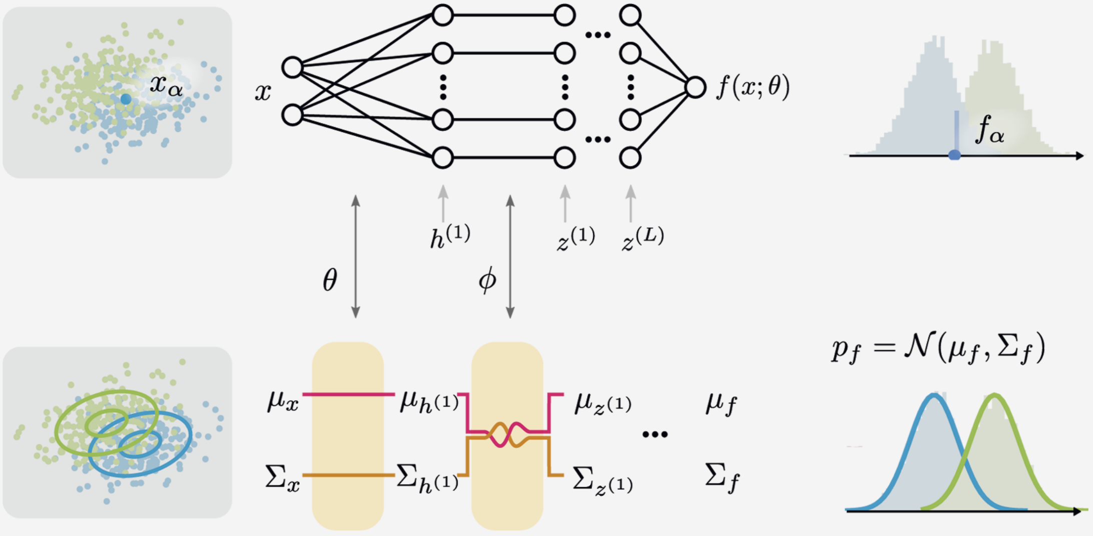


Mechanics of deep neural networks beyond the Gaussian limit

Kirsten Fischer

Information
Band / Volume 110
ISBN 978-3-95806-815-5

Mitglied der Helmholtz-Gemeinschaft

JÜLICH
Forschungszentrum

Schriften des Forschungszentrums Jülich
Reihe Information / Information Band / Volume 110

Forschungszentrum Jülich GmbH
Institute for Advanced Simulation (IAS)
Computational and Systems Neuroscience (IAS-6)

Mechanics of deep neural networks beyond the Gaussian limit

Kirsten Fischer

Schriften des Forschungszentrums Jülich
Reihe Information / Information Band / Volume 110

ISSN 1866-1777 ISBN 978-3-95806-815-5

Bibliografische Information der Deutschen Nationalbibliothek.
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der
Deutschen Nationalbibliografie; detaillierte Bibliografische Daten
sind im Internet über http://dnb.d-nb.de abrufbar.

Herausgeber und Vertrieb:	Forschungszentrum Jülich GmbH Zentralbibliothek, Verlag 52425 Jülich Tel.: +49 2461 61-5368 Fax: +49 2461 61-6103 zb-publikation@fz-juelich.de www.fz-juelich.de/zb
Umschlaggestaltung:	Grafische Medien, Forschungszentrum Jülich GmbH
Druck:	Grafische Medien, Forschungszentrum Jülich GmbH
Copyright:	Forschungszentrum Jülich 2025

Schriften des Forschungszentrums Jülich
Reihe Information / Information, Band / Volume 110

D 82 (Diss. RWTH Aachen University, 2025)

ISSN 1866-1777
ISBN 978-3-95806-815-5

Vollständig frei verfügbar über das Publikationsportal des Forschungszentrums Jülich (JuSER)
unter www.fz-juelich.de/zb/openaccess.

To my grandpa
- for giving me all
the abilities I needed

Author's List of Publications

The work presented in this thesis is in parts based on the following publications:

Decomposing neural networks as mappings of correlation functions
Kirsten Fischer, Alexandre René, Christian Keup, Moritz Layer, David Dahmen, and Moritz Helias
Published in Physical Review Research, 4(4) (2022): 043143.

Critical feature learning in deep neural networks
Kirsten Fischer*, Javed Lindner*, David Dahmen, Zohar Ringel, Michael Krämer, and Moritz Helias
Published in Proceedings of the 41st International Conference on Machine Learning, PMLR 235:13660-13690 (2024).
* These authors contributed equally.

Optimal signal propagation in ResNets through residual scaling
Kirsten Fischer, David Dahmen, and Moritz Helias
Preprint arXiv:2305.07715 (2023).

Author contributions are indicated at the beginning of the respective chapters.

Abstract

Current developments in the field of artificial intelligence and the neural network technology supersede our theoretical understanding of these networks. In the limit of infinite width, networks at initialization are well described by the neural network Gaussian process (NNGP): the distribution of outputs is a zero-mean Gaussian characterized by its covariance or kernel across data samples. Going to the lazy learning regime, where network parameters change only slightly from their initial values, the neural tangent kernel characterizes networks trained with gradient descent. Despite the success of these Gaussian limits for deep neural networks, they do not capture important properties such as network trainability or feature learning.

In this work, we go beyond Gaussian limits of deep neural networks by obtaining higher-order corrections from field-theoretic descriptions of neural networks. From a statistical point of view, two complimentary averages have to be considered: the distribution over data samples and the distribution over network parameters. We investigate both cases, gaining insights into the working mechanisms of deep neural networks.

In the former case, we study how data statistics are transformed across network layers to solve classification tasks. We find that, while the hidden layers are well described by a non-linear mapping of the Gaussian statistics, the input layer extracts information from higher-order cumulants of the data. The developed theoretical framework allows us to investigate the relevance of different cumulant orders for classification: On MNIST, Gaussian statistics account for most of the classification performance, and higher-order cumulants are required to fine-tune the networks for the last few percentages. In contrast, more complex data sets such as CIFAR-10 require the inclusion of higher-order cumulants for reasonable performance values, giving an explanation for why fully-connected networks perform subpar compared to convolutional networks.

In the latter case, we investigate two different aspects: First, we derive the network kernels for the Bayesian network posterior of fully-connected networks and observe a non-linear adaptation of the kernels to the target, which is not present in the NNGP. These feature corrections result from fluctuation corrections to the NNGP in finite-size networks, which allow the networks to adapt to the data. While fluctuations become larger near criticality, we uncover a trade-off between criticality and feature learning scales in networks as a driving mechanism for feature learning. Second, we study network trainability of residual networks by deriving the network prior at initialization. From this, we obtain the response function as a leading-order correction to the NNGP, which describes the signal propagation in networks. We find that scaling the residual branch by a hyperparameter improves signal propagation since it avoids saturation of the non-linearity and thus information loss. Finally, we observe a strong dependence of the optimal scaling of the residual branch on the network depth but only a weak dependence on other network hyperparameters, giving an explanation for the universal success of depth-dependent scaling of the residual branch.

Overall, we derive statistical field theories for deep neural networks that allow us to obtain systematic corrections to the Gaussian limits. In this way, we take a step towards a better mechanistic understanding of information processing and data representations in neural networks.

Zusammenfassung

Die aktuellen Entwicklungen im Bereich der künstlichen Intelligenz und neuronaler Netzwerke im Besonderen übersteigen unser theoretisches Verständnis dieser Netzwerke. Im Limes unendlicher Netzwerkbreite werden untrainierte Netzwerke bei Initialisierung als ein Gauß-Prozess, kurz NNGP, beschrieben: die Wahrscheinlichkeitsverteilung der Netzwerkausgaben ist eine Gaußverteilung mit Mittelwert Null, der durch seine Kovarianz charakterisiert wird. Der "Neural Tangent Kernel" beschreibt trainierte Netzwerke im sogenannten lazy learning Bereich, wo sich die Netzwerkparameter während des Trainings mit Gradientenabstieg nur geringfügig von ihren Anfangswerten unterscheiden. Trotz des Erfolgs dieser Gaußschen Charakterisierungen von tiefen neuronale Netze, erfassen diese wichtige Eigenschaften nicht, wie die Trainierbarkeit von Netzwerken oder das Lernen von Merkmalen aus den Daten.

In dieser Arbeit gehen wir über die Gaußschen Grenzwerte von tiefen neuronalen Netzwerken hinaus, indem wir Korrekturen höherer Ordnung mithilfe von feldtheoretischen Methoden bestimmen. Aus statistischer Sicht sind zwei komplementäre Beschreibungen von Bedeutung: die Wahrscheinlichkeitsverteilung der Datenpunkte und die Wahrscheinlichkeitsverteilung der Netzwerkparameter. Wir untersuchen beide Fälle und bekommen so unterschiedliche Einblicke in die Mechanismen tiefer neuronaler Netzwerke.

Im ersteren Fall untersuchen wir, wie die Datenstatistik durch die Netzwerkschichten transformiert wird um eine Klassifikationsaufgabe zu lösen. Wir stellen fest, dass die mittleren Netzwerkschichten durch eine nichtlineare Abbildung der Gaußschen Statistik gut beschrieben werden, während die erste Netzwerkschicht Informationen aus Kumulanten höherer Ordnung extrahiert. Die entwickelte Theorie ermöglicht es uns, die Bedeutung von Kumulanten verschiedener Ordnungen für die Klassifikation zu untersuchen: Bei MNIST ist die Gaußsche Statistik für den größten Teil der Klassifizierungsleistung verantwortlich und Kumulanten höherer Ordnung sind notwendig, um die Netzwerke für zusätzliche Prozente anzupassen. Im Gegensatz dazu erfordern komplexere Datensätze wie CIFAR-10 die Einbeziehung von Kumulanten höherer Ordnung. Dies könnte erklären, warum feedfoward Netzwerke im Vergleich zu Faltungsnetzwerken unterdurchschnittliche Ergebnisse liefern.

Im letzteren Fall untersuchen wir zwei verschiedene Aspekte: Erstens bestimmen wir die Kovarianzen für den Bayes'schen Netzwerk-Posterior von feedforward Netzwerken und stellen eine nichtlineare Anpassung der Kernel an den Zielwert fest, was beim NNGP nicht passiert. Diese Korrekturen der Kovarianzen resultieren aus Fluktuationskorrekturen des NNGP in Netzwerken endlicher Netzwerkbreite, was es den Netzwerken erlaubt sich an die Daten anzupassen. Während Fluktuationen in der Nähe der Kritikalität größer werden, entdecken wir einen Trade-off zwischen Kritikalität und Skalen in Netzwerken als treibenden Mechanismus für das feature learning. Zweitens untersuchen wir die Trainierbarkeit von residuellen Netzwerken, indem wir den Netzwerkprior bei der Initialisierung bestimmen. Daraus erhalten wir die Antwortfunktion als Korrektur führender Ordnung des NNGP, die die Signalpropagation in Netzwerken beschreibt. Wir stellen fest, dass die Skalierung des residuellen Netzwerkzweigs durch einen Hyperparameter die Signalpropagation im Netzwerk verbessert, da sie eine Sättigung der Nichtlinearität und damit einhergenden Informationsverlust vermeidet. Schließlich beobachten wir eine starke Abhängigkeit der optimalen Skalierung des residuellen Netzwerkzweigs von der Netzwerktiefe, aber nur eine schwache Abhängigkeit von anderen Netzwerkhyperparametern, was den breiten Erfolg der tiefenabhängigen Skalierung des residuellen Netzwerkzweigs erklärt.

Insgesamt bestimmen wir statistische Feldtheorien für tiefe neuronale Netzwerke, mithilfe welcher wir systematische Korrekturen zu den Gaußschen Beschreibungen neuronaler Netzwerke berechnen. Auf diese Weise machen wir einen Schritt hin zu einem besseren mechanistischen Verständnis der Informationsverarbeitung und der Datenrepräsentation in neuronalen Netzwerken.

Acknowledgements

It takes a village to complete a doctoral thesis. I got incredibly lucky in that I got to be part of not a single but multiple villages. My deepest gratitude goes to all of them for accompanying me on this journey.

First and foremost, I thank my thesis advisor Prof. Moritz Helias: for all your support, for your incredibly deep understanding of physics paired with the willingness and patience to teach me, and for your kind guidance. I thank David Dahmen for always asking the relevant questions, for being to the point, and for having a sympathetic ear for me. I thank Prof. Michael Kraemer for his curiosity and being my second advisor. I am grateful to our secretaries Saskia, Janine, and Petra for being the heart and support of the institute. I am grateful to my wonderful colleagues at IAS-6, filling the past four years with so much life: I thank Heather for long conversations and fun adventures. I thank Jasper for all the shared interests outside of work. I thank Michael Dick for being my partner in student issues, thesis writing, etc. I thank Claudia for her infinite supply of jokes. I thank Peter for always seeing the glass half-full. I thank Jonas for his calm in any situation. I thank Javed for making any meeting both fun and productive. This is just to name a few of all my amazing colleagues and the support they have provided me with. Further, I thank Prof. Markus Diesmann and Prof. Sonja Grün for putting together this wonderful institute.

I am grateful to Prof. Zohar Ringel for giving me the opportunity to stay in his lab, many intense discussions, and providing me with new perspectives. I thank Jan Bauer for easing my arrival, being a friend in a new place and beyond. I thank my Israeli colleagues Noa, Itay, and Lior: for answering all my stupid questions and being the best coffee crew, also outside of the office. I thank the amazing people from my Hebrew class for many learnings and fun outings after class.

I am grateful to Gitta Doebert for bringing to life the wonderful TandemDok program, of which I got to be a part of. I thank my mentor for all the discussions and valuable inputs. I am grateful to Anna for our monthly coaching sessions and always having a sympathetic ear for me. I am grateful for our amazing mentoring group of incredibly capable and warm-hearted doctoral students, all our laughs, all our discussions and all the advice during our trips together. You are truly the soul of this program.

Most importantly, I thank my best friend for knowing me inside-out, knowing when to just listen and when to provide me with advice, and for accompanying me through all of my academic journey. I thank my brother for being my very anti-thesis and always sharing his infinite knowledge with me. I thank my parents for letting me find my own path, always being the serene support in the background. I thank my family for enduring yet another explanation of what I am actually doing. Finally, I thank my friends, all of them pursuing their amazing paths while supporting and cheering each other on.

Funding

This thesis received funding from the German Federal Ministry for Education and Research (BMBF grant 01IS19077A to Juelich), the Excellence Initiative of the German federal and state governments (ERS PF-JARA-SDS005), and the Helmholtz Association Initiative and Networking Fund under Project No. SO-092 (Advanced Computing Architectures, ACA). This thesis was supported by a fellowship of the German Academic Exchange Service (DAAD) and a short-term research grant by the Minerva foundation.

Contents

Chapter 1

Introduction

Recent years have shown a great success of deep neural networks; from image recognition (Krizhevsky, Sutskever, and Hinton, 2012), playing Go (Silver et al., 2016) to predicting protein structures (Abramson et al., 2024). Already today we interact with such systems on a daily basis via voice command for smart phones and other smart devices, refined internet search, or personalized algorithms shaping what we are exposed to on social media platforms. One of the most-prominent examples are large language models such as Chat-GPT (Vaswani et al., 2017; Radford et al., 2018), which has become ubiquitous at a rapid pace.

Despite the success of these technologies, the development in this field supersedes our theoretical understanding of such systems. Starting from the simple but already non-trivial perceptron (Rosenblatt, 1958) to foundation models (Bommasani et al., 2022), network architectures have become ever more complex. Due to their black-box character combined with inferring information directly from the given data sets, unwanted behaviors may occurr: In image recognition, neural networks use features that work well on the given data set but do not generalize well as they do not learn the desired features (Ribeiro, Singh, and Guestrin, 2016). Large language models are prone to hallucinating information and in particular references (Ji et al., 2023), but fail at simple reasoning tasks (Nezhurina et al., 2024). For safe employment of these technologies in sensitive fields such as autonomous driving (Hofmarcher et al., 2019) or automated medical diagnosis (Al Kuwaiti et al., 2023), we require a better understanding of their inner mechanics as well as the learned features. Such insights can then also be used to guide future development of network architectures in a principled way, as opposed to current empirical approaches, thereby reducing both compute and energy costs (Yang and Shami, 2020).

Shedding light onto the inner mechanics of the black box, also referred to as explainable AI (Gunning et al., 2019), is a field of research with various approaches from different disciplines: Shapley values, which are originally utilized in game theory, measure the contribution of different input features to the network output (Aas, Jullum, and Løland, 2021). Deep Taylor decomposition determines the relevance of neuron activations by decomposing the network output using Taylor expansions of the non-linearity (Montavon et al., 2017). For convolutional networks, the learned filters can be visualized and linked to detecting different image elements (Zeiler and Fergus, 2014). In the context of generative models, invertible neural networks allow sampling from the learned distribution and thus learned features (Ardizzone et al., 2019).

From a physics point of view, neural networks can be viewed as complex systems with a large number of degrees of freedom. In practice, the realization of individual network parameters is less relevant, but one is interested in the collective behavior of the constituents of the system, e.g. neurons, measured by scalar quantities such as the generalization performance. Such emergent characteristics can be studied using methods from statistical physics. In recent decades, much research has been done at this intersection between statistical physics and machine learning: from applying replica methods for studying learning curves (Mézard, Parisi, and Virasoro, 1987; Loureiro et al., 2022), dynamic mean-field theory for studying scaling properties of neural networks (Sompolinsky and Zippelius, 1982; Bordelon and Pehlevan, 2023; Bordelon et al., 2024) to uncovering phase transitions in their learning dynamics (Baldassi et al., 2022; Cui et al., 2024).

In the limit of infinite network width, neural networks can be characterized as a centered Gaussian process, which is referred to as the Neural Network Gaussian process (NNGP), and the covariance over network samples, also called NNGP kernel, emerges as the relevant order parameter. At initialization, the network parameters are independent and identically distributed, so that the signal in linear network layers becomes Gaussian due to the central limit theorem. This characterization was first found by Neal (1996) for networks with a single hidden layer; Lee et al. (2018) extended this result to deep neural networks. However, Bayesian inference with the NNGP kernel corresponds to training the output layer only and leaves a generalization gap to finite-width networks trained with gradient based methods (Lee et al., 2019; Yang, 2019).

Another Gaussian limit involves training all network layers: the Neural Tangent kernel (NTK) is the outer product of gradients across data samples and describes the evolution of the network during gradient descent (Jacot, Gabriel, and Hongler, 2018). For the mean-squared error loss function, the trained network may again be characterized as a Gaussian process with a different mean and covariance than the NNGP (Lee et al., 2019). However, in the NTK limit, network parameters change only negligibly and thus the network is linearized with respect to the network parameters (Lee et al., 2019), which corresponds to the lazy regime that was first introduced by (Chizat, Oyallon, and Bach, 2019).

Despite the accomplishments of these Gaussian theories, they ultimately are free theories that do not capture interactions within the network and are thus limited in which phenomena of the system they describe. An analogy from physics is the ideal gas, which yields the general gas equation, but requires the interaction between gas particles to approximately describe condensation phenomena with the van der Waals equation (Boltzmann, 1964). The decoupling of different neurons in the network, which leads to a Gaussian theory, results directly from the infinite-width limit; interactions become relevant in finite-size neural networks and and thus require to go beyond the Gaussian limit: image recognition using convolutional networks is based on non-Gaussian statistics of the data (Ingrosso and Goldt, 2022; Refinetti, Ingrosso, and Goldt, 2023); grokking refers to rapid changes of the generalization performance after longer, constant phases of poor generalization (Power et al., 2022; Rubin, Seroussi, and Ringel, 2024); in the feature learning regime network parameters adapt strongly to the data and thus violate the stochastic independence assumption of the central limit theorem (Chizat, Oyallon, and Bach, 2019; Geiger et al., 2020). In this thesis we will use statistical field theories to study neural networks and in particular to systematically determine corrections to these Gaussian theories, which shed light onto the inner mechanics of deep neural networks.

For studying the emergent phenomena in neural networks, there are two components to their statistical nature: the distribution of data samples, which drives network training, and the distribution of network parameters, which at initialization influences learning dynamics and after training determines the learned features. We will study each of these settings separately, asking how non-Gaussian corrections in finite-size networks determine their inner mechanics.

One key question relating to the first setting is how neural networks on supervised classification tasks fit their training data set and also generalize well, which is often explained by simplicity biases of the networks (Lin, Tegmark, and Rolnick, 2017; Kalimeris et al., 2019; Bowman and Montufar, 2022). Valle-Perez, Camargo, and Louis (2019) argue that neural networks are biased towards learning simple functions that avoid overfitting the data. While linear networks first learn relevant target directions (Krogh and Hertz, 1992; Saxe, Mcclelland, and Ganguli, 2014; Advani, Saxe, and Sompolinsky, 2020), non-linear networks first learn linear functions before finding more complex solutions (Saad and Solla, 1995; Mei, Montanari, and Nguyen, 2018). Further, Rahaman et al. (2019) find a spectral simplicity bias where networks initially learn lower frequencies of the target function. While these works focus on the implemented mapping, we are interested in how this mapping extracts and processes information that is encoded in the data statistics. By shifting the perspective to the transformation of data distributions, we are able to shed light on the functional properties of different network components. We find a simplicity bias in fully-connected networks towards primarily learning Gaussian statistics, which receives corrections in the input layer to fine-tune the networks for final generalization performance.

In the second setting, NNGP and NTK describe networks at initialization and in the lazy learning regime, respectively, well. However, they do not capture feature learning, where network parameters adapt strongly to the data; this regime typically outperforms networks in the lazy regime (Novak et al., 2019; Lee et al., 2020; Geiger et al., 2020; Petrini et al., 2022). One line of research proposes kernel rescaling as the adaptation mechanism, where the NNGP kernel receives a scalar factor so that the predictor stays the same but its variance changes; previous works investigate linear networks (Li and Sompolinsky, 2021; Hanin and Zlokapa, 2023) as well as different non-linear network architectures (Pacelli et al., 2023; Aiudi et al., 2023; Baglioni et al., 2024). Other works such as (Seroussi, Naveh, and Ringel, 2023) instead find kernel adaptation beyond rescaling towards the relevant task directions, corresponding to richer mechanics of the networks. We will investigate the question of kernel adaptation, observing a non-linear adaptation towards the target that results from fluctuation corrections to the NNGP limit.

Finding network hyperparameters that lead to successfull training and yield high generalization performance has become even more relevant as the model size has increased significantly in recent years, imbuing large compute demands for hyperparameter search (Yang and Shami, 2020). Methods include Bayesian optimization (Snoek, Larochelle, and Adams, 2012), multi-fidelity methods (Falkner, Klein, and Hutter, 2018) or genetic algorithms (Itano, Sousa, and Del-Moral-Hernandez, 2018). Schoenholz et al. (2017) propose to initialize networks at the edge-of-chaos in hyperparameter space, where networks transition from an ordered to a chaotic phase regarding their signal dynamics and as a result information propagation scales in the network diverge, so that signals and gradients can propagate to great depths. By investigating the scales of network training, Yang et al. (2021) propose μP-scaling for initializing feed-forward networks, which allows zero-shot hyperparameter transfer from smaller to larger network models. Following these lines of research, we study signal propagation and trainability for residual networks and investigate the effect of scaling the residual branch as an additional hyperparameter.

This thesis is organized as follows: In Chap. 2, we study the transformation of the data distribution by fully-connected, feed-forward networks and how these networks utilize information encoded in the data distribution to solve a given classification task. We then turn towards the distribution over network parameters and first investigate feature learning in fully-connected, feed-forward networks by computing the posterior network kernels in Chap. 3, analyzing in detail how non-Gaussian corrections drive kernel adaptation. In Chap. 4 we consider residual networks and study the residual scaling of these network and its relation to signal propagation. Finally, we discuss our contributions towards understanding the inner mechanics of neural networks beyond Gaussian limits in Chap. 5 and provide an outlook on future directions of research.

Chapter 2

Decomposing neural networks as mappings of correlation functions

This chapter, App. A, and parts of the discussion are based on the following publication:

Kirsten Fischer, Alexandre René, Christian Keup, Moritz Layer, David Dahmen, and Moritz Helias. "Decomposing neural networks as mappings of correlation functions." Physical Review Research, 4(4) (2022): 043143.

Author contributions
Under the supervision of David Dahmen and Moritz Helias, the author worked on all parts of the above publication presented in this chapter. The author contributed to the general formalism and performed the corresponding numerical experiments. All authors contributed to writing the manuscript. The idea of tracing the transformation of cumulants in feed-forward networks is also present in the author's master thesis (Fischer, 2020); however, the inclusion of higher-order cumulants was explored in a preliminary fashion. In this work, we properly include higher-order cumulants in Sec. 2.5.4 as well as distinguish between Gaussian statistics and higher-order cumulants for MNIST in Sec. 2.5.3. Further, we extended the theoretical results to the practically more relevant ReLU non-linearity and provide numerical experiments. Finally, we add experiments on CIFAR-10.

2.1 Introduction

Neural networks are complex systems with a large number of degrees of freedom. Their statistical properties depend on both the distribution of network parameters and the joint distribution of input data and labels. While we consider the former case in the next two chapters, we here focus on how neural networks transform the distribution of input data for a fixed set of network parameters. To solve a task, the network aims to match a certain target distribution on the output side. We study how networks transform the input statistics to match the target and in particular which statistical quantities are most relevant for doing so.

The main contributions of this chapter are:

- in fully-connected feed-forward networks, the principal part of the computation performed by the network is captured by the mapping of Gaussian statistics in each network layer;
- for low-dimensional data, higher-order cumulants of the input data are primarily extracted and processed by the input layer;
- classifying image data relies heavily on higher-order cumulants, explaining why fully-connected networks fail while convolutional networks that involve sparse structured weight matrices and thus higher-order cumulants in each layer succeed.

2.2 Setup

We study fully-connected feed-forward neural networks with L layers and N_l neurons in layer l and a linear readout layer as shown in Fig. 2.1(a). In each layer $l = 1, \ldots, L$ we have a linear mapping

$$z_i^{(l)} = \sum_{j=1}^{N_{l-1}} W_{ij}^{(l)} y_j^{(l-1)} + b_i^{(l)}, \tag{2.1}$$

parameterized by a weight matrix $W^{(l)} \in \mathbb{R}^{N_l \times N_{l-1}}$ and bias vector $b^{(l)} \in \mathbb{R}^{N_l}$. Then we apply a non-linear activation function ϕ in a pointwise manner

$$y_i^{(l)} = \phi\big(z_i^{(l)}\big) = \phi\left(\sum_{j=1}^{N_{l-1}} W_{ij}^{(l)} y_j^{(l-1)} + b_i^{(l)}\right). \tag{2.2}$$

We write $y^{(0)} = x \in \mathbb{R}^{D_{\text{in}}}$ for input data of dimension $N_0 = D_{\text{in}}$. The linear readout layer yields the network output $y \in \mathbb{R}^{D_{\text{out}}}$ with $N_{L+1} = D_{\text{out}}$, so that $y_i = z_i^{(L+1)}$. The full network mapping $y = g(x;\theta)$ results from iterating over network layers; it depends on parameters $\theta := \{W^{(l)}, b^{(l)}\}_{l=1,\dots,L+1}$.

All network parameters are initialized randomly by drawing from i.i.d. centered Gaussians $W_{ij}^{(l)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, \sigma_w^2/N_{l-1}\right)$ and $b_i^{(l)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, \sigma_b^2\right)$. We scale the covariance of $W_{ij}^{(l)}$ such that the covariance of $z^{(l)}$ is independent of the layer width N_l.

2.3 Theoretical Background

We first provide some background on empirical risk minimization that forms the basis of this study. Then we introduce the concept of cumulants as a parametrization of probability distributions that will form the central object of our analysis.

2.3.1 Empirical risk minimization

We here adapt the presentation of empirical risk minimization to classification problems that are studied in this chapter. The existence of a joint distribution $p(x,t)$ of data samples x and class labels t that is the same for training and evaluation is the essential premise behind classification (Bishop, 2006). According to Bayes' theorem, the distribution of the input data can be regarded as a mixture model $p(x) = \sum_t p(t)\, p(x|t)$. The network's task is then to implement a mapping $g : x \mapsto y$ that minimizes the expectation value of a loss function $\ell(y,t)$ that depends on both network outputs $y = g(x;\theta)$ and labels t.

Consequently, for each label t a mapping of probability distributions is induced by the network mapping

$$p(x|t) \mapsto p(y|t;\theta) = \int \delta(y - g(x;\theta))\, p(x|t)\, dx, \tag{2.3}$$

where $\delta(\circ)$ refers to the Dirac Delta distribution. The output distribution over all labels t is then given by the weighted sum $p(y) = \sum_t p(t)\, p(y|t;\theta)$. Ideally, the network output y is exactly the true label t; the target distribution is thus $p(y|t) = \delta(y - t)$.

Any training algorithm aims to minimize the expected loss or risk functional

$$\mathcal{L}(\theta) = \langle \ell(y,t) \rangle_{y|\theta} = \sum_t p(t)\, \langle \ell(y,t) \rangle_{y|t;\theta}, \tag{2.4}$$

where the expectation value $\langle \dots \rangle_{y|t;\theta}$ refers to the class-conditional output distributions $p(y|t;\theta)$ (Vapnik, 1992). However, in general the mixture components of the input distribution $p(x|t)$ and the induced class-conditional output distributions $p(y|t;\theta)$ are unknown. Therefore, one instead minimizes the empirical loss or risk

$$\mathcal{L}_{\text{emp}}(\theta) = \frac{1}{P}\sum_{\alpha=1}^{P} \ell(g(x_\alpha;\theta), t_\alpha), \tag{2.5}$$

that is being evaluated for a training set $\{(x_\alpha, t_\alpha)\}_\alpha$ of size P with sample indices α. The empirical risk minimization principle makes the following assumption: the mapping $g(\circ;\theta^*)$ that minimizes the empirical risk $\theta^* = \text{argmin}_\theta\, \mathcal{L}_{\text{emp}}(\theta)$ yields an expected risk $\mathcal{L}(\theta^*)$ that is close to its minimum $\min_\theta R(\theta)$ (Vapnik, 1992).

2.3.2 Parameterizing probability distributions by cumulants

One way to think of neural networks is as complex systems that generate interactions between data components. Rather than focusing on the probability distributions themselves, generating functions of moments or cumulants are frequently used in order to study such systems. Since cumulants are additive when it comes to the addition of independent variables, they typically offer a useful parameterization of probability distributions. This makes it easier to derive equations for the transformation of statistics between layers. These concepts are widely used in mathematical statistics and statistical physics.

The network mapping $g : x \mapsto y$ connects the cumulant-generating function of network outputs y to the distribution of the inputs x:

$$\mathcal{W}_{y|t;\theta}(j) = \ln \left\langle \exp\left(j^\mathsf{T} y\right)\right\rangle_{y|t;\theta} \tag{2.6}$$

$$= \ln \left\langle \exp\left(j^\mathsf{T} g(x;\theta)\right)\right\rangle_{x|t}. \tag{2.7}$$

The distribution of inputs is expected to be different for different classes, therefore the cumulant-generating function appears per class t. We are interested in the class-conditional output cumulant of order n which is given by

$$G^{(n)}_{y|t;\theta} = \left.\frac{d^n \mathcal{W}_{y|t;\theta}(j)}{dj^n}\right|_{j=0}. \tag{2.8}$$

In general, $G^{(n)}_{y|t;\theta}$ can be related to the input cumulants $G^{(n')}_{x|t;\theta}$ by evaluating Eq. (2.7), but the fact that the network mapping $g(x;\theta)$ is defined by the iterations in Eq. (2.2) poses difficulties for doing so. Deep neural networks are effective as universal function approximators due to their iterative non-linear character, but this also complicates the analysis of networks with respect to their data-processing properties. However, by examining each layer separately, we can study how cumulants change from input to output.

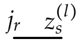
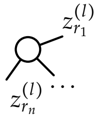
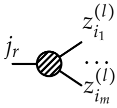
Meaning	Algebraic term	Graphical representation
External line	$j_r \, \delta_{rs}$	
Cumulant vertex with n internal lines	$G^{(n)}_{z^{(l)},\,(r_1,\ldots,r_n)}$	
ϕ-vertex with m internal lines and one external line	$j_r \, \frac{1}{m!} \, \phi^{(m)}\big\rvert_{z^{(l)}=0}$ $\times \, \delta_{ri_1} \, \ldots \, \delta_{ri_m}$	

Table 2.1: Diagrammatic language for performing a perturbative expansion of the cumulant-generating function $\mathcal{W}_{y^{(l)}}(j)$ for post-activations $y^{(l)} = \phi(z^{(l)})$.

Since pre-activations $z^{(l)}$ result from a linear transformation, the cumulant-generating function of pre-activations $z^{(l)}$ in layer l is trivially connected to the cumulant-generating function of post-activations $y^{(l-1)}$ of layer $l-1$ as

$$
\begin{aligned}
\mathcal{W}_{z^{(l)}}(j) &= \ln \left\langle \exp\left(j^{\mathsf{T}} z^{(l)}\right) \right\rangle_{z^{(l)}} \\
&= \ln \left\langle \exp\left(j^{\mathsf{T}} W^{(l)} \, y^{(l-1)} + j^{\mathsf{T}} b^{(l)}\right) \right\rangle_{y^{(l-1)}} \\
&= \mathcal{W}_{y^{(l-1)}}\left(\left(W^{(l)}\right)^{\mathsf{T}} j\right) + j^{\mathsf{T}} b^{(l)},
\end{aligned}
\tag{2.9}
$$

which for the first-order cumulant gives

$$
G^{(1)}_{z^{(l)}} = W^{(l)} \, G^{(1)}_{y^{(l-1)}} + b^{(l)},
\tag{2.10}
$$

and for second- and higher-order cumulants gives

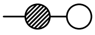
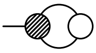
$$
G^{(n)}_{z^{(l)},\,(r_1,\ldots,r_n)} = \sum_{s_1,\ldots,s_n} W^{(l)}_{r_1 s_1} \ldots W^{(l)}_{r_n s_n} \, G^{(n)}_{y^{(l-1)},\,(s_1,\ldots,s_n)}.
\tag{2.11}
$$

Thus, for each index r_k of the resulting cumulant we contract one index s_i with one factor $W^{(l)}_{r_k s_i}$. Consequently, cumulants of pre-activations $z^{(l)}$ are generated from cumulants of post-activations $y^{(l-1)}$ of the same order by a linear tensor transformations.

The pre-activations $z^{(l)}$ in layer l result from the corresponding post-activations $y^{(l)}$ by applying the non-linear activation function ϕ in a pointwise manner

$$
\mathcal{W}_{y^{(l)}}(j) = \ln \left\langle \exp\left(j^{\mathsf{T}} y^{(l)}\right) \right\rangle_{y^{(l)}} = \ln \left\langle \exp\left(j^{\mathsf{T}} \phi(z^{(l)})\right) \right\rangle_{z^{(l)}}.
\tag{2.12}
$$

In general, the cumulant-generating function of the post-activations $y^{(l)}$ does not have an exact analytical solution. One can obtain approximate expression for the appearing average by using a perturbative expansion that commonly appears in statistical physics (Helias and Dahmen, 2020). We here employ such an expansion in the following way: by replacing $\phi(z^{(l)})$ with its Taylor expansion $\sum_m \frac{\phi^{(m)}|_{z^{(l)}=0}}{m!}(z^{(l)})^m$ in Eq. (2.12) and treating non-linear terms ($m > 1$) as perturbations, we obtain cumulants $G^{(n)}_{y^{(l)}}$ as series of Feynman diagrams composed of the graphical elements shown in Tab. 2.1. For instance, we get the following diagrams for the first layer's mean:

$G^{(1)}_{y^{(1)},i} =$  $+$  $+ \ldots$

$$= \frac{\phi^{(1)}\big|_{x=0}}{1!} G^{(1)}_{x,i} + \frac{\phi^{(2)}\big|_{x=0}}{2!} G^{(2)}_{x,ii} + \ldots \tag{2.13}$$

These expressions appearing in the perturbation expansion involve two types of factors that are represented with two types of vertices: hatched circles with one external line j that originate from Taylor coefficients $\frac{\phi^{(m)}\big|_{z^{(l)}=0}}{m!}$ of the non-linearity, and empty circles with internal lines, representing cumulants $G^{(n)}_{z^{(l)}}$ of pre-activations $z^{(l)}$.

To obtain the cumulant $G^{(n)}_{y^{(l)}}$ of the post-activations $y^{(l)}$ of order n, we construct all diagrams with n external lines. External lines appear on both hatched and cumulant vertices. Moreover, external lines cannot be connected to one another; they must always be connected to a cumulant vertex. Finally, from the linked cluster theorem follows that only connected diagrams contribute to cumulants. Symmetries within diagrams imply that they appear repeatedly in the perturbation expansion, which is accounted for by so-called symmetry factors. To obtain these combinatorial prefactors of the generated diagrams, all permutations of indices $(r_1, \ldots, r_n)$ for both internal and external lines are determined (for more details, see (Helias and Dahmen, 2020)).

There are two key benefits to using this perturbative approach for determining the cumulants $G^{(n)}_{y^{(l)}}$ of the post-activations $y^{(l)}$: Firstly, it offers a principled way to include higher-order cumulants, thereby going beyond Gaussian statistics. Secondly, the information transfer from cumulants $G^{(n)}_{z^{(l)}}$ of the pre-activations $z^{(l)}$ to cumulants $G^{(m)}_{y^{(l)}}$ of the post-activations $y^{(l)}$ can be visually represented by the diagrammatic language in Tab. 2.1.

The diagrammatic language requires that the activation function ϕ has a convergent representation as a Taylor series. If this is not the case, e.g. for non-differentiable functions such as ReLU, we can use a Gram-Charlier expansion of the probability distribution $p(z^{(l)})$ to approximate the expectation value in Eq. (2.12): the computation then simplifies to a weighted sum of Gaussian integrals, which can be calculated either analytically (see App. A.1 for ReLU as an example) or numerically.

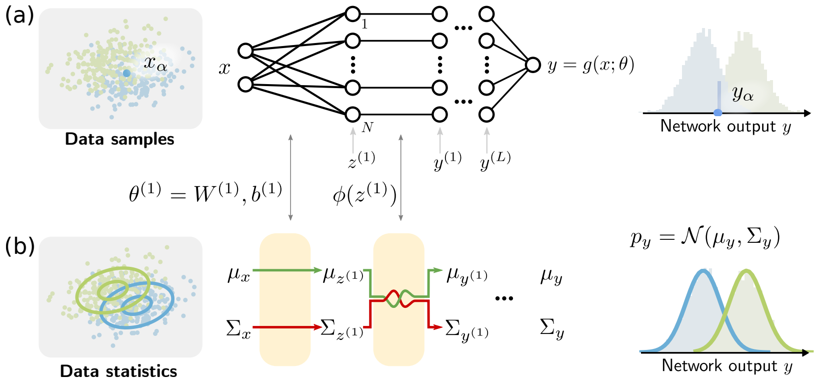


Figure 2.1: (a) Sample-based network analysis considers a single output y_α for each data sample x_α as it flows through the network. A linear transformation $(W^{(l)}, b^{(l)})$ precedes the componentwise application of a non-linearity ϕ in each layer. (b) Network analysis grounded in data statistics takes into account the network's transformation of the data distribution $p(x)$. In each layer we parameterize the intermediate distribution by its cumulants; the mean μ and the covariance Σ are the lowest-order and most-relevant cumulants for wide networks. While μ and Σ are transformed independently by the linear step, a non-trivial interaction between the two is caused by the non-linearity ϕ.

2.4 Theory

In this section we derive how cumulants of the input data are iteratively transformed by deep neural networks, as shown in Fig. 2.1(b), and how this shapes the expected loss.

2.4.1 Cumulants drive network training

We here study the relation of the expected loss in Eq. (2.4) with cumulants of the data. For general loss functions $\ell(y,t)$, the expected risk $R(\theta)$ depends on the class labels t and the class-conditional cumulants $G^{(n)}_{y|t;\theta}$ of arbitrary orders n:

$$\begin{aligned}\mathcal{L}(\theta) &= \sum_t \int \mathrm{d}y\, \ell(y,t)\, p(y|t;\theta) \\ &=: \sum_t \sigma_t(\{G^{(n)}_{y|t;\theta}\}_n) \\ &=: \sigma(\{\{G^{(n)}_{y|t;\theta}\}_n; t\}_t).\end{aligned}$$

For the commonly used mean squared error $\ell_{\mathrm{MSE}}(y,t) = \|y-t\|^2$, the expected risk $\mathcal{L}(\theta)$ is a function of solely the mean μ^t_y and variance Σ^t_y of outputs of each class t

and given by

$$\mathcal{L}_{\mathrm{MSE}}(\{\mu_y^t, \Sigma_y^t; t\}_t) = \sum_t p(t) \left(\operatorname{tr} \Sigma_y^t + \|\mu_y^t - t\|^2\right). \tag{2.14}$$

From this follows directly that network training tries to match class means and labels due to the term $\|\mu_y^t - t\|^2$, while reducing the variance of each class's output in $\operatorname{tr}\Sigma_y^t$. Although similar in structure, Eq. (2.14) differs from the bias-variance decomposition (Kohavi and Wolpert, 1996): we take the expectation over the input distribution $p(x)$ directly instead of the expectation over finite data sets of fixed size.

As a direct consequence, it is only mean and covariance of the output layer that drive network training, making these the important statistics. There are two main consequences from this result: 1.) Only non-Gaussian statistics that occur in network layers prior to the output layer can affect the first two cumulants in the output layer, which in turn influences the learnt information processing. 2.) Non-Gaussian statistics generated in the last layer may result from previous layers acting on higher-order cumulants, but they do not have a specific functional purpose in network training.

Overall, we conclude that to understand the network mapping it is sufficient to understand how the Gaussian cumulants (μ_y^t, Σ_y^t) of the output arise from the input data. Thus, network training and the resulting information processing by the trained network are closely related to the ways in which cumulants of the input data are transformed by the network across network layers.

2.4.2 Transformation of cumulants by the network

In the previous section, we showed that the only relevant cumulants of the network output are mean and covariance. We now derive how these depend on the cumulants of previous layers and in particular the input distribution. As we derived in Eq. (2.9)-(2.11), the linear transformation in every layer l yields a one-to-one relation between mean and covariance of pre-activations $z^{(l)}$ and post-activations $y^{(l)}$ (see Fig. 2.1(b)) that is given by

$$\mu_{z^{(l)}} = W^{(l)}\,\mu_{y^{(l-1)}} + b^{(l)}, \quad \Sigma_{z^{(l)}} = W^{(l)}\,\Sigma_{y^{(l-1)}}(W^{(l)})^{\mathsf{T}}. \tag{2.15}$$

In general, the non-linear activation function $\phi : z^{(l)} \mapsto y^{(l)}$ yields a non-Gaussian distribution of post-activations $y^{(l)}$ that thus involves cumulants of higher orders in the pre-activations $z^{(l)}$ (see Sec. 2.3.2)

$$\mu_{y^{(l)}} = \left.\frac{d\mathcal{W}_{y^{(l)}}(j)}{dj}\right|_{j=0} = \langle \phi(z^{(l)}) \rangle_{z^{(l)}}, \tag{2.16}$$

$$\Sigma_{y^{(l)}} = \left.\frac{d^2\mathcal{W}_{y^{(l)}}(j)}{dj\,dj^{\mathsf{T}}}\right|_{j=0} = \langle \phi(z^{(l)})\,\phi(z^{(l)})^{\mathsf{T}} \rangle_{z^{(l)}} - \mu_{y^{(l)}}\mu_{y^{(l)}}^{\mathsf{T}}. \tag{2.17}$$

To make these equations tractable again, we utilize that due to the central limit theorem cumulants of order $n > 2$ are suppressed in the subsequent linear layer $y^{(l-1)} \mapsto z^{(l)}$: by initializing the weights independently, higher-order cumulants scale as $G^{(n)}_{z^{(l)},(i_1,\dots,i_n)} = \langle\!\langle z^{(l)}_{i_1} z^{(l)}_{i_2} \dots z^{(l)}_{i_n} \rangle\!\rangle \sim \mathcal{O}((N_{l-1})^{-\frac{n}{2}+1})$. Since we are also interested in the case of trained networks, we derive in App. A.3 sufficient conditions under which the Gaussian approximation continues to hold in the presence of weak correlations among network weights. We find that it is sufficient if network weights scale as $W \sim \mathcal{O}(N^{-\frac{1}{2}})$ and rows of network weights project to approximately orthogonal subspace of the covariance matrix from the upstream post-activations. These conditions go beyond the lazy learning regime, where weights change only slightly compared to their initialized values (Chizat, Oyallon, and Bach, 2019). Thus, for networks with i.i.d. initialization or networks that are trained and fulfill these sufficient conditions, expectations over pre-activations $\langle \dots \rangle_{z^{(l)}}$ can be taken with respect to Gaussian distributions $z^{(l)} \sim \mathcal{N}(\mu_{z^{(l)}}, \Sigma_{z^{(l)}})$. It follows that mean and covariance of post-activations then are non-linear functions of mean and covariance of pre-activations

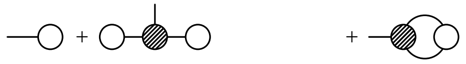
$$\mu_{y^{(l)}} = f_\mu(\mu_{z^{(l)}}, \Sigma_{z^{(l)}}), \quad \Sigma_{y^{(l)}} = f_\Sigma(\mu_{z^{(l)}}, \Sigma_{z^{(l)}}), \tag{2.18}$$

which due to their non-linear nature effectuate interactions between mean and covariance.

Taking the transformation of cumulants in linear and non-linear layers together and iterating backwards across network layers $l = L+1, L, \dots, 2$, we find that the information processing in the hidden network layers can be understood as an iterated, non-linear mapping of mean and covariance. For arbitrary activation functions ϕ, we can always compute the interaction functions f_μ and f_Σ numerically; for certain choices of activation functions, we can determine analytic expressions that allow for an analysis of the interactions between mean and covariance in the network. In Tab. 2.2, expressions for the interaction functins are given in the case of $\phi = \text{ReLU}$ and $\phi(z) = z + \epsilon z^2$.

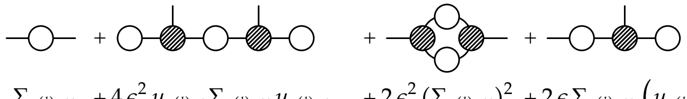
The quadratic activation function is minimally non-linear in the parameter ϵ and thus leads to especially interpretable interaction functions; the resulting interactions can be visualized and calculated with diagrams as follows

$\mu_{y^{(l)},i} =$



$$= \mu_{z^{(l)},i} + \epsilon\,(\mu_{z^{(l)},i})^2 + \epsilon\,\Sigma_{z^{(l)},ii}\,, \tag{2.19a}$$

$\Sigma_{y^{(l)},ij} =$



$$= \Sigma_{z^{(l)},ij} + 4\epsilon^2\,\mu_{z^{(l)},i}\,\Sigma_{z^{(l)},ij}\,\mu_{z^{(l)},j} + 2\epsilon^2\,(\Sigma_{z^{(l)},ij})^2 + 2\epsilon\,\Sigma_{z^{(l)},ij}\left(\mu_{z^{(l)},i} + \mu_{z^{(l)},j}\right). \tag{2.19b}$$

The most-right diagram that contributes to $\Sigma_{y^{(l)},ij}$ is transcribed into two terms that appear in brackets; these two terms correspond to the permutation of the indices (i, j) (see Sec. 2.3.2).

While network training generates correlations between weights and thereby breaks the independence of network weights that is required for the central limit theorem, both the independence assumption as well as the conditions derived for weakly correlated weights in App. A.3 are only sufficient and not necessary conditions for the Gaussian description of the network to hold. Instead, we empirically show in later section that tracing solely mean and covariance through the hidden network layers continues to be a useful approximation also for trained networks. Before going to these experiments, we study the information processing by the input layer.

2.4.3 Input layer extracts higher-order cumulants from data

The main difference between the input layer and hidden network layers is that the sum appearing in the definition of the pre-activations $z_i^{(1)} = \sum_{j=1}^{N_0} W_{ij}^{(1)} x_j + b_i^{(1)}$ is over the input dimension $N_0 = D_{\text{in}}$ instead of the network width N. Since the input dimension is fixed by the given task, we often have $D_{\text{in}} \ll N$. As we have discussed in the previous section, the higher-order cumulants of the pre-activations $z^{(1)}$ scale as $G_{z^1}^{(n>2)} \propto D_{\text{in}}^{1-\frac{n}{2}}$. In consequence, we need to take higher-order cumulants into account for the interaction functions in the input layer

$$\mu_{y^{(1)}} = h_\mu(\{G_{z^{(1)}}^{(n)}\}_n), \quad \Sigma_{y^{(1)}} = h_\Sigma(\{G_{z^{(1)}}^{(n)}\}_n)\,. \tag{2.20}$$

Mean and covariance of the post-activations $y^{(1)}$ are then passed on through the subsequent layers of the network as discussed in the previous sections.

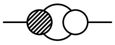
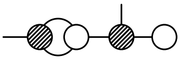
While we cannot calculate the interaction functions h_μ and h_Σ exactly for arbitrary activation functions ϕ, we can approximate these systematically: for non-differentiable functions (see App. A.1 for ReLU as an example) one can use a Gram-Charlier expansion or for differentiable functions one can use the diagrammatic techniques discussed in Sec. 2.3.2.

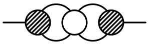
For polynomial activation functions, we are able to calculate the interaction functions h_μ and h_Σ exactly. We illustrate this here for the quadratic activation function $\phi(z) = z + \epsilon z^2$: the mean remains the same as in Eq. (2.19a) since it does not receive a correction from $G_{z^{(1)}}^{(n>2)}$ while the covariance in Eq. (2.19b) receives additional

Non-linearity	Interaction function
$\phi(z) = \mathrm{ReLU}(z)$	$f_{\mu,i} = \frac{\sqrt{\Sigma_{z,ii}}}{\sqrt{2\pi}} \exp\left(-\frac{\mu_{z,i}^2}{2\Sigma_{z,ii}}\right) + \frac{\mu_{z,i}}{2}\left(1 + \mathrm{erf}\left(\frac{\mu_{z,i}}{\sqrt{2\Sigma_{z,ii}}}\right)\right)$
	$\begin{aligned} f_{\Sigma,ii} &= \frac{\Sigma_{z,ii}}{2}\left(1 + \mathrm{erf}\left(\frac{\mu_{z,i}}{\sqrt{2\Sigma_{z,ii}}}\right)\right) + \frac{\mu_{z,i}^2}{4} \\ &\quad - \left(\frac{\sqrt{\Sigma_{z,ii}}}{\sqrt{2\pi}} \exp\left(-\frac{\mu_{z,i}^2}{2\Sigma_{z,ii}}\right) + \frac{\mu_{z,i}}{2} \mathrm{erf}\left(\frac{\mu_{z,i}}{\sqrt{2\Sigma_{z,ii}}}\right)\right)^2 \end{aligned}$
	$\begin{aligned} f_{\Sigma,ij} &= \frac{\sqrt{\det(\tilde{\Sigma}_z)}}{2\pi} \exp\left(-\tfrac{1}{2}\tilde{\mu}_z^{\mathsf{T}}\tilde{\Sigma}_z^{-1}\tilde{\mu}_z\right) \\ &\quad + \frac{\sqrt{\det(\tilde{\Sigma}_z)}}{2\pi}\, \mu_{z,j} \frac{\sqrt{\pi\, \tilde{\Sigma}_{z,jj}^{-1}}}{\sqrt{2}} \exp\left(-\tfrac{1}{2}\tilde{\Sigma}_{z,ii}^{-1}\, \mu_{z,i}^2\right) \\ &\quad \times \exp\left(\frac{\left(\tilde{\Sigma}_{z,ji}^{-1}\, \mu_{z,i}\right)^2}{2\tilde{\Sigma}_{z,jj}^{-1}}\right) \left(1 + \mathrm{erf}\left(\frac{\left(\tilde{\Sigma}_z^{-1}\tilde{\mu}_z\right)_j}{\sqrt{2\tilde{\Sigma}_{z,jj}^{-1}}}\right)\right) \\ &\quad + \frac{\sqrt{\det(\tilde{\Sigma}_z)}}{2\pi}\, \mu_{z,i} \frac{\sqrt{\pi\, \tilde{\Sigma}_{z,ii}^{-1}}}{\sqrt{2}} \exp\left(-\tfrac{1}{2}\tilde{\Sigma}_{z,jj}^{-1}\, \mu_{z,j}^2\right) \\ &\quad \times \exp\left(\frac{\left(\tilde{\Sigma}_{z,ij}^{-1}\, \mu_{z,j}\right)^2}{2\tilde{\Sigma}_{z,ii}^{-1}}\right) \left(1 + \mathrm{erf}\left(\frac{\left(\tilde{\Sigma}_z^{-1}\tilde{\mu}_z\right)_i}{\sqrt{2\tilde{\Sigma}_{z,ii}^{-1}}}\right)\right) \\ &\quad + \left[\mu_{z,i}\, \mu_{z,j} - \tilde{\Sigma}_{z,ij}^{-1} \det\left(\tilde{\Sigma}_z\right)\right] \\ &\quad \times \left[\frac{1}{2}\mathrm{erf}\left(\frac{\sqrt{2}\, \mu_{z,i}}{\sqrt{\Sigma_{z,ii}}}\right) + \frac{1}{2}\mathrm{erf}\left(\frac{\sqrt{2}\, \mu_{z,j}}{\sqrt{\Sigma_{z,jj}}}\right) + F_{\tilde{\mu}_z, \tilde{\Sigma}_z}(0,0)\right] \\ &\quad - \left[\frac{\sqrt{\Sigma_{z,ii}}}{\sqrt{2\pi}} \exp\left(-\frac{\mu_{z,i}^2}{2\Sigma_{z,ii}}\right) + \frac{\mu_{z,i}}{2}\left(1 + \mathrm{erf}\left(\frac{\mu_{z,i}}{\sqrt{2\Sigma_{z,ii}}}\right)\right)\right] \\ &\quad \times \left[\frac{\sqrt{\Sigma_{z,jj}}}{\sqrt{2\pi}} \exp\left(-\frac{\mu_{z,j}^2}{2\Sigma_{z,jj}}\right) + \frac{\mu_{z,j}}{2}\left(1 + \mathrm{erf}\left(\frac{\mu_{z,j}}{\sqrt{2\Sigma_{z,jj}}}\right)\right)\right] \end{aligned}$
$\phi(z) = z + \epsilon\, z^2$	$f_{\mu,i} = \mu_{z,i} + \epsilon\, (\mu_{z,i})^2 + \epsilon\, \Sigma_{z,ii}$
	$f_{\Sigma,ij} = \Sigma_{z,ij} + 2\,\epsilon\, \Sigma_{z,ij}\, \left(\mu_{z,i} + \mu_{z,j}\right) + 2\,\epsilon^2\, (\Sigma_{z,ij})^2 + 4\,\epsilon^2\, \mu_{z,i}\, \Sigma_{z,ij}\, \mu_{z,j}$

Table 2.2: Overview of interaction functions for different non-linear activation functions ϕ. In the derivation for both examples, we use that the pre-activations are approximately Gaussian distributed as $z \sim \mathcal{N}(\mu_z, \Sigma_z)$. We drop the layer index here for better readability. For the off-diagonal elements of the covariance in the case of ReLU, we use the marginalized distribution with respect to $\tilde{z} = (z_i, z_j)^{\mathsf{T}}$; writing the marginalized mean and covariance as $\tilde{\mu}_z$ and $\tilde{\Sigma}_z$ and the corresponding cumulative distribution function as $F_{\tilde{\mu}_z, \tilde{\Sigma}_z}(x, y)$.

contributions from third- and fourth-order input cumulants

$\Sigma_{y^{(l)},ij}\,|_{\text{add.}} =$  $+$ 

$+$ 

$$
\begin{aligned}
&= \epsilon \left(G^{(3)}_{z^{(l)},(i,j,j)} + G^{(3)}_{z^{(l)},(j,i,i)} \right) + 2\epsilon^2 \left(G^{(1)}_{z^{(l)},(i)}\, G^{(3)}_{z^{(l)},(i,j,j)} + G^{(1)}_{z^{(l)},(j)}\, G^{(3)}_{z^{(l)},(j,i,i)} \right) \\
&\quad + \epsilon^2\, G^{(4)}_{z^{(l)},(i,i,j,j)}.
\end{aligned}
$$

We here again have to account for permutations of the indices (i,j) in the diagrams (see Sec. 2.3.2), yielding multiple summands for individual diagrams.

