Seminar Theoretical Foundations of Deep Learning

Seminar Report: Network Representations

Department of Statistics
Ludwig-Maximilians-Universität München

Jonas Schernich

Munich, February 11, 2025

Supervised by Prof. Dr. David Rügamer & Julius Kobialka

Abstract

Neural network representations are an emerging field of research. Approaching this topic, Ziyin et al. (2024) introduced the Canonical Representation Hypothesis, as well as the Polynomial Alignment Hypothesis, which deepen the understanding of how neural networks align throughout their training. Zeger et al. (2024) provide insight into how neural networks can alternatively be represented through Lasso problems, giving key insight on how deep layers can learn complex relationships via the concept of reflection features. Together, these approaches help to understand different phenomena in the field of deep learning as well as support practical model selection decisions.

Contents

1 Different Perspectives on Network Representations

Network representations are the latent states that form as data passes through the layers of a neural network. They are the transformed versions of the input data. The lack of understanding of these representations is a reason why neural networks are called black boxes. Since gaining insight into these representations is crucial not only for advancing research in deep learning but also for improving model selection and construction in practice, investigating in the field of network representations is essential. Ziyin et al. (2024) and Zeger et al. (2024) contributed by offering different perspectives on this topic. While Ziyin et al. (2024) specifically want to 'open the black box problem and advance our scientific understanding of modern AI systems', Zeger et al. (2024) seek to provide insight in the expressive power and learning capabilities of deeper neural networks for low-dimensional data. Thus, Ziyin et al. (2024) focus on the learning dynamics of neural networks, introducing the Canonical Representation Hypothesis (CRH) which suggest that representations align with the weights and gradients after training, as well as the Polynomial Alignment Hypothesis (PAH) as an alternative method by which the network organizes itself. While they focus on understanding how network representations emerge, Zeger et al. (2024) focus on simplifying the representations a network learns from the data. Instead of focusing on the learning process within the network, they simplify the optimization problem by using piecewise linear functions.

2 Formation of Representations in Neural Networks (Ziyin et al., 2024)

This section discusses the perspective on the topic of network representations introduced by Ziyin et al. (2024), who are sometimes referred to as 'the authors' and the discussed work is referred to as 'the paper' throughout this section.

2.1 Background of the Paper

Ziyin et al. (2024) motivate the topic approached in their paper by discussing previous work which studied how network representations emerge and certain components of a neural network align. Concretely, the authors reference the neural collapse phenomenon (NC), introduced by Papyan et al. (2020), and the neural feature ansatz (NFA), proposed by Radhakrishnan et al. (2023). Papyan et al. (2020) describe the emergence of proportionalities in the penultimate fully connected layer in the terminal phase of training of deep neural nets, while Radhakrishnan et al. (2023) show how, for fully connected layers, weight matrices and gradients can align during training. How the NC and the NFA are related to the CRH is discussed in detail in Section 2.6. Furthermore, Kaplan et al. (2020) and Bahri et al. (2024) explored the relations of network components by empirical power laws relating model performance to network size. Ziyin et al. (2024) apply this very concept of power laws through the PAH as an extension to the CRH concerning the reciprocal relations of different network components.

2.2 Formal Foundation

2.2.1 Fully Connected Feedforward Neural Network

The authors consider a fully connected feedforward neural network with the structure:

$$x \to h^1 \to h^2 \to \cdots \to h^D \to \hat{y}$$

For an arbitrary layer of any model following a linear transformation, they define the *"preactivation"* h_b as $h_b = Wh_a(x)$, with h_a (*"postactivation"*) being the output after applying any

nonlinear activation function to the previous layer and $f(x) = f(h_b(x))$ being another arbitrary transformation.
They also account for the situation including a trainable bias by $h_a = (h'_a(x), 1)$. The bias will not be further discussed in this report as it is not a key part of the main ideas.

2.2.2 Gradient Definitions

The authors define gradients g_a and g_b as the negative derivatives of the sample-wise loss function ℓ with respect to h_a and h_b, respectively, i.e., $g_a = -\nabla_{h_a}\ell$ and $g_b = -\nabla_{h_b}\ell$.

2.2.3 Matrix Definitions for Proofs

To perform their proofs and illustrate the key concepts from their paper, the authors introduce some additional notation. Specifically, they define the following matrices:

For $c \in \{a, b\}$: $\quad H_c = \mathbb{E}[h_c h_c^\top], \quad G_c = \mathbb{E}[g_c g_c^\top], \quad Z_c = M_c M_c^\top, \quad$ where $M_a = W^\top = M_b^\top$.

Here, the values a and b correspond to the postactivation and preactivation notations in the network. The expectations are taken with respect to the input values, as h_c and g_c are input-dependent, while W is not. Evaluating the three matrices on a and b results in a total of six matrices.

2.2.4 Assumptions

Ziyin et al. (2024) base the CRH on minimal assumptions. Specifically, their paper uses stochastic gradient descent (SGD) to optimize the neural network and is primarily based on the assumption of mean-field norms, assuming that the norms of g and h to approximate their empirical averages, so $\|h_a\|^2 = \mathbb{E}[\|h_a\|^2]$ and $\|g_b\|^2 = \mathbb{E}[\|g_b\|^2]$ (Assumption 1).

2.3 Relevance of the Key Ideas

Research on the principles of understanding the learning dynamics of neural networks is a relatively new area that has only started to be investigated a few decades ago, for instance, with backpropagation by Rumelhart et al. (1986) and the vanishing gradient problem by Hochreiter (1991). The CRH improves the understanding of such learning dynamics by its description of the alignment of weights, gradients, and representations. Moreover, Ziyin et al. (2024) demonstrate how the CRH can be used to explain the ability of neural networks to learn representations that are invariant to task-irrelevant features and how it offers a unified framework to explain NC and NFA. The PAH becomes relevant when the CRH does not hold, a situation the authors refer to as the 'breakage of CRH'.

2.4 Canonical Representation Hypothesis

2.4.1 Definition of the Canonical Representation Hypothesis

Formally, the CRH uses the previously defined matrices H_c, G_c and Z_c. Applying them to both a and b results in the following six matrices:

$$1.\, H_a = \mathbb{E}[h_a h_a^T] \quad 2.\, G_a = \mathbb{E}[g_a g_a^T] \quad 3.\, Z_a = W^T W$$
$$4.\, H_b = \mathbb{E}[h_b h_b^T] \quad 5.\, G_b = \mathbb{E}[g_b g_b^T] \quad 6.\, Z_b = W W^T$$

To fulfill the requirements of the CRH, two alignment categories must fully hold, resulting in all six of these matrices being involved in two alignments each. Firstly, there are three backward alignments concerning the postactivation, based on H_a, G_a, and Z_a: representation-gradient alignment (RGA), where $H_a \propto G_a$; representation-weight alignment (RWA), where $H_a \propto Z_a$; and gradient-weight alignment (GWA), where $G_a \propto Z_a$.

Secondly, there are three forward alignments, which can be considered the analogs of the backward alignments for the preactivation case. They are based on H_b, G_b, and Z_b and specifically defined by $H_b \propto G_b$ (RGA), $H_b \propto Z_b$ (RWA) and $G_b \propto Z_b$ (GWA).
Only if all six alignments hold after the training, the CRH holds. If just one of the alignments is not fulfilled, the CRH breaks.

2.4.2 Theoretical Foundation

Under the assumption of mean field norms, the authors prove that noise regularization balance can lead to the proportionality of H_c and G_c, when considering a SGD training process. They define the resulting alignments of weights, gradients, and representations in Theorem 1 (A.1) of the paper. Specifically, when h_a, g_b, and W reach stationarity, meaning their expected changes $\mathbb{E}[\Delta(h_a h_a^T)] = 0$, $\mathbb{E}[\Delta(g_b g_b^T)] = 0$, $\mathbb{E}[\Delta(WW^T)] = 0$, and $\mathbb{E}[\Delta(W^T W)] = 0$, two relationships emerge. Firstly, the forward weight-based matrix WW^T combined with a scaled gradient covariance aligns with the scaled covariance of the preactivations, i.e., $WW^\top + c_1\mathbb{E}[g_b g_b^\top] = c_2\mathbb{E}[h_b h_b^\top]$. Secondly, the backward weight-based matrix $W^T W$ combined with a scaled postactivation covariance aligns with the respective gradient covariance, i.e., $W^\top W + c_3\mathbb{E}[h_a h_a^\top] = c_4\mathbb{E}[g_a g_a^\top]$. Finally, if the training process is also at a local minimum, the proportionalities
$WW^\top \propto \mathbb{E}[g_b g_b^\top] \propto \mathbb{E}[h_b h_b^\top]$ and $W^\top W \propto \mathbb{E}[h_a h_a^\top] \propto \mathbb{E}[g_a g_a^\top]$ hold, which defines the CRH. In the proof of this theorem, the authors consider the time evolution of hh^T in an SGD training step to be zero at the end of the training. They decompose such a SGD step into the following three terms: a learning term, a weight decay term, and a noise term induced by SGD. At stationarity, these terms cancel each other out due to the noise-regularization balance. With the support of Lemma 5(A.4) of their paper, this proves the CRH. The exact process of the proof, including Lemma 5(A.4), is moved to the appendix (A.1.1) due to space constraints.

