When Representations Align:
Universality in Representation Learning Dynamics

Loek van Rossem [1]   **Andrew M. Saxe** [1,2]

Abstract

Deep neural networks come in many sizes and architectures. The choice of architecture, in conjunction with the dataset and learning algorithm, is commonly understood to affect the learned neural representations. Yet, recent results have shown that different architectures learn representations with striking qualitative similarities. Here we derive an effective theory of representation learning under the assumption that the encoding map from input to hidden representation and the decoding map from representation to output are arbitrary smooth functions. This theory schematizes representation learning dynamics in the regime of complex, large architectures, where hidden representations are not strongly constrained by the parametrization. We show through experiments that the effective theory describes aspects of representation learning dynamics across a range of deep networks with different activation functions and architectures, and exhibits phenomena similar to the "rich" and "lazy" regime. While many network behaviors depend quantitatively on architecture, our findings point to certain behaviors that are widely conserved once models are sufficiently flexible.

1. Introduction

One of the major challenges currently faced in the theory of deep learning is the issue of scalability. Although exact solutions to the learning dynamics have been found for some simple networks (Saad & Solla, 1995; Saxe et al., 2014; Mei et al., 2018; Arora et al., 2019; Saxe et al., 2022), under these analyses, any small change to the network architecture

requires one to significantly amend the analysis. A solution to the learning dynamics of one architecture does not necessarily transfer to another architecture, a rather inconvenient fact considering the large range of different models available. Moreover, state of the art models have a complexity far exceeding what can be reasonably solved analytically.

Studies have shown that although the exact representation between different neural networks may differ, some important aspects of the computational structure they learn are universal (Maheswaranathan et al., 2019; McMahan et al., 2021; Huang et al., 2021). Even for certain areas in the brain, similarities in representational structure have been found with deep neural networks trained on natural image data (Yamins et al., 2014; Yamins & DiCarlo, 2016; Saxe et al., 2021; Conwell et al., 2023; Bracci et al., 2023). The data used in learning may play a larger role than the implementation details of the model, a fact which has been empirically demonstrated in vision transformers and convolutional neural networks (Bouchacourt et al., 2021; Conwell et al., 2023) and studied theoretically (Goldt et al., 2019). Additionally, neural scaling laws describe a formal relationship between model or dataset size and performance that appears to hold without making direct reference to architecture (Kaplan et al., 2020; Hoffmann et al., 2022). More recently, it has been found that an alternative architecture performs roughly the same as the transformer, given a similar parameter count (Wang et al., 2022). These observations together license universal theories: mathematical descriptions of learning that sidestep certain implementation details in order to model behavior common among architectures.

Although a universal theory by necessity cannot completely capture all details of any particular architecture, it has distinct advantages. Derived results may apply to many different learning systems. In particular, this could include highly complicated machine learning models as well as the brain, systems which are difficult to study theoretically otherwise. Furthermore, such an abstract approach forces us to ignore many details not essential to describing specific learning behaviors. This may aid in deriving results, as it reduces the mathematical complexity of the theory. It will also focus our attention to only the most crucial concepts needed to understand learning dynamics, providing a better sense of how

[1]Gatsby Computational Neuroscience Unit, University College London [2]Sainsbury Wellcome Centre, University College London. Correspondence to: Loek van Rossem <loek.rossem.22@ucl.ac.uk>.

we should be thinking about these highly complex systems.

Here we propose an approach to modelling universal representation learning, striving to provide conceptual explanations for phenomena universally observed in learning systems. We seek to summarize known results in deep learning within a single framework. Our main contributions are as follows:

- In Section 2, we derive an effective theory for the interaction between two datapoints with nearby representations during training in the expressive regime, where the neural network is large and complex enough to not be strongly constrained by its parametrization.

- In Section 3, we demonstrate the existence of universal behavior within representational learning dynamics by showing that the derived theory describes the dynamics of a range of deep networks with varying activation functions and architectures.

- In Section 4, we look at the final representational structure predicted by the theory, and find two qualitatively different regimes, one based on random initializations, and one based on structure in the data. We run experiments on deep neural networks trained on toy datasets, and find similar representations as predicted by the theory.

1.1. Related Work

Exact solutions in simple architectures. Considerable progress has been made in the theoretical analysis of deep linear neural networks, for example, the loss landscape is well understood (Baldi & Hornik, 1989; Kawaguchi, 2016), and exact solutions have been obtained for specific initial conditions (Saxe et al., 2014; Lampinen & Ganguli, 2019; Braun et al., 2022; Pesme & Flammarion, 2023). This is in contrast to other architectures, for which only few exact analyses of learning dynamics exists, see e.g. (Seung et al., 1992; Saad & Solla, 1995; Mei et al., 2018; Saxe et al., 2022). Although some of these results derived for linear networks reflect behavior in more complex architectures, there is no precise mathematical connection in the dynamics of linear and nonlinear neural networks. In this work, we will attempt to make this relationship more explicit.

