The optimal activation function for the MLP

A first-principled physics-based approach to deep learning

Maarten van Tartwijk

Supervisors: prof. dr. Marco Loog and dr. David Tax

Thesis Advisor: prof. dr. Jan van Gemert

Thesis committee member: dr. Klaus Hildebrandt

Special thanks to: prof. dr. Jos Thijssen (Faculty of Applied Sciences)

To obtain the degree of Master of Science in Computer Science

at the Delft University of Technology

May 2024

Preface

This report is a result of my personal interest in trying to understand deep learning on a first-principled basis. That is, my interest lies in building my understanding of deep learning from some fundamental building blocks or axioms, rather than an empirical investigation. Luck has it that two theoretical physicists published a book [21] in 2022 that offers a first-principled approach to deep learning theory. This book forms the basis for my report.

A note on language
In the rest of this report, I will use the first-person singular form ("I") to refer to decisions I made, and to convey reasoning and points of discussion of personal origin. I will use first-person plural ("we") when I am explaining the work of Roberts, Yaida, and Hanin to the reader, in a way that I hope is accessible to fellow computer science students/scholars that are not experts in theoretical physics.

Acknowledgements

I would like to express my profound gratitude to Marco and David, without whose guidance and support this project would never have materialized. They have without a doubt been the best supervisors I have had during my academic career, and I was struck by their genuine interest in my ideas and generous amounts of time they spent on supervising me. I would also like to thank prof. dr. Thijssen from the faculty of applied science for helping me understand the physics required for this project. I am grateful to prof. dr. Van Gemert and dr. Hildebrandt for willing to be a member of my thesis committee.

Contents

1 Introduction

What is the optimal activation function for a fully connected deep neural network, also known as multilayer perceptron (MLP)? To answer this question, we must first define what makes an activation function 'optimal'. To arrive at such a definition we require a type of understanding of MLPs that goes beyond the universal approximation theorem [6]–[8] and typical initialization schemes such as Xavier-initialization [12].

In 2022, two theoretical physicists published a book [21] that offers a first-principled theoretical approach to deep learning theory. The book, "Deep Learning Theory" [21] written by Daniel Roberts and Sho Yaida (with contributions of Boris Hanin), forms the basis for my thesis project. Throughout this report, I will refer to the authors of the book [21] simply as 'the authors'. To my knowledge, the book is the first to offer a comprehensive first-principled theory of deep-learning that is able to explain real, practical networks (as opposed to idealized infinite-width networks, as is the case for the universal approximation theorems). The book borrows heavily from theoretical physics, which can make it particularly challenging to digest for people with a different background, such as computer scientists. Thus, my contribution is two-fold. First, I give an overview of the book's theory of MLPs at initialization[1], while providing the missing links for those without a background in statistical mechanics, critical phenomena and renormalization group theory. Second, I attempt to use this theoretical basis to devise a method of judging an activation function's quality, thereby providing an answer to the question "What is the optimal activation function for the MLP?". Furthermore, I make an attempt to design an optimal activation function and provide experimental results to verify its optimality and compare it to other commonly used activation functions.

In the remainder of this introductory chapter, I will first introduce the foundation on which to build the theory, the so-called first principles or axioms. Section 3.1 introduces this foundation, as well as the notation used to describe it. Next, in section 1.1, we will take a very brief detour into the history of the steam engine and subsequent development of the fields of thermodynamics and statistical mechanics, which will help explain the key dilemma as described by the authors: the difference between a microscopic and macroscopic view of deep learning. In section 1.2, I will explain why physics turns out to be especially useful for building a first-principled theory of deep learning. Finally, in section section 1.3, I give a high-level summary of the main results of [21] on MLPs at initialization, so it is clear to the reader what we are working towards in subsequent chapters.

1.1 Parallels with statistical mechanics: the macroscopic vs. the microscopic

In 1712 Thomas Newcomen developed the first commercially viable steam engine. It was still a very inefficient machine, but this hardly mattered as it was used primarily to pump water from British coal mines: fuel was cheap and readily available. Half a century later, in 1764, James Watt dramatically improved the Newcomen engine [14]. Importantly, both Newcomen and Watt did not have the theorems of thermodynamics to guide them in their inventions. It was only in 1824 that Carnot laid the foundations for the field of thermodynamics in the only book he ever published, with the express purpose of understanding and improving the steam engine:

> *" Notwithstanding the work of all kinds done by steam-engines, notwithstanding the satisfactory condition to which they have been brought to-day, their theory is very little understood, and the attempts to improve them are still directed almost by chance. [...] We propose now to submit these questions to a deliberate examination"*
>
> Nicolas Sadi Carnot, 1824 [1]

The authors suggest that we are at a similar point in history regarding deep learning as Carnot was in 1824. In the decades following the work of Carnot, the laws of thermodynamics were derived through empirical observations[2]. The laws of thermodynamics gave rise to a high-level understanding of the steam engine, of the **macroscopic** machine that it unquestionably is. However, the laws of thermodynamics were not based on some fundamental theory, a theory derived from first principles: not until the work of Maxwell, Boltzmann and Gibbs showed that the laws of thermodynamics can be derived by combining statistics and mechanics. They connected the macroscopic laws of thermodynamics to fundamental **microscopic** properties of matter

[1] It should be noted that the book goes *much* further than this. It includes an analysis of the training procedure, a new understanding of model complexity and the beginnings of applying the theorems to transformers and residual networks.

[2] I posit that there is a historical parallel with the empirical scaling laws derived for deep learning architectures [19], [22].

(particles). In short, for a given system, by considering an infinite number of mental copies of that system (referred to as an **ensemble**), each with a different setting of the microscopic states of particles, the resulting ensemble can be studied through statistical means [16]. Note that by the time Maxwell, Boltzmann, and Gibbs were working on statistical mechanics, the consensus within the scientific community regarding the particulate nature of matter was fairly-well established, but some details were missing. Thus, Maxwell, Boltzmann and Gibbs started out with a clear macroscopic picture of thermodynamics, had to postulate the appropriate microscopic properties, and build a bridge that would connect the microscopic to the macroscopic. The authors state that the field of deep learning is a similar but *opposite* situation. Indeed, we have a crystal clear picture of the microscopic: we know exactly how the microscopic constituents of neural networks behave: it is by our design. The question that remains is, what does the macroscopic view of deep learning look like? Can we build this view from the ground up from the microscopic world of neurons, weights, biases and activations?

1.2 Learning physics vs. learning deep learning

Physics has a long history of coming up with simple theories to describe complex systems. To define what we mean by "simple" and "theory", consider some physical system. Let us define a theory as an equation that relates one state of the system to another state, either as the system progresses through time or as the minimization process of some total amount of energy. Next, consider the set of all possible experiments one could perform on the system. Then, a theory is simple if it can predict the outcome of many more experiments than the number of experiments it takes to learn the value of all the parameters involved in the theory. In this section I will discuss why the nature of our reality forces physicists to find simple theories, why this bias towards simple theories is seemingly absent within the context of deep learning, and how we can nonetheless introduce simplicity to deep learning theory by borrowing ideas from physics. This section is based on [20].

We refer to the parameters that define the state of the system as **degrees of freedom**. We could define the state of the system through the state of all its constituents, e.g. the positions and momenta of all particles in a system. In statistical mechanics, such a snapshot of all the microscopic details of a system is referred to as a **microstate**. Naively, a theory making predictions about such microstates would require many parameters as each degree of freedom could potentially interact with any number of other degrees of freedom.

Fortunately, reality has numerous ways of constraining the space of possible theories a physicist might consider. Some of these constraints are also relevant to deep learning theory. Firstly, in our universe, interactions between degrees of freedom are limited through the principle of **locality**: making a local change to the state of a system only has an effect on the immediate surroundings. This rules out the majority of potential interactions, because we can drop any interaction spanning more than some short distance. Secondly, the exact microstate of a system is generally inaccessible. This forces the physicist to define **macrostates** of the system that are actually observable through experiments. In statistical mechanics, macrostates are derived through statistical means by leveraging the law of large numbers (of particles). In summary, the locality of the universe and the macroscopic nature of its inhabitants, in this case, the physicists, gives physics scholarship a bias towards finding simple theories.

At first glance, a deep learning theorist's reality offers much weaker constraints. Indeed, we have access to the exact microstate of the system: the settings of the weights and biases, the values of the preactivations, etc. A notion of locality does exist in the sense that feed-forward networks are recursive in nature, enabling the consideration of just two subsequent layers at a time. However, from a different perspective, the locality of interactions between neurons is rather limited: in a fully connected network, all neurons in one layer affect all neurons in the subsequent layer. Thus, none of the activations of the neurons in a single layer are statistically independent, or in physics-speak: for any neuron in a given layer, we potentially have to model interactions between up to n_l neurons at a time.

Nevertheless, we can borrow ideas from physics to impose our own constraints to find a simple theory. First, even though we *do* have access to the exact microstate of the system at all times, that does not mean we cannot define a macrostate. For example, we could take the probability distribution of the preactivations of the last layer as our macrostate. Then, it is our hope that we can somehow limit the number of parameters involved in this distribution, leading to a **sparse** description of the macrostate. In statistical mechanics, the sparsest of descriptions is obtained when one considers the thermodynamic limit, i.e. an infinite number of degrees of freedom, as a sort of physical manifestation of the Central Limit Theorem. Similarly, in deep learning we can achieve a sparse description by assuming an infinite network width. In this case, our macrostate

becomes a simple Gaussian distribution completely defined by its mean and covariance matrix. However, the infinite-width assumption contradicts the express goal of the authors: obtaining a theory that describes real neural networks. To remedy this shortcoming we can take the infinite-width solution as a starting point and introduce a correction by assuming a large width rather than an infinite width. This results in a distribution that is almost Gaussian, but not quite. The correction involves modeling interactions between up to four different neurons within the same layer. Limiting the number of degrees of freedom partaking in an interaction to some number k is known as k-**locality**. To conclude, the self-imposed constraints borrowed from physics to help find a 'simple' theory of deep learning are threefold:

- To reduce the number of parameters involved in a theory for MLPs, we define a macrostate. For example, the probability distribution of the preactivations of the last layer of the MLP. We can derive the possible macrostates of the MLP using yet another technique from physics: the renormalization group, which I discuss in the next two chapters.
- To further reduce the number of parameters required for the equations of the theory, we only model interactions between up to four neurons in the same layer. This in an assumption of k-locality with $k = 4$. The resulting probability distribution that describes our macrostate is nearly-Gaussian. Section 3.2 defines the nearly-Gaussian distribution.
- The assumption of 4-locality yields a valid model only when the width of the MLP is large. Thus, mirroring statistical mechanics, the authors first study the 'thermodynamic limit' of deep learning, i.e. the infinite-width MLP, which yields a basic Gaussian solution. Next, they make a finite-width correction to this base solution, resulting in a nearly-Gaussian distribution. This approach whereby a simplified base solution is modified with small corrections is an example of perturbation theory, which I discuss in section 3.3.

1.3 Summary: a first-principled theory of MLPs at initialization

In this section, I give a high-level overview of the authors' main results regarding MLPs at initialization, while avoiding most of the physics for now. At the heart of the authors' theory of MLPs at initialization is the perspective of viewing the MLP as a critical phenomenon. Critical phenomena can occur in all kinds of physical systems, when they find themselves at a special point (the critical point) in the phase diagram, somewhere at the boundary between two phases (e.g. liquid $\leftrightarrow$ gas, ferromagnetic $\leftrightarrow$ paramagnetic). A system at criticality has unusual behavior: it exhibits scale-invariance, meaning that it appears similar no matter the observation scale, not unlike a fractal. The authors define a notion of criticality for the MLP: if the initialization hyperparameters are chosen correctly, MLPs also display a kind of scale-invariance in the sense that correlations between inputs are preserved in the preactivations of the network during the forward pass.

At the core of the authors' analysis is a theoretical framework called the **renormalization group** (RG). The renormalization group framework is applicable when a system is self-similar, in the sense that it can be described by the same mathematical model at all observation scales. MLPs are also self-similar: the authors show that each layer can be described by the same model. Without making any prior assumptions about the nature of the macrostates of a system, an RG analysis yields insights into the distinct possible macrostates and the transitions between them. These insights are all put together in the centerpiece of a renormalization group analysis: RG flow diagram. This diagram is best described as a topology of macrostates, and tells us for any given microscopic state of the system, what macrostate the system must be in. The diagram also includes the critical point, where the parameters of the model are the same at all observation scales. Based on the behavior of physical systems near the critical point, they can be categorized into **universality classes**. Likewise, the authors find that activation functions of MLPs can also be divided into separate classes, where each class has some characteristic behavior and can be described by the characteristic set of equations.

The initialization hyperparameters of the MLP that correspond to the critical point in the RG flow diagram are referred to as the **critical tuning**. The authors provide a generally applicable algorithm to determine the existence and values of the critical tuning of an MLP for any analytic or ReLU-like activation function. It reproduces commonly used initialization schemes, such as Xavier initialization [12].

Finally, to satisfy the condition that makes an RG analysis applicable, namely that the same model is an accurate description of the system at all observation scales (layer depths for the MLP), the authors assume

that the MLP has a large-but-finite width. At the same time, the authors make argue that the wide network regime is precisely when deep networks are most useful in practice.

1.4 Report overview

In the two subsequent chapters, I will show, in a computer scientist-friendly way, how the authors use physics to build upon the initialization scheme and forward-pass equations to arrive at deeper understanding of the MLP, the end-goal being an algorithm for critical initialization parameters and the division of activation functions into universality classes. To this end, chapter 2 paints a backdrop of statistical mechanics against which we can compare and contrast the most important concepts put forward in [21]. Chapter 3 explains how the techniques borrowed from physics can be applied to understand MLPs at initialization.

Chapter 4 covers my own contributions. I use the theoretical basis laid out in the previous chapters to compile a list of properties that should enable the objective comparison of activation functions. Moreover, I design two polynomial activation functions that optimize some properties from the aforementioned list. Additionally, I derive some computational rules that could be useful for those seeking to modify or design their own activation functions. Furthermore, I analyze how fluctuations across initializations of the size of the network output can be minimized for ReLU-like activation functions. Lastly, I also introduce a new metric, which must be computed numerically, that can explain the behavior of activation functions when the inputs are very large (which is something the original theory does not cover).

In chapter 5, I describe the experiments I set up to compare the behavior of commonly used activation functions and my specially designed functions empirically. Chapter 6 compiles the most important results from these experiments. Chapter 7 contains the discussion and conclusions.

2 The computer scientist's guide to critical phenomena

This chapter paints a backdrop of statistical mechanics that should help to understand how critical phenomena and MLPs are related. The term 'critical phenomena' refers to the behavior of systems that are close to- or at a critical point. Later in this chapter, I will clarify the definition and importance the critical point, but for now, it suffices to know that critical points are special points in the phase diagram of matter, where a phase transition is accompanied by unusual behavior of various properties of the system. Near the critical point, these properties often exhibit scale invariance, meaning they behave the same way regardless of the scale at which they are observed. Furthermore, various observables behave according to power laws in their approach to the critical point. The exponents of such power laws are referred to as critical exponents. Remarkably, completely different phenomena, such as the liquid-gas transition or the ferromagnetic-paramagnetic phase transition of a magnet, can be described by the same set of critical exponents, hinting at some kind of universal behavior of complex systems, regardless of their microscopic details. Finally, fluctuations (over time and distance) in the properties of a system at criticality become highly correlated and large in magnitude, in the order of the size of the system. [5]

The authors show that the MLP, at initialization, can be understood as critical phenomenon, including phases, critical points, power law behavior of certain observables, critical exponents, universality, and large fluctuations[3]. Therefore, we can borrow powerful, established mathematical tools from statistical mechanics and repurpose them to study the MLP. One such mathematical tool is **renormalization group flow** (RG flow). RG flow finds its origin in quantum electrodynamics, but has since been generalized and applied to numerous other fields, such as solid-state physics, chemical physics, and statistical mechanics. The framework of renormalization group flow enables the systematic investigation of a system at different observation scales by means of a so-called "coarse-graining" procedure. Studying a system using RG flow leads to a derivation of the system's macroscopic states and the possible transitions (and transition points) between those states. For this reason, RG flow can be described as yielding a topological description of the macroscopic states of the system under investigation. [5]

Due to its general applicability, there are many ways to introduce the framework of RG flow. In this chapter, I have chosen to give a surface-level introduction to RG flow within the context of magnetic phase transitions, where I cover only those concepts that have a counterpart in the authors' analysis of MLPs at initialization. With this goal in mind:

- Section 2.1 introduces the phases of- and a model for the magnetic system.
- Section 2.2 explains how a topology of macrostates emerges from this model when subjected to the renormalization group framework.
- Section 2.3 describes the importance of the critical point and how it relates to concepts such as critical exponents and universality.
- Finally, in section 2.4 we circle back from physics to deep learning to see how the MLP can be understood as a critical phenomenon.

2.1 The 2D Ising model of a magnetic system

Consider an iron magnet. It has two phases of magnetism: the ferromagnetic phase and the paramagnetic phase. In the ferromagnetic phase, the magnetic moments of the atoms are aligned, and the material acts as a magnet all on its own. In the paramagnetic phase, the magnetic moments are oriented randomly, unless a magnetic field is applied. This behavior is visualized in fig. 1. The two phases are separated by a critical temperature T_c called the Curie temperature. Below the Curie temperature, the weak interactions of the magnetic moments keep them aligned and the material is ferromagnetic. Above the Curie temperature, the thermal disorder overcomes the weak interactions of the magnetic moments and the material is in the paramagnetic phase. Precisely at the critical point T_C the magnet will exhibit special behavior that is neither ferromagnetism nor paramagnetism. Instead, there are regions of the material where the magnetic moments are ordered, and there are regions of disorder. Moreover, the sizes of these patches of order and disorder occur at all length-scales. Much like a fractal, the material is indistinguishable across observation scales. [5], [23].

[3] Fluctuations over initializations, instead of time or distance

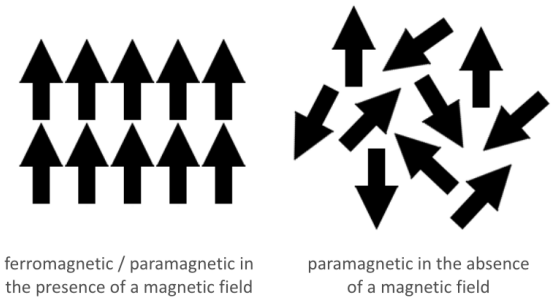

Figure 1: Two magnetic phases, adapted from [28].

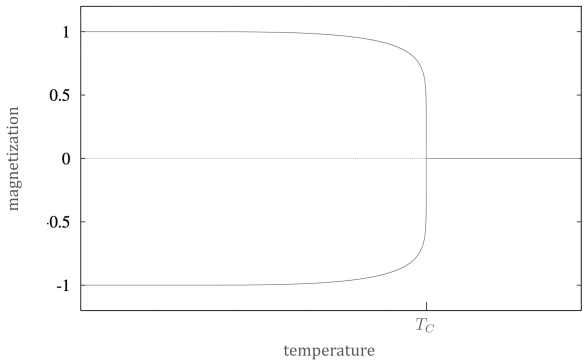

Figure 2: The magnetization of the system modeled by the 2D Ising model. Adapted from [23].

Why is the behavior of the magnet characterized by two distinct phases? Why is there such a stark transition point between the phases? These are questions we can answer by studying the 2D Ising model of a magnetic system within the framework of RG flow. The general Ising model considers a system with degrees of freedom that can assume only two values, often taken to be $+1$ and -1. Ising models have a wide range of applications, including (but not limited to) problems in epidemiology, neuroscience and the study of alloys [23]. Here, we are concerned with the 2D Ising model describing a simplified magnetic system in the absence of an external magnetic field. The model consists of a square lattice where on each lattice site there is a particle that can have 'spin-up' $(+1)$ or 'spin-down' (-1). This 'spin' represents the direction of the magnetic moment at each lattice site. Let the spin at lattice site i be denoted as s_i. If we have N lattice sites, the phase space of our system is given by

$$S = \{-1, +1\}^N \ . \tag{2.1.1}$$

A microstate of the system is a particular configuration of all spins in the lattice, which we denote as $\{s_i\} \in S$. Let us define a dimensionless quantity for the magnetization of the system which is given by the relative difference between the number of up-spins N_+ and down-spins N_-:

$$m = \frac{N_+ - N_-}{N} \tag{2.1.2}$$

The behavior of this magnetization in relation to the temperature is given by fig. 2. The two distinct branches for temperatures below the critical temperature T_C represent the spin-up and spin-down paths that the system can end up in. It takes a prohibitively long time (exponential in the number of lattice sites) to spontaneously switch between the two paths, unless we apply an external magnetic field, or increase the temperature above the critical point and then cool down the system (and hope the system ends up in the desired branch). Note the stark transition between the two phases, which we will explain in the next section using the renormalization group framework.

We model the behavior of the system through its Hamiltonian $H(\{s_i\})$, which gives the total energy of the system. J and K are positive couplings that represent the strength between nearest-neighbor and next-to-nearest-neighbor interactions, respectively. The lattice's structure and neighbor 'types' are described in fig. 3. The Hamiltonian is given by

$$H(\{s_i\}) = -\frac{1}{\beta}\left(J\sum_{\langle i,j\rangle} s_i s_j + K\sum_{\langle\langle i,j\rangle\rangle} s_i s_j\right) \tag{2.1.3}$$

where $\langle i,j\rangle$ are the nearest-neighbor pairs, $\langle\langle i,j\rangle\rangle$ are the next-to-nearest-neighbor pairs, and $\beta = 1/(k_B T)$ where k_B is the Boltzmann constant. Using the Hamiltonian, we say something about the relative probabilities of all microstates of the system. The probability that the system is in a certain microstate is proportional to the microstate's Boltzmann factor:

$$P(\{s_i\}) \propto e^{-\beta H(\{s_i\})} \tag{2.1.4}$$

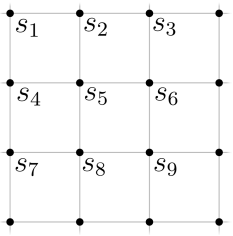

Figure 3: The lattice structure of the 2D Ising model for a magnetic system. The nearest neighbors of s_5 are s_2, s_4, s_6 and s_8. The next-to-nearest neighbors are s_1, s_3, s_7 and s_9.

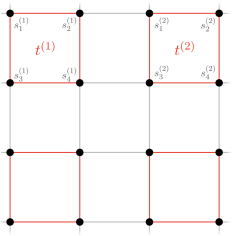

Figure 4: A visualization of the coarse-graining procedure performed in a renormalization transformation.

The so-called partition function $Z(\beta)$ can be used to turn these relative probabilities into absolute probabilities:

$$Z(\beta) = \sum_{\{s_i\}} e^{-\beta H(\{s_i\})} \text{ , where } P(\{s_i\}) = \frac{e^{-\beta H(\{s_i\})}}{Z(\beta)} . \tag{2.1.5}$$

More than just a normalization factor, the partition function can be viewed as encoding all the energy levels and their degeneracies (the number of microstates with a particular energy level) of the system. Many quantities of interest can be directly derived from the partition function, such as the Helmholtz free energy ($F = -k_B T \log Z(\beta)$) and the expected energy of the system ($\langle E \rangle = -\frac{\partial}{\partial \beta} \log Z(\beta)$). If two systems have identical partition functions, they are said to have the same thermodynamic behavior. This is important for RG flow: during the coarse-graining procedure, the partition function must be invariant as our objective is to understand the system at larger and larger observation scales, *not* to change the dynamics of the system.

2.2 Renormalization group flow for the 2D Ising model

At the core of RG flow is the concept of self-similarity. We can apply the framework of RG flow when a system can be described by the same model regardless of observation scale (or energy scale in quantum electrodynamics) [16]. Such models are said to be **renormalizable**. The basic playbook of RG flow is as follows: first, we consider our model at the microscopic level, and then increase our observation scale. As we increase our observation scale, we need to change the value of certain parameters in the model. Thus, there exists a *flow* in the model's parameters as we increase the observation scale. By studying the direction and fixed points of this flow, we can gain valuable insights into the macroscopic behavior of our system.