2.4.3 Why Can the Canonical Representation Hypothesis Break

The authors primarily focus on two mechanisms leading to breakage of the CRH, the competition between zero training loss & stationary fluctuation (1) and breakage due to finite training time (2). Additionally, in their "Insights" section they suggest that the lack of independence between size and direction of h can also break the CRH (3).

1. Competition between zero training loss & stationary fluctuation

For the six alignments to hold both $\mathbb{E}[\Delta W]$, $\mathbb{E}[\Delta Z]$ have to be zero (stationarity), where the expectation $\mathbb{E}$ is taken over all random components of SGD (such as the mini-batch selection). In this situation, the training can have reached a local minimum. Furthermore, $\mathrm{Var}[\Delta W]$ has to be zero, with variance taken with respect to all random components of SGD. This is called zero fluctuation and is formally implied as follows: Since $\mathbb{E}[\Delta W] = \mathbb{E}[\Delta Z] = 0$, it follows that $\mathrm{Var}[\Delta W] = 0$. More explicitly, we have $\mathrm{Var}[\Delta W] = \mathbb{E}[\Delta W \Delta W^T] + \mathbb{E}[\Delta W]\mathbb{E}[\Delta W]^T$, and substituting $\mathbb{E}[\Delta W] = 0$ and $\mathbb{E}[\Delta Z] = 0$ results in $\mathrm{Var}[\Delta W] = \mathbb{E}[\Delta Z] + \mathbb{E}[\Delta W]\mathbb{E}[\Delta W]^T = 0 + 0 = 0$.
This creates a fundamental competition: For the six alignments to hold both stationarity ($\mathbb{E}[\Delta W] = \mathbb{E}[\Delta Z] = 0$) and zero fluctuation ($\mathrm{Var}[\Delta W] = 0$) are required, but the latter requirement contradicts the essential stochastic nature of SGD that is needed for optimization. Citing Xu et al. (2023), the authors argue that $\Delta W \neq 0$ even for $t \to \infty$ if the mini-batch size is not large enough. While, as indicated by Ziyin and Wu (2024), this condition can be true for some subspaces, there are also scenarios where it does not hold, leading to the breakage of the CRH.

2. Finite Time Breaking
Finite time breaking describes the issue that not all alignments occur in finite time. While some alignments occur relatively quickly, others can take impractically or even infinitely long. Formally, the authors approach this issue using Lemma 5 (A.4), where they state that $\mathbb{E}[g_b h_b^T] = \mathbb{E}[g_b h_b^T]^T = \gamma W W^T$. As shown in the proof of Theorem 1 (A.1) of the paper, this statement is used for the noise regularization balance described by the equation $z_b \mathbb{E}[g_b h_b^T] + z_b \mathbb{E}[h_b g_b^T] + \eta {z_b}^2 \mathbb{E}[g_b g_b^T] = 2\gamma \mathbb{E}[h_b h_b^T]$. By substituting $\mathbb{E}[g_b h_b^T] = \gamma W W^T$, this leads to $2\gamma z_b W W^T + \eta {z_b}^2 G_b = 2\gamma H_b$ (2.2.3), and thus $G_b = \frac{2\gamma}{\eta {z_b}^2}(H_b - z_b W W^T)$. In practice, H_b and $z_b W W^T$ might not end up being equal and instead, $H_b - z_b W W^T = O(\gamma) H_b$. It follows that $G_b + O(\gamma) \propto \gamma^2 H_b$. Since $H_b \propto W W^T$, we obtain $G_b + O(\gamma) \propto \gamma^2 W W^T$. Analogously, it can be shown that $H_a + O(\gamma) \propto \gamma^2 G_a \propto \gamma^2 W^T W$. It should be noted that this explicit derivation of the finite time breaking is not provided by the authors and only holds under the assumption that there is a mistake in the paper, specifically in the definitions for the proof of Theorem 1 (A.1). The mistake is noted in the discussion of the proof in the appendix (A.1.1).
It can be seen that the forward RGA and GWA as well as the backward RGA and RWA alignment all depend on quadratic terms in the weight decay γ. This can cause alignment to take very long for small γ. Nevertheless, the forward RWA between H_b and $W W^T$ and the backward GWA between G_a and $W^T W$ can be strong. The authors also support these result empirically, displayed in Figure 8 (A.14.1) of the paper.

3. Additional Reason
Though not mentioned in the main section on the reason why the CRH can break, in their section 'Insights', the authors mention this possible third reason for the breakage. This reason was observed through the empirical work conducted by Ziyin et al. (2024). While Figure 7 (A.14.1) of the paper demonstrates strong alignment between the matrices H and G, it reveals that the activation norm $\|h\|$ is not independent of its direction $h/\|h\|$. This is illustrated in the plot by three distinct branches in the relationship between gradient and activation covariances. According to the authors, there should be a single, well-defined relationship between gradients and activations, which would appear as a single branch in the plot, and the activation norm should remain independent of its direction. However, the figure (A.14.1) showing three distinct branches indicates that the same gradient values correspond to multiple different activation values and thus that the activation norm depends on its direction. This discrepancy between theoretical prediction and empirical observation is another limitation of the CRH theory and mechanism through which the CRH can break.

2.5 Alternative Network Organization for Incomplete Alignments

2.5.1 CRH Master Theorem

Theorem 2 (A.2) of the paper, the 'CRH Master Theorem', describes the network's organizational transition from the CRH to the PAH. This transition is characterized by varying alignment restrictions in four properties.
The most stringent condition, Property 4 (Canonical Alignment II), requires all six alignments to be satisfied simultaneously resulting in the relationship $\mathbb{E}[hh^T] \propto \mathbb{E}[gg^T] \propto Z \propto P$, where P denotes an orthogonal projection matrix. By requiring the perfect alignment between all six matrices Property 4 defines the conditions that must be met for the CRH to hold.
Property 3 (Canonical Alignment I) is less stringent and defined by the need for one forward alignment, one backward alignment, and any additional alignment. When these conditions are fulfilled, the CRH holds in a Z^0 subspace and at a local minimum.

The transition to the PAH is then described by Properties 1 and 2. Property 1 (Directional Redundancy) states the existence of redundant relations between the six matrices. The redundancy is used by Property 2 (Reciprocal Polynomial Alignments) to define power laws that allow the network's self-organization in a case where only one forward and one backward alignment hold.

2.5.2 Polynomial Alignment Hypothesis

Via the redundancy defined in Property 1, the power laws in Property 2 define the PAH. Specifically, if only one forward and one backward alignment hold, all of the forward and backward alignments hold via power laws and is formally established by the proportionalities $H^{\alpha_c} \propto G^{\beta_c} \propto Z^{\delta_c}$, for $\alpha_c, \beta_c, \delta_c$ satisfying $-1 \leq \alpha_c, \beta_c, \delta_c \leq 3$.
Via this definition the PAH is an extension to the CRH, imposing weaker requirements to the matrix proportionalities. However, it also includes the definition of the CRH, since exponents ranging between -1 and 3 include 1 as a valid case. If all $\alpha_c, \beta_c, \delta_c = 1$ for $c \in \{a, b\}$, this definition describes matrix proportionality as in the CRH, ensuring all six alignments hold. For other values of $\alpha_c, \beta_c, \delta_c$ however, the less strict alignment conditions become relevant. An example of the polynomial relation in the case of forward RGA with powers $\alpha_a = 2$ and $\beta_a = 0.5$ looks as follows:

$$H_a = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \quad \Rightarrow \quad H_a^2 = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad G_a = \begin{bmatrix} 64 & 0 \\ 0 & 324 \end{bmatrix} \quad \Rightarrow \quad G_a^{0.5} = \begin{bmatrix} 8 & 0 \\ 0 & 18 \end{bmatrix}.$$

Thus, $H_a^{\alpha_a} \propto G_a^{\beta_a}$, despite H_a note being proportional to G_a.

2.5.3 Phases

When the PAH applies in the case of CRH breakage, distinct phases emerge, which describe the alignment status of an arbitrary hidden layer, meaning the specific polynomial relation of all six matrices in that layer. Any given hidden layer exhibits a combination of alignments that either hold or do not hold. Since there are six possible alignments, each of which can either hold or not hold, a total of $2^6 = 64$ phases are possible for a layer.
Figure 8 (A.14.1) from the paper illustrates how, for a given layer, some alignments may be strong while others may be weak, as discussed in detail in the context of the competition between zero training loss and stationary fluctuations(2.4.3).
When accounting for the in Theorem 2 (A.2) introduced redundancies, the number of unique phases reduces to $6 + \binom{6}{2} + 1 = 22$ phases.
Additionally, the authors touch on the idea of adding ordering to the phases, which would result in $6! = 720$ phases.