Since the pre-activations $z^{(1)}$ in the input layer result from a linear transformation of the inputs x, their cumulants $G^{(n)}_{z^{(1)}}$ depend on the cumulants of the input data $G^{(n)}_x$ in a one-to-one mapping of orders n as $G^{(n)}_x \mapsto G^{(n)}_{z^{(1)}}$ as derived in Eq. (2.11) and we write the interaction functions as a function of the input cumulants

$$
\begin{aligned}
\mu_{y^{(1)}} &= \tilde{h}_\mu(\{G^{(n)}_x\}_n; \{W^{(1)}, b^{(1)}\}), \\
\Sigma_{y^{(1)}} &= \tilde{h}_\Sigma(\{G^{(n)}_x\}_n; \{W^{(1)}, b^{(1)}\}).
\end{aligned}
$$

Overall, we find that higher-order cumulants of the data $G^{(n>2)}_x$ are predominantly extracted by the input layer because the higher-order cumulants are not suppressed as strongly for $D_{\text{in}} \ll N$.

2.4.4 Statistical representation of feed-forward networks

Building on the previous section, we introduce a statistical representation of the network. It describes the information processing performed by the network on the level of the cumulants of the data $\{G^{(n)}_x\}_n$.

We obtain the mean and covariance of the network output $y = g(x;\theta)$ as functions of the statistics of the input x by iterating Eq. (2.15), Eq. (2.20), and Eq. (2.18) across layers

$$
\begin{aligned}
\mu_y &= W^{(L+1)}(f_\mu(\ldots\{\tilde{h}_\nu(\{G^{(n)}_x\}_n)\}_{\nu=\mu,\Sigma}\ldots)) + b^{(L+1)} \\
&=: g_\mu(\{G^{(n)}_x\}_n; \theta, \phi), \qquad (2.22) \\
\Sigma_y &= W^{(L+1)}(f_\Sigma(\ldots\{\tilde{h}_\nu(\{G^{(n)}_x\}_n)\}_{\nu=\mu,\Sigma}\ldots))(W^{(L+1)})^\mathsf{T} \\
&=: g_\Sigma(\{G^{(n)}_x\}_n; \theta, \phi). \qquad (2.23)
\end{aligned}
$$

Since we expect the statistics of inputs x to differ between class labels t, these iteration equations apply per class t, yielding the distribution of the network output as a Gaussian mixture $p(y) = \sum_t p(t)\mathcal{N}(\mu_y^t, \Sigma_y^t)(y)$. Mean and covariance of the network output (μ_y^t, Σ_y^t) are determined using the transformation of the data cumulants $\{G_x^{(n),t}\}_{n,t}$ by the whole network from Eq. (2.22)-(2.23). It is important to here point out again that these equations are not exact since we approximate the pre-activations $z^{(l)}$ as Gaussian at each hidden layer l. In the remainder of this chapter, we call the mapping

$$g_{\text{stat}} : (\{G_x^{(n),t}\}_{n,t}, \theta, \phi) \mapsto p(y) \tag{2.24}$$

the statistical model of the network. We note that one important property of the statistical model is that it shares the set of parameters $\theta = \{W^{(l)}, b^{(l)}\}_{l=1,\ldots,L+1}$ with the network; more specifically, there is a one-to-one correspondence between the statistical model Eq. (2.24) and the network model $g\colon (x;\theta) \mapsto y$ given a fixed set of parameters θ.

The statistical model provides a framework to study information processing in the network for different tasks as well as to investigate the contribution of cumulants of the data $\{G_x^{(n),t}\}_{n,t}$ for different tasks. In Sec. 2.4.1, we showed that the expected mean squared error loss $\mathcal{L}_{\text{MSE}}(\{\mu_y^t, \Sigma_y^t\}_t)$ can be written as Eq. (2.14); this expression is a function of mean and covariance of the output. Combining this formulation of the error with the statistical model in Eq. (2.24), we can write the mean squared error $\mathcal{L}_{\text{MSE}}$ as a function of the cumulants of the data $\{G_x^{(n),t}\}_{n,t}$ and the network parameters θ:

$$\mathcal{L}_{\text{MSE}}\big(\{\mu_y^t, \Sigma_y^t\}_t\big) \approx \mathcal{L}_{\text{MSE}}\big(\{G_x^{(n),t}\}_{n,t}; \theta\big). \tag{2.25}$$

Thus, we can train the statistical model on a given task by minimizing this loss with standard methods. Given the trained parameters θ^* for the statistical model, we use the one-to-one relationship between statistical model and network to transfer these parameters to the network. This network $g(x;\theta^*)$ then only 'sees' the set of cumulants of the data $\{G_x^{(n),t}\}_{n,t}$ used to minimize Eq. (2.25). In this way, we can test different sets of cumulants of the data to investigate their contribution for solving a particular task.

2.5 Experiments

We utilize the presented framework to investigate the information processing in feed-forward networks on multiple tasks: the XOR problem, the MNIST data set, and the CIFAR-10 data set. For the network architecture defined in Sec. 2.2, we set the network width $N_l = N$ to be fixed for $l \geq 1$ and use either the ReLU activation function or the quadratic activation function $\phi(z) = z + \epsilon z^2$ with $\epsilon = 0.5$.

2.5.1 Training details

To initialize network parameters θ, we sample from Gaussians with $\sigma_w^2 = \sigma_b^2 = 0.75$. Networks are trained by minimizing the empirical risk of the MSE loss over mini-batches $\{(x_\alpha, t_\alpha)\}_\alpha$ of size B:

$$\mathcal{L}_{\text{emp, MSE}}(\theta) = \frac{1}{B} \sum_{\alpha=1}^{B} \ell_{\text{MSE}}(g(x_\alpha;\theta), t_\alpha) \,. \tag{2.26}$$

We set the batch size $B = 10$ on XOR and $B = 100$ on MNIST. While the choice of optimizer for training may change the correlation structure of parameters in trained networks, it does not affect the theoretical framework itself. We employ Adam (Kingma and Ba, 2015; Loshchilov and Hutter, 2019) with learning rate 10^{-3} and otherwise standard settings (momenta $\beta_1 = 0.9$ and $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and $\lambda = 0$).

2.5.2 Statistical information encoding for the XOR task

We begin by examining the XOR task as an example of a non-linearly separable task, thus requiring information exchange between cumulants of different orders mediated by the non-linear activation function. We use setting of this task as a Gaussian mixture model, which has two conceptual advantages: First, by having an exact representation of the input distribution in terms of Gaussians, we can isolate the information processing by hidden layers of the network. Second, the formulation of both class-conditional distributions as a Gaussian mixture allows us to study the class-conditional behavior using two different statistical representations that isolate different cumulants orders of the input (mean and covariance). Our findings indicate that although the network can solve the task using either representation, the networks converge to different local minima in the empirical loss landscape.

XOR task as a Gaussian mixture

For the XOR task, we define the input distribution as a Gaussian mixture with four mixture components (see Fig. 2.2(a)); thus we have real-valued instead of binary inputs. For the class label $t = +1$, the mean values of its two components ± are set to $\mu_x^{t=+1,\pm} = \pm(0.5, 0.5)^\intercal$. For the class label $t = -1$, they are set to $\mu_x^{t=-1,\pm} = \pm(-0.5, 0.5)^\intercal$. The covariances are identical for all components and are set to $\Sigma_x^{t,\pm} = 0.05\,\mathbb{I}$. All components receive the same weight $p(t) = p_\pm = \frac{1}{2}$. Thus, the input distribution is given by

$$p(x,t) = p(t) \sum_{\pm} p_\pm \mathcal{N}(\mu_x^{t,\pm}, \Sigma_x^{t,\pm})(x). \tag{2.27}$$

Each data sample x_α gets assigned a target label $t_\alpha \in \{\pm 1\}$ depending on which mixture component it has been drawn from. From the illustration in Fig. 2.2(a), one

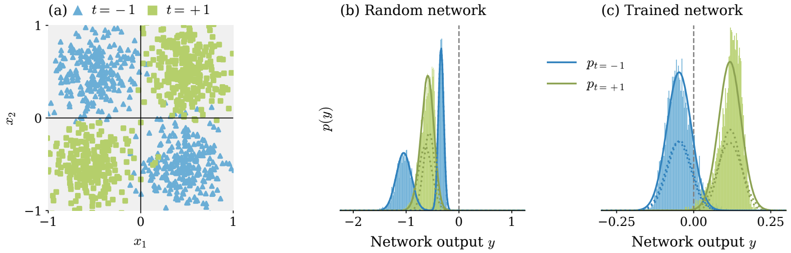


Figure 2.2: Information flow in ReLU networks on the XOR task. (a) We characterize the input distribution as a Gaussian mixture model. The class labels $t = \pm 1$ are assigned to data samples x_α (blue and green dots) based on the mixture component they were drawn from. (b)-(c) Distribution of the network output for (b) a random network and (c) a network trained to convergence. Each mixture component (dashed curves) is propagated through the network as in Eq. (2.22)-(2.23), yielding the class-conditional distributions (solid curves) as a superposition in Eq. (2.28). Mapping a set of test data points by the network yields empirical estimates of the class-conditional distributions (blue and green histograms). For binary classification, we set the classification threshold to be $y = 0$ (gray lines). The trained network in (c) achieves $\mathcal{P} = 93.82\%$ performance compared to $\mathcal{P}_{\text{opt}} = 97.5\%$. Other parameters: $\phi = \text{ReLU}$, depth $L = 1$, width $N = 10$.

can directly read off the optimal decision boundaries that correspond to the axes in data space; the optimal performance is then given by $\mathcal{P}_{\text{opt}} = 97.5\%$. Training and test data set sizes are set to $P_{\text{train}} = 10^5$ and $P_{\text{test}} = 10^4$.

Information processing in internal network layers by cumulants

Since we know the exact input distribution of the XOR task in terms of Gaussian distributions, we apply the statistical model in Eq. (2.22)-(2.23) to each mixture component $(t, \pm)$ separately

$$p_{\text{theo.}}(y) = \sum_t p(t) \sum_{\pm} p_{\pm} \mathcal{N}(\mu_y^{t,\pm}, \Sigma_y^{t,\pm})(y), \tag{2.28}$$

with $\mu_y^{t,\pm} = g_\mu(\mu_x^{t,\pm}, \Sigma_x^{t,\pm}; \theta, \phi)$ and $\Sigma_y^{t,\pm} = g_\Sigma(\mu_x^{t,\pm}, \Sigma_x^{t,\pm}; \theta, \phi)$ being functions of the input statistics, and the network parameters; their particular shape further depends on the choice of activation function. We compare this theoretical result with the empirical estimate of the output distribution $p_{\text{emp.}}(y)$, given as a histogram over the test data, for both an untrained network with random initialization (see Fig. 2.2(b)) and a network trained to convergence (see Fig. 2.2(c)).

In both cases, the output distribution results from the superposition of the Gaussian distributions corresponding to each mixture component. For the random network,

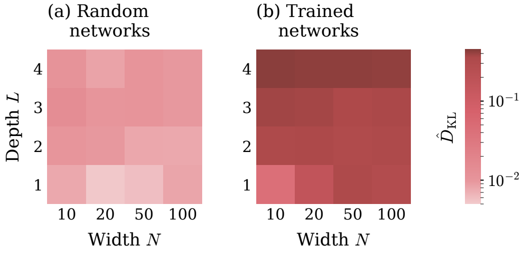


Figure 2.3: Comparison between theoretical and empirical output distribution for (a) random networks and (b) networks trained to convergence. The normalized Kullback-Leibler divergence $\hat{D}_{\mathrm{KL}}(p_{\mathrm{emp.}}\|p_{\mathrm{theo.}})$ is used as a deviation measure and averaged over 50 network realizations. Networks are trained on the XOR task; trained networks achieve average performance values of $\mathcal{P} = 97.00\% \pm 0.05\%$ relative to $\mathcal{P}_{\mathrm{opt}} = 97.5\%$. Other parameters: $\phi = \mathrm{ReLU}$.

these yield a complex distribution, where each mixture component matches the theoretical predictions well (see Fig. 2.2(b)). Network training changes the output distribution such that the class-conditional distributions $p(y|t)$ concentrate around the labels and separate from one another (see Fig. 2.2(c)); the remaining overlap between class-conditional distributions corresponds to the classification error of the network. We provide results for the quadratic activation function in App. A.4.

As a quantitative comparison measure between theoretically predicted output distribution $p_{\mathrm{theo.}}(y)$ and empirically estimated output distribution $p_{\mathrm{emp.}}(y)$, we use the Kullback-Leibler divergence D_{KL} between theoretical and empirical distributions and normalize by the entropy H of the empirical distribution $p_{\mathrm{emp.}}(y)$, yielding

$$\hat{D}_{\mathrm{KL}}(p_{\mathrm{emp.}}\|p_{\mathrm{theo.}}) = D_{\mathrm{KL}}(p_{\mathrm{emp.}}\|p_{\mathrm{theo.}})/H(p_{\mathrm{emp.}}). \tag{2.29}$$

This quantity measures deviations between theory and simulation. Again, we compare between untrained networks with randomly drawn parameters (see Fig. 2.3(a)) and networks trained to convergence (see Fig. 2.3(b)). Overall, theory and simulation match well. We observe a slight increase in deviations with the network depth L, which is expected since approximation errors accumulate across network layers. For random networks, we would expect a decrease of deviations for wider networks; however, this may require going to even wider networks while the approximations already hold well for rather narrow networks as shown here. Due to the correlation of network parameters in trained networks, deviations are overall larger but remain modest, indicating that the theoretical predictions continue to be applicable in this case. We provide results for the quadratic activation function in App. A.4.

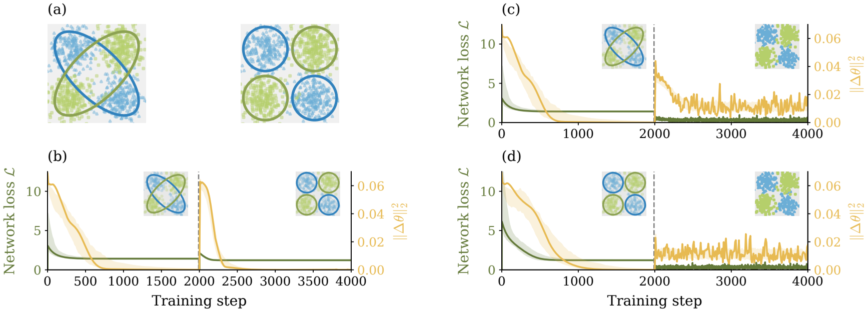


Figure 2.4: Different statistical information encodings of the XOR task lead to different local minima of the loss landscape. (a) Statistical encodings of the XOR task. Left: Covariance coding; class membership (blue and green ellipses) is encoded in the covariance. Right: Mean coding; the Gaussian components of each class (blue/ green circles) differ in their means but have identical covariances. (b)-(d) Evolution of network loss $\mathcal{L}$ ($\mathcal{L}_{\mathrm{emp,MSE}}$ Eq. (2.26) for training the network model, $\mathcal{L}_{\mathrm{MSE}}$ Eq. (2.14) for the statistical model) and change of network parameters $||\Delta\theta||_2^2$ across training steps: For the first $T_1 = 2000$ steps, we train the network based on one encoding, starting from random initialization. The statistical encoding is changed at T_1, starting from parameters $\theta(T_1)$ obtained in the preceding step. We measure the change of network parameters every 10 training steps $\|\Delta\theta(T)\|_2^2 = \|\theta(T) - \theta(T-10)\|_2^2$. Shaded areas indicate lower and upper quartiles across 10^2 network realizations. Solid curves correspond to a single network realization. (b) First part: statistical model with covariance coding; second part: statistical model with mean coding. (c) First part: statistical model with covariance coding; second part: network model on data samples. (d) First part: statistical model with mean coding; second part: network model on data samples. Other parameters: $P_{\mathrm{train}} = 10^4$, 2 epochs, $\phi(z) = z + \epsilon z^2$, depth $L = 1$, width $N = 10$.

Different information encodings and their relation

We have seen that the mapping implemented by the network can be cast into a mapping of cumulants by the statistical model in Eq. (2.24). From the perspective of mapping cumulants, the network's ability to solve a given task can be broken down into two parts: (1) the ability of the network architecture to implement a desired mapping of cumulants from input to output and (2) the encoding of class membership in the cumulants of the input data.

While the first part requires probing the space of possible mappings, which goes beyond the scope of this chapter, we can study the second part by considering two different information encodings for the XOR task: (a) the class membership is encoded in different means, while covariances and higher-order cumulants are all identical and (b) class membership is encoded in different covariances, while means and higher-order cumulants are all identical. We refer to these two encodings as (a) mean coding and (b) covariance coding in the following. For the XOR task, we can use these two encodings to restrict information on class membership to a single cumulant order. In general, one expects that class membership is encoded in multiple cumulants of different orders, allowing the network to recombine information as necessary to maximize performance.

For comparing these statistical information encodings, we use the statistical model in Eq. (2.24): For (a) mean coding, we keep both the class labels t and the specific mixture component ± from which a sample was drawn, yielding four sets of statistics $\{\mu_x^m, \Sigma_x^m\}_{m=(t,\pm)}$ with different means but identical covariances. For (b) covariance coding, we keep only the class label t, yielding two sets of statistics $\{\mu_x^m, \Sigma_x^m\}_{m=(t)}$ with different covariances $\Sigma_x^{t=\pm1} = \left(\begin{smallmatrix} 0.3 & \pm0.25 \\ \pm0.25 & 0.3 \end{smallmatrix}\right)$ but identical means. In both cases, we set all higher-order cumulants of the component distributions $m = (t, \pm)$ and $m = (t)$, respectively, to zero. Note that mean coding $p(x, t = \pm1) = \sum_\pm p_\pm \mathcal{N}(\mu_x^{t,\pm}, \Sigma_x^{t,\pm})(x)$ in fact involves higher-order cumulants for the class-conditional distributions. We visualize the different statistical encodings in Fig. 2.4(a).

We compare mean and covariance coding in the statistical model with training networks on batches of data samples, which we refer to as sample coding. In principle, network training may utilize cumulants of arbitrary orders from the data. We now tackle three main questions: First, which statistical encoding corresponds most closely to the information representation in networks trained on data samples? Second, how are these statistical encodings different and can both be efficiently used by the network to solve the given task? Third, do networks trained on data samples recombine information of different cumulants to improve network performance?

We use the following methodology to probe these questions: In a first part, we optimize models with either information representation (mean coding, covariance coding, sample coding) until convergence. Then, we change the information representation and finetune the network parameters for the same number of steps as before, thereby investigating the stability of the local minima. We measure the loss and the change of network parameters across training steps.

For the first part, all trained models yield performance values of at least $\mathcal{P} = 91\%$, thus have converged to network parameters suitable for solving the XOR task. For covariance coding, we find that, when switching to mean coding, $||\Delta\theta||_2^2$ is of the same order of magnitude as in the initial training steps, indicating complete re-training of the model (see Fig. 2.4(b)). This behavior suggests that mean and covariance coding lead to different solutions. On the other hand, when switching to sample coding (see Fig. 2.4(c)), $||\Delta\theta||_2^2$ changes only slightly. When switching from mean coding to sample coding, $||\Delta\theta||_2^2$ changes only negligibly (see Fig. 2.4(d)). Thus, while mean and covariance coding yield different solutions due to their different information encodings, both form local minima in the loss landscape of the network.

From a theoretical perspective, covariance coding is of particular interest: this encoding is an example where the non-linear activation function is required to transfer information from higher-order cumulants to the class means since classification utilizes different mean values in the network output. For the quadratic activation function $\phi(z) = z + \epsilon z^2$ used in Fig. 2.4, the transfer function is particularly easy to interpret

$$\mu_{y^{(l)},i} = \mu_{z^{(l)},i} + \epsilon \left(\mu_{z^{(l)},i}\right)^2 + \epsilon \Sigma_{z^{(l)},ii}. \tag{2.30}$$

It is worth noting that the information transfer occurs on the diagonal elements of the covariance, whereas for the XOR task the class-conditional input covariances $\Sigma_x^{t=\pm1} = \left(\begin{smallmatrix} 0.3 & \pm0.25 \\ \pm0.25 & 0.3 \end{smallmatrix}\right)$ differ on the off-diagonal. Transferring information from diagonal to off-diagonal is mediated by the linear transformation. This case illustrates how our theoretical framework can be used to trace the information flow across layers that is required for successful classification.

With respect to our initial questions, we conclude that both mean and covariance coding can efficiently be utilized by the network. Network training resembles more closely mean coding, thus recombining information from higher-order cumulants of the data.

2.5.3 Gaussian statistics are essential to the MNIST data set

We here study the MNIST data set (LeCun, Cortes, and Burges, 1998) that consists of 28×28 pixel, grayscale images of handwritten digits from zero to nine (see Fig. 2.5(a) for digit three). The task is to classify the images into ten classes according to the shown digit. This data set was constructed to be highly structured: approximating the class-conditional distributions as multivariate Gaussians, drawn samples from these Gaussians already appear rather realistic (see Fig. 2.5(b)). Based on this observation, one expects that mean and covariance are the main drivers for classification by the network. We test this hypothesis here by comparing different data sets and different sets of input cumulants. When optimizing the parameters θ^* of the statistical model, we can use different sets of cumulants $\{G_x^{(n)}\}_{n=1,\ldots,\hat{n}}$, yielding different performance values for the corresponding network on the test data set. By comparing these values to that of a network trained on data samples, the difference in performance serves as a measure for the importance of the cumulants kept in $\{G_x^{(n)}\}_{n=1,\ldots,\hat{n}}$.

For the statistical model in Eq. (2.22)-(2.23), we use two different approximations: we approximate the input data distribution $p_x \approx \hat{p}_x(\{G_x^{(n)}\}_{n=1,\ldots,\hat{n}})$ up to a certain cumulant order $\hat{n}$ and we approximate the signal distributions $p_{z^l} \approx \hat{p}_{z^l}(\mu_{z^l}, \Sigma_{z^l})$ as Gaussians. The approximation of the input distribution limits what information from the data itself is available to the network; the approximation of the signal distributions limits how this information is being processed by the network. To focus on the truncation of higher-order cumulants in the input distribution, we sample from the approximation of the input distribution $x \sim \hat{p}_x(\{G_x^{(n)}\}_{n=1,\ldots,\hat{n}})$ and train the network on these samples, thereby not limiting the information processing by the network.

To obtain a Gaussian approximation of the class-conditional input distributions, we flatten the 28×28 pixel images into 784-dimensional vectors and then estimate means $\hat{\mu}_x^t$ and covariances $\hat{\Sigma}_x^t$ for each class t empirically from the training data set. The estimated covariances $\hat{\Sigma}_x^t$ are only positive semi-definite instead of positive definite because the image edges exhibit no variance. We use a principal component analysis of the covariance matrix to take care of zero eigenvalues: For each class t, the covariance matrix can be written as

$$\hat{\Sigma}_x^t = V\mathcal{D}V^{\mathsf{T}}, \tag{2.31}$$

where $V = (v_1|\ldots|v_{N_0})$ consists of unit-length eigenvectors v_i and $\mathcal{D} = \mathrm{diag}(\lambda_1, \ldots, \lambda_{N_0})$ of the respective eigenvalues λ_i of $\hat{\Sigma}_x^t$. The eigenvalues are ordered according to size $\lambda_1 \geq \cdots \geq \lambda_{N_0} \geq 0$. By choosing a threshold $\vartheta_{\mathrm{PCA}} > 0$ that defines a subspace U spanned by the eigenvectors $\{v_i\}_{i=1,\ldots,N_{\mathrm{PCA}}}$ for which $\lambda_i > \vartheta_{\mathrm{PCA}}$, we generate data samples $\hat{x}_\alpha|_U$ in this subspace U and project these samples back to the input space $\mathbb{R}^{D_{\mathrm{in}}}$:

$$\hat{x}_\alpha|_U \sim \mathcal{N}(0, \mathrm{diag}(\lambda_1, \ldots, \lambda_{N_{\mathrm{PCA}}})), \tag{2.32}$$

$$\hat{x}_\alpha = \hat{\mu}_x^t + V \begin{pmatrix} \hat{x}^{(d)}|_U \\ 0 \end{pmatrix}. \tag{2.33}$$

We set $\vartheta_{\mathrm{PCA}} = 10^{-2}$ in the following, keeping relevant eigenvalues and excluding noise due to finite numerical precision. We generate a data set of $P = 60{,}000$ Gaussian samples (same size as MNIST training data set). We use one-hot encoding for training, so that the output dimensionality is $D_{\mathrm{out}} = 10$.

We start by training networks on either the MNIST data set (see Fig. 2.5(a)) or the corresponding Gaussian samples (see Fig. 2.5(b)) using the standard empirical loss in Eq. (2.26). In both setting, processing of information in hidden network layers is not restricted to any cumulant order. We observe that removing higher-order cumulants from the input distribution leads to performance values that are $\Delta\mathcal{P} \simeq 2.4\% \pm 0.7\%$ lower than when training on the original data set (see Fig. 2.5(c)). Further, the achieved performance of $\mathcal{P} \approx 91\%$ is accountable to class-conditional means and covariances of the data, demonstrating that the Gaussian input statistics are already highly informative for the classification task.

To study the internal information processing by the network, we next optimize the statistical model in Eq. (2.24) on the Gaussian approximation of MNIST. This gives a modest drop of performance by about $0.9 \pm 0.4\%$ (see Fig. 2.5(c)), from which we conclude that the information processing by the internal layers is well described in terms of Gaussian statistics. Altogether, we find that on MNIST the Gaussian statistics of the data account for the main part of classification performance, although

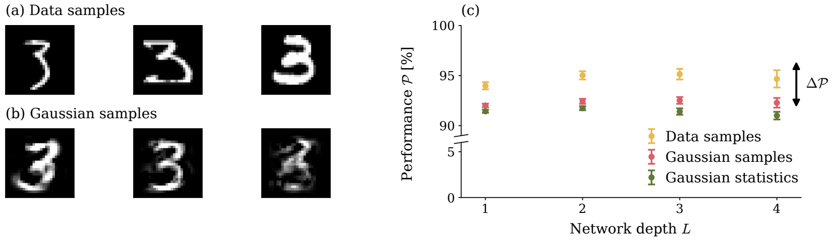


Figure 2.5: Class-conditional means and covariances of MNIST give main contribution to classification performance. (a) Three example data samples for the digit three from the MNIST training data set. (b) Data samples for the digit three that are drawn from the Gaussian approximation of the class-conditional input distribution. (c) Classification performance on the MNIST test data set for different statistical encodings. Network training achieves performances of $\mathcal{P} \approx 94\%$ or more (yellow), while the statistical model with Gaussian statistics yields performances that are $3.3\% \pm 0.6\%$ lower (green). Training networks on Gaussian input samples leads to performance values comparable to Gaussian statistics (red). Performance is always evaluated on the MNIST test data set. Error bars indivate mean and one standard deviation across 10^2 network realizations. Other parameters: $\phi(z) = z + \epsilon z^2$, width $N = 100$.

higher-order cumulants of the data are required for the last few percents. In the next section, we demonstrate how higher-order cumulants can effectively be included into our theoretical framework.

2.5.4 Extracting higher-order cumulants in the input layer

Up to here, we focused on the class-conditional means μ_x^t and covariances Σ_x^t of the input data; however, for complex tasks we expect higher-order cumulants to be just as important. As an extreme case, there can be tasks where these two cumulants are not even informative for discriminating between the classes: One can construct tasks with two classes $t = \pm 1$, where both the class-conditional means and covariances of the data are identical – $\mu_x^{t=-1} = \mu_x^{t=+1}$, $\Sigma_x^{t=-1} = \Sigma_x^{t=+1}$. Thus, it is impossible to distinguish between classes solely based on Gaussian statistics; instead classification must rely on discriminative information in higher-order cumulants.

We study a low-dimensional example of such a task in the following. The input distribution for this task is defined as a Gaussian mixture with four mixture components. The task consists of two classes, $t = \pm 1$, each comprised of two Gaussian components $+$ and $-$, with means

$$\mu_x^{t=+1,-} = (-0.5, 0)^\top, \qquad \mu_x^{t=-1,-} = (-1.5, 0)^\top, \tag{2.34}$$

$$\mu_x^{t=+1,+} = (1.5, 0)^\top, \qquad \mu_x^{t=-1,+} = (0.5, 0)^\top. \tag{2.35}$$

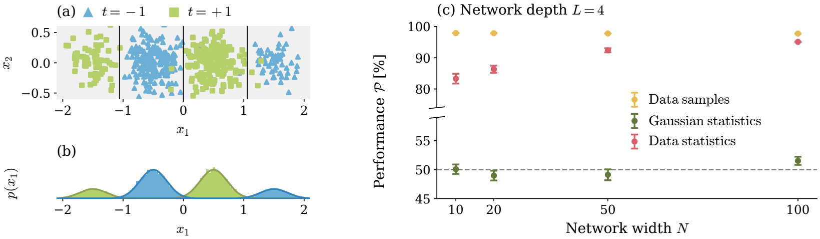


Figure 2.6: Networks extract higher-order cumulants in the input layer. (a) The distribution of input data is given by a Gaussian mixture. Data samples x_α (blue and green dots) are assigned class labels $t = \pm 1$ based on the mixture component that they were drawn from. Both classes have zero mean and identical covariances, but differ in their higher-order cumulants. (b) Projection of data samples onto x_1-axis (histograms) corresponds to the marginalization of the data distribution over x_2 (solid lines), indicating the different relative weights of the mixture components. (c) Classification performance for models incorporating different sets of statistics. Training networks on data samples achieves performance values of $\mathcal{P} \approx 96\%$ or more (yellow). While the statistical model $g_{\mathrm{stat}}(\{G_x^{(n)}\}_{n=1,2}, \theta)$ optimized on Gaussian statistics (green) results in performance values at chance level $\mathcal{P} \approx 50\%$ (dotted line), incorporating higher-order cumulants into the statistical model $\tilde{g}_{\mathrm{stat}}(\{G_x^{(n)}\}_{n=1,2,3,4}, \theta)$ (red) almost completely bridges the gap to training networks on data samples. Error bars indicate the standard deviation over 10^2 different network initializations. Other parameters: quadratic non-linearity $\phi(z) = z + \epsilon z^2$.

The covariances of all components are isotropic

$$\Sigma_x^{t,\pm} = 0.05\,\mathbb{I}. \tag{2.36}$$

We weight the components $(t = -1, -)$ and $(t = +1, +)$ by $p_{\mathrm{outer}} = \frac{1}{8}$ and the components $(t = -1, +)$ and $(t = +1, -)$ by $p_{\mathrm{inner}} = \frac{3}{8}$, as shown in Fig. 2.6(a)-(b). We assign a target label $t_\alpha \in \{\pm 1\}$ to each data sample x_α based on the mixture component it is drawn from. For this particular setup, the class-conditional means and covariances of the input data are identical

$$\mu_x^{t=\pm 1} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \qquad \Sigma_x^{t=\pm 1} = \begin{pmatrix} 0.8 & 0 \\ 0 & 0.5 \end{pmatrix}. \tag{2.37}$$

However, the third-order cumulants have different signs

$$G_{x,(i,j,k)}^{(3),t=\pm 1} = \pm 0.75\,\delta_{ij}\delta_{jk}\delta_{ki}\delta_{i1}. \tag{2.38}$$

Training and test data sets are of size $P = 10^4$.

Since the third-order cumulants are different for the two classes ($G_x^{(3),t=-1} = -G_x^{(3),t=+1}$), we anticipate that incorporating them into the statistical model will enable it to solve this task. In this section, we illustrate that our approach is capable of handling such higher-order cumulants. Specifically, we confirm the assertion in Sec. 2.4.3 that considering higher-order cumulants is necessary only in the first layer.

As expected, training the network results in performance values close to the theoretical upper bound of performance of 97%, which is due to the overlaps of the Gaussian mixture components. However, a statistical model $g_{\mathrm{stat}}(\{G_x^{(n)}\}_{n=1,2}, \theta)$ that incorporates only class-conditional means μ_x^t and covariances Σ_x^t fails to perform well and yields chance-level performance (Fig. 2.6(c)). We can almost completely bridge this performance gap by including the third-order input cumulants $G_x^{(3)}$ (via Eq. (A.26)) in the first layer of the statistical model $\tilde{g}_{\mathrm{stat}}(\{G_x^{(n)}\}_{n=1,2,3,4}, \theta)$; we take into account cumulants up for fourth order here to ensure that covariances are properly positive-definite. For successful classification, we require different means $G_y^{(1)}$ in the network output; this transfer of information from the third-order cumulant $G_x^{(3)}$ to lower-order cumulants is mediated by the non-linear activation function ϕ. More specifically, this information transfer depends entirely on the non-linear part of ϕ, highlighting its importance for the computational power of the network.

2.5.5 High dimensionality of input data justifies Gaussian description of fully-connected deep networks

As an example of a structurally more diverse task compared to MNIST, we here study the CIFAR-10 data set (Krizhevsky and Hinton, 2009), which comprises of ten classes of 32×32 pixel images with three color channels. Compared to the previous section, we anticipate two antagonistic effects: Firstly, due to the significant heterogeneity across images within each class of CIFAR-10, we expect class-conditional distributions to be more complex than for MNIST, necessitating higher-order cumulants to accurately represent their statistical structure. Secondly, the larger input dimensionality of $D_{\mathrm{in}} = 3072$ compared to $D_{\mathrm{in}} = 784$ for MNIST means higher-order cumulants are more strongly suppressed in the input layer (see Sec. 2.4.2). To investigate the trade-off between these two effects, we utilize the methods from previous sections to determine the contribution of cumulants of different orders, similar to Sec. 2.5.3.

For CIFAR-10, we are primarily interested in the contributions from mean and covariance vs. higher-order cumulants. To this end, we train networks on the CIFAR-10 data set, i.e. involving cumulants of all orders, and compare them to the statistical model trained on the Gaussian approximation of CIFAR-10 (see Fig. 2.7). We evaluate network performance in both cases on the CIFAR-10 test data set. Networks trained on data samples achieve $\mathcal{P} = 34.8\% \pm 1.4\%$. In contrast, the statistical model trained on the Gaussian statistics consistently achieves higher performance values of $\mathcal{P} = 37.6\% \pm 1.3\%$; we find the opposite relation between these two settings compared to MNIST. We can understand this result in the context of the two aforementioned

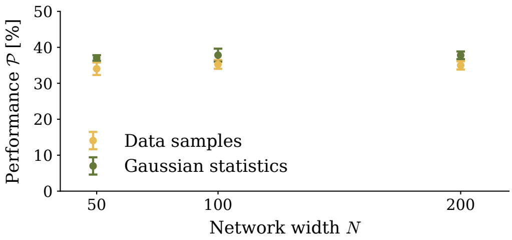


Figure 2.7: Fully-connected networks trained on CIFAR-10 operate only on Gaussian statistics. Training the statistical model on Gaussian statistics (green) consistently yields higher performance than training networks on data samples (yellow). Network performance is evaluated on the test data set. Error bars indicate mean and standard deviation across 10 network initializations. Other parameters: depth $L = 2$, quadratic activation function $\phi(z) = z + \epsilon z^2$.

antagonistic effects: The suppression of higher-order cumulants due to the larger input dimensionality appears to dominate over the networks' need to extract and process them for correctly solving the task. Thus, networks predominantly process only Gaussian statistics of CIFAR-10 and in consequence, the statistical model is a good representation of the networks. A possible explanation for the slightly higher performance of the statistical method is that the statistical model is based on a more accurate estimate of the Gaussian statistics (averaged over the entire training set of $50,000$ images) compared to training on data samples (averaged over mini-batches of 10^2 images). Note that while the performance values are significantly lower than what is reported for state-of-the-art network architectures such as convolutional ResNets (Zagoruyko and Komodakis, 2016), they match what is typically reported for fully-connected feed-forward networks on CIFAR-10 (Lee et al., 2018). Based on this observation, one possible explanation is that these different architectures extract and utilize different sets of cumulants and thus statistical information to learn CIFAR-10: our theoretical framework predicts that fully-connected feed-forward networks are limited to Gaussian statistics in the case of high-dimensional input data and are thus only able to capture part of the structure for more complex data sets such as CIFAR-10, limiting performance on such tasks. In conclusion, studying the information processing of neural networks on the level of cumulants gives us a handle to link computational power of neural networks to the statistical structure of the data.

2.6 Conclusion

In this chapter, we derived a theoretical framework to trace the transformation of the data distribution in fully-connected networks across layers and link this transformation to the information processing performed by the network. We parameterize the data distribution in terms of cumulants and investigate the relevance of different cumulant orders for the classification task. Including higher-order cumulants requires a perturbative treatment of the non-linearity in the network layer.

For wide networks, we argue that it is sufficient to trace the class-conditional means and covariances. We validate this Gaussian hypothesis for the XOR task, which we model as a Gaussian mixture model and thus have an exact representation of the input distribution in terms of Gaussian distributions. While network training introduces correlations between network parameters, we show that under certain conditions our theory continues to apply to trained networks and validate this empirically.

Further, we show for the MNIST data set that class-conditional means and covariances account for the largest part of the classification accuracy, while higher-order cumulants are required to fine-tune for a few additional percentages. We show for a low-dimensional toy task that higher-order cumulants are predominantly extracted in the input layer while the internal information processing is primarily governed by Gaussian statistics. Finally, for more complex image data like CIFAR-10, we find that higher-order cumulants are required to match state-of-the-art performance on this data set. Since higher-order cumulants are suppressed by the large input dimension, fully-connected networks do not have the capacity to solve this task; in contrast, convolutional networks with their sparse structured weight matrices likely involve higher-order cumulants also for their internal information processing. This observation indicates how the theoretical approach presented in this chapter may be applied to probe the computational properties of different network architectures.

2.6.1 Limitations

While the theoretical framework in this chapter and its perturbative methods apply naturally to polynomial activation functions, it can be extended to non-polynomial and even non-differentiable activation functions using Gram-Charlier or Edgeworth expansions. The main advantage of polynomial activation functions is that the information exchange between cumulants of different orders is described by an intuitive diagrammatic language and yields exact analytical expressions. However, since networks with polynomial activation functions do not yield universal function approximators (Cybenko, 1989; Leshno et al., 1993; Pinkus, 1999), commonly used activation functions such as ReLU are non-polynomial. Therefore, we provide results with ReLU for Gaussian statistics and show how to generalize to higher-order cumulants in App. A.1.

For including higher-order cumulants, it is important to note that there are mathematical constraints for truncating cumulants at a certain order. Such truncations need to be done systematically to maintain positivity of the probability distribution as well as conform to properties of the cumulants, e.g. the positive semi-definiteness of the covariance. This is especially relevant for extending the theoretical framework to other network architectures such as convolutional networks that may require tracing higher-order cumulants at hidden network layers.

The main limitation for the applicability of this framework lies in the dimensionality D_{in} of the data and the hidden representations N_l. The cumulant of order n is a tensor with N_l^n elements. Determining higher-order cumulants from the data requires significant amounts of both available data and compute, which increases exponentially with the order n. Further, storing and tracing higher-order cumulants across network layers leads to high memory demands. To ease these constraints, one can use the inherent symmetries of the cumulants under index permutations, as done in (Merger et al., 2023).

2.6.2 Relation to other works

Deco and Brauer (1994) are the first to describe the data and signal distribution in terms of cumulants. They consider the special case of volume-preserving networks and derive a learning rule for decorrelating the network outputs, but are limited to two-layer networks.

Closest in spirit is a line of research that studies how networks learn distributions of increasing complexity, where complexity is measured in terms of cumulants of the data distribution. (Refinetti, Ingrosso, and Goldt, 2023; Belrose et al., 2024) measure learning curves of networks on approximations of the input distributions involving cumulants of different orders. They show that both convolutional networks and transformers first learn Gaussian statistics and then continuously learn more accurate representations of the input distribution. While they study the effect of different approximations of the input distribution on the learning dynamics, our approach also captures the information processing within the network and allows us to link the input cumulants to the networks' computational properties. For invertible networks, a special class of networks that map the data distributions to a latent Gaussian distribution in order to generate new samples, Merger et al. (2023) show that these networks generate interactions of increasing orders across network layers. While their approach captures the information processing within the network, it applies to a different network architecture and traces interaction coefficients instead of cumulants.

Multiple lines of research consider the reciprocal setting to ours where one considers the distribution over network parameters for a fixed training data set. (Goldt et al., 2020; Goldt et al., 2022; Loureiro et al., 2022) study teacher-student settings, where the overlaps between teacher and student weights emerge as natural order parameters that permit a Gaussian approximation for wide networks, mapping high-dimensional integrals to low-dimensional Gaussian integrals to determine quantities like the generalization error. In the limit of infinite width, (Neal, 1996; Williams, 1998; Lee et al., 2018; Garriga-Alonso, Rasmussen, and Aitchison, 2019) find an exact equivalence between neural networks at initialization and Gaussian processes, the so-called neural network Gaussian process (NNGP), where the covariance function becomes the central object. This approach studies how overlaps between samples and thus merely their global relation to one another is transformed by the network, while our approach goes further in that it captures how the internal structure of the samples is processed by the network. To make this explicit, we consider classification on image data as an example: the NNGP involves the scalar product $\sum_i x_i^\alpha x_i^\beta$ between pixels x_i^α, x_i^β of any pair of images α, β, while our approach considers the image structure in the sense of correlations between pixel values x_i^α and x_j^α.

Chapter 3

Critical feature learning in deep neural networks

This chapter, App. B, and parts of the discussion are based on the following publication:

Kirsten Fischer, Javed Lindner, David Dahmen, Zohar Ringel, Michael Krämer, and Moritz Helias. "Critical feature learning in deep neural networks." Proceedings of the 41st International Conference on Machine Learning, PMLR 235:13660-13690 (2024).

Author contributions
Javed Lindner and the author contributed equally to the publication and share the first authorship. Moritz Helias was behind the idea of the publication, implemented preliminary numerical tests of the theory, and helped directing the research. Javed Lindner performed the numerical experiments involving the training of networks using Langevin dynamics in Sec. 3.5.1 and Sec. 3.5.2. The author performed the analysis close to criticality in Sec. 3.5.2 and of the output scale Sec. 3.5.3 as well as the numerical solution of the self-consistency equations in all parts of Sec. 3.5. Zohar Ringel provided the idea of width annealing to solve the self-consistency equations. Both Zohar Ringel and Michael Krämer helped directing the research. Javed Lindner, the author, David Dahmen, and Moritz Helias all contributed to writing the manuscript. All authors contributed to revising and finalizing the manuscript.

3.1 Introduction

We here consider the reciprocal setting to the previous chapter: we study ensembles of neural networks over the distribution of network parameters for a fixed training data set and fully-connected, feed-forward networks as in the previous chapter. We are interested in the Bayesian posterior of the network, which corresponds to selecting only those networks from the ensemble that implement the correct input-output mapping on the training data set. Due to the vast number of parameters in neural networks, it is impossible to keep track of all of them. Instead, when computing the network prior at initialization from a field-theoretic perspective as in (Segadlo et al., 2022), network kernels emerge as natural order parameters of the system. These network kernels describe the covariance structure between different input samples of the Gaussian signal distribution that results from a Gaussian prior on the network parameters.

While the characterization of the signal distribution as Gaussian is rigorous in the infinite-width limit at initialization, it continues to hold approximately when going away from initialization for finite-size networks; however, the network kernels change as shown in (Seroussi, Naveh, and Ringel, 2023; Rubin, Seroussi, and Ringel, 2024). In this chapter, we tackle the question of the emergence and structure of these feature-corrected kernels by determining leading-order feature corrections to the kernels at initialization in a field-theoretic formulation of the network.

The main contributions of this chapter are:

- we derive analytic expressions for the posterior kernels in the Bayesian setting for finite-size networks under small but non-zero training load;
- we find that the forward-backward propagation in the self-consistency equations for the feature-corrected kernels captures the input-label relationship, correctly predicting non-linear kernel adaptation to the target kernel in trained neural networks;
- we identify a connection between kernel fluctuations close to a critical point in hyperparameter space and the networks' ability for feature learning, revealing that the underlying mechanims for feature learning results from a tradeoff between criticality and feature learning scales of the network output.

3.2 Setup

We study deep, fully-connected, feed-forward networks defined as

$$\begin{aligned} h_\alpha^{(0)} &= W^{(0)} x_\alpha + b^{(0)}, \\ h_\alpha^{(l)} &= W^{(l)} \phi\left(h_\alpha^{(l-1)}\right) + b^{(l)} \quad l = 1, \ldots, L, \\ f_\alpha &= h_\alpha^{(L)}. \end{aligned} \tag{3.1}$$

We consider P training samples $x_\alpha \in \mathbb{R}^D$ with data indices $\alpha \in \{1, \ldots, P\}$. We denote the hidden representations as $h_\alpha^{(l)} \in \mathbb{R}^{N_l}$ and network output as $f_\alpha \in \mathbb{R}$. To ease notation, we assume identical width $N_l = N$ for all network layers; for the general case one needs to consider layer size ratios N_l/N as in Segadlo et al. (2022). The here obtained theoretical framework holds for arbitrary activation functions $\phi : \mathbb{R} \mapsto \mathbb{R}$; numerical experiments in subsequent sections are performed for $\phi = \mathrm{erf}$ because this choice allows for exact analytical solutions of the appearing Gaussian integrals. At initialization, we use Gaussian i.i.d. priors for all weights $W^{(0)} \in \mathbb{R}^{N \times D}$, $W^{(l)} \in \mathbb{R}^{N \times N}$, $W^{(L)} \in \mathbb{R}^{1 \times N}$ and biases $b^{(l)} \in \mathbb{R}^N$, $b^{(L)} \in \mathbb{R}$, so that $W_{ij}^{(0)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}\big(0, \sigma_{w,0}^2/D\big)$, $W_{ij}^{(l)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}\big(0, \sigma_w^2/N\big)$ for $i, j = 1, \ldots, N$ and $l = 1, \ldots, L$, and $b_i^{(l)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_b^2)$ for $i = 1, \ldots, N$ and $l = 0, \ldots, L$. To keep the notation concise, we use the same weight variance σ_w^2 and bias variance σ_b^2 across all hidden layers; the theoretical framework can straightforwardly incorporate layer-dependent weight variances $\sigma_{w,l}^2$ and bias variances $\sigma_{b,l}^2$ by adding a layer index l. In this chapter, we derive the Bayesian posterior distribution conditioned on a training data set consisting of inputs $X = (x_\alpha)_{\alpha=1,\ldots,P}$ and corresponding labels $Y = (y_\alpha)_{\alpha=1,\ldots,P}$ as outlined in the following Sec. 3.3.1. This view corresponds to training the network with stochastic Langevin dynamics (see Appendix App. B.5).

3.3 Theoretical background

We first shortly present the concept of Bayesian supervised learning, which is a setting commonly studied in machine learning theory. Next, we state the Neural Network Gaussian Process (NNGP) kernel (Neal, 1996; Lee et al., 2018) for feed-forward networks that describes these at initialization. We then reiterate the derivation of the network prior in a field-theoretic framework following Segadlo et al. (2022), which served as a starting point for our work. Then, we discuss next-to-leading-order corrections to the NNGP kernel in this framework, introducing the response function and fluctuation corrections, which we identify as the driving forces of feature learning in deep neural networks. Finally, we give a brief overview of large deviation theory following Touchette (2009), which appears when approximating the network posterior to determine the feature-corrected kernels.

3.3.1 Bayesian supervised learning

We here briefly review the Bayesian approach to supervised learning (MacKay, 2003). Consider a model $p(y|x, \theta)$ that maps from inputs $x \in \mathbb{R}^D$ to outputs $y \in \mathbb{R}$ given an arbitrary but fixed set of network parameters θ. Then common network training methods find a set of network parameters $\hat{\theta}$ that maximizes the likelihood of the data $p(Y|X, \theta)$ given the training set $\mathcal{D} = \{X, Y\}$. Any observable $\mathcal{A} = \mathcal{A}(X, Y, \theta)$

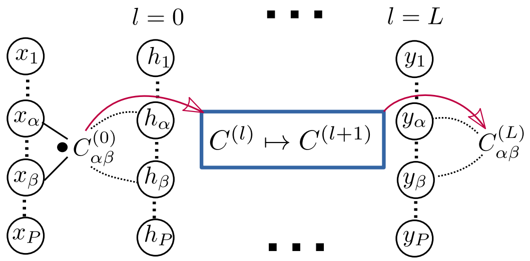


Figure 3.1: The Neural Network Gaussian Process (NNGP) for feed-forward networks exhibits a forward dependence, mapping from one layer to the next.

of the network is then given by $\mathcal{A}(X,Y,\hat{\theta})$; in particular we get a prediction for an unseen test point x^* from $p(y^*|x^*,\hat{\theta})$.

In the Bayesian setting, we instead assume a prior $p(\theta)$ on the network parameters θ and compute the posterior after conditioning on the training data using Bayes' theorem

$$p(\theta|X,Y) = \frac{p(Y|X,\theta)\,p(\theta)}{\int \mathrm{d}\theta\, p(Y|X,\theta)\,p(\theta)}. \tag{3.2}$$

Conditioning on the training data can be seen as selecting all those sets of network parameters θ from the network prior $p(\theta)$ that implement the correct input-ouput mapping $X \mapsto Y$ on the training data set. For any observable $\mathcal{A} = \mathcal{A}(X,Y,\theta)$, we get the Bayesian posterior by marginalizing over the likelihood of the posterior distribution of the network parameters

$$p(\mathcal{A}|X,Y) = \int \mathrm{d}\theta\, p(\mathcal{A}|\theta)\,p(\theta|X,Y), \tag{3.3}$$

which can be rewritten as

$$p(\mathcal{A}|X,Y) = \frac{p(Y,\mathcal{A}|X)}{p(Y|X)}.$$

The term $p(Y|X) = \int \mathrm{d}\theta\, p(Y|X,\theta)\,p(\theta)$ denotes the model-dependent network prior which describes all input-output mappings compatible with the network mapping $p(y|x,\theta)$ and the network prior $p(\theta)$. The observables that we will be interested in are the network kernels

$$C^{(l)}_{\alpha\beta}(\theta) = \frac{\sigma_w^2}{N}\phi(h^{(l-1)}(x_\alpha,\theta))\,\phi(h^{(l-1)}(x_\beta,\theta)) + \sigma_b^2 \tag{3.4}$$

under Gaussian priors $W^{(l)}_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0,\sigma_w^2/N\right)$ and $b^{(l)}_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,\sigma_b^2)$ that appear as natural order parameters of the network as we will show in the following.

3.3.2 Neural Network Gaussian Process

In the limit of infinite width $N \to \infty$, feed-forward networks can be described as a Gaussian process, which is referred to as the Neural Network Gaussian Process (NNGP) (Neal, 1996; Lee et al., 2018). As the signal $h_{i,\alpha}^{(l)} = \sum_{j=1}^{N} W_{ij}^{(l)} \phi(h_{j,\alpha}^{(l-1)}) + b_i^{(l)}$ in layer l contains a sum over N i.i.d. random variables, the central limit theorem implies that for $N \to \infty$ the signal $h^{(l)} \sim \mathcal{N}(0, C_{\alpha\beta}^{(l)})$ is Gaussian distributed with covariance function

$$
\begin{aligned}
C_{\alpha\beta}^{(0)} &= \frac{\sigma_w^2}{D} x_\alpha \cdot x_\beta + \sigma_b^2, \\
C_{\alpha\beta}^{(l)} &= \sigma_w^2 \langle \phi_\alpha^{(l-1)} \phi_\beta^{(l-1)} \rangle_{\mathcal{N}(0,C^{(l-1)})} + \sigma_b^2 \quad l = 1, \ldots, L,
\end{aligned}
$$

which is also often referred to as the NNGP kernel. Here, the dot-product is over the element indices $x_\alpha \cdot x_\beta = \sum_{i=1}^{D} x_{i,\alpha} x_{i,\beta}$. The covariance in layer l depends solely on the covariance in the previous layer and thus only on the inputs $(x_\alpha)_\alpha$ as shown in Fig. 3.1; it does not account for the input-output relation between samples and labels. Bayesian inference on the training data with the NNGP kernel corresponds to training only the output layer of the network (Lee et al., 2018; Yang, 2019).

3.3.3 Network prior in a field-theoretic formulation

We here retrace the derivation of the network prior for feed-forward networks in a field-theoretic formulation as done by Segadlo et al. (2022). They derive the network prios in a more general formulation that includes both feed-forward and recurrent neural networks jointly while we here focus on feed-forward networks.