The neural tangent kernel. A notable exception in terms of universal solutions is that of the neural tangent kernel literature (Lee et al., 2019; Jacot et al., 2018; Golikov et al., 2022), which does provide exact solutions applicable to a wide range of models. The limitation here is that the neural tangent kernel solutions are only valid for infinitely wide neural networks at a specific weight initialization scale (Woodworth et al., 2020). In particular, the initialization

regime used is one of relatively high weights, where the dataset does not impact the representation (Chizat et al., 2019; Flesch et al., 2022). We seek a similar universal theory, applicable to the feature learning regime, where the dataset controls the representation.

Implicit biases in gradient descent. A large body of work has investigated gradient descent as a source of generalization performance in deep neural networks, as opposed to explicit regularization and inductive biases in the architecture (Neyshabur et al., 2015; Gunasekar et al., 2018; Chizat & Bach, 2020; Soudry et al., 2018). This work is mainly either empirical or focused on analyses for examples of specific architectures. Here instead we consider a modeling approach, aiming to provide intuition into the universal nature of implicit bias and observed similarities between representational structures across architectures.

Local elasticity. One motivating factor for the method used here is the empirical phenomenon of local elasticity (He & Su, 2020; Zhang et al., 2021), observed for non-linear neural networks performing classification. A model is said to exhibit local elasticity in the case that updating one feature vector does not significantly perturb feature vectors dissimilar to it. This allows us to consider neural network learning dynamics as a local process, suggesting that the local interaction between representations may already provide insight into the behavior of the entire dataset.

Physics inspired approaches. The approach to modelling a two point interacting between datapoints taken here is inspired by intuitions from physics surrounding interacting particles. Physics-inspired particle approaches are prominent in the deep learning theory (Seung et al., 1992; Rotskoff & Vanden-Eijnden, 2018a;b; Baek et al., 2024), most notably the application of mean field theory (Saul et al., 1996; Poole et al., 2016; Xiao et al., 2018; Mei et al., 2018; 2019; Pandey & Xu, 2019).

2. The Model

The approach we will be taking here is motivated by universal approximation theorems (Hornik et al., 1989; Barron, 1993; Csáji, 2001), which state that given enough parameters, non-linear neural networks can learn to approximate any arbitrary smooth function. We can thus consider the process of training a neural network as an optimization of a smooth function to fit the data. Constraints coming from the architecture of the neural network may affect the dynamics of the optimization process, but since here we are interested in modelling these function dynamics without reference to the details of the architecture, we will choose to ignore this.

2.1. Function Dynamics

More explicitly, let us consider some unspecified deep neural network $f_\theta : X \to Y$, that is trained on a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N, \forall x_i \in X, \forall y_i \in Y$. During training its set of parameters θ is optimized to reduce the mean squared error loss function:

$$L(f_\theta) = \frac{1}{2}\langle ||f_\theta(x_i) - y_i||^2 \rangle_\mathcal{D}, \tag{1}$$

where $\langle \cdot \rangle_\mathcal{D}$ is the average over the dataset $\mathcal{D}$.

Since we are interested in studying the dynamics of the representational space, we will split f_θ into two smooth maps, an encoder map $h_\phi : X \to H$ which assigns to each input in X a representation in some intermediate layer H in the neural network, and a decoder map $y_\psi : H \to Y$ which assigns a predicted output to each possible representation in H. We may then write the loss in Equation (1) as:

$$L(h_\phi, y_\psi) = \frac{1}{2}\langle ||y_\psi(h_\phi(x_i)) - y_i||^2 \rangle_\mathcal{D}. \tag{2}$$

The dynamics on these maps follow from applying the gradient descent update rule to the parameters ϕ and ψ defining them:

$$\Delta\phi = -\eta\nabla_\phi L \quad \text{and} \quad \Delta\psi = -\eta\nabla_\psi L, \tag{3}$$

where η is the learning rate. For sufficiently small η we can take the continuous time limit

$$\frac{\mathrm{d}}{\mathrm{d}t}\phi = -\frac{1}{\tau}\nabla_\phi L \quad \text{and} \quad \frac{\mathrm{d}}{\mathrm{d}t}\psi = -\frac{1}{\tau}\nabla_\psi L, \tag{4}$$

where $\tau := \frac{1}{\eta}$. The problem with Equation (4) is that it explicitly depends on the parametrization of the neural network, whereas a universal approach requires ignoring such implementation details. Ideally we would like to optimize the loss L directly with respect to the maps h_ϕ and y_ψ, since if the model has high *expressivity*, i.e. the architecture is large and complex enough, perhaps it has enough freedom to behave as an arbitrary smooth map. However, it is unclear how this can be achieved mathematically[1].