2.5.4 Gradual Alignment

Ziyin et al. (2024) claim to expect the CRH to hold better in later layers, due to the lower rank of $\mathbb{E}[\Delta W]$, the reasons for which are discussed in the context of the first reason for CRH breakage (2.4.3) which is supported by empirical work. This is also illustrated in Figure 1, displaying the results of empirical work performed for this seminar, which shows a comparison of different layers regarding the alignment strength of the forward RGA. The authors state that if any hidden layer shows an almost perfect alignment, then all the following layers do as well. This phenomenon shows clear similarities to the NC and, according to the authors, is also consistent with the tunnel phenomenon discovered by Masarczyk et al. (2024).

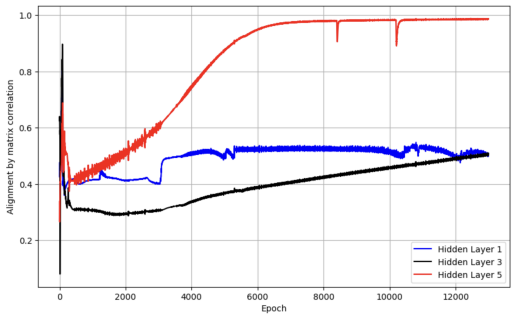

Figure 1: Comparison of layer one, three and five regarding the alignment strength of the forward RGA. Details can be found on **GitHub**.

2.6 Connection to Previous Research

Ziyin et al. (2024) mention several related works, such as Ziyin and Wu (2024) which shows the alignment of the latent representation in a deep linear network with a linear transformation of the prediction residual. However, this section will focus on the relation to the neural networks invariance task-irrelevant features, NC, and NFA.

2.6.1 Invariance to Task-Irrelevant Features

Citing Zeiler and Fergus (2014) and Selvaraju et al. (2017), the authors claim that the latent representations in well-trained CNNs would become essentially invariant to task-irrelevant features and suggest the CRH to be a reason for this invariance. In proposition 1 (A.5) of the paper, they argue that if hidden states h of a model $f(h(x))$ satisfy the RGA and $\hat{n}$ is any unit vector in the latent space, having a loss that is invariant up to second-order terms under perturbations in the direction $\hat{n}$ is equivalent to the latent covariance satisfying $\mathbb{E}[hh^\top]\hat{n} = 0$. If the CRH holds, it is additionally implied that both the weight matrix and the gradient covariance vanish along $\hat{n}$ meaning $W\hat{n} = 0$ and $G\hat{n} = 0$. In summary, the latent direction that does not affect the loss is suppressed, leading to a representation that is invariant to task-irrelevant features, which provides a theoretical explanation for the invariance observed in well-trained CNNs.

2.6.2 Neural Collapse

Papyan et al. (2020) introduce NC as a phenomenon in deep networks that arises particularly in the final stages of training and define it via four key properties:
NC1: Variability Collapse describes the vanishing of within-class variability, meaning that the last-layer activations of each class converge to their corresponding class mean.
NC2: Simplex Property states that after centering around the global mean, the ordered class means form an approximate equilateral simplex, resulting in equal mutual distances/angles.
NC3: Self-Duality refers to final classifiers aligning themselves with these class means.
NC4: Nearest-Class-Center states that the decision rule degeneration to a nearest-class-mean criterion, i.e., when considering feature representation $h \in \mathbb{R}^d$ in the last-layer feature space, a set of class weights w_c corresponding to class c, and class means μ_c, the classification decision

follows: $\arg\max_c \langle w_c, h\rangle \approx \arg\min_c \|h - \mu_c\|$.
These properties describes how neural networks simplify during training, which leads to improved generalization, robustness, and interpretability. Ziyin et al. (2024) formally explore the relationship between NC and the CRH in Theorem 3 (A.3). They state that for penultimate-layer activations h_a, a quasi-interpolating model $f(x_c)$ with $f(x_c) = h_b = W h_a(x_c) = \zeta \mathbf{1}_c$ for each class c and a loss covariance proportional to the identity, formally $\mathbb{E}\left[\nabla_f \ell \nabla_f \ell^\top\right] \propto I$, h satisfies NC1–NC4 if it also satisfies the CRH.
Concretely this process can be described by NC1 first enforcing that the penultimate-layer activations satisfy $h_a(x_c) = \mu_c$, secondly, NC2 ensuring that the covariance H_a is proportional to an orthogonal projection and thirdly, NC3 leading to the structural constraint $W^\top W \propto \sum_c \mu_c \mu_c^\top$. Together, these conditions establish the backward CRH, expressed as $H_a \propto G_a \propto Z_a$. Furthermore, the forward CRH follows with $Z_b \propto I$ and $\mathbb{E}[h_b h_b^\top] \propto I$ as well.
One can conclude that the CRH is a generalization of NC. While NC is specific to the penultimate layer in classification tasks, CRH extends to more general network architectures and layers.

2.6.3 Neural Feature Alignment

By introducing the NFA Radhakrishnan et al. (2023) demonstrate how weight matrices and gradients in fully connected layers can align during training. They introduce the Neural Feature Matrix(NFM), which defines how features are scaled and rotated and is formalized as $W_i^\top W_i$, with W_i denoting the weights of layer i in a neural network. Radhakrishnan et al. (2023) show that this NFM is proportional to the Average Gradient Outer Product. The NFA states that $W^\top W \propto \mathbb{E}[\nabla_{h_a} f \nabla_{h_a} f^\top]$, which directly links to GWA introduced by Ziyin et al. (2024). The NFM can be related to Z_c and the Average Gradient Outer Product to G_c.
Ziyin et al. (2024) discuss the equivariance properties of NFA and CRH, stating that while the NFA is not invariant under trivial model and loss redefinitions, the GWA of CRH are invariant. Formally, when considering $f'(x) := Zf(x)$, with Z invertible, and a corresponding loss transformation $l'(f) := l(Z^{-1} f)$, the NFA leads to inconsistent conditions:
for f, $W^\top W \propto \mathbb{E}[\nabla f \nabla f^\top]$, yet for f', $W^\top W \propto \mathbb{E}[\nabla f \, Z \, Z^\top \nabla f^\top]$. They cannot both hold simultaneously. In contrast, the GWA remains invariant, since $\nabla l(f) \nabla l(f)^\top = \nabla l'(f') \nabla l'(f')^\top$.

2.7 Empirical Work

The paper provides empirical work, validating and illustrating the CRH for different networks, including ResNet-18 and Transformer, as well as conditions for CRH violation. It also shows how different phases emerge in different layers, the relation of batch sizes and γ with the phase of different layers, and CRH's relation to neural collapse. Due to limited space and since the report should focus on the theoretical foundations of the papers, specifically discussed empirical results are limited to those mentioned in previous chapters.

3 A Library of Mirrors: Deep Neural Nets in Low Dimensions are Convex Lasso Models with Reflection Features (Zeger et al., 2024)

This section discusses the perspective on the topic of network representations introduced by Zeger et al. (2024), who are sometimes referred to as 'the authors' and the discussed work is referred to as 'the paper' throughout this section.

3.1 Background of the Paper

Neural networks face challenges regarding their non-convexity and complexity, including computational costs and the uncertainty in finding a global solution. In this regard, they are at a notable disadvantage when compared to some traditional optimization techniques, for instance, Lasso optimization. To introduce their approach, the authors cite Ergen and Pilanci (2021a, 2020) as well as Pilanci and Ergen (2020) , arguing that these works have shown that networks with absolute value activation functions can be perfectly described by piecewise linear functions when the network inputs are one-dimensional. Similarly, Savarese et al. (2019) and Ergen and Pilanci (2021a, 2021b) demonstrated this property for networks with ReLU activation functions. Zeger et al. (2024) sought to extend this approach in order to investigate a wider range of network architectures by reformulating a neural network optimization as a Lasso problem.

3.2 Formal Foundations

This section outlines the formal foundations used in Zeger et al. (2024), focusing on two key aspects: the requirements to the network structure and the reformulation of a neural network optimization problem into a Lasso optimization problem.

3.2.1 Requirements to the Network Structure

While the paper briefly discusses higher-dimensional input data, the main findings are for one-dimensional inputs only. For showing their findings, the authors use parallel neural networks, stating that these can be converted into standard neural networks.
Parallel neural networks are introduced to simplify the transition to a lasso formulation. These networks use several parallel units which independently process input data and whose results are eventually aggregated. Due to limited space, the mathematical details are omitted. There are four relevant types of networks. The first type is the deep narrow network which has one neuron per hidden layer. The second network type being the three-layer symmetrized network, which uses symmetrical weight parameters for different layers and neurons. The third one being rectangular networks, which consist of at least three layers of hidden neurons, all of which have the same number of neurons. The last network structure is called tree networks, which use a recursive design with a branching depth of at least three.