Marginalization over network parameters

Assuming sample-wise i.i.d. Gaussian regularization noise of variance κ, the network prior is given by

$$
p(Y|X) = \int d\theta \prod_{\alpha=1}^{P} \mathcal{N}(y_\alpha | f_\alpha, \kappa)\, p(f|X;\theta) \tag{3.5}
$$

where $f = (f_\alpha)_{\alpha=1,\ldots,P}$ denotes the network outputs and we marginalize over the network parameters $\theta = \{W^{(l)}, b^{(l)}\}_l$. For fixed network parameters θ, the probability $p(f|X,\theta)$ is given by enforcing the network architecture Eq. (3.1) with Dirac δ-distributions as

$$
\begin{aligned}
p(f|X,\theta) = \prod_{\alpha=1}^{P} \int \mathrm{d}h_\alpha^{(0)} \ldots \int \mathrm{d}f_\alpha\, & \delta\left(f_\alpha - W^{(L)} \phi\left(h_\alpha^{(L-1)}\right) - b^{(L)}\right) \\
& \times \prod_{l=1}^{L-1} \delta\Big(h_\alpha^{(l)} - W^{(l)}\, \phi\left(h_\alpha^{(l-1)}\right) - b^{(l)}\Big) \\
& \times \delta\left(h_\alpha^{(0)} - W^{(0)} x_\alpha - b^{(0)}\right).
\end{aligned} \tag{3.6}
$$

We compute the marginalization over network parameters in Eq. (3.5) of the second term as $p(f|X) = \int d\theta\, p(f|X;\theta)$, giving

$$p(f|X) = \prod_{l=0}^{L-1} \int \mathrm{d}h^{(l)} \int \mathrm{d}f \prod_{\alpha=1}^{P} \left\langle \delta\left(f_\alpha - W^{(L)}\phi\left(h_\alpha^{(L-1)}\right) - b^{(L)}\right)\right\rangle_{W^{(L)},b^{(L)}} \tag{3.7}$$

$$\times \prod_{l=1}^{L-1} \left\langle \delta\left(h_\alpha^{(l)} - W^{(l)}\,\phi\left(h_\alpha^{(l-1)}\right) - b^{(l)}\right)\right\rangle_{W^{(l)},b^{(l)}} \tag{3.8}$$

$$\times \left\langle \delta\left(h_\alpha^{(0)} - W^{(0)}x_\alpha - b^{(0)}\right)\right\rangle_{W^{(0)},b^{(0)}} . \tag{3.9}$$

Here $\langle \dots \rangle_{\{W,b\}}$ refers to the Gaussian average over weights W and biases b. To marginalize over weights and biases, we rewrite the Dirac δ-distributions using their Fourier transform $\delta(h) = \int \mathcal{D}\tilde{h} \exp(h\tilde{h})$ with $\int \mathcal{D}\tilde{h} = \frac{1}{2\pi i}\int_{-i\infty}^{i\infty} d\tilde{h}$, yielding

$$\begin{aligned} &\delta\left(h_{k,\alpha}^{(l)} - \sum_{j=1}^{N} W_{kj}^{(l)}\phi\left(h_{j,\alpha}^{(l-1)}\right) - b_k^{(l)}\right) \\ &= \int d\tilde{h}_{k,\alpha} \exp\left(h_{k,\alpha}^{(l)}\tilde{h}_{k,\alpha}^{(l)} - \tilde{h}_{k,\alpha}^{(l)}\sum_{j=1}^{N} W_{kj}^{(l)}\phi\left(h_{j,\alpha}^{(l-1)}\right) - \tilde{h}_{k,\alpha}^{(l)}b_k^{(l)}\right), \end{aligned} \tag{3.10}$$

As a result of the Fourier transform, we introduce conjugate variables $\tilde{f}$ for the network output f and $\tilde{h}^{(l)}$ for the layer pre-activations $h^{(l)}$. When we perform the Gaussian expectation values over network parameters $\theta = \{W^{(l)}, b^{(l)}\}_l$, the moment-generating function (MGF) of these variables appears naturally; for a centered Gaussian it computes to $\langle \exp(k\,\theta_{ij})\rangle_{\theta_{ij}\sim\mathcal{N}(0,\sigma^2)} = \exp\left(\sigma^2/2\,k^2\right)$. For the input layer, this yields

$$\prod_\alpha \left\langle \exp\left(\sum_{i,j} \tilde{h}_{i,\alpha}^{(0)} W_{ij}^{(0)} x_{j,\alpha} + \tilde{h}_{i,\alpha}^{(0)} b_i^{(0)}\right)\right\rangle_{W^{(0)},b^{(0)}} \tag{3.11}$$

$$= \left\langle \exp\left(\sum_{\alpha,i,j} \tilde{h}_{i,\alpha}^{(0)} W_{ij}^{(0)} x_{j,\alpha}\right)\right\rangle_{W^{(0)}} \left\langle \exp\left(\sum_{\alpha,i} \tilde{h}_{i,\alpha}^{(0)} b_i^{(0)}\right)\right\rangle_{b^{(0)}} \tag{3.12}$$

$$\overset{\mathrm{MGF}}{=} \exp\left(\frac{\sigma_w^2}{2D}\sum_{\alpha\beta}\left(\tilde{h}_\alpha^{(0)}\right)^{\mathsf{T}}[XX^{\mathsf{T}}]_{\alpha\beta}\tilde{h}_\beta^{(0)}\right)\exp\left(\frac{\sigma_b^2}{2}\sum_{\alpha\beta}\left(\tilde{h}_\alpha^{(0)}\right)^{\mathsf{T}}\tilde{h}_\beta^{(0)}\right), \tag{3.13}$$

where $\left[XX^{\mathsf{T}}\right]_{\alpha\beta} = x_\alpha^{\mathsf{T}} x_\beta = x_\alpha \cdot x_\beta$ denotes the input overlaps. All subsequent network layers give

$$\prod_\alpha \left\langle \exp\left(\sum_{i,j} \tilde{h}_{i,\alpha}^{(l)}\, W_{ij}^{(l)} \phi_{j,\alpha}^{(l-1)} + \sum_i \tilde{h}_{i,\alpha}^{(l)} b_i^{(l)}\right)\right\rangle_{W^{(l)},b^{(l)}} \tag{3.14}$$

$$= \left\langle \exp\left(\sum_{\alpha,i,j} \tilde{h}_{i,\alpha}^{(l)} W_{ij}^{(l)} \phi_{j,\alpha}^{(l-1)}\right)\right\rangle_{W^{(l)}} \left\langle \exp\left(\sum_{\alpha,i} \tilde{h}_{i,\alpha}^{(l)} b_i^{(l)}\right)\right\rangle_{b^{(l)}} \tag{3.15}$$

$$\overset{\text{MGF}}{=} \exp\left(\frac{\sigma_w^2}{2N}\sum_{\alpha\beta}\left(\tilde{h}_\alpha^{(l)}\right)^\mathsf{T}\left[\phi^{(l-1)}\phi^{(l-1)\mathsf{T}}\right]_{\alpha\beta}\tilde{h}_\beta^{(l)}\right)\exp\left(\frac{\sigma_b^2}{2}\sum_{\alpha\beta}\left(\tilde{h}_\alpha^{(l)}\right)^\mathsf{T}\tilde{h}_\beta^{(l)}\right), \quad (3.16)$$

where we introduce $\left[\phi^{(l-1)}\phi^{(l-1)\mathsf{T}}\right]_{\alpha\beta} \coloneqq \sum_j \phi_{j,\alpha}^{(l-1)}\phi_{j,\beta}^{(l-1)}$ with shorthand $\phi_{j,\alpha}^{(l-1)} = \phi\left(h_{j,\alpha}^{(l-1)}\right)$.

Auxiliary variables

Quadratic terms in h and $\tilde{h}$ can be solved as Gaussian integrals. However, in the above expressions terms we also have terms proportional to $\propto \left[\tilde{h}^l\right]^\mathsf{T}\tilde{h}^l\,\phi(h^{l-1})^\mathsf{T}\phi(h^{l-1})$, which are at least quartic in h and $\tilde{h}$. To treat these terms, we introduce auxiliary variables

$$C_{\alpha\beta}^{(0)} \coloneqq \frac{\sigma_w^2}{D}[XX^\mathsf{T}]_{\alpha\beta} + \sigma_b^2, \quad (3.17)$$

$$C_{\alpha\beta}^{(l)} \coloneqq \frac{\sigma_w^2}{N}\left[\phi^{(l-1)}\phi^{(l-1)\mathsf{T}}\right]_{\alpha\beta} + \sigma_b^2 \quad l = 1, \ldots, L, \quad (3.18)$$

so that the appearing terms simplify to

$$\int \mathcal{D}\tilde{h}_{i,\alpha}^{(l)} \exp\left(-\tilde{h}_{i,\alpha}^{(l)}h_{i,\alpha}^{(l)} + \frac{1}{2}\sum_{\alpha\beta}\tilde{h}_{i,\alpha}^{(l)}C_{\alpha\beta}^{(l)}\tilde{h}_{i,\beta}^{(l)}\right) = \mathcal{N}\left(h_i^{(l)}|0, C_{\alpha\beta}^{(l)}\right) \quad 0 \le l < L. \quad (3.19)$$

We enforce the definition of the auxiliary variables using Dirac δ-distributions

$$\begin{aligned}&\delta\left(-C_{\alpha\beta}^{(l)} + \frac{\sigma_w^2}{N}[\phi^{(l-1)}\phi^{(l-1)\mathsf{T}}]_{\alpha\beta} + \sigma_b^2\right) \quad (3.20)\\ &= \int \mathcal{D}\tilde{C}_{\alpha\beta}^{(l)} \exp\left(-\tilde{C}_{\alpha\beta}^{(l)}C_{\alpha\beta}^{(l)} + \tilde{C}_{\alpha\beta}^{(l)}\frac{\sigma_w^2}{N}[\phi^{(l-1)}\phi^{(l-1)\mathsf{T}}]_{\alpha\beta} + \tilde{C}_{\alpha\beta}^{(l)}\sigma_b^2\right),\end{aligned}$$

where $\int \mathcal{D}\tilde{C}_{\alpha\beta}^{(l)} = \int_{-i\infty}^{i\infty}\frac{d\tilde{C}_{\alpha\beta}^{(l)}}{2\pi i}$ and $\tilde{C}_{\alpha\beta}$ is the conjugate variable of $C_{\alpha\beta}$. Combining all terms in Eq. (3.7) yields

$$\begin{aligned}p(f|X) &= \int \mathcal{D}\{\tilde{C}, C\}\,\mathcal{N}\left(f|0, C_{\alpha\beta}^{(L)}\right)\left\langle\exp(\mathcal{S}(C, \tilde{C}))\right\rangle_h, \quad (3.21)\\ \mathcal{S}(C, \tilde{C}) &= -\sum_{l=1}^{L}\tilde{C}_{\alpha\beta}^{(l)}C_{\alpha\beta}^{(l)} + \tilde{C}_{\alpha\beta}^{(l)}\left(\frac{\sigma_w^2}{N}[\phi^{(l-1)}\phi^{(l-1)\mathsf{T}}]_{\alpha\beta} + \sigma_b^2\right),\end{aligned}$$

where $\int \mathcal{D}\{\tilde{C}, C\} = \prod_l \prod_{\alpha,\beta}\int \mathcal{D}\tilde{C}_{\alpha\beta}^{(l)}\int dC_{\alpha\beta}^{(l)}$, the expectation value $\langle\ldots\rangle_h$ goes over all hidden layers $h^{(l)} \sim \mathcal{N}\left(0, C_{\alpha\beta}^{(l)}\right)$ and repeated indices α, β are summed over.

Since these distributions are independent across neuron indices j, averages reduce to

$$\left\langle \exp\left(\tilde{C}^{(l)}_{\alpha\beta}\frac{\sigma_w^2}{N}\sum_{j=1}^{N}\phi^{(l-1)}_{j,\alpha}\phi^{(l-1)}_{j,\beta}\right)\right\rangle_{\{\mathcal{N}\left(h_j^{(l-1)}|0,C^{(l-1)}_{\alpha\beta}\right)\}_j} \tag{3.22}$$

$$\overset{h_j^{(l-1)}\text{ i.i.d. in}\,j}{=} \left\langle \exp\left(\frac{\sigma_w^2}{N}\tilde{C}^{(l)}_{\alpha\beta}\phi^{(l-1)}_{\alpha}\phi^{(l-1)}_{\beta}\right)\right\rangle^{N}_{\mathcal{N}\left(h^{(l-1)}|0,C^{(l-1)}_{\alpha\beta}\right)}. \tag{3.23}$$

Thus, we obtain

$$p(f|X) = \int \mathcal{D}\{\tilde{C},C\}\,\mathcal{N}\left(f|0,C^{(L)}_{\alpha\beta}\right)\,\exp\big(-\sum_{l=1}^{L}\tilde{C}^{(l)}_{\alpha\beta}C^{(l)}_{\alpha\beta} + \mathcal{W}(\tilde{C}|C)\big), \tag{3.24}$$

$$\mathcal{W}(\tilde{C}|C) = \sum_{l=1}^{L}\sum_{\alpha\beta}\tilde{C}^{(l)}_{\alpha\beta}\sigma_b^2 + N\sum_{l=1}^{L}\ln\big\langle \exp\big(\frac{\sigma_w^2}{N}\tilde{C}^{(l)}_{\alpha\beta}\phi^{(l-1)}_{\alpha}\phi^{(l-1)}_{\beta}\big)\big\rangle_{\mathcal{N}(0,C^{(l-1)})}, \tag{3.25}$$

$$C^{(0)} = \frac{\sigma_w^2}{D}XX^{\mathsf{T}} + \sigma_b^2. \tag{3.26}$$

Network prior as a superposition of Gaussians

Since we assume regularization noise on the labels, the network prior reads

$$p(Y|X) = \int \mathcal{D}\{\tilde{C},C\}\prod_{\alpha} df_{\alpha}\,\mathcal{N}(y_{\alpha}|f_{\alpha},\kappa)\,\mathcal{N}\left(f_{\alpha}|0,C^{(L)}_{\alpha\beta}\right)\,\exp(-\mathrm{tr}\,\tilde{C}^{\mathsf{T}}C + \mathcal{W}(\tilde{C}|C)).$$

Here we use the shorthand $\mathrm{tr}\,\tilde{C}^{\mathsf{T}}C = \sum_{\alpha\beta l}\tilde{C}^{(l)}_{\alpha\beta}C^{(l)}_{\alpha\beta}$. The integral over f_{α} is a convolution of the Gaussian distributions $\mathcal{N}(y_{\alpha}|f_{\alpha},\kappa)$ and $\mathcal{N}\left(f_{\alpha}|0,C^{(L)}_{\alpha\beta}\right)$; in consequence their covariances add to $C^{(L)}_{\alpha\beta} + \kappa\mathbb{I}$.
We obtain the final expression for the network prior

$$p(Y|X) = \int \mathcal{D}C\,\mathcal{N}\left(Y|0,C^{(L)} + \kappa\mathbb{I}\right)\,p(C), \tag{3.27}$$

$$p(C) = \int \mathcal{D}\tilde{C}\,\exp\left(-\mathrm{tr}\,\tilde{C}^{\mathsf{T}}C + \mathcal{W}(\tilde{C}|C)\right). \tag{3.28}$$

From this expression one can read off that the network output is a superposition of centered Gaussians $\mathcal{N}\left(0,C^{(L)} + \kappa\mathbb{I}\right)$. The covariance is given by the label noise $\kappa\mathbb{I}$ and $C^{(L)}$ that itself is distributed as $p(C^{(L)}) = \int dC^{(1\le l<L)}\,p(C)$. The joint distribution $p(C) = p(C^{(L)}|C^{(L-1)})\cdots p(C^{(1)}|C^{(0)})$ of all $C^{(1\le l\le L)}$ decomposes into a chain of conditionals with

$$p\left(C^{(l)}|C^{(l-1)}\right) = \int \mathcal{D}\tilde{C}^{(l)}\exp\left(-\mathrm{tr}\,\tilde{C}^{(l)\mathsf{T}}C^{(l)} + \mathcal{W}\left(\tilde{C}^{(l)}|C^{(l-1)}\right)\right), \tag{3.29}$$

$$\mathcal{W}\left(\tilde{C}^{(l)}|C^{(l-1)}\right) = \tilde{C}^{(l)}\sigma_b^2 + N\ln\big\langle \exp\big(\frac{\sigma_w^2}{N}\phi^{(l-1)\mathsf{T}}\tilde{C}^{(l)}\phi^{(l-1)}\big)\big\rangle_{\mathcal{N}(0,C^{(l-1)})} \quad 1\le l<L. \tag{3.30}$$

The distribution $p(C^{(L)})$ is written in terms of its cumulant-generating function

$$\mathcal{W}(\tilde{C}|C) = N \sum_{l=0}^{L-1} \ln \left\langle \exp\left(\frac{\sigma_w^2}{N} \phi^{(l)\mathsf{T}} \tilde{C}^{(l+1)} \phi^{(l)} \right) \right\rangle_{\mathcal{N}(0,C^{(l)})} + \tilde{C}\sigma_w^2 + \tilde{C}^{(0)\mathsf{T}} C^{(0)}, \tag{3.31}$$

with $\phi^{\mathsf{T}}\tilde{C}\phi = \sum_{\alpha\beta} \phi_\alpha \tilde{C}_{\alpha\beta} \phi_\beta$. The formulation of the network prior in Eq. (3.27) as a superposition of Gaussians is exact.

Saddle point approximation recovers NNGP kernel

We can write Eq. (3.27) in the form

$$p(Y|X) = \int \mathcal{D}C \int \mathcal{D}\tilde{C} \mathcal{N}\left(Y|0, C^{(L)} + \kappa \mathbb{I}\right) \exp\left(S(C,\tilde{C})\right), \tag{3.32}$$

with an action $S(C,\tilde{C}) = -\mathrm{tr}\,\tilde{C}^{\mathsf{T}}C + \mathcal{W}(\tilde{C}|C)$. The action $\mathcal{S}$ scales with the network width N. In the limit of infinite width $N \to \infty$, we can thus perform a saddle point approximation to evaluate integrals of the form

$$\int \mathcal{D}C \int \mathcal{D}\tilde{C} f(C,\tilde{C}) \exp\left(\mathcal{S}(C,\tilde{C})\right) \overset{N\to\infty}{=} f(C_*, \tilde{C}_*), \tag{3.33}$$

where C_* and $\tilde{C}_*$ are the saddle points of the action $\mathcal{S}$. We compute these using the conditions

$$\frac{\partial \mathcal{S}}{\partial C} \overset{!}{=} 0, \ \frac{\partial \mathcal{S}}{\partial \tilde{C}} \overset{!}{=} 0, \tag{3.34}$$

and recover the NNGP

$$C_*^{(0)} = \frac{\sigma_w^2}{D} X X^{\mathsf{T}} + \sigma_b^2, \tag{3.35}$$

$$C_*^{(l)} = \sigma_w^2 \langle \phi^{(l-1)} \phi^{(l-1)} \rangle_{\mathcal{N}(0,C^{(l-1)})} + \sigma_b^2 \quad l = 1, \ldots, L, \tag{3.36}$$

$$\tilde{C}_*^{(l)} = 0. \tag{3.37}$$

Thus, the network prior reduces to a single Gaussian with the NNGP kernel

$$p(Y|X) = \mathcal{N}\left(Y|0, C_*^{(L)} + \kappa \mathbb{I}\right). \tag{3.38}$$

3.3.4 Next-to-leading-order corrections

For infinitely wide networks $N \to \infty$, the auxiliary variables in Eq. (3.18) concentrate around the NNGP kernel. For finite-width networks, the auxiliary variables fluctuate around their saddle points. The above field-theoretic framework allows computing next-to-leading-order corrections as in Segadlo et al. (2022). To lowest-order, the auxiliary variables fluctuate in a Gaussian manner

$$\begin{aligned} p(Y|X) &\simeq \int \mathcal{D}\delta C \int \mathcal{D}\delta\tilde{C}\, \exp\Big(\frac{1}{2}(\delta C, \delta\tilde{C})^{\mathsf{T}} \mathcal{S}^{(2)}\,(\delta C, \delta\tilde{C})\Big) \\ &= \int \mathcal{D}\delta C \int \mathcal{D}\delta\tilde{C}\, \exp\Big(-\frac{1}{2}(\delta C, \delta\tilde{C})^{\mathsf{T}} \Big[\Delta^{(2)}\Big]^{-1}\,(\delta C, \delta\tilde{C})\Big), \end{aligned}$$

where $\delta C = C - C_*$ and $\delta\tilde{C} = \tilde{C} - \tilde{C}_*$. We obtain these fluctuations by computing the negative inverse of the Hessian of the action

$$\begin{pmatrix} \langle \delta C\, \delta C \rangle & \langle \delta C\, \delta\tilde{C} \rangle \\ \langle \delta\tilde{C}\, \delta C \rangle & \langle \delta\tilde{C}\, \delta\tilde{C} \rangle \end{pmatrix} = \Big[-\mathcal{S}^{(2)}\Big]^{-1} = \Delta^{(2)}. \tag{3.39}$$

The terms of the Hessian are given by

$$\begin{aligned} &\frac{\partial^2}{\partial C^{(l)}_{\alpha\beta} \partial C^{(m)}_{\gamma\delta}} \mathcal{S}|_{(C_*,0)} = 0, \\ &\frac{\partial^2}{\partial C^{(l)}_{\alpha\beta} \partial \tilde{C}^{(m)}_{\gamma\delta}} \mathcal{S}|_{(C_*,0)} = -\delta_{l,m}\, \delta_{(\alpha\beta),(\gamma\delta)} + \delta_{m-1,l}\, \sigma_w^2\, \frac{\partial \langle \phi^{(m-1)}_\gamma \phi^{(m-1)}_\delta \rangle_{\mathcal{N}(0,C^{(l)}_*)}}{\partial C^{(l)}_{\alpha\beta}}, \\ &\frac{\partial^2}{\partial \tilde{C}^{(l)}_{\alpha\beta} \partial \tilde{C}^{(m)}_{\gamma\delta}} \mathcal{S}|_{(C_*,0)} = \delta_{l,m}\, \frac{\sigma_w^4}{N}\, \langle \phi^{(l-1)}_\alpha \phi^{(l-1)}_\beta, \phi^{(l-1)}_\gamma \phi^{(l-1)}_\delta \rangle^c_{\mathcal{N}(0,C^{(l)}_*)}. \end{aligned}$$

Here appears the connected correlation function

$$\begin{aligned} \langle \phi^{(l-1)}_\alpha \phi^{(l-1)}_\beta, \phi^{(l-1)}_\gamma \phi^{(l-1)}_\delta \rangle^c_{\mathcal{N}(0,C^{(l)}_*)} \coloneqq{}& \langle \phi^{(l-1)}_\alpha \phi^{(l-1)}_\beta \phi^{(l-1)}_\gamma \phi^{(l-1)}_\delta \rangle_{\mathcal{N}(0,C^{(l)}_*)} \\ &- \langle \phi^{(l-1)}_\alpha \phi^{(l-1)}_\beta \rangle_{\mathcal{N}(0,C^{(l)}_*)} \langle \phi^{(l-1)}_\gamma \phi^{(l-1)}_\delta \rangle_{\mathcal{N}(0,C^{(l)}_*)}. \end{aligned}$$

Using the block structure of the Hessian $\mathcal{S}^{(2)} = \begin{pmatrix} \mathcal{S}_{11} & \mathcal{S}_{12} \\ \mathcal{S}_{21} & \mathcal{S}_{22} \end{pmatrix}$ and that $\mathcal{S}_{11} = 0$, we compute its negative inverse as

$$\Delta_{11} = \Delta_{12}\, \mathcal{S}_{22}\, \Delta_{21}, \tag{3.40}$$

$$\Delta_{12} = -\mathcal{S}_{21}^{-1}, \tag{3.41}$$

$$\Delta_{22} = 0. \tag{3.42}$$

This yields for the off-diagonal terms

$$\Delta_{12}^{(lm,\alpha\beta)} = \delta_{lm} + 1_{l>m}\sigma_w^2 \prod_{s=m+1}^{l} \Big\langle \Big(\phi^{(s)}_\alpha\Big)' \Big(\phi^{(s)}_\beta\Big)' \Big\rangle_{\mathcal{N}(0,C^{(s)}_*)},$$

which corresponds to $\Delta_{12}^{(lm,\alpha\beta)} = \mathrm{Cov}\left(C^{(l)}, \tilde{C}^{(m)}\right)$. Since the auxiliary fields $\tilde{C}^{(l)}$ represent changes in the fields $C^{(l)}$, it can be understood as the response of the network residual in layer l to a perturbation of the kernel in layer m. Due to the feed-forward architecture any response can only propagate forward in the network, which is reflected by the indicator function $1_{k>l}$. While the statistics of all intermediate layers are determined by $(C, \tilde{C}) \sim \exp\big(S(C, \tilde{C})\big)$, we can describe the effect of variability in the input layer $l = 0$ by

$$C^{(l)} = C_*^{(l)} + \Delta_{12}^{(l0)}\, \delta C^{(0)} + \mathcal{O}(\delta^2). \tag{3.43}$$

This can also be understood in terms of linear response theory since the response function can be written as $\Delta_{12}^{(l0)} = \frac{\partial}{\partial C^{(0)}} C^{(l)}|_{C_*}$.

The diagonal terms of the covariance Δ are given by

$$\Delta_{22}^{(lm,\alpha\beta\gamma\delta)} = \sum_{k=1}^{l} \Delta_{12}^{(lk,\alpha\beta)}\, \sigma_w^4 \langle \phi_\alpha^{(k-1)} \phi_\beta^{(k-1)}, \phi_\gamma^{(k-1)} \phi_\delta^{(k-1)} \rangle^c_{\mathcal{N}(0,C_*^{(l)})}\, \Delta_{21}^{(km,\gamma\delta)},$$

where the fluctuations $\sigma_w^4 \langle \phi_\alpha^{(k-1)} \phi_\beta^{(k-1)}, \phi_\gamma^{(k-1)} \phi_\delta^{(k-1)} \rangle^c_{\mathcal{N}(0,C_*^{(l)})}$ of the kernels $C^{(l)}$ are convolved with the response function and summed over all layers.

3.3.5 Criticality in neural networks

Here we discuss properties of the response function, which exhibits long-range behavior across layers at critical points of the network. In this section, we follow the presentation in Schoenholz et al. (2017). They show that for infinite depth $L \to \infty$, the kernels converge to a fixed point for all inputs x_α that can be determined self-consistently

$$q_{\alpha\alpha}^* = \sigma_w^2 \langle \phi(h)\, \phi(h) \rangle_{h \sim \mathcal{N}(0, q_{\alpha\alpha}^*)} + \sigma_b^2, \tag{3.44}$$

$$q_{\alpha\beta}^* = \sigma_w^2 \langle \phi(h^{(1)})\, \phi(h^{(2)}) \rangle_{(h^{(1)}, h^{(2)}) \sim \mathcal{N}(0,Q)} + \sigma_b^2, \tag{3.45}$$

where $Q = \begin{pmatrix} q_{\alpha\alpha}^* & q_{\alpha\beta}^* \\ q_{\beta\alpha}^* & q_{\beta\beta}^* \end{pmatrix}$ with $q_{\alpha\alpha}^* = q_{\beta\beta}^*$ and $q_{\alpha\beta}^* = q_{\beta\alpha}^*$ since the fixed point is independent of the sample index. Depending on the hyperparameters σ_w^2 and σ_b^2, the correlation $c_{\alpha\beta}^{(l)} = C_{\alpha\beta}^{(l)} / \sqrt{C_{\alpha\alpha}^{(l)} C_{\beta\beta}^{(l)}}$ converges to a fixed point $c^* = q_{\alpha\beta}^* / q_{\alpha\alpha}^*$ that is either $c^* = 1$ or $c^* < 1$. These two cases define two different phases in the network dynamics: For $c^* = 1$, all signals $h_\alpha^{(l)}$ for different inputs x_α correlate, which corresponds to the ordered phase. For $c^* < 1$, all signals $h_\alpha^{(l)}$ for different inputs x_α decorrelate, which correponds to the chaotic phase. Points (σ_w^2, σ_b^2) in hyperparameters space where the network transitions from one phase to the other are so-called critical points, following the notion of criticality in dynamical systems. To determine when the fixed point

$c^* = 1$ is stable, one calculates

$$\chi_1 = \frac{\partial c_{\alpha\beta}^{(l)}}{\partial c_{\alpha\beta}^{(l-1)}} = \sigma_w^2 \left\langle \phi' \phi' \right\rangle_{\mathcal{N}(0,q_*)}, \tag{3.46}$$

For $\chi_1 < 1$, the fixed point $c^* = 1$ is stable and the system is in the ordered phase. For $\chi_1 > 1$, the system converges to the fixed point $c^* < 1$ corresponding to the chaotic phase. Thus, $\chi_1 = 1$ marks the order-chaos transition.

The decay to the fixed point can be described in terms of the response function as

$$C_{\alpha\beta}^{(l)} \approx q^* + \Delta_{12}^{(l0,\alpha\beta)} C_{\alpha\beta}^{(0)}. \tag{3.47}$$

At large depth, the response function shows exponential behavior

$$\begin{aligned}
\Delta_{12}^{(l0,\alpha\beta)} &= \prod_{s=1}^{l} \sigma_w^2 \left\langle \left(\phi_\alpha^{(s)}\right)' \left(\phi_\beta^{(s)}\right)' \right\rangle_{\mathcal{N}(0,C_*^{(s)})} \\
&\approx \prod_{s=1}^{l} \sigma_w^2 \left\langle \left(\phi_\alpha^{(s)}\right)' \left(\phi_\beta^{(s)}\right)' \right\rangle_{\mathcal{N}(0,Q)} \\
&= \left[\sigma_w^2 \left\langle \left(\phi_\alpha^{(s)}\right)' \left(\phi_\beta^{(s)}\right)' \right\rangle_{\mathcal{N}(0,Q)}\right]^l \\
&= \exp\left(-l/\xi_c\right),
\end{aligned}$$

where $\xi_c^{-1} = -\ln\left(\sigma_w^2 \left\langle \left(\phi_\alpha^{(s)}\right)' \left(\phi_\beta^{(s)}\right)' \right\rangle_{\mathcal{N}(0,Q)}\right)$ is the depth scale that controls how far information about the inputs propagates within the network. In the ordered phase, $c^* = 1$ and the integral simplifies to $\xi_c^{-1} = \ln \chi_1$. At the transition to chaos $\chi_1 = 1$ and the depth scale diverges, leading to information propagation to arbitrary depths in the network. Thus, initializing infinitely wide networks at critical points might be beneficial for network training, which is also called the edge-of-chaos initialization (Schoenholz et al., 2017).

3.3.6 Large deviation theory

Large deviation theory studies the asymptotic tail behavior of probability distributions and is based on the idea that probability distributions $p(x)$ can be expressed in terms of entropy functions $\Gamma(x)$ (Touchette, 2009). This idea takes form in the so-called large deviation principle $p(x) \simeq \exp(-N\,\Gamma(x))$ where N is a large parameter, e.g. the system size, and Γ is called a rate function in the context of large deviation theory. There are different theorems yielding a large deviation principle; we here present the Gärtner-Ellis theorem which will appear in Sec. 3.4.

For a real random variable X_N that is parameterized by a positive integer N, we define the scaled cumulant-generating function as

$$\lambda(k) = \lim_{N\to\infty} \frac{1}{N} \ln\langle \exp(NkX_N)\rangle_{X_N} \tag{3.48}$$

where $k \in \mathbb{R}$. If $\lambda(k)$ exists and is differentiable, then X_N follows a large deviation principle so that

$$p(X_N) \simeq \exp(-N\Gamma(X_N)) \tag{3.49}$$

with the rate function given by the Legendre transform of the scaled cumulant-generating function

$$\Gamma(X_N) = \sup_{k\in\mathbb{R}} kX_N - \lambda(k). \tag{3.50}$$

Note that the Gärtner-Ellis theorem applies to a one-dimensional random variable.

To make the large deviation principle more tangible, we study the example of a sum of Gaussian random i.i.d. variables $S_N = \frac{1}{N}\sum_{i=1}^N Y_i$ where $Y_i \overset{\text{i.i.d}}{\sim} \mathcal{N}(\mu, \sigma^2)$. The scaled cumulant-generating function is then

$$\begin{aligned}\lambda(k) &= \lim_{N\to\infty} \frac{1}{N} \ln\langle \exp(NkS_N)\rangle_{S_N}\\ &= \lim_{N\to\infty} \frac{1}{N} \ln \prod_{i=1}^{N} \langle \exp(kY_i)\rangle_{Y_i}\\ &= \ln\langle \exp(kY)\rangle_Y.\end{aligned}$$

We get the cumulant-generating function of the individual Gaussians as $\ln\langle\exp(kY)\rangle_Y = \mu k + \frac{1}{2}\sigma^2 k^2$, fulfilling the conditions of the Gärtner-Ellis theorem. For the rate function we obtain

$$\Gamma(S_N) = \frac{(S_N - \mu)^2}{2\sigma^2}, \tag{3.51}$$

thus recovering Cramer's theorem.

3.4 Theory

We here present a theoretic framework to describe the posterior of the network in a Bayesian setting. This theory captures non-linear kernel adaptation in trained networks.

3.4.1 Large deviation approach for the posterior kernels

The cumulant-generating function in Eq. (3.31) scales as $\mathcal{W}(\tilde{C}) = N\lambda(\tilde{C}/N)$ with the network width N, where the function λ itself does not depend on N. Thus λ

follows a scaling form (Touchette, 2009) and in consequence, the auxiliary variables C concentrate in the limit $N \to \infty$ around its mean while cumulants of order $k > 1$ are suppressed with $1/N^{k-1}$. For finite network width N, all auxiliary variables $C^{(1 \le l \le L)}$ fluctuate with a variance of order $\mathcal{W}'' \sim \mathcal{O}(N^{-1})$; except for the input kernel $C^{(0)}$, which is deterministic. As the cumulant-generating function has a scaling form, we can do a saddle point approximation for the integrals over the conjugate variables $\tilde{C}^{(l)}$ as

$$\begin{aligned} -\ln p(C^{(l+1)}|C^{(l)}) &= \ln \int \mathcal{D}\tilde{C}^{(l)} \exp\left(\operatorname{tr} \tilde{C}^{(l+1)\mathsf{T}} C^{(l+1)} - \mathcal{W}(\tilde{C}^{(l+1)}|C^{(l)})\right) && (3.52) \\ &\simeq \sup_{\tilde{C}^{(l+1)}} \operatorname{tr} \tilde{C}^{(l+1)\mathsf{T}} C^{(l+1)} - \mathcal{W}(\tilde{C}^{(l+1)}|C^{(l)}) && (3.53) \\ &= \Gamma(C^{(l+1)}|C^{(l)}). \end{aligned}$$

Here, Γ can be understood as a rate function that one would get by using the Gärtner-Ellis theorem on a one-dimensional variable to get a large deviation principle (l.d.p.) (Touchette, 2009). However, one needs to be careful in which limits this saddle point approximation is valid, whether correction terms become non-negligible or even dominate the integral. For linear networks, the rate function takes an intuitive form as the Kullback-Leibler divergence between the Gaussian distributions of the pre-activations of two adjacent layers (see App. B.2), indicating that the network prior keeps the distributions in adjacent layers similar. We get the joint probability $p(C)$ in Eq. (3.28) across all layers as

$$\begin{aligned} \ln p(C) &= \ln p(C^{(L)}|C^{(L-1)}) \cdots p(C^{(1)}|C^{(0)}) \\ &\simeq -\sum_{l=1}^{L} \Gamma(C^{(l)}|C^{(l-1)}) =: -\Gamma(C). \end{aligned}$$

We use the rate function to write the network prior $p(Y|X)$ as

$$p(Y|X) \simeq \int \mathcal{D}C \mathcal{N}\left(Y|0, C^{(L)} + \kappa \mathbb{I}\right) \exp\left(-\Gamma(C)\right). \qquad (3.54)$$

From the saddle point in Eq. (3.52) follows

$$C^{(l+1)}_{\alpha\beta} \equiv \frac{\partial \mathcal{W}}{\partial \tilde{C}^{(l+1)}_{\alpha\beta}} = \sigma_w^2 \left\langle \phi^{(l)}_\alpha \phi^{(l)}_\beta \right\rangle_{\mathcal{P}^{(l)}} + \sigma_b^2, \qquad (3.55)$$

$$\left\langle \dots \right\rangle_{\mathcal{P}^{(l)}} \propto \left\langle \dots \exp\left(\frac{\sigma_w^2}{N} \phi^{(l)\mathsf{T}} \tilde{C}^{(l+1)} \phi^{(l)}\right)\right\rangle_{\mathcal{N}(0,C^{(l)})}, \qquad (3.56)$$

where the expectation value is with respect to the non-Gaussian measure

$$\langle \dots \rangle_{\mathcal{P}^{(l)}} \equiv \langle \dots \rangle_{h^{(l)} \sim \mathcal{P}(\tilde{C}^{(l+1)}, C^{(l)})}, \qquad (3.57)$$

with proper normalization given by $\left\langle \exp\left(\frac{\sigma_w^2}{N} \phi^{(l)\mathsf{T}} \tilde{C}^{(l+1)} \phi^{(l)}\right) \right\rangle_{\mathcal{N}(0,C^{(l)})}$.

We obtain the posterior distribution of the kernels C as

$$p(C|Y) \propto p(Y,C) \equiv \mathcal{N}\left(Y|0, C^{(L)} + \kappa\mathbb{I}\right) p(C), \tag{3.58}$$

where we conditioned on the training data $\{(x_\alpha, y_\alpha)\}_\alpha$. Then, the posterior estimates for the auxiliary variables C are given by the stationary points of the action

$$\begin{aligned} \mathcal{S}(C) &\coloneqq \ln p(C|Y) \simeq \mathcal{S}_\mathrm{D}(C^{(L)}) - \Gamma(C) + \circ, \\ \mathcal{S}_\mathrm{D}(C^{(L)}) &\coloneqq -\frac{1}{2} Y^\mathsf{T} (C^{(L)} + \kappa\mathbb{I})^{-1} Y - \frac{1}{2} \ln\det(C^{(L)} + \kappa\mathbb{I}). \end{aligned} \tag{3.59}$$

Here, we approximate $p(C)$ by its rate function in Eq. (3.52) and drop any terms $\circ$ constant in C. The action $S(C)$ consists of two terms: the rate function $-\Gamma(C) \sim \mathcal{O}(N)$ from the network prior and the log-likelihood of the training labels $\mathcal{S}_\mathrm{D}(C^{(L)}) \sim \mathcal{O}(P)$ from conditioning on the training data. As the log-likelihood depends only on the auxiliary variable $C^{(L)}$ in the output, its stationary point $\partial S(C)/\partial C^{(L)} \overset{!}{=} 0$ results from a trade-off between the network prior term in the form of Γ and the data term $\mathcal{S}_\mathrm{D}$, yielding

$$\tilde{C}^{(L)} = \frac{1}{2}(C^{(L)} + \kappa\mathbb{I})^{-1} Y Y^\mathsf{T} (C^{(L)} + \kappa\mathbb{I})^{-1} - \frac{1}{2}(C^{(L)} + \kappa\mathbb{I})^{-1}. \tag{3.60}$$

We obtain the conjugate kernel $\tilde{C}^{(L)}$ of the network output as a function of the network kernel $C^{(L)}$ and the training labels Y. Further, we show in App. B.1 that the conjugate kernel $\tilde{C}^{(L)}$ corresponds to the second moment of the discrepancies between labels and network output; its trace yields the training loss. While the input kernel $C^{(0)}$ is deterministic, we get for the kernels $C^{(l)}$ in the hidden network layers $1 \le l < L$ from the stationary condition $\partial S(C)/\partial C^{(l)} \overset{!}{=} 0$ that

$$\begin{aligned} \tilde{C}^{(l)}_{\alpha\beta} &= -\frac{\partial\Gamma(C^{(l+1)}|C^{(l)})}{\partial C^{(l)}_{\alpha\beta}} \overset{\text{Legendre}}{\equiv} \frac{\partial\mathcal{W}(\tilde{C}^{(l+1)}|C^{(l)})}{\partial C^{(l)}_{\alpha\beta}} \\ &\overset{\text{Price}}{=} \sigma_w^2\, \tilde{C}^{(l+1)}_{\alpha\beta} \left\langle \left(\phi^{(l)}_\alpha\right)' \left(\phi^{(l)}_\beta\right)' \right\rangle_{\mathcal{P}^{(l)}} + \delta_{\alpha\beta}\, \sigma_w^2\, \tilde{C}^{(l+1)}_{\alpha\alpha} \left\langle \left(\phi^{(l)}_\alpha\right)'' \phi^{(l)}_\alpha \right\rangle_{\mathcal{P}^{(l)}} \\ &\quad + 2\frac{\sigma_w^4}{N} \sum_{\gamma,\delta} \tilde{C}^{(l+1)}_{\alpha\gamma}\, \tilde{C}^{(l+1)}_{\beta\delta} \left\langle \left(\phi^{(l)}_\alpha\right)' \phi^{(l)}_\gamma \left(\phi^{(l)}_\beta\right)' \phi^{(l)}_\delta \right\rangle_{\mathcal{P}^{(l)}} \qquad (3.61) \\ &= \sigma_w^2\, \tilde{C}^{(l+1)}_{\alpha\beta} \left\langle \left(\phi^{(l)}_\alpha\right)' \left(\phi^{(l)}_\beta\right)' \right\rangle_{\mathcal{P}^{(l)}} + \delta_{\alpha\beta}\, \sigma_w^2\, \tilde{C}^{(l+1)}_{\alpha\alpha} \left\langle \left(\phi^{(l)}_\alpha\right)'' \phi^{(l)}_\alpha \right\rangle_{\mathcal{P}^{(l)}} + \mathcal{O}(N^{-1}), \\ &\qquad (3.62) \end{aligned}$$

where we used Price's theorem (Price, 1958; Papoulis and Pillai, 2002) and the fundamental property of the Legendre transform in Eq. (3.52) as well as dropped subleading terms $\propto \mathcal{O}(N^{-1})$ in the last line. Thus, we get the conjugate kernel $\tilde{C}^{(l)}$ in layer l as a function of the conjugate kernel $\tilde{C}^{(l+1)}$ in the subsequent layer and the kernel $C^{(l)}$ in the same layer, propagating the conjugate kernel $\tilde{C}^{(l)}$ backwards across layers to mediate information about the relation between inputs and outputs. We will investigate this backpropagation further in Sec. 3.5.2, drawing a link to criticality and signal scales in the network.

3.4.2 Forward-backward kernel propagation equations

As a main result, we get the pair of equations (3.55) and (3.62)

$$C^{(l+1)} \overset{(3.55)}{=} F(C^{(l)}, \tilde{C}^{(l+1)}), \tag{3.63}$$

$$\tilde{C}^{(l)} \overset{(3.62)}{=} G(C^{(l)}, \tilde{C}^{(l+1)})\, \tilde{C}^{(l+1)}, \tag{3.64}$$

with initial and final conditions, respectively, given by the fixed input kernel $C^{(0)}$ in Eq. (3.26) and the conjugate output kernel $\tilde{C}^{(L)}$ in (3.60) as

$$\tilde{C}^{(L)} = \frac{1}{2}(C^{(L)} + \kappa\mathbb{I})^{-1}(YY^{\mathsf{T}} - C^{(L)} - \kappa\mathbb{I})(C^{(L)} + \kappa\mathbb{I})^{-1}. \tag{3.65}$$

This set of equations describes the mode of the posterior distribution of network kernels C in the proportional limit $P = \nu N \to \infty$. The ratio $\nu = P/N$ can be understood as the training load on the network.

We can already obtain some insights and intuition from this set of equations, starting with Eq. (3.63) for the feature-corrected kernels $C^{(l)}$. These are propagated forward through the network as $C^{(l)} \mapsto C^{(l+1)}$. The forward-propagation takes a form similar to the NNGP but with respect to a different non-Gaussian measure that depends on the activation function ϕ, the kernels $C^{(l)}$, and the conjugate kernels $\tilde{C}^{(l+1)}$. Importantly, in contrast to the NNGP limit, i.e. P fixed at a finite value and thus $\nu = 0$, where the conjugate kernels $\tilde{C} \equiv 0$ vanish (see Eq. (3.35)-(3.37) in Sec. 3.3.3), the kernel $C^{(l)}$ receives a correction from the conjugate kernel $\tilde{C}^{(l+1)}$ in the subsequent layer. The Gaussian limits of both the NNGP and the NTK are contained as special cases in our theoretical framework; we derive the NTK in App. B.4 under the assumption that the network output depends linearly on all network parameters.

While the NNGP does not describe feature learning as shown by Yang and Hu (2020), our theoretical framework captures feature learning whenever the log-likelihood of the data $\mathcal{S}_D$ is significant relative to the rate function Γ encoding the network prior. This applies to the proportional limit with $N \to \infty$ with $P = \nu N$ and for large but finite values of N, P. We treat the latter case in the following Sec. 3.4.4, where feature learning corrections result from leading-order fluctuation corrections in N^{-1}. The posterior kernels $C^{(l)}$ balance between the likelihood of the data $\mathcal{S}_D$ and the rate function $-\Gamma$ from the network prior; the maximization yields the backward-propagating equation Eq. (3.64) that maps $\tilde{C}^{(l+1)} \to \tilde{C}^{(l)}$ for the conjugate kernels. Due to the data term S_D, the output kernel $C^{(L)}$ receives a correction towards the target kernel YY^{T} in Eq. (3.60) as we will see in experiments and derive explicitly for deep linear networks in App. B.3. Thereby, the network more closely reproduces the training labels. In contrast, both NNGP and NTK depend only on the training data X and thus cannot represent the relation between pairs of inputs and outputs $\{(x_\alpha, y_\alpha)\}_\alpha$. In the extreme case when the output kernel matches the target kernel up to the regularization noise $C^{(L)} + \kappa\mathbb{I} = YY^\mathsf{T}$, the conjugate kernel $\tilde{C}^{(L)} = 0$ of the output layer vanishes and subsequently from Eq. (3.64) all conjugate kernels $\tilde{C}^{(l)} = 0$ vanish. In this case, there are no additional corrections driven by the conjugate kernels $\tilde{C}^{(l)}$ since the kernel is already optimally aligned with the target kernel YY^T up to the regularization noise $\kappa\mathbb{I}$.

3.4.3 Perturbative, leading-order solution of the forward-backward kernel equations

For general activation functions ϕ, the expectation values with regard to the measure *Eq.* (3.56) in the forward-backward equations are non-Gaussian and thus hardly tractable. The non-Gaussianity in the measure *Eq.* (3.56) results from the term $\frac{\sigma_w^2}{N}\phi_\alpha^{(l)}\tilde{C}_{\alpha\beta}^{(l+1)}\phi_\beta^{(l)}$ in the exponent. For large network width N, this term is suppressed by N^{-1} relative to the second term $-\frac{1}{2}h_\alpha^{(l)}[C^{(l)}]_{\alpha\beta}h_\alpha^{(l)}$ from the Gaussian part of the measure. Thus, we perform an expansion in N^{-1}, which corresponds to expanding to linear order in $\tilde{C}$, and obtain the perturbative, leading-order solution of the forward equation in Eq. (3.55) as

$$C_{\alpha\beta}^{(l+1)} = \sigma_w^2 \left\langle \phi_\alpha^{(l)}\phi_\beta^{(l)} \right\rangle_{\mathcal{N}(0,C^{(l)})} + \sigma_b^2 + \frac{\sigma_w^4}{N}\sum_{\gamma,\delta} V_{\alpha\beta,\gamma\delta}^{(l)}\tilde{C}_{\gamma\delta}^{(l+1)} + \mathcal{O}\left(N^{-2}\right), \tag{3.66}$$

$$V_{\alpha\beta,\gamma\delta}^{(l)} := \langle \phi_\alpha^{(l)}\phi_\beta^{(l)}\phi_\gamma^{(l)}\phi_\delta^{(l)} \rangle_{\mathcal{N}(0,C^{(l)})} - \langle \phi_\alpha^{(l)}\phi_\beta^{(l)} \rangle_{\mathcal{N}(0,C^{(l)})} \langle \phi_\gamma^{(l)}\phi_\delta^{(l)} \rangle_{\mathcal{N}(0,C^{(l)})}. \tag{3.67}$$

As a result of the expansion, all appearing expectation values become Gaussian $\langle \ldots \rangle_{\mathcal{N}(0,C^{(l)})}$. For the backward equation, the same expansion yields the perturbative, leading-order solution by replacing all non-Gaussian expectation values $\langle \ldots \rangle_{\mathcal{P}^{(l)}}$ by Gaussian expectation values $\langle \ldots \rangle_{\mathcal{N}(0,C^{(l)})}$ in Eq. (3.62). In this form, we can explicitly identify the correction of the kernels $C_{\alpha\beta}^{(l+1)}$ from the backpropagated error signal $\tilde{C}_{\gamma\delta}^{(l+1)}$, which results from the interaction with all other data samples via the four-point interaction term $\sum_{\gamma,\delta} V_{\alpha\beta,\gamma\delta}^{(l)}\tilde{C}_{\gamma\delta}^{(l+1)}$. We visualize the forward-backward dependence of the posterior, feature-corrected kernels in Fig. 3.2.

For some non-linear activation functions such as $\phi = \mathrm{erf}(x)$, we can derive closed-form solutions for the two-point integrals in Eq. (3.62), Eq. (3.66) and Eq. (3.67); however, the four-point integral in Eq. (3.67) is determined numerically (for details see App. B.6).

3.4.4 Fluctuation corrections lead to feature learning

We here show that the perturbative, leading-order solution Eq. (3.66) of the kernel equations can alternatively be derived in a field-theoretic framework, yielding a for-

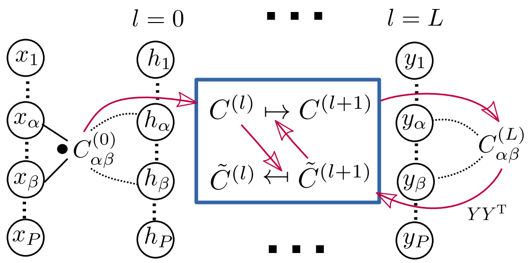


Figure 3.2: Visualization of the perturbative, leading-order solution to the forward-backward equations across network layers. The feature-corrected kernel $C^{(l)}$ is being forward propagated and thus depends on the kernel $C^{(l-1)}$ from the previous layer; in addition it receives a correction from the conjugate kernel $\tilde{C}^{(l+1)}$. The conjugate kernel $\tilde{C}^{(l)}$ is being backwards propagated and thus depends on the kernel $\tilde{C}^{(l+1)}$ from the subsequent layer; the expectation values appearing in Eq. (3.62) are with respect to the feature-corrected kernel $C^{(l)}$.

mulation as fluctuation corrections to the NNGP for large but finite network width N and number of training samples P for small training load $\nu = P/N$.

When deriving the network prior Eq. (3.27), we introduce auxiliary variables $C^{(l)}_{\alpha\beta} = \sigma_w^2/N\,\phi_\alpha^{(l-1)} \cdot \phi_\beta^{(l-1)\mathsf{T}} + \sigma_b^2$. In the limit of infinite network width $N \to \infty$, these quantities concentrate: the appearing sum over neuron indices in the scalar product becomes an expectation value, yielding the NNGP kernel $C^{(l)}_* = \sigma_w^2 \,\langle \phi^{(l-1)}\phi^{(l-1)} \rangle_{\mathcal{N}(0,C_*^{(l-1)})} + \sigma_b^2$. For large but finite network width $N < \infty$, the realizations of the auxiliary variables for individual network realizations with network parameters $\theta = \left\{W^{(l)}, b^{(l)}\right\}_l$ remain close to the NNGP kernel but exhibit fluctuations around this value

$$C^{(l)} = C^{(l)}_* + \delta C^{(l)}_{\alpha\beta}. \tag{3.68}$$

To account for these fluctuations, we utilize the expression in Eq. (3.27) for the network prior, writing it as an integral over the pair of fields $(C, \tilde{C})$

$$\begin{aligned} p(Y|X) &= \int DC \int D\tilde{C}\,\mathcal{N}(Y|0, C^{(L)} + \kappa\mathbb{I}) \exp\left(\mathcal{S}(C,\tilde{C})\right), \\ \mathcal{S}(C,\tilde{C}) &= -\mathrm{tr}\,\tilde{C}^{\mathsf{T}}C + \mathcal{W}(\tilde{C}|C), \end{aligned} \tag{3.69}$$

with cumulant-generating function $\mathcal{W}$ given by Eq. (3.31). Rewriting the Gaussian term $\mathcal{N}(Y|0, C^{(L)} + \kappa\mathbb{I})$ in Eq. (3.69) as $\exp(\mathcal{S}_\mathrm{D})$, we obtain the joint distribution for the pair of variables $(C, \tilde{C})$ up to its normalization constant as

$$(C, \tilde{C}) \sim \exp\left(\mathcal{S}(C,\tilde{C}) + \mathcal{S}_\mathrm{D}(C^{(L)}|Y)\right),$$

with the data term given by

$$\begin{aligned}\mathcal{S}_{\mathrm{D}}(C^{(L)}|Y) &\coloneqq \ln\mathcal{N}(Y|0,C^{(L)}+\kappa\mathbb{I})\\ &= -\frac{1}{2}Y^{\mathsf{T}}(C^{(L)}+\kappa\mathbb{I})^{-1}Y-\frac{1}{2}\ln\det(C^{(L)}+\kappa\mathbb{I}).\end{aligned}$$

We expand $S(C,\tilde{C})$ around its saddle point $(C_*,\tilde{C}_*)$, which is given by the NNGP as shown in Sec. 3.3.3. We use the next-to-leading-order in N^{-1} as computed in Sec. 3.3.4

$$\mathcal{S}^{(2)(l,m)}_{(\alpha\beta)(\gamma\delta)}\Big|_{C_*,\tilde{C}\equiv 0} =$$
$$\begin{pmatrix} 0 & -\delta_{(l\alpha\beta),(m\gamma\delta)}+\delta_{m-1,l}\,\sigma_w^2\,\frac{\partial\langle\phi_\gamma^{(m-1)}\phi_\delta^{(m-1)}\rangle_{\mathcal{N}(0,C_*^{(l)})}}{\partial C_{\alpha\beta}^{(l)}}\\ -\delta_{(l\alpha\beta),(m\gamma\delta)}+\delta_{l-1,m}\,\sigma_w^2\,\frac{\partial\langle\phi_\alpha^{(l-1)}\phi_\beta^{(l-1)}\rangle_{\mathcal{N}(0,C_*^{(l)})}}{\partial C_{\gamma\delta}^{(m)}} & \delta_{l,m}\,\sigma_w^4\,\langle\phi_\alpha^{(l-1)}\phi_\beta^{(l-1)},\phi_\gamma^{(l-1)}\phi_\delta^{(l-1)}\rangle^c_{\mathcal{N}(0,C_*^{(l)})}\end{pmatrix}.$$

The Laplace approximation of the network prior in Eq. (3.69) then takes the form

$$p(Y|X)\simeq\int D\delta C\int D\delta\tilde{C}\,\exp\Big(\frac{1}{2}(\delta C,\delta\tilde{C})^{\mathsf{T}}\mathcal{S}^{(2)}\,(\delta C,\delta\tilde{C})+\mathcal{S}_{\mathrm{D}}(C_*^{(L)}+\delta C^{(L)}|Y)\Big) \quad (3.70)$$

since the constant Taylor term is $\mathcal{S}(C_*,\tilde{C}_*=0)\equiv 0$ and the linear term vanishes at the saddle point $(C_*,\tilde{C}_*)$ by definition.