To make things a bit more precise, let us consider the Hamiltonian $H(\{s_i\})$ for the 2D Ising model of a magnetic system. As is, $H(\{s_i\})$ is a function of the microscopic state of the system, i.e. the configuration of fine-grained spins $\{s_i\}$. We can define a coarser view of the system by increasing our observation scale: we group the original spins s_i into blocks of four and define an effective spin $t^{(k)}$ for each block. The coarse-graining procedure is depicted in fig. 4. At the new observation scale, the state of the system is described by a configuration of coarse-grained spins $\{t^{(k)}\} \in \{-1, +1\}^{N/4}$. A discrete probability distribution defines the effective spin of each block, denoted by $P\left(t^{(k)};\ s_1^{(k)}, s_2^{(k)}, s_3^{(k)}, s_4^{(k)}\right)$, where the $s_i^{(k)}$ are the fine-grained spins that make up the block:

	$P\left(+1;\ s_1^{(k)}, s_2^{(k)}, s_3^{(k)}, s_4^{(k)}\right)$	$P\left(-1;\ s_1^{(k)}, s_2^{(k)}, s_3^{(k)}, s_4^{(k)}\right)$
$s_1^{(k)} + s_2^{(k)} + s_3^{(k)} + s_4^{(k)} > 0$	1	0
$s_1^{(k)} + s_2^{(k)} + s_3^{(k)} + s_4^{(k)} < 0$	0	1
$s_1^{(k)} + s_2^{(k)} + s_3^{(k)} + s_4^{(k)} = 0$	$\frac{1}{2}$	$\frac{1}{2}$

Note that for any input spins $s_1^{(k)}, \ldots, s_4^{(k)}$, we have $\sum_{t^{(k)}=-1,+1} P\left(t^{(k)};\ s_1^{(k)}, s_2^{(k)}, s_3^{(k)}, s_4^{(k)}\right) = 1$. Using this block-level distribution for the coarse spins, we can define a distribution for the entire configuration of coarse spins:

$$P\left(\{t^{(k)}\}\middle|\{s_i\}\right) = \prod_{t^{(k)}} P\left(t^{(k)};\ \ s_1^{(k)}, s_2^{(k)}, s_3^{(k)}, s_4^{(k)}\right) \ . \tag{2.2.1}$$

As mentioned earlier, we need to check that the partition function remains invariant under our coarse-graining procedure in order for our RG flow analysis to be valid. To this end, consider the probability of the system being in a particular configuration of coarse-grained spins:

$$\begin{aligned} P\left(\{t^{(k)}\}\right) &= \sum_{\{s_i\}} P\left(\{t^{(k)}\}\middle|\{s_i\}\right) P(\{s_i\}) \\ &\propto \sum_{\{s_i\}} P\left(\{t^{(k)}\}\middle|\{s_i\}\right) e^{-\beta H(\{s_i\})} \ , \end{aligned} \tag{2.2.2}$$

where on the first line we used the law of total probability, and on the second line we used eqs. (2.1.4) and (2.2.1). Next, we calculate the partition function $Z'(\beta)$ for the coarse-grained model by summing the previous result over all configurations of coarse-grained spins:

$$\begin{aligned} Z'(\beta) &= \sum_{\{t^{(k)}\}} \sum_{\{s_i\}} P\left(\{t^{(k)}\}\middle|\{s_i\}\right) e^{-\beta H(\{s_i\})} \\ &= \sum_{\{s_i\}} e^{-\beta H(\{s_i\})} \sum_{\{t^{(k)}\}} P\left(\{t^{(k)}\}\middle|\{s_i\}\right) \\ &= \sum_{\{s_i\}} e^{-\beta H(\{s_i\})} \cdot 1 \\ &= Z(\beta) \ . \end{aligned} \tag{2.2.3}$$

Indeed, our coarse-graining procedure leaves the partition function invariant. Therefore, we have succeeded in increasing our observation scale, integrating out fine-grained details, while keeping the thermodynamic behavior of the model the same. An updated Hamiltonian $H'(\{t^{(k)}\})$ will describe the coarse-grained model. In general, $H'(\{t^{(k)}\})$ can have an arbitrary form with many terms and different couplings, but we assume that is has two dominant terms representing nearest- and next-to-nearest-neighbor interactions[4]:

$$H'(\{t^{(k)}\}) = -\frac{1}{\beta}\left(J' \sum_{\langle k,k'\rangle} t^{(k)} t^{(k')} + K' \sum_{\langle\langle k,k'\rangle\rangle} t^{(k)} t^{(k')} + \text{ other much smaller terms}\right) \ . \tag{2.2.4}$$

There exists a mapping from the fine-grained to the coarse-grained couplings. This mapping is known as the **renormalization transformation** (RT):

$$(J, K) \xrightarrow{RT} (J', K') \ . \tag{2.2.5}$$

The derivation of the equations that define this mapping is outside the scope of this report, but I should note that the 2D Ising model was solved analytically in 1944 by Nobel laureate Lars Onsager [2]. However, we do need to discuss the consequence of the existence of this mapping. First, note that there is no reason we must stop the coarse-graining procedure after just one iteration. The existence of the renormalization transformation and its recursive nature imply the existence of a flow in the so-called J, K-space: for any point (J, K) we can calculate where the renormalization transformation would bring us, thus defining a vector field. This vector field, or renormalization group flow diagram, will be the central object of study for the rest of this section and is depicted in fig. 7. Second, since the total energy of the coarse grained system cannot change under the renormalization transformation, i.e. $H'(\{t^{(k)}\}) = H(\{s_i\})$, it is necessary to change the temperature of the coarse-grained model to compensate for the change $(J, K) \xrightarrow{RT} (J', K')$. We call this new temperature the **effective temperature** T_{RG}. The change in temperature *does not imply* that the temperature of the system has changed, but rather that at our new observation scale, the system behaves as if we were looking

[4]This must be true for the RG flow analysis to be valid. For the 2D Ising model this is a valid assumption, but the reasons are out of the scope of this thesis report. When applying RG flow to deep learning, this is an important detail, which I discuss in chapter 3.

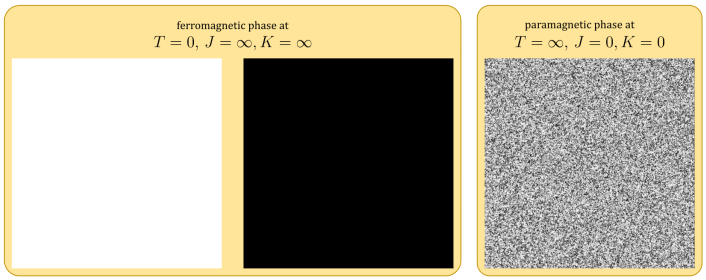

Figure 5: Visualization of the Ising model lattice at two temperature extremes. Here a white pixel is a lattice site with spin $+1$, and a black pixel is a lattice site with spin -1. In the absolute zero limit, all lattice sites have the same spin, either spin-up or spin-down, and the system is in the ferromagnetic phase. In the infinite temperature limit, the spins of the lattice sites are uncorrelated, and the system is in the paramagnetic phase. Adapted from [13].

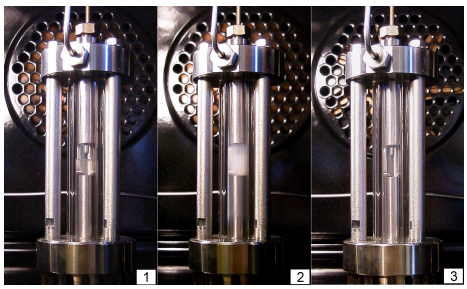

Figure 6: A demonstration of critical opalescence of ethane. (1) The liquid phase. (2) The system is at the critical point. Critical opalescence can be observed as the fluid turns milky white. (3) The gas phase. Taken from [24].

at the microscopic level with an actual temperature $T = T_{RG}$. To illustrate, recall that at low temperature, there is a strong coupling between the spins (J and K large). Vice versa, at high temperature, there is a weak coupling between the spins (J and K small). If we start with a point in the RG flow corresponding to a low temperature, the flow will bring us to a point corresponding to an even lower temperature. Conversely, if we start with a point in the RG flow corresponding to a high temperature, the flow will bring us to an even higher effective temperature. Again, this *does not* imply that the temperature of the system changes as our observation scale grows (that would be rather unphysical), but it means that the system starts to look more and more as if it were at one of two temperature extremes as we zoom out, as depicted in fig. 5.

The key takeaways of the RG flow diagram (fig. 7) are as follows:

- **Fixed points**. A fixed point in an RG flow diagram is a point in the parameter space that remains unchanged under the renormalization transformation. In this case, the RG flow has two attractive fixed points at the temperature extremes, and one repulsive fixed point at the critical temperature. Fixed points that are approached exponentially quickly under the RG transformations, are referred to as 'trivial' fixed points. The two attractive fixed points of the 2D Ising model's RG flow diagram are examples of trivial fixed points.
- **Criticality**. Critical points are the nontrivial fixed points of the RG flow. At the critical temperature, the system is perfectly self-similar, meaning that the structures of up- and down spins look similar at any observation scale, a bit like fractal. The flow near a critical point can be linearized, resulting in power law dynamics of various properties of the system.
- **Separatrix**. The boundary between the two trivial fixed points is a line whose points all flow in to the critical point. This boundary is called the separatrix.
- **A topology of macrostates**. To get from the microscopic lattice of atoms to a real life magnet, we have to increase our observation scale by a factor of multiple millions. Therefore, any point in the parameter space will flow to one of the fixed points as we increase our observation scale to match our macroscopic reality. Thus, *the fixed points of the RG flow diagram reflect the possible macrostates of the system*. The attraction basins of the two trivial fixed-points cover the entire parameter space (with the exception of the critical point and the separatrix). This explains the two distinct phases of our magnetic system: any system with a temperature below the critical temperature will, at the macroscopic scale, look like a system at $T = 0$, and hence ferromagnetic. Conversely, any system with a temperature higher than the critical temperature will look like a system at $T = \infty$, and hence paramagnetic.

Figure 8 attempts to make the coarse-graining procedure more intuitive using the same black-and-white pixel technique as fig. 5. However, I *strongly* advise the reader to watch the animation on which fig. 8 is based[5]. In the video, the red dot on the temperature axis on the bottom visualizes the evolution of the effective

[5]For convenience: https://www.youtube.com/watch?v=MxRddFrEnPc

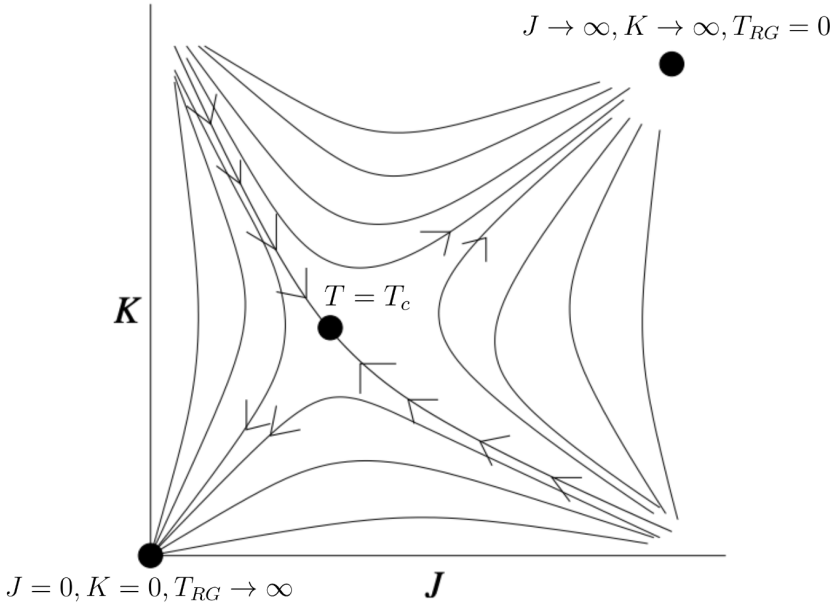

Figure 7: The RG flow diagram for the 2D Ising model with nearest- and next-to-nearest-neighbor interactions. There are two trivial fixed points at $T_{RG} = \infty$ and $T_{RG} = 0$. There is one nontrivial fixed point, also referred to as a critical point, at the critical temperature T_c. Adapted from [23].

temperature T_{RG} during the coarse-graining procedure.

2.3 The importance of the critical point

For several reasons, critical points are of great interest in physics [3], [4], [11], [25], [26]. Treating the MLP as a critical phenomenon is an interesting idea for the very same reasons, which will be discussed in section 2.4.

First, near critical points, many physical systems exhibit universal properties, meaning they behave similarly regardless of their microscopic details. This universality is expressed through the **critical exponents** of those systems. Critical exponents are the exponents of the power laws according to which various properties of the system behave near the critical point. Systems in the same **universality class** have identical critical exponents. Universality allows physicists to use the same theoretical models and mathematical tools across different systems.

Second, at the critical point, the system looks similar at all observation scales in a fractal-like manner. Due to this self-similarity across observation scales, observing the macrostate of a system at criticality provides a window into the microscopic state of the system. In other words, the microscopic and macroscopic world become strongly correlated, exemplified in the divergence of the so-called correlation length at the critical point. The correlation length quantifies the typical distance at which fluctuations or perturbations in the system become uncorrelated or decay to zero, indicating the extent of spatial coherence or ordering within the material. To illustrate, take the liquid-gas phase transition of ethane. Under normal conditions, small fluctuations in the density of the fluid do not produce any visual cues at the macroscopic observation scale. However, if one were to adjust the temperature and pressure just right as to be at the critical point, the fluctuations in density become so large that they become visible to the naked eye, turning the fluid into a milky white substance (fig. 6), a phenomenon called critical opalescence [9].

Third, by studying the RG flow near the critical point, one can determine if the model includes any irrelevant interactions. For example, take the 2D Ising model of a magnet. If, no matter our starting point (J, K), the renormalization transformations would always bring the next-to-nearest neighbor coupling K to zero, we can conclude that K is not relevant for the macroscopic behavior of the system. In such a case, K is deemed an irrelevant coupling. Couplings that remain fixed under RG flow are called 'marginal', while couplings that

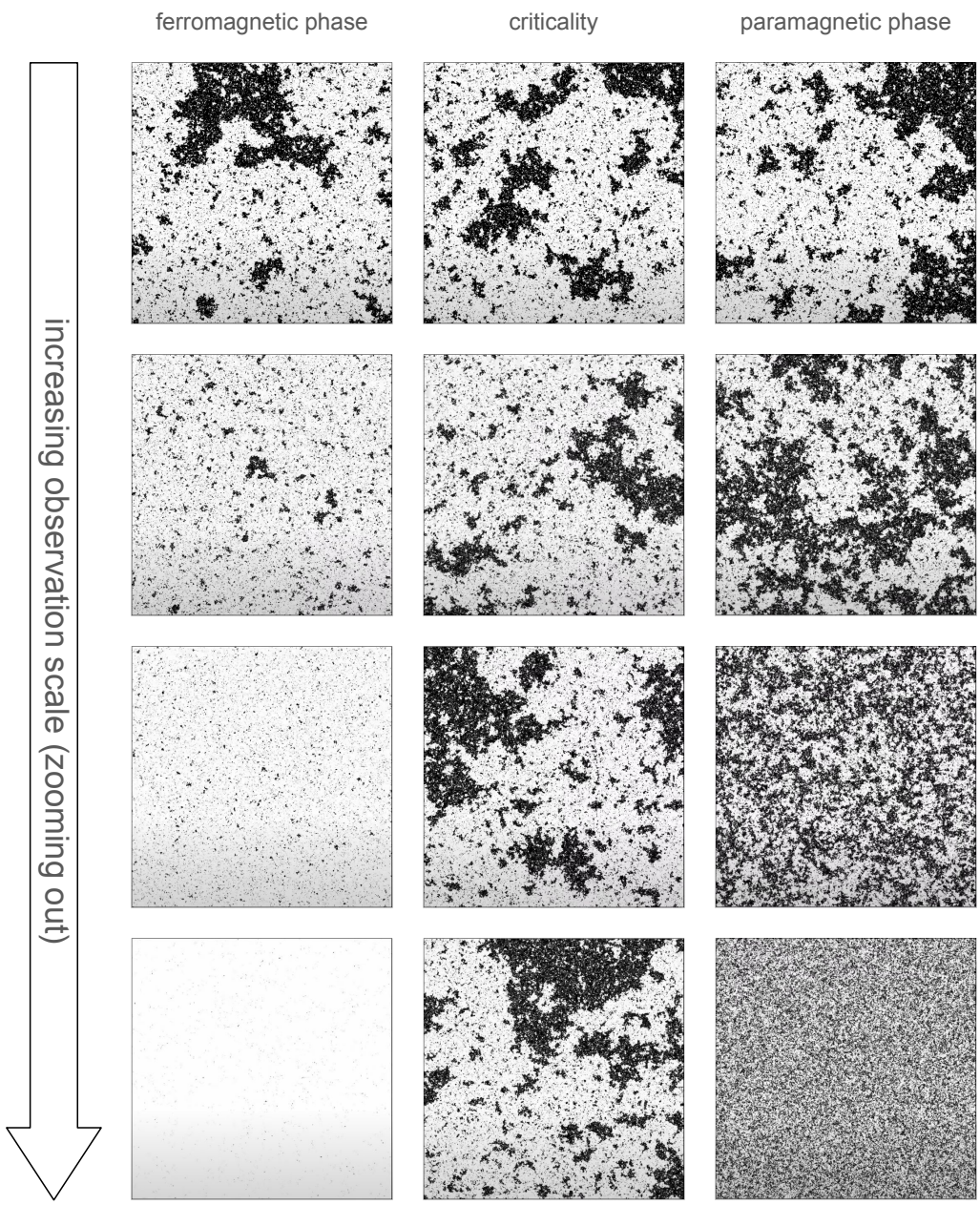

Figure 8: Visualization of the Ising lattice for a magnetic system, (white = spin up, black = spin down). In the ferromagnetic and paramagnetic phase, the system starts to look more and more like one of the temperature extremes at the microscopic level, as depicted in fig. 5. Conversely, at the critical point, the system looks similar no matter the observation scale. Please disregard the slight vertical gradient in the diagrams. It is an artifact from the screenshots being taken from a paused YouTube video. Adapted from [13].

grow with the flow are named 'relevant'. Adding irrelevant interactions to a model has no effect on the critical exponents.

For those who are interested, I have included appendix A, which delves further into critical exponents and (ir)relevant interactions, using the 2D Ising model of a magnet.

2.4 The MLP at initialization as a critical phenomenon

After this crash course in the physics of critical phenomena, we are now ready to discuss why the MLP at initialization can be viewed as a critical phenomenon. Due to its recursive nature, where one layer feeds into the next, the MLP lends itself for analysis through the renormalization group framework. When the authors apply the RG flow framework to the MLP, the layer depth l takes the role of the (observation) scale, and the model is taken to be a probability distribution of the preactivations in layer l. The RG flow diagram then shows how the probability distribution's parameters evolve when we go deeper and deeper into the network. We find that the RG flow has three fixed points, and we can view the MLP as having two distinct phases separated by a critical point. In one phase, the network output vanishes, while in the other phase, the network output explodes. At the critical point, the network finds itself exactly in between the two phases, and many of the same concepts for physical critical phenomena apply.

First, the expected magnitude of the preactivations is preserved from one layer to the next. In this regard, the network displays self-similarity in the size of the preactivations. However, there is a more comprehensive notion of self-similarity: the correlations/differences between different network inputs are also preserved from one layer to the next when an MLP is tuned to criticality. Just like the critical opalescence of ethane, correlations between the microscopic (inputs + first layer) and the macroscopic (the output of the last layer) are strongest when the system (the MLP) is at criticality.

Second, different types of activation functions lead to different toplogies in the RG flow diagrams. Furthermore, the behavior of various observables near the critical point can be described by power laws with corresponding critical exponents. Combined, the different RG flow topologies and the critical exponents enable the activation functions to be divided into distinct universality classes, each with their own nuances that are relevant for practical networks.

Finally, just like physical critical phenomena, a critical tuning of the MLP is accompanied by large fluctuations in various observables, the only difference being that these observables fluctuate from initialization to initialization instead of over some kind of time or distance metric.

3 Renormalization group flow for the MLP

This chapter explains how the authors apply the renormalization group framework to the MLP, and what insights can be gained from such an analysis. Recall from chapter 2 that the basic idea of RG flow is as follows:

1. We define a model for the system at the microscopic scale.
2. We hope (or make sure) that this model can be translated to larger scales, by finding a renormalization transformation.
3. The renormalization transformation induces a flow in the parameter space of the model, the so-called RG flow.
4. The topology of the RG flow, namely its fixed points and attraction basins, yields insights into the macrostates of the system.

With these steps in mind, section 3.2 introduces the model that we will subject to RG flow analysis: the nearly-Gaussian distribution. There are some preconditions that must hold for the nearly-Gaussian distribution to be a good approximation. This is best explained through the lens of perturbation theory. Moreover, perturbation theory provides the mathematical tools that enable us to write down a renormalization transformation. Thus, section 3.3 introduces perturbation theory. In section 3.4, I summarize the renormalization transformation(s) found by the authors. Finally, sections 3.5 to 3.8 give an account of the RG flow of the MLP and what we can learn from the flow. This includes an algorithm to find critical initialization hyperparameters in section 3.6.

3.1 Notation and problem setting: the MLP

The authors' work focuses on fully connected networks with non-linear activation functions[6] (and no other common modifications to an MLP). For future reference, I have included table 1 to give a clear overview of all the notation involved. We consider a generic MLP with L layers, where each layer l has n_l neurons. The input to the network is given by a data set X of size $N_{\mathcal{D}}$ where $x_\alpha \in X$ is a vector-valued input with sample index α. For convenience, we also introduce the set of sample indices $\mathcal{D} = \{\alpha \mid \alpha = 1, \dots, N_{\mathcal{D}}\}$. The size of all vector-valued inputs is assumed to be the same and given by n_0. We use $x_{i;\alpha}$ to refer to the i^{th} element of input x_α. For a given input with sample index α, the preactivation $z^{(l)}_{i;\alpha}$ of a neuron i in layer l is given by the following iteration equations:

$$z^{(1)}_{i;\alpha} = b^{(1)}_i \sum_{i,j=1}^{n_0} W^{(1)}_{ij} x_{j;\alpha} \ , \tag{3.1.1}$$

$$z^{(l+1)}_{i;\alpha} = b^{(l+1)}_i \sum_{i,j=1}^{n_l} W^{(l+1)}_{ij} \ \sigma\left(z^{(l)}_{j;\alpha}\right) \ , \tag{3.1.2}$$

where $b^{(l)}_i$ and $W^{(l)}_{ij}$ are the biases and weights of the neuron and σ is the activation function.It will also be convenient to define $z^{(l)}$ as a matrix of size $n_l \times N_{\mathcal{D}}$, which captures all the preactivations in layer l for all inputs in the dataset X. Finally, I should stress that the dataset X is assumed to be *given*, i.e. deterministic. In contrast, the weights and biases are random variables (non-deterministic), since they are assumed to be independently initialized from the following Gaussian distributions:

$$b^{(l)}_i \ \sim \ \mathcal{N}\left(0, \ C^{(l)}_b\right) \ , \tag{3.1.3}$$

$$W^{(l)}_{ij} \ \sim \ \mathcal{N}\left(0, \ \frac{C^{(l)}_W}{n_{l-1}}\right) \ , \tag{3.1.4}$$

where $C^{(l)}_b$ and $C^{(l)}_W$ are layer-specific initialization hyperparameters. The non-determinism of the initialization induces a distribution over the preactivations $z^{(l)}$ in each layer. We will use $p(\cdot)$ to denote the probability

[6] Chapter 3 of [21] discusses MLPs with linear activation functions. I covered the work in this chapter extensively in my literature study [27].

density function (PDF) of any observable that is of interest. For example, the PDF of the preactivations of layer l would be denoted as $p\left(z^{(l)}\right)$. We will use the following notation to reduce clutter when multiple sums are involved:

$$\sum_{i_1,...,i_m=1}^{N} f(i_1,...,i_m) \equiv \sum_{i_1=1}^{N} \cdots \sum_{i_m=1}^{N} f(i_1,...i_m) \tag{3.1.5}$$

$$\sum_{\alpha_1,...,\alpha_m \in A} f(\alpha_1,...,\alpha_m)\ , \equiv \sum_{\alpha_1 \in A} \cdots \sum_{\alpha_m \in A} f(\alpha_1,...,\alpha_m)\ . \tag{3.1.6}$$

Lastly, we use a braket notation to denote Gaussian expectations. For a univariate Gaussian expectation of a function $f(z)$, we define

$$\langle f(z) \rangle_k \equiv \frac{1}{\sqrt{2\pi k}} \int_{-\infty}^{+\infty} \exp\left(-\frac{1}{2k} z^2\right) f(z) dz\ , \tag{3.1.7}$$

where k is the standard deviation. For the Gaussian expectation of a function F of the preactivations for an arbitrary neuron but all inputs $\alpha \in \mathcal{D}$, we define For a Gaussian expectation with covariance matrix K and an arbitrary function $F(z_{\alpha_1},...,z_{\alpha_{N_\mathcal{D}}})$ over variables with sample indices only, we define:

$$\begin{aligned} \langle F(z_{\alpha_1},...,z_{\alpha_{N_\mathcal{D}}}) \rangle_K \equiv |2\pi K|^{\frac{n_\mathcal{D}}{2}} \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} \\ \exp\left(-\frac{1}{2} \sum_{\beta_1,\beta_2 \in \mathcal{D}} K^{\beta_1\beta_2} z_{\beta_1} z_{\beta_2}\right) F(z_{\alpha_1},...,z_{\alpha_{N_\mathcal{D}}}) \left[\prod_{\alpha \in \mathcal{D}} dz_\alpha\right]\ , \end{aligned} \tag{3.1.8}$$

where K is the covariance matrix, $|\cdot|$ denotes the determinant of a matrix, and the raised indices $\beta_1\beta_2$ on K in the integrand denote the element in row β_1 and column β_2 of the inverse covariance matrix K^{-1}.