3.2.2 Turning a Neural Net into a Lasso Problem

The authors define the following transition from a neural network optimization problem to a lasso optimization problem:

$$\min_{\theta\in\Theta} \frac{1}{2}\|f_L(\mathbf{X};\theta) - \mathbf{y}\|_2^2 + \frac{\beta}{\tilde{L}}\|\theta_w\|^{\tilde{L}} \quad \Rightarrow \quad \min_{\mathbf{z},\xi} \frac{1}{2}\|\mathbf{A}\mathbf{z} + \xi\mathbf{1} - \mathbf{y}\|_2^2 + \tilde{\beta}\|\mathbf{z}\|_1$$

In the neural network problem $\beta > 0$ is a regularization coefficient, $\theta_w \subset \theta$ is a subset of parameters with $\|\theta_w\|^{\tilde{L}} = \sum_{q\in\theta_w} \|q\|_2^{\tilde{L}}$ penalizing the total network weight, and $\tilde{L} = L$ for ReLU, leaky ReLU, and absolute value activations, or $\tilde{L} = 2$ for threshold and sign activations. In the Lasso problem $\mathbf{z}$ is a vector, $\xi \in \mathbb{R}$, $\mathbf{1}$ is a vector of ones, and $\tilde{\beta} = \frac{\beta}{2}$ for 3-layer symmetrized networks, or $\tilde{\beta} = \beta$ for other networks. $\mathbf{A}$ is the dictionary matrix, with columns $\mathbf{A}_i \in \mathbb{R}^N$.
The authors show that the dictionary matrix transfers structural information from the neural network to the Lasso problem, allowing the optimization to proceed equivalently. This relationship is further elaborated in Section 3.7.

3.3 Relevance of the Key Ideas

The reformulation of neural networks into Lasso problems provides both theoretical insight into representative structures of neural networks as a base for further research, as well as practical implications regarding model selection and optimization. Specifically, the findings from Zeger et al. (2024) provide insight into the differentiation between layer depth, regarding the possible modeling complexity of the neural network.

3.4 Types of Features

There are several different types of features discussed by Zeger et al. (2024). Every network can learn generic features based on its corresponding activation function. Different layers can learn different features, specifically, in some cases deeper layers can learn more complex feature types, including variations of the generic activation-based feature, as well as reflection features.

3.4.1 Activation Based Features

The generic features based on the network's activation function can typically be learned by the first hidden layer of the network. What these features look like exactly depends on the activation function and is not specifically stated for all activation functions. The activation functions used in the paper can be categorized into two subsets. The first one contains continuously piecewise linear functions. This subset includes the ReLU activation, $\sigma(x) = (x)_+ := \max\{x, 0\}$, which outputs x for positive inputs and zero otherwise, the absolute value activation, $\sigma(x) = |x|$, maps negative inputs to their positive counterparts, as well as the leaky ReLU,
$\sigma(x) = (a^+\mathbf{1}\{x > 0\} + a^-\mathbf{1}\{x < 0\})x$, which allows different scaling factors a^+ and a^- for positive and negative inputs, with $a^+ \neq a^-$.
The second category consists of sign-determined functions. This category includes the threshold activation described by $\sigma(x) = \mathbf{1}\{x \geq 0\}$, which outputs 1 for non-negative inputs and 0 otherwise, as well as the sign activation, defined as $\sigma(x) = \text{sign}(x)$, where $\text{sign}(x) = -1$ if $x < 0$ and $\text{sign}(x) = 1$ if $x \geq 0$. For sign-determined activations, features are described as switching sets because such features are not linear but instead show a switching pattern between two distinct values, based on the data on which the feature is evaluated. The switching sets are defined as vectors $h \in \{-1, 1\}^N$. Absolute value features are not defined specifically. However, in Theorem 3.2 (A.6), Zeger et al. (2024) define the later discussed reflection features for absolute value networks which suggest that the standard absolute value feature is of the form $|x - a|$. The most specifically discussed feature is the ReLU feature, for which the authors differentiate between positive ReLU features and negative ReLU features, which are the authors define as $\text{ReLU}_a^+(x) = (x - a)_+$ and $\text{ReLU}_a^-(x) = (a - x)_+$, respectively. The authors state how, for such ReLU networks, complexity is added by the fact that, for example, deep narrow nets can also learn capped ramp features in their second hidden layer. These are piecewise linear functions that increase linearly between two thresholds before flattening at a maximum or minimum value.
Let's consider a deep narrow net with one hidden layer and $a, b, c \in \mathbb{R}$ with $a = 2$, $b = 3$, and $c = 8$ as three one-dimensional input data points. In this case the layer learns ReLU features with break points, so called kinks, at the data points.
Figure 2 illustrates the case of positive ReLU features. Analogously, the layer can learn negative ReLU features with breakpoints at the data points.

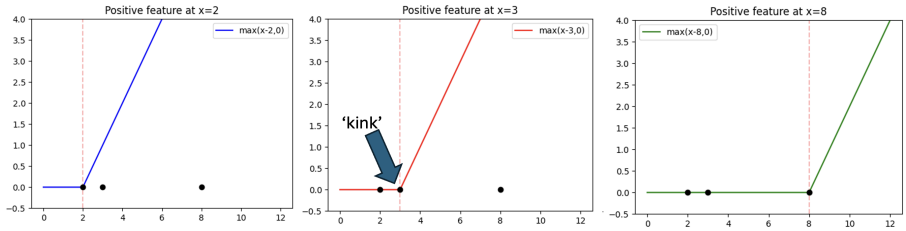

Figure 2: Illustration of positive ReLU features for data points 2, 3 and 8. The breakpoints are called 'kinks'.

3.4.2 Reflection Features

Another feature type that can be learned by some networks, generally only by networks that have at least two hidden layers, is called reflections features. Such reflection features, or just 'reflections', are one of the key findings of the paper and the reason why the paper title uses the metaphor of a mirror. Reflection features create new, artificial data points. These new artificial data points lead to more breakpoints and can thus allow for a more accurate modeling of the underlying relationships within the data, just as having more input data points would. Zeger et al. (2024) introduce three types of reflections features. Which reflection features can be learned by a network depends on the network's activation functions as well as its architectural aspects. An overview of the information that the paper provides regarding this question is available in the appendix (A.12).

Basic Reflections

Basic reflections are just referred to as 'reflections' by Zeger et al. (2024) and define the fundamental idea of reflection features by introducing the general mathematical formula for all known reflection features. All reflection features are defined as $R_{(a,b)} = 2b - a$, for $a, b \in \mathbb{R}$.
For these basic reflections a and b both have to be non-artificial data points from the input data. So if we consider our previously defined data points a, b, c, then the reflection of b about a is $R_{(3,2)} = 2 \cdot 2 - 3 = 1$.
This very simple formula now introduced a new, artificial data point which can be considered on a scale, together with the original data points. In Figure 3 the green point now represents the reflection of 3 about 2, while the blue points represent original data points.

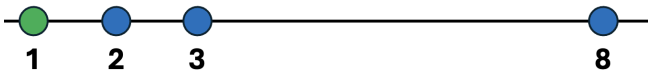

Figure 3: Illustration of original data points and an artificial data point created by the reflection of 2 about 3.

This can be done analogously for all other input data points which would results in $N \cdot (N-1)$ reflections, in this case $3 \cdot (3-1) = 6$ reflections.

Double Reflections

Double reflections extend the concept of basic reflection by relaxing the restriction on the values a and b. In the case of double reflections, a and b don't have to both be non-artificial input data points; instead, one of the points is created by a basic reflection. Thus, double reflections are defined as $R_{(R_{(a,c)},b)}$ or $R_{(b,R_{(a,c)})}$. Applying the previously defined values a, b, c again, the double reflection of the reflection of b about a, about c, would be
$R_{(R_{(b,a)},c)} = 2 \cdot c - (2 \cdot a - b) = 2 \cdot 8 - (2 \cdot 2 - 3) = 15$.
Such double reflections can also be calculated for all other combinations of reflections and input data points and can be displayed on the one-dimensional scale as well, as they are also just an artificial data point.
Illustratively, double reflections can be seen as a nesting process of reflections, an idea which is further researched in the later paper Zeger and Pilanci (2024), where the nesting of reflections is extended in a recursive manner, which the authors call '*order-k reflections*'.

Generalized Reflections
Generalized reflections are the third type of reflection introduced in the paper and are another extension to the basic reflections. Here, analogously to the double reflection, one of the input points for the reflections formula is replaced. However, instead of replacing it by a basic reflection, for generalized reflections, it is replaced by the average of two real input data points. Formally, the authors define a generalized reflection of a and c about b as $a + c - b = R_{(b,\frac{a+c}{2})}$. The simplification on the left stems from simply rewriting the definition of the reflection as $R_{(b,\frac{a+c}{2})} = 2 \cdot \left(\frac{a+c}{2}\right) - b = a + c - b$.
If we once again consider the example data points a, b, c, such a reflection of a and c about b would be $R_{(3,\frac{2+8}{2})} = 2 \cdot 5 - 3 = 7$, which is again just another artificial data point which could be displayed on the one-dimensional scale and can again be calculated for all combinations of input data points.