We now examine how the data term $\mathcal{S}_{\mathrm{D}}$ affects the saddle point of δC and $\delta\tilde{C}$. The data term only has an effect on $\delta C^{(L)}$, functioning as a source term $\mathrm{tr}J^{\mathsf{T}}\,\delta C^{(L)}$ with

$$J_{\alpha\beta}\coloneqq\frac{\partial\mathcal{S}_{\mathrm{D}}}{\partial C_{\alpha\beta}^{(L)}}. \quad (3.71)$$

Thus, the saddle point equation for the shift $(\delta C,\delta\tilde{C})$ of the saddle points is given by

$$0=\left[\mathcal{S}^{(2)}\begin{pmatrix}\delta C\\ \delta\tilde{C}\end{pmatrix}\right]+\begin{pmatrix}J\,\delta_{l,L}\\ 0\end{pmatrix}. \quad (3.72)$$

The first line of Eq. (3.72) is

$$0=-\delta\tilde{C}_{\alpha\beta}^{(l)}+\sigma_w^2\sum_{\gamma\delta}\frac{\partial\langle\phi_\gamma^{(l)}\phi_\delta^{(l)}\rangle_{\mathcal{N}(0,C^{(l)})}}{\partial C_{\alpha\beta}^{(l)}}\,\delta\tilde{C}_{\gamma\delta}^{(l+1)}+\delta_{l,L}\,J_{\alpha\beta}. \quad (3.73)$$

Using Price's theorem (Price, 1958; Papoulis and Pillai, 2002) to compute the derivate with respect to $C_{\alpha\beta}^{(l)}$ then yields

$$\sum_{\gamma\delta}\frac{\partial\langle\phi_\gamma^{(l)}\phi_\delta^{(l)}\rangle_{\mathcal{N}(0,C^{(l)})}}{\partial C_{\alpha\beta}^{(l)}}\,\delta\tilde{C}_{\gamma\delta}^{(l+1)}=\frac{1}{2}\sum_{\gamma\delta}\langle\frac{\partial}{\partial h_\alpha^{(l)}}\frac{\partial}{\partial h_\beta^{(l)}}\,\phi_\gamma^{(l)}\phi_\delta^{(l)}\rangle_{\mathcal{N}(0,C^{(l)})}\,\delta\tilde{C}_{\gamma\delta}^{(l+1)}$$

$$
\begin{aligned}
&= \left\langle (\phi_\alpha^{(l)})'(\phi_\beta^{(l)})' \right\rangle_{\mathcal{N}(0,C^{(l)})} \delta\tilde{C}_{\alpha\beta}^{(l+1)} \\
&\quad + \delta_{\alpha\beta} \sum_\gamma \left\langle (\phi_\alpha^{(l)})''(\phi_\gamma^{(l)}) \right\rangle_{\mathcal{N}(0,C^{(l)})} \delta\tilde{C}_{\alpha\gamma}^{(l+1)},
\end{aligned}
$$

where we make use of the fact that $\tilde{C}$ and C are both symmetric in (α,β). By substituting this expression in Eq. (3.73), we arrive at the perturbative leading-order solution for $\tilde{C}$ in Eq. (3.62).

The second line of Eq. (3.72) gives

$$
\begin{aligned}
0 = -\delta C_{\alpha\beta}^{(l)} &+ \sigma_w^2 \sum_{\gamma\delta} \frac{\partial\langle \phi_\alpha^{(l-1)}\phi_\beta^{(l-1)} \rangle_{\mathcal{N}(0,C^{(l-1)})}}{\partial C_{\gamma\delta}^{(l-1)}} \delta C_{\gamma\delta}^{(l-1)} \\
&+ \sigma_w^4 \sum_{\gamma\delta} \langle \phi_\alpha^{(l-1)}\phi_\beta^{(l-1)}, \phi_\gamma^{(l-1)}\phi_\delta^{(l-1)} \rangle_{c,\mathcal{N}(0,C^{(l-1)})} \delta\tilde{C}_{\gamma\delta}^{(l)},
\end{aligned}
$$

which can be rewritten as

$$
\delta C_{\alpha\beta}^{(l)} = \sigma_w^2 \sum_{\gamma\delta} \frac{\partial\langle \phi_\alpha^{(l-1)}\phi_\beta^{(l-1)} \rangle_{\mathcal{N}(0,C^{(l-1)})}}{\partial C_{\gamma\delta}^{(l-1)}} \delta C_{\gamma\delta}^{(l-1)} + \sigma_w^4 \sum_{\gamma\delta} V_{\alpha\beta,\gamma\delta}^{(l-1)} \delta\tilde{C}_{\gamma\delta}^{(l)}, \tag{3.74}
$$

where we identify $V_{\alpha\beta,\gamma\delta}^{(l-1)} = \langle \phi_\alpha^{(l-1)}\phi_\beta^{(l-1)}, \phi_\gamma^{(l-1)}\phi_\delta^{(l-1)} \rangle_{\mathcal{N}(0,C^{(l-1)})}^c$ from Eq. (3.66). The first term in Eq. (3.74) arises from a linear correction of the NNGP result due to the shift in δC

$$
\sigma_w^2 \left\langle \phi_\alpha^{(l-1)}\phi_\beta^{(l-1)} \right\rangle_{\mathcal{N}(0,C^{(l-1)}+\delta C^{(l-1)})} + \sigma_b^2 = C_{*,\alpha\beta}^{(l)} + \sum_{\gamma\delta} \frac{\partial C_{*,\alpha\beta}^{(l)}}{\partial C_{\gamma\delta}^{(l-1)}} \delta C_{\gamma\delta}^{(l-1)} + \mathcal{O}(\delta C)^2.
$$

The second term in Eq. (3.74) matches the corrections found in Eq. (3.66): We observe that the expression $\sigma_w^4 V_{\alpha\beta,\gamma\delta}^{(l-1)} = \langle \sigma_w^2 \phi_\alpha^{(l-1)}\phi_\beta^{(l-1)}, \sigma_w^2 \phi_\gamma^{(l-1)}\phi_\delta^{(l-1)} \rangle_{\mathcal{N}(0,C^{(l-1)})}^c$ corresponds to the covariance of the auxiliary variables $C_{\alpha\beta}^{(l)} = \sigma_w^2/N \phi_\alpha^{(l-1)} \cdot \phi_\beta^{(l-1)\mathsf{T}} + \sigma_b^2$; thus, the feature learning corrections in the self-consistency equations are in fact fluctuation corrections. In consequence, we can relate feature learning corrections to the notion of criticality as presented in Sec. 3.3.4: Stronger feature learning results from larger fluctuations and fluctuations become especially large close to critical points (see Fig. 3.3).

3.5 Experiments

We test the presented theoretical predictions against networks trained with Langevin gradient descent to ensure that the networks relax to the Bayesian posterior. Further, we reexamine the self-consistency equations Eq. (3.62) for the network kernels, finding a link to the response function of the network, criticality, and the scales within the

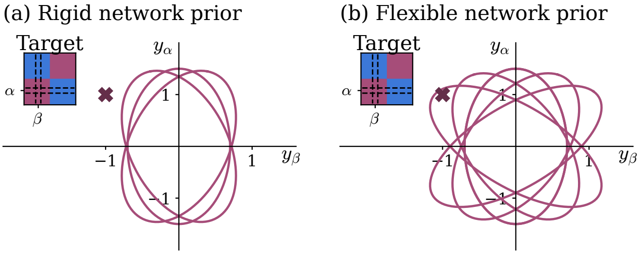


Figure 3.3: Stronger feature learning results from larger kernel fluctuations. The network prior is given by a superposition of Gaussians as $f \sim \int \mathcal{N}(0,C)\,p(C)\,dC$ (pink ellipses). The distribution of kernels is more (a) concentrated or (b) wider depending on the network hyperparameters, which corresponds to smaller or larger kernel fluctuations. The target kernel that the network aims to learn is given by YY^T (inset); the target value for the indicated example samples α, β (dashed lines in inset) from different classes lies at $(+1,-1)$ (pink cross). Gaussian components in the Bayesian posterior are reweighed according to their evidence given the data. Stronger adaptation to data is made possible by larger kernel fluctuations in (b), which results in richer feature learning.

network. Thereby, we shed light on the driving mechanisms for kernel adaptation in deep neural networks.

3.5.1 Comparative analysis with trained networks

To obtain the theoretical predictions for the posterior kernels, the self-consistency equations for both kernels $C^{(l)}$ and conjugate kernels $\tilde{C}^{(l)}$ in Eq. (3.66) and Eq. (3.62) are solved iteratively; details can be found in App. B.6. We compare the theoretical predictions obtained for the feature-corrected output kernel $C^{(L)}$ to sampling the empirical output kernel $C_{\mathrm{emp}}^{(L)}$ from the posterior distribution using Langevin stochastic gradient descent (for details see App. B.5). The comparison utilizes kernel alignment (Canatar and Pehlevan, 2022), defined for two kernels A, $B \in \mathbb{R}^{P\times P}$ as

$$\frac{\mathrm{Tr}(A\,B)}{\sqrt{\mathrm{Tr}(A\,A)\,\mathrm{Tr}(B\,B)}}. \tag{3.75}$$

For intuition, this measure can be understood as the cosine similarity between the flattened kernels A and B; thus this measure is invariant to scaling either kernel by a scalar $A \mapsto aA$ or $B \mapsto bB$. One typically centers the kernels so that constant components in the eigendecomposition of the kernels are removed (Cortes, Mohri, and Rostamizadeh, 2012); this is called the centered kernel alignment (CKA). The transformation of both kernels is done as $A \mapsto HAH$ and $B \mapsto HBH$ with a centering matrix $H = \mathbb{I} - \frac{1}{P}11^\mathsf{T}$, where 1 is the matrix with all elements equal to one.

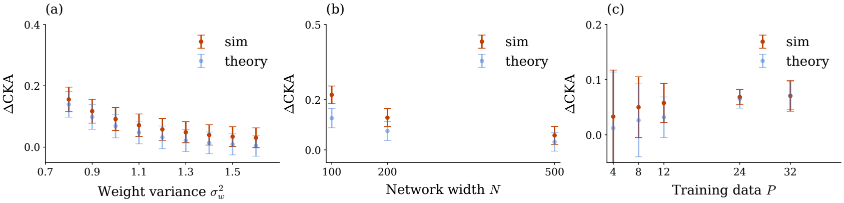


Figure 3.4: Theoretical predictions match kernels measured in trained networks for the XOR task. We plot the difference $\Delta\text{CKA} = \text{CKA}(C^{(L)}, YY^\mathsf{T}) - \text{CKA}(C_*^{(L)}, YY^\mathsf{T})$ to the NNGP, measuring the increase in kernel adaptation due to feature corrections. Kernel alignment ΔCKA of the feature-corrected kernels (blue: theory; red: empirical) increases (a) for smaller weight variance, (b) for narrower networks, and (c) for more training data. Other parameters: XOR task with $\sigma^2_{\text{XOR}} = 0.4, D = 100$, $L = 3$, $\sigma_b^2 = 0.05$, (a) $N = 500$, $P = 12$, (b) $P = 12$, $\sigma_w^2 = 1.2$, (c) $N = 500$, $\sigma_w^2 = 1.2$. Error bars indicate mean and one standard deviation over 10 training data sets.

We compute the CKA for both the theoretical predictions of the kernel $C^{(L)}$ and the Langevin-sampled empirical kernels $C^{(L)}_{\text{emp}}$, comparing each with the target kernel $YY^\top$. Our theoretical framework does not presuppose any assumptions on the data. We study two different tasks: XOR and binary classification of MNIST digits. The numerical results consistently align with our theoretical predictions for both tasks.

XOR

Refinetti et al. (2021) show that random feature models, which are known to correspond to the NNGP (Mei and Montanari, 2022), are unable to solve the non-linearly separable task XOR optimally. We study the XOR task in a setting where neural networks exhibit feature learning compared to random feature models (Refinetti et al., 2021). The feature-corrected kernels that we obtain from our theory have a larger CKA than the NNGP (see Fig. 3.4), indicating that finite-width effects lead to kernel corrections in the direction of the target kernel. Note that the CKA is by construction invariant to a global rescaling of the kernel and instead captures the kernel structure. Thus, the difference between NNGP and empirical kernels is further numerical evidence that the kernels acquire structure beyond a global rescaling, in contrast to deep linear networks (Li and Sompolinsky, 2021) and opposed to approximate results employing Gaussian equivalence results (Pacelli et al., 2023; Baglioni et al., 2024). We observe a stronger kernel alignment for smaller weight variance σ_w^2 (see Fig. 3.4(a)). As expected, the feature-corrected kernels approach the NNGP limit for $\nu = P/N \to 0$ when keeping P fixed in Fig. 3.4(b). Deviations in Fig. 3.4(b) for small N and in Fig. 3.4(c) for increasing P at fixed N result from the perturbative treatment of $\tilde{C}$ in the numerical solution of the self-consistency equations, which is strictly valid only for $\nu = P/N \ll 1$.

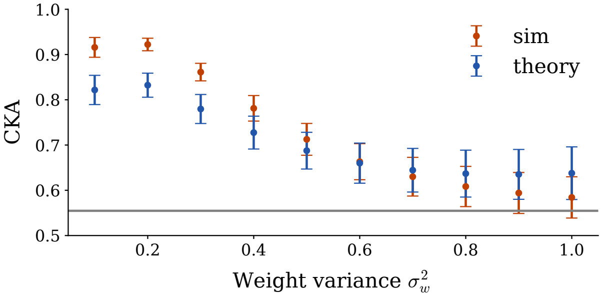


Figure 3.5: Theoretical predictions match kernels measured in trained networks for MNIST. In comparison to the NNGP (gray), theory (blue) and simulation (red) show significant kernel adaptation towards the target kernel measured by $\mathrm{CKA}(C^{(l)}, YY^{\mathsf{T}})$. Kernel adaptation becomes maximal for small weight variance σ_w^2 that is coherent for theory (blue) and simulation (red). Other parameters: MNIST task with $L = 2$, $N = 2000$. Error bars indicate mean and one standard deviation over 10 training data sets.

MNIST

For MNIST (LeCun, Cortes, and Burges, 1998), we consider binary classification between digits 0 and 3. The theoretical predictions for the kernels involving feature-corrections match the kernels measured in trained networks (see Fig. 3.5). Further, they exhibit stronger kernel alignment with the target kernel YY^{T} than the NNGP does, as expected.

3.5.2 Link between feature learning corrections and criticality

In the feature-corrected kernel equations in Eq. (3.66) appears a term that measures the fluctuations of the auxiliary variables due to the finite-size of the network. Since fluctuations typically become large close to critical points, we here investigate the connection between the feature-corrected kernels and criticality in neural networks (Schoenholz et al., 2017). The corrections of the network kernel $C^{(l-1)}$ are mediated by the conjugate kernel $\tilde{C}^{(l)}$, so we examine the self-consistency equations for the conjugate kernels. For the off-diagonal terms $\alpha \neq \beta$, we can identify two contributions

$$\tilde{C}_{\alpha\beta}^{(l)} = \tilde{C}_{\alpha\beta}^{(L)} \chi_{\alpha\beta}^{(l),\leftarrow}, \tag{3.76}$$

$$\chi_{\alpha\beta}^{(l),\leftarrow} := \prod_{s=l}^{L-1} \sigma_w^2 \left\langle \left(\phi_\alpha^{(s)}\right)' \left(\phi_\beta^{(s)}\right)' \right\rangle_{h^{(s)} \sim \mathcal{N}(0, C^{(s)})}, \tag{3.77}$$

with $\tilde{C}^{(L)}$ as in Eq. (3.60). While the term $\tilde{C}^{(L)}$ acts as an error signal and relates to the mismatch between the output kernel $C^{(L)}$ and the target kernel given by YY^{T}

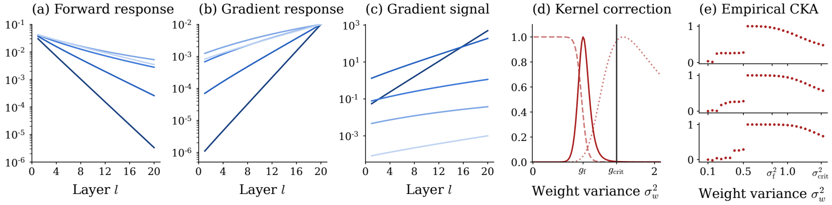


Figure 3.6: Finite-size effects close to criticality that drive feature corrections. (a)-(b) Forward response $\chi^{l,\rightarrow}$ and gradient response $\chi^{l,\leftarrow}$ describe relative signal propagation across network layers. The signal propagates to deeper network layers close to criticality (σ_w^2 increases from dark to light). (c) Conjugate kernel $\tilde{C}^{(l)}$ backpropagated across network layers for varying weight variance σ_w^2 (increasing from dark to light). Both the response function and the kernel mismatch $\tilde{C}^{(L)}$ in the output depend on the weight variance σ_w^2. The curves of $\tilde{C}^{(l)}$ cross at different depths. Larger $\tilde{C}^{(l)}$ lead to larger feature learning corrections in Eq. (3.66). (d) The kernel correction term $\tilde{C}^{(0)}$ in the readin layer (solid line, slice for $l = 1$ in (c)) is composed of the gradient response $\chi^{1,\leftarrow}$ (dotted line, slice for $l = 1$ in (b)) and the error signal $\tilde{C}^{(L)}$ (dashed line). Largest feature learning corrections arise for a weight variance σ_f^2 shifted away from the critical point σ_{crit}^2 (vertical line). (e) CKA for trained networks between $C_{\mathrm{emp}}^{(l)}$ and YY^{T} ($l = 10, 15, 20$ from top to bottom). Other parameters: XOR task with $\sigma_{\mathrm{XOR}}^2 = 0.4$, $\sigma_w^2 \in \left\{0.6, \sigma_f^2 \approx 0.825, 1.1, \sigma_{\mathrm{crit}}^2 \approx 1.38, 2.2\right\}$, $\sigma_b^2 = 0.05$, $L = 20$, $N = 500$,$\kappa = 10^{-3}$, $P = 12$.

as discussed before, we identify $\chi^{(l),\leftarrow}$ as the gradient response function of the network. The gradient response function is the equivalent to the forward response function discussed in Sec. 3.3.5 and measures how variations in the network output propagate backwards through the network. The response functions naturally appear in the next-to-leading-order corrections to the NNGP kernel. Close to criticality, both reponse function indicate long-range correlations across network layers (see Fig. 3.6(a)-(b)) and the signal can propagate to large depths, leading to improved network trainability in deep feed-forward networks (Schoenholz et al., 2017).

For the feature corrections in Eq. (3.66), we now observe two competing effects in hyperparameter space (σ_w^2, σ_b^2): On the one hand, since the gradient response function backpropagates the conjugate kernel $\tilde{C}^{(L)}$ of the output, feature corrections propagate furthest in the network close to criticality (see Fig. 3.6(d)). On the other hand, the error signal given by $\tilde{C}^{(L)}$ itself also depends non-linearly on both weight variance σ_w^2 and bias variance σ_b^2. For fixed bias variance σ_b^2, it is largest for small weight variances σ_w^2 (see Fig. 3.6(d)), which is primarily a scaling effect between the output kernel $C^{(L)} = \mathcal{O}_{\sigma_w^2}(\sigma_w^2)$ and the target kernel $YY^{\mathsf{T}} = \mathcal{O}_{\sigma_w^2}(1)$. Since criticality as in (Schoenholz et al., 2017) applies to networks at initialization, we consider the tradeoff between these two competing effects at the NNGP and perform one iteration step for solving the self-consistency equations. We observe that the conjugate kernels $\tilde{C}^{(l)}$ then show a particular depth dependence for different weight variances σ_w^2 (see Fig. 3.6(c)) due to the dependence on both the kernel mismatch $\tilde{C}^{(L)}$ and the gradient response $\chi^{(l),\leftarrow}$. We find that the conjugate kernel $\tilde{C}^{(1)}$ in the first later is maximal at a weight variance $\sigma_w^2 \simeq \sigma_f^2$ below the critical value σ_{crit}^2 (see Fig. 3.6(d)). Indeed, we see a similar effect in trained networks where kernel adaptation is large above values $\sigma_w^2 \simeq \sigma_f^2$ for the weight variance but drops sharply below. Thus, we identify criticality and output scale as the main components in our theoretical framework that drive feature learning corrections.

3.5.3 Downscaling network outputs boosts feature learning

Having observed how feature learning corrections arise from the interplay between response function and kernel mismatch in the network output, we now explore a strategy to boost feature learning in the network. The response function reflects behavior across all network layers, while the kernel mismatch is only measured in the output layer. While the output kernel $C^{(L)}$ depends directly on the weight variance of the output layer $\sigma_{w,L}^2$, the target kernel YY^T does not. In consequence, we can decrease the weight variance of the output layer $\sigma_{w,L}^2$ to reduce the scale of the output kernel $C^{(L)}$ relative to the target kernel YY^T, thereby directly increasing the kernel mismatch and thus feature learning corrections in Eq. (3.62).

We study the setting where the output weight variance is scaled by a factor γ_0, which is not extensive in the number of hidden units N, modifying σ_w^2 in the output layer L as $\sigma_{w,L}^2 \mapsto \sigma_{w,L}^2/\gamma_0$. We investigate the impact of this factor on the feature-corrected kernels given by the self-consistency equations Eq. (3.62): The feature-corrections are mediated by the conjugate kernels $\tilde{C}^{(l)} = \tilde{C}^{(L)}\chi^{(l),\leftarrow}$. Since the output kernel scales as $C_{\alpha\beta}^{(L)} \propto \sigma_{w,L}^2/\gamma_0$, we get $\tilde{C}_{\alpha\beta}^{(L)} \overset{(3.60)\text{ for }\kappa=0}{\propto} \gamma_0^2 + \mathcal{O}(\gamma_0)$. Using Eq. (3.77), we find that the response function scales as $\chi^{(L),\leftarrow} \propto \sigma_{w,L}^2/\gamma_0$. Thus, the feature corrections mediated by the conjugate kernel in the input layer $\tilde{C}_{\alpha\beta}^{(0)}$ increase linearly with downscaling the output variance

$$\tilde{C}_{\alpha\beta}^{(0)} = \chi^{(L),\leftarrow}\tilde{C}_{\alpha\beta}^{(L)} \propto \gamma_0 + \mathcal{O}_{\gamma_0}(1). \tag{3.78}$$

Thereby, the network shows stronger kernel adaptation to the given training data. Due to the linear dependence of the feature corrections in Eq. (3.62) on the conjugate kernels $\tilde{C}^{(l)}$ and their linear scaling with the feature scale γ_0, feature corrections increase consistenly across all network layers in Fig. 3.7(a) when progressively increasing the feature scale γ_0. The underlying intuition for the feature scale is that the reduced scale of the output kernel $C^{(L)}$ prompts the network kernels $C^{(l)}$ to expand towards the direction of the target kernel YY^T. Introducing the feature scale γ_0 does not change the interplay between criticality and weight variance σ_w^2 for $l < L$ discussed in the previous section, but increasing the feature scale γ_0 overall boosts feature learning as shown in Fig. 3.7(b).

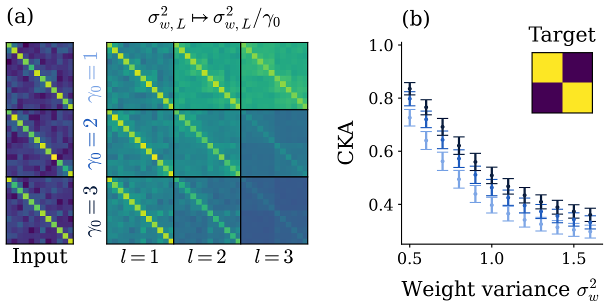


Figure 3.7: Increase of feature scale leads to stronger kernel adaption. (a) Feature-corrected kernels $C^{(l)}$ across network layers $l = 1, 2, 3$ for different feature scales γ_0. Downscaling the output layer already increases feature corrections across all network layers. Other parameters: XOR task with $\sigma^2_{\mathrm{XOR}} = 0.4$, $P = 12$, $N = 200$, $L = 3$, $\sigma_w^2 = 0.5$. (b) Dependence of CKA between output kernel $C^{(L)}$ and target kernel YY^T on feature scaling ($\gamma_0 = 1, 2, 3$ from dark to light) and weight variance σ_w^2. Increasing the feature scale linearly increases kernel adaptation across weight variances σ_w^2. Error bars indicate mean and one standard deviation over 10 training data sets. Other parameters: XOR task with $\sigma^2_{\mathrm{XOR}} = 0.4$, $P = 12$, $L = 3$.

3.6 Conclusion

In this chapter, we derive the posterior kernels of fully-connected networks in the Bayesian setting. We obtain a set of forward-backward propagation equations for the kernels that mediate the input-output relation between training data and labels. On both XOR and MNIST, we observe non-linear adaptation of the kernels towards the target kernel given by the outer product of the labels.

In finite-size networks, the feature corrections result from fluctuation corrections of the kernels as a direct consequence of their finite width. Since fluctuations typically become large close to critical points in hyperparameter space, we investigate their relation to criticality: While being close to criticality allows the error signal to backpropagate to deeper layers, the error signal itself exhibits a scaling effect and is largest for small weight variances. We find that the trade-off between these two effects leads to stronger feature corrections at values slightly lower than the critical ones, shedding light onto the driving mechanisms of feature learning.

3.6.1 Limitations

We obtain the perturbative leading-order solution of the posterior kernels in Sec. 3.4 by expanding the forward-backward equations to linear order in the conjugate kernels. For this approximation to be valid, we require corrections to the NNGP limit to be small, i.e. $\nu = P/N \ll 1$. This limitation applies to non-linear networks only; in linear networks the forward-backward equations directly involve Gaussian averages (see App. B.2). We can bypass this assumption to a certain degree by computing feature-corrected kernels iteratively from wider to narrower networks (see App. B.6), but are still limited if the training load $\nu = P/N$ becomes too large.

3.6.2 Relation to other works

Describing the Bayesian posterior of neural networks is a highly active field of research. One line of research focuses on deep linear networks, where the non-linearity is replaced by the identity mapping. Li and Sompolinsky (2021) find that in the proportional limit, where $N \to \infty$, $P \to \infty$ with fixed training load $P/N = \nu$, kernels adapt to the data by changing only their overall scale compared to the NNGP result, which they call kernel renormalization. Hanin and Zlokapa (2023) derive a rigorous non-asymptotic solution in terms of Meijer-G functions and find that infinitely deep linear networks with data-agnostic priors yield the same result as shallow networks with evidence-maximizing data-dependent priors. Zavatone-Veth, Tong, and Pehlevan (2022) determine perturbation corrections and find that the generalization error in deep linear networks receives corrections only at quadratic order or higher. Yang et al. (2023) study deep kernel machines, for which they find a similar tradeoff between network prior and data terms as in this work; however, they consider a different limit where the number of training samples P is fixed but the network is trained on N copies of the training data. While we consider the special case of linear networks in App. B.2, recovering the result by Yang et al. (2023), our framework encompasses both linear and non-linear networks.

Kernel renormalization in non-linear networks has been studied in multiple works: Pacelli et al. (2023) use the Breuer-Major theorem to derive an effective action for non-linear single-hidden-layer networks, which exhibit kernel renormalization. In (Aiudi et al., 2023), they extend their results to convolutional networks and in (Baglioni et al., 2024), they empirially investigate the validity of their theory across different hyperparameter sets. Ingrosso et al. (2024) apply their approach to study the efficiency of fine-tuning in transfer learning. Tiberi et al. (2024) apply a similar approach as Li and Sompolinsky (2021) to a transformer-like architecture, uncovering different attention paths that recombine kernels to enhance generalization performance. Meegen and Sompolinsky (2024) investigate coding schemes in neural networks, uncovering a direct dependence on the used activation function. In contrast to those works, we uncover non-linear feature corrections to the NNGP kernel that agree well with kernels empirically measured in trained networks.

Closest in spirit to our work is (Seroussi, Naveh, and Ringel, 2023): They derive a similar forward-backward relation between kernels, but use a variational Gaussian approximation of the posterior hidden representations to obtain the posterior kernels. While their approach involves not only Gaussian fluctuations of the auxiliary variables but also higher-order ones, we find empirically that the Gaussian fluctuations seem to be dominating. Further, our results allow us to investigate the driving mechanisms of feature learning: we find a link to criticality, uncovering a trade-off between kernel fluctuations and network scales.

Another line of research considers perturbative approaches. Halverson, Maiti, and Stoner (2021) perform a perturbation expansion where the non-linear terms of the activation constitute the expansion parameter. Other works build on the Edgeworth expansion using non-Gaussian cumulants as the expansion parameters: (Dyer and Gur-Ari, 2020; Huang and Yau, 2020; Aitken and Gur-Ari, 2020; Roberts, Yaida, and Hanin, 2022; Bordelon and Pehlevan, 2023) consider gradient-based training while (Yaida, 2020; Antognini, 2019; Naveh et al., 2021; Cohen, Malka, and Ringel, 2021; Roberts, Yaida, and Hanin, 2022) study Bayesian inference. In contrast to these works, the forward-backward propagation equations do not require a perturbation expansion but result directly from a saddle point approximation in both auxiliary and conjugate variables.

Chapter 4

Field theory for optimal signal propagation in residual networks

This chapter, App. C, and parts of the discussion are based on the following preprint:

Kirsten Fischer, David Dahmen, and Moritz Helias. "Optimal signal propagation in residual networks through residual scaling." arXiv preprint. arXiv 2305:07715.

Author contributions
Under the supervision of David Dahmen and Moritz Helias, the author performed all theoretical and numerical parts of the publication. The author wrote the original draft of the publication and all authors contributed equally to revising and finalizing the publication.

4.1 Introduction

In the previous chapter, we determined feature corrections to network kernels in trained networks, pointing out that the structure of these feature corrections reflects the network architecture being fully-connected. Naturally, one may ask how these feature corrections change for other commonly used network architectures such as convolutional networks, residual networks, or transformers. The theoretical framework in the previous chapter can be applied to other architectures given knowledge of the network prior in a field-theoretical framework.

In this chapter, we derive the network prior of residual networks and study finite-size characteristics of networks at initialization that are linked to the networks' trainability. More specifically, we are interested in the effect of scaling the residual branch of the network by a hyperparameter that empirically leads to higher performance values, as shown in (Szegedy et al., 2017), and can be connected to signal propagation in the networks.

The main contributions of this chapter are:

- we derive the Bayesian network prior for residual networks in a field-theoretic formulation, yielding a framework for determining finite-size properties of residual networks;
- at infinite network width, we recover the Neural Network Gaussian Process kernel as the saddle point of the action and for finite width, we compute next-to-leading-order corrections to the saddle point, yielding kernel fluctuations and the response function that measures the networks' sensitivity to varying inputs;
- we find that the response function as a function of the residual scaling has a unique maximum that links to optimal signal propagation and imroved trainability of the network;
- we find a $1/\sqrt{\text{depth}}$ dependence of the optimal residual scaling and merely a weak dependence on other hyperparameters, explaining the universal succes of this scaling across different residual architectures.

In the following, we will repeatedly draw comparisons between feed-forward networks and residual networks. Therefore we introduce the abbreviations FFNets for feed-forward networks and ResNets for residual networks.

4.2 Setup

In this chapter, we consider the residual network architecture defined as

$$\begin{aligned} h^{(0)} &= W^{\text{in}}x + b^{\text{in}}, \\ h^{(l)} &= h^{(l-1)} + \rho\left[W^{(l)}\phi(h^{(l-1)}) + b^{(l)}\right] \quad l = 1, \dots, L, \\ y &= W^{\text{out}}\phi(h^{(L)}) + b^{\text{out}}. \end{aligned} \tag{4.1}$$

The network implements a mapping from inputs $x_\alpha \in \mathbb{R}^{D_{\text{in}}}$ to outputs $y_\alpha \in \mathbb{R}^{D_{\text{out}}}$ for different samples α given by $x_\alpha \mapsto f(x_\alpha;\theta) = y_\alpha$. Here, we denote the joint set of trainable network parameters by $\theta = \left\{W^{\text{in}}, b^{\text{in}}, W^{(l)}, b^{(l)}, W^{\text{out}}, b^{\text{out}}\right\}$. The network includes a linear readin and a fully-connected readout layer, similar to state-of-the-art models like ResNet-50 (He et al., 2016). These two architecture components allow us to choose different dimensionalities: inputs $x_\alpha \in \mathbb{R}^{D_{\text{in}}}$ are of dimension D_{in}, signals $h^{(l)}(x_\alpha) \in \mathbb{R}^N$ in layer l of dimension N, and outputs $y_\alpha \in \mathbb{R}^{D_{\text{out}}}$ of dimension D_{out}. The residual branch is denoted as $\mathcal{F}(h^{(l-1)}) = \rho\left[W^{(l)}\phi(h^{(l-1)}) + b^{(l)}\right]$. Each network layer l consists of the skip connection $h^{(l-1)}$ and the residual branch $\mathcal{F}(h^{(l-1)})$ in Eq. (4.1), illustrated in Fig. 4.1(a). The network depth L corresponds to the total number of layers with skip connections. In the theoretical derivations, we assume the non-linear activation function ϕ to be saturating and twice differentiable almost everywhere. One common choice fulfilling both conditions is the error function $\phi = \text{erf}$, which we will use throughout this chapter. We scale the residual branch by a residual scaling parameter ρ, which we take to be a hyperparameter of the network. We consider networks with Gaussian initialization so that $W_{ij}^{\text{in}} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_{w,\text{in}}^2/D_{\text{in}})$, $b_i^{\text{in}} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_{b,\text{in}}^2)$, $W_{ij}^{(l)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_w^2/N)$, $b_i^{(l)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_b^2)$, $W_{ij}^{\text{out}} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_{w,\text{out}}^2/N)$, and $b_i^{\text{out}} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_{b,\text{out}}^2)$.

4.3 Theoretical background

As reference we here state the Neural Network Gaussian Process result for residual networks.

4.3.1 NNGP for ResNets

The Neural Network Gaussian Process for residual networks was derived for different residual network architectures by (Huang et al., 2020; Tirer, Bruna, and Giryes, 2022; Barzilai et al., 2023). Adapted to the residual architecture in Eq. (4.1), the

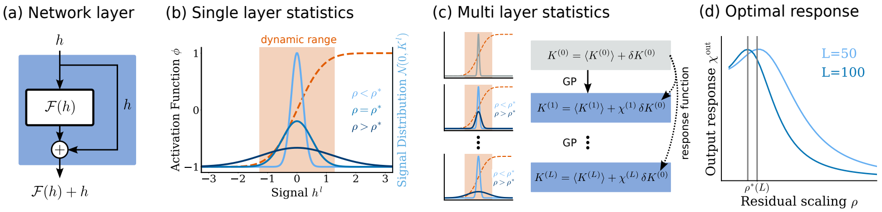

(a) Network layer
h
$\mathcal{F}(h)$
h
$\mathcal{F}(h)+h$
(b) Single layer statistics
dynamic range
Activation Function ϕ
Signal Distribution $\mathcal{N}(0,K^{l})$
$\rho<\rho^*$
$\rho=\rho^*$
$\rho>\rho^*$
1
0
−1
−3 −2 −1 0 1 2 3
Signal h^{l}
(c) Multi layer statistics
$K^{(0)}=\langle K^{(0)}\rangle+\delta K^{(0)}$
GP
$K^{(1)}=\langle K^{(1)}\rangle+\chi^{(1)}\,\delta K^{(0)}$
GP
$K^{(L)}=\langle K^{(L)}\rangle+\chi^{(L)}\,\delta K^{(0)}$
response function
$\rho<\rho^*$
$\rho>\rho^*$
$\rho<\rho^*$
$\rho>\rho^*$
(d) Optimal response
Output response χ^{out}
L=50
L=100
$\rho^*(L)$
Residual scaling ρ

Figure 4.1: (a) The skip connection bypasses the residual branch, mapping to $\mathcal{F}(h)+h$ in each layer. (b) Signal distribution $h^{(l)}$ in layer l (solid curves) before passing through the activation function ϕ = erf (dashed curve). The dynamic range $\mathcal{V}$ of the activation function is indicated by the orange area. The signal $h^{(l)} \sim \mathcal{N}(0, K^{(l)})$ follows a Gaussian distribution with variance $K^{(l)} = K^{(l)}(\rho)$; this variance is a function of the residual scaling parameter ρ. If $\rho > \rho^*$ is too large, the signal is chopped off due to the saturation of the activation function ϕ (dark blue). If $\rho < \rho^*$ is too small, the signal is passed through the part of the activation function that is primarly linear (light blue). In the intermediate regime $\rho = \rho^*$, the signal fills the dynamic range $\mathcal{V}$ of the activation function ϕ (middle blue). (c) Assuming a perturbation $\delta K^{(0)}$ of the input kernel $\langle K^{(0)}\rangle$, the response function $\chi^{(l)}$ measures the resulting change of the kernel $K^{(l)}$ in layer l to linear order. Due to the skip connections, the kernel $K^{(l)}$ receives additions from the residual branches across all layers and thus increases. Since the residual branches are scaled by ρ, the residual scaling parameter ρ determines the rate of increase. If the signal goes into saturation ($\rho > \rho^*$) or stays close to zero ($\rho < \rho^*$), the network output is rather insensitive to changes in the input, which impediments network training. (d) The output response has a unique maximum $\rho^*(L)$ that links to good signal propagation in the network.

corresponding NNGP is given by

$$K_{\alpha\beta}^{(l)} = \begin{cases} \frac{\sigma_{w,\mathrm{in}}^2}{D_{\mathrm{in}}} x_\alpha^{\mathrm{T}} x_\beta + \sigma_{b,\mathrm{in}}^2 & l = 0, \\ K_{\alpha\beta}^{(l-1)} + \rho^2 \sigma_w^2 \left\langle \phi(h_\alpha^{(l-1)})\, \phi(h_\beta^{(l-1)}) \right\rangle_{h^{(l-1)} \sim \mathcal{N}(0, K^{(l-1)})} + \rho^2 \sigma_b^2 & 1 \le l \le L, \\ \sigma_{w,\mathrm{out}}^2 \left\langle \phi(h_\alpha^{(L)})\, \phi(h_\beta^{(L)}) \right\rangle_{h^{(L)} \sim \mathcal{N}(0, K^{(L)})} + \sigma_{b,\mathrm{out}}^2 & l = L+1. \end{cases} \tag{4.2}$$

In contrast to FFNets, the skip connections lead to a recursive structure of the expressions.

4.4 Theory

We first derive the prior of the network in a field-theoretic formulation. From the network prior we recover the NNGP result as the saddle point of the action as well as the response function and fluctuation corrections as the next-to-leading-order corrections to the saddle point. Finally, we link the response function to linear response theory.

4.4.1 Network prior in a field-theoretic framework

We study the network prior of the residual architecture in Eq. (4.1). The network prior $p(Y|X)$ represents the joint distribution of all inputs $X = (x_\alpha)_{\alpha=1,\dots,P}$ and corresponding network outputs $Y = (y_\alpha)_{\alpha=1,\dots,P}$; its derivation is similar to the replica calculation in physics (Zinn-Justin, 1996; Hertz, Krogh, and Palmer, 1991) with one replicon per input x_α with a fixed, shared set of network parameters θ across all replica. We employ the same field-theoretic approach as in Segadlo et al. (2022), who obtain a joint description for both deep feed-forward and recurrent neural networks. The network prior is defined as

$$p(Y|X) = \int \mathrm{d}\theta \prod_\alpha p(y_\alpha|x_\alpha,\theta)\, p(\theta). \tag{4.3}$$

Given a particular realization of network parameters θ, we get the probability $p(Y|X,\theta)$ by enforcing the network mapping with Dirac δ-distributions as

$$\begin{aligned} p(Y|X,\theta) = \prod_\alpha \int \mathrm{d}h_\alpha^{(0)} \cdots \int \mathrm{d}h_\alpha^{(L)}\, & \delta(h_\alpha^{(0)} - W^{\mathrm{in}} x_\alpha - b^{\mathrm{in}}) \\ & \times \prod_{l=1}^{L} \delta(h_\alpha^{(l)} - h_\alpha^{(l-1)} - \rho W^{(l)} \phi(h_\alpha^{(l-1)}) - \rho b^{(l)}) \qquad (4.4) \\ & \times \delta(y_\alpha - W^{\mathrm{out}} \phi(h_\alpha^{(L)}) - b^{\mathrm{out}}). \qquad (4.5) \end{aligned}$$

Marginalization over network parameters

We now take the expectation value with respect to the network parameters

$$\begin{aligned} p(Y|X) = \prod_\alpha \int \mathrm{d}h_\alpha^{(0)} \cdots \int \mathrm{d}h_\alpha^{(L)}\, & \langle \delta(h_\alpha^{(0)} - W^{\mathrm{in}} x_\alpha - b^{\mathrm{in}}) \rangle_{\{W^{\mathrm{in}}, b^{\mathrm{in}}\}} \\ & \times \prod_{l=1}^{L} \langle \delta(h_\alpha^{(l)} - h_\alpha^{(l-1)} - \rho W^{(l)} \phi(h_\alpha^{(l-1)}) - \rho b^{(l)}) \rangle_{\{W^{(l)}, b^{(l)}\}} \\ & \times \langle \delta(y_\alpha - W^{\mathrm{out}} \phi(h_\alpha^{(L)}) - b^{\mathrm{out}}) \rangle_{\{W^{\mathrm{out}}, b^{\mathrm{out}}\}}, \end{aligned} \tag{4.6}$$

where we denote by $\langle \dots \rangle_{\{W,b\}}$ the Gaussian expectation over weights W and biases b. We substitute the Dirac δ-distributions by its Fourier representation

$$\delta(h) = \int \mathrm{d}\tilde{h}\, \exp\left(\tilde{h}^{\mathsf{T}} h\right), \tag{4.7}$$

with $\tilde{h}^{\mathsf{T}} h = \sum_{i=1}^{N} \tilde{h}_i h_i$, where $\int \mathrm{d}\tilde{h} = \prod_k \int_{i\mathbb{R}} \frac{\mathrm{d}\tilde{h}_k}{2\pi i}$ denotes the integration measure and $\tilde{h}$ is the conjugate variable to h.

We obtain

$$\begin{aligned} p(Y|X) = \prod_{\alpha} \Big\{ \int \mathcal{D}\tilde{y}_{\alpha} \int \mathcal{D}\tilde{h}_{\alpha} \int \mathcal{D}h_{\alpha} \Big\} & \Big\langle \exp\Big(\sum_{\alpha,i} \tilde{h}^{(0)}_{i,\alpha} \big(h^{(0)}_{i,\alpha} - \sum_{j} W^{\mathrm{in}}_{ij} x_{j,\alpha} - b^{\mathrm{in}}_{i}\big)\Big)\Big\rangle_{\{W^{\mathrm{in}}, b^{\mathrm{in}}\}} \\ \times \prod_{l=1}^{L} & \Big\langle \exp\Big(\sum_{\alpha,i} \tilde{h}^{(l)}_{i,\alpha} \big(h^{(l)}_{i,\alpha} - h^{(l-1)}_{i,\alpha} - \rho \sum_{j} W^{(l)}_{ij} \phi^{(l-1)}_{j,\alpha} - \rho b^{(l)}_{i}\big)\Big)\Big\rangle_{\{W^{(l)}, b^{(l)}\}} \\ \times & \Big\langle \exp\Big(\sum_{\alpha,i} \tilde{y}_{i,\alpha} \big(y_{i,\alpha} - \sum_{j} W^{\mathrm{out}}_{ij} \phi^{(L)}_{j,\alpha} - b^{\mathrm{out}}_{i}\big)\Big)\Big\rangle_{\{W^{\mathrm{out}}, b^{\mathrm{out}}\}} \,, \end{aligned} \tag{4.8}$$

where $\int \mathcal{D}h_{\alpha} = \prod_{l=0}^{L} \int \mathrm{d}h^{(l)}_{\alpha}$ and $\int \mathcal{D}\tilde{h}_{\alpha} = \prod_{l=0}^{L} \int \mathrm{d}\tilde{h}^{(l)}_{\alpha}$ and we employ the shorthand $\phi^{(l-1)}_{j,\alpha} = \phi(h^{(l-1)}_{j,\alpha})$ for brevity. Since the network parameters θ are distributed independently, the probability distribution factorizes across all parameters and one can calculate the expectation value for each parameter individually as $\int \mathrm{d}\theta_k \, p(\theta_k) \exp\big(z\theta_k\big)$. This integral corresponds to the moment-generating function of the distribution $p(\theta_k)$. For parameters $\theta_k \sim \mathcal{N}(0, \sigma^2)$ that are Gaussian distributed, the moment-generating function gives

$$\int \mathrm{d}\theta_k \, p(\theta_k) \exp\big(z\theta_k\big) = \exp\big(\frac{1}{2}\sigma^2 z^2\big). \tag{4.9}$$

Computing all terms in Eq. (4.8) separately, we have

$$\begin{aligned} \Big\langle \exp\Big(- \sum_{i,j} W^{\mathrm{in}}_{ij} \sum_{\alpha} \tilde{h}^{(0)}_{i,\alpha} x_{j,\alpha} \Big) \Big\rangle_{W^{\mathrm{in}}} &= \exp\Big(\frac{1}{2} \frac{\sigma^2_{w,\mathrm{in}}}{D_{\mathrm{in}}} \sum_{i,j} \Big(\sum_{\alpha} \tilde{h}^{(0)}_{i,\alpha} x_{j,\alpha} \Big)^2 \Big) \\ &= \exp\Big(\frac{1}{2} \frac{\sigma^2_{w,\mathrm{in}}}{D_{\mathrm{in}}} \sum_{\alpha,\beta} \sum_{i,j} \tilde{h}^{(0)}_{i,\alpha} x_{j,\alpha} \, \tilde{h}^{(0)}_{i,\beta} x_{j,\beta} \Big), \\ \Big\langle \exp\Big(- \sum_{i} b^{\mathrm{in}}_{i} \sum_{\alpha} \tilde{h}^{(0)}_{i,\alpha} \Big) \Big\rangle_{b^{\mathrm{in}}} &= \exp\Big(\frac{1}{2} \sigma^2_{b,\mathrm{in}} \sum_{i} \Big(\sum_{\alpha} \tilde{h}^{(0)}_{i,\alpha} \Big)^2 \Big) \\ &= \exp\Big(\frac{1}{2} \sigma^2_{b,\mathrm{in}} \sum_{\alpha,\beta} \sum_{i} \tilde{h}^{(0)}_{i,\alpha} \, \tilde{h}^{(0)}_{i,\beta} \Big), \\ \Big\langle \exp\Big(- \sum_{i,j} W^{(l)}_{ij} \, \rho \sum_{\alpha} \tilde{h}^{(l)}_{i,\alpha} \phi^{(l-1)}_{j,\alpha} \Big) \Big\rangle_{W^{(l)}} &= \exp\Big(\frac{1}{2} \frac{\sigma^2_{w}}{N} \sum_{i,j} \Big(\rho \sum_{\alpha} \tilde{h}^{(l)}_{i,\alpha} \phi^{(l-1)}_{j,\alpha} \Big)^2 \Big) \\ &= \exp\Big(\frac{1}{2} \rho^2 \frac{\sigma^2_{w}}{N} \sum_{\alpha,\beta} \sum_{i,j} \tilde{h}^{(l)}_{i,\alpha} \phi^{(l-1)}_{j,\alpha} \, \tilde{h}^{(l)}_{i,\beta} \phi^{(l-1)}_{j,\beta} \Big), \\ \Big\langle \exp\Big(- \sum_{i} b^{(l)}_{i} \, \rho \sum_{\alpha} \tilde{h}^{(l)}_{i,\alpha} \Big) \Big\rangle_{b^{(l)}} &= \exp\Big(\frac{1}{2} \sigma^2_{b} \sum_{i} \Big(\rho \sum_{\alpha} \tilde{h}^{(l)}_{i,\alpha} \Big)^2 \Big) \\ &= \exp\Big(\frac{1}{2} \rho^2 \sigma^2_{b} \sum_{\alpha,\beta} \sum_{i} \tilde{h}^{(l)}_{i,\alpha} \, \tilde{h}^{(l)}_{i,\beta} \Big), \end{aligned}$$

$$
\begin{aligned}
\left\langle \exp\Big(-\sum_{i,j} W^{\mathrm{out}}_{ij} \sum_{\alpha} \tilde{y}_{i,\alpha}\phi^{(L)}_{j,\alpha}\Big)\right\rangle_{W^{\mathrm{out}}} &= \exp\Big(\frac{1}{2}\frac{\sigma^2_{w,\mathrm{out}}}{N}\sum_{i,j}\Big(\sum_{\alpha}\tilde{y}_{i,\alpha}\phi^{(L)}_{j,\alpha}\Big)^2\Big) \\
&= \exp\Big(\frac{1}{2}\frac{\sigma^2_{w,\mathrm{out}}}{N}\sum_{\alpha,\beta}\sum_{i,j}\tilde{y}_{i,\alpha}\phi^{(L)}_{j,\alpha}\,\tilde{y}_{i,\beta}\phi^{(L)}_{j,\beta}\Big), \\
\left\langle \exp\Big(-\sum_{i} b^{\mathrm{out}}_{i} \sum_{\alpha} \tilde{y}_{i,\alpha}\Big)\right\rangle_{b^{\mathrm{out}}} &= \exp\Big(\frac{1}{2}\sigma^2_{b,\mathrm{out}}\sum_{i}\Big(\sum_{\alpha}\tilde{y}_{i,\alpha}\Big)^2\Big) \\
&= \exp\Big(\frac{1}{2}\sigma^2_{b,\mathrm{out}}\sum_{\alpha,\beta}\sum_{i}\tilde{y}_{i,\alpha}\tilde{y}_{i,\beta}\Big).
\end{aligned}
$$

In the following, we adopt an implicit summation convention for repeated lower indices in the exponent, for example, $\sum_\alpha \sum_i \tilde{h}^{(l)}_{i,\alpha}\tilde{h}^{(l)}_{i,\alpha} = \tilde{h}^{(l)}_{i,\alpha}\tilde{h}^{(l)}_{i,\alpha}$. Additionally, we jointly denote $\int \mathcal{D}\tilde{h} = \prod_\alpha \int \mathcal{D}\tilde{h}_\alpha$. By rewriting the sums over neuron indices in terms of scalar products, we obtain the following expression for the network prior

$$
\begin{aligned}
p(Y|X) = \int \mathcal{D}\tilde{y}\int \mathcal{D}\tilde{h}\int \mathcal{D}h\, &\exp\left(\tilde{y}_\alpha^{\mathsf{T}} y_\alpha + \frac{1}{2}\frac{\sigma^2_{w,\mathrm{out}}}{N}\,\tilde{y}_\alpha^{\mathsf{T}}\tilde{y}_\beta\,\left[\phi^{(L)}_\alpha\right]^{\mathsf{T}}\phi^{(L)}_\beta + \frac{1}{2}\sigma^2_{b,\mathrm{out}}\sum_{\alpha,\beta}\tilde{y}_\alpha^{\mathsf{T}}\tilde{y}_\beta\right) \\
\times &\exp\left(\sum_{l=1}^{L}\left[\tilde{h}^{(l)}_\alpha\right]^{\mathsf{T}}\left[h^{(l)}_\alpha - h^{(l-1)}_\alpha\right]\right) \\
\times &\exp\left(\sum_{l=1}^{L}\left(\frac{1}{2}\rho^2\frac{\sigma_w^2}{N}\left[\tilde{h}^{(l)}_\alpha\right]^{\mathsf{T}}\tilde{h}^{(l)}_\beta\left[\phi^{(l-1)}_\alpha\right]^{\mathsf{T}}\phi^{(l-1)}_\beta + \frac{1}{2}\rho^2\sigma_b^2\sum_{\alpha,\beta}\left[\tilde{h}^{(l)}_\alpha\right]^{\mathsf{T}}\tilde{h}^{(l)}_\beta\right)\right) \\
\times &\exp\left(\left[\tilde{h}^{(0)}_\alpha\right]^{\mathsf{T}} h^{(0)}_\alpha + \frac{1}{2}\frac{\sigma^2_{w,\mathrm{in}}}{D_{\mathrm{in}}}\left[\tilde{h}^{(0)}_\alpha\right]^{\mathsf{T}}\tilde{h}^{(0)}_\beta\, {x_\alpha}^{\mathsf{T}} x_\beta + \frac{1}{2}\sigma^2_{b,\mathrm{in}}\sum_{\alpha,\beta}\left[\tilde{h}^{(0)}_\alpha\right]^{\mathsf{T}}\tilde{h}^{(0)}_\beta\right) \\
=: \int \mathcal{D}\tilde{y}\int \mathcal{D}\tilde{h}\int \mathcal{D}h\, &\exp\left(\mathcal{S}(Y,\tilde{Y},H,\tilde{H}|X)\right).
\end{aligned}
$$

The action $\mathcal{S}$ of the network prior is given by

$$
\mathcal{S}(Y,\tilde{Y},H,\tilde{H}|X) = \mathcal{S}_{\mathrm{in}}(H^{(0)},\tilde{H}^{(0)}|X) + \mathcal{S}_{\mathrm{net}}(H,\tilde{H}) + \mathcal{S}_{\mathrm{out}}(Y,\tilde{Y}|H^{(L)}), \tag{4.10}
$$

where we distinguish between the readin layer containing the dependence on the inputs X as

$$
\mathcal{S}_{\mathrm{in}}(H^{(0)},\tilde{H}^{(0)}|X) \coloneqq \left[\tilde{h}^{(0)}_\alpha\right]^{\mathsf{T}} h^{(0)}_\alpha + \frac{1}{2}\frac{\sigma^2_{w,\mathrm{in}}}{D_{\mathrm{in}}}\left[\tilde{h}^{(0)}_\alpha\right]^{\mathsf{T}}\tilde{h}^{(0)}_\beta\,(XX^{\mathsf{T}})_{\alpha\beta} + \frac{1}{2}\sigma^2_{b,\mathrm{in}}\sum_{\alpha,\beta}\left[\tilde{h}^{(0)}_\alpha\right]^{\mathsf{T}}\tilde{h}^{(0)}_\beta, \tag{4.11}
$$

the hidden layers of the network involving skip connections as

$$
\begin{aligned}
\mathcal{S}_{\mathrm{net}}(H,\tilde{H}) \coloneqq \sum_{l=1}^{L}\left[\tilde{h}^{(l)}_\alpha\right]^{\mathsf{T}}\left[h^{(l)}_\alpha - h^{(l-1)}_\alpha\right] &+ \frac{1}{2}\rho^2\frac{\sigma_w^2}{N}\left[\tilde{h}^{(l)}_\alpha\right]^{\mathsf{T}}\tilde{h}^{(l)}_\beta\left[\phi^{(l-1)}_\alpha\right]^{\mathsf{T}}\phi^{(l-1)}_\beta \\
&+ \frac{1}{2}\rho^2\sigma_b^2\sum_{\alpha,\beta}\left[\tilde{h}^{(l)}_\alpha\right]^{\mathsf{T}}\tilde{h}^{(l)}_\beta,
\end{aligned} \tag{4.12}
$$

and the readout layer containing the dependence on the network outputs Y as

$$\mathcal{S}_{\text{out}}(Y,\tilde{Y}|H^{(L)}) \coloneqq \tilde{y}_\alpha^\mathsf{T} y_\alpha + \frac{1}{2}\frac{\sigma^2_{w,\text{out}}}{N}\tilde{y}_\alpha^\mathsf{T}\tilde{y}_\beta \left[\phi_\alpha^{(L)}\right]^\mathsf{T}\phi_\beta^{(L)} + \frac{1}{2}\sigma^2_{b,\text{out}}\tilde{y}_\alpha^\mathsf{T}\tilde{y}_\beta. \tag{4.13}$$

In contrast to feed-forward networks, the conjugate variables $\tilde{h}^{(l)}$ of layer l do not only couple to the signal $h^{(l)}$ of layer l, but also to the signal $h^{(l-1)}$ of the previous layer $l-1$. This coupling across layers arises due to the skip connections in residual networks. The interdependence between layers induced by the coupling prohibits the marginalization over the intermediate signals $h^{(l)}$ in a direct manner as for feed-forward networks.