3.2 The model: the nearly-Gaussian distribution

The RG flow analysis requires a model that can accurately describe our system (the MLP at initialization) at different observation scales. The authors choose to model the probability distribution of the preactivations in each layer, where the layer depth functions as the observation scale. We denote the probability density as $p(z^{(l)})$. The authors choose to approximate $p(z^{(l)})$ with a so-called **nearly-Gaussian distribution**. In order to define the nearly-Gaussian distribution, and to verify that it is a good approximation of $p(z^{(l)})$, we need the concept of the **connected M-point correlator**[7]. The connected M-point correlator is similar to the moment of order M, and it is also defined in terms of moments. For an exact definition, see section 2.3 of [27]. The Gaussian distribution, be it univariate or multivariate, has the special property that it is the *only* probability distribution for which the higher order ($M > 2$) connected correlators are zero [21]. Thus, these connected correlators are a measure of non-Gaussianity. The authors define the nearly-Gaussian distribution as a distribution for which the connected M-point correlators with $M > 2$ are small, and there is a hierarchy among the correlators where they become progressively smaller as M is incremented. To verify that the nearly-Gaussian distribution is an accurate approximation of $p(z^{(l)})$, the authors show that the connected correlators of the distribution fit this definition.

Recall from section 1.2 that simple theories generally have sparse descriptions. For the MLP, the sparsest of descriptions is obtained in the infinite-width limit, when the authors show that $p(z^{(l)})$ is Gaussian. Note that $z^{(l)}$ has dimensions $n_l \times N_\mathcal{D}$, i.e. it gives the value of the preactivations of each neuron in layer l for all samples in the dataset. As such, it is convenient to refer to the covariance between any two elements of $z^{(l)}$ by two neural indices i_1 and i_2, and two sample indices α_1 and α_2. The problem with the Gaussian infinite-width solution is that the covariance of the preactivations of two distinct neurons ($i_1 \neq i_2$) is always zero, i.e. it does not model interactions between different neurons. Even if we back off from the infinite-width limit, the authors show that the covariance between different neurons remains 0 (see sections 4.2 and 4.2 of [21]). However, what does change when we back off the infinite-width limit, is that the higher order connected correlators become nonzero, and hence the nearly-Gaussian distribution becomes a candidate model, so long as the higher order connected correlators are small and conform to the aforementioned hierarchy. The nearly-Gaussian distribution *is* able to model interactions between different neurons, because it has more (and different) parameters than the Gaussian distribution.

[7] Known in statistics as *cumulants*.

Notation	Explanation
X	Dataset of inputs (samples) for the MLP ($\lvert X\rvert = N_{\mathcal{D}}$).
$N_{\mathcal{D}}$	The size of the dataset.
$\mathcal{D}$	The set of sample indices, $\mathcal{D} = \{\alpha \mid \alpha = 1, \ldots, N_{\mathcal{D}}\}$.
x_α	The α^{th} sample of the dataset (vector-valued).
$x_{i;\alpha}$	The i^{th} element of the α^{th} sample of the dataset (scalar-valued).
L	The total number of layers of the MLP.
n_l	The number of neurons in layer l. Note: n_0 denotes size of the input vectors.
$z_{i;\alpha}^{(l)}$	The preactivation of neuron i in layer l for a given input x_α, where i is referred to as the neural index and α as the sample index. This is a random variable and it is scalar valued.
$z^{(l)}$	The preactivations of the entire layer l for all samples in the dataset. This is a tensor-valued quantity of size $n_l \times N_{\mathcal{D}}$. It is a random tensor.
$b_i^{(l)}$	The bias of neuron i in layer l. This is random variable and it is scalar-valued.
$W_{ij}^{(l)}$	The weight of neuron i in layer l for an incoming signal of neuron j in layer $l-1$ (or incoming input when $l = 1$). This is a random variable and it is scalar-valued.
$C_b^{(l)}$ and $C_W^{(l)}$	Initialization hyperparameters which set the distributions from which the biases and weights are initialized, respectively.
$\sigma(\cdot)$	The activation function used throughout the network ($\sigma : \mathbb{R} \to \mathbb{R}$).
$p(\cdot)$	The probability density function of some observable of the MLP.
$\langle\cdot\rangle_K$	Either a univariate or multivariate Gaussian expectation, see eqs. (3.1.7) and (3.1.8).

Table 1: An overview of all the notation involved in the authors' analysis of the MLP at initialization [21].

The nearly-Gaussian distribution is defined by a so-called **action**. Let x be an N-dimensional random vector. An action $S(x)$ defines the functional form of a probability density function $p(x)$ through the relation:

$$p(x) \propto e^{-S(x)} \ . \tag{3.2.1}$$

This relation completely determines the shape of the distribution, but to obtain the full probability density function, the partition function (normalization constant) Q must be computed:

$$Q \equiv \int e^{-S(x)} \ d^N x \ . \tag{3.2.2}$$

After which the full density function is given by:

$$p(x) = \frac{e^{-S(x)}}{Q} \ . \tag{3.2.3}$$

Note that the so-called quadratic action yields the Gaussian distribution. Let K be the covariance matrix of this Gaussian distribution. Let K^{ij} denote the element on the i^{th} row and j^{th} column of the inverse of the covariance matrix K. Then the quadratic action is given by:

$$S(x) = \frac{1}{2} \sum_{\mu,\nu=1}^{N} K^{\mu\nu} x_\mu x_\nu \ . \tag{3.2.4}$$

$K^{\mu\nu}$ is also referred to as the **quadratic coupling**. The nearly-Gaussian distribution is defined by extending the quadratic action with a quartic term, giving us the quartic action:

$$S(x) = \frac{1}{2} \sum_{\mu,\nu=1}^{N} K^{\mu\nu} x_\mu x_\nu + \frac{\epsilon}{4!} \sum_{\rho_1,\rho_2,\rho_3,\rho_4=1}^{N} V^{\rho_1\rho_2\rho_3\rho_4} \, x_{\rho_1} x_{\rho_2} x_{\rho_3} x_{\rho_4} \ . \tag{3.2.5}$$

The small positive parameter ϵ ensures that the quartic term is a small correction to the base quadratic action. The **quartic coupling** V is an $(N \times N \times N \times N)$-dimensional tensor that encodes interactions between up to four random variables. There is a direct relation between the quartic coupling and the connected 4-point correlator (see eq. (1.78) of [21]). The larger the quartic coupling, the larger the connected 4-point correlators, and hence the more non-Gaussian the distribution becomes.

The quartic action is the simplest member of the family of non-Gaussian distributions, as defined by eq. (1.81) in [21]. The quartic action can be made more complex systematically by adding terms with higher order couplings. Recall from section 1.2 the concept of k-locality, where one restricts a model to incorporate correlations between at most k degrees of freedom. The quartic action corresponds to an assumption of 4-locality. Whether this assumption is valid for modeling the preactivations in a layer of the MLP, depends on the width of the network. As we will see in the next sections, with a large width, the quartic term is a sufficient correction to the infinite-width solution, and we do not need higher order couplings in the action to have an accurate model.

3.3 Enabling the renormalization transformation: perturbation theory

Perturbative analysis has a wide range of applications, including statistical mechanics. To find a perturbative solution to some complex problem, one first considers a much simpler version of the problem, typically making use of some mathematical limit or idealized version of the problem. For example, the ideal gas law can be derived assuming the existence of molecules with zero volume that do not interact with each other, but only with the walls of the container:

$$pV = nRT \ , \tag{3.3.1}$$

where p is the pressure, V is the volume, n is the number of molecules, R is the gas constant, and T is the temperature. The next step in a perturbative analysis is to let go of (some of the) idealized assumptions, meaning we have to perturb the base solution with some correction terms. In the case of the ideal gas law, by assuming the gas molecules *do* have some volume and *do* interact with each other, one obtains the Van der Waals equation:

$$(p + \frac{n^2 a}{V^2})(V - nb) = nRT \ , \tag{3.3.2}$$

where we have introduced the corrections n^2a/V^2 and $-nb$ to the pressure and volume, respectively[8]. A more general way to define a perturbative solution is as follows. To find a solution A to some complex problem, we first solve as simpler version of the problem. This yields a solution A_0. The next step to find A is to perturb the solution A_0 with small correction terms in a power series:

$$A = A_0 + \epsilon\, A_1 + \epsilon^2\, A_2 + \epsilon^3\, A_3 + \dots \tag{3.3.3}$$

Here, we use ϵ to denote a 'small' parameter, and $A_1, A_2, ...$ to denote the corrections. Note that a prerequisite for a perturbative solution to be valid is that the correction terms $\epsilon^k\, A_k$ with $k \geq 1$ are small compared to the base solution A_0. How small these correction terms need to be depends on the context of the problem. Making real-world predictions from a theory requires constants to be known. Thus, to be useful in practice, a perturbative solution needs to be truncated to some degree k to limit the number of constants that must be determined, meaning we only keep the first k correction terms. If the first-order correction term $\epsilon\, A_1$ is relatively large, we need to keep more higher-degree corrections in order for the overall solution A to be accurate. Conversely, if the first order correction term is small enough, we only need to keep this first order correction and can discard the higher order corrections.

Making the link with the authors' theory for MLPs, the base solution is obtained when taking the infinite-width limit: the model $p(z^{(l)})$ becomes completely Gaussian. In the language of actions we have

$$A_0 = \lim_{n\to\infty} S(z^{(l)}) = \frac{1}{2}\sum_{i=1}^{n_l}\sum_{\alpha_1,\alpha_2\in\mathcal{D}} g_{(l)}^{\alpha_1\alpha_2} z_{i;\alpha_1}^{(l)} z_{i;\alpha_2}^{(l)}\ , \tag{3.3.4}$$

where $g_{(l)}^{\alpha_1\alpha_2}$ is the quadratic coupling. Then to make a finite-but-large-width correction to this idealized assumption, we designate the inverse of the network width as the small parameter and use the fact that $v \propto 1/n$:

$$\epsilon\, A_1 = -\frac{1}{8}\sum_{i_1,i_2=1}^{n_l}\sum_{\alpha_1,\dots,\alpha_4\in\mathcal{D}} v_{(l)}^{(\alpha_1\alpha_2)(\alpha_3\alpha_4)} z_{i_1;\alpha_1}^{(l)} z_{i_1;\alpha_2}^{(l)} z_{i_2;\alpha_3}^{(l)} z_{i_2;\alpha_4}^{(l)}\ . \tag{3.3.5}$$

Here, $v_{(l)}^{(\alpha_1\alpha_2)(\alpha_3\alpha_4)}$ is the quartic coupling. The reason for the brackets surrounding its indices, is that the quartic coupling of the MLP exhibits the following symmetries (i.e. invariant under swapping): $\alpha_1 \leftrightarrow \alpha_2$, $\alpha_3 \leftrightarrow \alpha_4$, and $(\alpha_1\alpha_2) \leftrightarrow (\alpha_3, \alpha_4)$. Since the authors assume $n \gg 1$, we can truncate the perturbative series after only one correction term (the quartic coupling). In short, the Van der Waals equation is to the ideal gas law what the quartic action is to the infinite-width Gaussian solution of the MLP.

Finally, recall from section 2.2 that a renormalization transformation is only valid when the model (the nearly-Gaussian distribution) is an accurate approximation at all observation scales (layer depths). For the 2D Ising model, this corresponds to coupling J and K dominating the "other much smaller terms" resulting from the renormalization transformation in eq. (2.2.4). For the MLP, the authors show that the "other much smaller terms", i.e. sextic coupling and higher order couplings in the true action of $p(z^{(l)})$, remain small compared to the quartic coupling, so long as we assume the network width to be large (see section 4.3 of [21]).

3.4 The renormalization transformation: layer recursions

In this section, I give an overview of the recursions that define the renormalization transformation for the MLP, and discuss what intuitions they yield. For the derivations of these recursions, please refer to chapter 4 of [21]. The recursions require a base case, and so we start this section with the first layer of the MLP. Furthermore, there is a qualitative difference between the second layer and all subsequent layers, and so we will discuss them separately.

Recall that the model for the preactivations in a particular layer is a nearly-Gaussian distribution, whose parameters are given by the quadratic and quartic coupling. Combing the infinite-width solution eq. (3.3.4) and finite-width correction eq. (3.3.5), the action of this distribution is given by:

$$S(z^{(l)}) = \frac{1}{2}\sum_{i=1}^{n_l}\sum_{\alpha_1,\alpha_2\in\mathcal{D}} g_{(l)}^{\alpha_1\alpha_2} z_{i;\alpha_1}^{(l)} z_{i;\alpha_2}^{(l)} - \frac{1}{8}\sum_{i_1,i_2=1}^{n_l}\sum_{\alpha_1,\dots,\alpha_4\in\mathcal{D}} v_{(l)}^{(\alpha_1\alpha_2)(\alpha_3\alpha_4)} z_{i_1;\alpha_1}^{(l)} z_{i_1;\alpha_2}^{(l)} z_{i_2;\alpha_3}^{(l)} z_{i_2;\alpha_4}^{(l)}\ . \tag{3.4.1}$$

[8] It is not relevant here, but for the sake of completeness: a is the magnitude of intermolecular forces between gas particles, and b is the excluded volume of one mole of gas molecules.

To emphasize that the quartic action is an approximation whose accuracy is dependent on the large width-assumption, the couplings are further decomposed into:

$$g_{(l)}^{\alpha_1\alpha_2} = K_{(l)}^{\alpha_1\alpha_2} + \mathcal{O}\left(\frac{1}{n}\right) , \tag{3.4.2}$$

$$v_{(l)}^{(\alpha_1\alpha_2)(\alpha_3\alpha_4)} = \frac{1}{n_{l-1}} V_{(l)}^{(\alpha_1\alpha_2)(\alpha_3\alpha_4)} + \mathcal{O}\left(\frac{1}{n^2}\right) , \tag{3.4.3}$$

where $K_{(l)}^{\alpha_1\alpha_2}$ is an element from the inverse of the **kernel** matrix $K^{(l)}$, which defines the covariance matrix[9] for the preactivations at infinite width:

$$\mathbb{E}[z_{i_1;\alpha_1}^{(l)} z_{i_2;\alpha_2}^{(l)}] = \delta_{i_1 i_2}\left(K_{\alpha_1\alpha_2}^{(l)} + \mathcal{O}\left(\frac{1}{n}\right)\right) . \tag{3.4.4}$$

The authors introduce the object $V_{(l)}^{(\alpha_1\alpha_2)(\alpha_3\alpha_4)}$ for convenience, and it is defined as:

$$V_{(l)}^{(\alpha_1\alpha_2)(\alpha_3\alpha_4)} = \sum_{\beta_1,\ldots,\beta_4\in\mathcal{D}} K_{(l)}^{\alpha_1\beta_1} K_{(l)}^{\alpha_2\beta_2} K_{(l)}^{\alpha_3\beta_3} K_{(l)}^{\alpha_4\beta_4} V_{(\beta_1\beta_2)(\beta_3\beta_4)}^{(l)} , \tag{3.4.5}$$

where $V^{(l)}$ is referred to as the **four-point vertex**, which has the same symmetries as the coupling $v_{(l)}$. The four-point vertex encodes the correlations between the preactivations of distinct neurons ($j \neq k$) through:

$$\text{Cov}\left(z_{j;a_1}^{(l)} z_{j;a_2}^{(l)},\, z_{k;a_3}^{(l)} z_{k;a_4}^{(l)}\right) = \frac{1}{n_{l-1}} V_{(\alpha_1\alpha_2)(\alpha_3\alpha_4)}^{(l)} , \tag{3.4.6}$$

and encodes the fluctuations in the preactivations through:

$$\text{Var}\left(z_{i;\alpha_1}^{(l)} z_{i;\alpha_2}^{(l)}\right) = \text{Cov}\left(z_{i;\alpha_1}^{(l)} z_{i;\alpha_2}^{(l)},\, z_{i;\alpha_1}^{(l)} z_{i;\alpha_2}^{(l)}\right) = \frac{1}{n_{l-1}} V_{(\alpha_1\alpha_2)(\alpha_1\alpha_2)}^{(l)} . \tag{3.4.7}$$

In what follows, I will give the recursions/renormalization transformation for the kernel and the four-point vertex, rather than giving recursions for the couplings directly.

The authors find that the distribution of the preactivations in the first layer is Gaussian (for a detailed derivation, see appendix D.). Accordingly, we have no four-point vertex. In the first layer, we have:

$$K_{\alpha_1\alpha_2}^{(1)} = C_b^{(1)} + C_W^{(1)} \frac{1}{n_0} \sum_{j=1}^{n_0} x_{j;\alpha_1} x_{j;\alpha_2} , \tag{3.4.8}$$

$$\frac{1}{n_0} V_{(\alpha_1\alpha_2)(\alpha_3\alpha_4)}^{(1)} = 0 . \tag{3.4.9}$$

In the second layer, the four-point vertex becomes non-vanishing, i.e. the first bit of non-Gaussianity appears. The second layer recursions read:

$$K_{\alpha_1\alpha_2}^{(2)} = C_b^{(2)} + C_W^{(2)} \left\langle \sigma_{\alpha_1}\sigma_{\alpha_2}\right\rangle_{K^{(1)}} , \tag{3.4.10}$$

$$\frac{1}{n_1} V_{(\alpha_1\alpha_2)(\alpha_3\alpha_4)}^{(2)} = \frac{1}{n_1}\left(C_W^{(2)}\right)^2 \left[\left\langle \sigma_{\alpha_1}\sigma_{\alpha_2}\sigma_{\alpha_3}\sigma_{\alpha_4}\right\rangle_{K^{(1)}} - \left\langle \sigma_{\alpha_1}\sigma_{\alpha_2}\right\rangle_{K^{(1)}} \left\langle \sigma_{\alpha_3}\sigma_{\alpha_4}\right\rangle_{K^{(1)}}\right] , \tag{3.4.11}$$

where we have introduced the shorthand notation $\sigma_\alpha \equiv \sigma(z_\alpha)$. In deeper layers, each transition from layer l to $l+1$ generates new non-Gaussianity in layer $l+1$, while layer $l+1$ also inherits the already existing non-Gaussianity from the previous layer. As such, there is an additional term in the four-point vertex that reflects this accumulation of non-Gaussianity:

$$K_{\alpha_1\alpha_2}^{(l+1)} = C_b^{(l+1)} + C_W^{(l+1)} \left\langle \sigma_{\alpha_1}\sigma_{\alpha_2}\right\rangle_{K^{(l)}} , \tag{3.4.12}$$

$$\begin{aligned}\frac{1}{n_l} V_{(\alpha_1\alpha_2)(\alpha_3\alpha_4)}^{(l+1)} &= \frac{1}{n_l}\left(C_W^{(l+1)}\right)^2 \left[\left\langle \sigma_{\alpha_1}\sigma_{\alpha_2}\sigma_{\alpha_3}\sigma_{\alpha_4}\right\rangle_{K^{(l)}} - \left\langle \sigma_{\alpha_1}\sigma_{\alpha_2}\right\rangle_{K^{(l)}} \left\langle \sigma_{\alpha_3}\sigma_{\alpha_4}\right\rangle_{K^{(l)}}\right] \\ &\quad + \frac{1}{4}\left(C_W^{(l+1)}\right)^2 \frac{1}{n_{l-1}} \sum_{\beta_1,\ldots,\beta_4\in\mathcal{D}} V_{(l)}^{(\beta_1\beta_2)(\beta_3\beta_4)} \left\langle \sigma_{\alpha_1}\sigma_{\alpha_2}(z_{\beta_1}z_{\beta_2} - K_{\beta_1\beta_2}^{(l)})\right\rangle_{K^{(l)}} \\ &\quad \times \left\langle \sigma_{\alpha_3}\sigma_{\alpha_4}(z_{\beta_3}z_{\beta_4} - K_{\beta_3\beta_4}^{(l)})\right\rangle_{K^{(l)}} .\end{aligned} \tag{3.4.13}$$

[9] $\mathbb{E}[z_{i_1;\alpha_1}^{(l)} z_{i_2;\alpha_2}^{(l)}] = \text{Cov}(z_{i_1;\alpha_1}^{(l)}, z_{i_2;\alpha_2}^{(l)})$, since $\mathbb{E}[z_{i;\alpha}^{(l)}] = 0$, as shown by eq. (D.0.7) for the first layer. For deeper layers, see chapter 4 of [21].

We have now defined the renormalization transformation of the MLP:

$$\left(K^{(l)}, V^{(l)}\right) \xrightarrow{RT} \left(K^{(l+1)}, V^{(l+1)}\right) \ . \tag{3.4.14}$$

Further analysis by the authors shows that the four-point vertex always grows linearly with layer depth if the initialization hyperparameters are chosen such that there is a critical point in the kernel recursion eq. (3.4.12). I cover these results in section 3.8. Recall that one objective of this chapter is to divide the activation functions into distinct families. Since the linear growth of the four-point vertex at criticality is the same for all the yet-to-be-discussed families, we base the RG flow diagram of the MLP solely on the kernel recursion (leaving the quartic coupling out of the flow diagram). Sections 3.5 to 3.7 discuss how we can obtain the RG flow diagram from the kernel recursions.

3.5 Towards an RG flow diagram: analysis of the kernel

Analysis of the kernel recursion eq. (3.4.12) will form the basis for the RG flow diagram of the MLP. Here, we assume all layers have the same width n and initialization hyperparameters C_W and C_b. The goal is to find out how we should choose the initialization hyperparameters C_b and C_W such that the kernel recursion eq. (3.4.12) has a critical point K^*, defined implicitly through:

$$K^* = C_b + C_W \left\langle \sigma_{\alpha_1} \sigma_{\alpha_2} \right\rangle_{K^*} \ . \tag{3.5.1}$$

For the `linear` activation function $\sigma(z) = z$ only, the kernel recursion simplifies to:

$$K^{(l+1)}_{\alpha_1\alpha_2} = C_b + C_W K^{(l)}_{\alpha_1\alpha_2} \ . \tag{3.5.2}$$

Thus, under the critical tuning $(C_b, C_W) = (0, 1)$, any value $K^{(l)}_{\alpha_1\alpha_2}$ is a fixed point of the recursion. For non-linear activation functions, this is no longer true. Not only does $\langle \sigma_{\alpha_1} \sigma_{\alpha_2} \rangle_{K^{(l)}}$ become a non-linear function of the kernel, the recursion also 'mixes' different elements of the kernel. For example, for the quadratic activation function $\sigma(z) = z^2$ has the following recursion[10]:

$$K^{(l+1)}_{\alpha_1\alpha_2} = C_b + C_W \left(K^{(l)}_{\alpha_1\alpha_1} K^{(l)}_{\alpha_2\alpha_2} + 2 K^{(l)}_{\alpha_1\alpha_2} K^{(l)}_{\alpha_1\alpha_2} \right) \ , \tag{3.5.3}$$

where we see that $K^{(l+1)}_{\alpha_1\alpha_2}$ not only depends on $K^{(l)}_{\alpha_1\alpha_2}$, but also $K^{(l)}_{\alpha_1\alpha_1}$ and $K^{(l)}_{\alpha_2\alpha_2}$. This complicates the search for a critical tuning of C_W and C_b, because we need to find a fixed point for both the diagonal- and off-diagonal entries of the kernel. Since the elements of the kernel involve at most two inputs, let us consider two arbitrary inputs x_+ and x_-. Pretending that these are the only inputs to the network, we write the full kernel as:

$$K^{(l)} = \begin{pmatrix} K^{(l)}_{++} & K^{(l)}_{+-} \\ K^{(l)}_{-+} & K^{(l)}_{--} \end{pmatrix} \ , \tag{3.5.4}$$

noting that $K^{(l)}_{+-} = K^{(l)}_{-+}$ and that our objective is to find a setting of (C_b, C_W) that allows for the existence of a critical point satisfying eq. (3.5.1). To make the problem tractable, the authors first consider the so-called "coincident limit" where both inputs coincide, i.e. $x_+ = x_-$. With the coincident assumption, $K^{(l)}_{++} = K^{(l)}_{+-} = K^{(l)}_{--}$ and thus the critical point takes the form:

$$K^* = \begin{pmatrix} K^*_{00} & K^*_{00} \\ K^*_{00} & K^*_{00} \end{pmatrix} \ . \tag{3.5.5}$$

The coincident limit is equivalent to the problem of an MLP with only a single input x_0, where the full kernel and the critical point are scalars $K^{(l)}_{00}$ and K^*_{00}, respectively (hence the notation in eq. (3.5.5)). We use $K^{(l)}_{00}$ to denote the expected squared magnitude of preactivations for a single arbitrary input (see eq. (3.4.4)).