2.2. Two Point Interaction

In general, the modelling issue of parametrization dependent dynamics does not appear to have a resolution. So let us instead restrict ourselves to a simpler case, where the only property we do know about the unspecified network, the smoothness of the maps h_ϕ and y_ψ, can be exploited. During training the representations of different datapoints move around in the hidden space H, and may get near each other and interact. Understanding such an interaction on its own may already provide some insight into the representational learning behavior of the model on the full dataset.

To model this two point interaction, let us consider a dataset $\mathcal{D} = \{(x_1, y_1), (x_2, y_2)\}$. If the representations $h_\phi(x_1)$ and $h_\phi(x_2)$ are close enough, then by smoothness it is reasonable to a take a linear approximation of h_ϕ and y_ψ around the mean of the two points:

$$\begin{aligned} h_\phi(x_i) &\approx h_\phi(\frac{x_2 + x_1}{2}) + \frac{1}{2}D_h(x_i - x_{\neg i}) \\ y_\psi(x_i) &\approx y_\psi(h_\phi(\frac{x_2 + x_1}{2})) + \frac{1}{2}D_y(D_h(x_i - x_{\neg i})) \end{aligned} \tag{5}$$

where D_h and D_y are the Jacobian matrices of h_ϕ and y_ψ respectively.

Presumably, as gradient descent optimizes ϕ and ψ in order to reduce the loss, at the two point interaction this will effectively optimize the linearization parameters D_h, D_y and $\bar{y} := y_\psi(h_\phi(\frac{x_2+x_1}{2}))$ in order to minimize the loss[2]. Assuming expressivity, i.e. "enough freedom" in the architecture such that there are no additional constraints on the dynamics, we can model the dynamics by applying gradient descent directly to the linearization parameters, i.e.

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\bar{y} &= -\frac{1}{\tau_{\bar{y}}}\frac{\partial L}{\partial \bar{y}} \\ \frac{\mathrm{d}}{\mathrm{d}t}D_h &= -\frac{1}{\tau_h}\frac{\partial L}{\partial D_h} \\ \frac{\mathrm{d}}{\mathrm{d}t}D_y &= -\frac{1}{\tau_y}\frac{\partial L}{\partial D_y}, \end{aligned} \tag{6}$$

where $1/\tau_{\bar{y}}, 1/\tau_h$ and $1/\tau_y$ are effective learning rates.

Note that these dynamics are not necessarily equal to the actual dynamics on the linearization parameters induced by the model parameters ϕ and ψ. Equation (6) is the main modelling assumption, and intended as an *effective theory* in the regime of large complex architectures, where behavior is unconstrained by their parametrization. The details of how h_ϕ and y_ψ are parameterized determine the effective learning rates $1/\tau_{\bar{y}}, 1/\tau_h$ and $1/\tau_y$, but are otherwise not present in the effective theory.

In fact, if one were to interpret the linearization parameters as model weights, then the effective theory dynamics are mathematically the same as that of a two layer linear network with bias and differing learning rates for each of the layers. However, we are not using a two layer linear network to model the neural network here, instead we are locally modeling a two point interaction.

---

[1]One may be tempted to try replacing the right hand side of Equation (4) with functional derivatives of L with respect to h_ϕ and y_ψ, however this will result in a term involving the gradient of y_ψ, and the dynamics of this gradient explicitly depend on ψ.

[2]Potentially $h_\phi(\frac{x_2+x_1}{2})$ should also be considered as a linearization parameter, but it will turn out to not play a role in the following analysis.