Auxiliary variables

Quartic terms $\propto \left[\tilde{h}^{(l)}\right]^\mathsf{T}\tilde{h}^{(l)}\,\phi^{(l-1)^\mathsf{T}}\phi^{(l-1)}$ cannot be solved analytically for general activation functions ϕ. Instead, we introduce auxiliary variables

$$C_{\alpha\beta}^{(l)} \coloneqq \begin{cases} \frac{\sigma^2_{w,\text{in}}}{D_\text{in}}(XX^\mathsf{T})_{\alpha\beta} + \sigma^2_{b,\text{in}} & l=0, \\ \rho^2\frac{\sigma_w^2}{N}\phi_\alpha^{(l-1)}\cdot\phi_\beta^{(l-1)} + \rho^2\sigma_b^2 & 1\le l\le L, \\ \frac{\sigma^2_{w,\text{out}}}{N}\phi_\alpha^{(L)}\cdot\phi_\beta^{(L)} + \sigma^2_{b,\text{out}} & l=L+1, \end{cases}$$

which decouple these into quadratic terms again. While we need to decouple these terms for analytical feasability, we are interested precisely in effects that result from this coupling. Therefore, we keep track of the original interaction between $\tilde{h}$ and $\phi^{(l-1)}$ by enforcing the definition of the auxiliary variables. We again use Dirac δ-distributions and get

$$\begin{aligned} p(Y|X) = \int\mathcal{D}\tilde{y}\int\mathcal{D}\tilde{h}\int\mathcal{D}h\prod_{\alpha,\beta}\int\mathcal{D}C_{\alpha\beta}\,\exp\left(\tilde{y}_\alpha^\mathsf{T}y_\alpha + \frac{1}{2}C_{\alpha\beta}^{(L+1)}\tilde{y}_\alpha^\mathsf{T}\tilde{y}_\beta\right) \\ \times\,\delta\left(C_{\alpha\beta}^{(L+1)} - \frac{\sigma^2_{w,\text{out}}}{N}\left[\phi_\alpha^{(L)}\right]^\mathsf{T}\phi_\beta^{(L)} - \sigma^2_{b,\text{out}}\right) \\ \times\exp\left(\sum_{l=1}^{L}\left[\left[\tilde{h}_\alpha^{(l)}\right]^\mathsf{T}\left[h_\alpha^{(l)} - h_\alpha^{(l-1)}\right] + \frac{1}{2}C_{\alpha\beta}^{(l)}\left[\tilde{h}_\alpha^{(l)}\right]^\mathsf{T}\tilde{h}_\beta^{(l)}\right]\right) \\ \times\,\delta\left(C_{\alpha\beta}^{(l)} - \rho^2\frac{\sigma_w^2}{N}\left[\phi_\alpha^{(l-1)}\right]^\mathsf{T}\phi_\beta^{(l-1)} - \rho^2\sigma_b^2\right) \\ \times\exp\left(\left[\tilde{h}_\alpha^{(0)}\right]^\mathsf{T}h_\alpha^{(0)} + \frac{1}{2}C_{\alpha\beta}^{(0)}\left[\tilde{h}_\alpha^{(0)}\right]^\mathsf{T}\tilde{h}_\beta^{(0)}\right)\delta\left(C_{\alpha\beta}^{(0)} - \frac{\sigma^2_{w,\text{in}}}{D_\text{in}}x_\alpha\cdot x_\beta - \sigma^2_{b,\text{in}}\right), \end{aligned}$$

where $\int\mathcal{D}C_{\alpha\beta} = \prod_{l=0}^{L+1}\int\mathcal{D}C_{\alpha\beta}^{(l)}$. As before, we rewrite the Dirac δ-distributions using their Fourier representation $\delta(C_{\alpha\beta}^{(l)}) = \int_{i\mathbb{R}}\frac{\mathrm{d}\tilde{C}_{\alpha\beta}^{(l)}}{2\pi i}\exp\left(\tilde{C}_{\alpha\beta}^{(l)}\,C_{\alpha\beta}^{(l)}\right)$, which leads us to introducing conjugate variables $\tilde{C}^{(l)}_{\alpha\beta}$ to the auxiliary variables $C^{(l)}_{\alpha\beta}$:

$$\begin{aligned}
&\delta\left(D_{\text{in}}C^{(0)}_{\alpha\beta} - \sigma^2_{w,\text{in}}\, x_\alpha \cdot x_\beta - D_{\text{in}}\sigma^2_{b,\text{in}}\right) \\
&\quad = \int_{i\mathbb{R}} \frac{\mathrm{d}\tilde{C}^{(0)}_{\alpha\beta}}{2\pi i} \exp\left(D_{\text{in}}\tilde{C}^{(0)}_{\alpha\beta}C^{(0)}_{\alpha\beta} - \sigma^2_{w,\text{in}}\,\tilde{C}^{(0)}_{\alpha\beta}\, x_\alpha \cdot x_\beta - D_{\text{in}}\sigma^2_{b,\text{in}}\,\tilde{C}^{(0)}_{\alpha\beta}\right), \\
&\delta\left(NC^{(l)}_{\alpha\beta} - \rho^2\sigma^2_w \left[\phi^{(l-1)}_\alpha\right]^\mathsf{T} \phi^{(l-1)}_\beta - N\rho^2\sigma^2_b\right) \\
&\quad = \int_{i\mathbb{R}} \frac{\mathrm{d}\tilde{C}^{(l)}_{\alpha\beta}}{2\pi i} \exp\left(N\tilde{C}^{(l)}_{\alpha\beta}C^{(l)}_{\alpha\beta} - \rho^2\sigma^2_w\,\tilde{C}^{(l)}_{\alpha\beta} \left[\phi^{(l-1)}_\alpha\right]^\mathsf{T} \phi^{(l-1)}_\beta - N\rho^2\sigma^2_b\,\tilde{C}^{(l)}_{\alpha\beta}\right), \\
&\delta\left(NC^{(L+1)}_{\alpha\beta} - \sigma^2_{w,\text{out}} \left[\phi^{(L)}_\alpha\right]^\mathsf{T} \phi^{(L)}_\beta - N\sigma^2_{b,\text{out}}\right) \\
&\quad = \int_{i\mathbb{R}} \frac{\mathrm{d}\tilde{C}^{(L+1)}_{\alpha\beta}}{2\pi i} \exp\left(N\tilde{C}^{(L+1)}_{\alpha\beta}C^{(L+1)}_{\alpha\beta} - \sigma^2_{w,\text{out}}\,\tilde{C}^{(L+1)}_{\alpha\beta} \left[\phi^{(L)}_\alpha\right]^\mathsf{T} \phi^{(L)}_\beta - N\sigma^2_{b,\text{out}}\,\tilde{C}^{(L+1)}_{\alpha\beta}\right).
\end{aligned}$$

Then, the network prior can be written as

$$\begin{aligned}
p(Y|X) = \int \mathcal{D}\tilde{y} \int \mathcal{D}\tilde{h} \int \mathcal{D}h \int \mathcal{D}C \int \mathcal{D}\tilde{C} \, &\exp\left(\tilde{y}^\mathsf{T}_\alpha y_\alpha + \frac{1}{2}C^{(L+1)}_{\alpha\beta}\tilde{y}^\mathsf{T}_\alpha \tilde{y}_\beta\right) \\
&\times \exp\left(\sum_{l=1}^{L}\left[\left[\tilde{h}^{(l)}_\alpha\right]^\mathsf{T}\left[h^{(l)}_\alpha - h^{(l-1)}_\alpha\right] + \frac{1}{2}C^{(l)}_{\alpha\beta}\left[\tilde{h}^{(l)}_\alpha\right]^\mathsf{T}\tilde{h}^{(l)}_\beta\right]\right) \\
&\times \exp\left(\left[\tilde{h}^{(0)}_\alpha\right]^\mathsf{T} h^{(0)}_\alpha + \frac{1}{2}C^{(0)}_{\alpha\beta}\left[\tilde{h}^{(0)}_\alpha\right]^\mathsf{T}\tilde{h}^{(0)}_\beta\right) \\
&\times \exp\left(-N\sum_{l=0}^{L+1}\nu_l\, C^{(l)}_{\alpha\beta}\,\tilde{C}^{(l)}_{\alpha\beta} + \rho^2\sigma^2_w\sum_{l=1}^{L}\tilde{C}^{(l)}_{\alpha\beta}\left[\phi^{(l-1)}_\alpha\right]^\mathsf{T}\phi^{(l-1)}_\beta + N\rho^2\sigma^2_b\sum_{l=1}^{L}\sum_{\alpha,\beta}\tilde{C}^{(l)}_{\alpha\beta}\right) \\
&\times \exp\left(\sigma^2_{w,\text{out}}\,\tilde{C}^{(L+1)}_{\alpha\beta}\left[\phi^{(L)}_\alpha\right]^\mathsf{T}\phi^{(L)}_\beta + N\sigma^2_{b,\text{out}}\sum_{\alpha,\beta}\tilde{C}^{(L+1)}_{\alpha\beta}\right) \\
&\times \exp\left(\sigma^2_{w,\text{in}}\,\tilde{C}^{(0)}_{\alpha\beta}\,(XX^\mathsf{T})_{\alpha\beta} + D_{\text{in}}\,\sigma^2_{b,\text{in}}\sum_{\alpha,\beta}\tilde{C}^{(0)}_{\alpha\beta}\right),
\end{aligned}$$

where we use the shorthands $\int \mathcal{D}C = \prod_{\alpha,\beta}\left\{\int \mathcal{D}C_{\alpha\beta}\right\}$, $\int \mathcal{D}\tilde{C} = \prod_{\alpha,\beta}\prod_{l=0}^{L+1}\int_{i\mathbb{R}}\frac{\mathrm{d}\tilde{C}^{(l)}_{\alpha\beta}}{2\pi i}$, and $\nu_l = 1 + \delta_{0l}\,(d_{\text{in}}/N - 1)$. Since the variables $C^{(l)}_{\alpha\beta}$ and $\tilde{C}^{(l)}_{\alpha\beta}$ only couple to sums of $\tilde{h}^{(l)}$ and $\phi^{(l)}$ over all neuron indices and are scalar quantities themselves, we find that all components of $h^{(l)}$ and $\tilde{h}^{(l)}$ follow the same distribution. Thus, we replace all scalar products by products of scalar variables $h^{(l)}$ and $\tilde{h}^{(l)}$ and pull out a factor N, yielding

$$\begin{aligned}
p(Y|X) = \int \mathcal{D}\tilde{y} \int \mathcal{D}\tilde{h} \int \mathcal{D}h \int \mathcal{D}C \int \mathcal{D}\tilde{C} \, &\exp\left(\tilde{y}_\alpha^\mathsf{T} y_\alpha + \frac{1}{2} C_{\alpha\beta}^{(L+1)} \tilde{y}_\alpha^\mathsf{T} \tilde{y}_\beta\right) \\
&\times \exp\left(N \sum_{l=1}^{L} \left[\tilde{h}_\alpha^{(l)} \left[h_\alpha^{(l)} - h_\alpha^{(l-1)}\right] + \frac{1}{2} \tilde{h}_\alpha^{(l)} C_{\alpha\beta}^{(l)} \tilde{h}_\beta^{(l)}\right]\right) \\
&\times \exp\left(N \sum_\alpha \tilde{h}_\alpha^{(0)} h_\alpha^{(0)} + N \frac{1}{2} \sum_{\alpha,\beta} \tilde{h}_\alpha^{(0)} C_{\alpha\beta}^{(0)} \tilde{h}_\beta^{(0)}\right) \\
&\times \exp\left(-N \sum_{l=0}^{L+1} \nu_l \, C_{\alpha\beta}^{(l)} \, \tilde{C}_{\alpha\beta}^{(l)} + N\rho^2 \sigma_w^2 \sum_{l=1}^{L} \phi_\alpha^{(l-1)} \tilde{C}_{\alpha\beta}^{(l)} \phi_\beta^{(l-1)} + N\rho^2 \sigma_b^2 \sum_{l=1}^{L} \sum_{\alpha,\beta} \tilde{C}_{\alpha\beta}^{(l)}\right) \\
&\times \exp\left(N\sigma_{w,\,\mathrm{out}}^2 \phi_\alpha^{(L)} \tilde{C}_{\alpha\beta}^{(L+1)} \phi_\beta^{(L)} + N\sigma_{b,\,\mathrm{out}}^2 \sum_{\alpha,\beta} \tilde{C}_{\alpha\beta}^{(L+1)}\right) \\
&\times \exp\left(\sigma_{w,\,\mathrm{in}}^2 \, \tilde{C}_{\alpha\beta}^{(0)} \, (XX^\mathsf{T})_{\alpha\beta} + D_\mathrm{in} \, \sigma_{b,\,\mathrm{in}}^2 \sum_{\alpha,\beta} \tilde{C}_{\alpha\beta}^{(0)}\right).
\end{aligned}$$

The auxiliary variables $C_{\alpha\beta}^{(l)}$ will be the order parameters of the system, so we aim for an expression solely in these variables. To this end, we move the integrals over $h^{(l)}$ and $\tilde{h}^{(l)}$ into the exponent and obtain

$$\begin{aligned}
p(Y|X) &= \int \mathcal{D}\tilde{y} \int \mathcal{D}C \int \mathcal{D}\tilde{C} \, \exp\left(\tilde{y}_\alpha^\mathsf{T} y_\alpha + \frac{1}{2} C_{\alpha\beta}^{(L+1)} \tilde{y}_\alpha^\mathsf{T} \tilde{y}_\beta - N \sum_{l=0}^{L+1} \nu_l \, C_{\alpha\beta}^{(l)} \, \tilde{C}_{\alpha\beta}^{(l)}\right) \\
&\qquad \times \exp\Bigg[N \ln \prod_{l=1}^{L} \int \mathcal{D}\tilde{h}^{(l)} \int \mathcal{D}h^{(l)} \, \exp\left(\tilde{h}_\alpha^{(l)} \left[h_\alpha^{(l)} - h_\alpha^{(l-1)}\right] + \frac{1}{2} \tilde{h}_\alpha^{(l)} C_{\alpha\beta}^{(l)} \tilde{h}_\beta^{(l)}\right) \\
&\qquad\qquad \times \exp\left(\rho^2 \sigma_w^2 \sum_{l=1}^{L} \phi_\alpha^{(l-1)} \tilde{C}_{\alpha\beta}^{(l)} \phi_\beta^{(l-1)} + \rho^2 \sigma_b^2 \sum_{l=1}^{L} \sum_{\alpha,\beta} \tilde{C}_{\alpha\beta}^{(l)}\right) \\
&\qquad\qquad \times \exp\left(\sigma_{w,\,\mathrm{out}}^2 \phi_\alpha^{(L)} \tilde{C}_{\alpha\beta}^{(L+1)} \phi_\beta^{(L)} + \sigma_{b,\,\mathrm{out}}^2 \sum_{\alpha,\beta} \tilde{C}_{\alpha\beta}^{(L+1)}\right) \\
&\qquad\qquad \times \int \mathcal{D}\tilde{h}^{(0)} \int \mathcal{D}h^{(0)} \exp\left(\tilde{h}_\alpha^{(0)} h_\alpha^{(0)} + \frac{1}{2} \tilde{h}_\alpha^{(0)} C_{\alpha\beta}^{(0)} \tilde{h}_\beta^{(0)}\right) \\
&\qquad\qquad\qquad \times \exp\left(\sigma_{w,\,\mathrm{in}}^2 \, \tilde{C}_{\alpha\beta}^{(0)} \, (XX^\mathsf{T})_{\alpha\beta} + D_\mathrm{in} \sigma_{b,\,\mathrm{in}}^2 \sum_{\alpha,\beta} \tilde{C}_{\alpha\beta}^{(0)}\right)\Bigg] \\
&= \int \mathcal{D}\tilde{y} \left\langle \exp\left(\tilde{y}_\alpha^\mathsf{T} y_\alpha + \frac{1}{2} \tilde{y}_\alpha^\mathsf{T} C_{\alpha\beta}^{(L+1)} \tilde{y}_\beta\right)\right\rangle_{C,\tilde{C}}.
\end{aligned}$$

The expectation value in the last line is taken with respect to the auxiliary variables and their conjugate variables, both of which follow a distribution determined by the auxiliary action $(C,\tilde{C}) \sim \exp(S(C,\tilde{C}))$ that is defined as

$$
\begin{aligned}
\mathcal{S}(C,\tilde{C}) &\coloneqq -N\sum_{l=0}^{L+1}\nu_l\, C^{(l)}_{\alpha\beta}\,\tilde{C}^{(l)}_{\alpha\beta} + N\mathcal{W}(\tilde{C}|C),\\
\mathcal{W}(\tilde{C}|C) &\coloneqq \ln\prod_{l=1}^{L}\int\mathcal{D}\tilde{h}^{(l)}\int\mathcal{D}h^{(l)}\,\exp\left(\tilde{h}^{(l)}_{\alpha}\left[h^{(l)}_{\alpha}-h^{(l-1)}_{\alpha}\right]+\frac{1}{2}\tilde{h}^{(l)}_{\alpha}C^{(l)}_{\alpha\beta}\tilde{h}^{(l)}_{\beta}\right)\\
&\quad\times\exp\left(\rho^2\sigma_w^2\sum_{l=1}^{L}\phi^{(l-1)}_{\alpha}\tilde{C}^{(l)}_{\alpha\beta}\phi^{(l-1)}_{\beta}+\rho^2\sigma_b^2\sum_{l=1}^{L}\sum_{\alpha,\beta}\tilde{C}^{(l)}_{\alpha\beta}\right)\\
&\quad\times\exp\left(\sigma^2_{w,\mathrm{out}}\phi^{(L)}_{\alpha}\tilde{C}^{(L+1)}_{\alpha\beta}\phi^{(L)}_{\beta}+\sigma^2_{b,\mathrm{out}}\sum_{\alpha,\beta}\tilde{C}^{(L+1)}_{\alpha\beta}\right)\\
&\quad\times\int\mathcal{D}\tilde{h}^{(0)}\int\mathcal{D}h^{(0)}\,\exp\Bigg(N\tilde{h}^{(0)}_{\alpha}h^{(0)}_{\alpha}+N\frac{1}{2}\tilde{h}^{(0)}_{\alpha}C^{(0)}_{\alpha\beta}\tilde{h}^{(0)}_{\beta}\\
&\qquad\qquad+\sigma^2_{w,\mathrm{in}}\,\tilde{C}^{(0)}_{\alpha\beta}\,(XX^{\mathsf{T}})_{\alpha\beta}+D_{\mathrm{in}}\sigma^2_{b,\mathrm{in}}\sum_{\alpha,\beta}\tilde{C}^{(0)}_{\alpha\beta}\Bigg).
\end{aligned}
$$

Saddle point approximation yields NNGP result

The action $\mathcal{S}$ is proportional to the network width N. So in the limit of infinite width $N\to\infty$, we can evaluate averages by performing a saddle point approximation

$$
\int\mathcal{D}C\int\mathcal{D}\tilde{C}\,f(C,\tilde{C})\,\exp\left(\mathcal{S}(C,\tilde{C})\right)\overset{N\to\infty}{=}f(C_*,\tilde{C}_*),\tag{4.14}
$$

where C_* and $\tilde{C}_*$ are the saddle points of the action $\mathcal{S}$. These fulfill the conditions

$$
\frac{\partial\mathcal{S}}{\partial C^{(l)}_{\alpha\beta}}\Big|_{(C_*,\tilde{C}_*)}\overset{!}{=}0,\;\frac{\partial\mathcal{S}}{\partial\tilde{C}^{(l)}_{\alpha\beta}}\Big|_{(C_*,\tilde{C}_*)}\overset{!}{=}0.\tag{4.15}
$$

Solving for C_*, $\tilde{C}_*$ we get

$$
\begin{aligned}
C^{(l)}_{\alpha\beta,*} &= \begin{cases}\frac{\sigma^2_{w,\mathrm{in}}}{D_{\mathrm{in}}}(XX^{\mathsf{T}})_{\alpha\beta}+\sigma^2_{b,\mathrm{in}} & l=0,\\ \rho^2\sigma_w^2\,\langle\phi^{(l-1)}_{\alpha}\,\phi^{(l-1)}_{\beta}\rangle_p+\rho^2\sigma_b^2 & 1\le l\le L,\\ \sigma^2_{w,\mathrm{out}}\,\langle\phi^{(L)}_{\alpha}\,\phi^{(L)}_{\beta}\rangle_p+\sigma^2_{b,\mathrm{out}} & l=L+1,\end{cases}\\
\tilde{C}^{(l)}_{*} &= \;0 \qquad\qquad\qquad\qquad\qquad\qquad l=0,\dots,L+1,
\end{aligned}
$$

where

$$
\begin{aligned}
\langle\dots\rangle_p = \prod_{l=1}^{L}\int\mathcal{D}\tilde{h}^{(l)}\int\mathcal{D}h^{(l)}\,\dots\,\exp\left(\tilde{h}^{(l)}_{\alpha}\left[h^{(l)}_{\alpha}-h^{(l-1)}_{\alpha}\right]+\frac{1}{2}\tilde{h}^{(l)}_{\alpha}C^{(l)}_{\alpha\beta,*}\tilde{h}^{(l)}_{\beta}\right)\\
\times\int\mathcal{D}\tilde{h}^{(0)}\int\mathcal{D}h^{(0)}\,\exp\left(\tilde{h}^{(0)}_{\alpha}h^{(0)}_{\alpha}+\frac{1}{2}\tilde{h}^{(0)}_{\alpha}C^{(0)}_{\alpha\beta,*}\tilde{h}^{(0)}_{\beta}\right).
\end{aligned}
$$

The input kernel $C^0 = C^0_*$ is fixed by the inputs; the saddle points $C^{(l)}_*$ for all other layers l need to be determined self-consistently. We rewrite the expectation value $\langle \dots \rangle_p$ appearing in the self-consistency equations in terms of the residual $f^{(l)} = h^{(l)} - h^{(l-1)}$ for $1 \leq l \leq L$:

$$\begin{aligned}\langle \dots \rangle_p &= \int \mathcal{D}h^{(0)} \int \mathcal{D}\tilde{h}^{(0)} \exp\Big(\tilde{h}^{(0)}_\alpha h^{(0)}_\alpha + \frac{1}{2}\tilde{h}^{(0)}_\alpha C^{(0)}_{\alpha\beta,*}\tilde{h}^{(0)}_\beta\Big) \\ &\quad \times \prod_{l=1}^{L} \int \mathcal{D}f^{(l)} \int \mathcal{D}\tilde{h}^{(l)} \dots \exp\Big(\tilde{h}^{(l)}_\alpha f^{(l)}_\alpha + \frac{1}{2}\tilde{h}^{(l)}_\alpha C^{(l)}_{\alpha\beta,*}\tilde{h}^{(l)}_\beta\Big) \qquad (4.16) \\ &= \int \mathcal{D}h^{(0)} \mathcal{N}\Big(h^{(0)}|0, C^{(0)}_{\alpha\beta,*}\Big) \prod_{l=1}^{L} \int \mathcal{D}f^{(l)} \dots \mathcal{N}\Big(f^{(l)}|0, C^{(l)}_{\alpha\beta,*}\Big), \qquad (4.17)\end{aligned}$$

with $\mathcal{N}\Big(f^{(l)}|0, C^{(l)}_{\alpha\beta,*}\Big)$ being a multi-dimensional Gaussian with zero mean and covariance $C^{(l)}_{\alpha\beta,*}$. We see that the input signal $H^{(0)} = (h^{(0)}_\alpha)_{\alpha=1,\dots,P}$ is a centered Gaussian with covariance $C^{(0)}_{\alpha\beta,*}$ and the residuals $F^{(l)} = (f^{(l)}_\alpha)_{\alpha=1,\dots,P}$ are centerd Gaussians with covariance $C^{(l)}_{\alpha\beta,*}$. We are ultimately interested in the signal $h^{(l)}$ in layer l, which relates to the residuals by $h^{(l)} = h^{(0)} + \sum_{k=1}^{l} f^{(k)}$. Since the residuals $f^{(l)}$ are independent Gaussian variables, their means and covariances sum up accordingly. As a result, the signal $h^{(l)}$ follows a Gaussian distribution with covariance $K^{(l)} = \sum_{k=0}^{l} C^{(k)}_*$. The summation can be written as a recursion relation $K^{(l)} = K^{(l-1)} + C^{(l-1)}_*$, from which we recover the NNGP result (Huang et al., 2020; Tirer, Bruna, and Giryes, 2022; Barzilai et al., 2023) for residual networks

$$C^{(l)}_{\alpha\beta,*} = \rho^2\sigma_w^2 \langle \phi^{(l-1)}_\alpha \phi^{(l-1)}_\beta \rangle_{\mathcal{N}(0,K^{(l-1)})} + \rho^2\sigma_b^2 \quad 1 \leq l \leq L, \qquad (4.18)$$

$$K^{(l)}_{\alpha\beta} = \begin{cases} \frac{\sigma^2_{w,\mathrm{in}}}{D_{\mathrm{in}}}(XX^\mathsf{T})_{\alpha\beta} + \sigma^2_{b,\mathrm{in}} & l = 0, \\ K^{(l-1)}_{\alpha\beta} + C^{(l-1)}_{\alpha\beta,*} & 1 \leq l \leq L, \\ \sigma^2_{w,\mathrm{out}} \langle \phi^{(L)}_\alpha \phi^{(L)}_\beta \rangle_{\mathcal{N}(0,K^{(L)})} + \sigma^2_{b,\mathrm{out}} & l = L+1. \end{cases} \qquad (4.19)$$

4.4.2 Next-to-leading-order correction

The field-theoretic formulation of residual networks allows us to go beyond the saddle point, which is the NNGP result, determining next-to-leading-order corrections to these saddle points. For large but finite width N, the auxiliary variables $C^{(l)}$ fluctuate around the saddle point value $C^{(l)}_*$. In a first-order approximation, these fluctuations are Gaussian themselves. We determine these fluctuation corrections by evaluating the Hessian of the action $\mathcal{S}$ at the saddle point

$$\begin{aligned} p(Y|X) &\simeq \int \mathcal{D}\delta C \int \mathcal{D}\delta\tilde{C} \exp\Big(\frac{1}{2}(\delta C, \delta\tilde{C})^\mathsf{T} \mathcal{S}^{(2)} (\delta C, \delta\tilde{C})\Big) \\ &= \int \mathcal{D}\delta C \int \mathcal{D}\delta\tilde{C} \exp\Big(-\frac{1}{2}(\delta C, \delta\tilde{C})^\mathsf{T} \Delta^{(2)} (\delta C, \delta\tilde{C})\Big). \end{aligned}$$

We denote the fluctuations as $\delta C = C - C^{*}$, $\delta\tilde{C} = \tilde{C} - \tilde{C}^{*}$; the negative inverse of the Hessian corresponds to their covariance

$$\Delta = -(\mathcal{S}^{(2)})^{-1} =: \begin{pmatrix} \langle \delta C\, \delta C \rangle & \langle \delta C\, \delta\tilde{C} \rangle \\ \langle \delta\tilde{C}\, \delta C \rangle & \langle \delta\tilde{C}\, \delta\tilde{C} \rangle \end{pmatrix}. \tag{4.20}$$

Since we evaluate the Hessian at the saddle point, there is no linear term and all expectation values in the following are with respect to the Gaussian measure $\langle \dots \rangle_p$ in Eq. (4.16).

We calculate the diagonal entries of the Hessian as

$$\begin{aligned}
\frac{\partial^2}{\partial C^{(l)}_{\alpha\beta} \partial C^{(k)}_{\gamma\beta}} \mathcal{S}|_{(C_*,\tilde{C}_*)} &= 0\,, \qquad (4.21)\\
\frac{\partial^2}{\partial \tilde{C}^{(l)}_{\alpha\beta} \partial \tilde{C}^{(k)}_{\gamma\delta}} \mathcal{S}|_{(C_*,\tilde{C}_*)} &= \frac{\partial}{\partial \tilde{C}^{(l)}_{\alpha\beta}} \Big(N[\delta_{k,L+1} + (1-\delta_{k,L+1})\rho^2]\, \sigma_w^2 \langle \phi_\gamma^{(k-1)} \phi_\delta^{(k-1)} \rangle_p + \text{const.}(\tilde{C}) \Big) \\
&= \delta_{L0} N \left[\sigma_{w,\text{in}}^2 \, (XX^{\mathsf{T}})_{\alpha\beta} + D_{\text{in}} \right] \, [\delta_{k,L+1} + (1-\delta_{k,L+1})\rho^2]\, \sigma_w^2 \langle \phi_\gamma^{(k-1)} \phi_\delta^{(k-1)} \rangle_p \\
&\quad + N\sigma_w^4 \, 1_{l>0} 1_{k>0} \, \langle \phi_\alpha^{(l-1)} \phi_\beta^{(l-1)}, \phi_\gamma^{(k-1)} \phi_\delta^{(k-1)} \rangle_p^c \\
&\qquad \times \begin{cases} \rho^4 & k,l \neq L+1, \\ \rho^2 & k \neq l = L+1 \vee l \neq k = L+1, \\ 1 & \text{else,} \end{cases} \qquad (4.22)
\end{aligned}$$

with $1_{l>0}$ being the indicator function. Here $\langle \dots \rangle^c$ denotes connected correlations that are given by

$$\langle \phi_\alpha^{(l-1)} \phi_\beta^{(l-1)}, \phi_\gamma^{(k-1)} \phi_\delta^{(k-1)} \rangle_p^c = \langle \phi_\alpha^{(l-1)} \phi_\beta^{(l-1)} \phi_\gamma^{(k-1)} \phi_\delta^{(k-1)} \rangle_p - \langle \phi_\alpha^{(l-1)} \phi_\beta^{(l-1)} \rangle_p \, \langle \phi_\gamma^{(k-1)} \phi_\delta^{(k-1)} \rangle_p .$$

We calculate the off-diagonal entries of the Hessian as

$$\begin{aligned}
&\frac{\partial^2}{\partial C^{(l)}_{\alpha\beta} \partial \tilde{C}^{(k)}_{\gamma\delta}} \mathcal{S}|_{(C_*,\tilde{C}_*)} \\
&= -N\nu_l \delta_{kl} + N\, 1_{k>0}\, \sigma_w^2 \frac{\partial}{\partial C^{(l)}_{\alpha\beta}} \langle \phi_\gamma^{(k-1)} \phi_\delta^{(k-1)} \rangle_p \times \begin{cases} \rho^2 & k \leq L \\ 1 & k = L+1 \end{cases} \\
&= -N\nu_l \delta_{kl} + N\, 1_{k>0}\, \sigma_w^2 \frac{\partial}{\partial K^{(k-1)}_{\alpha\beta}} \langle \phi_\gamma^{(k-1)} \phi_\delta^{(k-1)} \rangle_{\mathcal{N}(0,K^{(k-1)})} \, \frac{\partial}{\partial C^{(l)}_{\alpha\beta}} K^{(k-1)}_{\alpha\beta} \times \begin{cases} \rho^2 & k \leq L \\ 1 & k = L+1 \end{cases} \\
&= -N\nu_l \delta_{kl} + N\, \delta_{(\alpha\beta),(\gamma\delta)}\, 1_{k>0} 1_{k>l}\, \sigma_w^2 \langle \left[\phi_\alpha^{(k-1)} \right]' \left[\phi_\beta^{(k-1)} \right]' + \delta_{\alpha\beta} \left[\phi_\alpha^{(k-1)} \right]'' \left[\phi_\beta^{(k-1)} \right] \rangle_{\mathcal{N}(0,K^{(k-1)})} \\
&\qquad \times \begin{cases} \rho^2 & k \leq L \\ 1 & k = L+1 \end{cases} , \qquad (4.23)
\end{aligned}$$

using Price's theorem (Price, 1958; Papoulis and Pillai, 2002) from the third to the fourth line. The indicator function $1_{k>l}$ enforces a causality condition $k > l$ across network layers: network kernels $K^{(k-1)}$ only depend on upstream residual kernels $C^{(l)}$ where $l < k$.

We are ultimately interested in the negative inverse of the Hessian. To this end, we write

$$\mathcal{S}^{(2)} = \begin{pmatrix} \frac{\partial^2}{\partial C^2}\mathcal{S} & \frac{\partial^2}{\partial C\,\partial\tilde{C}}\mathcal{S} \\ \frac{\partial^2}{\partial\tilde{C}\,\partial C}\mathcal{S} & \frac{\partial^2}{\partial\tilde{C}^2}\mathcal{S} \end{pmatrix} =: \begin{pmatrix} \mathcal{S}_{11} & \mathcal{S}_{12} \\ \mathcal{S}_{21} & \mathcal{S}_{22} \end{pmatrix}. \tag{4.24}$$

Then we use that $\mathcal{S}_{11} = 0$, yielding the following relation for the sub-blocks of the inverse

$$\Delta_{11} = \Delta_{12}\,\mathcal{S}_{22}\,\Delta_{21}, \tag{4.25}$$

$$\Delta_{12} = -\mathcal{S}_{21}^{-1}, \tag{4.26}$$

$$\Delta_{22} = 0. \tag{4.27}$$

Response function

The causality condition $l < k$ for the kernels $K^{(l)}$ makes the off-diagonal block matrix $\mathcal{S}_{21}$ a lower triangular matrix. Thus, its inverse can be determined using forward propagation

$$\begin{aligned} &\Delta_{12}^{(lm),(\alpha\beta),(\gamma\delta)} \\ &= N^{-1}\nu_l^{-1}\delta_{lm} + 1_{l>0}\,\delta_{(\alpha\beta),(\gamma\delta)}\,\sigma_w^2 \langle \left[\phi_\alpha^{(k-1)}\right]' \left[\phi_\beta^{(k-1)}\right]' + \delta_{\alpha\beta}\left[\phi_\alpha^{(k-1)}\right]'' \left[\phi_\beta^{(k-1)}\right] \rangle_{\mathcal{N}(0,K^{(l-1)})} \\ &\qquad\qquad \times \sum_{k=0}^{l-1} \Delta_{12}^{(km),(\alpha\beta),(\gamma\delta)} \times \begin{cases} \rho^2 & k \leq L \\ 1 & k = L+1 \end{cases}. \end{aligned}$$

We identify $\Delta_{12}^{lm,(\alpha\beta),(\gamma\delta)} = \mathrm{Cov}\left(C_{(\alpha\beta)}^{(l)}, \tilde{C}_{(\gamma\delta)}^{(m)}\right)$ as the forward response function in layer l to a perturbation of the residual in layer m.

We will study the response function with respect to network inputs, as it is related to network trainability. First, the residual response for all hidden layers $1 \leq l \leq L$ is given by

$$\eta_{\alpha\beta}^{(l)} \coloneqq \delta_{(\alpha\beta),(\gamma\delta)}\,\rho^2\sigma_w^2 \langle \left[\phi_\alpha^{(l-1)}\right]' \left[\phi_\beta^{(l-1)}\right]' + \delta_{\alpha\beta}\left[\phi_\alpha^{(l-1)}\right]'' \left[\phi_\beta^{(l-1)}\right] \rangle_{\mathcal{N}(0,K^{(l-1)})} \sum_{k=0}^{l-1} \eta_{\alpha\beta}^{(k)} \tag{4.28}$$

with initial condition $\eta_{\alpha\beta}^{(0)} = D_{\mathrm{in}}^{-1}$. Then, the response function of the kernels $K^{(l)}$ can be determined from their additive relation to the residual kernels $C^{(l)}$, yielding

$\chi^{(l)}_{\alpha\beta} \coloneqq \sum_{k=0}^{l} \eta^{(k)}_{\alpha\beta}$. Finally, we obtain for the output response to perturbations in the input

$$\chi^{\text{out}}_{\alpha\beta} = \sigma^2_{w,\text{out}} \langle \left[\phi^{(L)}_{\alpha}\right]' \left[\phi^{(L)}_{\beta}\right]' + \delta_{\alpha\beta} \left[\phi^{(L)}_{\alpha}\right]'' \left[\phi^{(L)}_{\beta}\right] \rangle_{\mathcal{N}(0,K^{(L)})} \sum_{k=0}^{L} \eta^{(k)}_{\alpha\beta}. \tag{4.29}$$

Kernel fluctuations

The diagonal term of the negative inverse Hessian in Eq. (4.25) corresponds to the covariance of the Gaussian fluctuations of the residual kernels $C^{(l)}$ around the NNGP value $C^{(l)} = C^{(l)}_* + \delta C^{(l)}$ in networks with large but finite network width N. It is given by

$$\Delta^{(lm)}_{11} = \sum_{k,n} \Delta^{(lk)}_{12}\, \mathcal{S}^{(kn)}_{22}\, \Delta^{(nm)}_{21}, \tag{4.30}$$

so that $\delta C^{(l)} \sim \mathcal{N}(0, \Delta^{(ll)}_{11})$. This quantity appears when determining finite-width corrections to quantities such as the posterior kernels, generalization error, etc. We obtain Gaussian fluctuations of the network kernels $K^{(l)}$ from their additive relation to the residual kernels $C^{(l)}$ as $K^{(l)} = \sum_k C^{(k)} = \sum_k C^{(k)}_* + \sum_k \delta C^{(k)}$.

4.5 Experiments

We now use the field-theoretic framework to study signal propagation in residual networks. The response function measures the networks' sensitivity to changes in the input and thus describes the signal propagation in networks. Signal propagation can be linked to network trainability and thus generalization performance of trained networks (Schoenholz et al., 2017; Yang and Schoenholz, 2017). We investigate how the residual scaling affects signal propagation in residual networks and its dependence on the network hyperparameters. We first focus on the diagonal elements of the covariance as it allows us to relate signal propagation to saturation effects from the non-linearity and then extend these results to off-diagonal elements.

4.5.1 Kernels and response function in networks at initialization

Residual kernels $C^{(l)}_*$ and the residual response function $\eta^{(l)}$ for both FFNets and ResNets agree well between theory and empirics in Fig. 4.2. In FFNets, the residual kernels decay to a fixed point so that the residual response function decays exponentially to zero. In contrast, the residual kernels in ResNets converge more slowly and consequently, the residual response function decays more slowly, converging to zero only asympotically (see App. C.2.1 for more details). These results agree with Schoenholz et al. (2017) for FFNets and Yang and Schoenholz (2017) for ResNets, who study the convergence rate of the kernels to their fixed points, whereas we explicitly obtain the response function.

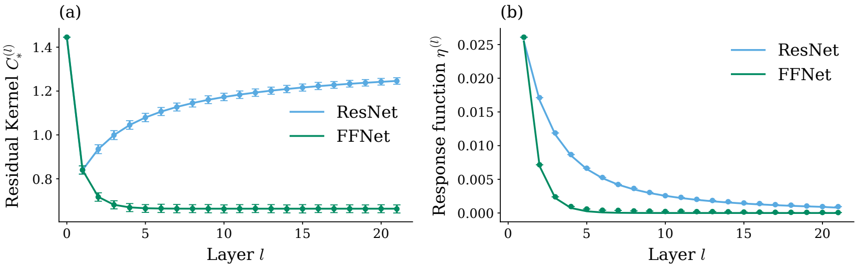


Figure 4.2: Behavior of (a) residual kernels $C_*^{(l)}$ and (b) residual response function $\eta^{(l)}$ in ResNets (blue) compared to FFNets (green). (a) Solid curves show theory values for $C_*^{(l)}$ in Eq. (4.18). Error bars show empirical mean and one standard deviation over 10^3 network realizations. (b) Solid curves show theory values for $\eta^{(l)}$ in Eq. (4.28). Dots show empirical means over 10^2 input samples and 10^3 network initialization; errors are not shown as they are of order 10^{-5}. FFNets converge exponentially to fixed point of kernels across network layers, whereas ResNets converge asymptotically slow. Other parameters: $\sigma^2_{w,\text{in}} = \sigma^2_w = \sigma^2_{w,\text{out}} = 1.2$, $\sigma^2_{b,\text{in}} = \sigma^2_b = \sigma^2_{b,\text{out}} = 0.2$, $D_{\text{in}} = D_{\text{out}} = 100$, $N = 500$, $\rho = 1$.

4.5.2 Optimal scaling in residual networks

To obtain a better intuition for how the residual scaling ρ affects the response function $\chi^{(l)}$, we show the behavior of both kernels $K^{(l)}$ and response function $\chi^{(l)}$ across network layers l for different values of the residual scaling ρ in Fig. 4.3. The residual kernels $C_*^{(l)}$ in Eq. (4.18) scale with ρ^2 and consequently also the kernels $K^{(l)}$, so that ρ governs the rate of increase of $K^{(l)}$ across layers (see Fig. 4.3(a)). The response function $\chi^{(l)}$ inherits the scaling with ρ^2; thus ρ also governs its rate of increase (see Fig. 4.3(b)).

We are ultimately interested in the output response χ^{out} since it measures how sensitive the network output is to changes in the network input, which is linked with network trainability. We find that the output response as a function of the residual scaling ρ has a unique maximum $\rho^*(L)$ as shown in Fig. 4.4(a)-(b), which depends on the network depth L (see Fig. 4.4(c)). In agreement with empirical observations by Szegedy et al. (2017), the optimal values ρ^* are in the value range of $[0.1, 0.3]$ for deep networks.

The recursive structure of Eq. (4.28)-(4.29) does not allow for an analytic solution of the optimal scaling value ρ^*. Instead, we can give an intuition for the diagonal elements of the covariance $K^{(l)}$ based on saturation effects of the non-linearity: The network kernels $K^{(l)}$ in Eq. (4.19) continuously increase across network layers. In consequence, the signal $h^{(l)}$ more likely lies outside the dynamic range $\mathcal{V}$ of the activation function ϕ (see Fig. 4.1(b)) and this part of the signal $h^{(L)}$ gets cut off by

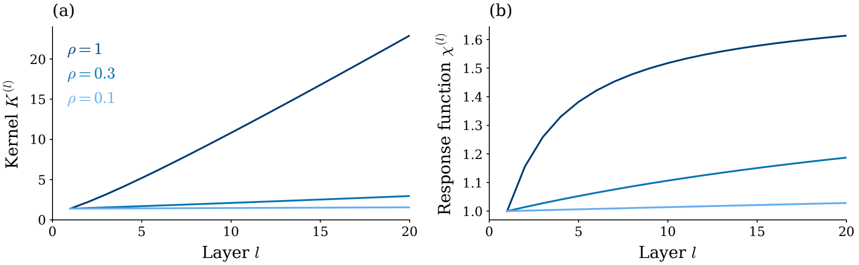


Figure 4.3: The residual scaling parameter ρ determines the rate of increase of both (a) network kernels $K^{(l)}$ and (b) response function $\chi^{(l)}$. We use $\rho \in [1.0, 0.3, 0.1]$ from dark to light. Other parameters: $\sigma^2_{w,\text{in}} = \sigma^2_w = \sigma^2_{w,\text{out}} = 1.2$, $\sigma^2_{b,\text{in}} = \sigma^2_b = \sigma^2_{b,\text{out}} = 0.2$, $D_\text{in} = D_\text{out} = 100$, $N = 500$.

the readout layer, which in turn decreases the output response χ^out. Since the rate of increase for the kernels $K^{(l)}$ scales with ρ^2, one can avoid information loss in the readout layer by decreasing the residual scaling ρ (see Fig. 4.1(c)). However, if ρ becomes too small, it can suppress the residual branch to such a degree that the hidden layers are dominated by the skip connection and the network reduces to a single layer perceptron.

Building on this intuition, we now derive an approximate expression for the optimal scaling ρ^*. Assuming that the signal stays in the linear part of the activation function $\phi(h^{(l)}) \approx \phi'(0)\, h^{(l)}$, the residual kernels can be explicitly written as

$$C_*^{(l)} = \rho^2 \sigma_w^2\, \phi'(0)^2 \sum_{k=0}^{l-1} C_*^{(k)} + \rho^2 \sigma_b^2. \tag{4.31}$$

Here, we used that $\phi(0) = 0$ and $\phi'(0)$ is the slope of the activation function at zero. We get a recursion relation for the residual kernels $C_*^{(l)} = C_*^{(l-1)} + \rho^2\sigma_w^2\,\phi'(0)^2\, C_*^{(l-1)}$ that can be solved as $C_*^{(l)} = (1 + \rho^2\sigma_w^2\,\phi'(0)^2)^{l-1}\,(\rho^2\sigma_w^2\,\phi'(0)^2\, K^{(0)} + \rho^2\sigma_b^2)$. For the network kernels $K^{(l)}$, we then use the geometric series to obtain

$$K^{(\mathrm{L})} = (1 + \rho^2\sigma_w^2\,\phi'(0)^2)^L K^{(0)} + \frac{\sigma_b^2}{\phi'(0)^2\sigma_w^2}\left((1 + \rho^2\sigma_w^2\,\phi'(0)^2)^L - 1\right). \tag{4.32}$$

We require the signal to utilize the full dynamic range $\mathcal{V}$ of the activation function ϕ, yielding the condition $\mathcal{V}/2 \overset{!}{=} \sqrt{K^{(L)}}$. Solving for the optimal residual scaling, we get

$$\rho^* \approx \frac{1}{\sigma_w\,\phi'(0)} \sqrt{\left(\frac{\sigma_w^2\,\phi'(0)^2\,\left(\mathcal{V}/2\right)^2 + \sigma_b^2}{\sigma_w^2\,\phi'(0)^2\, K^{(0)} + \sigma_b^2}\right)^{\frac{1}{L}} - 1}. \tag{4.33}$$

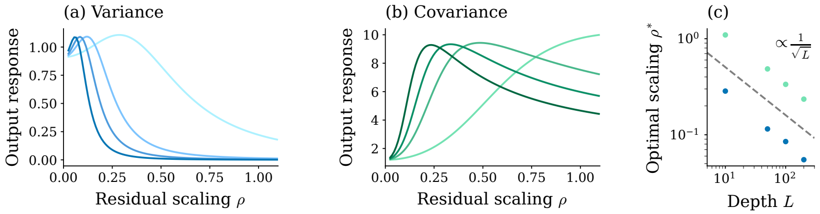


Figure 4.4: The response function indicates optimal scaling of the residual branch. Output response χ^{out} for (a) diagonal and (b) off-diagonal elements of the network kernels $K^{(l)}_{\alpha\beta}$. Light to dark curves represent different depths $L \in [10, 50, 100, 200]$. The unique maximum of the output response indicates optimal scaling ρ^*. (c) Optimal residual scaling $\rho^* = \operatorname{argmax}(\chi^{\text{out}})$ for diagonal (blue) and off-diagonal (green) elements of the network kernel $K^{(l)}_{\alpha\beta}$. Optimal scalings depend on the network depth as $1/\sqrt{L}$ (gray). Other parameters: input kernel $K^{(0)} = \left(\begin{smallmatrix} 0.05 & 0.03 \\ 0.03 & 0.05 \end{smallmatrix}\right)$, $\sigma_w^2 = 1.25$, $\sigma_b^2 = 0.05$, $D_{\text{in}} = D_{\text{out}} = 100$, $N = 500$.

We can alternatively obtain the condition $\mathcal{V}/2 \overset{!}{=} \sqrt{K^{(L)}}$ from a maximum entropy argument for the signal distribution (see App. C.1), where we use the arguments by Bukva et al. (2023) who study trainability of feed-forward networks. Due to the assumptions in Eq. (4.32), the expression in Eq. (4.33) does not capture the effect of the non-linearity fully but allows for an analytically tractable solution. Note that this is exclusively a limitation of Eq. (4.33); the response function accounts for all non-linear effects in the network.

4.5.3 Optimal scaling depends strongly on depth hyperparameter

For deep networks $L \gg 1$, we can expand the L-th root in Eq. (4.33) in $1/L$, yielding to leading order

$$\rho^* \approx \sqrt{\frac{1}{L}} \sqrt{\frac{1}{\sigma_w^2\, \phi'(0)^2} \log\left(\frac{\sigma_w^2\, \phi'(0)^2\, (\mathcal{V}/2)^2 + \sigma_b^2}{\sigma_w^2\, \phi'(0)^2\, K^{(0)} + \sigma_b^2} \right)}. \tag{4.34}$$

Thus, we obtain the $\propto 1/\sqrt{L}$ scaling in Fig. 4.4(c) from the theoretical expression; (Hayou et al., 2021b; Hayou et al., 2021a; Zhang et al., 2022) suggested this scaling using different theoretical approaches. Our result goes beyond these earlier works in that we additonally obtain the dependence on other hyperparameters but the network depth. Due to the appearing logarithm, we find a strong dependence on the network depth relative to a weak dependence on other hyperparameters as shown in Fig. 4.5, explaining the universal success of the $1/\sqrt{L}$ scaling across different architectures (Bordelon et al., 2024).

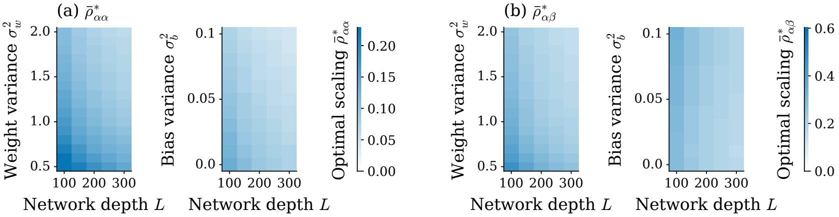


Figure 4.5: Optimal scalings depend weakly on all hyperparameters but the network depth. We show the dependence of both (a) variances and (b) covariances on weight variance σ_w^2 and bias variance σ_b^2 relative to the dependence on network depth L. We use CIFAR-10 with samples being either dogs or airplanes. The scaling with maximal output response is averaged over all diagonal or all off-diagonal elements of the covariance, respectively, $\bar{\rho}^*_{\alpha\alpha} = \frac{1}{N}\sum_\alpha \mathrm{argmax}(\chi^{\mathrm{out}}_{\alpha\alpha})$ or $\bar{\rho}^*_{\alpha\beta} = \frac{1}{N(N-1)}\sum_{\alpha\neq\beta} \mathrm{argmax}(\chi^{\mathrm{out}}_{\alpha\beta})$. Other parameters: data set size $P = 20$, input scale $K^{(0)} = 0.05$, $\sigma_w^2 = 1.25$, $\sigma_b^2 = 0.05$, $D_{\mathrm{in}} = D_{\mathrm{out}} = 100$, $N = 500$.

4.5.4 Optimal scaling across data sets

So far, we have considered the statistics of individual samples in the form of the diagonal elements of the network kernels. However, network trainability generally requires efficient signal propagation on the whole data set. To this end, we first study the dependence of the optimal scaling with maximal output response $\rho^*(K_{\alpha\beta}) = \mathrm{argmax}(\chi^{\mathrm{out}}_{\alpha\beta})$ on different alignment between data samples: we generate samples of unit length with different angles in the intervall $[0, \pi]$ by steps of π/P, where P is the number of data samples. As shown in Fig. 4.6(a), the optimal scaling ρ^* varies strongly as a function of the angle. Nevertheless, when we study common tasks such as MNIST and CIFAR-10 in Fig. 4.6(b)-(c), we find that optimal scalings $\rho^*(K_{\alpha\beta})$ for the off-diagonal elements are very homogeneous across the whole data set, despite samples belonging to two different classes. Therefore, we expect average scalings $\bar{\rho}^*_{\alpha\beta} = \frac{1}{N(N-1)}\sum_{\alpha\neq\beta} \mathrm{argmax}(\chi^{\mathrm{out}}_{\alpha\beta})$ on the off-diagonal elements to be representative for the whole data set.

The optimal scalings for the covariance also exhibit a $1/\sqrt{L}$ scaling as shown in Fig. 4.4(c). While optimal scalings $\bar{\rho}^*_{\alpha\beta}$ depend strongly on the network depth, there is only a weak dependence on other hyperparameters as shown in Fig. 4.5(b). As this behavior is consistent across data sets, it could be an explanation for the universal success of the $1/\sqrt{L}$ scaling across different residual networks (Bordelon et al., 2024).

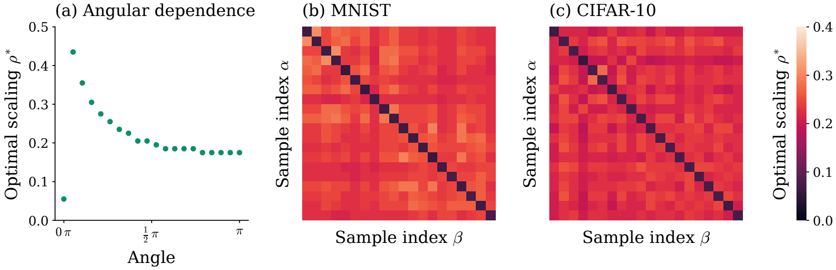


Figure 4.6: Dependence of optimal scaling ρ^* on variation across data sets. (a) Angular dependence of the optimal scaling: Close to parallel samples allow for larger scaling values, while orthogonal samples require smaller scaling values. For data sets such as (b) MNIST and (c) CIFAR-10, optimal scalings ρ^* for off-diagonal elements of the kernels (samples sorted by classes) are similar. For MNIST, samples are either digit 0 or digit 3, and for CIFAR-10, samples are either dogs or airplanes. Other parameters: data set size $P = 20$, input scale $K^{(0)} = 0.05$, $\sigma_w^2 = 1.25$, $\sigma_b^2 = 0.05$, $D_{\text{in}} = D_{\text{out}} = 100$, $N = 500$.

4.6 Conclusion

In this chapter, we derive a field theory for residual networks, which proves to be a versatile method for studying neural networks: In the limit of infinite network width, we recover the NNGP result (Huang et al., 2020; Tirer, Bruna, and Giryes, 2022; Barzilai et al., 2023). More importantly, this approach allows us to determine finite-size properties in a systematic manner.

In this work, we study the response function, which measures signal propagation in the network and thus links to trainability. We focus on the influence of the residual scaling on the response function, discovering that there is a unique maximal response for a particular residual scaling. We link the maximization of the response function to improved signal propagation in the network, which avoids saturation effects due to the non-linear activation function. We obtain the $1/\sqrt{\text{depth}}$ dependence of the optimal scaling as suggest by (Arpit, Campos, and Bengio, 2019; Hayou et al., 2021b; Hayou et al., 2021a; Zhang et al., 2022; Bordelon et al., 2024) and uncover a weak dependence on other hyperparameters of the network. In that sense, the theoretical framework presented in this chapter not only unites multiple existing observations in a joint theory but goes beyond previous works, allowing us to systematically investigate finite-size effects such as feature learning in the future.

4.6.1 Limitations

In this chapter, we consider the network prior, which corresponds to networks at initialization. Nevertheless, (Schoenholz et al., 2017; Yang and Schoenholz, 2017) show that the behavior at initialization can be indicative for generalization performance. More importantly, the network prior is a prerequesite for determining the network posterior and properties of trained networks such as the posterior kernels, as shown in the previous chapter.

The expression for the optimal scaling in Sec. 4.4 is based on the assumption that the signal passes through the linear part of the non-linear activation function for all layers up to the output layer and solely applies to the diagonal elements of the kernels. Despite these simplifications, it correctly predicts the $1/\sqrt{\text{depth}}$ scaling for deep residual networks. Note that the response function itself accounts for non-linear effects due to the activation function and correctly captures the $1/\sqrt{\text{depth}}$ scaling for off-diagonal elements of the network kernels.