Next, the authors consider two perturbations to the coincident limit, where the inputs x_+ and x_- are modified such that they are slightly different, as depicted in fig. 9. The parallel perturbation keeps the orientation of

[10] Derived using Wick's theorem [21], [27].

the inputs the same, but gives them a slightly different magnitude. The perpendicular perturbation keeps the magnitude of the inputs (practically) the same, while slightly changing their orientations. The authors introduce two layer-dependent difference metrics to track the downstream effects of these perturbations on the preactivations:

$$R^{(l)} \equiv \mathbb{E}\left[\frac{1}{n_l}\sum_{i=1}^{n_l}\left(z_{i;+}^{(l)}\right)^2\right] - \mathbb{E}\left[\frac{1}{n_l}\sum_{i=1}^{n_l}\left(z_{i;-}^{(l)}\right)^2\right] = K_{++}^{(l)} - K_{--}^{(l)} \ , \tag{3.5.6}$$

$$D^{(l)} \equiv \mathbb{E}\left[\frac{1}{n_l}\sum_{i=1}^{n_l}\left(z_{i;+}^{(l)} - z_{i;-}^{(l)}\right)^2\right] = K_{++}^{(l)} + K_{--}^{(l)} - 2K_{+-}^{(l)} \ . \tag{3.5.7}$$

$R^{(l)}$ tracks the difference between the magnitudes of the preactivations for the two inputs x_+ and x_-, so it quantifies the effect of the parallel perturbation. Conversely, $D^{(l)}$ tracks the magnitude of the difference between the preactivations for the two inputs, quantifying the effect of the perpendicular perturbation. Importantly, $D^{(l)}$ depends on the off-diagonal elements of the kernel. The authors show that the kernel matrix $K^{(l)}$ can be decomposed into a weighted sum of three specially chosen matrices with $K_{00}^{(l)}$, $R^{(l)}$ and $D^{(l)}$ as the coefficients. Following a recursion analysis of this decomposed kernel matrix[11], the authors come to the following conclusions:

- A critical point in the recursion of the full kernel matrix can only exist if the three metrics $K_{00}^{(l)}$, $R^{(l)}$ and $D^{(l)}$ are fixed (invariant under the renormalization transformation) at the critical point.
- The problem of preserving the parallel perturbations is just the problem of the preactivation magnitude $K_{00}^{(l)}$ in disguise, reflected by $R^{(l)}$'s dependence on the diagonal kernel elements only.
- Thus, despite the three metrics, the authors find only two conditions for criticality. One condition ensures that $K_{00}^{(l)}$ and $R^{(l)}$ are preserved at the fixed point, while the other ensure the preservation of $D^{(l)}$.

The authors define the **parallel susceptibility** and the **perpendicular susceptibility**, which control the behavior of the three metrics:

$$\chi_{\parallel}(k) \equiv \frac{C_W}{2k^2}\left\langle \sigma(z)\sigma(z)\left(z^2 - k\right)\right\rangle_k \ , \tag{3.5.8}$$

$$\chi_{\perp}(k) \equiv C_W \left\langle \sigma'(z)\sigma'(z)\right\rangle_k \ , \tag{3.5.9}$$

where k is a scalar and $\langle\cdot\rangle_k$ is a single-variable Gaussian expectation as defined by eq. (3.1.7). The two conditions that ensure that the three metrics are preserved at the critical point, as defined by eq. (3.5.5), are:

$$\chi_{\parallel}(K_{00}^*) = 1 \qquad \text{and} \qquad \chi_{\perp}(K_{00}^*) = 1 \ . \tag{3.5.10}$$

3.6 An algorithm for critical initialization hyperparameters

Using the criticality conditions on the susceptibilities (eq. (3.5.10)), the authors provide, for any choice of activation function, a general algorithm to decide whether a critical tuning of the network exists, and what the values of the critical tuning $(C_b, C_W)^{\text{critical}}$ are. The most naive implementation works as follows (a more efficient implementation is given sec. 5.3.1 of [21]):

1. For each point in the (C_b, C_W) plane, where $C_b \geq 0$ and $C_W > 0$, find a fixed point $K_{00}^* \geq 0$ of the kernel recursion. Note that a fixed point in the kernel recursion might not exist for some points in the (C_b, C_W) plane, in which case we can discard it immediately.
2. Scan over the (C_b, C_W) plane until one finds a point for which a fixed point K_{00}^* exists in the kernel recursion, *and* the conditions $\chi_{\parallel}(K_{00}^*) = 1$ and $\chi_{\perp}(K_{00}^*) = 1$ are met.
3. Verify that the behavior of the kernel recursion is such that it brings points in the neighborhood of the critical point *towards* the critical point, as opposed to further away from the critical point. This ensures that any preactivations whose squared magnitudes are smaller or larger than K_{00}^* are, on average, brought closer to the critical point, avoiding the exploding / vanishing network output problem. Mathematically, since the expected magnitude of the preactivations $K_{00}^{(l)}$ is governed by the parallel susceptibility, we require that for $k > K_{00}^*$ we have $\chi_{\parallel}(k) \leq 1$, and for $k < K_{00}^*$ we have $\chi_{\parallel}(k) \geq 1$.

[11] See section 5.1 of [21].

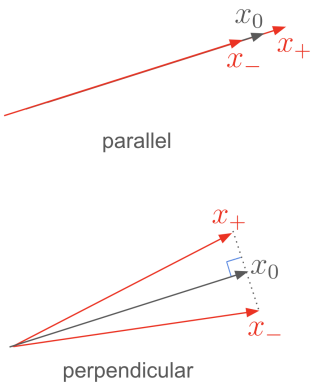

Figure 9: Visualization of the two types of perturbations considered by the authors in their criticality analysis of the kernel. x_0 represents the coincident limit.

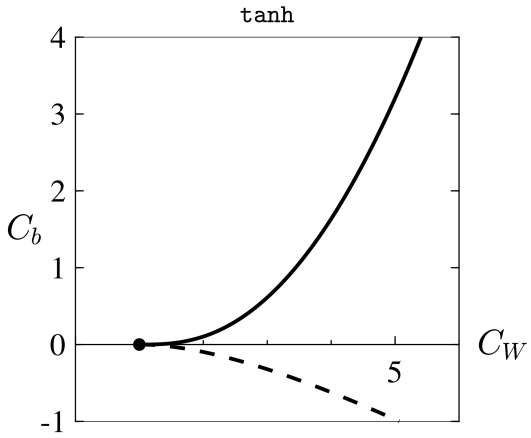

Figure 10: Illustration of step 1 and 2 of the authors' algorithm to determine the existence and value of the critical initialization hyperparameters $(C_b, C_W)^{\text{critical}}$, for the `tanh` function. Solid line: all points (C_b, C_W) for which $\chi_\perp(K_{00}^*) = 1$. Dashed line: all points (C_b, C_W) for which $\chi_\parallel(K_{00}^*) = 1$. The intersection point of these lines corresponds to a candidate critical tuning (which can be accepted/rejected in step 3). Taken from [21].

The first two steps of this naive implementation of the algorithm are illustrated in fig. 10. Note that this algorithm reproduces the Xavier [12] initialization $(C_b, C_W)^{\text{critical}} = (0, 1)$ for `tanh`, and the Kaiming [17] initialization $(C_b, C_W)^{\text{critical}} = (0, 2)$ for the `ReLU` activation function.

3.7 The RG flow diagram & universality classes

Based on the nature of the critical point K_{00}^* for an MLP initialized with critical hyperparameters $(C_b, C_W)^{\text{critical}}$, the authors divide activation functions into universality classes. The behavior of the MLP for activation functions in the same universality class can be described by the same mathematical model. Moreover, the authors find formulas for the critical hyperparameters for two of the universality classes. Thus, if one knows that an activation function belongs to one of these universality classes, there is no need to execute the algorithm discussed in section 3.6, as $(C_b, C_W)^{\text{critical}}$ can be calculated directly from the details of the activation function. Each of the universality classes has a differently looking RG flow diagram. Although the authors do not construct RG flow diagrams themselves, I created them based on the authors' findings, as depicted in fig. 11. As introduced in section 2.3, each universality class is characterized by its own critical exponent(s). In this section, we will see that the metric for the expected squared magnitude of the preactivations has a power-law dependence on the layer depth, taking the form $K_{00}^{(l)} \propto (1/l)^p$ where p is the critical exponent.

The first universality class is that of the scale-invariant activation functions, which satisfy $\lambda\sigma(z) = \sigma(\lambda z)$ for all z, and take the form:

$$\sigma(z) = \begin{cases} a_+ z, & z \geq 0 \\ a_- z, & z < 0 \end{cases} . \tag{3.7.1}$$

Note that this universality class includes the `linear` activation function with $a_+ = a_- = 1$ and the `ReLU` with $a_- = 0$ and $a_+ = 1$. The authors find that scale-invariant activation functions have the property that the susceptibilities $\chi_\parallel(k)$ and $\chi_\perp(k)$ are independent of k, and thus any value of $K_{00}^{(l)}$ is a critical point of the kernel recursion. In other words, the scale-invariant activation functions have a line of critical points in the $K_{00}^{(l)}$ space, as shown in the first RG flow diagram of fig. 11. Since there is no flow in the $K_{00}^{(l)}$ space, there should be no power-law behavior in $K_{00}^{(l)}$, and so the scale-invariant universality class is characterized by critical exponent $p = 0$. The authors show that the critical tuning for scale-invariant activation functions is given by:

$$(C_b, C_W)^{\text{critical}} = \left(0, \frac{1}{A_2}\right) , \quad \text{where} \quad A_2 = \frac{a_+^2 + a_-^2}{2} , \tag{3.7.2}$$

where I should note that, just like the general algorithm, this is consistent with the Kaiming initialization scheme [17] for the ReLU.

The authors find that $\sigma(0) = 0$ and $\sigma'(0) \neq 0$ are necessary conditions for analytic activation functions to allow a critical tuning of the MLP. The second universality class encompasses all analytic functions for which $\sigma(0) = 0$, $\sigma'(0) \neq 0$, and the RG flow in the $K_{00}^{(l)}$ space is directed towards the critical point (this will be made precise shortly). For these activation functions, the critical point is $K_{00}^{*} = 0$, and so their class is referred to as the $K_{00}^{*} = 0$ universality class. The corresponding RG flow diagram is shown in the second row of fig. 11. A critical point at $K_{00}^{(l)} = 0$ might seem counterproductive at first: does this not imply that the magnitudes of the preactivations vanish under RG flow? Indeed, the diagonal kernel entries are brought closer to zero, but due to the critical tuning, they get stuck in the neighborhood of the critical point for networks of practical depths. In this region near the critical point, the authors show that $K_{00}^{(l)}$ has a power-law dependence on the layer depth l:

$$K_{00}^{(l)} = \frac{1}{-a_1}\left(\frac{1}{l}\right)^{p} + \mathcal{O}\left(\frac{\log(l)}{l^2}\right) , \tag{3.7.3}$$

where the meaning of a_1 will become clear shortly, and the critical exponent $p = 1$ is universal for the $K_{00}^{*} = 0$ universality class. Let

$$\sigma(z) = \sum_{p=0}^{\infty} \frac{\sigma_p}{p!} z^p \ , \ \sigma_p = \left.\frac{d^p}{dz^p}\sigma(z)\right|_{z=0} \tag{3.7.4}$$

be the Taylor series around $z = 0$ of an arbitrary analytical function $\sigma(z)$. Then the critical tuning for members of the $K_{00}^{*} = 0$ universality class is given by:

$$(C_b, C_W)^{\text{critical}} = \left(0, \frac{1}{(\sigma_1)^2}\right) . \tag{3.7.5}$$

Let a point $K_{00}^{(l)}$ near the fixed point K_{00}^{*} be expressed as $K_{00}^{(l)} = K_{00}^{*} + \Delta K_{00}^{(l)}$. Then, the authors find that the following recursions hold in the neighborhood of the critical point:

$$\Delta K_{00}^{(l+1)} = \Delta K_{00}^{(l)} + a_1\left(\Delta K_{00}^{(l)}\right)^2 + a_2\left(\Delta K_{00}^{(l)}\right)^3 + O\left(\left(\Delta K_{00}^{(l)}\right)^4\right) , \tag{3.7.6}$$

$$R^{(l+1)} = R^{(l)}\left[1 + 2a_1\Delta K_{00}^{(l)} + 3a_2\left(\Delta K_{00}^{(l)}\right)^2 + O\left(\left(\Delta K_{00}^{(l)}\right)^3\right)\right] , \tag{3.7.7}$$

$$D^{(l+1)} = D^{(l)}\left[1 + b_1\Delta K_{00}^{(l)} + O\left(\left(\Delta K_{00}^{(l)}\right)^2\right)\right] , \tag{3.7.8}$$

where the coefficients a_1, a_2 and b_1 are given by:

$$a_1 = \left(\frac{\sigma_3}{\sigma_1}\right) + \frac{3}{4}\left(\frac{\sigma_2}{\sigma_1}\right)^2 , \tag{3.7.9}$$

$$a_2 = \frac{1}{4}\left(\frac{\sigma_5}{\sigma_1}\right) + \frac{5}{8}\left(\frac{\sigma_4}{\sigma_1}\right)\left(\frac{\sigma_2}{\sigma_1}\right) + \frac{5}{12}\left(\frac{\sigma_3}{\sigma_1}\right)^2 , \tag{3.7.10}$$

$$b_1 = \left(\frac{\sigma_3}{\sigma_1}\right) + \left(\frac{\sigma_2}{\sigma_1}\right)^2 . \tag{3.7.11}$$

Thus, even though the power-law behavior of all K_{00}^{*} activation functions is similar near the critical point, nuances exist, expressed through the coefficients a_1, a_2 and b_1. Furthermore, since near the critical point we have $\Delta K_{00}^{(l)} < 1$, the term $a_1\Delta(K_{00}^{(l)})^2$ in eq. (3.7.6) dominates the higher order terms. Therefore, the sign of a_1 decides the orientation of the RG flow. Since the $K_{00}^{*} = 0$ universality class requires that the flow is oriented towards the critical point, we must have $a_1 < 0$.

Some analytical functions have a nonzero critical point in the kernel recursion. For example, the ReLU-like SWISH and the Gaussian Error Linear Unit (GELU), given by:

$$\sigma_{\texttt{SWISH}}(z) = \frac{z}{1 + \exp(-z)} , \qquad \sigma_{\texttt{GELU}}(z) = \left(\frac{1}{2} + \frac{1}{2}\texttt{erf}\left(\frac{z}{\sqrt{z}}\right)\right) z , \tag{3.7.12}$$

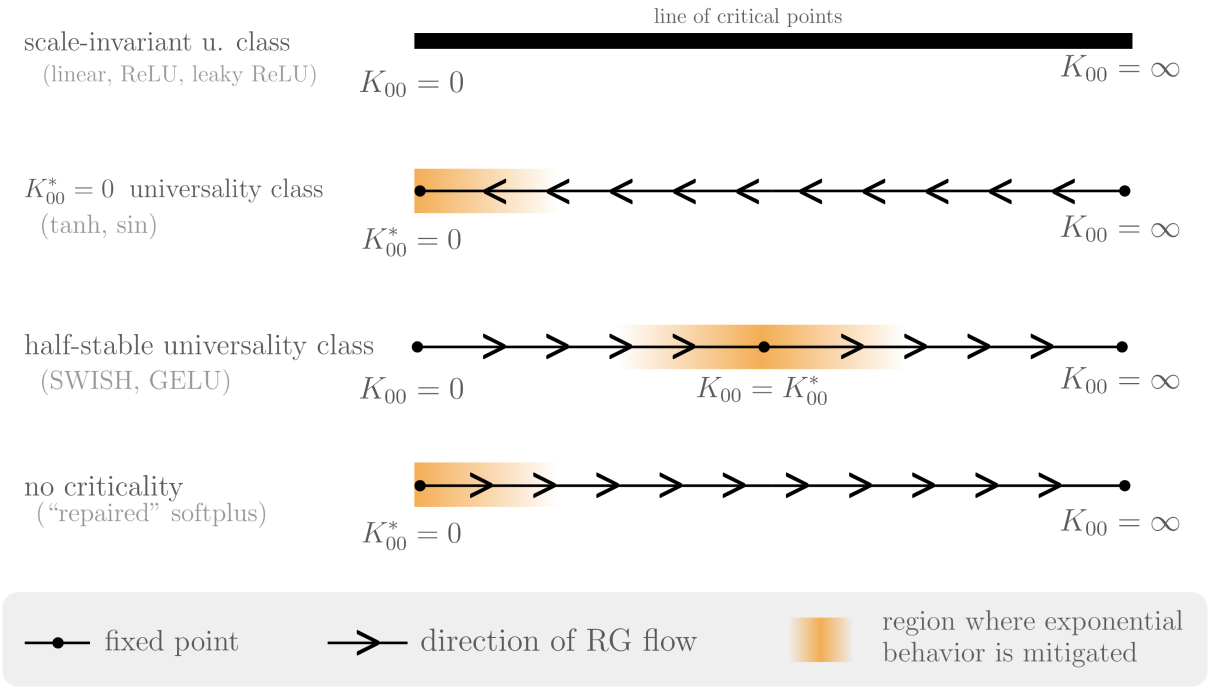

Figure 11: An overview of the RG flow diagram for various universality classes and the shifted softplus, given by $\sigma(z) = \log(1 + e^z) - \log(2)$.

have a fixed point at $K_{00}^{(l)} \approx 14.3$ and $K_{00}^{(l)} \approx 3.56$, respectively. The authors refer to these activation functions as 'half-stable', because the RG flow in the $K_{00}^{(l)}$ space is directed away from the critical point when $K_{00}^{(l)} > K_{00}^*$. The RG flow diagram of the half-stable activation functions is shown on the third row of fig. 11.

Finally, some analytical functions do not allow a critical tuning of the MLP at all. For example, consider another ReLU-like function, the `softplus`, given by:

$$\sigma_{\texttt{softplus}}(z) = \texttt{log}\left(1 + e^{-z}\right) \ . \tag{3.7.13}$$

As is, the `softplus` does not satisfy the condition $\sigma(0) = 0$, which is required for the $K^*_{00} = 0$ universality class. If we shift the `softplus` downwards by a constant $\log(2)$, it does satisfy the conditions $\sigma(0) = 0$ and $\sigma'(0) \neq 0$, and even satisfies the criticality conditions in eq. (3.5.10). However, we find that $a_1 > 0$, and so the RG flow of the `softplus` is oriented away from its critical point at $K^*_{00} = 0$. Therefore, the shifted `softplus` suffers from exponential growth in the magnitudes of the preactivations, no matter the choice of C_b and C_W, and the authors conclude that its use should be discouraged unless the MLP is shallow. The RG flow diagram of the shifted `softplus` is shown on the last row of fig. 11.

The rest of this report focuses on the scale-invariant and $K^*_{00} = 0$ universality classes.

3.8 Finite-width corrections & fluctuations

Recall from section 3.3 that in the infinite-width limit, the distribution of the preactivations is completely Gaussian. In other words, the quartic coupling vanishes, which we can also see from the $1/n$ factor in eq. (3.4.3). Furthermore, from eq. (3.4.7) we see that initialization-to-initialization fluctuations of preactivations' magnitudes also vanish in infinite-width limit. Thus, these fluctuations are a finite-width phenomenon. In this section, I discuss the finite-width corrections to the infinite-width solution of the PDF for the preactivations $p(z^{(l)})$. There are two important corrections. First, the quadratic coupling receives an $\mathcal{O}(1/n)$ correction, which has implications for the critical initialization tunings found in section 3.5. Second, the quartic coupling is nonzero (for $l > 1$) in the finite-width regime, which we already saw in the recursions of the four-point vertex in section 3.4. Here, I give an account of the authors' non-recursive solution.

Recall from eq. (3.4.2) that the kernel defines the infinite-width solution of quadratic coupling, and recall that

the criticality analysis in section 3.5, with its associated tunings for the initialization hyperparameters, relied on the recursions of the kernel. Therefore, some critical tunings found in section 3.5 need to be adjusted to account for the finite-width correction to the quadratic coupling. The critical initialization for the scale-invariant universality class, given by eq. (3.7.2), remains unchanged in the finite-width regime. However, the critical tuning for the $K_{00}^{*} = 0$ universality class must be corrected to:

$$(C_b, C_W)^{\text{critical}} = \left(0, \quad \left[1 + \frac{2}{3n}\right] \frac{1}{(\sigma_1)^2} \right) . \tag{3.8.1}$$

For the derivation of this correction, please refer to section 5.4.2 of [21].

We saw in section 3.4 how finite-width correlations accumulate from layer to layer in the four-point vertex $V^{(l)}_{(\alpha_1\alpha_2)(\alpha_3\alpha_4)}$, which is directly related to the quartic coupling through eqs. (3.4.3) and (3.4.5). Here, I give the authors' non-recursive solution of the four-point vertex recursions, where they only look at a single input α to simplify the math. This single-input assumption also simplifies the notation through:

$$\mathcal{K}^{(l)}_{\alpha\alpha} = K^{(l)} , \qquad V^{(l)}_{(\alpha\alpha)(\alpha\alpha)} = V^{(l)} . \tag{3.8.2}$$

The non-recursive solution for the four-point vertex relies on the large (but finite) width assumption, and also assumes a critical tuning of the initialization hyperparameters. For the scale-invariant activation functions, the authors find:

$$\frac{V^{(l)}}{n\left(K^{(l)}\right)^2} = \left(\frac{3A_4}{A_2^2} - 1\right)\frac{l}{n} + O\left(\frac{1}{n}\right) , \tag{3.8.3}$$

where

$$A_4 = \frac{a_+^4 + a_-^4}{2} . \tag{3.8.4}$$

For the $K_{00}^{*} = 0$ universality class, they find:

$$\frac{V^{(l)}}{n\left(K^{(l)}\right)^2} = \left(\frac{2}{3}\right)\frac{l}{n} + O\left(\frac{\log(l)}{n}\right) . \tag{3.8.5}$$

4 Contributions to the analysis of activation functions

In this chapter I describe some additions to the authors' theory of MLPs at initialization, as well as my proposal for a list of properties that should enable the objective comparison of activation functions. Furthermore, I design two custom polynomial activation functions that optimize some of these properties.

4.1 Rules for activation functions from the $K_{00}^{*}=0$ universality class

The systemic nature of the authors' treatment of the recursions for the metrics $K^{(l)}$, $R^{(l)}$ and $D^{(l)}$ allowed me to (i) add an extra term to the recursion of $D^{(l)}$, and (ii) find some general computational rules for modifications of a given activation function. Adding an extra term with coefficient b_2 to the recursion in eq. (3.7.8), enables the design of an activation function that has $b_1 = 0$. To ensure that the flow in $D^{(l)}$ is oriented towards zero[12], we must then have $b_2 < 0$. The expanded recursion reads:

$$D^{(l+1)} = D^{(l)} \left[1 + b_1 \Delta K_{00}^{(l)} + b_2 \left(\Delta K_{00}^{(l)} \right)^2 + O\left(\left(\Delta K_{00}^{(l)} \right)^3 \right) \right] , \tag{4.1.1}$$

where

$$b_2 = \frac{3}{4} \left(\frac{\sigma_3}{\sigma_1} \right)^2 + \frac{\sigma_2}{\sigma_1} \frac{\sigma_4}{\sigma_1} + \frac{1}{4} \frac{\sigma_5}{\sigma_1} . \tag{4.1.2}$$

To understand the effect of some simple modifications to an activation function on its behavior in terms of $K^{(l)}$, $R^{(l)}$ and $D^{(l)}$, I derived some general rules:

- For even analytic activation functions, $\sigma_1 = 0$ and so the network cannot be tuned to criticality.
- Given an activation function $\sigma(z)$ from the $K_{00}^{*}=0$ universality class, multiplying this function by some constant $\sigma(z) \to \alpha\sigma(z)$ has no effect on a_1, a_2, b_1, or b_2. The only thing that changes is the critical initialization hyperparameter for the weights: $C_W \to C_W/\alpha^2$.
- Given a function $\sigma(z)$ from $K_{00}^{*}=0$ universality class, scaling the function along the z-axis, i.e. $\sigma(z) \to \sigma(\beta z)$ for some real constant β, has the effects of multiplying a_1 and b_1 by a factor of β^2, while multiplying a_2 and b_2 by β^4. Therefore, when $\beta < 1$, this substitution should, in theory, reduce the decay of all metrics three metrics $K_{00}^{(l)}$, $R^{(l)}$, and $D^{(l)}$.