2.3. Reduction to a 3-dimensional System

To simplify the dynamics, we study solutions where the representations of the two datapoints either move together or apart but do not rotate. Strikingly, as derived in Appendix A.1, this results in a self-contained system of three scalar variables:

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}||dh||^2 = & -\frac{1}{\tau_h}||x_2 - x_1||^2 w \\
\frac{\mathrm{d}}{\mathrm{d}t}||dy||^2 = & -w(\frac{1}{\tau_y}||dh||^2 + \frac{1}{\tau_h}||x_2 - x_1||^2 \frac{||dy||^2}{||dh||^2}) \\
\frac{\mathrm{d}}{\mathrm{d}t}w = & -\frac{1}{\tau_y}\frac{1}{2}(3w - ||dy||^2 + ||y_2 - y_1||^2)||dh||^2 \\
& -\frac{1}{\tau_h}\frac{1}{2}||x_2 - x_1||^2 \frac{(||dy||^2 + w)w}{||dh||^2},
\end{aligned}
\tag{7}
$$

where

$$
\begin{aligned}
||dh||^2 := & ||D_h(x_2 - x_1)||^2 \\
||dy||^2 := & ||D_y D_h(x_2 - x_1)||^2 \\
w := & ||dy||^2 - (D_y D_h(x_2 - x_1))^\top(y_2 - y_1).
\end{aligned}
\tag{8}
$$

We will find that this system still exhibits interesting behavior worth studying. Figure 1 illustrates the relevant variables of this system. The main quantity we care about here is the representational distance $||dh||$, as it may provide insight into what kind of representational structures are learned. The difference in predicted outputs $||dy||$ is useful for modelling loss curves. Finally, there is the additional variable w, which controls the representational velocity. This can be interpreted as an alignment as it only contains additional information about the angle between the predicted outputs and the target outputs.
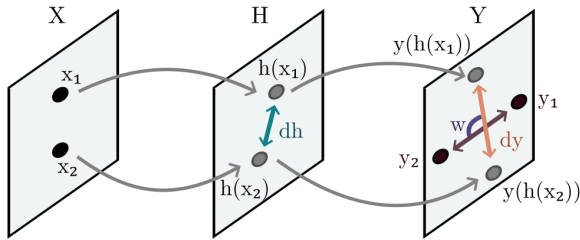

Figure 1. Overview of the effective theory for the two point interaction. For two datapoints x_1 and x_2 self-contained dynamics are defined on their representational difference $||dh||$, predicted output difference $||dy||$, and output alignment w.

3. Universal Dynamics

The dynamics in Equation (7) contain no direct reference to the implementation details of the neural network, only two effective learning rates $1/\tau_h$ and $1/\tau_y$, which are unknown quantities and may differ across neural networks. Implementation details have essentially been abstracted away into these two constants. Given the right effective learning rates, solutions to these dynamics should model the learning dynamics of neural networks regardless of the architecture.

3.1. Comparison of the Dynamics

In Figure 2 we compare the learning dynamics of a neural network (20 layer, leaky ReLU, 500 units per layer) trained on a two point dataset to numerical solutions of the effective theory, as well as variations of this network with different architectures and nonlinearities. Despite only needing to fit two constants, the theory matches all three curves quite well for all networks. Note that this is not solely a result of having two degrees of freedom (see Appendix C.1).

3.2. Varying Initial Weights

For small initial weights, the variables $||dh||, ||dy||$ and w start small[3], so the right hand side of Equation (7) must be small too. Therefore, the dynamics are initially stagnant resulting in plateau-like behavior, whereas for large initial weights we have immediate exponential decay, as derived analytically in Appendix A.5. Such a transition from an initial plateau to immediate decay is observed experimentally, as can be seen in Figure 3 (top). Going to even larger initial weights, the representations start off far apart, so the linear approximation breaks down and the effective theory no longer matches the network.

3.3. The Training Loss

In Appendix A.2, we derive an expression for the training loss as a function of $||dy||^2$ and w:

$$
\begin{aligned}
L = & \frac{1}{2}||(y(0) - \frac{y_2 + y_1}{2})e^{-\frac{t}{\tau_{\bar{y}}}}||^2 \\
& + \frac{1}{4}(w + \frac{1}{2}(||y_2 - y_1||^2 - ||dy||^2))),
\end{aligned}
\tag{9}
$$

where the third effective time constant $1/\tau_{\bar{y}}$ shows up. The first term in the expression is an exponential decay coming from fitting the output mean. The second term depends on $||dy||^2$ and w, which causes a plateau in the loss at small weights, due to the jump discussed in Section 3.2. The appearance of such a plateau at small weights is observed for the true dynamics, as can be seen in Figure 3 (bottom).