4.6.2 Relation to other works

From a practitioner's perspective, the idea of residual scaling was first introduced by Szegedy et al. (2017) to resolve training instabilities in residual networks with a large number of convolutional filters. The common approach of batch normalization (Ioffe and Szegedy, 2015) turned out to be insufficent, but downscaling the residual branch by a factor between 0.1 and 0.3 proved to work well. In a similar vein, Zhang et al. (2019) find a value of 0.1 to result in the best generalization performance by doing a systematic grid search for this hyperparameter.

An adjacent field of research studies scaling the skip connections instead of the residual branch. Zhang et al. (2024) show improved semantic feature learning in autoencoders when scaling the skip connections; they find a dominant dependence on the total downscaling across layers rather than on the particular scaling scheme per layer. Doshi, He, and Gromov (2023) find critical points in residual networks using empirically computed partial Jacobians that show only a weak dependence on hyperparameters. While we here focus on the scaling of the residual branch, it is straightforward to integrate the scaling of the skip connections into our field-theoretic framework and study the trade-off between these two scalings.

There are multiple lines of theoretical research studying the mechanics of residual scaling. Closest in spirit are (Yang and Schoenholz, 2017; Hanin and Rolnick, 2018) who study signal propagation in residual networks in terms of the decay rate of sample correlation to a fixed point, at which all discriminatory information between samples is lost; they show that signal propagation links to trainability of networks. While feed-forward networks exhibit an exponential decay for all points in hyperparameter space except for a low-dimensional critical manifold (Poole et al., 2016; Schoenholz et al., 2017; Hanin, 2018), residual networks exhibit a sub-exponential decay so that they remain close to criticality for all sets of hyperparameters. Our framework is more versatile than their approach in that we recover these results as well as obtain additional finite-size properties of the network such as kernel fluctuations in a systematic way.

Another line of research considers the limit of infinite network depth, for which residual networks can be described by a set of differential equations, so-called NeuralODEs (Chen et al., 2018). According to Marion et al. (2024), residual scaling is linked to the regularity of the network weights, which then affects generalization performance. For different non-linearities and architectures, Cohen et al. (2021) discover different scaling regimes of the network. In contrast to these works, we focus specifically on finite-size effects in residual networks.

Empirical works such as Bachlechner et al. (2021) suggest a strong dependence of proper residual scaling on the network depth, which is also a subject of theoretical considerations. From a neural tangent kernel perspective, Tirer, Bruna, and Giryes (2022) find that smaller residual scalings result in smoother kernels and thus interpolation between data samples. While Huang et al. (2020) argue for a 1/depth scaling, they consider the double limit of infinite width and depth that does not generally apply to finite-size networks. Based on different approaches, (Arpit, Campos, and Bengio, 2019; Hayou et al., 2021b; Hayou et al., 2021a; Zhang et al., 2022; Bordelon et al., 2024) suggest a $1/\sqrt{\text{depth}}$ scaling: Using a mean-field analysis of residual networks, Arpit, Campos, and Bengio (2019) show that this scaling avoids exploding and vanishing information in the network. Similarly, Zhang et al. (2022) observe that this scaling stabilizes the forward and backward propagation of signals. (Hayou et al., 2021b; Hayou et al., 2021a) study the NTK for this case and argue that it becomes universal in the sense that it is capable of expressing any function. Finally, Bordelon et al. (2024) use dynamic mean field theory to argue that the $1/\sqrt{\text{depth}}$ scaling allows hyperparameter transfer across different network widths for residual networks, which was initially done for fully-connected networks in the μP-scaling framework (Yang et al., 2021; Bordelon and Pehlevan, 2023). In contrast to these works, we additionally uncover a weak dependence on all hyperparameters but the network depth, providing a possible expanation for the universal success of the $1/\sqrt{\text{depth}}$ scaling.

Chapter 5

Discussion

The previous three chapters constitute the heart of this thesis and present the key studies of this work. In this final chapter, we briefly review each chapter before putting each one into the context of the overarching topic - mechanistic theories of neural networks beyond the Gaussian limit - as well as providing an outlook per chapter. We conclude by merging the contributions of all chapters to address the relevance of this topic and end on an outlook regarding future avenues of research.

Decomposing neural networks as mappings of correlation functions

In this chapter, we derived a statistical representation of fully-connected networks as a mapping of data correlations across layers. We observed that for the information processing performed by the hidden layers it is sufficient to trace how Gaussian statistics are transformed by the network layers, whereas the input layer additionally extracts information from higher-order cumulants. We showed how this statistical representation can be used to probe which correlations in the data set are essential: For MNIST, class membership is largely encoded in class-conditional means and covariances, while higher-order cumulants are required to fine-tune for few additional performance percentages. For CIFAR-10, we found that more complex data sets require higher-order cumulants to solve the task, providing a possible explanation for the shortcomings of fully-connected networks on such tasks.

While we found that for the hidden network layers it is sufficient to trace only Gaussian statistics, we require corrections from higher-order cumulants in the input layer due to the finite data dimensionality. Further, we expect that for narrow networks, such corrections are also necessary for hidden network layers. For CIFAR-10, we observed that tracing only the behavior of Gaussian statistics, albeit consistent with the behavior of fully-connected networks, is insufficient to solve this task. State-of-the-art architectures on CIFAR-10 typically use convolutional layers that can be understood as sparse, highly-structured, fully-connected layers. Both their structure and sparseness leads to convolutional layers requiring the inclusion of higher-order cumulants in an accurate description of the information processing performed by convolutional networks.

Extending the theoretical framework in this chapter to network architectures such as convolutional or residual networks is an interesting point for future research, allowing us to contrast their computational capabilities with respect to different cumulant orders. Furthermore, we may study expressivity of neural networks with this work: Reversely tracing cumulants through the network allows us to investigate which input distributions can be mapped to particular output distributions, thereby constructing statistical receptive fields of network layers and the network as a whole. Measuring how the complexity of data distributions is reduced by deep neural networks is an important step to a better mechanistic understanding of neural networks.

Critical feature learning in deep neural networks

In this chapter, we derived a set of forward-backward propagation equations that describe the Bayesian posterior kernels in fully-connected networks. We found that they capture non-linear kernel adaptation in trained networks. Further, feature corrections result from fluctuation corrections in a field-theoretic description, indicating a link to criticality. Analyzing the obtained set of equations unveiled a trade-off between criticality and network scales, which drives feature learning in the network.

We here explicitly went beyond the Gaussian limits of the NNGP and NTK for infinitely wide network by calculating fluctuation corrections to the Gaussian kernels for finite-size networks with non-zero training load. While the posterior kernels contained a term that structurally resembles the NNGP result, we explicitly derived the fluctuation corrections to this result. On a conceptual level, the fluctuation corrections directly result from the finite network width: in the infinite width limit, the auxiliary variables concentrate on the NNGP but fluctuate around this value at finite width, allowing for adaption to the training data. Furthermore, another leading-order correction to the NNGP appeared naturally: when disassembling the forward-backward propagation equations, we found that the response function of the network mediates the backward propagation of the kernel mismatch via the conjugate kernels. Thereby, we link the driving mechanisms of feature learning to finite-size corrections beyond the Gaussian limit.

A next step will be to determine the network predictor for the network posterior. To this end, we need to compute corrections to the kernel predictor, in a similar vein as Lindner et al. (2023). Knowledge of the network predictor then will allow us to study the generalization performance of the network posterior.

Field theory for optimal signal propagation in residual networks

In this chapter, we derived a field-theoretic description of residual networks and obtained the response function as the leading-order correction to the NNGP result. The response function is linked to signal propagation in the network and thus network trainability. We investigated how scaling the residual branch affects signal propagation and found that optimal scaling avoids information loss due to signal saturation in the non-linearity. Further, we found a strong dependence of the optimal scaling on the network depth, favoring a $1/\sqrt{\mathrm{depth}}$ scaling, while other hyperparameters only show a weak dependence, explaining the empirically observed universality of the $1/\sqrt{\mathrm{depth}}$ scaling across different networks.

We here studied networks at initialization, which, in the infinite width limit, are governed by a Gaussian limit, the NNGP. In the field-theoretic formulation, the response function appears naturally as the first-order correction to this Gaussian limit at finite width. The definition of the response function directly entails its finite-size nature: it measures the susceptibility of network kernels to changes in the input kernel. In the previous chapter, we have further seen how the response function is linked to feature learning corrections of the NNGP kernel in finite-size networks after training.

With the network prior of residual networks derived in this chapter, we may study the network posterior in a similar vein as Chap. 3 of this thesis. In particular, investigating how skip connections affect the learned features in deep neural networks will be an interesting question for the future.

Statistical field theories of neural networks

In this thesis, we derived statistical field theories of neural networks to study various finite-size properties of these networks. While known theoretical concepts like the NNGP or the NTK correspond to Gaussian descriptions of neural networks, the theoretical frameworks presented in this work allow us to systematically study effects beyond the Gaussian limit, which is the unique benefit of this approach.

Studying finite-size effects in neural networks has emerged as a highly relevant field of research for a theoretical understanding of the mechanics of neural networks. Finite-size effects include phenomena such as grokking (Rubin, Seroussi and Ringel, 2024; Cohen, Levi and Oz, 2024), feature learning (Geiger et al., 2020; Naveh et al., 2021), and non-convex loss landscapes (Geiger et al., 2019; Sarao Mannelli, Vanden-Eijnden and Zdeborová, 2020; Mignacco, Urbani and Zdeborová, 2021). Our contributions in this work are concerned with signal propagation in different network architectures and feature learning in terms of kernel adaptation to data in trained networks.

Outlook

For neural networks, there are two main sources of stochasticity: the data distribution and the distribution of network parameters. In this thesis, we have studied each case separately, considering the data distribution in the first chapter and the distribution of network parameters in the latter two chapters. Understanding interaction effects between these two will be an important next step for future research with highly relevant applications such as transfer learning (Zhuang et al., 2021; Wan et al., 2021; Kora et al., 2022), where networks are trained on a source task with typically large amounts of data available and then are fine-tuned on a target task with limited data availability. Ingrosso et al. (2024) make a first step into this direction and find a link between the source-target correlation and the effectiveness of fine-tuning; however, they consider only kernel-rescaling in their work and do not consider kenel adaptation. Further, Lindner et al. (2023) build a field-theoretic framework for studying how data variability affects kernel regression with the NNGP. Their framework can naturally be integrated with the field-theoretic approaches presented in this work, with the goal of investigating the effect of data variability on the network posterior. In conclusion, this forms a promising avenue for investigating the interplay of learned features with different data sets for, among others, transfer learning.

Appendix

Appendix A

Decomposing neural networks as mappings of correlation functions

A.1 Interaction functions for different activation functions

We define in Sec. 2.4.2 the interaction functions between mean and covariance that result from the application of the non-linearity ϕ; they are given by

$$
\begin{aligned}
f_\mu(\mu_{z^{(l)}}, \Sigma_{z^{(l)}}) &= \langle \phi(z^{(l)}) \rangle_{z^{(l)}}, \\
f_\Sigma(\mu_{z^{(l)}}, \Sigma_{z^{(l)}}) &= \langle \phi(z^{(l)})\, \phi(z^{(l)})^\mathsf{T} \rangle_{z^{(l)}} - \mu_{y^{(l)}} \mu_{y^{(l)}}^\mathsf{T}.
\end{aligned}
$$

In Tab. 2.2 in Sec. 2.4.2 we present expressions for ReLU and quadratic non-linearities. In this appendix we derive these expressions step by step.

A.1.1 ReLU

The ReLU activation function is defined as $\mathrm{ReLU}(z) = \max(0, z)$. We start from the distribution of pre-activations $z^{(l)}$, which we assume to be Gaussian with mean $\mu_{z^{(l)}}$ and covariance $\Sigma_{z^{(l)}}$. Then we compute the mean post-activations as

$$
\begin{aligned}
\mu_{y^{(l)},i} =& \langle \max(0, z_i^{(l)}) \rangle_{z^{(l)} \sim \mathcal{N}(\mu_{z^{(l)}}, \Sigma_{z^{(l)}})} && \text{(A.1)} \\
=& \frac{1}{\sqrt{2\pi \Sigma_{z^{(l)},ii}}} \int_0^\infty \mathrm{d}z_i^{(l)}\, z_i^{(l)} \exp\left(-\frac{\left(z_i^{(l)} - \mu_{z^{(l)},i}\right)^2}{2\Sigma_{z^{(l)},ii}} \right) && \text{(A.2)} \\
=& -\frac{\sqrt{\Sigma_{z^{(l)},ii}}}{\sqrt{2\pi}} \int_{-\mu_{z^{(l)},i}}^\infty \mathrm{d}z_i^{(l)}\, \frac{-z_i^{(l)}}{\Sigma_{z^{(l)},ii}} \exp\left(-\frac{(z_i^{(l)})^2}{2\,\Sigma_{z^{(l)},ii}} \right) && \text{(A.3)} \\
&+ \mu_{z^{(l)},i} \frac{1}{\sqrt{2\pi \Sigma_{z^{(l)},ii}}} \int_{-\mu_{z^{(l)},i}}^\infty \mathrm{d}z_i^{(l)} \exp\left(-\frac{(z_i^{(l)})^2}{2\,\Sigma_{z^{(l)},ii}} \right) && \text{(A.4)} \\
=& \frac{\sqrt{\Sigma_{z^{(l)},ii}}}{\sqrt{2\pi}} \exp\left(-\frac{\mu_{z^{(l)},i}^2}{2\,\Sigma_{z^{(l)},ii}} \right) + \frac{\mu_{z^{(l)},i}}{2} \left(1 + \mathrm{erf}\left(\frac{\mu_{z^{(l)},i}}{\sqrt{2\,\Sigma_{z^{(l)},ii}}} \right) \right). && \text{(A.5)}
\end{aligned}
$$

For the covariance of post-activations, we first calculate the second moment. We here distinguish between diagonal elements with $i = j$ and off-diagonal elements with $i \neq j$. The diagonal elements can be calculated as

$$\begin{aligned}
&\langle \phi(z_i^{(l)})\, \phi(z_i^{(l)}) \rangle_{z^{(l)} \sim \mathcal{N}(\mu_{z^{(l)}}, \Sigma_{z^{(l)}})} && \text{(A.6)}\\
=&\frac{1}{\sqrt{2\pi\, \Sigma_{z^{(l)},ii}}} \int_0^\infty \mathrm{d}z_i^{(l)}\, (z_i^{(l)})^2 \exp\left(-\frac{1}{2\Sigma_{z^{(l)},ii}} (z_i^{(l)} - \mu_{z^{(l)},i})^2 \right) && \text{(A.7)}\\
=&-\frac{\sqrt{\Sigma_{z^{(l)},ii}}}{\sqrt{2\pi}}\, \mu_{z^{(l)},i} \exp\left(-\frac{\mu_{z^{(l)},i}^2}{2\Sigma_{z^{(l)},ii}} \right) + \frac{\Sigma_{z^{(l)},ii}}{2} \left(1 + \mathrm{erf}\left(\frac{\mu_{z^{(l)},i}}{\sqrt{2\, \Sigma_{z^{(l)},ii}}} \right) \right) \\
&+ \sqrt{\frac{2}{\pi}}\, \mu_{z^{(l)},i} \sqrt{\Sigma_{z^{(l)},ii}} \exp\left(-\frac{\mu_{z^{(l)},i}^2}{2\Sigma_{z^{(l)},ii}} \right) + \frac{\mu_{z^{(l)},i}^2}{2} \left(1 + \mathrm{erf}\left(\frac{\mu_{z^{(l)},i}}{\sqrt{2\, \Sigma_{z^{(l)},ii}}} \right) \right) && \text{(A.8)}\\
=&\frac{\sqrt{\Sigma_{z^{(l)},ii}}\, \mu_{z^{(l)},i}}{\sqrt{2\pi}} \exp\left(-\frac{\mu_{z^{(l)},i}^2}{2\Sigma_{z^{(l)},ii}} \right) + \frac{\Sigma_{z^{(l)},ii} + \mu_{z^{(l)},i}^2}{2} \left(1 + \mathrm{erf}\left(\frac{\mu_{z^{(l)},i}}{\sqrt{2\, \Sigma_{z^{(l)},ii}}} \right) \right). && \text{(A.9)}
\end{aligned}$$

Using the relation between second moment, mean, and covariance, we get the diagonal elements of the covariance

$$\begin{aligned}
\Sigma_{y^{(l)},ii} =& \langle \phi(z_i^{(l)})\, \phi(z_i^{(l)}) \rangle_{z^{(l)} \sim \mathcal{N}(\mu_{z^{(l)}}, \Sigma_{z^{(l)}})} - \left(\mu_{y^{(l)},i} \right)^2 && \text{(A.10)}\\
=& \frac{\Sigma_{z^{(l)},ii}}{2} \left(1 + \mathrm{erf}\left(\frac{\mu_{z^{(l)},i}}{\sqrt{2\Sigma_{z^{(l)},ii}}} \right) \right) + \frac{\mu_{z^{(l)},i}^2}{4} \\
&- \left(\frac{\sqrt{\Sigma_{z^{(l)},ii}}}{\sqrt{2\pi}} \exp\left(-\frac{1}{2\Sigma_{z^{(l)},ii}} \mu_{z^{(l)},i}^2 \right) + \frac{\mu_{z^{(l)},i}}{2} \mathrm{erf}\left(\frac{\mu_{z^{(l)},i}}{\sqrt{2\Sigma_{z^{(l)},ii}}} \right) \right)^2. && \text{(A.11)}
\end{aligned}$$

For the off-diagonal elements $i \neq j$, we marginalize over all other indices and just consider the joint distribution of $(z_i^{(l)}, z_j^{(l)})$. We write the marginalized mean and covariance as $\tilde{\mu}_{z^{(l)}} = (\mu_{z^{(l)},i}, \mu_{z^{(l)},j})^\mathsf{T}$ and $\tilde{\Sigma}_{z^{(l)}} = \begin{pmatrix} \Sigma_{z^{(l)},ii} & \Sigma_{z^{(l)},ij} \\ \Sigma_{z^{(l)},ji} & \Sigma_{z^{(l)},jj} \end{pmatrix}$. We get for the second moment

$$\begin{aligned}
&\langle \phi(z_i^{(l)})\, \phi(z_j^{(l)}) \rangle_{z^{(l)} \sim \mathcal{N}(\mu_{z^{(l)}}, \Sigma_{z^{(l)}})} \\
=&\frac{1}{\sqrt{(2\pi)^2 \det(\tilde{\Sigma}_{z^{(l)}})}} \int_0^\infty \mathrm{d}z_i^{(l)} \int_0^\infty \mathrm{d}z_j^{(l)}\, z_i^{(l)}\, z_j^{(l)} \exp\left(-\frac{1}{2} (\tilde{z}^{(l)} - \tilde{\mu}_{z^{(l)}})^\mathsf{T} \tilde{\Sigma}_{z^{(l)}}^{-1} (\tilde{z}^{(l)} - \tilde{\mu}_{z^{(l)}}) \right) && \text{(A.12)}\\
=&\frac{\sqrt{\det(\tilde{\Sigma}_{z^{(l)}})}}{2\pi} \exp\left(-\frac{\tilde{\mu}_{z^{(l)}}^\mathsf{T} \tilde{\Sigma}_{z^{(l)}}^{-1} \tilde{\mu}_{z^{(l)}}}{2} \right) \\
&+ \frac{\sqrt{\det(\tilde{\Sigma}_{z^{(l)}})}}{2\pi} \tilde{\Sigma}_{z^{(l)},jj}^{-1}\, \mu_{z^{(l)},j} \frac{\sqrt{\pi}}{\sqrt{2\tilde{\Sigma}_{z^{(l)},jj}^{-1}}} \exp\left(-\frac{\tilde{\Sigma}_{z^{(l)},ii}^{-1}\, \mu_{z^{(l)},i}^2}{2} \right)
\end{aligned}$$

$$
\begin{aligned}
&\times \exp\left(\frac{\left(\tilde{\Sigma}^{-1}_{z^{(l)},ji}\,\mu_{z^{(l)},i}\right)^2}{2\tilde{\Sigma}^{-1}_{z^{(l)},jj}}\right)\left[1+\operatorname{erf}\left(\frac{\left(\tilde{\Sigma}^{-1}_{z^{(l)}}\tilde{\mu}_{z^{(l)}}\right)_j}{\sqrt{2\tilde{\Sigma}^{-1}_{z^{(l)},jj}}}\right)\right]\\
&+\frac{\sqrt{\det(\tilde{\Sigma}_{z^{(l)}})}}{2\pi}\,\tilde{\Sigma}^{-1}_{z^{(l)},ii}\,\mu_{z^{(l)},i}\,\frac{\sqrt{\pi}}{\sqrt{2\tilde{\Sigma}^{-1}_{z^{(l)},ii}}}\,\exp\left(-\frac{\tilde{\Sigma}^{-1}_{z^{(l)},jj}\,\mu^2_{z^{(l)},j}}{2}\right)\\
&\times \exp\left(\frac{\left(\tilde{\Sigma}^{-1}_{z^{(l)},ij}\,\mu_{z^{(l)},j}\right)^2}{2\tilde{\Sigma}^{-1}_{z^{(l)},ii}}\right)\left[1+\operatorname{erf}\left(\frac{(\tilde{\Sigma}^{-1}_{z^{(l)}}\tilde{\mu}_{z^{(l)}})_i}{\sqrt{2\tilde{\Sigma}^{-1}_{z^{(l)},ii}}}\right)\right]\\
&+\left(\mu_{z^{(l)},i}\,\mu_{z^{(l)},j}-\tilde{\Sigma}^{-1}_{z^{(l)},ij}\,\det\left(\tilde{\Sigma}_{z^{(l)}}\right)\right)\left[\frac{1}{2}\operatorname{erf}\left(\frac{\sqrt{2}\,\mu_{z^{(l)},i}}{\sqrt{\Sigma_{z^{(l)},ii}}}\right)+\frac{1}{2}\operatorname{erf}\left(\frac{\sqrt{2}\,\mu_{z^{(l)},j}}{\sqrt{\Sigma_{z^{(l)},jj}}}\right)+F_{\tilde{\mu}_{z^{(l)}},\tilde{\Sigma}_{z^{(l)}}}(0,0)\right].
\end{aligned}
\tag{A.13}
$$

We write the appearing cumulative distribution function as $F_{\tilde{\mu}_{z^{(l)}},\tilde{\Sigma}_{z^{(l)}}}(x,y)$; evaluating at the origin $F_{\tilde{\mu}_{z^{(l)}},\tilde{\Sigma}_{z^{(l)}}}(0,0)$ yields the so-called quadrant probability. To obtain the off-diagonal elements of the covariance, we subtract $\mu_{y^{(l)},i}\,\mu_{y^{(l)},j}$ from the second moment, yielding the expression in Tab. 2.2 in Sec. 2.4.2.

Corrections from higher-order cumulants using a Gram-Charlier expansion

For determining correction terms from higher-order cumulants, we use the Gram-Charlier expansion (Blinnikov and Moessner, 1998) of the probability density function $p_{z_i^{(l)}}(z_i^{(l)})$. This expansion assumes the probability density function to be close to a Gaussian, performing a Taylor expansion in higher-order cumulants. Up to third order, the Gram-Charlier expansion is given by

$$
p_{z_i^{(l)}}(z_i^{(l)}) \approx \left(1+\frac{G^{(3)}_{z^{(l)},(i,i,i)}}{3!\sqrt{\Sigma^3_{z^{(l)},ii}}}\left[\left(\frac{z_i^{(l)}-\mu_{z^{(l)},i}}{\sqrt{\Sigma_{z^{(l)},ii}}}\right)^3-3\frac{\left(z_i^{(l)}-\mu_{z^{(l)},i}\right)}{\sqrt{\Sigma_{z^{(l)},ii}}}\right]\right)\frac{1}{\sqrt{2\pi\,\Sigma_{z^{(l)},ii}}}\exp\left(-\frac{\left(z_i^{(l)}-\mu_{z^{(l)},i}\right)^2}{2\Sigma_{z^{(l)},ii}}\right).
$$

Based on this, we now calculate corrections to the mean of the post-activations $y^{(l)}$ up to linear order in $G^{(3)}_{z^{(l)}}$ for ReLU activations.

Using the Gram-Charlier expansion, we write the mean $\mu_{y^{(l)},i}$ of the post-activations $y^{(l)}$ as

$$
\begin{aligned}
\mu_{y^{(l)},i} &= \langle \phi(z_i^{(l)}) \rangle_{z^{(l)}\sim\mathcal{N}(\mu_{z^{(l)}},\Sigma_{z^{(l)}})} && \text{(A.14)}\\
&= \int_0^\infty \mathrm{d}z_i^{(l)}\, z_i^{(l)}\, p_{z_i^{(l)}}(z_i^{(l)}) && \text{(A.15)}\\
&\approx \int_0^\infty \mathrm{d}z_i^{(l)}\, z_i^{(l)}\, \frac{1}{\sqrt{2\pi\,\Sigma_{z^{(l)},ii}}}\exp\left(-\frac{\left(z_i^{(l)}-\mu_{z^{(l)},i}\right)^2}{2\Sigma_{z^{(l)},ii}}\right)\\
&\quad+\frac{G^{(3)}_{z^{(l)},(i,i,i)}}{3!\sqrt{\Sigma^3_{z^{(l)},ii}}}\int_0^\infty \mathrm{d}z_i^{(l)}\, z_i^{(l)}\left[\left(\frac{z_i^{(l)}-\mu_{z^{(l)},i}}{\sqrt{\Sigma_{z^{(l)},ii}}}\right)^3-3\frac{\left(z_i^{(l)}-\mu_{z^{(l)},i}\right)}{\sqrt{\Sigma_{z^{(l)},ii}}}\right] && \text{(A.16)}
\end{aligned}
$$

$$\times \frac{\exp\Big(-\big(z_i^{(l)} - \mu_{z^{(l)},i}\big)^2 / 2\Sigma_{z^{(l)},ii}\Big)}{\sqrt{2\pi\,\Sigma_{z^{(l)},ii}}} \tag{A.17}$$

$$\begin{aligned}
=& \frac{\sqrt{\Sigma_{z^{(l)},ii}}\,\mu_{z^{(l)},i}}{\sqrt{2\pi}}\,\exp\Big(-\frac{\mu^2_{z^{(l)},i}}{2\,\Sigma_{z^{(l)},ii}}\Big) + \frac{\mu_{z^{(l)},i}}{2}\left(1+\mathrm{erf}\left(\frac{\mu_{z^{(l)},i}}{\sqrt{2\,\Sigma_{z^{(l)},ii}}}\right)\right)\\
&- \frac{G^{(3)}_{z^{(l)},(i,i,i)}}{2\,\Sigma^2_{z^{(l)},ii}}\left(\Sigma^2_{z^{(l)},ii} - 1\right)\frac{1}{2}\left(1+\mathrm{erf}\left(\frac{\mu_{z^{(l)},i}}{\sqrt{2\,\Sigma_{z^{(l)},ii}}}\right)\right)\\
&+ \frac{G^{(3)}_{z^{(l)},(i,i,i)}}{3!\,\Sigma^3_{z^{(l)},ii}}\left(3\,\mu_{z^{(l)},i}\Sigma^2_{z^{(l)},ii} + 2\,\Sigma_{z^{(l)},ii} + \mu^3_{z^{(l)},i} + \mu^2_{z^{(l)},i} - 3\right) && \text{(A.18)}\\
&\times \frac{\sqrt{\Sigma_{z^{(l)},ii}}\,\mu_{z^{(l)},i}}{\sqrt{2\pi}}\,\exp\left(-\frac{\mu^2_{z^{(l)},i}}{2\Sigma_{z^{(l)},ii}}\right). && \text{(A.19)}
\end{aligned}$$

Corrections by other cumulant orders can be determined in a similar way.

A.1.2 Quadratic activation function

We here study the case of a quadratic activation function $\phi(z) = z + \epsilon\, z^2$. The mean post-activations are given by

$$\begin{aligned}
\mu_{y^{(l)},i} &= \langle z_i^{(l)} + \epsilon\,(z_i^{(l)})^2\rangle_{z^{(l)}} && \text{(A.20)}\\
&= \mu_{z^{(l)},i} + \epsilon\,(\mu_{z^{(l)},i})^2 + \epsilon\,\Sigma_{z^{(l)},ii}. && \text{(A.21)}
\end{aligned}$$

This holds for any distribution of pre-activations $z^{(l)}$, not only a Gaussian.

The second moment is given by

$$\begin{aligned}
\langle \phi(z_i^{(l)})\,\phi(z_j^{(l)})\rangle_{z^{(l)}} &= \left\langle \left[z_i^{(l)} + \epsilon\,(z_i^{(l)})^2\right]\left[z_j^{(l)} + \epsilon\,(z_j^{(l)})^2\right]\right\rangle_{z^l} && \text{(A.22)}\\
&= \Sigma_{z^{(l)},ij} + \mu_{z^{(l)},i}\,\mu_{z^{(l)},j} + \epsilon\,M^{(3)}_{z^{(l)},(i,j,j)} + \epsilon\,M^{(3)}_{z^{(l)},(j,i,i)} + \epsilon^2\,M^{(4)}_{z^{(l)},(i,i,j,j)}, && \text{(A.23)}
\end{aligned}$$

where $M^{(n)}_{z^{(l)}}$ stands for the n-th moment of pre-activations $z^{(l)}$. Using the relation between second moment, mean, and covariance, we get for the covariance

$$\begin{aligned}
\Sigma_{y^{(l)},ij} =& \langle \phi(z_i^{(l)})\,\phi(z_j^{(l)})\rangle_{z^{(l)}} - \langle \phi(z_i^{(l)})\rangle_{z^{(l)}}\,\langle \phi(z_j^{(l)})\rangle_{z^{(l)}} && \text{(A.24)}\\
=& \Sigma_{z^{(l)},ij} + 2\,\epsilon\,\Sigma_{z^{(l)},ij}\left(\mu_{z^{(l)},i} + \mu_{z^{(l)},j}\right) + 2\,\epsilon^2\,(\Sigma_{z^{(l)},ij})^2\\
&+ 4\,\epsilon^2\,\mu_{z^{(l)},i}\,\Sigma_{z^{(l)},ij}\,\mu_{z^{(l)},j} + \Sigma_{y^{(l)},ij}|_{n>2}. && \text{(A.25)}
\end{aligned}$$

We collect all contributions from higher-order cumulants in $\Sigma_{y^l,ij}|_{n>2}$, which is given by

$$\Sigma_{y^{(l)},ij}\big|_{n>2} = \epsilon\,(1 + 2\epsilon\,\mu_{z^{(l)},i})\,G^{(3)}_{z^{(l)},(i,j,j)} + \epsilon\,(1 + 2\alpha\,\mu_{z^{(l)},j})\,G^{(3)}_{z^{(l)},(j,i,i)} + \epsilon^2\,G^{(4)}_{z^{(l)},(i,i,j,j)}. \quad \text{(A.26)}$$

For Gaussian distributed pre-activations $z^{(l)} \sim \mathcal{N}(\mu_{z^{(l)}}, \Sigma_{z^{(l)}})$, higher-order cumulants vanish $G^{(n>2)}_{z^{(l)}} = 0$, so that $\Sigma_{y^{(l)},ij}\big|_{n>2} = 0$ and we obtain the expression in Tab. 2.2 in Sec. 2.4.2.

A.2 Higher-order cumulants of post-activations from weak correlations

In this section, we study the influence of weak correlations between pre-activations on higher-order cumulants of the post-activations. The activation function ϕ is assumed to be piece-wise differentiable. In the following, all functions applied to pre-activations are the activation function ϕ, but we denote them differently to keep track of different indices, so $f = f_k = g = \phi$. We start by considering only two pre-activations x, y that are Gaussian distributed with zero mean and weakly correlated with covariance $C = \left(\begin{smallmatrix} a & c \\ c & a \end{smallmatrix}\right)$. To keep the notation concise, we use $\langle \dots \rangle \coloneqq \langle \dots \rangle_{(x,y)\sim\mathcal{N}(0,C)}$. We look at the second moment $\langle f(x)g(y)\rangle$ of the post-activations and expand for small c as

$$\begin{aligned}\langle f(x)g(y)\rangle &= \langle f(x)g(y)\rangle_{c=0} + \langle f'(x)g'(y)\rangle_{c=0}\, c + \mathcal{O}(c^2)\\ &= \langle f(x)\rangle\langle g(y)\rangle + \langle f'(x)\rangle\langle g'(y)\rangle\, c + \mathcal{O}(c^2),\end{aligned} \quad \text{(A.27)}$$

where we used Price's theorem (Price, 1958; Papoulis and Pillai, 2002; Schuecker et al., 2016, Appendix A) to determine the first Taylor coefficient

$$\frac{\partial}{\partial c}\langle f(x)g(y)\rangle = \langle f'(x)g'(y)\rangle.$$

The expansion Eq. (A.27) corresponds to Eq. (A4) in (Goldt et al., 2020), but is there derived with a different method than Price's theorem. One needs to substitue $\langle u f(u)\rangle = \langle f'(u)\rangle$ in Eq. (A4) in (Goldt et al., 2020), which is possible since $\langle u^2\rangle = 1$ is assumed.

In order to generalize to higher-order cumulants, we use centered post-activations

$$\begin{aligned}\tilde{f}(x) &\coloneqq f(x) - \langle f(x)\rangle,\\ \tilde{g}(x) &\coloneqq g(x) - \langle g(x)\rangle,\end{aligned}$$

yielding an expansion of the second moment and its dependence on the weak correlation c between pre-activations as

$$\langle \tilde{f}(x)\tilde{g}(y)\rangle = \langle f'(x)\rangle\langle g'(y)\rangle\, c + \mathcal{O}(c^2).$$

We now study how expectation values of arbitrary orders depend on the weak correlation c between pre-activations; we denote these expectations as

$$F_n(x) := \Big\langle \prod_{k=1}^{n} \tilde{f}_k(x_k) \Big\rangle. \tag{A.28}$$

To evaluate the appearing expectation values, we use the marginalization property of Gaussian distributions: We assume that $x = (x_i)_i$ is Gaussian distributed with covariance C. Then the joint distribution of any subset of x_i is also Gaussian distributed; the covariance matrix is the corresponding submatrix $C_{ij} = \langle\langle x_i x_j \rangle\rangle$. Building on the result for two variables, we define for any index pair (i, j)

$$\begin{aligned} F_n(x\backslash\{x_i, x_j\}) &= \Big\langle \prod_{k=1}^{n} \tilde{f}_k(x_k) \Big\rangle_{(x_i, x_j)} \\ &= \langle \tilde{f}_i(x_i) \tilde{f}_j(x_j) \rangle_{(x_i, x_j)} \prod_{k\backslash\{i,j\}} \tilde{f}_k(x_k). \end{aligned}$$

We can apply the previous result Eq. (A.27) to the first term and get

$$F_n(x\backslash\{x_i, x_j\}) = \big[c_{ij} \, \langle f_i'(x_i) f_j'(x_j) \rangle_{(x_i, x_j), c_{ij}=0} + \mathcal{O}(c_{ij}^2) \big] \prod_{k\backslash\{i,j\}} \tilde{f}_k(x_k). \tag{A.29}$$

Then we determine the expectation across the remaining variables $x\backslash\{x_i, x_j\}$ with probability distribution $p(x\backslash\{x_i, x_j\})$. We rewrite the probability distribution over all variables in terms of conditionals $p(x_1, \dots, x_N) = p(x_i, x_j | x\backslash\{x_i, x_j\}) \, p(x\backslash\{x_i, x_j\})$ and use Eq. (A.29) for the conditional expectation value over x_i, x_j with regard to $p(x_i, x_j | x\backslash\{x_i, x_j\})$, yielding

$$\big\langle F_n(x\backslash\{x_i, x_j\}) \big\rangle_{x\backslash\{x_i, x_j\}} = \Big\langle \big[c_{ij} \, f_i'(x_i) f_j'(x_j) + \mathcal{O}(c_{ij}^2) \big] \prod_{k\backslash\{i,j\}} \tilde{f}_k(x_k) \Big\rangle_{x, c_{ij}=0}.$$

For the expectation over all variables, we look at arbitrary pairings of all indices as any such pair yields a non-zero contribution. Expanding all pairs in a similar manner as before, we obtain

$$\begin{aligned} \big\langle F_n(x) \big\rangle_x = \sum_{\sigma\in\Pi} & c_{\sigma(1)\sigma(2)} \, \langle f_{\sigma(1)}'(x_{\sigma(1)}) \rangle \langle f_{\sigma(2)}'(x_{\sigma(2)}) \rangle \\ & \cdots c_{\sigma(n-1)\sigma(n)} \, \langle f_{\sigma(n-1)}'(x_{\sigma(n-1)}) \rangle \langle f_{\sigma(n)}'(x_{\sigma(n)}) \rangle \\ + \mathcal{O}(c_{\circ\circ}^{\frac{n}{2}+1}), & \end{aligned} \tag{A.30}$$

where $\sum_{\sigma\in\Pi}$ sums over all disjoint pairings of indices. This expression matches Eq. (A16) in (Goldt et al., 2020) apart from minor typos; the factors b_i seem to be missing, p needs to be m, and we interpret the upper case of their Eq. (A16) as $b_1 \cdots b_m \sum_{\sigma\pi\Pi} m_{\sigma_1\sigma_2} \cdots m_{\sigma_{m-1}\sigma_m}$.

The expansion Eq. (A.30) applies to arbitrary cumulant orders n and is correct up to terms of order $\mathcal{O}(c^{\frac{n}{2}})$. In consequence, all cumulants beyond Gaussian order $n > 2$ vanish.

We illustrate the latter for the fourth-order cumulant, where we drop the argument x for brevity:

$$\langle\langle \tilde{f}_1 \tilde{f}_2 \tilde{f}_3 \tilde{f}_4 \rangle\rangle = \langle \tilde{f}_1 \tilde{f}_2 \tilde{f}_3 \tilde{f}_4 \rangle - \langle \tilde{f}_1 \tilde{f}_2 \rangle \langle \tilde{f}_3 \tilde{f}_4 \rangle - \langle \tilde{f}_1 \tilde{f}_3 \rangle \langle \tilde{f}_2 \tilde{f}_4 \rangle - \langle \tilde{f}_1 \tilde{f}_4 \rangle \langle \tilde{f}_2 \tilde{f}_3 \rangle. \tag{A.31}$$

Using Eq. (A.30), the first term is given by

$$\begin{aligned}\langle \tilde{f}_1 \tilde{f}_2 \tilde{f}_3 \tilde{f}_4 \rangle = c_{12} \langle f_1' \rangle \langle f_2' \rangle c_{34} \langle f_3' \rangle \langle f_4' \rangle + c_{13} \langle f_1' \rangle \langle f_3' \rangle c_{24} \langle f_2' \rangle \langle f_4' \rangle \\ + c_{14} \langle f_1' \rangle \langle f_4' \rangle c_{23} \langle f_2' \rangle \langle f_3' \rangle + \mathcal{O}(c^3).\end{aligned}$$

Further, we expand all negative terms of Eq. (A.31) using Eq. (A.30), yielding e.g. for the first term

$$-\langle \tilde{f}_1 \tilde{f}_2 \rangle \langle \tilde{f}_3 \tilde{f}_4 \rangle = -c_{12} \langle f_1' \rangle \langle f_2' \rangle c_{34} \langle f_3' \rangle \langle f_4' \rangle + \mathcal{O}(c^3),$$

precisely cancelling all terms in Eq. (A.31) up to order $\mathcal{O}(c^3)$.

For arbitrary cumulant orders n, odd orders vanish for centered variables and even orders exhibit similar cancellation effects so that

$$\left\langle\!\!\left\langle \prod_{k=1}^{n} \tilde{f}_k(x_k) \right\rangle\!\!\right\rangle = \mathcal{O}\big(c^{\frac{n}{2}+1}\big). \tag{A.32}$$

Up to here, we only considered the case that all indices differ from one another. In general, we also need to look at the case that indices in Eq. (A.31) are repeated, e.g. $\langle\langle \tilde{f}_1 \tilde{f}_1 \tilde{f}_2 \tilde{f}_2 \rangle\rangle$. To formalize this case, we assume that for a cumulant of order n there are r different indices $j_1, \ldots, j_r$ with $n > r$. Within a set of repeated variables correlations are of order $\mathcal{O}(1)$ instead of $\mathcal{O}(c)$. In the expansion Eq. (A.30) variables with repeated indices must be treated as a single variable.

As an example for r even, we look at $\langle\langle \tilde{f}_1 \tilde{f}_1 \tilde{f}_2 \tilde{f}_2 \rangle\rangle$. We define $g_i := \tilde{f}_i^2$, $i = 1, 2$ and centered variables $\tilde{g}_i := \tilde{f}_i^2 - \langle \tilde{f}_i^2 \rangle$. Using Eq. (A.30), we then get

$$\langle \tilde{g}_1 \tilde{g}_2 \rangle = c_{12} \langle g_1' \rangle \langle g_2' \rangle + \mathcal{O}(c^2), \tag{A.33}$$

which is the same as the second cumulant since $\tilde{g}$ is centered. We now expand the fourth cumulant analogous to Eq. (A.31), yielding

$$\langle\langle \tilde{f}_1 \tilde{f}_1 \tilde{f}_2 \tilde{f}_2 \rangle\rangle = \langle \tilde{f}_1 \tilde{f}_1 \tilde{f}_2 \tilde{f}_2 \rangle - \langle \tilde{f}_1 \tilde{f}_1 \rangle \langle \tilde{f}_2 \tilde{f}_2 \rangle - \langle \tilde{f}_1 \tilde{f}_2 \rangle \langle \tilde{f}_1 \tilde{f}_2 \rangle - \langle \tilde{f}_1 \tilde{f}_2 \rangle \langle \tilde{f}_1 \tilde{f}_2 \rangle. \tag{A.34}$$

Using the definitions of g and $\tilde{g}$, and Eq. (A.33), we expand the fourth moment as

$$\begin{aligned}\langle\tilde{f}_1\tilde{f}_1\tilde{f}_2\tilde{f}_2\rangle &= \langle g_1 g_2\rangle\\ &= \langle\tilde{g}_1\tilde{g}_2\rangle + \langle g_1\rangle\langle g_2\rangle\\ &\overset{(A.33)}{=} c_{12}\,\langle g_1'\rangle\,\langle g_2'\rangle + \langle g_1\rangle\langle g_2\rangle + \mathcal{O}(c^2).\end{aligned}$$

Dropping all terms of order $\mathcal{O}(c^2)$ such as $\langle\tilde{f}_1\tilde{f}_2\rangle\,\langle\tilde{f}_1\tilde{f}_2\rangle = \mathcal{O}(c_{12}^2)$ in Eq. (A.34) and using $\langle\tilde{f}_1\tilde{f}_1\rangle\,\langle\tilde{f}_2\tilde{f}_2\rangle = \langle g_1\rangle\langle g_2\rangle$, we get

$$\langle\!\langle\tilde{f}_1\tilde{f}_1\tilde{f}_2\tilde{f}_2\rangle\!\rangle = c_{12}\,\langle g_1'\rangle\,\langle g_2'\rangle + \mathcal{O}(c^2).$$

For r odd, we look at the example of

$$\langle\!\langle\tilde{f}_1\tilde{f}_2\tilde{f}_2\tilde{f}_3\rangle\!\rangle = \langle\tilde{f}_1\tilde{f}_2\tilde{f}_2\tilde{f}_3\rangle - \langle\tilde{f}_1\tilde{f}_2\rangle\,\langle\tilde{f}_2\tilde{f}_3\rangle - \langle\tilde{f}_1\tilde{f}_2\rangle\,\langle\tilde{f}_2\tilde{f}_3\rangle - \langle\tilde{f}_1\tilde{f}_3\rangle\,\langle\tilde{f}_2\tilde{f}_2\rangle. \tag{A.35}$$

Then the fourth moment is given by

$$\begin{aligned}\langle\tilde{f}_1\tilde{f}_2\tilde{f}_2\tilde{f}_3\rangle &= \langle\tilde{f}_1 g_2\tilde{f}_3\rangle\\ &= \underbrace{\langle\tilde{f}_1\tilde{g}_2\tilde{f}_3\rangle}_{\mathcal{O}(c^2)} + \langle g_2\rangle\,\langle\tilde{f}_1\tilde{f}_3\rangle\\ &\overset{(A.33)}{=} \langle g_2\rangle\,\langle f_1'\rangle\langle f_3'\rangle\,c_{13} + \mathcal{O}(c^2).\end{aligned}$$

Inserting into Eq. (A.35), we get

$$\langle\!\langle\tilde{f}_1\tilde{f}_2\tilde{f}_2\tilde{f}_3\rangle\!\rangle = \langle g_2\rangle\,\langle f_1'\rangle\langle f_3'\rangle\,c_{13} - \langle\tilde{f}_1\tilde{f}_3\rangle\,\langle\tilde{f}_2\tilde{f}_2\rangle + \mathcal{O}(c^2) = \mathcal{O}(c^2),$$

where all terms $\propto c$ cancel precisely.

For the case of repeated indices, these two examples illustrate the structure of the expansion in weak correlations: For r different indices $j_1,\ldots,j_r$, the n-th cumulant with $n > r$ will be of the order in c that equals the number of pairs to join all different indices. Altogether, we get for arbitrary r that the n-th cumulant of post-activations scales with weak correlations of the pre-activations as

$$\left\langle\!\!\left\langle\prod_{k=1}^{n}\tilde{f}_{j_k}(x_{j_k})\right\rangle\!\!\right\rangle = \mathcal{O}\big(c^{\lceil\frac{r}{2}\rceil}\big). \tag{A.36}$$

A.3 Suppression of higher-order cumulants in wide networks

Even when the pre-activations $z^{(l)}$ are Gaussian distributed, the non-linear activation function ϕ leads to post-activations $y^{(l)}$ that are non-Gaussian distributed as the non-linearity generates higher-order cumulants $G^{(n)}_{y^{(l)}}$ of the post-activations $y^{(l)}$, as discussed in Sec. 2.4.2. We here show how higher-order cumulants $G^{(n)}_{y^{(l)}}$ of the post-activations scale with the network width N in the subsequent linear layer. More specifically, they become negligible for wide networks $N \gg 1$.

We build on the arguments of the previous App. A.2 on weakly correlated Gaussian variables. In layer l we have pre-activations

$$z_i^{(l)} = \sum_{a=1}^{N} W_{ia}^{(l)} \, y_a^{(l-1)} + b_i^{(l)}, \tag{A.37}$$

yielding post-activations

$$y_i^{(l)} = \phi\big(z_i^{(l)}\big). \tag{A.38}$$

We study the case that pre-activations $z_i^{(l)}$ are Gaussian distributed and weakly correlated to order c

$$\langle\langle z_i^{(l)} z_j^{(l)} \rangle\rangle \overset{i \neq j}{=} \mathcal{O}(c). \tag{A.39}$$

We now derive conditions for which also pre-activations $z_i^{(l+1)}$ in the next layer have this property. It then follows by induction that this holds true for all network layers, justifying our approximation of the pre-activation distribution as Gaussian in all hidden layers. We start by defining centered variables

$$\tilde{z} := z - \langle z \rangle$$

and

$$\begin{aligned} y = f(\tilde{z}) &:= \phi\big(\langle z \rangle + \tilde{z}\big) \\ \tilde{y} = \tilde{f}(\tilde{z}) &:= f(\tilde{z}) - \big\langle f(\tilde{z}) \big\rangle. \end{aligned}$$

To ensure that the pre-activations $z^{(l)}$ do not explode in magnitude and stay in the dynamic range of the activation function that we assume to be $\mathcal{O}(1)$, we require the variance of pre-activations to be of order one

$$\big\langle \big(\tilde{z}_a^{(l)}\big)^2 \big\rangle = \mathcal{O}(1).$$

Thus, the post-activations also remain of order one

$$\big\langle \big(\tilde{f}_a^{(l)}\big) \big\rangle = \mathcal{O}(1).$$

Such regularity assumptions are also often enforced by normalization methods such as batch-normalization. In the case that the post-activations $y_a^{(l)}$ are uncorrelated, we get for the variance of pre-activations in the next layer

$$\mathcal{O}(1) \overset{!}{=} \langle \big(\tilde{z}_i^{(l+1)}\big)^2 \rangle = \sum_a \big[W_{ia}^{(l+1)}\big]^2 \langle \big(\tilde{f}_a^{(l)}\big)^2 \rangle \overset{!}{=} \sum_a \big[W_{ia}^{(l+1)}\big]^2 \mathcal{O}(1).$$

For both conditions to hold, we require

$$W_{ia}^{(l+1)} = \mathcal{O}(N^{-\frac{1}{2}}), \tag{A.40}$$

i.e. rows and columns of the matrix $W^{(l+1)}$ are vectors of unit length.

Next we look at the case that the post-activations $y_a^{(l)}$ are weakly correlated to order c so that

$$\langle \tilde{y}_a^{(l)} \tilde{y}_b^{(l)} \rangle = \langle \tilde{f}_a^{(l)} \tilde{f}_b^{(l)} \rangle =: C_{ab} = \mathcal{O}(c) \tag{A.41}$$

between the outputs of layer l across different neuron indices $a \neq b$. Then the variance of pre-activations in the subsequent layer is given by

$$\langle \big(\tilde{z}_i^{(l+1)}\big)^2 \rangle = \sum_{a,b} W_{ia}^{(l+1)} W_{ib}^{(l+1)} C_{ab}. \tag{A.42}$$

We want the correlations between neurons to stay controlled, so we require

$$\mathcal{O}(c) \overset{!}{=} \langle \tilde{z}_i^{(l+1)} \tilde{z}_j^{(l+1)} \rangle = \sum_{a,b} W_{ia}^{(l+1)} W_{jb}^{(l+1)} C_{ab} \quad \forall i \neq j, \tag{A.43}$$

which can be understood as requiring that different rows $W_{i\circ}^{(l+1)}$ and $W_{j\circ}^{(l+1)}$ project out mutually nearly orthogonal sub-spaces of the space of principal components of C. Alternatively, this can be seed as different neurons i and j each specializing on sub-spaces with little mutual overlap.

Finally, we consider higher-order cumulants. For $n \geq 3$, we get from Eq. (A.32) and from the condition of weak pairwise correlations Eq. (A.39) that

$$\Big\langle\!\Big\langle \prod_{i=1}^{n} y_i^{(l)} \Big\rangle\!\Big\rangle = \Big\langle\!\Big\langle \prod_{i=1}^{n} \tilde{f}_i(\tilde{z}_i^{(l)}) \Big\rangle\!\Big\rangle = \mathcal{O}\big(c^{\frac{n}{2}+1}\big). \tag{A.44}$$

We get the cumulants of the pre-activations $z_i^{(l+1)}$ from the post-activations $y_i^{(l)}$ as

$$\langle\!\langle z_{i_1}^{(l+1)} \dots z_{i_n}^{(l+1)} \rangle\!\rangle = \sum_{j_1,\dots,j_n=1}^{N} W_{i_1 j_1}^{(l+1)} \dots W_{i_n j_n}^{(l+1)} \Big\langle\!\Big\langle \prod_{k=1}^{n} \tilde{f}_{j_k}(\tilde{z}_{j_k}^{(l)}) \Big\rangle\!\Big\rangle. \tag{A.45}$$

To determine the scaling of higher-order cumulants with the network width N and the weak correlations c, we look at three different cases:

1.) Diagonal contributions: This refers to the special case that all indices $j_1 = \cdots = j_n$

are identical. For order $n \geq 3$, we then get contributions to Eq. (A.45) of order

$$\sum_{j=1}^{N} W_{i_1 j}^{(l+1)} \dots W_{i_n j}^{(l+1)} \underbrace{\langle\!\langle (\tilde{f}_j)^n \rangle\!\rangle}_{\mathcal{O}(1)} = \mathcal{O}\big(N^{1-\frac{n}{2}}\big) \overset{n\geq 3}{<} \mathcal{O}\big(N^{-\frac{1}{2}}\big), \tag{A.46}$$

which are suppressed by large network width $N \gg 1$.

2.) Off-diagonal contributions with all indices being different: Next we look at the case that all neuron indices differ from one another $j_1 \neq j_2 \neq \dots \neq j_n$, allowing us to apply Eq. (A.44). For cumulants of odd order n, any contributions vanish since Eq. (A.44) vanishes. For cumulants of even order n, we have

$$\sum_{(j_1 \neq j_2, \dots, \neq j_n)=1}^{N} W_{i_1 j_1}^{l+1} \cdots W_{i_n j_n}^{l+1} \Big\langle\!\!\Big\langle \prod_{k=1}^{n} \tilde{f}_{j_k}(\tilde{z}_{j_k}^{l}) \Big\rangle\!\!\Big\rangle = \mathcal{O}\Big(\frac{N!}{(N-n)!} N^{-\frac{n}{2}} c^{\frac{n}{2}+1}\Big) \tag{A.47}$$

$$\overset{N \gg n}{=} \mathcal{O}\big(N^{\frac{n}{2}} c^{\frac{n}{2}+1}\big) \tag{A.48}$$

To ensure that these contributions are suppressed by the network width N for $n \geq 3$, we must require that the order of pairwise correlations c is at most

$$c = \mathcal{O}(N^{-1}),$$

so that the contribution in Eq. (A.47) becomes

$$\mathcal{O}\big(N^{-1}\big) \tag{A.49}$$

and is hence suppressed by the network width N also for cumulants of very high order n.

3.) Off-diagonal contributions with two or more indices being identical: In this case a subset of $j_a, j_b, j_c, \dots$ has the same value. We denote the number of different indices $j_1 \neq j_2 \neq \dots \neq j_r$ by $r < n$. On the one hand, each pair of identical indices leads to the appearance of one Kronecker $\delta_{j_a j_b}$, which eliminates one summation $\sum_{j=1}^{N}$ and hence one factor N. On the other hand, identical indices yield higher-order moments of the correlation among weights by Eq. (A.34), so that $\langle \prod_{k=1}^{n} \tilde{f}_{j_k}(\tilde{z}_{j_k}^{l}) \rangle = \mathcal{O}\big(c^{\lceil \frac{r}{2} \rceil}\big)$. Combining both effects, we get that contributions are of order

$$\begin{aligned} \mathcal{O}\Big(\frac{N!}{(N-r)!} N^{-\frac{n}{2}} c^{\lceil \frac{r}{2} \rceil}\Big) &\overset{N \gg 1, \epsilon = \mathcal{O}(N^{-1})}{=} \mathcal{O}\big(N^{r} N^{-\frac{n}{2}} N^{-\lceil \frac{r}{2} \rceil}\big) \\ &= \mathcal{O}\big(N^{\lfloor \frac{r}{2} \rfloor - \frac{n}{2}}\big) < \mathcal{O}\big(N^{-\frac{1}{2}}\big), \end{aligned} \tag{A.50}$$

where we used $r < n$ in the last step and wrote the worst case upper bound where $n = r + 1$ for n odd. In consequence, also contributions from partial diagonal terms are suppressed by the network width N.