4.2 The $r(k)$ metric: understanding recursions far away from the critical point

The authors' analysis of the single input kernel recursion is valid only in the neighborhood of the critical point. Further away from the critical point, the Taylor expansion on which the authors' analysis of the recursion of $K^{(l)}$ relies is no longer valid. Recall that the recursion of the single input kernel, before doing any expansions, is given by:

$$K_{00}^{(l+1)} = C_b + C_W \left\langle \sigma\left(z\right) \sigma\left(z\right) \right\rangle_{K_{00}^{(l)}} . \tag{4.2.1}$$

We cannot rely on a Taylor expansion to find an equation that is valid at any distance from the critical point, and the presence of the Gaussian expectation $\left\langle \sigma\left(z_0\right) \sigma\left(z_0\right) \right\rangle_{K^{(l)}}$ makes an analytical investigation of the recursion for $K^{(l)}$ challenging. Therefore, I defined a metric that we can compute numerically using the `quad()` function from Scipy's `integrate` module [29]. The metric is found by dividing $K_{00}^{(l+1)}$ by $K_{00}^{(l)}$, where I substitute $k = K_{00}^{(l)}$ to avoid clutter and emphasize the layer independence of the metric:

$$\frac{K_{00}^{(l+1)}}{K_{00}^{(l)}} = \frac{C_b + C_W \left\langle \sigma\left(z\right) \sigma\left(z\right) \right\rangle_{k}}{k} \equiv r(k) . \tag{4.2.2}$$

Thus, as long as $r(k) = 1$, the preactivation magnitudes will be preserved.0 If $r(k) > 1$ for all $k > \kappa$, where κ is some positive constant, we can expect the preactivation magnitudes to explode when the inputs are too big. Conversely, we can expect them to vanish quickly in regions where $r(k) \ll 1$, or to decay slowly in regions where $r(k)$ is close to but smaller than one.

[12] To prevent $D^{(l)}$ from blowing up.

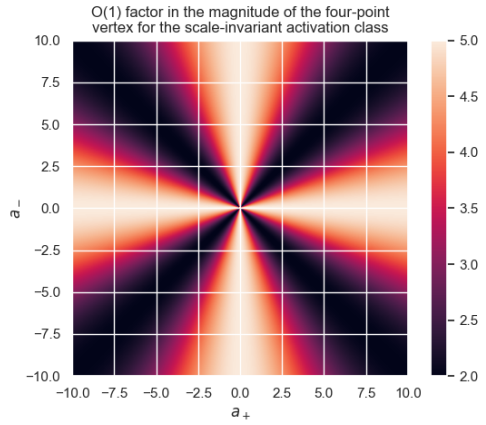

Figure 12: A visualization of the strength of the fluctuations for the scale-invariant activation functions, for different choices of a_+ and a_-.

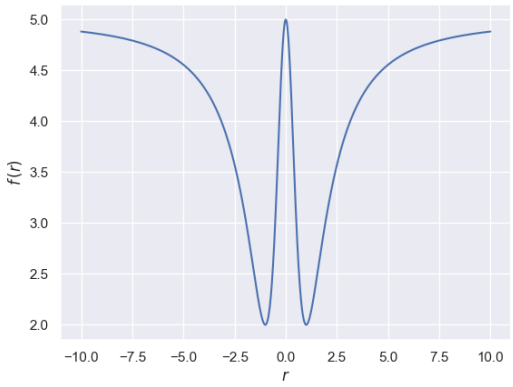

Figure 13: The relation between the ratio $r = a_+/a_-$ and the strength of the fluctuations for the scale-invariant activation functions.

4.3 Minimizing the fluctuations of the scale-invariant activation functions

From eq. (3.8.2) we see that fluctuations in the preactivations of an MLP with scale-invariant activation functions are proportional to the following factor f:

$$f(a_+, a_-) = 3\,\frac{A_4}{A_2^2} - 1, \qquad \text{where } A_2 = \frac{a_+^2 + a_-^2}{2},\ A_4 = \frac{a_+^4 + a_-^4}{2}\ . \tag{4.3.1}$$

The heatmap (fig. 12) already suggests that the best we can do in terms of fluctuations is $f = 2$, whereas the worst we can do is $f = 5$. To see this analytically, suppose we consider the ratio between a_+ and a_-:

$$r = \frac{a_+}{a_-}\ . \tag{4.3.2}$$

Then, by substituting $a_+ = r a_-$ into eq. (4.3.1), we obtain a solution that is independent of a_+ and a_- that only depends on r:

$$f(r) = 3\frac{\frac{1}{2}(1+r^4)\ a_-^4}{\frac{1}{4}(1+r^2)^2\, a_-^4} - 1 = 6\frac{1+r^4}{(1+r^2)^2} - 1\ . \tag{4.3.3}$$

By means of its derivative(s) it we find that this function has two minima $f(r = \pm 1) = 2$ and one maximum $f(r = 0) = 5$. Furthermore, in the limit of $r \to \pm\infty$ the function also evaluates to this maximum value of 5. In this respect, the ReLU at $r = 0$ seems a particularly bad choice for a scale invariant activation function. If we plot $f(r)$ (fig. 13), there is really only a small region where the fluctuations are small. It is only when the absolute values of a_+ and a_- are equal that the fluctuations are minimized. For example, this is the case for linear activations, but if we, as deep learning practitioners, require some form of non-linearity, it seems $\sigma(z) = |z|$ could be an interesting choice, because it also minimizes the fluctuations while being non-linear.

4.4 Properties to facilitate the comparison of activation functions

In this section, I propose a list of properties to facilitate the comparison of activation functions. For future reference, the list of properties is summarized in table table 2. The list incorporates the insights gained from [21] so far, and links the metrics $K_{00}^{(l)}$, $R^{(l)}$ and $D^{(l)}$ to some of the properties enabling quantitative comparisons.

Viewing the MLP as a critical phenomenon, we saw that the critical point divides the two phases of exploding and vanishing network output. Thus, the first property is binary: whether an activation function allows for a critical tuning of the MLP.

Assuming henceforth that we are comparing activation functions at criticality, the second property deals with decay of the preactivation magnitudes. Even when an MLP is tuned to criticality, the expected squared

property	explanation
Enabling a critical tuning of the MLP	A binary property (yes/no). Question can be answered through the algorithm in section 3.6, or based on the activation function's universality class.
Signal stability	The degree to which, at criticality, the activation function manages to preserve the expected squared magnitude of the preactivations $K_{00}^{(l)}$ under RG flow.
Preservation of the training data's structure	The degree to which, at criticality, the off-diagonal entries of the full kernel matrix are preserved under RG flow (in expectation). This behavior is captured in the metrics $R^{(l)}$ and $D^{(l)}$.
Fluctuations	The degree to which, at criticality, the metrics $K_{00}^{(l)}$, $R^{(l)}$ and $D^{(l)}$ tend to fluctuate, from initialization to initialization. This is captured by the magnitude of the four-point vertex $V^{(l)}$.
Gradient stability	The degree to which the activation function mitigates exponential growth or decay of the gradient in the backwards pass of gradient descent, during the early epochs.
Computational cost	The computational cost of the activation function and its derivative.
Non-linearity / model complexity	The degree to which the non-linearity introduced by the activation function enables sufficient model complexity to learn the target function.

Table 2: An overview of properties based on which to compare activation functions.

preactivation magnitude $K_{00}^{(l)}$ might exhibit a decay towards 0. As we saw in section 3.7, this is the case for members of the $K_{00}^{*} = 0$ universality class. Furthermore, my results in chapter 6 will show that, for practical networks, the scale-invariant universality class also displays a decay of $K_{00}^{(l)}$ towards 0. The slower this decay, the deeper we can make the MLP without a loss of signals. Since bigger models generally perform better [19], [22], I decided to include this property in the list and name it "signal stability".

Besides the preservation of the expected magnitude of the preactivations $K^{(l)}$, we also want the MLP to preserve the differences between any two inputs in the training set (at initialization), or else the network loses its ability to compare and contrast inputs. In other words, the off-diagonal entries of the full kernel matrix should not vanish under RG flow, or else the network loses the ability to compare and contrast inputs. The third property, dubbed "preservation of the training data's structure", is the degree to which an activation function preserves the expected difference metrics $R^{(l)}$ and $D^{(l)}$.

The previous two properties dealt with the behavior of the MLP *in expectation*. However, many observables, including $K_{00}^{(l)}$, $R^{(l)}$ and $D^{(l)}$, are subject to fluctuations from initialization to initialization. As we saw in **??**, the strength of the fluctuations grows with layer depth. Therefore, fluctuations are largest at the output (last layer L) of the network. If an activation function is characterized by particularly large fluctuations over initializations, there could be a practically cumbersome percentage of initializations where the MLP exhibits a loss of signal or loss of contrast between the inputs at the output layer, despite using a critical initialization scheme (the critical initialization scheme guarantees that these metrics are well-behaved in expectation). Thus, the fourth property is the strength of the fluctuations.

Even though the theory covered in this report only concerns the MLP at initialization, I think we should consider the first epoch(s) of the training procedure. In the backwards pass of the gradient descent algorithm (backpropagation), during the early epochs, it is important for proper learning to occur in all layers that the gradients of the weights and biases w.r.t. the loss function neither vanish nor explode [12], [15]. Backpropagation relies on the chain rule and the graphical structure of the MLP to recursively calculate the gradient of the weights and biases, meaning that the computation of the gradient in shallow layers involves the product

of the gradients of all deeper layers [18]. Thus, if an activation function has a particularly small or large derivative at typical preactivation values, the gradient of weights and biases in the shallow layers would vanish or explode, respectively. This can slow down or prevent training [12]. Therefore, the fifth property is gradient stability: the degree to which an activation function mitigates exponential growth or decay of the gradient in the backwards pass of gradient descent, during the early epochs.

Another property included in the list is computational cost: not all mathematical operators are equal in terms of clock cycles and ease of computation on the GPU. In this respect, we should consider both the activation function itself and its derivative. Finally, the activation function should be the source of non-linearity (unless the target function is linear), and enable a model complexity that is sufficient for a good training/test performance.

This report does not cover theory nor experiments regarding the last three properties. Nevertheless, I included them to facilitate the discussion in chapter 7.

4.5 Designing custom activation functions

Based on the list of properties introduced in section 4.4, I designed two new activation functions and modified the hyperbolic tangent (`tanh`) and sine (`sin`) activation functions. These new- and modified activation functions should have improved performance in terms of signal stability and the preservation of the training data's structure compared to the unmodified `tanh` and `sin`, quantified by the metrics $K_{00}^{(l)}$, $R^{(l)}$ and $D^{(l)}$.

To mitigate the decay in $K_{00}^{(l)}$ and $R^{(l)}$, I modified `tanh` and sinh by substituting $z \to \beta z$, with $\beta = 0.05$. As discussed in section 4.1, this changes the coefficient: $a_1 \to \beta^2 a_1$ in the recursion of $K_{00}^{(l)}$ and $R^{(l)}$, and thus the decay of the preactivation magnitudes near the critical point should be reduced. The `modified tanh` and `modified sin` are given by:

$$\sigma_{\texttt{modified tanh}}(z) = \texttt{tanh}(\beta z)\ , \qquad \sigma_{\texttt{modified sin}}(z) = \texttt{sin}(\beta z)\ . \tag{4.5.1}$$

Likewise, I designed the two new activation functions to mitigate the decay in the metrics $K_{00}^{(l)}$, $R^{(l)}$, and/or $D^{(l)}$. Since the authors' theory predicts that fluctuations in the metrics $K_{00}^{(l)}$, $R^{(l)}$ and $D^{(l)}$ are always stronger for the scale-invariant universality class than for the $K_{00}^{*} = 0$ universality class, and the $K_{00}^{*} = 0$ universality class has a much broader range of functions, I decided to focus my efforts on designing a $K_{00}^{*} = 0$ activation function. The goal was to design an activation function with smaller fluctuations than the scale-invariant activation function, while mitigating decay of the metrics, which disadvantages, for deep networks, `tanh` and `sin` compared to the scale-invariant activation functions. I decided to use polynomials, because their derivatives are easily manipulated through their coefficients.

The first polynomial, denoted `custom polynomial 1`, mitigates the decay of $K_{00}^{(l)}$ and $R^{(l)}$, disregarding[13] the difference metric $D^{(l)}$, by setting

$$a_1 = 0, \qquad a_2 = -\varepsilon\ , \tag{4.5.2}$$

thereby eliminating the second term in the recursion eq. (3.7.6), and setting a_2 to an arbitrarily small negative value in both eqs. (3.7.6) and (3.7.7). Here, ε is a positive real number, because when setting $a_1 = 0$, it is important that $a_2 < 0$ to ensure the flow of $K_{00}^{(l)}$ points in the direction of the critical point $K_{00}^{*} = 0$. To limit the number of free variables, besides eq. (4.5.2), I also assume:

$$\sigma_1 = 1, \qquad \sigma_2 = 1, \qquad \sigma_5 = 0\ . \tag{4.5.3}$$

Under these assumptions, using eqs. (3.7.9) and (3.7.10), we find

$$a_1 = 0 = \frac{\sigma_3}{1} + \frac{3}{4}\left(\frac{1}{1}\right)^2 \quad \Longrightarrow \quad \sigma_3 = -\frac{3}{4} \tag{4.5.4}$$

$$\Longrightarrow \quad a_2 = -\varepsilon = \frac{1}{4}\left(\frac{0}{1}\right) + \frac{5}{8}\left(\frac{\sigma_4}{1}\right)\left(\frac{1}{1}\right) + \frac{5}{12}\left(\frac{-3/4}{1}\right)^2 \quad \Longrightarrow \quad \sigma_4 = -\left(\frac{3}{8} + \frac{8}{5}\varepsilon\right)\ . \tag{4.5.5}$$

[13] This was my first successful attempt at creating a custom polynomial activation function. The second custom polynomial *does* take into account the difference metric $D^{(l)}$. Nonetheless, I decided to include the first polynomial in this report because it displayed some peculiar behavior in the experiments.

Thus, `custom polynomial 1` is a 4th degree polynomial:

$$\sigma_{\texttt{custom polynomial 1}}(z) = z + \frac{1}{2!}z^2 - \frac{1}{3!}\frac{3}{4}z^3 - \frac{1}{4!}\left(\frac{3}{8} + \frac{8}{5}\varepsilon\right)z^4 \,. \tag{4.5.6}$$

The second polynomial, `custom polynomial 2`, mitigates the decay of all metrics $K_{00}^{(l)}$, $R^{(l)}$ and $D^{(l)}$ by setting

$$a_1 = -\varepsilon_a, \qquad b_1 = 0, \qquad b_2 = -\varepsilon_b \,, \tag{4.5.7}$$

where ε_a and ε_b are small positive real numbers. Note that I expanded the recursion of $D^{(l)}$ to include the term with coefficient b_2, as discussed in section 4.1. Since $b_1 = 0$, we must have $b_2 < 0$ to ensure that the flow is oriented correctly. To limit the number of free variables, besides eq. (4.5.7), I also assume:

$$\sigma_1 = 1, \qquad \sigma_5 = 0 \,. \tag{4.5.8}$$

Plugging all assumptions into eqs. (3.7.9) to (3.7.11) and (4.1.2) and solving for σ_2, σ_3 and σ_4, we find:

$$\sigma_2 = \pm 2\sqrt{\varepsilon_a}, \qquad \sigma_3 = -4\varepsilon_a, \qquad \sigma_4 = -\frac{\varepsilon_b + 12\varepsilon_a^2}{\sigma_2} \,, \tag{4.5.9}$$

which, if we choose $\sigma_2 = +2\sqrt{\varepsilon_a}$, gives the following activation function:

$$\sigma_{\texttt{custom polynomial 2}}(z) = z + \frac{2\sqrt{\varepsilon_a}}{2!}z^2 - \frac{4\varepsilon_a}{3!}z^3 - \frac{\varepsilon_b + 12\varepsilon_a^2}{4! \cdot 2\sqrt{\varepsilon_a}}z^4 \,. \tag{4.5.10}$$

Figure 14 shows the two custom activation functions with $\varepsilon = \varepsilon_a = \varepsilon_b = 0.01$.

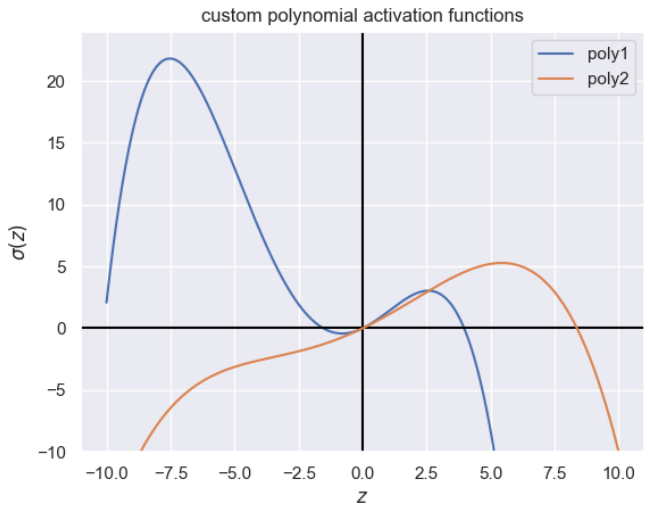

Figure 14: Two custom polynomial activation functions designed to minimize the decay of (some of) the metrics $K_{00}^{(l)}$, $R^{(l)}$, and $D^{(l)}$.

5 Experiments

To (i) verify the predictions of the theory in [21] on MLPs at initialization, (ii) explore potential reasons for any discrepancies found between the theory and experiments, and (iii) study the behavior of the modified `tanh` and `sin` and the custom polynomials, I set up experiments to study the MLP at initialization. For members of both the scale-invariant and K_{00}^{*} universality classes, I ran the same three experiments which recorded the metrics $K_{00}^{(l)}$, $R^{(l)}$ and $D^{(l)}$ and their fluctuations. Unless mentioned otherwise, all experiments involved $N_W = 10,000$ initializations, an MLP of $L = 100$ layers each with $n = 1000$ neurons, where all neurons use the same activation function. For reproducibility, I set the same seed for all experiments.

5.1 Implementation

The experiments were implemented using Python and the PyTorch package [30]. The source code is available on https://github.com/Maarten0110/master-thesis-public. I did not use the standard PyTorch models for the MLP, as I ran into runtime bottlenecks concerning model initialization overhead, weight initialization on the CPU and transferring data between the GPU and CPU. Instead, I developed a custom implementation that does all the work, i.e. weight initializations, the forward pass, and computation of the average metrics and their quantiles, on the GPU. Furthermore, the custom implementation runs in batches that execute multiple initializations, forward passes and metrics computations in parallel. The results are only transferred from the GPU memory to the CPU memory on a per-batch basis. This brought the runtime of an experiment (for a single activation function) down from 3-4 hours to 10-15 minutes. I put the configuration of all experiments in a single Jupyter notebook [31] (filename: `notebook.ipynb`). I made it so that anyone with the "Open in Colab" Chrome extension [32] can open the notebook file on GitHub in their browser and load my entire project in Google Colab [33] with one click. The experiments can then be run from within Colab, although the runtime type must be set to a GPU with enough memory. The first few cells in the notebook fetch the git repository and ask you to mount your Google Drive to store the results of the experiments (which should not be more than a few MBs, unless you explicitly enable the option to store all the preactivation data, which is multiple GBs).

5.2 The magnitudes of preactivations

To study the effect of the activation function on the average squared magnitude of the preactivations in relation to the layer depth, its expected value denoted by $K_{00}^{(l)}$, I initialize the MLP with a single input x_α of size $n_0 = n = 1000$ elements with a Euclidean norm $||x_\alpha|| = (x_\alpha^T x_\alpha)^{1/2}$ that varies depending on the experiment. I set the value of each element of x_α to $x_{i;\alpha} = \sqrt{\frac{1}{n_0}||x_\alpha||^2}$. For each layer l and each initialization t, I record the Euclidean norm of the preactivations:

$$||z_\alpha^{(l;t)}|| = \left(\sum_{i=1}^{n} \left(z_{i;\alpha}^{(l;t)} \right)^2 \right)^{\frac{1}{2}} \tag{5.2.1}$$

Then, I compute the mean of $||z_\alpha^{(l;t)}||$ and the 2.5% and 97.5% quantiles over all initializations $t = 1, 2, \ldots, N_W$. I will refer to this type of experiment as the $K_{00}^{(l)}$ experiments.

5.3 Parallel perturbations

To study the effect of the activation function on how parallel perturbations propagate through the MLP, quantified in the theory by the expected difference-in-magnitude metric $R^{(l)}$, I initialize the MLP with two inputs x_α and x_β, each with $n_0 = n = 1000$ elements. The two inputs are oriented in the same direction, i.e. $\cos(x_\alpha, x_\beta) = 0$, but have different norms. The different inputs are generated by starting with a 'midpoint' vector x_0 with Euclidean norm $||x_0|| = 1$, where each element of x_0 is set to $x_{i;0} = \sqrt{1/n_0}$, and then scaling it such that:

$$x_\alpha = (1 - \varepsilon_R)x_0 \quad \text{and} \quad x_\beta = (1 + \varepsilon_R)x_0 \ , \tag{5.3.1}$$

where $\varepsilon_R < 1$ is a real positive constant. For each layer l and each initialization t, I record the difference of the magnitudes, i.e. eq. (3.5.6) without the expectations:

$$r^{(l;t)} = \frac{1}{n}\sum_{i=1}^{n}\left(z_{i;\alpha}^{(l)}\right)^2 - \frac{1}{n}\sum_{i=1}^{n}\left(z_{i;\beta}^{(l)}\right)^2 \tag{5.3.2}$$

Then, I compute the mean of $r^{(l;t)}$ (to estimate $R^{(l)}$) and the 2.5% and 97.5% quantiles over all initializations $t = 1, 2, \ldots, N_W$. I will refer to this type of experiment as the $R^{(l)}$ experiments.

5.4 Perpendicular / rotational perturbations

The theory dealt with small perpendicular perturbations only, the effect of which on the preactivations is quantified by $D^{(l)}$. However, experimentally it is trivial to test larger perturbations where the angle between the inputs is increased, but the magnitudes are kept the same. For larger angles but unchanged magnitudes, the qualification 'perpendicular' is no longer correct. From here on, I will refer to $D^{(l)}$ as the rotational difference metric, as my experiments involve both small and large angles.

To study the effect the activation function on how rotational perturbations propagate through the MLP, I initialize the network with two inputs x_α and x_β, each with Euclidean norm $||x_\alpha|| = ||x_\beta|| = 1$ and two elements, i.e. $n_0 = 2$. I set the two elements of x_α to $x_{i;\alpha} = \frac{1}{2}\sqrt{2}$. Then, I generate x_β by rotating x_α around the origin:

$$x_\beta = \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix} x_\alpha \,, \tag{5.4.1}$$

where ϕ is the angle in radians. For each layer l and each initialization t, I record magnitude of the differences, i.e. eq. (3.5.7) without the expectation:

$$d^{(l;t)} = \frac{1}{n}\sum_{i=1}^{n}\left(z_{i;\alpha}^{(l)} - z_{i;\beta}^{(l)}\right)^2 \tag{5.4.2}$$

Then, I compute the mean of $d^{(l;t)}$ (to estimate $D^{(l)}$) and the 2.5% and 97.5% quantiles over all initializations $t = 1, 2, \ldots, N_W$. I will refer to this type of experiment as the $D^{(l)}$ experiments.