3.5 Dictionary

The dictionary is the fundamental concept of this paper. Other concepts like the dictionary matrix and library build upon it. The dictionary is a set of features, which can be applied to input data as feature functions. Each layer of a neural network has its own dictionary, which contains the features that layer can learn. For absolute value, leaky ReLU, and ReLU activation functions, the features contained in the dictionaries for the layers of such networks are continuously piecewise linear functions. In constrast, for threshold- and sign-activated nets Zeger et al. (2024) define the dictionaries as switching set in Lemma 3.16 (A.7). Which features a layer can learn depends on various factors, such as the depth of the layer, as well as factors regarding the network architecture, like the activation function used and whether it is a symmetrized net, a deep narrow net, etc. The authors don't practically define what dictionaries looks like, but rather describe them theoretically. For illustrative purposes and based on the authors theoretical descriptions, I opt for defining them in this report as follows. Let the dictionary of layer i be defined as $\mathcal{D}_i = \{f_a \mid f \in F_i,\, a \in \{x_1, \ldots, x_N\}\}$, with x_i being input data point i, N being the number of input data points and f being a feature from the feature space F_i, describing all the features that layer i can learn. In general, the exact dictionary for a specific layer of a specific network can't be inferred in a simple way; instead, the features contained in the dictionary have to be investigated distinctly for different dictionaries. For the case of a deep narrow net with one hidden layer and ReLU activation, the authors state that the dictionary of that hidden layer contains positive and negative ReLU features. Thus, $\mathcal{D}_1$, which defines that dictionary, contains the features $F_1 = \{\mathrm{ReLU}^+, \mathrm{ReLU}^-\}$.
So for $N = 2$, the dictionary is given by $\mathcal{D}_1 = \{\mathrm{ReLU}^+_{x_1}, \mathrm{ReLU}^+_{x_2}, \mathrm{ReLU}^-_{x_1}, \mathrm{ReLU}^-_{x_2}\}$.

3.6 Library

The library is an extension of the idea of the dictionary. The key difference is that a network can have many dictionaries, but always just one library, as the library is the union of all the dictionaries. Since there is one dictionary per layer containing the features which the network can learn through that specific layer, the union of all dictionaries covers the features from all layers and thus the library contains all the features which the network can learn.
As for the dictionary, the authors also don't specifically define what a library looks like, so to illustrate it, the following definition is introduced for this report:
$\mathcal{L} = \bigcup_{i=1}^{h} \mathcal{D}_i = \bigcup_{i=1}^{h} \{f_a \mid f \in F_i,\, a \in \{x_1, \ldots, x_N\}\}$, where h is the number of hidden layers.

3.6.1 What Does a Library Look Like

Since a single-layer network only has one dictionary and the union of a set with itself yields the same set, the library of the previously specified single hidden layer deep narrow net with ReLU activation is given by $\mathcal{L} = \mathcal{D}_1 = \{\mathrm{ReLU}_a^+, \mathrm{ReLU}_a^-\}$.
This equivalence between library and dictionary is specific to this simple case and does not generally hold for deeper architectures where the library can encompass multiple dictionaries across different layers. Specifically, for such a ReLU network, adding an additional feature adds complexity to the library by adding the capped ramp functions. Such capped ramps are contained in the dictionary of the second layer, and thus the library of a deep narrow net with two hidden layers and ReLU activation looks like this:
$\mathcal{L} = \mathcal{D}_1 \cup \mathcal{D}_2 = \{f_a \mid f \in F_1,\, a \in \{x_1, \ldots, x_N\}\} \cup \{f_a \mid f \in F_2,\, a \in \{x_1, \ldots, x_N\}\}$
$= \{\mathrm{ReLU}_a^+, \mathrm{ReLU}_a^-, \mathrm{Ramp}_{a_1,a_2}^+, \mathrm{Ramp}_{a_1,a_2}^- \mid a, a_1, a_2 \in \{x_1, \ldots, x_N\}\}$.
The exact structure of the libraries depends on the exact network architecture. For some networks, such as ReLU, leaky ReLU, and absolute value-activated networks, the library can include the previously discussed reflection features, while for others, such as sign and threshold-activated networks, the library won't include reflection features. A comprehensive overview of all mentioned libraries can be found in the appendix (A.11).

3.6.2 Growth of the Library

Figure 4 illustrates how, in many cases, the library grows with increasing depth of the network. For specific types of networks, the authors define general formulas for the growth of the library, depending on the depth of the network. They mention the following scenarios: For a deep narrow net with ReLU activation, the library grows as $O(N^2)$. With leaky ReLU or absolute value activation, these networks obtain a faster growth rate of $O(N^{L-1}2^L L!)$. When using sign or threshold activation functions, the growth behavior is different. Rectangular networks have a growth rate of $H^{m_{(L-2)}}$, where H is the size of the switching set, while tree networks grow by $H^{(\prod_{l=1}^{L-1} m_l)}$.

3.6.3 Freezing of the Library

For some networks architectures the libraries do not grow any further when additional hidden layers are added. According to the authors, this freezing happens in specific situations. For ReLU, libraries freeze if the number of neurons is the same in the middle and final layers, whereas they continue to grow if the middle layer has twice as many neurons as the final layer. In contrast, no freezing occurs at all for Leaky ReLU and absolute value. For sign activations, libraries freeze if the number of neurons remains constant across all layers, but no freezing takes place if there is a tree structure in the architecture. Finally, for threshold activations, the authors do not provide a clear statement.
Regarding the freezing of the libraries, Zeger et al. (2024) state in Theorem 3.4 (A.10) of their paper, that it would imply the stagnation of the representation capabilities of a network, even when more layers are added.

3.6.4 Deep Library

The deep library, as introduced in Definition 3.11 by Zeger et al. (2024), is a sub-concept of the normal library which consists of all network outputs $X^L(X)$ and is characterized by how features are influenced by data or midpoint feature biases, with $W^{(1)}$ elements being constrained to ± 1. This applies to three-layer symmetrized and L-layer deep narrow networks. The biases come from either training data points or normalized midpoints between data points, which allows the library capture multi-level symmetries.

3.7 Dictionary Matrix

The concept of the dictionary matrix is based on the dictionary/library. While, in theory, one could construct a dictionary matrix for every dictionary of a neural net, there is really only one relevant dictionary matrix regarding the idea of the paper of turning a neural net into a Lasso problem. This dictionary matrix uses all the features of the neural net and thus it might be more intuitive to think of it as a library matrix. The dictionary matrix uses the abstract features from the dictionaries/library of a neural net and applies them to the values of the input data points. Specifically, each column of the matrix represents one feature and each row evaluates that feature for a given input data point.
When again considering the ReLU-activated deep narrow network with a single hidden layer, previously used for illustrations, the dictionary matrix is the concatenation of the positive and the negative feature matrices for the ReLU features, formally: $A = [A^+; A^-] \in \mathbb{R}^{N \times 2N}$, where $A^+, A^- \in \mathbb{R}^{N \times N}$, with $[A^+]_{i,n} = (x_i - x_n)_+$ and $[A^-]_{i,n} = (x_n - x_i)_+$.
Given the three previously used input data points a, b, c, so $N = 3$, one ends up with the following positive and negative feature matrices and dictionary matrix:

$$A^+ = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 6 & 5 & 0 \end{bmatrix}, \quad A^- = \begin{bmatrix} 0 & 1 & 6 \\ 0 & 0 & 5 \\ 0 & 0 & 0 \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 6 \\ 1 & 0 & 0 & 0 & 0 & 5 \\ 6 & 5 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The dictionary matrix is the interface to transition from the neural net to the Lasso problem, as it contains all the information from the neural net, which is used in the Lasso problem (except for the varying definitions of β and $\tilde{L}$). When using the dictionary matrix from the previous example to optimize the Lasso problem, the goal would be to find a sparse vector $\mathbf{z} = (z_1, \ldots, z_6)$ which solves the Lasso problem optimally. The sparsity of that vector $\mathbf{z}$ defines which features from the library, which are all represented by a column, are relevant for the current optimization task.

3.8 Empirical Work

Zeger et al. (2024) provide empirical evidence to support their theoretical framework. The most relevant findings will be briefly discussed here. In their experiments, they show that solutions from standard training with Adam are as predicted by their Lasso formulation. Notable is also their observation of reflection breakpoints in deep networks with ReLU and absolute-value activations. They find that these breakpoints emerge as predicted by their theoretical findings. For networks with sign activation, it is illustrated that the dictionary matrices behave as expected and they are bounded by training set size. Furthermore, Zeger et al. (2024) use autoregressive models to demonstrate that their convex Lasso-based approach performs better than non-convex optimization when predicting Bitcoin prices. In addition to the empirical work in the paper, I ran an experiment regarding the effect of different library growth rates on the time required to solve the lasso problem, displayed in Figure 4. This illustration also indicates a possible problem regarding the dictionary matrix. Due to the polynomial growth rate of the number of features, that is for example obtained by absolute value networks, large data sets are likely to cause computational issues when solving the Lasso problem.