To summarize, we find that in the presence of weak pairwise correlations of order $c < \mathcal{O}(N^{-1})$ among network weights, a Gaussian approximation of the pre-activation distribution continues to hold under two conditions: a.) if the network weights scale as Eq. (A.40) and b.) if the rows of the weight matrix $W^{(l)}$ in every layer l additionally obey the approximate orthonormality condition Eq. (A.43). The second condition ensures that all N neurons per layer are used efficiently to represent the entire variability of the distribution from the previous layer. Thereby, redundancy in the representations learned by the neurons is avoided, which makes sense from a functional perspective.

Finally, we point out that Eq. (A.43) always holds for untrained networks that are initialized in a Gaussian manner: the network parameters $\theta = \{W^{(l)}, b^{(l)}\}_{l=1,\ldots,L+1}$ are drawn i.i.d. from centered Gaussians $W_{rs}^{(l)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, \sigma_w^2/N_{l-1}\right)$, $b_r^{(l)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, \sigma_b^2\right)$. This initialization precisely ensures that the magnitude of the covariance of pre-activations $z^{(l)}$ stays the same across layers

$$\begin{aligned}
\langle \tilde{z}_i^{(l+1)} \, \tilde{z}_j^{(l+1)} \rangle &= \sum_{a,b} W_{ia}^{(l+1)} W_{jb}^{(l+1)} \, C_{ab}^{(l)} \\
&\overset{N \gg 1}{\approx} \langle \sum_{a,b} W_{ia}^{(l+1)} W_{jb}^{(l+1)} \, C_{ab}^{(l)} \rangle_W \\
&= \delta_{ij} \frac{\sigma_w^2}{N} \sum_a C_{aa}^{(l)} = \mathcal{O}(c).
\end{aligned}$$

As the covariance in the next layer is then approximately diagonal, the above derivations simplify significantly. We highlight that the above considerations apply to the case of trained networks under certain conditions where correlations between network weights lead to correlations between pairs of pre-activations $(z_i^{(l)}, z_j^{(l)})$.

A.4 Networks with quadratic activation function

In Sec. 2.5.2, we show in Fig. 2.2 the transformation of the data distribution in networks with ReLU activations. Here, we supplement the corresponding figures for networks with a quadratic activation function in Fig. A.1 and Fig. A.2.

A.5 Depth scales of signal propagation in neural networks

Here we examine the connection between our work and that of Poole et al. (2016). To this end, we denote the pre-activation of neuron k in layer l for a specific data sample x_α in a network with parameters θ as $z_{\theta k,\alpha}^{(l)}$. Poole et al. (2016) investigate ensembles of networks over random realizations of network parameters θ. On the level of pre-activations, they investigate the following family of distributions

$$\tilde{p}_{\{\alpha\}}^{(l)}(\{z_k\}) = \langle \prod_{k,\alpha} \delta(z_{k,\alpha} - z_{k\theta,\alpha}^{(l)}) \rangle_\theta. \tag{A.51}$$

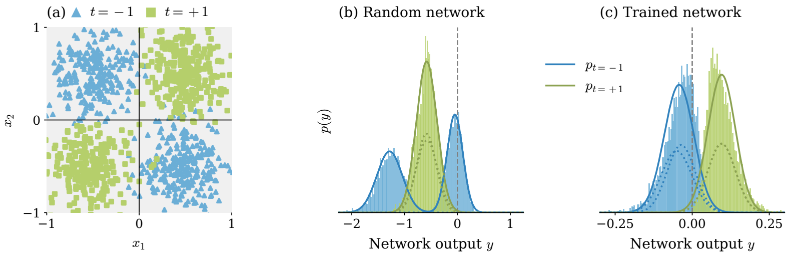


Figure A.1: Information flow in networks with quadratic activation function on the XOR task. (a) We characterize the input distribution as a Gaussian mixture model. The class labels $t = \pm 1$ are assigned to data samples x_α (blue and green dots) based on the mixture component they were drawn from. (b)-(c) Distribution of the network output for (b) a random network and (c) a network trained to convergence. Each mixture component (dashed curves) is propagated through the network as in Eq. (2.22)-(2.23), yielding the class-conditional distributions (solid curves) as a superposition in Eq. (2.28). Mapping a set of test data points by the network yields empirical estimates of the class-conditional distributions (blue and green histograms). For binary classification, we set the classification threshold to be $y = 0$ (gray lines). The trained network in (c) achieves $\mathcal{P} = 90.46\%$ performance. Other parameters: $\phi(z) = z + \epsilon z^2$, network depth $L = 1$, width $N = 10$.

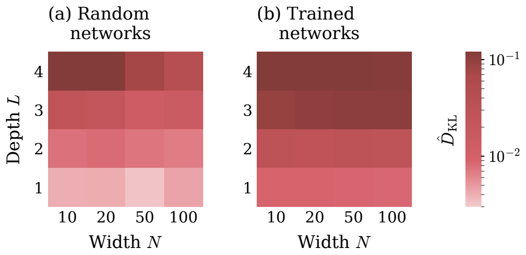


Figure A.2: Comparison between theoretical and empirical output distribution for (a) random networks and (b) networks trained to convergence. The normalized Kullback-Leibler divergence $\hat{D}_{\mathrm{KL}}(p_{\mathrm{emp.}} \| p_{\mathrm{theo.}})$ is used as a deviation measure and averaged over 100 network realizations. Networks are trained on the XOR task; trained networks achieve average performance values of $\mathcal{P} = 96.52\% \pm 0.15\%$ relative to $\mathcal{P}_{\mathrm{opt}} = 97.5\%$. Other parameters: $\phi(z) = z + \epsilon z^2$.

This is a joint distribution for all pre-activations $\{z_{k,\alpha}\}_{\alpha=1,\ldots,P;k=1,\ldots,N}$ for a set of P data samples x_α and for each layer l. For infinitely-wide networks, $\tilde{p}^{(l)}$ factorizes across different neuron indices, making $z_{i,\alpha}^{(l)}$ and $z_{k,\beta}^{(l)}$ independent for $i \neq k$. In addition, these variables are zero-mean Gaussian distributed, meaning a single covariance matrix suffices for their description. Therefore, it is enough to consider the joint statistics of all networks on all pairs of inputs α, β

$$\tilde{p}^{(l)}_{\alpha\beta}(z, z') = \langle \delta(z - z^{(l)}_{\theta,\alpha})\, \delta(z' - z^{(l)}_{\theta,\beta}) \rangle_\theta . \tag{A.52}$$

Thus, correlation functions in Poole et al.'s work describe fluctuations across different realizations of network parameters. Their mean-field theory for deep feed-forward networks is analogous to the classical mean-field theory of random recurrent networks (Molgedey, Schuchhardt, and Schuster, 1992); the reason being that in recurrent networks with discrete-time updates the equal-time statistics are equal to the same-layer statistics of a deep feed-forward network (Segadlo et al., 2022).

In contrast, our work focuses on individual networks characterized by a fixed set of parameters θ over the distribution $p(x)$ of data samples x_α. Consequently, correlations in our work measure the variability of the network state over different data points. To make this explicit, we examine the following family of distributions

$$p^{(l)}_\theta(\{z_k\}) = \langle \prod_k \delta(z_k - z^{(l)}_{\theta,\alpha}) \rangle_\alpha . \tag{A.53}$$

This is a joint distribution of all neurons k for each layer l and network parameters θ, but for the ensemble of data points α.

A key difference lies in the expectation across network parameters θ in Eq. (A.51) versus the expectation over data points α in Eq. (A.53). In the limit of large network width, however, networks exhibit self-averaging behavior: The ensemble of network parameters θ in Poole et al. (2016) tends to concentrate on a typical behavior that is representative of any (likely) individual realization. For the theoretical expressions, this means that the empirical distribution of (z_k, z'_k) across neurons k for any random choice of parameters θ converges to the same form as $\tilde{p}$. Thus, for large N, Eq. (A.52) approaches the empirical average over neuron activations

$$\tilde{p}^{(l)}_{\alpha\beta}(z, z') \overset{\text{self-averaging}}{\simeq} N^{-1} \sum_k \delta(z - z^{(l)}_{\theta k,\alpha})\, \delta(z' - z^{(l)}_{\theta k,\beta}), \quad \forall \theta.$$

This self-averaging property can be demonstrated using a saddle point approximation of the moment-generating function after averaging over the disorder given by the network parameters θ (e.g, Schuecker et al., 2016; Crisanti and Sompolinsky, 2018; Helias and Dahmen, 2020; Segadlo et al., 2022; Bordelon and Pehlevan, 2023).

To obtain the results by Poole et al. (2016) using our formalism, we begin from the definition of pre-activations $z_i^{(l+1)} = \sum_k W_{ik}^{(l+1)} y_k^{(l)} + b_i^{(l+1)}$. Given that $W_{ik}^{(l+1)}$ and $b_i^{(l+1)}$ are Gaussian distributed, pre-activations for a single fixed data sample x_α also become Gaussian. We will suppress the superscript α in the following for brevity. The mean and covariance of the pre-activations are given by

$$\begin{aligned} M_{z^{(l+1)},i} &:= \langle \sum_k W_{ik}^{(l+1)} y_k^{(l)} + b_i^{(l+1)} \rangle_{W,b} \\ &= \sum_k \left\langle W_{ik}^{(l+1)} \right\rangle_{W^{(l+1)}} \left\langle y_k^{(l)} \right\rangle_{W,b} + \left\langle b_i^{(l+1)} \right\rangle_{b^{(l+1)}} = 0, \end{aligned} \tag{A.54}$$

$$\begin{aligned} S_{z^{(l+1)},ij} &:= \langle \sum_{k,m} W_{ik}^{(l+1)} W_{jm}^{(l+1)} y_k^{(l)} y_m^{(l)} + b_i^{(l+1)} b_j^{(l+1)} \rangle_{W,b} \\ &= \sum_{k,m} \left\langle W_{ik}^{(l+1)} W_{jm}^{(l+1)} \right\rangle_{W^{(l+1)}} \left\langle y_k^{(l)} y_m^{(l)} \right\rangle_{W,b} + \left\langle b_i^{(l+1)} b_j^{(l+1)} \right\rangle_{b^{(l+1)}} \\ &= \delta_{ij} \left(\sigma_w^2 \left\langle y^{(l)} y^{(l)} \right\rangle_{W,b} + \sigma_b^2 \right) \\ &=: \delta_{ij}\, S_{z^{(l+1)}}, \end{aligned} \tag{A.55}$$

where

$$S_{z^{(l+1)}} = \sigma_w^2 \, \langle \phi(z^{(l)}) \phi(z^{(l)}) \rangle_{z^{(l)} \sim \mathcal{N}(0, S_{z^{(l)}})} + \sigma_b^2. \tag{A.56}$$

Here, we employed the transformation by the activation function Eq. (2.16). For the moments $\langle y_k^{(l)} \rangle_{W,b}$ and $\langle y_k^{(l)} y_k^{(l)} \rangle_{W,b}$ in layer l, we take the average over weights and biases in all previous layers $l' \leq l$ at once; they are the same for all neurons k so we denote $\langle y_k^{(l)} \rangle_{W,b} = \langle y^{(l)} \rangle_{W,b}$ and $\langle y_k^{(l)} y_k^{(l)} \rangle_{W,b} = \langle y^{(l)} y^{(l)} \rangle_{W,b}$. According to Eq. (A.54) and Eq. (A.55), there are no correlations between different neurons when taking the average over networks.

Analogously, we get for the covariance between pre-activations of a pair of networks for two different inputs x_α and x_β:

$$S_{z_\alpha^{(l+1)} z_\beta^{(l+1)}} = \sigma_w^2 \, \langle \phi(z_\alpha^{(l)}) \phi(z_\beta^{(l)}) \rangle_{(z_\alpha^{(l)}, z_\beta^{(l)}) \sim \mathcal{N}\left(0, \{S_{z_\alpha^{(l)} z_\beta^{(l)}}\}\right)} + \sigma_b^2. \tag{A.57}$$

Here $\mathcal{N}\big(0, \{S_{z_\alpha^{(l)} z_\beta^{(l)}}\}\big)$ refers to the centered Gaussian distribution for $(z_\alpha^{(l)}, z_\beta^{(l)})$ with covariance $\begin{pmatrix} S_{z_\alpha^{(l)}} & S_{z_\alpha^{(l)} z_\beta^{(l)}} \\ S_{z_\beta^{(l)} z_\alpha^{(l)}} & S_{z_\beta^{(l)}} \end{pmatrix}$.

We can link to our results by looking at a pair of inputs x_α and x_β. These inputs are fed into the network as $y_\alpha^{(0)}$ and $y_\beta^{(0)}$. According to Poole et al. (2016), the overlap $O_{\alpha\beta}^{(l)}$ of network states after l layers is determined by solving the joint iterative equations above. For the network width going to infinity $N \to \infty$, this overlap becomes self-averaging and concentrates around its mean value over y:

$$\begin{aligned} O^{(l)}_{\alpha\beta} &:= N^{-1}\sum_k y^{(l)}_{k,\alpha} y^{(l)}_{k,\beta} \\ &\simeq \langle y^{(l)}_\alpha y^{(l)}_\beta \rangle_{W,b} \\ &= \langle \phi(z^{(l)}_\alpha)\phi(z^{(l)}_\beta)\rangle_{(z^{(l)}_\alpha, z^{(l)}_\beta)\sim\mathcal{N}\left(0,\{S_{z^{(l)}_\alpha z^{(l)}_\beta}\}\right)} \\ &= \sigma_w^{-2}\left(S_{z^{(l+1)}_\alpha z^{(l+1)}_\beta} - \sigma_b^2\right). \end{aligned} \tag{A.58}$$

We consider the same case as in Poole et al. (2016), which is a deep network at initialization. Since the statistics of $z^{(l)}$ and $y^{(l)}$ converge to a fixed point, we can assume the covariance to be constant for deep enough layers l:

$$\begin{aligned} S_{z^{(l)}_\alpha} &= A_0, \quad \forall \alpha, \\ S_{z^{(l)}_\alpha z^{(l)}_\beta} &= C_0, \quad \forall \alpha \neq \beta, \end{aligned} \tag{A.59}$$

These constants are given as stationary solutions to self-consistency equations: For A_0, we have Eq. (A.56)

$$A_0 = \sigma_w^2 \, \langle \phi(z)\phi(z)\rangle_{z\sim\mathcal{N}(0,A_0)} + \sigma_b^2, \tag{A.60}$$

and C_0 is determined by Eq. (A.57)

$$C_0 = \sigma_w^2 \, \langle \phi(z_1)\phi(z_2)\rangle_{(z_1,z_2)\sim\mathcal{N}\left(0,\left[\begin{smallmatrix} A_0 & C_0 \\ C_0 & A_0 \end{smallmatrix}\right]\right)} + \sigma_b^2 \,.$$

Consider pairs of inputs $(y^{(0)}_\alpha, y^{(0)}_\beta)$ where the pre-activation statistics are nearly at the fixed-point values. Suppose each data point has variance $S_{z^{(l)}} = A_0$, and for any pair (α, β) the covariance of pre-activations in the first layer can be written as

$$S_{z^{(1)}_\alpha z^{(1)}_\beta} = C_0 + \delta C^{(1)}_{\alpha\beta}, \tag{A.61}$$

with $\delta C^{(1)}_{\alpha\beta} \ll C_0$. Given these conditions, we can then determine decay constants as a function of the layer index l.

We determine the transformation of $\delta C^{(l)}_{\alpha\beta}$ across layers by linearizing Eq. (A.57):

$$\delta C^{(l+1)}_{\alpha\beta} = \sigma_w^2 \, \langle \phi'(z^{(l)}_\alpha)\phi'(z^{(l)}_\beta)\rangle \, \delta C^{(l)}_{\alpha\beta} + \mathcal{O}\left[\left(\delta C^{(l)}_{\alpha\beta}\right)^2\right].$$

For the derivative by the covariance, we used Price's theorem (Price, 1958; Papoulis and Pillai, 2002, Appendix A), which gives $\partial\langle\phi(z)\phi(z')\rangle/\partial\Sigma_{zz'} = \langle\phi'(z)\phi'(z')\rangle$ with $\phi' = d\phi/dz$ and $\Sigma_{zz'}$ is the covariance of z and z'. Under the homogeneity assumption across data samples Eq. (A.59) and for stationary statistics across layers Eq. (A.60), we obtain

$$\begin{aligned}\langle \phi'(z_\alpha)\phi'(z_\beta)\rangle &= \langle \phi'(z_1)\phi'(z_2)\rangle_{(z_1,z_2)\sim\mathcal{N}\left(0,\left[\begin{smallmatrix} A_0 & C_0 \\ C_0 & A_0\end{smallmatrix}\right]\right)} \\ &=: \langle \phi'\phi'\rangle.\end{aligned}$$

Overall, we get an exponential decay across network layers l as

$$\begin{aligned}\delta C_{\alpha\beta}^{(l+1)} &= \left(\sigma_w^2 \langle \phi'\phi'\rangle\right)^l \delta C_{\alpha\beta}^{(1)} \\ &= e^{-\frac{l}{\xi}}\,\delta C_{\alpha\beta}^{(1)},\end{aligned} \tag{A.62}$$

governed by depth scale

$$\xi = -1/\ln\left[\sigma_w^2 \langle \phi'\phi'\rangle\right], \tag{A.63}$$

which is precisely the depth scale found in Poole et al. (2016) for network ensembles. Its inverse ξ^{-1} corresponds to the Lyapunov exponent in Molgedey, Schuchhardt, and Schuster (1992). The depth scale diverges at the transition to chaos, which is at $\sigma_w^2 \langle \phi'\phi'\rangle = 1$. The overlaps of activations Eq. (A.58) decay with the same depth scale, because its change $\delta O_{\alpha\beta}^{(l-1)}$ relates linearly to $\delta C_{\alpha\beta}^{(l)}$ as $\delta C_{\alpha\beta}^{(l)} = \sigma_w^2\,\delta O_{\alpha\beta}^{(l-1)}$ and thus

$$\delta O_{\alpha\beta}^{(l)} = \left(\sigma_w^2 \langle \phi'\phi'\rangle\right)^l \delta O_{\alpha\beta}^{(0)}. \tag{A.64}$$

In consequence, both the covariance of pre-activations $S_{z_\alpha^{(1)} z_\beta^{(1)}}$ and the overlaps of activations $O_{\alpha\beta}^{(l)}$ converge to fixed points, which are related by $O_0 = \sigma_w^{-2}(C_0 - \sigma_b^2)$.

To link these results to our work for individual network realizations, we write the overlap $O_{\alpha\beta}^{(l)}$ in terms of the probability distribution over different data samples, yielding with Eq. (A.58):

$$\frac{1}{P\,(P-1)} \sum_{(\alpha\neq\beta)=1}^{P} O_{\alpha\beta}^{(l)} \simeq N^{-1} \sum_{k=1}^{N} \left[\frac{1}{P}\sum_{\alpha=1}^{P} y_{k,\alpha}^{(l)}\right]\left[\frac{1}{P}\sum_{\beta=1}^{P} y_{k,\beta}^{(l)}\right] + \mathcal{O}(P^{-1}) \tag{A.65}$$

$$\overset{P\gg 1}{\simeq} N^{-1} \sum_{k=1}^{N} \langle y_{k,\alpha}^{(l)}\rangle_\alpha^2 \tag{A.66}$$

$$\simeq N^{-1} \sum_{k=1}^{N} \left(\mu_{y^{(l)},\{k\}}\right)^2. \tag{A.67}$$

In the last line, the mean post-activation $\mu_{y^{(l)},\{k\}}$ of neuron k in layer l appears, where the average was taken with respect to the ensemble of all data points α. This quantity results from iterating Eqs. (2.15) and (2.18).

In Fig. A.3, we demonstrate that our predictions from Eq. (A.67) match the depth scales obtained from Poole et al.'s theory Eq. (A.63). Our theory even goes beyond this, since it applies to individual networks and Eq. (A.67) allows us to account for

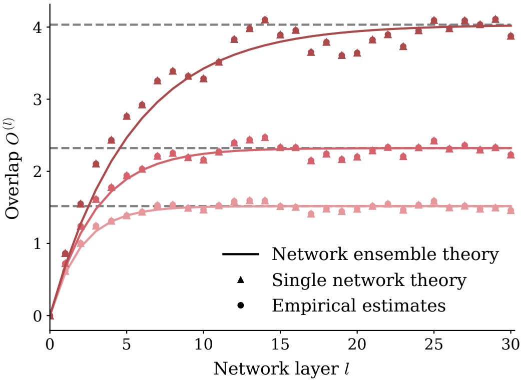


Figure A.3: Depth scale of cumulant propagation at initialization with randomly drawn weights. We plot the evolution of overlaps $O^{(l)}$ in Eq. (A.58) and (A.67) across layers l. Solid lines indicate the predicted decay with $e^{-l/\xi}$ and depth scale ξ as in Eq. (A.63) from Poole et al. (2016) for network ensembles. Dashed lines mark the fixed point values O_0 to which the overlaps $O^{(l)}$ converge. Dots correspond to empirical estimates of the overlaps $O^{(l)}_{\alpha\beta}$ in Eq. (A.65). Triangles correspond to the overlaps based on the cumulant propagation for individual network realizations in Eq. (A.67). Empirical estimates and the cumulant predictions agree well; the symbols are overlapping. We draw input data $(x_\alpha)_\alpha$ from a Gaussian $\mathcal{N}(0, A_0)$, where A_0 is specified in Eq. (A.60). Other parameters: $\sigma_w = \sigma_b \in [0.76, 0.81, 0.85]$ (from light to dark colors), quadratic activation function $\phi(z) = z + \epsilon z^2$, $\epsilon = 0.1$.

variability due to individual network realizations. Notably, while the depth scale ξ for network ensembles reflects the evolution of the second moments, we observe from Eq. (A.67) that the depth scale ξ characterizes the evolution of the squared means across data samples in individual network realizations.

Appendix B

Critical feature learning in deep neural networks

B.1 Conjugate kernels yield training error

We here aim to give physical meaning to the conjugate kernel $\tilde{C}^{(L)}$ in the output layer. To this end, we replace the regularizer $\kappa\mathbb{I}$ in the expression of the network prior in Eq. (3.27) by a generic covariance matrix $K_{\alpha\beta}$

$$p(Y|X,K) := \int \mathcal{D}C \int \mathcal{D}f\, \mathcal{N}(Y|f,K)\, \mathcal{N}(f|0,C^{(L)})\, p(C). \tag{B.1}$$

We observe that the statistics of Y result from a convolution of two centered Gaussian distributions with covariances $C^{(L)}$ and K, respectively. We obtain the action as

$$S(C|K) = -\frac{1}{2} y_\alpha \big[C^{(L)} + K\big]^{-1}_{\alpha\beta} y_\beta - \frac{1}{2}\ln\det(C+K) - \Gamma(C). \tag{B.2}$$

By expressing Eq. (B.1) explicitly

$$\begin{aligned} p(Y|X,K) = \frac{1}{(2\pi)^{\frac{M}{2}}\,(\det K)^{\frac{1}{2}}} \int \mathcal{D}C \int \mathcal{D}f\, \exp\Big(-\frac{1}{2}(y_\alpha - f_\alpha)\,\big[K^{-1}\big]_{\alpha\beta}\,(y_\beta - f_\beta)\Big) \\ \times \mathcal{N}(f|0,C^{(L)})\, p(C), \end{aligned}$$

we see that K^{-1} acts as a bi-linear source term to the second moment of the discrepancies

$$\begin{aligned} -\frac{1}{2}\,\langle (y_\alpha - f_\alpha)\,(y_\beta - f_\beta)\rangle &= \frac{\partial}{\partial[K^{-1}]_{\alpha\beta}}\Big(\ln p(Y|X,K) - \frac{1}{2}\det K^{-1}\Big)\Big|_{K=\kappa\mathbb{I}} \\ &\simeq \frac{\partial}{\partial[K^{-1}]_{\alpha\beta}} S(C|K) + \underbrace{\frac{\partial S}{\partial C}}_{=0}\,\frac{\partial C}{\partial (K^{-1})_{\alpha\beta}} - \frac{1}{2}K\Big|_{K=\kappa\mathbb{I}} \\ &= \Big[-\frac{1}{2}K\big[C^{(L)}+K\big]^{-1} YY^{\mathsf{T}}\big[C^{(L)}+K\big]^{-1}K + \frac{1}{2}K\,(C^{(L)}+K)^{-1}K\Big]_{\alpha\beta} - \frac{1}{2}\kappa\mathbb{I} \end{aligned} \tag{B.3}$$

$$
\begin{aligned}
&= \Big[-\frac{1}{2}\kappa^2 \left[C^{(L)} + \kappa\mathbb{I}\right]^{-1} (YY^{\mathsf{T}}) \left[C^{(L)} + \kappa\mathbb{I}\right]^{-1} + \frac{1}{2}\kappa^2 \left(C^* + \kappa\mathbb{I}\right)^{-1} - \frac{1}{2}\kappa\mathbb{I}\Big]_{\alpha\beta} \\
&\overset{(3.60)}{=} -\kappa^2 \tilde{C}^*_{\alpha\beta} - \frac{1}{2}\kappa\delta_{\alpha\beta},
\end{aligned}
$$

where we approximate the log-probability by the action from the first to second line. In this derivation, we used that $\partial[K^{-1}]_{\gamma\delta}/\partial K_{\alpha\beta} = -K^{-1}_{\gamma\alpha}\,K^{-1}_{\beta\delta}$ and by symmetry $\partial K_{\gamma\delta}/\partial[K^{-1}]_{\alpha\beta} = -K_{\gamma\alpha}\,K_{\beta\delta}$. Overall, we can write

$$
\begin{aligned}
\langle \Delta_\alpha\, \Delta_\beta \rangle &= 2\kappa^2 \tilde{C}^{(L)}_{\alpha\beta} + \kappa\delta_{\alpha\beta}, \\
\Delta_\alpha &= y_\alpha - f_\alpha,
\end{aligned}
\tag{B.4}
$$

and get the expected training error as

$$
\begin{aligned}
\langle \mathcal{L} \rangle :=& \frac{1}{2} \mathrm{tr} \, \langle \Delta\, \Delta \rangle \\
=& \kappa^2 \, \mathrm{tr} \tilde{C}^{(L)} + \frac{1}{2}\kappa P.
\end{aligned}
\tag{B.5}
$$

From these two expressions Eq. (B.4) and (B.5), we see that the conjugate kernel $\tilde{C}^{(L)}$ in the output layer corresponds to the expected squared errors Δ between target and network output plus a fixed offset.

B.2 Deep linear networks

To draw a link to (Li and Sompolinsky, 2021; Zavatone-Veth, Tong, and Pehlevan, 2022; Yang et al., 2023), we here consider the special case of deep linear networks. We obtain closed-form expressions for the forward-backward propagation equations of the posterior kernels in Eq. (3.55) and (3.62) without requiring the additional step of the perturbative treatment.

For deep linear networks, we recover the action in Yang et al. (2023), Eq. (1), for deep kernel machines. This result has three main implications:

1. In the proportional limit $P = \nu\, N$, deep linear networks converge to deep kernel machines.
2. The iterative forward-backward equations for the posterior kernels in this section apply to both deep linear neural networks and deep kernel machines.
3. We may use the alternative view of kernel adaptation generated by kernel fluctuations, as outlined in Sec. 3.5, also to deep linear networks and deep kernel machines; thus corrections apply to linear networks of finite size.

We here follow the same derivation steps as for the non-linear case in Sec. 3.4.1, indicating relevant differences. For deep linear networks, we choose $\phi = \mathrm{id}$, so that the hidden layers now take the form

$$h_\alpha^{(l)} = W^{(l)} h_\alpha^{(l-1)} + b^{(l)} \quad l = 1, \dots, L.$$

Given the same Gaussian priors on the network parameters, the network prior in Eq. (3.25) is given by

$$p(f|X) = \int \mathcal{D}\{\tilde{C}, C\} \mathcal{N}\left(f|0, C_{\alpha\beta}^{(L)}\right) \exp\left(-\sum_{l=1}^{L} \tilde{C}_{\alpha\beta}^{(l)} C_{\alpha\beta}^{(l)} + \mathcal{W}(\tilde{C}|C)\right), \tag{B.6}$$

$$\mathcal{W}(\tilde{C}|C) = \sum_{l=1}^{L} \sum_{\alpha,\beta} \tilde{C}_{\alpha\beta}^{(l)} \sigma_b^2 + N \sum_{l=1}^{L} \ln \left\langle \exp\left(\frac{\sigma_w^2}{N} \tilde{C}_{\alpha\beta}^{(l)} h_\alpha^{(l-1)} h_\beta^{(l-1)}\right)\right\rangle_{\mathcal{N}(0, C^{(l-1)})}, \tag{B.7}$$

$$C_{\alpha\beta}^{(0)} = \frac{\sigma_w^2}{D} \left(X X^\mathsf{T}\right)_{\alpha\beta} + \sigma_b^2. \tag{B.8}$$

A key difference to the non-linear case is that the expectation values in the definition of $\mathcal{W}(\tilde{C}|C)$ are with respect to Gaussians, allowing for a closed-form solution

$$\mathcal{W}(\tilde{C}|C) = \sum_{l=1}^{L} \mathcal{W}(\tilde{C}^{(l)}|C^{(l-1)}) \tag{B.9}$$

$$\begin{aligned} \mathcal{W}(\tilde{C}^{(l)}|C^{(l-1)}) &:= \sum_{\alpha,\beta} \tilde{C}_{\alpha\beta}^{(l)} \sigma_b^2 + N \ln \left\langle \exp\left(\frac{\sigma_w^2}{N} \tilde{C}_{\alpha\beta}^{(l)} h_\alpha^{(l-1)} h_\beta^{(l-1)}\right)\right\rangle_{\mathcal{N}(0, C^{(l-1)})} \\ &= \sum_{\alpha,\beta} \tilde{C}_{\alpha\beta}^{(l)} \sigma_b^2 - \frac{N}{2} \ln\det\left([C^{(l-1)}]^{-1} - 2\frac{\sigma_w^2}{N} \tilde{C}^{(l)}\right) - \frac{N}{2} \ln\det(C^{(l-1)}). \end{aligned}$$

This cumulant-generating function also follows the scaling form so that the limit $\lambda(k) := \lim_{N\to\infty} \mathcal{W}(Nk)/N$ exists. Thus, we perform a saddle point approximation of the conditional probabilities $p(C^{(l)}|C^{(l-1)})$ (cf. Eq. (3.29)), yielding a rate function Γ

$$\begin{aligned} -\ln p\left(C^{(l)}|C^{(l-1)}\right) &:= -\int \mathcal{D}\tilde{C}^{(l)} \exp\left(-\mathrm{tr}\, \tilde{C}^{(l)\mathsf{T}} C^{(l)} + \mathcal{W}\left(\tilde{C}^{(l)}|C^{(l-1)}\right)\right) && \text{(B.10)} \\ &\simeq \Gamma(C^{(l)}|C^{(l-1)}) && \text{(B.11)} \\ &= \frac{N}{2\sigma_w^2} \mathrm{tr}\left([C^{(l-1)}]^{-1}(C^{(l)} - \sigma_b^2)\right) - \frac{N}{2} \ln\det\left(C^{(l)} - \sigma_b^2\right) \\ &\quad + \frac{N}{2} \ln\det(C^{(l-1)}) + \text{const.}, \end{aligned}$$

Here, we neglected all terms constant in the kernels $C^{(l)}$ and inserted the stationarity condition, which gives

$$
\begin{aligned}
0 &\overset{!}{=} \frac{\partial}{\partial \tilde{C}^{(l)}_{\alpha\beta}} \Big(\mathrm{tr} \tilde{C}^{(l)\mathsf{T}} \tilde{C}^{(l)} - \mathcal{W}\left(\tilde{C}^{(l)} | C^{(l-1)}\right)\Big) \\
&= C^{(l)}_{\alpha\beta} - \sigma_b^2 - \sigma_w^2 \left([C^{(l-1)}]^{-1} - 2\frac{\sigma_w^2}{N} \tilde{C}^{(l)} \right)^{-1}_{\alpha\beta}.
\end{aligned}
\tag{B.12}
$$

From the stationarity condition, we obtain the propagation equation for the kernels $C^{(l)}$ as in Eq. (3.55) for the non-linear case. In the infinite width limit $N \to \infty$, this recovers the known NNGP result where $\tilde{C} = 0$.

We can now draw a link to the work by Yang et al. (2023) studying deep kernel machines. We can write the action as in Eq. (3.59) for the non-linear case using Eq. (B.7) and (B.11) as

$$
\begin{aligned}
\mathcal{S}(C) &:= \ln p(C|Y) \simeq \mathcal{S}_{\mathrm{D}}(C^{(L)}) - \Gamma(C) + \circ , \\
\Gamma(C) &= \sum_{l=1}^{L} \Gamma(C^{(l)} | C^{(l-1)}).
\end{aligned}
\tag{B.13}
$$

For linear networks, the rate function $\Gamma(C^{(l)}|C^{(l-1)})$ in Eq. (B.11) now takes the form of a Kullback-Leibler divergence between two pairs of centered Gaussian covariates with $\langle z^{(l-1)}_{\alpha i} z^{(l-1)}_{\beta j} \rangle = \delta_{ij} \sigma_w^2 C^{(l-1)}$ and $\langle z^{(l)}_{\alpha i} z^{(l)}_{\beta j} \rangle = \delta_{ij}\big(C^{(l)} - \sigma_b^2\big)$, respectively:

$$
\begin{aligned}
& D_{\mathrm{KL}}(\mathcal{N}(0, C^{(l)} - \sigma_b^2) || \mathcal{N}(0, \sigma_w^2 C^{(l-1)})) \\
&= -\big\langle \ln \mathcal{N}(z^{(l)} | 0, \sigma_w^2 C^{(l-1)}) \big\rangle_{z^{(l)} \sim \mathcal{N}(0, C^{(l)} - \sigma_b^2)} + \big\langle \ln \mathcal{N}(z^{(l)} | 0, C^{(l)} - \sigma_b^2) \big\rangle_{z^{(a)} \sim \mathcal{N}(0, C^{(a)} - \sigma_b^2)} \\
&= \frac{N}{2\sigma_w^2} \mathrm{tr}\, [C^{(l-1)}]^{\mathsf{T}} (C^{(l)} - \sigma_b^2) + \frac{N}{2} \ln\det\big(C^{(l-1)}\big) - \frac{N}{2} \ln\det\big(C^{(l)} - \sigma_b^2\big) + \mathrm{const.},
\end{aligned}
\tag{B.14}
$$

We here get additional factors N by using that $z_{\alpha i}$ are i.i.d. in $i = 1, \ldots, N$. Up to constant terms, this is the same result as Eq. (B.11). We recover with Eq. (B.13) the main result by Yang et al. (2023), their Eq. (1), for $\sigma_b^2 = 0$, $N_l/N = 1$ and using $K = \mathrm{id}$ for deep kernel machines. For linear networks, our theoretical framework thus agrees with Yang et al. (2023); however, our theoretical framework is more general and also applies to deep non-linear networks.

Next, we obtain the forward-backward equation of the posterior kernels for linear networks. We write the forward iteration in Eq. (B.12) as

$$
C^{(l)} = \sigma_b^2 + \sigma_w^2\, C^{(l-1)} \left(\mathbb{I} - 2\frac{\sigma_w^2}{N} \tilde{C}^{(l)} C^{(l-1)} \right)^{-1}.
\tag{B.15}
$$

For the backward equation, we calculate the saddle point for the kernels C from Eq. (B.13) as $\partial\mathcal{S}(C)/\partial C^{(l)}_{\alpha\beta} \overset{!}{=} 0$, yielding Eq. (3.60) for the output layer $l = L$ and otherwise

$$\begin{aligned}0 &\overset{!}{=} \frac{\partial}{\partial C^{(l)}_{\alpha\beta}}\left(\Gamma(C^{(l)}|C^{(l-1)})+\Gamma(C^{(l+1)}|C^{(l)})\right)\\ &= \tilde{C}^{(l)}_{\alpha\beta}-\frac{\partial}{C^{(l)}_{\alpha\beta}}\mathcal{W}(\tilde{C}^{(l+1)}|C^{(l)}).\end{aligned}$$

Using the explicit form of the cumulant-generating function $\mathcal{W}(\tilde{C}^{(l+1)}|C^{(l)})$ in Eq. (B.11) for the linear case, we get

$$\tilde{C}^{(l)}=\sigma_w^2\,\tilde{C}^{(l+1)}\left(\mathbb{I}-2\frac{\sigma_w^2}{N}\,\tilde{C}^{(l+1)}C^{(l)}\right)^{-1}. \tag{B.16}$$

Interestingly, forward and backward equation Eq. (B.15)-(B.16) adhere to the following symmetry relation

$$[\sigma_w^2\tilde{C}^{(l)}]^{-1}\tilde{C}^{(l-1)}=[\sigma_w^2 C^{(l-1)}]^{-1}\,(C^{(l)}-\sigma_b^2)=\left(\mathbb{I}-2\frac{\sigma_w^2}{N}\,\tilde{C}^{(l)}C^{(l-1)}\right)^{-1}. \tag{B.17}$$

We test our theoretical results for linear networks on a linearly separable Ising task: The data samples $x_\alpha\in\{\pm1\}^D$ belong to two classes. For class +1, the elements $x_{\alpha i}$ realize the value $x_{\alpha i}=+1$ with a probability of $p_1=0.5-\Delta p$ and the value $x_{\alpha i}=-1$ with a probability of $p_2=0.5+\Delta p$. Each element $x_{\alpha i}$ is drawn independently. For class -1, the probabilities for the value realizations are inverted. For larger Δp, the task becomes more separable.

We are interested in the behavior in N for fixed training load $\nu=P/N$. In Fig. B.1 we show results for a single-hidden-layer network. We plot the mean-squared error between the empirically measured kernels and the feature-corrected kernels derived in this section, the NNGP kernels and the linear approximation in $\tilde{C}$ of the feature corrections as in the main text. For large network width N, the feature-corrected kernels from the full theory converge to the empirically measured kernels, while their linear approximations in $\tilde{C}$ exhibit only slightly larger deviations, thus warranting the approximation in the main text. Finally, the empirically measured kernels consistently deviate from the NNGP result, which indicates the need for feature corrections.

B.3 Target-kernel-adaptation in linear networks

We revisit the topic of adaptation of the feature-corrected kernels $C^{(l)}$ towards the target kernel YY^T in the setting of a linear network; we explicitly derive a correction term towards the target kernel.

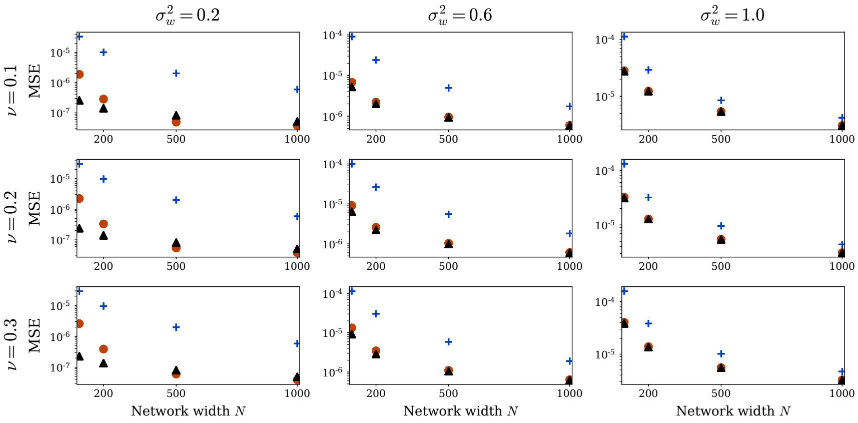


Figure B.1: Feature corrections in a single-hidden-layer network for fixed training load $\nu = P/N$. The mean squared error $\mathrm{MSE}(C, C_{\mathrm{emp}}) = 1/D^2 \sum_{\alpha,\beta=1}^{D} (C_{\alpha\beta} - C_{\alpha\beta}^{\mathrm{emp}})^2$ describes convergence to the empirically measured kernels. The feature-corrected kernels (red: linear approximation; black: full theory) are consistently closer to the empirically measured kernels than the NNGP kernel (blue). Error bars indicate mean and one standard deviation over 10 training data sets. Other parameters: Ising task $\Delta p = 0.2$, $D = 1000$, $\kappa = 0.001$, $\sigma_b^2 = 0.05$.

We start from the second-order expansion of the cumulant-generating function $\mathcal{W}$ in Eq. (B.9) as

$$\begin{aligned} &\mathcal{W}(\tilde{C}^{(l)}|C^{(l-1)}) \qquad \text{(B.18)} \\ &= \sum_{\alpha\beta} \tilde{C}_{\alpha\beta}^{(l)} \sigma_b^2 + \sigma_w^2 \, \tilde{C}_{\alpha\beta}^{(l)} C_{\alpha\beta}^{(l-1)} + \sum_{\gamma\delta} \frac{\sigma_w^4}{2N} \, \tilde{C}_{\alpha\beta}^{(l)} \left(C_{\alpha\gamma}^{(l-1)} C_{\beta\delta}^{(l-1)} + C_{\alpha\delta}^{(l-1)} C_{\beta\gamma}^{(l-1)} \right) \tilde{C}_{\gamma\delta}^{(l)} + \mathcal{O}(\tilde{C}^3). \end{aligned}$$

Here, we rewrote the connected correlation function in terms of the covariance: using its definition $\langle h_\alpha h_\beta, h_\gamma h_\delta \rangle^c = \langle h_\alpha h_\beta h_\gamma h_\delta \rangle - \langle h_\alpha h_\beta \rangle \langle h_\gamma h_\delta \rangle$, we write the fourth moment in terms of cumulants by Wick's theorem and are then left with the pairings $\langle h_\alpha h_\gamma \rangle \langle h_\beta h_\delta \rangle + \langle h_\alpha h_\delta \rangle \langle h_\beta h_\gamma \rangle = C_{\alpha\gamma} C_{\beta\delta} + C_{\alpha\delta} C_{\beta\gamma}$. Applying the stationarity condition Eq. (B.12) to Eq. (B.18), we get

$$C_{\alpha\beta}^{(l+1)} = \sigma_b^2 + \sigma_w^2 \, C_{\alpha\beta}^{(l)} + 2 \frac{\sigma_w^4}{N} \, C_{\alpha\gamma}^{(l)} \, \tilde{C}_{\gamma\delta}^{(l+1)} \, C_{\delta\beta}^{(l)}.$$

For simplicity, we consider $\sigma_b^2 = 0$ and $\kappa = 0$ in the following. Then, we can approximate $\sigma_w^2 C^{(l)} \simeq C^{(l+1)} + \mathcal{O}(N^{-1})$ since the correction term scales with N^{-1}. In the last layer $l = L$, we obtain in this approximation

$$C^{(L)}_{\alpha\beta} \simeq \sigma_w^2 \, C^{(L-1)}_{\alpha\beta} + \frac{2}{N} \, C^{(L)}_{\alpha\gamma} \, \tilde{C}^{(L)}_{\gamma\delta} C^{(L)}_{\delta\beta} + \mathcal{O}(N^{-1}).$$

By inserting $\tilde{C}^{(L)} = \frac{1}{2}(C^{(L)})^{-1} YY^{\mathsf{T}} (C^{(L)})^{-1} - \frac{1}{2}(C^{(L)})^{-1}$ from Eq. (3.60), the correction term is

$$\begin{aligned} \frac{2}{N} \, C^{(L)}_{\alpha\gamma} \, \tilde{C}^{(L)}_{\gamma\delta} C^{(L)}_{\delta\beta} &= \frac{2}{N} \, C^{(L)}_{\alpha\gamma} \left[\frac{1}{2}(C^{(L)})^{-1} YY^{\mathsf{T}} (C^{(L)})^{-1} - \frac{1}{2}(C^{(L)})^{-1} \right]_{\gamma\delta} C^{(L)}_{\delta\beta} \\ &= \frac{1}{N} \big(YY^{\mathsf{T}} - C^{(L)} \big). \end{aligned}$$

Finally, we get

$$C^{(L)}_{\alpha\beta} \simeq \sigma_w^2 \, C^{(L-1)}_{\alpha\beta} + \frac{1}{N} \big(YY^{\mathsf{T}} - C^{(L)} \big)_{\alpha\beta}. \tag{B.19}$$

Thus, we see that to increase the log-likelihood of the data, the kernel receives a correction into the rank-one direction of the target kernel YY^{T}.

B.4 Relation to the Neural Tangent Kernel

In this section, we draw a link to the Neural Tangent Kernel (NTK) (Jacot, Gabriel, and Hongler, 2018; Lee et al., 2018) and show under which assumption it emerges in our theoretical framework. To keep the notation concise, we here set biases to zero so that the network is defined as

$$\begin{aligned} h^{(0)}_\alpha &= W^{(0)} x_\alpha, \\ h^{(l)}_\alpha &= W^{(l)} \phi \left(h^{(l-1)}_\alpha \right) \quad l = 1, \dots, L, \\ f_\alpha &= h^{(L)}_\alpha \in \mathbb{R}. \end{aligned} \tag{B.20}$$

We use the squared error loss

$$\mathcal{L}(f; Y) = \frac{1}{2} \sum_{\alpha=1}^{P} \| f_\alpha - y_\alpha \|^2. \tag{B.21}$$

Further, we assume identical width N for all network layers as in the main text, including the input layer so that we require for the input data $x_\alpha \in \mathbb{R}^N$.

In the NTK setting, network weights at intialization scale as $W^{(l)} = w^{(l)} / \sqrt{N}$ with $\mathcal{O}(1) \sim w^{(l)} \sim \mathcal{N}(0, \sigma_w^2)$ with $w^{(l)}$ being the trainable parameters. As a result, gradients also scale with $1/\sqrt{N}$ and decrease for wider networks. In consequence, the weights $w^{(l)}$ stay close to their initial values. Furthermore, the scaling of the gradients can be translated to training on a rescaled loss $\bar{\mathcal{L}} = \mathcal{L}/\sqrt{N}$ with training dynamics being

$$\partial_t W = -\nabla_W \bar{\mathcal{L}}. \tag{B.22}$$

For infinitely-wide networks $N \to \infty$, the NTK is constant during training (Jacot, Gabriel, and Hongler, 2018), so we here focus on the NTK at initialization. Lee et al. (2018) show that the NTK at initialization corresponds to linearizing the network outputs f_α with respect to the initial weights θ_0, matching the assumption that the trained weights change only slightly from their initial values. We show that our theoretical framework reduces to the NTK under the linearization assumption. We substitute the network mapping as defined in Sec. 3.2 by

$$\begin{aligned} f(X|\theta) &\simeq f(X|\theta_0) + \nabla f(X|\theta_0)\,\omega \\ &=: f_0 + \nabla f\,\omega, \\ \omega &= \theta - \theta_0. \end{aligned} \tag{B.23}$$

Here, we denote all network weights as $\theta = \{w_{ij}^{(0)}....w_{ij}^{(L)}\}$ and ω measures the deviations of weights from their initial values. The inputs $X \in \mathbb{R}^{(P+1)\times N}$ consist of $1 \leq \alpha \leq P$ training points and a single test point $\alpha = *$. In the linearization, we have $\nabla f(X|\theta_0)\,\omega := \sum_{l,ij} \frac{\partial f(X|\theta)}{\partial w_{ij}^{(l)}}\Big|_{\theta_0} \omega_{ij}^{(l)} \in \mathbb{R}^{P+1}$, where $\nabla f \in \mathbb{R}^{(P+1)\times L\,N^2+N}$ denotes the Jacobi matrix with respect to the $L\,N^2 + N$ network weights.

The training dynamics in our framework (see App. B.5) differs from Eq. (B.22); it is given by

$$\begin{aligned} \partial_t\theta(t) &= -\gamma\theta(t) - \nabla\mathcal{L}(f(X,\theta(t));Y) + \sqrt{2T}\zeta(t), \\ \big\langle\zeta_i(t)\zeta_j(s)\big\rangle &= \delta_{ij}\delta(t-s). \end{aligned} \tag{B.24}$$

By adding a factor $\sqrt{N}$ to the time scale τ as well as setting both temperature T and weight decay γ to zero later, we can recover NTK training dynamics in Eq. (B.22) as

$$\begin{aligned} \tau\,\partial_t\theta(t) &= -\gamma\theta(t) - \nabla\mathcal{L}(f(X,\theta(t));Y) + \sqrt{2T\tau}\,\zeta(t), \\ \big\langle\zeta_i(t)\zeta_j(s)\big\rangle &= \delta_{ij}\delta(t-s). \end{aligned} \tag{B.25}$$

The equilibrium distribution is invariant to a change of the time scale; thus Eq. (B.24) yields the same equilibrium distribution as Eq. (B.25). We set $\gamma = 0$ directly, but keep T finite for the derivation and look at the limit $T \to 0$ only at the end. We recast Eq. (B.25) in terms of the linearized weights $\omega(t)$ and divide by τ, yielding

$$\partial_t\omega(t) = -\nabla_{\omega(t)}\,\bar{\mathcal{L}}(f_0 + \nabla f\,\omega(t)\,;\,Y) + \sqrt{2T/\tau}\,\zeta(t).$$

The equilibrium distribution of $\omega(t)$ is then given by

$$p_0(\omega|W_0) \propto \exp\big(-\frac{\tau}{T}\,\bar{\mathcal{L}}(f_0 + \nabla f\,\omega;\,Y)\big),$$

which for the squared loss function Eq. (B.21) is a Gaussian. Due to the linear dependence of the network outputs f_α on the linearized weights ω in Eq. (B.23), the joint distribution of all network outputs f and labels Y is also Gaussian, yielding for the network prior

$$\begin{aligned} p(Y,f|X,\theta_0) &\propto \int d\omega \,\exp\big(-\frac{\tau}{T}\,\bar{\mathcal{L}}(f;Y)\big)\,\delta(f-f_0-\nabla f\,\omega)) \\ &= \int d\omega\,\exp\big(-\frac{1}{T}\,\mathcal{L}(f;Y)\big)\,\delta(f-f_0-\nabla f\,\omega)). \end{aligned}$$

For conditioning the network prior on the training labels Y, we use the cumulant-generating function of the conditional $p(f|Y,X,\theta_0) \coloneqq p(Y,f|X,\theta_0)/\int df\,p(Y,f|X,\theta_0)$ with $j^\mathsf{T} f = \sum_{\alpha=1}^{P+1} j_\alpha f_\alpha$, yielding

$$\begin{aligned} &\mathcal{W}(j|Y,X,\theta_0) \qquad\qquad (B.26)\\ &= \ln \frac{\int df\,p(Y,f|X,\theta_0)\,e^{j^\mathsf{T} f}}{\int df\,p(Y,f|X,\theta_0)} \\ &= \ln \big\langle e^{j^\mathsf{T} f}\big\rangle_{f\sim p(Y,f|X,\theta_0)} + \text{const.} \\ &= \ln \int df \int d\omega\,\exp\big(j^\mathsf{T} f - \frac{1}{2T}\,\|Y-f\|_P^2\big)\,\delta\big(f-f_0-\nabla f\,\omega\big) + \text{const.} \\ &= \ln \int d\omega\,\exp\big(j^\mathsf{T}(f_0+\nabla f\,\omega) - \frac{1}{2T}\,\|Y-f_0-\nabla f\,\omega\|_P^2\big) + \text{const.} \\ &= \ln \int d\omega\,\exp\big(j^\mathsf{T}(f_0+\nabla f\,\omega) - \frac{1}{2T}\omega^\mathsf{T}\,\nabla f^\mathsf{T}\nabla f\,\omega + \frac{1}{T}\,(Y-f_0)^\mathsf{T}\,\nabla f\,\omega\big) + \text{const.} \\ &= \ln \int d\omega\,\exp\big(j^\mathsf{T} f_0 + \big(j^\mathsf{T}\nabla f + \frac{1}{T}\,(Y-f_0)^\mathsf{T}\nabla f\big)\,\omega - \frac{1}{2T}\omega^\mathsf{T}\,\nabla f^\mathsf{T}\nabla f\,\omega\big) + \text{const.} \\ &= j^\mathsf{T} f_0 + \frac{T}{2}\,\big(j^\mathsf{T}\nabla f + \frac{1}{T}\,(Y-f_0)^\mathsf{T}\,\nabla f\big)\,\big[\nabla f^\mathsf{T}\nabla f\big]^{-1}\big(j^\mathsf{T}\nabla f + \frac{1}{T}\,(Y-f_0)\,\nabla f\big)^\mathsf{T} + \text{const.}, \end{aligned}$$

where we drop all constant terms. Since the squared loss is computed only on the P training points, the norm $\|Y-f\|_P^2$ is with regard to these P training points and consequently all scalar products resulting from it; the test point appears only in $j^\mathsf{T}\nabla f$ and $j^\mathsf{T} f_0$. Since the cumulant-generating function is a polynomial of degree two, the posterior is Gaussian and we compute the mean for the test point $\alpha = *$ as

$$\mu_* = \frac{\partial}{\partial j_*}\mathcal{W}\Big|_{j=0} = f_{0,*} + \nabla f_*\,\big[\nabla f^\mathsf{T}\nabla f\big]^{-1}\,\nabla f^\mathsf{T}\,(Y-f_0). \qquad (B.27)$$

This result does not depend on T and thus the limit $T \to 0$ exists. We get the variance from the second derivative; it scales linearly with the temperature T and thus vanishes in the limit $T \to 0$. We rewrite Eq. (B.27) to obtain the NTK result by using the associativity of matrices X as

$$(X^\mathsf{T}\,X)\,X^\mathsf{T} = X^\mathsf{T}\,(X\,X^\mathsf{T});$$

multiplying with $(X^{\mathsf{T}}X)^{-1}$ from the left and $(XX^{\mathsf{T}})^{-1}$ from the right, we get

$$X^{\mathsf{T}}\,(X\,X^{\mathsf{T}})^{-1} = (X^{\mathsf{T}}\,X)^{-1}\,X^{\mathsf{T}}.$$

Thus, we can write the mean of the predictor as

$$\mu_* = \frac{\partial}{\partial j_*}\mathcal{W}\big|_{j=0} = f_{0,*} + \nabla f_* \nabla f^{\mathsf{T}}\left[\nabla f \nabla f^{\mathsf{T}}\right]^{-1}(Y - f_0), \tag{B.28}$$

where we recover the NTK kernel as

$$\Theta_{\alpha\beta} = [\nabla f \nabla f^{\mathsf{T}}]_{\alpha\beta} \equiv \sum_{l,ij} \frac{\partial f_\alpha}{\partial \theta_{ij}^{(l)}} \frac{\partial f_\beta}{\partial \theta_{ij}^{(l)}}. \tag{B.29}$$

Here, the matrix $\left[\nabla f \nabla f^{\mathsf{T}}\right]_{1\leq\alpha,\beta\leq P}$ goes over all P training points since it results from the norm $\|\dots\|_P^2$, whereas $[\nabla f_* \nabla f^{\mathsf{T}}]_{1\leq\beta\leq P}$ is a vector of dimension P. In (B.28) we recover the stationary point of the NTK predictor for linearized networks (cf. Eqs. (10)-(11) in Lee et al. (2018)).