5.5 Activation functions

For the scale-invariant class, I ran $K_{00}^{(l)}$, $R^{(l)}$ and $D^{(l)}$ experiments for the `ReLU`, `absolute value`, and various `leaky ReLUs`. For the $K_{00}^* = 0$ universality class, I ran the same three experiments on the activation functions listed in table 3. Note that for the modified `tanh` and `sin`, I set $\beta = 0.05$, and for the custom polynomials, I set $\epsilon = \epsilon_a = \epsilon_b = 0.01$. For future reference, table 3 also includes the coefficients a_1, a_2, b_1 and b_2 that should determine the behavior of the three metrics near the critical point. Finally, all experiments used the critical initialization hyperparameters as predicted by the theory: for scale-invariant activations, this is eq. (3.7.2), while for the $K_{00}^* = 0$ activations, this is eq. (3.8.1).

identifier	**formula**	a_1	a_2	b_1	b_2
`tanh`	$\sigma(z) = \texttt{tanh}(z)$	-2	5.67	-2	7
`sin`	$\sigma(z) = \texttt{sin}(z)$	-1	0.67	-1	1
`shifted sigmoid`	$\sigma(z) = \frac{1}{1+e^{-z}} - \frac{1}{2}$	-0.5	0.35	-0.5	0.44
`modified tanh`, $\beta = 0.05$	$\sigma(z) = \texttt{tanh}(\beta z)$	-0.0050	$3.4 \cdot 10^{-5}$	-0.0050	$4.4 \cdot 10^{-5}$
`modified sin`, $\beta = 0.05$	$\sigma(z) = \texttt{sin}(\beta z)$	-0.00025	$4.2 \cdot 10^{-6}$	-0.0025	$6.3 \cdot 10^{-6}$
`custom polynomial 1`	see eq. (4.5.6)	0	-0.010	0.25	0.031
`custom polynomial 2`	see eq. (4.5.10)	-0.01	-0.0063	0	-0.01

Table 3: An overview of the $K_{00}^* = 0$ activation functions used in the experiments.

6 Results

Figures 15 and 16 show the results of the $K_{00}^{(l)}$ experiments for the `linear`, `absolute value`, `ReLU`, and two leaky `ReLU`s. Here I set the norm of the single input $||x_\alpha|| = 1$. We see that the `linear` and `absolute value` functions have the smallest fluctuations, while the ReLU and `leaky ReLU` with $a_- = 0.1$ have the largest fluctuations. If we zoom in on the mean preactivation norms in fig. 16, we see that the scale invariant activation functions exhibit a small decay in the norms. Figures 17 and 18 show the results of the $K_{00}^{(l)}$ experiment for the `ReLU`, with input norm $||x_\alpha|| = 1$, where for fig. 17 the experiment was run at different numbers of initializations, while for fig. 18 the experiment was run at different network widths. Notably, we see no significant difference in behavior of the preactivation norms when the number of initializations is changed, as opposed to changes to the network width: the decay in preactivation norms is stronger for narrower networks. Figure 22 shows the results of the $D^{(l)}$ experiments for the `linear`, `absolute value`, `ReLU`, and multiple leaky `ReLU`s. The plot shows the ratio between $D^{(L)}$ (the last layer) and the $D^{(1)}$ (the first layer). Thus, if the MLP were to perfectly preserve the rotational perturbations, we expect $D^{(L)}/D^{(1)} = 1$, or if they were completely lost we would have $D^{(L)}/D^{(1)} = 0$. I ran the $D^{(l)}$ experiments with angles between inputs $\phi = \frac{\pi}{256}, \frac{\pi}{16}, \frac{\pi}{4}, \frac{\pi}{2}, \pi$. We see that the `absolute value` function has the largest decrease in the ratio $D^{(L)}/D^{(1)}$ for larger angles, followed by the `ReLU` and then the progressively 'leakier' `ReLU`s. Notably, the absolute value function is incapable of preserving an angle of $\phi = \pi$. The `linear` activation function fully preserves the rotational differences at any angle.

Figures 20 and 21 show the results of the $K_{00}^{(l)}$ experiments for the $K_{00}^* = 0$ activation functions listed in table 3. The norm of the single input for these experiments was again $||x_\alpha|| = 1$. Zooming in on the mean preactivation norms in fig. 21, we observe a slight decay in the norms for `tanh`, `sin`, `shifted sigmoid` functions, whereas the modified `tanh` and `sin`, as well as the custom polynomials, all display a slight upwards drift. I repeated[14] the aforementioned $K_{00}^{(l)}$ experiments for the $K_{00}^* = 0$ activation functions with different input norms $||x_\alpha|| = 1, 10, 31.6, 70.7, 141$ such that the average squared magnitudes of the preactivations in the first layer were $K_{00}^{(1)} = 0.0316, 0.1, 1, 5, 10, 20$, respectively. Figure 24 shows the results of these experiments, where the top-left plot is the same as fig. 21 but repeated for convenience. In figs. 24b to 24f we see that at larger input sizes the difference between the standard activation functions (`tanh` and `sin`) and the modified/custom activation functions are much more pronounced. We also see that `custom polynomial 1` appears unstable for $K_{00}^{(1)} \gtrapprox 0.0316$, as the preactivation norms explode at larger input sizes. The same is true for `custom polynomial 2`, but the instability appears at a larger input size, $K_{00}^{(1)} \gtrapprox 5$. I computed the $r(k)$ metric introduced in section 4.2 for the (modified) `tanh`, `sin` and the custom polynomial activation functions. The result is shown in fig. 19. We see that for the (modified) `tanh` and `sin`, we have $r(k) < 1$ for all tested k, while for `custom polynomial 1` we have $r(k) > 1$ if $k > 0$, and for `custom polynomial 2` we have $r(k) > 1$ when $k > \gtrapprox 12$. Figure 23 shows the results of the $D^{(l)}$ experiments for the activation functions listed in table 3. We see no loss of rotational differences at larger angles like we saw for the scale-invariant universality class.

Finally, I also ran $R^{(l)}$ experiments both universality classes to test the behavior of the activation functions for parallel perturbations. However, the shape of these plots turned out identical to the plots from the $K_{00}^{(l)}$ experiments, which is consistent with the fact that both the $K_{00}^{(l)}$ and $R^{(l)}$ metrics are governed by the parallel susceptibility $\chi_\parallel(k)$. Therefore, I included these results in appendix C rather than the main report.

[14] Except for the `shifted sigmoid`, due to time constraints.

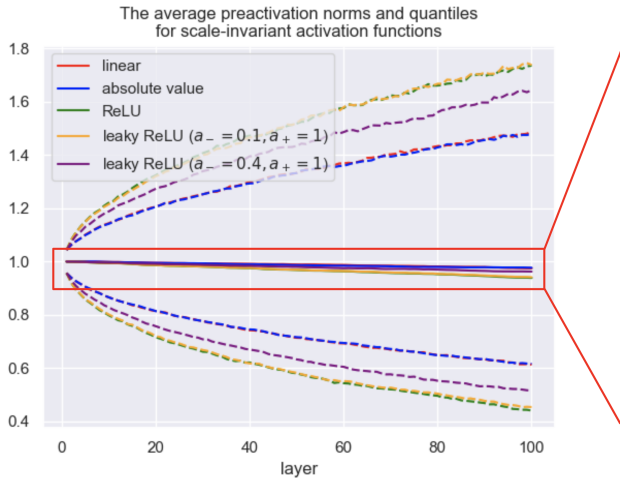

Figure 15: Solid lines: the mean Euclidean norm of the vector of preactivations in each layer, for the scale-invariant activation functions. Dashed lines: the 2.5% and 97.5% percentiles.

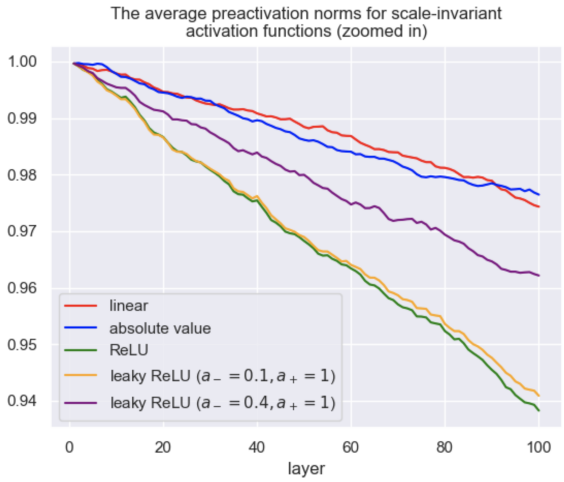

Figure 16: Close up of the mean Euclidean norm of the vector of preactivations in each layer, for the scale-invariant activation functions.

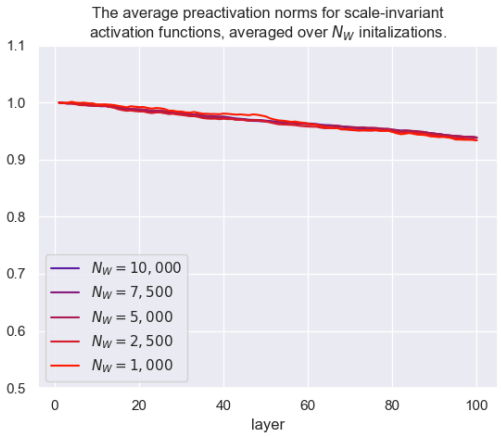

Figure 17: The mean Euclidean norm of the vector of preactivations in each layer, for the `ReLU` activation function, at different numbers of initializations.

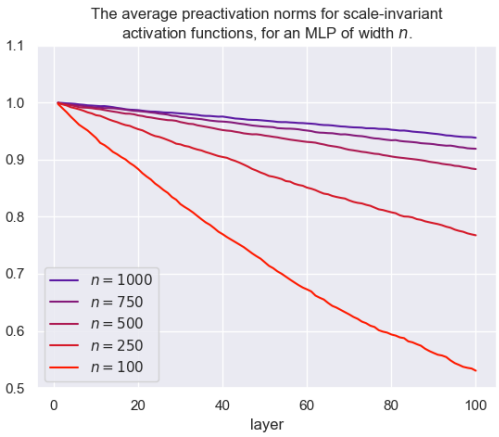

Figure 18: The mean Euclidean norm of the vector of preactivations in each layer, for the `ReLU` activation function, at different network widths.

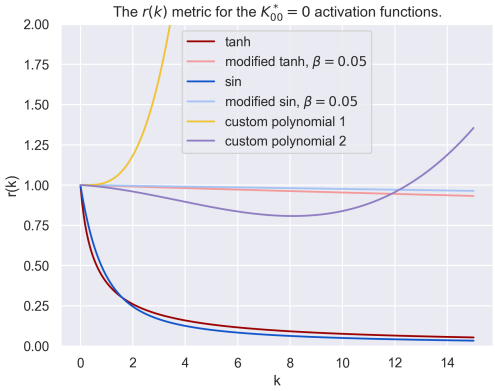

Figure 19: The $r(k)$ metric for various $K^*_{00} = 0$ activation functions.

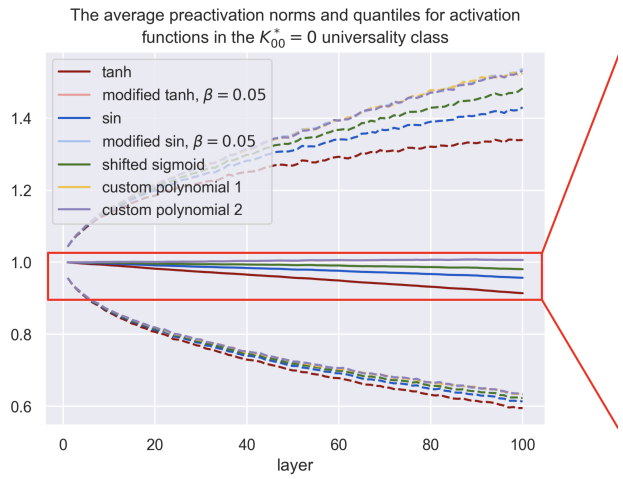

Figure 20: Solid lines: the mean Euclidean norm of the vector of preactivations in each layer, for the $K^*_{00} = 0$ activation functions. Dashed lines: the 2.5% and 97.5% percentiles.

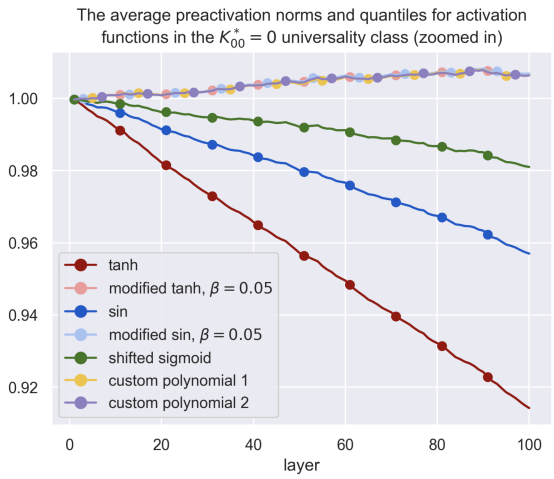

Figure 21: Close up of the mean Euclidean norm of the vector of preactivations in each layer, for the $K^*_{00} = 0$ activation functions. The markers are included to distinguish coinciding graphs. They *do not* represent sample points.

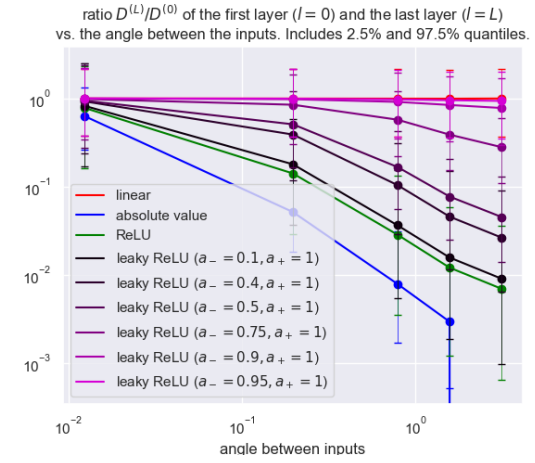

Figure 22: The ratio between $D^{(L)}$ (the last layer) and the $D^{(1)}$ (the first layer) vs. the angle between the two inputs, for the scale-invariant activation functions. The colored I-shapes represent the 2.5% and 97.5% quantiles. Note the logarithmic scale of both axes.

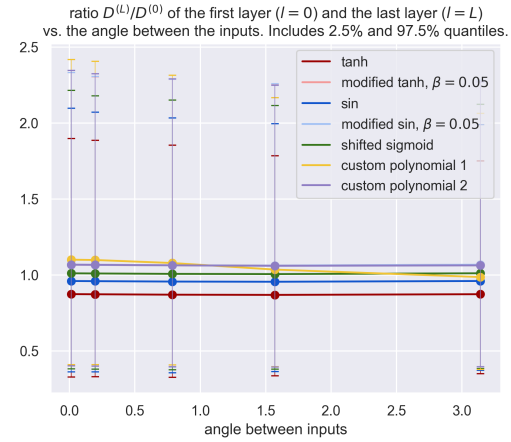

Figure 23: The ratio between $D^{(L)}$ (the last layer) and the $D^{(1)}$ (the first layer) vs. the angle between the two inputs, for the $K^*_{00} = 0$ activation functions. The colored I-shapes represent the 2.5% and 97.5% quantiles.

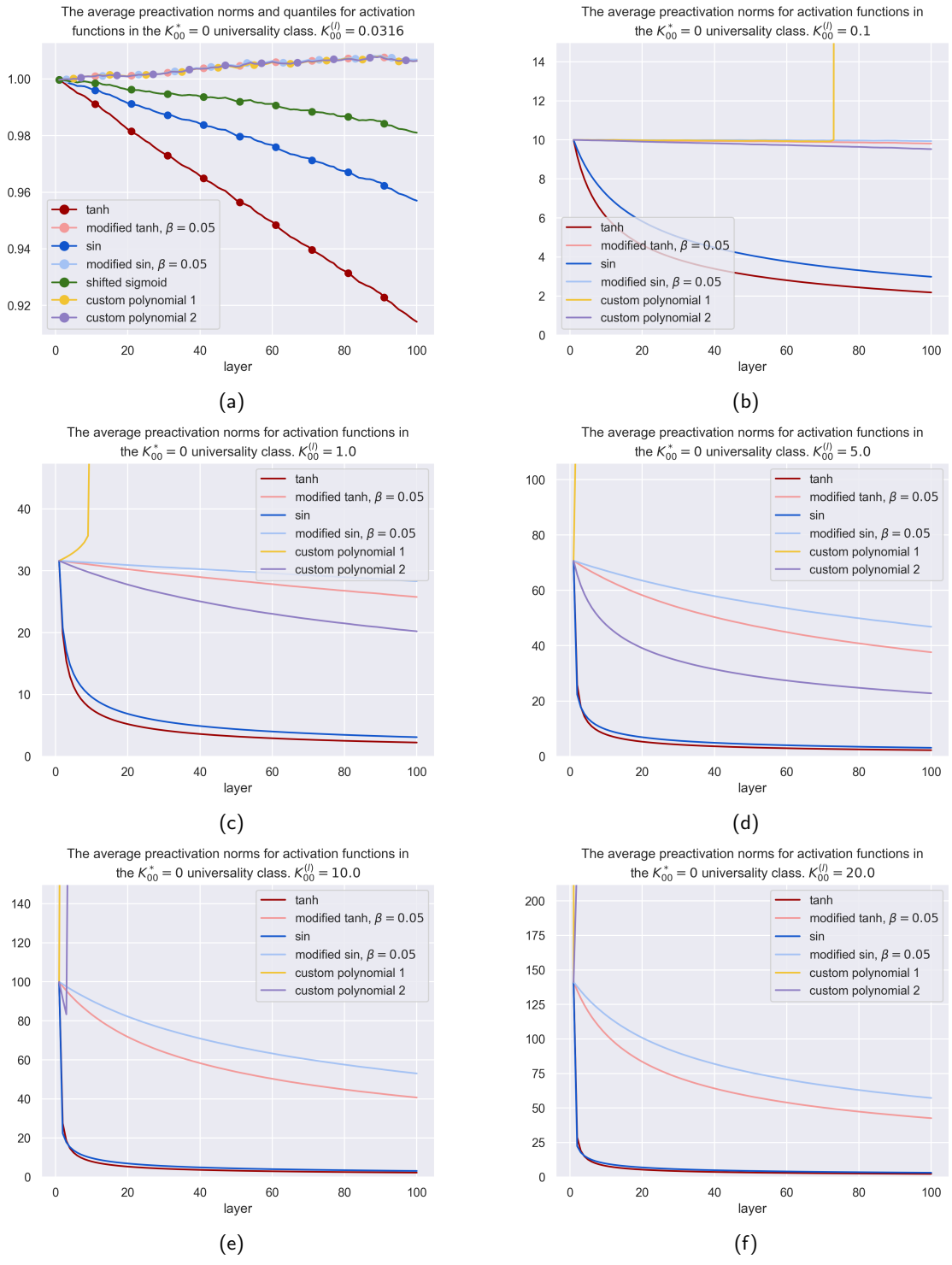

Figure 24: The mean Euclidean norm of the vector of preactivations in each layer, for the $K^*_{00} = 0$ activation functions, at different input sizes.

7 Discussion and conclusion

This final chapter is organized into four parts. In sections 7.1 and 7.2, I discuss whether my experimental results are in line with the theory, and how we might explain discrepancies. Section 7.3 discuss whether an optimal activation function exists, and what it might look like. Furthermore, I discuss whether the custom polynomial activation functions succeeded in their goal. Finally, section 7.4 covers some open questions and interesting avenues to pursue in future work.

7.1 The scale-invariant universality class

As shown in fig. 16, the average squared preactivations $K_{00}^{(l)}$ exhibit a small decay from layer to layer. The kernel analysis described in chapter 3 predicts that any value of $K_{00}^{(l)}$ is a critical point, and thus should be perfectly preserved under the RG transformation. This begs the question of why we observe this slight decay in $K_{00}^{(l)}$. Recall from chapter 3 that the kernel recursion is the infinite-width solution of the RG transformation. Therefore, one hypothesis is that the decay is a finite-width effect not accounted for in the kernel analysis. Another hypothesis is that the decay could be attributed to a sampling effect: the sample mean could be biased because the number of initializations was too small to capture samples from a heavy tail of the true distribution of $K_{00}^{(l)}$. To test these two hypotheses, I ran experiments with the `ReLU` activation function, where I varied the number of initializations and the layer width. The results of which are depicted in figs. 17 and 18. Figure 17 shows that there is no significant difference in the strength of the decay when we vary the number of initializations. Conversely, fig. 18 shows that the decay becomes stronger with decreasing layer width. This is in line with the result from [21] that finite-width effects should become stronger when the layer width decreases. Thus, we can conclude that the slight decay in $K_{00}^{(l)}$ for the scale-invariant universality class is a finite-width effect.

From my analysis in section 4.3, I expect the fluctuations of the `ReLU` to be the largest, while $\sigma(z) = z$ and $\sigma(z) = |z|$ should have the smallest fluctuations. A `leaky ReLU` should have fluctuations that are somewhere in between the two aforementioned cases. This is indeed what we observe in fig. 15. However, it is too soon to conclude that deep learning practitioners should all swap their `ReLUs` for `absolute value` activation functions. When we look at how well these scale-invariant activation functions can preserve the rotational perturbations, as quantified in the metric $D^{(l)}$, we see that the `absolute value` is actually the worst of all scale-invariant candidates, especially for larger angles. The pattern in fig. 22 seems to be that the more your scale-invariant activation function resembles the `linear` activation function, the better it preserves angles between inputs. This is consistent with the result in chapter 3 of [21] that the `linear` activation function is the only activation function to perfectly preserve the full kernel matrix under RG flow. Noteworthy is also the complete collapse of $D^{(l)}$ for the `absolute value` activation function when the angle between inputs $\varphi = \pi$. This is not some bug or numerical artifact, but an effect that can be shown to occur for any input size or layer width, which I do in appendix B.

7.2 The $K_{00}^{*} = 0$ universality class

As we can see in fig. 24a, when $K_{00}^{(l)}$ is near the critical point $K_{00}^{*} = 0$, the custom polynomial activation functions and the modified `tanh` and `sin` are better at preserving $K_{00}^{(l)}$ than the regular `tanh`, `sin` and `shifted sigmoid` (I will refer to these latter three as the 'standard' activation functions from here on out). However, for the custom/modified activation functions, there is a slight upwards drift in $K_{00}^{(l)}$. This is unexpected, because their coefficients a_1 and/or a_2 are such that the RG flow points towards $K_{00}^{*} = 0$. Perhaps the coefficients a_1 and a_2 are so small that the renormalization transformation is dominated by random noise: maybe the upwards drift would be reduced by running the experiments again with more initializations. Alternatively, it could be another finite-width effect. To test these two hypotheses, one could run experiments where the coefficients a_1 and a_2 are larger to increase the decay (and overcome hypothetical noise), experiments with more initializations (to reduce the noise), and experiments where one varies the width of the network. Unfortunately, due to time constraints, I did not run these experiments.

When we increase the input size, such that $K_{00}^{(1)}$ is further away from the critical point $K_{00}^{*} = 0$, the differences between the custom/modified activation functions and the standard activation functions become more pronounced. Figures 24b to 24d show that the standard activation functions quickly bring $K_{00}^{(l)}$ to the

neighborhood of the critical point. The modified `tanh` and `sin` manage to preserve the input size much better. However, the custom polynomials display some peculiar behavior. They appear to be unstable for larger input sizes, at seemingly random layer depths depending on the input size $K_{00}^{(1)}$. In an attempt to understand what is causing this behavior, I computed the $r(k)$ metric which I introduced in section 4.2. Figure 19 shows this metric for the standard- and custom/modified activation functions. Looking at the ranges of k where $r(k) \approx 1$ and $r(k) \leq 1$, we find that we should not expect `custom polynomial 1` to behave non-exponentially anywhere outside the neighborhood of the critical point, while `custom polynomial 2` should be stable so long as $K_{00}^{(1)} < 12$. However, I am unsure why, in fig. 24b, `custom polynomial 1` appears stable until a depth of $l \approx 70$, since I would expect $K_{00}^{(l)}$ to explode from the very first layer, based on the $r(k)$ metric. `Custom polynomial 2` does indeed appear stable for higher values of $K_{00}^{(1)}$ than `custom polynomial 1`, but the point at which it starts to exhibit an instability occurs sooner than expected at $K_{00}^{(1)} = 10$, instead of at $K_{00}^{(1)} = 12$, as predicted by the $r(k)$ metric. Perhaps this discrepancy is another finite-width effect? For `custom polynomial 1`, there also seems to be a conflict between the authors' recursion of ΔK (eq. (3.7.6)) and the $r(k)$ metric. Based the fact that $a_1 = 0$ and $a_2 = -\epsilon = -0.01$, we would expect the RG flow near the critical point to point towards the critical point. However, even if we compute the $r(k)$ metric for values very close to $k = 0$, we still find $r(k) > 1$. Thus, the $r(k)$ metric predicts that the flow is oriented away from the critical point while a_1 and a_2 predict that the flow is oriented towards the critical point. Perhaps the linearization around the critical point, which the recursion of eq. (3.7.6) relies on, is not valid all analytical functions?