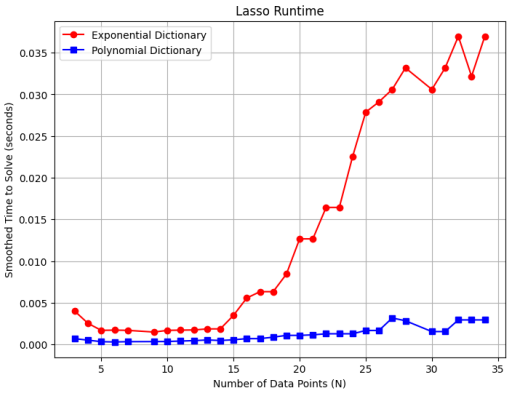

Figure 4: Time needed to solve Lasso problems for exponential ($O(N^2)$) vs. polynomial ($O(N^{L-1} \cdot 2^L \cdot L!)$) library growth. Details can be found on **GitHub**.

3.9 Higher Dimensions

The authors only briefly discuss the scenario of higher-dimensional inputs in this paper, specifically for sign-activated neural networks trained on 2D data points in the upper half-plane, with a 2-layer network and a rectangular network without internal bias parameters. In Theorem C.3 (A.8) they state the main result to be that training such networks is equivalent to solving a Lasso problem with a dictionary of switching sets $H(K)$, where $K = m_L - 2$ for the rectangular networks and $m_L \geq m^*$ (with m^* denoting the number of non-zero elements in the Lasso solution). Additionally, in Lemma C.4 (A.9), the authors provide the optimal parametrization for the 2-layer case, which uses a counterclockwise rotation matrix $R_{\pi/2}$. Even though this is a starting point showing that the idea can be extended to higher dimensions, a lot more research on the topic of higher dimensional inputs remains to be done. Thus, a follow-up paper by Zeger and Pilanci (2024), already mentioned in the Section 3.4.2 on reflection features, took up this topic once again. The paper discusses higher-dimensional inputs for absolute value-activated networks and the paper shows, for example, that reflections also exist in higher dimensions, which the authors illustrate on two-dimensional planes. These planes become more crinkled when reflection features are learned by the model.

4 Complementarity Regarding Network Representations

Both papers approach the topic of network representations from different perspectives. While Ziyin et al. (2024) focus on understanding how representations evolve during the network training by considering the alignment of different network components, Zeger et al. (2024) analyze what the final network representations look like and focus on simplifying them to better understand their implications and potentially identify a more practical model as an alternative to neural networks. These two different viewpoints give a more complete picture of the topic of network representations. Both papers provide insight into how neural networks reduce dimension. For Ziyin et al. (2024) this relates to the elaborated topic of invariance to task-irrelevant features through the alignments of the network components, while in Zeger et al. (2024) this is described by the sparsity of $\mathbf{z}$ in the Lasso problem. A possible hypothesis might be that this sparsity, in combination with more complex features learned in deeper layers, could enable more nuanced filtering of irrelevant information and thus also explain the invariance to task-irrelevant features from the perspective of Zeger et al. (2024). Generally, the relevance of network depth is a clear connection between the two papers. A possible second emerging hypothesis connecting the two approaches might be that the more complex features in deeper layers described by Zeger et al. (2024) might also be the reason for the stronger alignment in deeper layers, described by Ziyin et al. (2024). Though not proven, these questions might be an interesting avenue for further research.

5 Cross-Connections with Symmetries

5.1 Symmetries

When considering other presentations of the seminar, cross-connections can be made to Zeger et al. (2024) and Ziyin et al. (2024). Specifically the topic of 'Symmetries' seems relevant. Both Ziyin et al. (2024) and Zeger et al. (2024) are related to the ideas of symmetries discussed by Zhang et al. (2022). Zhang et al. (2022) show how symmetries lead to to non-convex optimization problems and steer the loss landscapes of these problems. This theoretical framework uses a similar perspective to the non-convex optimization perspective in Zeger et al. (2024), although the connection is not very direct. Furthermore, Zhang et al. (2022) also describes

how the symmetries they introduce can explain the practical occurrence of NC, an important phenomenon regarding network representations (see Section 2.6.2).
More direct is the connection to the symmetry concepts discussed by Ziyin (2024), a paper on symmetries referenced by Ziyin et al. (2024). In particular, Ziyin (2024) defines mirror symmetries of the form $w \mapsto (I - 2nn^\top)w$, with w being the weight vectors of the neural net, n being a unit vector in the direciton of the mirroring and I the identity matrix. This complements the representation alignment principles in the CRH introduced by Ziyin et al. (2024). Both papers show that symmetries affect the model's behavior, either through explicit parameter constraints, as in Ziyin (2024), or through alignment of representations, as in Ziyin et al. (2024). Specifically, Ziyin et al. (2024) state that the findings of Ziyin (2024) regarding permutation symmetries in the latent layers, which lead to neuron merging and low rankness in the representation, are related to their own work.

5.2 Other Cross-Connections

Though not as clearly connected to or cited in Ziyin et al. (2024) and Zeger et al. (2024), connecting the topics 'Generalization' as well as 'SGD Learning Dynamics' to the topic of this report seems reasonable. Regarding generalization, Bartlett et al. (2017) show that spectral norms of the weight matrices are a central measure for generalization bounds. The alignments which Ziyin et al. (2024) describe via the CRH and PAH might be a possible explanation of how the weight matrices evolve in a way that allows spectral norms to be a key measure for generalization bounds. Furthermore, the approach by Bartlett et al. (2017) on the topic of generalizations is complementary to the work by Zeger et al. (2024). While Bartlett et al. (2017) describe how training can lead to strong generalization via the margins and spectral norms, Zeger et al. (2024) offer a perspective on this topic through the L1-regularization used in the Lasso problem.
Another possible complementary perspective to Ziyin et al. (2024) can be found in the work of Bach and Chizat (2021) on SGD learning dynamics. Bach and Chizat (2021) describe how qualitative convergence guarantees can be derived through the analysis of gradient flow dynamics in the mean-field limit of infinitely wide neural networks. This complements the perspective on convergence that Ziyin et al. (2024) provide via the CRH and PAH.

6 Summary

Research on the topic of network representations is still in its early stages. Both Ziyin et al. (2024) and Zeger et al. (2024) provide a great service to the field of deep learning research by offering different perspectives on this topic. Not only do both papers provide complementary approaches to the topic, but they also further the understanding of neural networks by connecting different deep learning concepts, such as symmetries, and deepening the understanding of phenomena like NC. While they differ in how they approach this topic, Ziyin et al. (2024) focusing on learning dynamics and Zeger et al. (2024) on simplification, there might be room for future research connecting findings from both papers, such as by investigating the hypothesis regarding invariance to task-irrelevant features or strong alignment in deep layers.

A Appendix

A.1 Theorem 1 (Ziyin et al., 2024)

Assuming the assumption of mean-field norms holds, for any given hidden layer (after transformation), we have: $h_b = Wh_a$, and the following stationarity conditions are satisfied:

$$\mathbb{E}\big[\Delta(h_a h_a^\top)\big] = 0, \quad \mathbb{E}\big[\Delta(g_b g_b^\top)\big] = 0, \quad \mathbb{E}\big[\Delta(WW^\top)\big] = 0, \quad \mathbb{E}\big[\Delta(W^\top W)\big] = 0.$$

Then, there exist real positive constants $c_1, c_2, c_3, c_4 > 0$ such that:

$$WW^\top + c_1\mathbb{E}[g_b g_b^\top] = c_2\mathbb{E}[h_b h_b^\top], \quad W^\top W + c_3\mathbb{E}[h_a h_a^\top] = c_4\mathbb{E}[g_a g_a^\top].$$

Furthermore, at a local minimum:

$$WW^\top \propto \mathbb{E}\big[g_b g_b^\top\big] \propto \mathbb{E}\big[h_b h_b^\top\big] and W^\top W \propto \mathbb{E}\big[h_a h_a^\top\big] \propto \mathbb{E}\big[g_a g_a^\top\big].$$

A.1.1 Detailed Proof of Theorem 1 (Ziyin et al., 2024)

This is a description of the proof regarding the forward alignments. It can be done analogously for the backward alignments

Step 1: Notation and Setup

Assumption 1 ensures that the norms of h_a and g_b approximately match their empirical means, simplifying the proof.

Mistake in the paper:

The authors give the following definitions in Section B.3 of their paper regarding this proof: 'Let $B_a = h_a h_a^\top$, and $B_b = h_b h_b^\top$. Let $H_a = \mathbb{E}[B_b]$ and $H_b = \mathbb{E}[B_a]$.' These definitions result in $H_a = \mathbb{E}[h_b h_b^T]$ and $H_b = \mathbb{E}[h_a h_a^T]$. Those definitions are not consistent with definitions previously stated in the paper and also do not allow for an accurate solution to the proof. Instead, the correct definitions should be $B_a = h_a h_a^\top$, $B_b = h_b h_b^\top$, $H_a = \mathbb{E}[B_a]$ and $H_b = \mathbb{E}[B_b]$.