In conclusion, we find that our framework yields the NTK under the assumption that the dependence of the network output on the network parameters is linear. This assumption holds if the network weights remain close to their initial values, as is the case for the NTK architecture in the limit of infinitely wide networks $N \to \infty$. Our approach is more general and does not require this assumption. While we here considered zero weight decay, we may extend these results to non-zero weight decay. Further, this derivation shows that for non-zero temperature T, we get the same mean predictor Eq. (B.28) but with a non-zero variance from Eq. (B.26).

Similar to the NNGP, a key difference to the NTK kernel Θ in Eq. (B.29) is that it depends solely on the network architecture and the data points X but not on the labels Y and thus does not capture the input-label dependence. In contrast, the feature-corrected kernels $C^{(l)}$ in our theoretical framework result from the interplay between the network prior and the likelihood of the labels in the data term $\mathcal{S}_D$ in Eq. (3.59) and thus depend on the joint statistics of X and Y. In consequence, the CKA between $C^{(L)}$ and YY^{T} increases, as shown explicitly in the expressions for linear networks in Eq. (B.19).

B.5 Langevin stochastic gradient descent

In order to compare our theoretical predicitions of the kernels to the network posterior that has been conditioned on the training data $X = (x_\alpha)_{\alpha=1,\dots,P}, Y = (y_\alpha)_{\alpha=1,\dots,P}$, we use Langevin stochastic gradient descent (LSGD) to train networks. Building on

Naveh et al. (2021), we let network parameters θ evolve as

$$\begin{aligned}\partial_t\theta(t) &= -\gamma\theta(t) - \nabla_\Theta\mathcal{L}(\theta(t);Y) + \sqrt{2T}\zeta(t), \\ \left\langle\zeta_i(t)\zeta_j(s)\right\rangle &= \delta_{ij}\delta(t-s),\end{aligned} \tag{B.30}$$

where $\mathcal{L}(\theta;Y) = \sum_{\alpha=1}^{P}(f_\alpha(\theta)-y_\alpha)^2$ denotes the squared error loss and $f_\alpha(\theta)$ the network output for sample α. This ensures that we sample from the equilibrium distribution for θ for large times t given by

$$\lim_{t\to\infty} p\left(\theta(t)\right) \sim \exp\left(-\frac{\gamma}{2T}\|\theta\|^2 - \frac{1}{T}\mathcal{L}(\theta;Y)\right), \tag{B.31}$$

which can be derived from the Fokker-Planck equation (Risken, 1996). By marginalizing over the posterior distribution of network parameters θ, we obtain the network posterior as

$$\begin{aligned}p(Y|X) &\propto \int d\theta\,\exp\left(-\frac{\gamma}{2T}\,\|\theta\|^2 - \frac{1}{T}\,\|f-Y\|^2\right) \\ &\propto \left\langle\exp\left(-\frac{1}{T}\,\|f-Y\|^2\right)\right\rangle_{\Theta_k\overset{\text{i.i.d.}}{\sim}\mathcal{N}(0,T/\gamma)} \\ &\propto \mathcal{N}(Y|f,T/2)\,\langle\delta\big[f-f(\theta)\big]\rangle_{\Theta_k\overset{\text{i.i.d.}}{\sim}\mathcal{N}(0,T/\gamma)}.\end{aligned} \tag{B.32}$$

By inserting $p(f|X) \equiv \langle\delta\big[f-f(\theta)\big]\rangle_{\theta_k\overset{\text{i.i.d.}}{\sim}\mathcal{N}(0,T/\gamma)}$ and identifying $\kappa = T/2$ with the regularization noise and $T/\gamma = \sigma_w^2/N$ with the variance of the parameter θ_k, it follows that the posterior is the same as Eq. (3.27). To account for different initialization variances for different parameters θ_k, we use a different weight decay γ_k for each parameter θ_k.

For network training, we use the time discrete version of Eq. (B.30)as

$$\begin{aligned}\theta_t &= \theta_{t-1} - \eta\left(\gamma\theta_{t-1} + \nabla_\theta\mathcal{L}(\theta_{t-1};Y)\right) + \sqrt{2T\eta}\,\zeta_t, \\ \langle\zeta_t\zeta_s\rangle &= \delta_{ts},\end{aligned} \tag{B.33}$$

where ζ_t is standard normally distributed. The time step η can be interpreted as a learning rate and needs to be sufficiently small in order to match the differential equation in Eq. (B.30). The regularizer κ calibrates the trade-off between the network prior and the data term in the loss term: In network training, larger κ leads to large $T = 2\kappa$, which corresponds to more noise in LSGD, and thus favors the parameter priors. Meanwhile, smaller κ emphasizes the loss term in the exponent.

Overall, Langevin stochastic gradient descent performs full-batch gradient descent with additional i.i.d. standard normal noise on the gradients and weight decay regularization (Krogh and Hertz, 1991). To ensure that we sample from the Bayesian network posterior when training with LSGD, we require that the system has fully relaxed and that drawn samples are uncorrelated: we use an initial warmup of $50{,}000$ training steps and measure sample every $1{,}000$ time steps.

B.6 Additional details on numerical evaluation of theory

B.6.1 Weight variance of the input layer

We choose the weight variance of the input layer $\sigma_{w,0}^2$ such that the diagonal elements of the network kernels $C_{\alpha\alpha}^{(l)}$ are at their fixed point value for large depth according to Schoenholz et al. (2017). Then the response functions of the networks relax only on one depth scale determined by the off-diagonal kernel elements.

B.6.2 Gaussian integrals

The self-consistency equations in Eq. (3.62) involve several two-point and four-point Gaussian integrals. For ϕ = erf, we derive analytical expressions for these two-point integrals as

$$\left\langle \phi(h_\alpha)\phi(h_\beta)\right\rangle_{h\sim\mathcal{N}(0,C)} = \begin{cases} \frac{4}{\pi}\arctan\left(\sqrt{1+4C_{\alpha\alpha}}\right)-1 & \alpha=\beta, \\ \frac{2}{\pi}\arcsin\left(\frac{2C_{\alpha\beta}}{\sqrt{1+2C_{\alpha\alpha}}\sqrt{1+2C_{\beta\beta}}}\right) & \text{else,} \end{cases}$$

$$\left\langle \phi'(h_\alpha)\phi'(h_\beta)\right\rangle_{h\sim\mathcal{N}(0,C)} = \begin{cases} \frac{4}{\pi}\frac{1}{\sqrt{4C_{\alpha\alpha}+1}} & \alpha=\beta, \\ \frac{4}{\pi}\left(2\left(C_{\alpha\alpha}+C_{\beta\beta}\right)+1+4\left(C_{\alpha\alpha}C_{\beta\beta}-C_{\alpha\beta}^2\right)\right)^{-0.5} & \text{else,} \end{cases}$$

$$\left\langle \phi(h_\alpha)\phi''(h_\beta)\right\rangle_{h\sim\mathcal{N}(0,C)} = \begin{cases} -\frac{8}{\pi}\frac{C_{\alpha\alpha}}{(2C_{\alpha\alpha}+1)\sqrt{4C_{\alpha\alpha}+1}} & \alpha=\beta, \\ -\frac{8}{\pi}\frac{C_{\beta\alpha}}{(2C_{\alpha\alpha}+1)\sqrt{2\left(C_{\alpha\alpha}+C_{\beta\beta}\right)+1+4\left(C_{\alpha\alpha}C_{\beta\beta}-C_{\alpha\beta}^2\right)}} & \text{else.} \end{cases}$$

To our knowledge, there exists no analytical solution for the four-point integral $\left\langle \phi(h_\alpha)\phi(h_\beta)\phi(h_\gamma)\phi(h_\delta)\right\rangle_{h\sim\mathcal{N}(0,C)}$; instead we calculate this integral numerically using Monte-Carlo sampling with $n_{MC}=10^5$ samples.

B.6.3 Annealing in network width

We solve the self-consistency equations for the posterior kernels in Eq. (3.62) iteratively: (i) Set $C^{(0)}$ by Eq. (3.26) and $\tilde{C}=0$ initially. (ii) Propagate Eq. (3.66) forward until $C^{(L)}$; in the first iteration this yields the NNGP. (iii) Calculate $\tilde{C}^{(L)}$ from Eq. (3.60). (iv) Propagate $\tilde{C}$ backward with Eq. (3.62) (using the Gaussian measure $\langle\ldots\rangle_{\mathcal{N}(0,C^{(l)})}$). (v) Iterate from step (ii) with $\tilde{C}\neq 0$ until convergence. For stability of this iteration scheme, we damp the solution from each new iteration i by a factor $\gamma=0.5$ as

$$\begin{aligned} C^{(l),i} &\mapsto (1-\gamma)C^{(l),i+1}+\gamma C^{(l),i}, \\ \tilde{C}^{(l),i} &\mapsto (1-\gamma)\tilde{C}^{(l),i+1}+\gamma\tilde{C}^{(l),i}. \end{aligned}$$

Algorithm 1 Width annealing of kernels

```
**Input:** data $X$, labels $Y$, network widths $\{N_i\}_i$
Compute NNGP kernel $C^{(l)}_{\mathrm{NNGP}}$ from data $X$.
Set start values to NNGP kernel $C^{(l)}_{\mathrm{init}} = C^{(l)}_{\mathrm{NNGP}}$ and $\tilde{C}^{(l)}_{\mathrm{init}} = 0$.
**for** $N$ **in** $\{N_i\}_i$ **do**
    Compute corrected kernels $C^{(l)}_{\mathrm{corr}} = f(C^{(l)}_{\mathrm{init}}, \tilde{C}^{(l)}_{\mathrm{init}}, Y, N)$ and conjugate kernels $\tilde{C}^{(l)}_{\mathrm{corr}} = g(C^{(l)}_{\mathrm{init}}, \tilde{C}^{(l)}_{\mathrm{init}}, Y, N)$.
    Reset start values $C^{(l)}_{\mathrm{init}} = C^{(l)}_{\mathrm{corr}}$ and $\tilde{C}^{(l)}_{\mathrm{init}} = \tilde{C}^{(l)}_{\mathrm{corr}}$.
**end for**
```

To solve these equations iteratively, we begin with the NNGP kernel as the initial value. For wide networks and in the limit of negligible training load $\nu = P/N \to 0$, the corrections to the NNGP kernel remain small, allowing the posterior kernels to be accurately described by including corrections to linear order in the conjugate kernels. For smaller network widths, we take advantage of the fact that corrections are small when solving for the posterior kernels based on the posterior kernels of a slightly wider network: Specifically, we start from wide networks and compute the corrections to the NNGP kernel. Using these corrected kernels as the new starting point, we then compute corrections for slightly narrower networks. This process is repeated iteratively, progressively narrowing the network width until we reach the desired width (see pseudocode in Alg. 1).

In Fig. B.2, we illustrate width annealing by measuring the CKA between the output kernel $C^{(L)}$ and target kernel YY^{T} relative to the NNGP kernel for decreasing network width N. Feature corrections continuously increase for narrower networks.

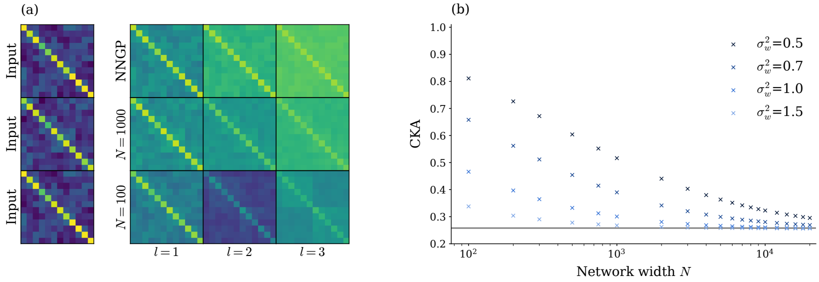


Figure B.2: Solving the self-consistency equations in Eq. (3.62) by annealing solutions in the network width. (a) Network kernel $C^{(l)}$ across layers $l = 1, 2, 3$ for different network widths N and $\sigma_w^2 = 0.5$. Narrower networks exhibit stronger adaptation to the target kernel YY^T. (b) CKA between network kernels $C^{(L)}$ and target kernel YY^T annealed in the network width N. The CKA (blue markers) is close to that of the NNGP (solid line) for wide networks, increasing continuously towards the target kernel for narrower networks. Feature corrections depend on network hyperparameters such as the weight variance σ_w^2 (increasing from dark to light). Other parameters: XOR task with $\sigma_{\mathrm{XOR}}^2 = 0.4$, $\sigma_w^2 \in \{0.5, 0.7, 1.0, 1.5\}$, $\sigma_b^2 = 0.05$, $L = 3$, $\kappa = 10^{-3}$, $P = 12$.

Appendix C

Field theory for optimal signal propagation in residual networks

C.1 Maximum entropy condition for optimal scaling

We present an alternative approach to determining the condition for optimal signal propagation in Sec. 4.5: In their work on trainability in feed-forward networks, Bukva et al. (2023) conjecture that networks are more expressive if hidden signal distributions are approximately uniform, making them maximally entropic.

For networks of large but finite width, the signal distribution of each neuron is identical and can approximately be described as a Gaussian

$$p(h;\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}h^2\right). \tag{C.1}$$

Then, the distribution of the post-activation $z = \phi(h)$ can be written as

$$p(z;\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}\phi'(\phi^{-1}(x))} \exp\left(-\frac{1}{2\sigma^2}\phi^{-1}(z)^2\right). \tag{C.2}$$

For $\phi = \mathrm{erf}$, the dynamic range is roughly given by $[-1, 1]$. For this intervall, we determine the Kullback-Leibler divergence between the distribution of the post-activation and a uniform distribution

$$\begin{aligned}
D_{\mathrm{KL}}(p_{\mathrm{uni}}|p_\phi) &= \int_{-1}^{1} \mathrm{d}z\, p_{\mathrm{uni}}(z)\, [\ln p_{\mathrm{uni}}(z) - \ln p_\phi(z)] \\
&= \int_{-1}^{1} \mathrm{d}z\, \frac{1}{2} \ln\left(\frac{1}{2}\right) + \frac{1}{2}\frac{1}{2\sigma^2}\phi^{-1}(z)^2 + \frac{1}{2}\ln\left(\sqrt{2\pi}\sigma\phi'(\phi^{-1}(z))\right) \\
&= \ln\left(\frac{1}{2}\right) + \ln(\sqrt{8}\sigma) + \frac{1}{2}\int_{-1}^{1} \mathrm{d}z\, \left(\frac{1}{2\sigma^2} - 1\right)\phi^{-1}(z)^2 \\
&= \ln\left(\sqrt{2}\right) + \frac{1}{2}\ln(\sigma^2) + \frac{1}{2}\left(\frac{1}{2\sigma^2} - 1\right)\int_{-\infty}^{\infty} \mathrm{d}h\, h^2 \frac{2}{\sqrt{\pi}}\exp(-h^2) \\
&= \ln\left(\sqrt{2}\right) + \frac{1}{2}\ln(\sigma^2) + \frac{1}{2}\left(\frac{1}{2\sigma^2} - 1\right).
\end{aligned}$$

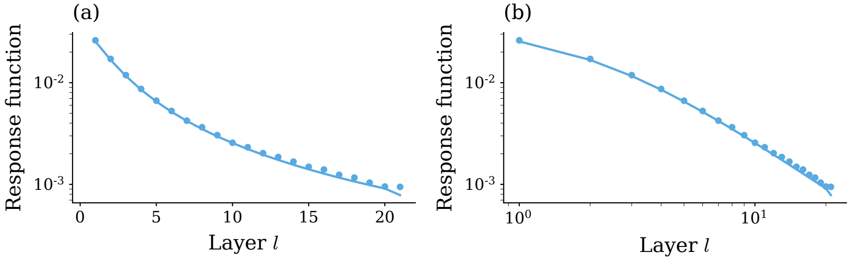


Figure C.1: (a) Log-plot and (b) log-log-plot of the response function $\eta^{(l)}$ for a residual network of depth $L = 20$. Theory (curve) and simulation (dots) agree well. Simulation values are averaged over 10^2 input samples and 10^3 network initializations. The response function decays sub-exponentially (a), showing power law behavior in later layers (b). Other parameters: $\sigma^2_{w,\mathrm{in}} = \sigma^2_w = \sigma^2_{w,\mathrm{out}} = 1.2$, $\sigma^2_{b,\mathrm{in}} = \sigma^2_b = \sigma^2_{b,\mathrm{out}} = 0.2$, $D_\mathrm{in} = D_\mathrm{out} = 100$, $N = 500$, $\rho = 1$.

Determining the maximum of the Kullback-Leibler divergence yields the following condition for the first derivative:

$$0 \overset{!}{=} \frac{\partial}{\partial \sigma^2} D_\mathrm{KL}(p_\mathrm{uni}|p_\phi) = \frac{1}{\sigma^2} - \frac{1}{4}\frac{1}{\sigma^4}.$$

Solving for the signal variance before the readout layer yields $\sigma^2 \overset{!}{=} 1/4$, matching the condition in Sec. 4.5.

C.2 Supplementary figures

C.2.1 Sub-exponential decay of response function

The response function in residual networks exhibits a sub-exponential decay (see Fig. C.1), as noted in Yang and Schoenholz (2017) using a different approach.

C.2.2 Normalized input kernels

We investigate signal propagation and scaling behavior in residual networks across various tasks in Sec. 4.5. For reference, we display in Fig. C.2 the normalized overlap kernels $\frac{1}{\max_{\alpha\beta} x_\alpha \cdot x_\beta} X^\mathsf{T} X$ for P data samples related to these tasks. For the MNIST data set, we focus on binary classification between 0 and 3, with an equal number of samples from both classes ($P_0 = P_3 = \frac{1}{2}P$). For the CIFAR-10 data set, we focus on binary classification between airplanes and dogs, also with an equal number of samples from both classes ($P_{\text{airplane}} = P_{\text{dog}} = \frac{1}{2}P$).

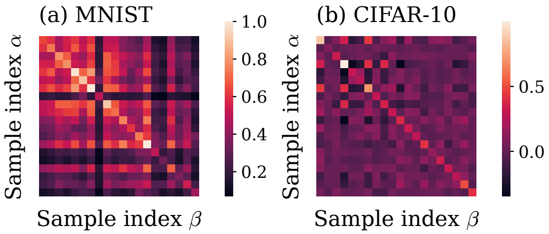


Figure C.2: Input kernels of different tasks for $P = 20$ samples, normalized over all kernel elements. We look at binary classification tasks on the common data sets (a) MNIST and (b) CIFAR-10 with equal number of samples for both classes.

Bibliography

Aas, Kjersti, Martin Jullum, and Anders Løland (2021). "Explaining individual predictions when features are dependent: More accurate approximations to Shapley values." In: *Artificial Intelligence* 298, p. 103502. URL: https://www.sciencedirect.com/science/article/pii/S0004370221000539.

Abramson, Josh et al. (2024). "Accurate structure prediction of biomolecular interactions with AlphaFold 3." In: *Nature* 630.8016, pp. 493–500. URL: https://www.nature.com/articles/s41586-024-07487-w.

Advani, Madhu S., Andrew M. Saxe, and Haim Sompolinsky (2020). "High-dimensional dynamics of generalization error in neural networks." In: *Neural Networks* 132, pp. 428–446.

Aitken, Kyle and Guy Gur-Ari (2020). *On the asymptotics of wide networks with polynomial activations*. URL: https://arxiv.org/abs/2006.06687.

Aiudi, R. et al. (2023). "Local Kernel Renormalization as a mechanism for feature learning in overparametrized Convolutional Neural Networks." In: arXiv:2307.11807 [cs.LG]. URL: https://arxiv.org/abs/2307.11807.

Al Kuwaiti, Ahmed et al. (2023). "A Review of the Role of Artificial Intelligence in Healthcare." In: *Journal of Personalized Medicine* 13.6. URL: https://www.mdpi.com/2075-4426/13/6/951.

Antognini, Joseph M (2019). "Finite size corrections for neural network Gaussian processes." In: *ArXiv*, 1908.10030 [cs.LG]. URL: https://arxiv.org/abs/1908.10030.

Ardizzone, Lynton et al. (2019). "Guided Image Generation with Conditional Invertible Neural Networks." In: arXiv:1907.02392 [cs.CV]. URL: https://arxiv.org/abs/1907.02392.

Arpit, Devansh, Víctor Campos, and Yoshua Bengio (2019). "How to Initialize your Network? Robust Initialization for WeightNorm &amp; ResNets." In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/e520f70ac3930490458892665cda6620-Paper.pdf.

Bachlechner, Thomas et al. (2021). "ReZero is all you need: fast convergence at large depth." In: *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*. Ed. by Cassio de Campos and Marloes H. Maathuis. Vol. 161. Proceedings of Machine Learning Research. PMLR, pp. 1352–1361. URL: https://proceedings.mlr.press/v161/bachlechner21a.html.

Baglioni, P. et al. (2024). “Predictive Power of a Bayesian Effective Action for Fully Connected One Hidden Layer Neural Networks in the Proportional Limit.” In: *Physical Review Letters* 133 (2), p. 027301. URL: https://link.aps.org/doi/10.1103/PhysRevLett.133.027301.

Baldassi, Carlo et al. (2022). “Learning through atypical phase transitions in overparameterized neural networks.” In: *Physical Review E* 106 (1), p. 014116. URL: https://link.aps.org/doi/10.1103/PhysRevE.106.014116.

Barzilai, Daniel et al. (2023). “A Kernel Perspective of Skip Connections in Convolutional Networks.” In: *The Eleventh International Conference on Learning Representations*. URL: https://openreview.net/forum?id=6H_uOfcwiVh.

Belrose, Nora et al. (2024). “Neural Networks Learn Statistics of Increasing Complexity.” In: *Proceedings of the 41st International Conference on Machine Learning*. Ed. by Ruslan Salakhutdinov et al. Vol. 235. Proceedings of Machine Learning Research. PMLR, pp. 3382–3409. URL: https://proceedings.mlr.press/v235/belrose24a.html.

Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Blinnikov, S. and R. Moessner (May 1998). “Expansions for nearly Gaussian distributions.” In: *Astron. Astrophys. Suppl. Ser.* 130.1, pp. 193–205. URL: http://aas.aanda.org/10.1051/aas:1998221.

Boltzmann, Ludwig (1964). *Lectures on Gas Theory*. Ed. by Stephen G. Brush. Berkeley: University of California Press. URL: https://doi.org/10.1525/9780520327474.

Bommasani, Rishi et al. (July 2022). “On the Opportunities and Risks of Foundation Models.” In: arXiv:2108.07258. arXiv:2108.07258 [cs]. URL: http://arxiv.org/abs/2108.07258.

Bordelon, Blake and Cengiz Pehlevan (2023). “Self-consistent dynamical field theory of kernel evolution in wide neural networks*.” In: *Journal of Statistical Mechanics: Theory and Experiment* 2023.11, p. 114009. URL: https://dx.doi.org/10.1088/1742-5468/ad01b0.

Bordelon, Blake et al. (2024). “Depthwise Hyperparameter Transfer in Residual Networks: Dynamics and Scaling Limit.” In: *The Twelfth International Conference on Learning Representations*. URL: https://openreview.net/forum?id=KZJehvRKGD.

Bowman, Benjamin and Guido Montufar (2022). “Implicit Bias of MSE Gradient Optimization in Underparameterized Neural Networks.” In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=VLgmhQDVBV.

Bukva, Aleksandar et al. (2023). “Criticality versus uniformity in deep neural networks.” In: arXiv:2304.04784. arXiv:2304.04784 [cs, stat]. URL: http://arxiv.org/abs/2304.04784.

Canatar, Abdulkadir and Cengiz Pehlevan (2022). “A Kernel Analysis of Feature Learning in Deep Neural Networks.” In: *2022 58th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1–8. URL: https://doi.org/10.1109/Allerton49937.2022.9929375.

Chen, Ricky TQ et al. (2018). “Neural ordinary differential equations.” In: *Advances in neural information processing systems*, pp. 6571–6583. URL: https://proceedings.

neurips.cc/paper_files/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf.

Chizat, Lenaic, Edouard Oyallon, and Francis Bach (2019). "On lazy training in differentiable programming." In: *Advances in Neural Information Processing Systems*. Vol. 32. URL: https://openreview.net/pdf?id=rkgxDVSlLB.

Cohen, Alain-Sam et al. (2021). "Scaling Properties of Deep Residual Networks." In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, pp. 2039–2048. URL: https://proceedings.mlr.press/v139/cohen21b.html.

Cohen, Khen, Noam Levi, and Yaron Oz (2024). "Classifying Overlapping Gaussian Mixtures in High Dimensions: From Optimal Classifiers to Neural Nets." In: arXiv:2405.18427 [stat.ML]. URL: https://arxiv.org/abs/2405.18427.

Cohen, Omry, Or Malka, and Zohar Ringel (2021). "Learning curves for overparametrized deep neural networks: A field theory perspective." In: *Physical Review Research* 3 (2), p. 023034. URL: https://link.aps.org/doi/10.1103/PhysRevResearch.3.023034.

Cortes, Corinna, Mehryar Mohri, and Afshin Rostamizadeh (2012). "Algorithms for Learning Kernels Based on Centered Alignment." In: *Journal of Machine Learning Research* 13.28, pp. 795–828. URL: http://jmlr.org/papers/v13/cortes12a.html.

Crisanti, A. and H. Sompolinsky (2018). "Path integral approach to random neural networks." In: *Physical Review E* 98 (6), p. 062120. URL: https://link.aps.org/doi/10.1103/PhysRevE.98.062120.

Cui, Hugo et al. (2024). "A phase transition between positional and semantic learning in a solvable model of dot-product attention." In: arXiv:2402.03902 [cs.LG]. URL: https://arxiv.org/abs/2402.03902.

Cybenko, G. (1989). "Approximation by superpositions of a sigmoidal function." In: *Mathematics of Control, Signals, and Systems* 2.4, pp. 303–314. URL: https://doi.org/10.1007/bf02551274.

Deco, Gustavo and Wilfried Brauer (Jan. 1994). "Higher order statistical decorrelation without information loss." In: *Proceedings of the 7th International Conference on Neural Information Processing Systems*. NIPS'94. Cambridge, MA, USA: MIT Press, pp. 247–254. URL: https://proceedings.neurips.cc/paper/1994/hash/892c91e0a653ba19df81a90f89d99bcd-Abstract.html.

Doshi, Darshil, Tianyu He, and Andrey Gromov (2023). "Critical Initialization of Wide and Deep Neural Networks using Partial Jacobians: General Theory and Applications." In: *Advances in Neural Information Processing Systems*. Ed. by A. Oh et al. Vol. 36. Curran Associates, Inc., pp. 40054–40095. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/7e02f2910ea7911a37c4691f4201c878-Paper-Conference.pdf.

Dyer, Ethan and Guy Gur-Ari (2020). "Asymptotics of Wide Networks from Feynman Diagrams." In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=S1gFvANKDS.

Falkner, Stefan, Aaron Klein, and Frank Hutter (2018). "BOHB: Robust and Efficient Hyperparameter Optimization at Scale." In: *Proceedings of the 35th International Con-*

ference on Machine Learning. Vol. 80. Proceedings of Machine Learning Research. PMLR, pp. 1437–1446. URL: https://proceedings.mlr.press/v80/falkner18a.html.

Fischer, Kirsten (2020). "Decomposition of deep neural networks into correlation functions." Master thesis. RWTH Aachen University.

Fischer, Kirsten, David Dahmen, and Moritz Helias (2023). "Optimal signal propagation in ResNets through residual scaling." In: arXiv:2305.07715 [cond-mat.dis-nn]. URL: https://arxiv.org/abs/2305.07715.

Fischer, Kirsten et al. (2022). "Decomposing neural networks as mappings of correlation functions." In: *Physical Review Research* 4.4, p. 043143. URL: https://link.aps.org/doi/10.1103/PhysRevResearch.4.043143.

Fischer, Kirsten et al. (2024). "Critical feature learning in deep neural networks." In: *Proceedings of the 41st International Conference on Machine Learning*. Ed. by Ruslan Salakhutdinov et al. Vol. 235. Proceedings of Machine Learning Research. PMLR, pp. 13660–13690. URL: https://proceedings.mlr.press/v235/fischer24a.html.

Garriga-Alonso, Adrià, Carl Edward Rasmussen, and Laurence Aitchison (2019). "Deep Convolutional Networks as shallow Gaussian Processes." In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=Bklfsi0cKm.

Geiger, Mario et al. (2019). "Jamming transition as a paradigm to understand the loss landscape of deep neural networks." In: *Phys. Rev. E* 100 (1), p. 012115. URL: https://link.aps.org/doi/10.1103/PhysRevE.100.012115.

Geiger, Mario et al. (2020). "Disentangling feature and lazy training in deep neural networks." In: *Journal of Statistical Mechanics: Theory and Experiment* 2020.11, p. 113301.

Goldt, Sebastian et al. (2020). "Modeling the Influence of Data Structure on Learning in Neural Networks: The Hidden Manifold Model." In: *Physical Review X* 10.4, p. 041044. URL: https://link.aps.org/doi/10.1103/PhysRevX.10.041044.

Goldt, Sebastian et al. (2022). "The Gaussian equivalence of generative models for learning with shallow neural networks." In: *Proceedings of the 2nd Mathematical and Scientific Machine Learning Conference*. Ed. by Joan Bruna, Jan Hesthaven, and Lenka Zdeborova. Vol. 145. Proceedings of Machine Learning Research. PMLR, pp. 426–471. URL: https://proceedings.mlr.press/v145/goldt22a.html.

Gunning, David et al. (2019). "XAI — Explainable artificial intelligence." In: *Science Robotics* 4.37, eaay7120. URL: https://www.science.org/doi/abs/10.1126/scirobotics.aay7120.

Halverson, James, Anindita Maiti, and Keegan Stoner (2021). "Neural networks and quantum field theory." In: *Machine Learning: Science and Technology* 2.3, p. 035002. URL: https://doi.org/10.1088/2632-2153/abeca3.

Hanin, Boris (2018). "Which Neural Net Architectures Give Rise to Exploding and Vanishing Gradients?" In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/13f9896df61279c928f19721878fac41-Paper.pdf.

Hanin, Boris and David Rolnick (2018). "How to Start Training: The Effect of Initialization and Architecture." In: *Advances in Neural Information Processing Systems*. Ed.

by S. Bengio et al. Vol. 31. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/d81f9c1be2e08964bf9f24b15f0e4900-Paper.pdf.

Hanin, Boris and Alexander Zlokapa (2023). "Bayesian interpolation with deep linear networks." In: *Proceedings of the National Academy of Sciences of the United States of America* 120.23, e2301345120. URL: https://www.pnas.org/doi/abs/10.1073/pnas.2301345120.

Hayou, Soufiane et al. (2021a). "Robust Pruning at Initialization." In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=vXj_ucZQ4hA.

Hayou, Soufiane et al. (2021b). "Stable ResNet." In: *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*. Ed. by Arindam Banerjee and Kenji Fukumizu. Vol. 130. Proceedings of Machine Learning Research. PMLR, pp. 1324–1332. URL: https://proceedings.mlr.press/v130/hayou21a.html.

He, Kaiming et al. (2016). "Deep Residual Learning for Image Recognition." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. URL: https://ieeexplore.ieee.org/document/7780459.

Helias, Moritz and David Dahmen (2020). *Statistical Field Theory for Neural Networks*. Springer International Publishing, p. 203.

Hertz, John, Anders Krogh, and Richard G. Palmer (1991). *Introduction to the Theory of Neural Computation*. Cambridge, MA, USA: Perseus Books.

Hofmarcher, Markus et al. (2019). "Visual Scene Understanding for Autonomous Driving Using Semantic Segmentation." In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer International Publishing, pp. 285–296. URL: https://doi.org/10.1007/978-3-030-28954-6_15.

Huang, Jiaoyang and Horng-Tzer Yau (2020). "Dynamics of Deep Neural Networks and Neural Tangent Hierarchy." In: *Proceedings of the 37th International Conference on Machine Learning*. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 4542–4551. URL: https://proceedings.mlr.press/v119/huang20l.html.

Huang, Kaixuan et al. (2020). "Why Do Deep Residual Networks Generalize Better than Deep Feedforward Networks? — A Neural Tangent Kernel Perspective." In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., pp. 2698–2709. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1c336b8080f82bcc2cd2499b4c57261d-Paper.pdf.

Ingrosso, Alessandro and Sebastian Goldt (2022). "Data-driven emergence of convolutional structure in neural networks." In: *Proceedings of the National Academy of Sciences* 119.40, e2201854119. URL: https://www.pnas.org/doi/abs/10.1073/pnas.2201854119.

Ingrosso, Alessandro et al. (2024). "Statistical mechanics of transfer learning in fully-connected networks in the proportional limit." In: arXiv:2407.07168 [cond-mat.dis-nn]. URL: https://arxiv.org/abs/2407.07168.

Ioffe, Sergey and Christian Szegedy (2015). "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift." In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei.

Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, pp. 448–456. URL: https://proceedings.mlr.press/v37/ioffe15.html.

Itano, Fernando, Miguel Angelo de Abreu de Sousa, and Emilio Del-Moral-Hernandez (2018). "Extending MLP ANN hyper-parameters Optimization by using Genetic Algorithm." In: *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. URL: https://doi.org/10.1109/IJCNN.2018.8489520.

Jacot, Arthur, Franck Gabriel, and Clément Hongler (2018). "Neural Tangent Kernel: Convergence and Generalization in Neural Networks." In: *Advances in Neural Information Processing Systems 31*, pp. 8580–8589. URL: https://proceedings.neurips.cc/paper/2018/file/5a4be1fa34e62bb8a6ec6b91d2462f5a-Paper.pdf.

Ji, Ziwei et al. (2023). "Survey of Hallucination in Natural Language Generation." In: *ACM Comput. Surv.* 55.12. URL: https://doi.org/10.1145/3571730.

Kalimeris, Dimitris et al. (2019). "SGD on Neural Networks Learns Functions of Increasing Complexity." In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/b432f34c5a997c8e7c806a895ecc5e25-Paper.pdf.

Kingma, Diederik P and Jimmy Lei Ba (2015). "Adam: A method for stochastic gradient descent." In: *ICLR: International Conference on Learning Representations*, pp. 1–15.

Kohavi, Ron and David H. Wolpert (1996). "Bias Plus Variance Decomposition for Zero-One Loss Functions." In: *Proceedings of the Thirteenth International Conference on Machine Learning*. Vol. 96, pp. 275–283.

Kora, Padmavathi et al. (2022). "Transfer learning techniques for medical image analysis: A review." In: *Biocybernetics and Biomedical Engineering* 42.1, pp. 79–107. URL: https://www.sciencedirect.com/science/article/pii/S0208521621001297.

Krizhevsky, Alex and Geoffrey Hinton (2009). *Learning multiple layers of features from tiny images*. Tech. rep.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "Imagenet classification with deep convolutional neural networks." In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., pp. 1097–1105. URL: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Krogh, A and J A Hertz (1992). "Generalization in a linear perceptron in the presence of noise." In: *Journal of Physics A: Mathematical and General* 25.5, p. 1135. URL: https://dx.doi.org/10.1088/0305-4470/25/5/020.

Krogh, Anders and John Hertz (1991). "A Simple Weight Decay Can Improve Generalization." In: *Advances in Neural Information Processing Systems*. Ed. by J. Moody, S. Hanson, and R.P. Lippmann. Vol. 4. Morgan-Kaufmann. URL: https://proceedings.neurips.cc/paper_files/paper/1991/file/8eefcfdf5990e441f0fb6f3fad709e21-Paper.pdf.

LeCun, Yann, Corinna Cortes, and Christopher JC Burges (1998). *The MNIST database of handwritten digits*.

Lee, Jaehoon et al. (2018). "Deep Neural Networks as Gaussian Processes." In: *International Conference on Learning Representations*. Vancouver: OpenReview.net. URL: https://openreview.net/forum?id=B1EA-M-0Z.

Lee, Jaehoon et al. (2019). "Wide Neural Networks of Any Depth Evolve as Linear Models Under Gradient Descent." In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/0d1a9651497a38d8b1c3871c84528bd4-Paper.pdf.

Lee, Jaehoon et al. (2020). "Finite Versus Infinite Neural Networks: an Empirical Study." In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., pp. 15156–15172. URL: https://proceedings.neurips.cc/paper/2020/file/ad086f59924fffe0773f8d0ca22ea712-Paper.pdf.

Leshno, Moshe et al. (1993). "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function." In: *Neural Networks* 6.6, pp. 861–867. URL: https://www.sciencedirect.com/science/article/pii/S0893608005801315.

Li, Qianyi and Haim Sompolinsky (2021). "Statistical Mechanics of Deep Linear Neural Networks: The Backpropagating Kernel Renormalization." In: *Physical Review X* 11.3, p. 031059. URL: https://journals.aps.org/prx/abstract/10.1103/PhysRevX.11.031059.

Lin, Henry W., Max Tegmark, and David Rolnick (2017). "Why Does Deep and Cheap Learning Work So Well?" In: *Journal of Statistical Physics* 168.6, pp. 1223–1247. URL: https://doi.org/10.1007/s10955-017-1836-5.

Lindner, Javed et al. (2023). "A theory of data variability in Neural Network Bayesian inference." In: *arXiv preprint arXiv:2307.16695*. URL: https://arxiv.org/abs/2307.16695.

Loshchilov, Ilya and Frank Hutter (2019). "Decoupled Weight Decay Regularization." In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=Bkg6RiCqY7.

Loureiro, Bruno et al. (2022). "Learning curves of generic features maps for realistic datasets with a teacher-student model." In: *Journal of Statistical Mechanics: Theory and Experiment* 2022.11, p. 114001. URL: https://dx.doi.org/10.1088/1742-5468/ac9825.

MacKay, David JC (2003). *Information theory, inference and learning algorithms*. Cambridge university press.

Marion, Pierre et al. (2024). "Scaling ResNets in the Large-depth Regime." In: arXiv:2206.06929 [cs.LG]. URL: https://arxiv.org/abs/2206.06929.

Meegen, Alexander van and Haim Sompolinsky (2024). "Coding schemes in neural networks learning classification tasks." In: arXiv:2406.16689 [cs.LG]. URL: https://arxiv.org/abs/2406.16689.

Mei, Song and Andrea Montanari (2022). "The Generalization Error of Random Features Regression: Precise Asymptotics and the Double Descent Curve." In: *Commun. pure appl. math.* 75.4, pp. 667–766. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.22008.

Mei, Song, Andrea Montanari, and Phan-Minh Nguyen (2018). "A mean field view of the landscape of two-layer neural networks." In: *Proceedings of the National Academy of Sciences* 115.33, E7665–E7671. URL: https://www.pnas.org/doi/abs/10.1073/pnas.1806579115.

Merger, Claudia et al. (2023). "Learning Interacting Theories from Data." In: *Phys. Rev. X* 13 (4), p. 041033. URL: https://link.aps.org/doi/10.1103/PhysRevX.13.041033.

Mézard, M., Giorgio Parisi, and M. Virasoro (1987). *Spin Glass Theory and Beyond (World Scientific Lecture Notes in Physics, Vol 9)*. World Scientific Publishing Company. URL: http://www.worldcat.org/isbn/9971501163.

Mignacco, Francesca, Pierfrancesco Urbani, and Lenka Zdeborová (2021). "Stochasticity helps to navigate rough landscapes: comparing gradient-descent-based algorithms in the phase retrieval problem." In: *Machine Learning: Science and Technology* 2.3, p. 035029. URL: https://doi.org/10.1088/2632-2153/ac0615.

Molgedey, L, J Schuchhardt, and HG Schuster (1992). "Suppressing chaos in neural networks by noise." In: *Physical Review Letters* 69.26, p. 3717. URL: https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.69.3717.

Montavon, Grégoire et al. (2017). "Explaining nonlinear classification decisions with deep Taylor decomposition." In: *Pattern Recognition* 65, pp. 211–222. URL: https://www.sciencedirect.com/science/article/pii/S0031320316303582.

Naveh, Gadi et al. (2021). "Predicting the outputs of finite deep neural networks trained with noisy gradients." In: *Physical Review E* 104 (6), p. 064301. URL: https://link.aps.org/doi/10.1103/PhysRevE.104.064301.

Neal, Radford M. (1996). *Bayesian Learning for Neural Networks*. Springer New York. URL: https://doi.org/10.1007/978-1-4612-0745-0.

Nezhurina, Marianna et al. (2024). "Alice in Wonderland: Simple Tasks Showing Complete Reasoning Breakdown in State-Of-the-Art Large Language Models." In: arXiv:2406.02061 [cs.LG]. URL: https://arxiv.org/abs/2406.02061.

Novak, Roman et al. (2019). "Bayesian Deep Convolutional Networks with Many Channels are Gaussian Processes." In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=B1g30j0qF7.

Pacelli, R. et al. (2023). "A statistical mechanics framework for Bayesian deep neural networks beyond the infinite-width limit." In: *Nat. Mach. Intell.* 5.12, pp. 1497–1507. URL: https://doi.org/10.1038/s42256-023-00767-6.

Papoulis, Athanasios and S. Unnikrishna Pillai (2002). *Probability, Random Variables, and Stochastic Processes*. 4th. Boston: McGraw-Hill.

Petrini, Leonardo et al. (2022). "Learning sparse features can lead to overfitting in neural networks." In: arXiv:2206.12314 [cs.stat]. URL: http://arxiv.org/abs/2206.12314.

Pinkus, Allan (1999). "Approximation theory of the MLP model in neural networks." In: *Acta Numer.* 8, pp. 143–195. URL: https://doi.org/doi:10.1017/S0962492900002919.

Poole, Ben et al. (2016). "Exponential expressivity in deep neural networks through transient chaos." In: *Advances in Neural Information Processing Systems 29*. URL: https://proceedings.neurips.cc/paper/2016/file/148510031349642de5ca0c544f31b2ef-Paper.pdf.

Power, Alethea et al. (2022). "Grokking: Generalization Beyond Overfitting on Small Algorithmic Datasets." In: arXiv:2201.02177 [cs.LG]. URL: https://arxiv.org/abs/2201.02177.

Price, Robert (1958). "A useful theorem for nonlinear devices having Gaussian inputs." In: *IRE Transactions on Information Theory* 4.2, pp. 69–72. URL: https://doi.org/10.1109/TIT.1958.1057444.

Radford, Alec et al. (2018). *Improving Language Understanding by Generative Pre-Training*.

Rahaman, Nasim et al. (2019). "On the Spectral Bias of Neural Networks." In: *Proceedings of the 36th International Conference on Machine Learning*. Vol. 97. Proceedings of Machine Learning Research. PMLR, pp. 5301–5310. URL: https://proceedings.mlr.press/v97/rahaman19a.html.

Refinetti, Maria, Alessandro Ingrosso, and Sebastian Goldt (2023). "Neural networks trained with SGD learn distributions of increasing complexity." In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by Andreas Krause et al. Vol. 202. Proceedings of Machine Learning Research. PMLR, pp. 28843–28863. URL: https://proceedings.mlr.press/v202/refinetti23a.html.

Refinetti, Maria et al. (2021). "Classifying high-dimensional Gaussian mixtures: Where kernel methods fail and neural networks succeed." In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, pp. 8936–8947. URL: https://proceedings.mlr.press/v139/refinetti21b.html.

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin (2016). ""Why Should I Trust You?": Explaining the Predictions of Any Classifier." In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 1135–1144. URL: https://doi.org/10.1145/2939672.2939778.

Risken, Hannes (1996). *The Fokker-Planck Equation*. Springer Verlag Berlin Heidelberg. URL: https://doi.org/10.1007/978-3-642-61544-3_4.

Roberts, Daniel A., Sho Yaida, and Boris Hanin (May 2022). *The Principles of Deep Learning Theory*. Cambridge University Press. URL: https://doi.org/10.1017/9781009023405.

Rosenblatt, F (1958). "The perceptron: a probabilistic model for information storage and organization in the brain." In: *Psychol Rev* 65, pp. 386–408.

Rubin, Noa, Inbar Seroussi, and Zohar Ringel (2024). "Grokking as a First Order Phase Transition in Two Layer Networks." In: *The Twelfth International Conference on Learning Representations*. URL: https://openreview.net/forum?id=3ROGsTX3IR.

Saad, David and Sara A. Solla (1995). "Exact Solution for On-Line Learning in Multilayer Neural Networks." In: *Phys. Rev. Lett.* 74 (21), pp. 4337–4340. URL: https://link.aps.org/doi/10.1103/PhysRevLett.74.4337.

Sarao Mannelli, Stefano, Eric Vanden-Eijnden, and Lenka Zdeborová (2020). "Optimization and Generalization of Shallow Neural Networks with Quadratic Activation Functions." In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., pp. 13445–13455. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/9b8b50fb590c590ffbf1295ce92258dc-Paper.pdf.

Saxe, Andrew, James Mcclelland, and Surya Ganguli (2014). "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks." In: *International Conference on Learning Represenatations*.

Schoenholz, Samuel S. et al. (2017). "Deep information propagation." In: *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*. URL: https://openreview.net/forum?id=H1W1UN9gg.

Schuecker, Jannis et al. (2016). "Functional methods for disordered neural networks." In: *ArXiv*. 1605.06758 [cond-mat.dis-nn]. URL: https://arxiv.org/abs/1605.06758.

Segadlo, Kai et al. (2022). "Unified field theoretical approach to deep and recurrent neuronal networks." In: *Journal of Statistical Mechanics: Theory and Experiment* 2022.10, p. 103401. URL: https://dx.doi.org/10.1088/1742-5468/ac8e57.

Seroussi, Inbar, Gadi Naveh, and Zohar Ringel (2023). "Separation of scales and a thermodynamic description of feature learning in some CNNs." In: *Nature Communications* 14.1, p. 908. URL: https://doi.org/10.1038/s41467-023-36361-y.

Silver, David et al. (2016). "Mastering the game of Go with deep neural networks and tree search." In: *Nature* 529.7587, pp. 484–489. URL: https://doi.org/10.1038/nature16961.

Snoek, Jasper, Hugo Larochelle, and Ryan P Adams (2012). "Practical Bayesian Optimization of Machine Learning Algorithms." In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf.

Sompolinsky, H and A Zippelius (1982). "Relaxational dynamics of the Edwards-Anderson model and the mean-field theory of spin-glasses." In: *Physical Review B* 25.11, pp. 6860–6875.

Szegedy, Christian et al. (2017). "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning." In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. AAAI'17. San Francisco, California, USA: AAAI Press, pp. 4278–4284. URL: https://doi.org/10.1609/aaai.v31i1.11231.

Tiberi, Lorenzo et al. (2024). "Dissecting the Interplay of Attention Paths in a Statistical Mechanics Theory of Transformers." In: arXiv:2405.15926 [cs.LG]. URL: https://arxiv.org/abs/2405.15926.

Tirer, Tom, Joan Bruna, and Raja Giryes (2022). "Kernel-Based Smoothness Analysis of Residual Networks." In: *Proceedings of the 2nd Mathematical and Scientific Machine Learning Conference*. Ed. by Joan Bruna, Jan Hesthaven, and Lenka Zdeborova. Vol. 145. Proceedings of Machine Learning Research. PMLR, pp. 921–954. URL: https://proceedings.mlr.press/v145/tirer22a.html.

Touchette, Hugo (2009). "The large deviation approach to statistical mechanics." In: *Physics Reports* 478.1, pp. 1–69. URL: https://www.sciencedirect.com/science/article/pii/S0370157309001410.

Valle-Perez, Guillermo, Chico Q. Camargo, and Ard A. Louis (2019). "Deep learning generalizes because the parameter-function map is biased towards simple functions." In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=rye4g3AqFm.

Vapnik, Vladimir (1992). "Principles of Risk Minimization for Learning Theory." In: *Advances in Neural Information Processing Systems*. Ed. by J. Moody, S. Hanson, and R. P. Lippmann. Vol. 4. Morgan-Kaufmann, pp. 831–838. URL: https://proceedings.neurips.cc/paper/1991/file/ff4d5fbbafdf976cfdc032e3bde78de5-Paper.pdf.

Vaswani, Ashish et al. (2017). "Attention is all you need." In: *Advances in Neural Information Processing Systems* 30.

Wan, Zitong et al. (2021). "A review on transfer learning in EEG signal analysis." In: *Neurocomputing* 421, pp. 1–14. URL: https://www.sciencedirect.com/science/article/pii/S0925231220314223.

Williams, Christopher KI (1998). "Computation with infinite neural networks." In: *Neural Computation* 10.5, pp. 1203–1216. URL: https://doi.org/10.1162/089976698300017412.

Yaida, Sho (2020). "Non-Gaussian processes and neural networks at finite widths." In: *Proceedings of The First Mathematical and Scientific Machine Learning Conference*. Ed. by Jianfeng Lu and Rachel Ward. Vol. 107. Proceedings of Machine Learning Research. PMLR, pp. 165–192. URL: http://proceedings.mlr.press/v107/yaida20a.html.

Yang, Adam X. et al. (2023). "A theory of representation learning gives a deep generalisation of kernel methods." In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by Andreas Krause et al. Vol. 202. Proceedings of Machine Learning Research. PMLR, pp. 39380–39415. URL: https://proceedings.mlr.press/v202/yang23k.html.

Yang, Ge and Samuel Schoenholz (2017). "Mean Field Residual Networks: On the Edge of Chaos." In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/81c650caac28cdefce4de5ddc18befa0-Paper.pdf.

Yang, Greg (2019). "Wide Feedforward or Recurrent Neural Networks of Any Architecture are Gaussian Processes." In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2019/file/5e69fda38cda2060819766569fd93aa5-Paper.pdf.

Yang, Greg and Edward J. Hu (2020). "Feature Learning in Infinite-Width Neural Networks." In: *ArXiv*. URL: https://arxiv.org/abs/2011.14522.

Yang, Greg et al. (2021). "Tuning Large Neural Networks via Zero-Shot Hyperparameter Transfer." In: *Advances in Neural Information Processing Systems*. Ed. by A. Beygelzimer et al. URL: https://openreview.net/forum?id=Bx6qKuBM2AD.

Yang, Li and Abdallah Shami (2020). "On hyperparameter optimization of machine learning algorithms: Theory and practice." In: *Neurocomputing* 415, pp. 295–316. URL: https://www.sciencedirect.com/science/article/pii/S0925231220311693.

Zagoruyko, Sergey and Nikos Komodakis (2016). "Wide Residual Networks." In: *Proceedings of the British Machine Vision Conference (BMVC)*. Ed. by Edwin R. Hancock Richard C. Wilson and William A. P. Smith. BMVA Press, pp. 87.1–87.12. URL: https://dx.doi.org/10.5244/C.30.87.

Zavatone-Veth, Jacob A., William L. Tong, and Cengiz Pehlevan (2022). "Contrasting random and learned features in deep Bayesian linear regression." In: *Physical*

Review E 105 (6), p. 064118. URL: https://journals.aps.org/pre/abstract/10.1103/PhysRevE.105.064118.

Zeiler, Matthew D. and Rob Fergus (2014). "Visualizing and Understanding Convolutional Networks." In: *Computer Vision – ECCV 2014*. Springer International Publishing, pp. 818–833. URL: https://link.springer.com/chapter/10.1007/978-3-319-10590-1_53.

Zhang, Huishuai et al. (2022). "Stabilize Deep ResNet with a Sharp Scaling Factor τ." In: *Machine Learning* 111.9, pp. 3359–3392. URL: https://doi.org/10.1007/s10994-022-06192-x.

Zhang, Jingfeng et al. (July 2019). "Towards Robust ResNet: A Small Step but a Giant Leap." In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, pp. 4285–4291. URL: https://doi.org/10.24963/ijcai.2019/595.

Zhang, Xiao et al. (2024). "Residual Connections Harm Abstract Feature Learning in Masked Autoencoders." In: arXiv:2404.10947 [cs.CV]. URL: https://arxiv.org/abs/2404.10947.

Zhuang, Fuzhen et al. (2021). "A Comprehensive Survey on Transfer Learning." In: *Proceedings IEEE* 109.1, pp. 43–76. URL: https://doi.org/10.1109/JPROC.2020.3004555.

Zinn-Justin, Jean (1996). *Quantum field theory and critical phenomena*. Clarendon Press, Oxford.

Band / Volume 96
Characterization and modeling of primate cortical anatomy and activity
A. Morales-Gregorio (2023), ca. 260 pp.
ISBN: 978-3-95806-698-4

Band / Volume 97
Hafnium oxide based memristive devices as functional elements of neuromorphic circuits
F. J. Cüppers (2023), vi, ii, 214 pp
ISBN: 978-3-95806-702-8

Band / Volume 98
Simulation and theory of large-scale cortical networks
A. van Meegen (2023), ca. 250 pp
ISBN: 978-3-95806-708-0

Band / Volume 99
Structure of two-dimensional multilayers and topological superconductors: surfactant mediated growth, intercalation, and doping
Y.-R. Lin (2023), x, 111 pp
ISBN: 978-3-95806-716-5

Band / Volume 100
Frequency mixing magnetic detection for characterization and multiplex detection of superparamagnetic nanoparticles
A. M. Pourshahidi (2023), X, 149 pp
ISBN: 978-3-95806-727-1

Band / Volume 101
Unveiling the relaxation dynamics of Ag/HfO2 based diffusive memristors for use in neuromorphic computing
S. A. Chekol (2023), x, 185 pp
ISBN: 978-3-95806-729-5

Band / Volume 102
Analysis and quantitative comparison of neural network dynamics on a neuron-wise and population level
R. Gutzen (2024), xii, 252 pp
ISBN: 978-3-95806-738-7

Band / Volume 103
3D Scaffolds with Integrated Electrodes for Neuronal Cell Culture
J. Abu Shihada (2024), vii, 163 pp
ISBN: 978-3-95806-756-1

Band / Volume 104
Advances in Photoemission Orbital Tomography
A. Haags (2024), ix, 254 pp
ISBN: 978-3-95806-766-0

Band / Volume 105
Quantitative investigation of point defects and their dynamics in focused ion beam-prepared group III-nitride lamellas by off-axis electron holography
K. Ji (2024), 164 pp
ISBN: 978-3-95806-782-0

Band / Volume 106
NeuCoNS and Stacked-Net: Facilitating the Communication for Accelerated Neuroscientific Simulations
R. Kleijnen (2024), xx, 110, xxi-xxxiv pp
ISBN: 978-3-95806-788-2

Band / Volume 107
Construction of a Spiking Network Model of Macaque Primary Visual Cortex: Towards Digital Twins
A. Kurth (2024), xvi, 207 pp
ISBN: 978-3-95806-800-1

Band / Volume 108
Spin selectivity of chiral molecules on surfaces
M.R. Safari (2025), xiv, 165 pp
ISBN: 978-3-95806-810-0

Band / Volume 109
Redox-based Random Access Memory Arrays for Computing-In-Memory and Neuromorphic Computing
H. Chen (2025), x, 154 pp
ISBN: 978-3-95806-814-8

Band / Volume 110
Mechanics of deep neural networks beyond the Gaussian limit
K. Fischer (2025), xvi, 138 pp
ISBN: 978-3-95806-815-5

Weitere ***Schriften des Verlags im Forschungszentrum Jülich*** unter
http://wwwzb1.fz-juelich.de/verlagextern1/index.asp

Information
Band / Volume 110
ISBN 978-3-95806-815-5

Mitglied der Helmholtz-Gemeinschaft

JÜLICH
Forschungszentrum