Finally, as fig. 23 shows, the scale-invariant activation functions generally preserve rotational perturbations, exemplified by the preservation of the $D^{(l)}$ metric.

7.3 The optimal activation function?

In the introduction of this report, I posed the question "What is the optimal activation function for an MLP?" In section 4.4, I proposed a list of properties based on which we can compare activation functions. Here, I go through the list of properties and discuss whether a notion of optimality exists, and if we have any reason to favor some activation functions over others, based on both the list of properties and experiments.

First, if the optimal activation function exists, certainly it must enable a critical tuning of the network with either:

- a line of critical points (as is the case for the scale-invariant universality class),
- or one critical point with the RG flow pointing towards the critical point,

where the susceptibilities satisfy $\chi_\perp(K_{00}^*) = \chi_\parallel(K_{00}^*) = 1$. This ensures that we have signal stability and that the training data's structure is preserved under RG flow, i.e. the metrics $K_{00}^{(l)}$, $R^{(l)}$ and $D^{(l)}$ are preserved or exhibit a mild non-exponential decay. A theoretical optimal activation function should perfectly preserve the metrics $K_{00}^{(l)}$, $R^{(l)}$ and $D^{(l)}$ and exhibit no decay. This covers the first three properties of the list.

The fourth property, fluctuations, is about the typical behavior of the network. A critical tuning ensures that the MLP exhibits signal stability and the preservation of the training set's structure, *in expectation*. However, the corresponding metrics fluctuate from initialization to initialization. For practical purposes, the probability that a network is unstable despite a critical tuning of the initialization hyperparameters should be small. Therefore, it is tempting to conclude that a theoretical optimal activation function exhibits no fluctuations. However, the authors show in chapter 4 of [21] that fluctuations are an inherent property of finite-width networks. Only infinite-width networks have no fluctuations, and the wider the network, the smaller the fluctuations. Based on the activation function alone, we have limited control over the fluctuations. As demonstrated in section 4.3, within the scale-invariant universality class, we have control over factor $f(a_+, a_-)$ with a fixed range $f(a_+, a_-) \in [2, 5]$, by tinkering with the details of the particular activation function. The authors' theory provides no such way to manipulate the fluctuations for the $K_{00}^* = 0$ universality class, although members in this class have smaller fluctuations than scale-invariant functions by default. To conclude, the fluctuations seem to me like a property of the network dimensions[15] first, and of the activation function's universality class second. Note that, even if there was a universality class that gave more control over fluctuations, or if we could make a network extremely wide, as to get rid of the fluctuations, this would not be desirable.

[15] Width n and depth L.

The fluctuations are proportional to the magnitude of the quartic coupling. Reducing the fluctuations to zero means that the quartic coupling vanishes. As we saw in eq. (3.4.6), the four-point vertex (and thus the quartic coupling) encodes correlations between the magnitudes of preactivations of distinct neurons. Furthermore, the authors describe in the epilogue chapter of [21], that the typical size of the quartic coupling is related to the complexity of the function that the MLP can learn. Therefore, there is a trade-off between the network's fluctuations from typicality and the model complexity that can be achieved during training. Finally, based on the 'fluctuations' property alone, the theoretical optimal activation function would be a member of the $K^*_{00} = 0$ universality class to achieve smaller fluctuations[16], but only if the activation function offers 'enough' model complexity for the particular problem at hand.

At this point of the discussion, we can take stock and see which real (non-hypothetical) activation functions are still in the race to be crowned 'optimal'. As described in section 7.1, even though the kernel analysis predicts that scale-invariant activations functions should perfectly preserve the average squared preactivations, we observe a small decay $K^{(l)}_{00}$ in the experiments, which I show to be a finite-width effect. As such, both the scale-invariant and $K^*_{00} = 0$ universality class exhibit a decay in $K^{(l)}_{00}$. Based on the experiments with the scale-invariant activation functions shown in figs. 15 and 16, it seems that the decay in $K^{(l)}_{00}$ is correlated with the fluctuations, with smaller fluctuations correlated with a smaller decay in $K^{(l)}_{00}$. Assuming this is true, the best case scenarios in terms of fluctuations, $\sigma(z) = z$ and $\sigma(z) = |z|$, still display a decay in $K^{(l)}_{00}$. Conversely, the $K^*_{00} = 0$ universality class provides more control over the decay in $K^{(l)}_{00}$: from fig. 24 we see that the custom polynomials and modified `tanh` and `sin` exhibit no decay in $K^{(l)}_{00}$ near the critical point. This is one reason to prefer the $K^*_{00} = 0$ universality class over the scale-invariant universality class, although the small decay would only become problematic in really deep networks. Comparing fig. 22 and fig. 23, a better reason to prefer the $K^*_{00} = 0$ universality class is that its members preserve the rotational difference metric $D^{(l)}$ much better, at all angles. Thus, unless the computational cost or gradient stability turn out to be problematic, the optimal activation function is from the $K^*_{00} = 0$ universality class.

Within the $K^*_{00} = 0$ universality class, there are differences among the class members to consider. Looking at fig. 24, the standard activation functions exhibit a much stronger decay in $K^{(l)}_{00}$ than the custom/modified activation functions, so the standard activation functions are out the proverbial optimality race. Of the remaining functions, the modified `tanh` and `sin` appear to be the most sensible choice, because they do not explode at larger input sizes. However, when we consider their gradient at and around $z = 0$, we see that $\sigma_1 = \beta = 0.05$. Therefore, I suspect that during training, the modified `tanh` and `sin` would suffer from the vanishing gradient problem. The astute reader might wonder if we could compensate the small derivative at $z = 0$ by multiplying the modified `tanh` and `sin` by a factor $1/\beta$, since I stated in section 4.1 that this has no effect on the recursion coefficients a_1, a_2, b_1 and b_2. However, what are we really doing when we substitute $z \to \beta z$ and multiply the activation function by a factor $1/\beta$? If $\beta \ll 1$, we are massively increasing the region around $z = 0$ where the activation function is (nearly) linear. This is another instance[17] where tending towards the linear activation function yields the best result if it were not for the need for non-linearity. It seems introducing non-linearity has an unavoidable cost. This cost might acceptable in the case of `custom polynomial 2`. Like the modified `tanh` and `sin`, it has a much smaller decay in $K^{(l)}_{00}$ than the standard activation functions, but it has a gradient at $z = 0$ of $\sigma_1 = 1$. Instead of the vanishing gradient problem, the cost we pay for using `custom polynomial 2` is its instability (exploding network output) when $K^{(l)}_{00} \gtrapprox 10$, but this seems like an easily avoidable problem by some preprocessing step on the training data.

In the end, I do not expect the minute differences in non-exponential (power-law) decay of the metrics $K^{(l)}_{00}$, $R^{(l)}$ or $D^{(l)}$ to be the most important factor to consider when choosing (or designing) a $K^*_{00} = 0$ activation function. Furthermore, there are issues that I did not (fully) explore, such as the vanishing/exploding gradient problem, computational cost, and whether all types of non-linearity are equal in terms of model complexity. Results in [12] suggest that the shape of the activation function has a nontrivial effect on learning and test performance. Thus, we cannot conclude that there exists one "optimal" activation function. In my opinion, the value of the theory of the authors and my contributions to the analysis of activation functions lies more in:

- Viewing the MLP at initialization as a critical phenomenon, enabling the analysis of the MLP through

[16] Smaller than the fluctuations of scale-invariant activation functions.

[17] We saw this trend before in section 7.1 when looking at the metric $D^{(l)}$, which was better preserved at larger angles by activation functions that were (almost) linear.

renormalization group theory. The RG flow framework provides a way of thinking about- and describing the MLP that is very insightful: it gives us the RG flow diagram, the notion of criticality and the universality classes.

- Understanding what are some fundamental properties of activation functions, as summarized in table 2.
- Having quantitative measures of an activation function's performance for some of these properties, such as $K_{00}^{(l)}$, $R^{(l)}$ or $D^{(l)}$.
- A generally applicable algorithm to determine whether any potential activation function has an associated critical tuning of the initialization hyperparameters, and what is the value of the initialization hyperparameters to achieve criticality. This allows deep learning practitioners to build their own activation functions suited to their needs *and* have a systematic way of finding the correct tuning of C_b and C_W.
- Understanding the effects of modifying an existing activation function, as described in section 4.1.

7.4 Future work

In section 7.1, we saw that in the case of scale-invariant activation functions, the rotational distance metric $D^{(l)}$ is preserved at all angles only by the `linear` activation function. Furthermore, the more a `leaky ReLU` resembles the `linear` activation function, i.e. the smaller the difference between a_- and a_+, the better it is at preserving $D^{(l)}$. Therefore, I wonder how much of a 'kink' in a `leaky ReLU` is enough to be able to learn the same target functions as MLPs with the standard `ReLU`. Certainly this could be investigated experimentally, but perhaps there is also a theoretical argument that shows that a standard `ReLU` can be emulated by a certain configuration of `leaky ReLUs`?

On a similar note, in section 7.3 I described how for $K_{00}^* = 0$ activation functions the substitution $z \to \beta z$ with $\beta \ll 1$ amounts to increasing the region around $z = 0$ where the activation is (nearly) linear. Similarly, `custom polynomial 2` is fairly linear in the neighborhood of $z = 0$. It would be interesting to see if an activation function with a large 'linear region' comes at the cost of model complexity or training/test performance. Perhaps the large (nearly) linear region prevents the MLP from learning a non-linear target function, or perhaps the gradient descent algorithm is able to adjust the weights to make use of the non-linearity outside this region whenever needed?

Additionally, there are still some open questions concerning the behavior of the custom polynomials and the modified `tanh` and `sin`. First, it would be interesting to find the cause of the upwards drift of $K_{00}^{(l)}$ near the critical point, as observed in fig. 24a. Is it a sampling effect, a finite-width effect, or is there some other yet to be determined cause? Second, I would love to know why the custom polynomials, outside the neighborhood of the critical point, exhibit exploding behavior at seemingly random layer depths.

I did not look into a first-principled theory of the vanishing/exploding gradient problem, nor did I run experiments to determine the expected gradient and its fluctuations for the various activation functions that I considered in my thesis. It would be interesting to see if the modified `tanh` and `sin` indeed suffer from the vanishing gradient problem, and if `custom polynomial 2` does better in comparison. Perhaps we could impose additional constraints on the design of a custom (polynomial) activation function that mitigate the vanishing/exploding gradient problem? Would these constraints contradict the constraints imposed in section 4.4, in which case we may find another trade-off?

Another avenue to explore is the design of custom activation functions that are not polynomial. I chose to base my custom activation functions on the polynomial, because its derivatives are easy to manipulate. Perhaps other types of functions can mitigate the decay in the three metrics while also not suffer from the instabilities for large inputs?

Finally, I wonder how the three metrics $K_{00}^{(l)}$, $R^{(l)}$, and $D^{(l)}$ at initialization affect the learning process. Do networks that better preserve the difference metrics better test performance or shorter training time?

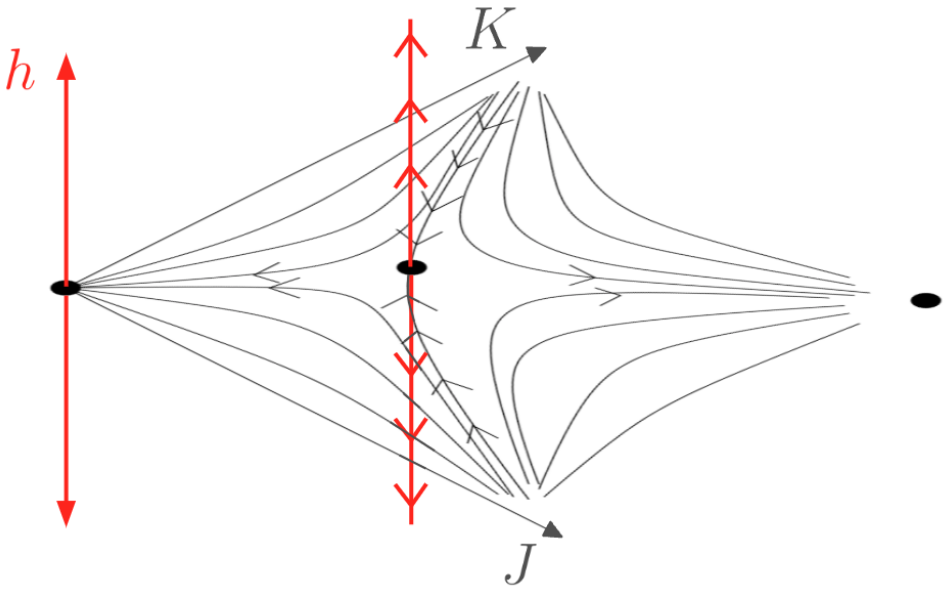

Figure 25: The RG flow diagram for the 2D Ising model with interactions J and K, and external magnetic field h. A nonzero external magnetic field destroys the critical behavior of the system.

A The 2D Ising model: critical exponents, scaling fields, (ir)relevant interactions

Starting with the critical exponents, in order to define them, we need to expand our model to include an external magnetic field h that points either upwards, downwards, or is zero:

$$H(\{s_i\}) = -\frac{1}{\beta}\left(J\sum_{\langle i,j\rangle} s_i s_j + K\sum_{\langle\langle i,j\rangle\rangle} s_i s_j + h\sum_i s_i \right) \tag{A.0.1}$$

A nonzero external magnetic field destroys any critical behavior: the inclusion of the external magnetic field expands the RG flow diagram to three dimensions, where the flow near the critical point in the h dimension is perpendicular to the J, K plane and points away from the critical point. This is depicted in fig. 25 Near the critical point, there are several quantities that are governed by power laws. In order to explain the concept of critical exponents, (ir)relevant interactions/fields, and universality, I need to list them here:

$$\begin{aligned} m(T) &\propto (T_c - T)^{\beta} && \text{for } T < T_c \\ m(h) &\propto h^{1/\delta} && \text{for } T = T_c \\ \chi &\propto |T - T_c|^{-\gamma} \\ C_h(T) &\propto |T - T_c|^{-\alpha} \\ g(r) &\propto 1/(r^{\,d-2+\eta}) \\ \xi &\propto |T - T_c|^{-\nu} \end{aligned} \tag{A.0.2}$$

Where m is the magnetization, χ is the magnetic susceptibility, C_h is the specific heat, $g(r)$ is the correlation function, ξ is the correlation length, and d is the number of dimensions of the system (in our case, $d = 2$).

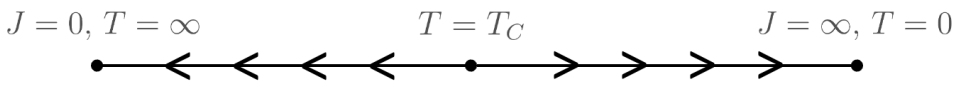

Figure 26: The RG flow diagram for the 2D Ising model with only nearest-neighbor interactions. There are two trivial fixed points at $J = 0$ and $J = \infty$. There is one nontrivial fixed point, also referred to as a critical point, at the critical temperature T_C.

The reader need not be familiar with all these physical quantities. What is important to our purposes, is that, near the critical point, all these quantities are governed by power laws with corresponding critical exponents α, β, γ, δ, η, and ν.

Next, we discuss the potential irrelevance of interactions/fields included in a model. Consider the 2D Ising model of a magnetic system with nearest- and next-to-nearest-neighbor interactions and an external magnetic field, as given by eq. (A.0.1). The parameter space of this model is the three-dimensional J, K, h-space. We could also have modeled the system using only nearest-neighbor interactions:

$$H(\{s_i\}) = -\frac{1}{\beta}\left(J\sum_{\langle i,j\rangle} s_i s_j + h\sum_i s_i\right) \tag{A.0.3}$$

Setting the external magnetic field to zero to explore critical phenomena, we obtain a 1D flow diagram under RG flow, as the only changeable parameter in the model is J. The RG flow diagram for this scenario is depicted in fig. 26. Remarkably, the RG flow diagram has the same fixed points as the more complicated model, i.e. this model predicts the same macrostates. Moreover, the critical exponents for both models are the same. By definition, this makes the inclusion of the next-to-nearest-neighbor interactions in the model irrelevant, but how could we have predicted this based on the RG flow?

Consider the model with the interaction couplings J and K, and the external magnetic field h. Since any nonzero magnetic field destroys the critical behavior and brings us to the trivial fixed points $h \to \pm\infty$ under RG flow, we initially focus on the J, K plane at $h = 0$. Let the critical point of its RG flow be denoted as (J^*, K^*), and a point near the critical point as $(J^* + \Delta J, K^* + \Delta K)$, where ΔJ and ΔK are small. A renormalization transformation gives us a new point:

$$(J^* + \Delta J, K^* + \Delta K) \xrightarrow{RT} (J^* + \Delta J', K^* + \Delta K') \tag{A.0.4}$$

Since we started close to the critical point, the distance we move in the J, K-space is small. Therefore, we can linearize the RG transformation close to the critical point:

$$\begin{pmatrix} \Delta J' \\ \Delta K' \end{pmatrix} = A \begin{pmatrix} \Delta J \\ \Delta K \end{pmatrix} \tag{A.0.5}$$

where A is a 2 by 2 matrix. If we diagonalize A we get eigenvector $\hat{s}$ with eigenvalue λ and eigenvector $\hat{t}$ with eigenvalue μ. The eigenvectors are depicted in fig. 27. $\hat{s}$ and $\hat{t}$ are called **scaling fields**. We have $\lambda < 1$, meaning that any vector in the same (or opposite) direction as $\hat{s}$ shrinks after one renormalization transformation. Thus, at the critical point, $\hat{s}$ corresponds to the direction along the separatrix. Conversely, we have $\mu > 1$, meaning any vector in the same (or opposite) direction as $\hat{t}$, grows after one renormalization transformation. One can repeat the process we just did, including the external magnetic field h. This yields a third eigenvector $\hat{h}$ with eigenvalue ν, that points in the direction perpendicular to the J, K plane. Now, we can re-express any point in the J, K, h-space in terms of $\hat{s}$, $\hat{t}$, and $\hat{h}$:

$$\begin{pmatrix} J \\ K \\ h \end{pmatrix} = c_1\,\hat{s} + c_2\,\hat{t} + c_3\,\hat{h}. \tag{A.0.6}$$

Let l be the lattice constant: the distance between horizontal/vertical neighbors in the lattice. According to Wegner's theorem [23], we have:

$$\lambda = l^y, \qquad \mu = l^z, \qquad \nu = l^v \tag{A.0.7}$$

where $y < 0$, $z > 0$, and $v > 0$. The exponents y, z and v are called **scaling dimensions**. Through derivations that are outside the scope of this report, it can be shown that the critical exponents related by a set of equations that *depend on z and v, but not on y*:

$$\begin{aligned} \alpha + 2\beta + \gamma &= 2 \\ 2 - \alpha &= d\nu \\ \beta(\delta - 1) &= \gamma \\ \nu(2 - \eta) &= \gamma \\ (d - v)\nu &= \beta \end{aligned} \tag{A.0.8}$$

Therefore, the scaling fields $\hat{t}$ and $\hat{h}$ are proclaimed **relevant**, while the scaling field $\hat{s}$ is designated **irrelevant**. Furthermore, through derivations that are once again outside the scope of this report, we can obtain the following intuition about the model near the critical point:

- Moving along the scaling field $\hat{s}$ changes the relative importance of J and K, but not the macroscopic behavior of the system.
- Moving along the scaling field $\hat{t}$ corresponds to changing the temperature, and we have $c_2 \propto T - T_c$.
- Moving along the scaling field $\hat{h}$ corresponds to changing the external magnetic field, and we have $c_3 \propto h$.

To summarize, in order to find out which interactions / fields included in a model are (ir)relevant:

1. We linearized the RG transformation near the critical point.
2. We determined the eigenvectors of the resulting matrix, which correspond to the scaling fields of the model.
3. If the scaling dimensions of a scaling field is positive, the scaling field is deemed relevant. The interactions/fields in our model that contribute to this scaling field are crucial to the macroscopic behavior of the system.
4. If the scaling dimension of a scaling field is negative, the scaling field is deemed irrelevant. The terms that gave rise to this scaling field are not important for the macroscopic behavior of the system.

We can add as many irrelevant terms to our model as we desire, but they will never change the critical exponents or their interdependencies. This leads us to the concept of **universality**: all models that differ from each other only by irrelevant terms, are said to be governed by the same critical point and thus, necessarily, belong to the same universality class.

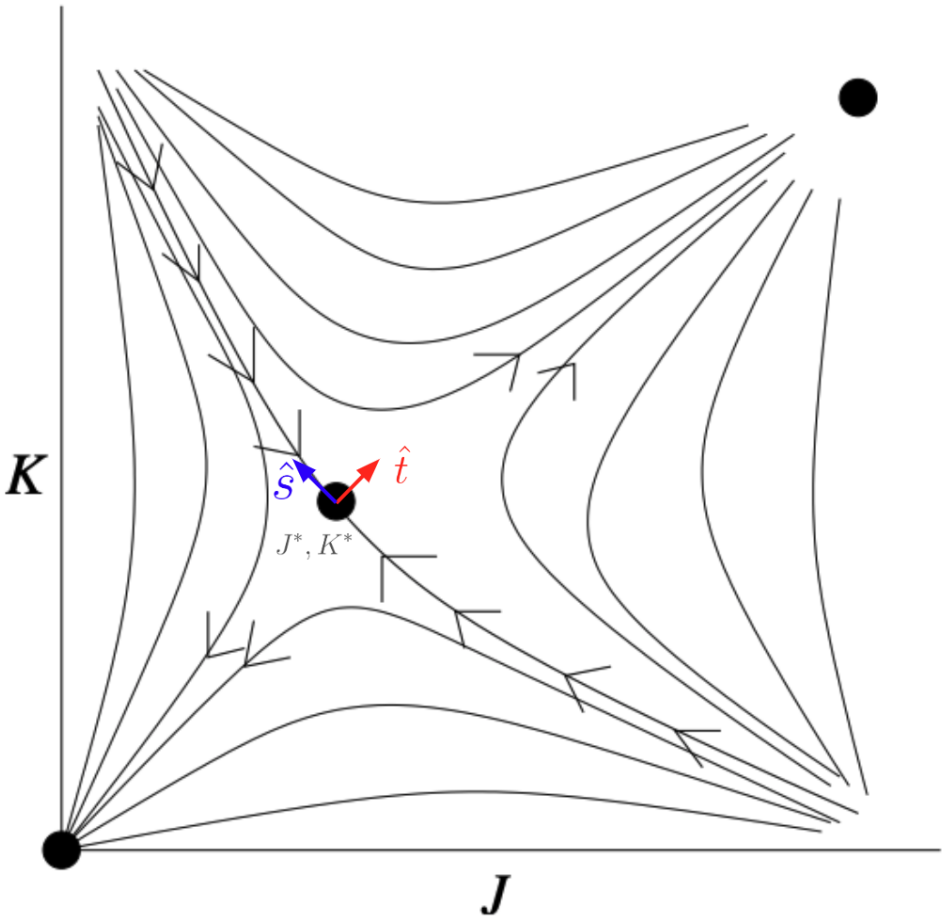

Figure 27: The scaling fields of the 2D Ising model of a magnetic system in the J, K plane at $h = 0$.