Step 1: Considering the time evolution of h_b

$$\Delta(h_b(x)h_b^\top(x)) = \Delta(Wh_a h_a^\top W^\top)$$

Step 2: Apply matrix multiplication $\Delta(AB) = (\Delta A)B + A(\Delta B)$

$$= \Delta W B_a W^\top + W B_a \Delta W^\top + \Delta W B_a \Delta W^\top + O(\Delta B_a)$$

Step 3: Apply SGD update step

$$\begin{aligned} &= -\eta(\nabla_{h_b}\ell h_a^\top + \gamma W)B_a W^\top - \eta W B_a(h_a \nabla_{h_b}^\top \ell + \gamma W^\top) \\ &\quad + \eta^2(\nabla_{h_b}\ell h_a^\top + \gamma W)B_a(h_a \nabla_{h_b}^\top \ell + \gamma W^\top) \end{aligned}$$

Step 4: Simplify by setting $g_b = \nabla_{h_b}\ell$ and $h_a h_a^\top = \|h_a\|^2$

$$\begin{aligned} &= -\eta\left(-\|h_a\|^2 g_b h_b^\top + \gamma B_b\right) - \eta\left(-\|h_a\|^2 h_b g_b^\top + \gamma B_b\right) \\ &\quad + \eta^2\|h_a\|^4 g_b g_b^\top + O(\eta^2\gamma) \end{aligned}$$

Step 5: Simplify by grouping terms

$$= \eta\left(\|h_a\|^2 g_b h_b^\top + \|h_a\|^2 h_b g_b^\top - 2\gamma B_b\right) + \eta^2\|h_a\|^4 g_b g_b^\top.$$

Step 6: Setting $z_b = \mathbb{E}||h_a||^2$ and taking expectation of both sides

$$0 = z_b\mathbb{E}[g_b h_b^\top] + z_b\mathbb{E}[h_b g_b^\top] + \eta z^2\mathbb{E}[g_b g_b^\top] - 2\gamma H_b$$

A.2 Theorem 2 (Ziyin et al., 2024)

CRH Master Theorem

Let A, B, C be a permutation of $\mathbb{E}[hh^\top]$, $\mathbb{E}[gg^\top]$, and Z, and let $\tilde{D} := PDP$ be a projected version of D for a projection matrix P. Then,

1. **(Directional Redundancy)** If any two forward (backward) alignments hold, all forward (backward) alignments hold.
2. **(Reciprocal Polynomial Alignments)** If one of any forward alignments and one of any backward alignments hold, there exist scalars $\alpha_c, \beta_c, \delta_c$ satisfying $-1 \leq \alpha_c, \beta_c, \delta_c \leq 3$ such that:

$$\tilde{A}_c^{\alpha_c} \propto \tilde{B}_c^{\beta_c} \propto \tilde{C}_c^{\delta_c},$$

where $c \in \{a, b\}$ denotes the backward and forward relations respectively, and the corresponding projection $P_c \in \{Z_c^0, \mathbb{E}[h_c h_c^\top]^0, \mathbb{E}[g_c g_c^\top]^0\}$, e.g., such that $\tilde{A} = P_c A P_c$.
3. **(Canonical Alignment I)** If (any) one more relation holds in addition to part 2, then all six alignments hold in the Z^0 subspace; in addition, at a local minimum, all six alignments hold.
4. **(Canonical Alignment II)** If all six alignments hold, $\mathbb{E}[hh^\top] \propto \mathbb{E}[gg^\top] \propto Z \propto P$, where P is an orthogonal projection matrix.

A.3 Theorem 3 (Ziyin et al., 2024)

Consider a classification task and the penultimate layer postactivation h_a. If the model is quasi-interpolating: $f(x_c) = h_b = W h_a(x_c) = \zeta \mathbf{1}_c,$) and the loss covariance is proportional to identity: $\mathbb{E}[\nabla_f \ell \nabla_f^\top \ell] \propto I$, then h satisfies all four properties of neural collapse (NC1–NC4) if and only if h satisfies the CRH.

A.4 Lemma 5 ((Ziyin et al., 2024))

First-order stationary point condition / local At any stationary point of the loss function, we have

$$E[g_b h_b^T] = E[g_b h_b^T]^T = \gamma W W^T,$$
$$E[g_a h_a^T] = E[g_a h_a^T]^T = \gamma W^T W.$$

A.5 Proposition 1 (Ziyin et al., 2024)

Let $f(h(x))$ be a model whose hidden states h obey RGA. Let $\hat{n}$ be a vector and ϵ a scalar. *The following statements are equivalent:* (1) $\ell(f(h + \epsilon\hat{n})) = \ell(f(h)) + O(\epsilon^2)$; (2) $\mathbb{E}[hh^\top]\hat{n} = 0$; (*if the CRH also holds*) $W\hat{n} = 0$, and $G\hat{n} = 0$.

A.6 Theorem 3.2 (Zeger et al., 2024)

Lasso equivalent of deep absolute value networks
A deep narrow network of arbitrary depth with $\sigma(x) = |x|$ is equivalent to a Lasso model with a finite set of features. Its dictionary matrix for 2 layers is $\mathbf{A}_{i,j} = |x_i - x_j|$. For 3 and 4 layers, its library includes features whose i^{th} element is $\big||x_i - x_{j_1}| - |x_{j_2} - x_{j_1}|\big|$ and $\Big|\big||x_i - x_{j_1}| - |x_{j_2} - x_{j_1}|\big| - |x_{j_3} - x_{j_1}| - |x_{j_2} - x_{j_1}|\Big|$, respectively, over all training samples $x_i, x_{j_1}, x_{j_2}, x_{j_3}$. A similar pattern holds for deeper networks. The 2-layer features have breakpoints exactly at training data. The libraries for 3 and 4 layers additionally include reflection and double reflection features, respectively.

A.7 Lemma 3.16 (Zeger et al., 2024)

The dictionary matrix for a 2-layer network with *sign activation* is $\mathbf{H}^{(1)}$.

A.8 Theorem C.3 (Zeger et al., 2024)

Consider a Lasso problem whose dictionary is the switching set $\mathbf{H}^{(K)}$, $\xi = 0$, and with solution $\mathbf{z}^*$. Let $m^* = \|\mathbf{z}^*\|_0$.
This Lasso problem is *equivalent* to the training problem for a *sign-activated network without internal biases* that is *2-layer or rectangular*, satisfies $m_L \geq m^*$, $m_{L-2} = K$, and is trained on *2-D data with unique angles in* $(0, \pi)$.

A.9 Lemma C.4 (Zeger et al., 2024)

Let $\mathbf{R}_{\frac{\pi}{2}} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$ be the counterclockwise rotation matrix by $\frac{\pi}{2}$. An optimal parameter set for the training problem in *Theorem C.3* when $L = 2$ is the unscaled version of

$$\theta = \left\{ \alpha_i = z_i^*, \mathbf{s}^{(i,1)} = 1, \mathbf{W}^{(i,1)} = \mathbf{R}_{\frac{\pi}{2}} \left(\mathbf{x}^{(i)}\right)^{\mathbf{T}}, \xi = 0 : z_i^* \neq 0 \right\},$$

where $\mathbf{z}^*$ is optimal in the Lasso problem.

A.10 Theorem 3.4 (Zeger et al., 2024)

A deep narrow network of arbitrary depth with *ReLU activation* is *equivalent* to a Lasso model with a finite set of features. Its library contains only *ramps and capped ramps*, and beyond 3 layers, *ReLU features do not change as the network deepens.*
In contrast to absolute value activation, for deep narrow networks, the ReLU library never gains reflection features. However, a symmetrized ReLU architecture creates reflections.

A.11 Library Overview

Activation	DNN with 1 L	DNN with 2 L	DNN with 3+ L	SN with 1L	SN with 2+ L
ReLU	$\text{ReLu}^{+/-}$	$\text{ReLU}^{+/-}$, $\text{Capped Ramps}^{+/-}$	might freeze	$\text{ReLU}^{+/-}$, $\text{Capped Ramps}^{+/-}$	RF, GRF, $\text{ReLU}^{+/-}$, $\text{Capped Ramps}^{+/-}$
Leaky ReLU	Leaky ReLU	-	-	Leaky ReLU	RF, GRF, Leaky ReLU
Absolut Value	Absolut Value	Absolut Value, RF	Abs. Value, RF, DRF	-	-
Threshold	Switching sets	Double switches	frozen	-	-
Sign	Switching sets	Double switches	Triple switches	-	Richer switching

Figure 5: Overview over the mentioned libraries. L=Hidden layers , DN= Deep narrow net, SN=Symmetrized net, RF=Reflections, GRF=Generalized reflections, DRF=Double reflections.

A.12 Reflections Overview

Activation functions	Normal Reflection	Double Reflection	Generalized Reflection
ReLU	symmetrized network & more than one hidden layer	not mentioned	symmetrized network & more than one hidden layer
Leaky ReLU	symmetrized network & more than one hidden layer	not mentioned	symmetrized network & more than one hidden layer
Absolute Value	more than one hidden layer	more than two hidden layers	not mentioned
Threshold	-	-	-
Sign	-	-	-

Figure 6: Overview about which network architectures can learn different reflection features.

A.13 Additional Feature Plots

A.13.1 Negative ReLU Features

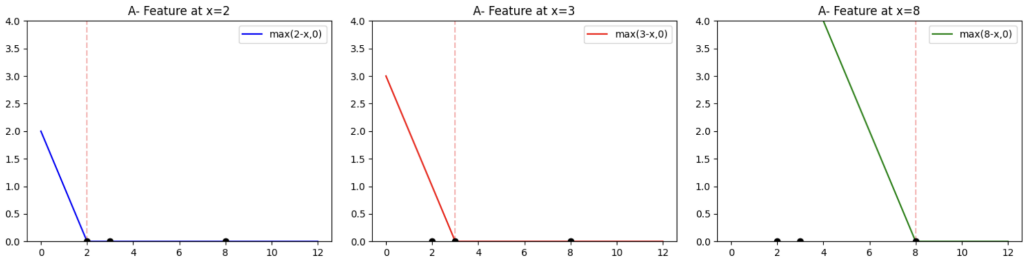

Figure 7: Illustration of the negative ReLU features for data points 2, 3, 8.

A.14 Mentioned Figures from the Papers

A.14.1 Figures from Ziyin et al. (2024)

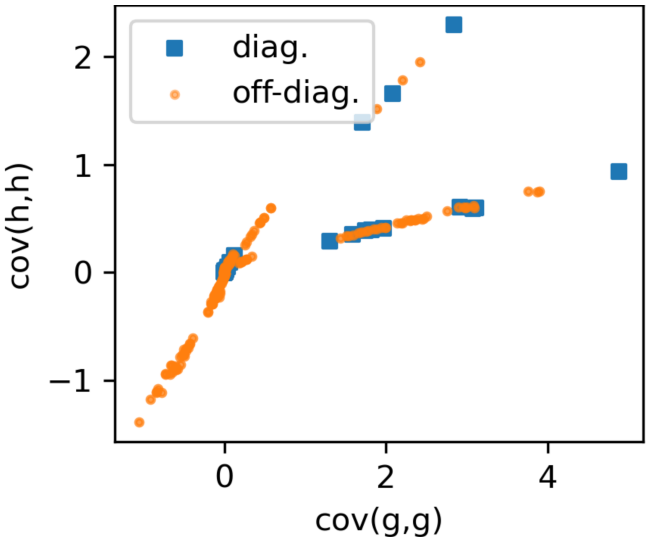

Figure 7 from Ziyin et al. (2024). 'The response diversity (the diagonal terms of the covariance) and correlation is coupled to the plasticity of neurons'

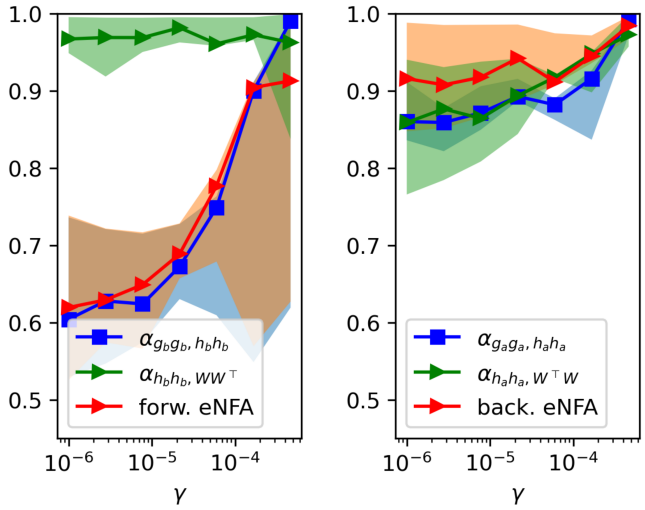

Figure 8 from Ziyin et al. (2024). 'After the training of a six-layer eight-head fully connected network, all six relations hold strongly at a large weight decay. For a small weight decay, at least one forward and backward relation holds strongly. The shaded region shows the variation across five hidden layers, and the solid lines show the median of these alignments. At a small γ, the best alignment is between H_a and $W^T W$ for the forward relation, and G_a and WW^T for the backward, in agreement with the theoretical prediction.'

B Electronic Appendix

B.1 GitHub

The code used to generate illustrations und run simulations for this project can be found on **GitHub**.

References

Bach, F. and Chizat, L. (2021). Gradient descent on infinitely wide neural networks: Global convergence and generalization, *arXiv preprint arXiv:2110.08084* .

Bahri, Y., Dyer, E., Kaplan, J., Lee, J. and Sharma, U. (2024). Explaining neural scaling laws, *Proceedings of the National Academy of Sciences* **121**(27): e2311878121.

Bartlett, P. L., Foster, D. J. and Telgarsky, M. J. (2017). Spectrally-normalized margin bounds for neural networks, *Advances in neural information processing systems* **30**.

Ergen, T. and Pilanci, M. (2020). Convex geometry of two-layer relu networks: Implicit autoencoding and interpretable models, *International Conference on Artificial Intelligence and Statistics*, PMLR, pp. 4024–4033.

Ergen, T. and Pilanci, M. (2021a). Convex geometry and duality of over-parameterized neural networks, *Journal of machine learning research* **22**(212): 1–63.

Ergen, T. and Pilanci, M. (2021b). Revealing the structure of deep neural networks via convex duality, *International Conference on Machine Learning*, PMLR, pp. 3004–3014.

Hochreiter, S. (1991). Untersuchungen zu dynamischen neuronalen netzen, *Diploma, Technische Universität München* **91**(1): 31.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J. and Amodei, D. (2020). Scaling laws for neural language models, *arXiv preprint arXiv:2001.08361* .

Masarczyk, W., Ostaszewski, M., Imani, E., Pascanu, R., Miłoś, P. and Trzcinski, T. (2024). The tunnel effect: Building data representations in deep neural networks, *Advances in Neural Information Processing Systems* **36**.

Papyan, V., Han, X. and Donoho, D. L. (2020). Prevalence of neural collapse during the terminal phase of deep learning training, *Proceedings of the National Academy of Sciences* **117**(40): 24652–24663.

Pilanci, M. and Ergen, T. (2020). Neural networks are convex regularizers: Exact polynomial-time convex optimization formulations for two-layer networks, *International Conference on Machine Learning*, PMLR, pp. 7695–7705.

Radhakrishnan, A., Beaglehole, D., Pandit, P. and Belkin, M. (2023). Mechanism of feature learning in deep fully connected networks and kernel machines that recursively learn features, *arXiv preprint arXiv:2212.13881* .

Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986). Learning representations by back-propagating errors, *nature* **323**(6088): 533–536.

Savarese, P., Evron, I., Soudry, D. and Srebro, N. (2019). How do infinite width bounded norm networks look in function space?, *Conference on Learning Theory*, PMLR, pp. 2667–2690.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. and Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization, *Proceedings of the IEEE international conference on computer vision*, pp. 618–626.

Xu, M., Rangamani, A., Liao, Q., Galanti, T. and Poggio, T. (2023). Dynamics in deep classifiers trained with the square loss: Normalization, low rank, neural collapse, and generalization bounds, *Research* **6**: 0024.

Zeger, E. and Pilanci, M. (2024). Black boxes and looking glasses: Multilevel symmetries, reflection planes, and convex optimization in deep networks, *arXiv preprint arXiv:2410.04279* .

Zeger, E., Wang, Y., Mishkin, A., Ergen, T., Candès, E. and Pilanci, M. (2024). A library of mirrors: Deep neural nets in low dimensions are convex lasso models with reflection features, *arXiv preprint arXiv:2403.01046* .

Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks, *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, Springer, pp. 818–833.

Zhang, Y., Qu, Q. and Wright, J. (2022). From symmetry to geometry: Tractable nonconvex problems, *arXiv preprint arXiv:2007.06753* .

Ziyin, L. (2024). Symmetry induces structure and constraint of learning, *Forty-first International Conference on Machine Learning*.

Ziyin, L., Chuang, I., Galanti, T. and Poggio, T. (2024). Formation of representations in neural networks, *arXiv preprint arXiv:2410.03006* .

Ziyin, L., W. M. L. H. and Wu, L. (2024). Loss symmetry and noise equilibrium of stochastic gradient descent.

Declaration of authorship

I hereby declare that the report submitted is my own unaided work. All direct or indirect sources used are acknowledged as references. I am aware that the Thesis in digital form can be examined for the use of unauthorized aid and in order to determine whether the report as a whole or parts incorporated in it may be deemed as plagiarism. For the comparison of my work with existing sources I agree that it shall be entered in a database where it shall also remain after examination, to enable comparison with future Theses submitted. Further rights of reproduction and usage, however, are not granted here. This paper was not previously presented to another examination board and has not been published.

Munich, Feb. 11, 2025
Location, date

JSchenich

Name