B The `absolute value` activation function at $\phi = \pi$

Let the elements of the two inputs be denoted as $x_{i;+}$ and $x_{i;-}$, where i is the element index and $+$ and $-$ are the sample indices. Let n_0 be the size of the inputs, and n_1 and n_2 be the width of the first and second layer, respectively. We denote the weights in layer l for neuron i as $w_{ij}^{(l)}$, where j is the index of either a neuron in the previous layer or an element of the input. The activation function is assumed to be the absolute value function, i.e. $\sigma(z) = |z|$. At $\phi = \pi$, we have $x_{i;+} = -x_{i;-}$. The preactivations in the first layer are given by:

$$\begin{aligned} z_{i;+}^{(1)} &= \sum_{j=1}^{n_0} w_{ij}^{(1)} x_{i;+} \\ z_{i;-}^{(1)} &= \sum_{j=1}^{n_0} w_{ij}^{(1)} x_{i;-} = \sum_{j=1}^{n_0} w_{ij}^{(1)} (-x_{i;+}) = -z_{i;+}^{(1)} \end{aligned} \tag{B.0.1}$$

This gives activations:

$$\begin{aligned} \sigma(z_{i;+}^{(1)}) &= |z_{i;+}^{(1)}| \\ \sigma(z_{i;-}^{(1)}) &= |-z_{i;+}^{(1)}| = \sigma(z_{i;+}^{(1)}) \end{aligned} \tag{B.0.2}$$

It follows that:

$$z_{i;+}^{(2)} = z_{i;-}^{(2)} = \sum_{j=1}^{n_1} w_{ij}^{(2)} \sigma(z_{i;+}^{(1)}) \tag{B.0.3}$$

Since $z_{i;+}^{(2)} = z_{i;-}^{(2)}$, we have $D^{(2)} = 0$. By an inductive argument we can show that $D^{(l)} = 0$ for $l \geq 2$.

C More results

Figures 28 and 29 show the results of the $R^{(l)}$ experiments for the scale-invariant and $K^*_{00} = 0$ universality classes, respectively.

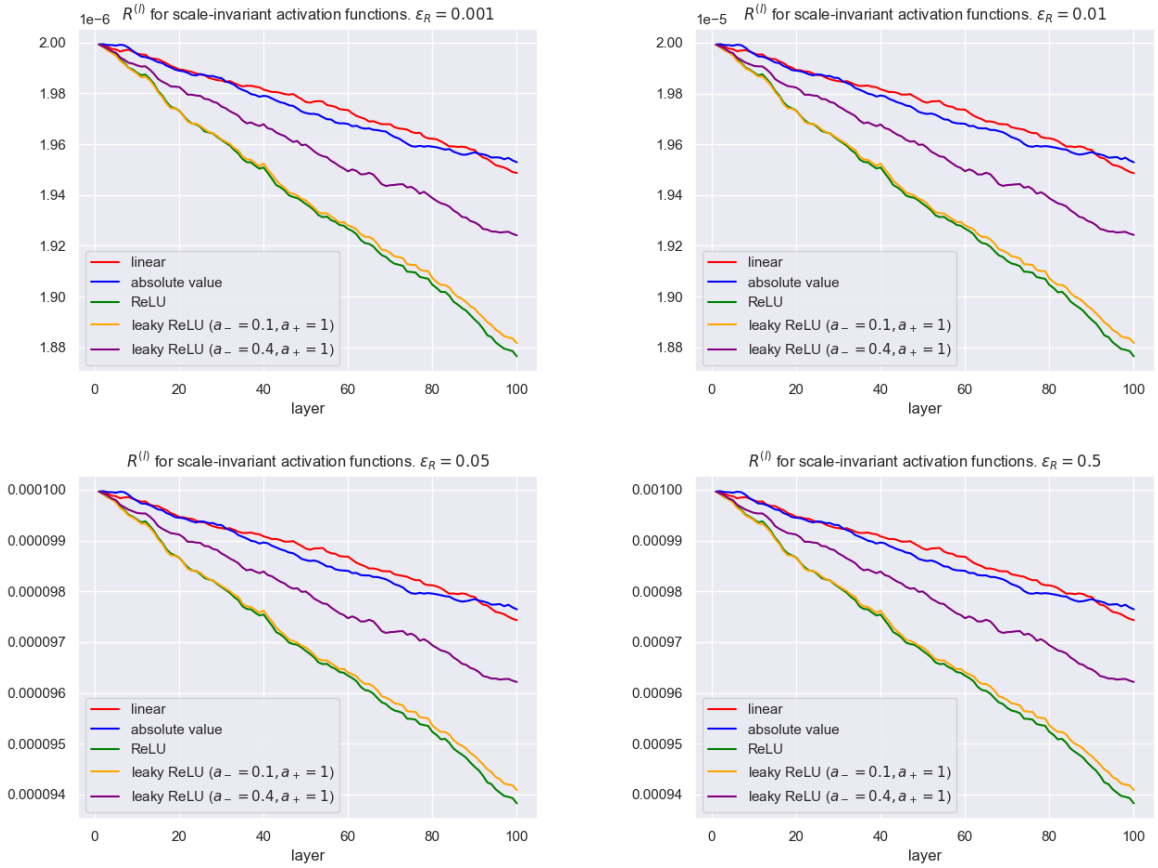

Figure 28: The $R^{(l)}$ metric for the scale-invariant universality class.

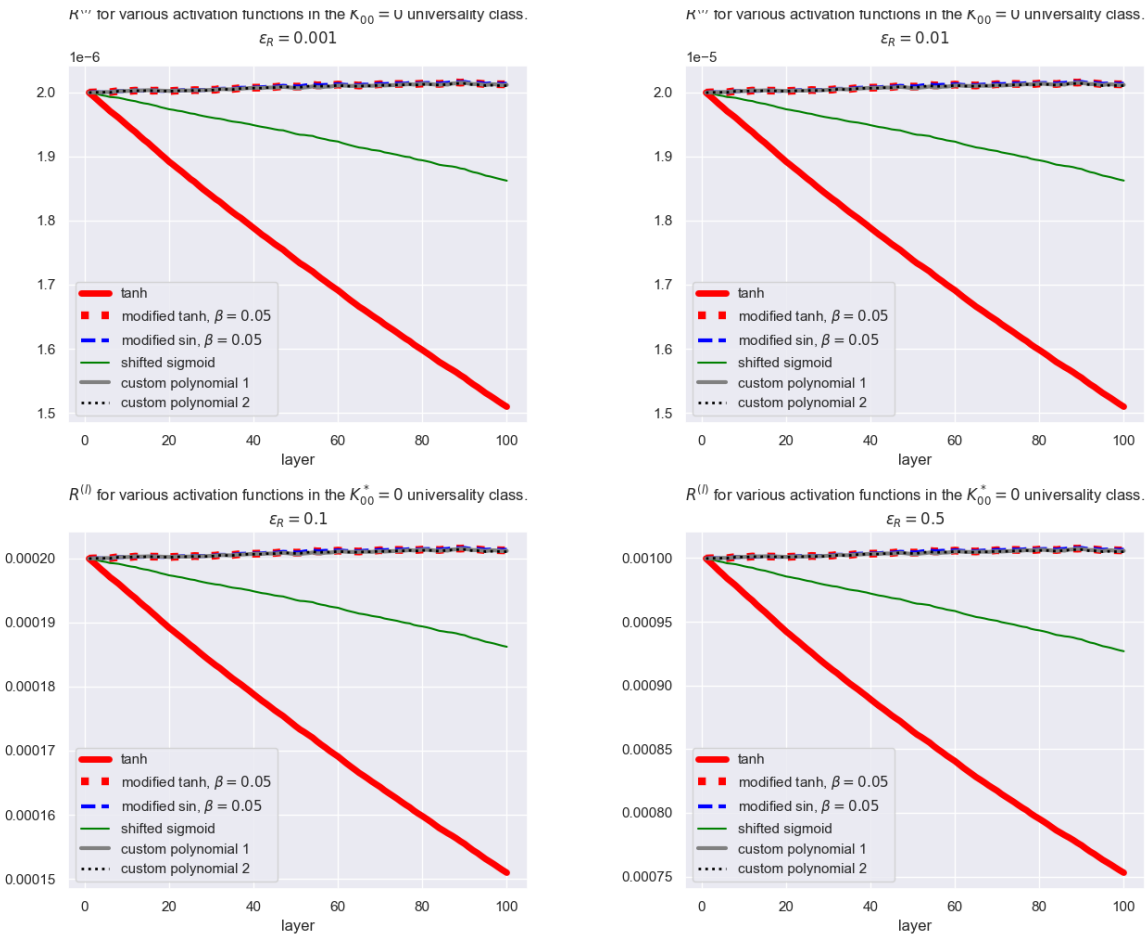

Figure 29: The $R^{(l)}$ metric for the $K^*_{00} = 0$ universality class.

D The preactivations in the first layer: a Gaussian distribution

This appendix gives a detailed account of the derivation of the distribution of the preactivations in the first layer of the MLP. We denote the density function as:

$$p\left(z^{(1)} \mid \mathcal{D}\right) \tag{D.0.1}$$

where $z^{(1)}$ is a 'random tensor' of size $n_1 \times |\mathcal{D}|$. In other words, we are looking for the joint distribution of the preactivations of all neurons in the first layer and all samples in the dataset.

Since the weights and biases are i.d.d. and zero-mean, we have:

$$\mathbb{E}\left[b_{i_1}^{(l)} b_{i_2}^{(l)}\right] = \delta_{i_1 i_2} C_b^{(1)} \tag{D.0.2}$$

$$\mathbb{E}\left[W_{i_1 j_1}^{(l)} W_{i_2 j_2}^{(l)}\right] = \delta_{i_1 i_2}\delta_{j_1 j_2} C_W^{(1)} \tag{D.0.3}$$

The equation for the preactivations in the first layer is given by:

$$z_{i;\alpha}^{(1)} = b_i^{(1)} + \sum_{j=1}^{n_0} W_{ij}^{(1)} x_{j;\alpha}, \quad \text{for} \quad i = 1, \ldots, n_1 \tag{D.0.4}$$

Recall that the bias $b_i^{(1)}$ and weights $W_{ij}^{(1)}$ are normally distributed random variables. The inputs $x_{j;\alpha}$ are real valued deterministic scalars. This makes $z_{i;\alpha}^{(1)}$ a weighted sum of normally distributed random variables. By the normal sum theorem [10], we know that $z_{i;\alpha}^{(1)}$ must also have a Gaussian distribution.

Let us refer to the joint distribution of all random variables in $z^{(1)}$ as D_z, which has probability density function $p\left(z^{(1)} \mid \mathcal{D}\right)$. With "all random variables in $z^{(1)}$" I refer to the following variables specifically:

$$z_{i;\alpha}^{(1)} \quad \text{for } i = 1, \ldots, n_1 \text{ and for all } \alpha \in \mathcal{D} \tag{D.0.5}$$

Knowing that D_z must be a multivariate Gaussian, i.e. $D_z \sim \mathcal{N}(\mu_D, \Sigma_D)$, all we need to fully specify this distribution are its mean μ_D and its covariance matrix Σ_D. However, this mean and covariance are typically a vector and matrix, respectively, while we have random variables $z_{i;\alpha}^{(1)}$ that are identified by two indices (the neural index and sample index). I found it helpful to think of $z^{(1)}$ as a random vector (instead of tensor) where all the $n_1 \cdot |\mathcal{D}|$ random variables are stacked on top of each other:

$$z^{(1)} = \begin{bmatrix} z_{i_1;\alpha_1}^{(1)} \\ \vdots \\ z_{i_1;\alpha_{|\mathcal{D}|}}^{(1)} \\ z_{i_2;\alpha_1}^{(1)} \\ \vdots \\ z_{i_2;\alpha_{|\mathcal{D}|}}^{(1)} \\ \vdots \\ \vdots \\ z_{i_{n_1};\alpha_1}^{(1)} \\ \vdots \\ z_{i_{n_1};\alpha_{|\mathcal{D}|}}^{(1)} \end{bmatrix} \tag{D.0.6}$$

Please note the specific order of the variables here, this will be important later. Starting with the mean of D_z, we see that for each element $z^{(1)}_{i;\alpha}$ of $z^{(1)}$ we have:

$$\begin{aligned}\mathbb{E}\left[z^{(1)}_{i;\alpha}\right] &= \mathbb{E}\left[b^{(1)}_i + \sum_{j=1}^{n_0} W^{(1)}_{ij} x_{j;\alpha_1}\right] \\ &= \mathbb{E}\left[b^{(1)}_i\right] + \sum_{j=1}^{n_0} \mathbb{E}\left[W^{(1)}_{ij}\right] x_{j;\alpha_1} \\ &= 0 + \sum_{j=1}^{n_0} 0 \cdot x_{j;\alpha_1} \\ &= 0\end{aligned} \tag{D.0.7}$$

Thus, $\mu_D = \vec{0}$. Next, since we have $\mathbb{E}[z^{(1)}] = \mu_D = \vec{0}$, the covariance matrix is equal to the two point correlator:

$$\Sigma_D = \mathbb{E}\left[z^{(1)} {z^{(1)}}^T\right] \tag{D.0.8}$$

Because $z^{(1)}$ is indexed by both the neural and sample indices, it is easiest to determine $\mathbb{E}\left[z^{(1)} {z^{(1)}}^T\right]$ by looking at it in an element-wise fashion:

$$\begin{aligned}\mathbb{E}\left[z^{(1)}_{i_1;\alpha_1} z^{(1)}_{i_2;\alpha_2}\right] &= \mathbb{E}\left[\left(b^{(1)}_{i_1} + \sum_{j_1}^{n_1} W^{(1)}_{i_1 j_1} x_{j_1;\alpha_1}\right)\left(b^{(1)}_{i_2} + \sum_{j_2}^{n_1} W^{(1)}_{i_2 j_2} x_{j_2;\alpha_2}\right)\right] \\ &= \mathbb{E}\left[b^{(1)}_{i_1} b^{(1)}_{i_2}\right] + \mathbb{E}\left[\left(\sum_{j_1}^{n_1} W^{(1)}_{i_1 j_1} x_{j_1;\alpha_1}\right)\left(\sum_{j_2}^{n_1} W^{(1)}_{i_2 j_2} x_{j_2;\alpha_2}\right)\right] \\ &\quad + \mathbb{E}\left[b^{(1)}_{i_1} \sum_{j_2}^{n_1} W^{(1)}_{i_2 j_2} x_{j_2;\alpha_2}\right] + \mathbb{E}\left[b^{(1)}_{i_2} \sum_{j_1}^{n_1} W^{(1)}_{i_1 j_1} x_{j_1;\alpha_1}\right] \\ &= \delta_{i_1 i_2} C^{(1)}_b + \sum_{j_1,j_2=1}^{n_1} \mathbb{E}\left[W^{(l)}_{i_1 j_1} W^{(l)}_{i_2 j_2}\right] x_{j_1;\alpha_1} x_{j_2;\alpha_2} \\ &\quad + \mathbb{E}\left[b^{(1)}_{i_1}\right] \mathbb{E}\left[\sum_{j_2}^{n_1} W^{(1)}_{i_2 j_2} x_{j_2;\alpha_2}\right] + \mathbb{E}\left[b^{(1)}_{i_2}\right] \mathbb{E}\left[\sum_{j_1}^{n_1} W^{(1)}_{i_1 j_1} x_{j_1;\alpha_1}\right] \\ &= \delta_{i_1 i_2} C^{(1)}_b + \sum_{j_1,j_2=1}^{n_1} \delta_{i_1 i_2} \delta_{j_1 j_2} \frac{C^{(1)}_W}{n_0} x_{j_1;\alpha_1} x_{j_2;\alpha_2} + 0 + 0 \\ &= \delta_{i_1 i_2}\left(C^{(1)}_b + \frac{C^{(1)}_W}{n_0} \sum_{j=1}^{n_1} x_{j;\alpha_1} x_{j;\alpha_2}\right) \\ &= \delta_{i_1 i_2} G^{(1)}_{\alpha_1 \alpha_2}\end{aligned} \tag{D.0.9}$$

where in the third equation we used the fact that any bias is statistically independent from any weight, in fourth equation we plugged in eq. (D.0.2) and eq. (D.0.3), in the fifth equation we simplified the summation by eliminating the Kronecker delta $\delta_{j_1 j_2}$, and on the final line we introduced a matrix $G^{(1)}$ of size $|\mathcal{D}| \times |\mathcal{D}|$ indexed by two sample indices α_1 and α_2:

$$G^{(1)}_{\alpha_1\alpha_2} \equiv C_b^{(1)} + C_W^{(1)} \frac{1}{n_0} \sum_{j=1}^{n_0} x_{j;\alpha_1} x_{j;\alpha_2} \tag{D.0.10}$$

If we take this element-wise formula for $\mathbb{E}[z^{(1)}_{i_1;\alpha_1} z^{(1)}_{i_2;\alpha_2}]$ and use it to populate $\Sigma_D = \mathbb{E}[z^{(1)} z^{(1)^T}]$ using the order of the indices in eq. (D.0.6), we obtain a block matrix of size $(n_1 \cdot |\mathcal{D}|) \times (n_1 \cdot |\mathcal{D}|)$ which is structured as follows:

$$\Sigma_D = \begin{bmatrix} G^{(1)} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & G^{(1)} & & \mathbf{0} \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & G^{(1)} \end{bmatrix} \tag{D.0.11}$$

where $\mathbf{0}$ is a matrix with only zero elements which has the same size as $G^{(1)}$, i.e. $|\mathcal{D}| \times |\mathcal{D}|$. Knowing the structure of Σ_D, we can write an expression for Σ_D by making use of a Kronecker product. Let I_D be an identity matrix of size $n_1 \times n_1$. Then:

$$\Sigma_D = I_D \otimes G^{(1)} \tag{D.0.12}$$

The inverse of Σ_D is found easily using the properties of the Kronecker product:

$$\Sigma_D^{-1} = \left(I_D \otimes G^{(1)}\right)^{-1} = I_D^{-1} \otimes \left(G^{(1)}\right)^{-1} = I_D \otimes \left(G^{(1)}\right)^{-1} \tag{D.0.13}$$

Note how the formula for Σ_D relates to the element-wise notation for $\mathbb{E}[z^{(1)}_{i_1;\alpha_1} z^{(1)}_{i_2;\alpha_2}]$: $\delta_{i_1 i_2}$ 'causes' I_D to appear in the formula for Σ_D. Similarly, $G^{(1)}_{\alpha_1\alpha_2}$ 'causes' the appearance of $G^{(1)}$. Using the same line of reasoning in reverse, the element-wise notation for the inverse of Σ_D is:

$$\delta_{i_1 i_2} G^{\alpha_1\alpha_2}_{(1)} \tag{D.0.14}$$

Using the properties of the Kronecker product once more, we can easily compute the determinant of Σ_D:

$$|\Sigma_D| \;=\; |I_D \otimes G^{(1)}| \;=\; |I_D|^{|\mathcal{D}|}\, |G^{(1)}|^{n_1} \;=\; 1^{|\mathcal{D}|}\, |G^{(1)}|^{n_1} \;=\; |G^{(1)}|^{n_1} \tag{D.0.15}$$

At last we can write down the probability density function for D_z. We will do so in terms of an action $S(z)$ and normalization constant Z, because this is convenient when we add correction terms to them when we look at deeper layers. Using the definition of a multivariate normal distribution, we write

$$p\left(z^{(1)} \mid \mathcal{D}\right) = \frac{1}{Z} e^{-S(z^{(1)})} \tag{D.0.16}$$

where

$$\begin{aligned} S\left(z^{(1)}\right) &= \frac{1}{2}(z^{(1)} - \mu_D)^T \Sigma_D^{-1} (z^{(1)} - \mu_D) \\ &= \frac{1}{2} z^{(1)^T} \Sigma_D^{-1} z^{(1)} \\ &= \frac{1}{2} \sum_{i_1, i_2 = 1}^{n_1} \sum_{\alpha_1, \alpha_2 \in \mathcal{D}} z^{(1)}_{i_1;\alpha_1} z^{(1)}_{i_2;\alpha_2} \delta_{i_1 i_2} G^{\alpha_1\alpha_2}_{(1)} \\ &= \frac{1}{2} \sum_{i=1}^{n_1} \sum_{\alpha_1, \alpha_2 \in \mathcal{D}} z^{(1)}_{i_1;\alpha_1} z^{(1)}_{i_2;\alpha_2} G^{\alpha_1\alpha_2}_{(1)} \end{aligned} \tag{D.0.17}$$

and

$$\begin{aligned} Z &= (2\pi)^{\frac{n_1}{2}|\mathcal{D}|}\, |\Sigma_D|^{\frac{1}{2}} \\ &= (2\pi)^{\frac{n_1}{2}|\mathcal{D}|} \left|G^{(1)}\right|^{\frac{n_1}{2}} \\ &= \left|2\pi G^{(1)}\right|^{\frac{n_1}{2}} \end{aligned} \tag{D.0.18}$$

References

[1] S. Carnot, *Reflections on the motive power of heat and on machines fitted to develop that power*. J. Wiley, 1890.

[2] L. Onsager, "Crystal statistics. i. a two-dimensional model with an order-disorder transition," *Phys. Rev.*, vol. 65, pp. 117–149, 3-4 1944. DOI: 10.1103/PhysRev.65.117. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRev.65.117.

[3] K. G. Wilson, "The renormalization group: Critical phenomena and the kondo problem," *Reviews of modern physics*, vol. 47, no. 4, p. 773, 1975.

[4] K. G. Wilson, "The renormalization group and critical phenomena," *Reviews of Modern Physics*, vol. 55, no. 3, p. 583, 1983.

[5] P. Pfeuty and G. Toulouse, *Introduction to the renormalization group and to critical phenomena*. John Wiley & Sons, 1987.

[6] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.

[7] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[8] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural networks*, vol. 4, no. 2, pp. 251–257, 1991.

[9] E. R. Gopal, "Critical opalescence," *Resonance*, vol. 5, no. 4, pp. 37–45, 2000.

[10] D. S. Lemons and P. Langevin, "5.2 normal sum theorem," in *An introduction to stochastic processes in physics*. Johns Hopkins University Press, 2002.

[11] A. R. Honerkamp-Smith, S. L. Veatch, and S. L. Keller, "An introduction to critical points for biophysicists; observations of compositional heterogeneity in lipid membranes," *Biochimica et Biophysica Acta (BBA)-Biomembranes*, vol. 1788, no. 1, pp. 53–63, 2009.

[12] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[13] 2012. [Online]. Available: https://www.youtube.com/watch?v=MxRddFrEnPc.

[14] C. R. Morris, *The dawn of innovation: The first American industrial revolution*. PublicAffairs, 2012.

[15] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International conference on machine learning*, Pmlr, 2013, pp. 1310–1318.

[16] J. W. Gibbs, *Elementary principles in statistical mechanics*. Dover Publications, 2015.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *CoRR*, vol. abs/1502.01852, 2015. arXiv: 1502.01852. [Online]. Available: http://arxiv.org/abs/1502.01852.

[18] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: A survey," *Journal of machine learning research*, vol. 18, no. 153, pp. 1–43, 2018.

[19] J. Kaplan, S. McCandlish, T. Henighan, *et al.*, "Scaling laws for neural language models," *arXiv preprint arXiv:2001.08361*, 2020.

[20] D. A. Roberts, *Why is ai hard and physics simple?* 2021. arXiv: 2104.00008 [hep-th].

[21] D. A. Roberts, S. Yaida, and B. Hanin, "The principles of deep learning theory," *CoRR*, vol. abs/2106.10165, 2021. arXiv: 2106.10165. [Online]. Available: https://arxiv.org/abs/2106.10165.

[22] J. S. Rosenfeld, "Scaling laws for deep learning," *arXiv preprint arXiv:2108.07686*, 2021.

[23] J. Thijssen, *Lecture notes statistical physics*, 2022.

[24] 2023. [Online]. Available: https://en.wikipedia.org/wiki/Critical_opalescence.

[25] G. Aarts, J. Aichelin, C. Allton, *et al.*, "Phase transitions in particle physics: Results and perspectives from lattice quantum chromo-dynamics," *Progress in Particle and Nuclear Physics*, p. 104 070, 2023.

[26] J. Shelton, *A physicist's journey to the "critical point" and the "strong force"*, 2023. [Online]. Available: https://news.yale.edu/2023/04/27/physicists-journey-critical-point-and-strong-force.

[27] M. v. Tartwijk, *Introduction to an Emergent Theory of Deep Learning*. TU Delft, 2023. [Online]. Available: https://drive.google.com/file/d/19luA1Les5qxRCvjgzHvZBLY9B6TkVh1v/view?usp=sharing.

[28] Wikipedia, the free encyclopedia, *Magnetic moments*, [Online; accessed December 18, 2023], 2023. [Online]. Available: https://upload.wikimedia.org/wikipedia/commons/5/56/Diagram_of_Paramagnetic_Magnetic_Moments.png.

[29] [Online]. Available: https://docs.scipy.org/doc/scipy/tutorial/integrate.html#general-integration-quad.

[30] [Online]. Available: https://pytorch.org/.

[31] [Online]. Available: https://jupyter.org/.

[32] [Online]. Available: https://chromewebstore.google.com/detail/open-in-colab/iogfkhleblhcpcekbiedikdehleodpjo?hl=en.

[33] [Online]. Available: https://colab.google/.