---

[3]There is an assumption here about the architecture. In principle, small weights could mean anything if nothing is know about how the smooth maps are parameterized, however, for typical neural network architectures these variables are small.
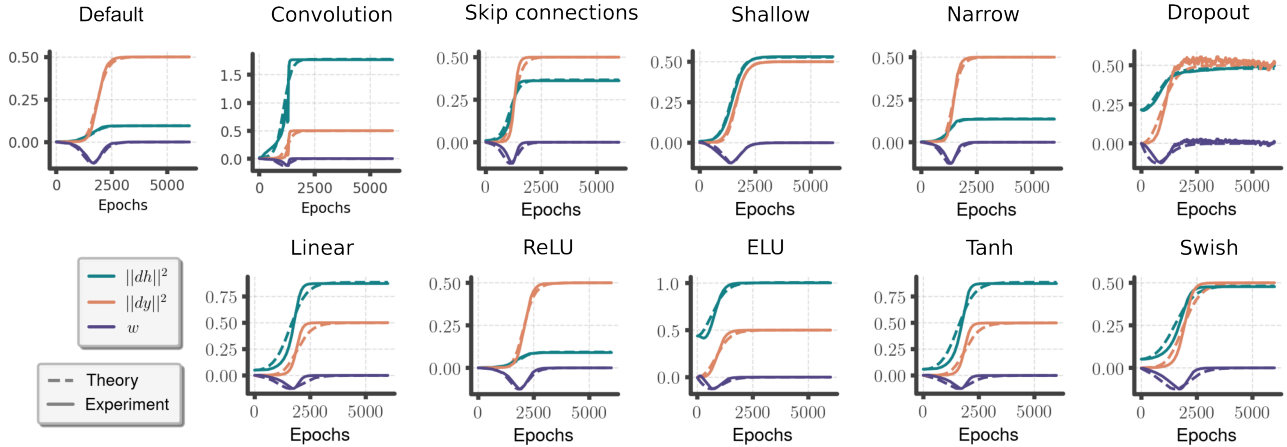
Figure 2. Universal learning dynamics among different architectures. The representational distance $||dh||^2$, prediction difference $||dy||^2$ and output alignment w during training on a two point dataset amongst architectures with varying connectivity (top) and nonlinearities (bottom), matches the theory after fitting two constants. The architectures used are all variations of the default architecture and initialized at small weights so as to be in the expressive feature learning regime. Details for all experiments can be found in Appendix B.

3.4. Larger Datasets

We have only considered the interaction of two points in *isolation*. Real datasets consist of many points, which may influence the behavior two interacting datapoints exhibit. To investigate the validity of the theory in this setting, we trained a model on the MNIST dataset, and tracked two distinguishable datapoints. The resulting dynamics (Figure 4) still qualitatively resemble the theory.
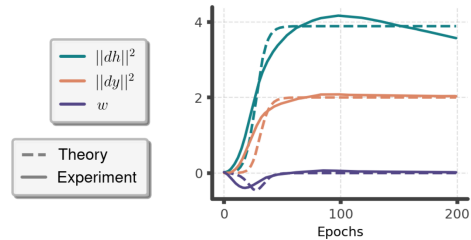

Figure 4. Learning dynamics of a randomly selected zero digit and one digit in MNIST compared against the theory after fitting two constants. The deep neural network has 4 fully connected layers, 100 units per layer, leaky ReLU activation and is initialized at small weights.
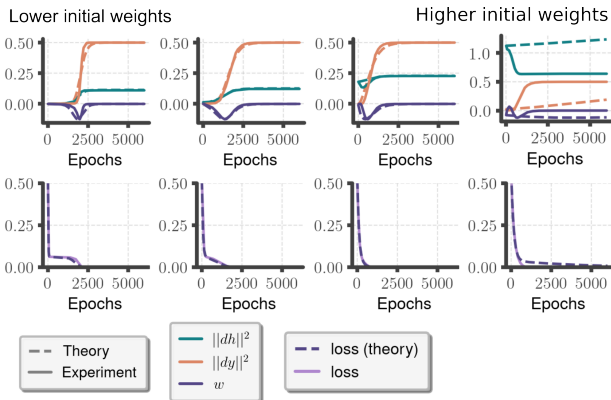

Figure 3. Dynamics of the 3-dimensional system (**top**) and training loss (**bottom**) at varying initial weights. The default architecture (20 fully connected layers, 500 units per layer, leaky ReLU) is trained on two datapoints and compared to the effective theory, after fitting two effective learning rates for the 3-dimensional system and one additional effective learning rate for the loss. Plateau-like behavior in the representational distance and loss can be seen at small initializations, but disappears at larger initial weights. At very high initial weights, when the representational distance starts off already large, the approximation breaks down as expected.

4. Structured Representations

If it is true that constraints coming from smoothness determine part of the behavior in neural network learning dynamics, what effect do these constraints have on the learned representation? To investigate this, we compute the final representational distance in the theory and compare it to neural networks trained on a few toy datasets.

4.1. Lazy and Rich Learning Regimes

We derive the fixed points of Equation (7) in Appendix A.3, using a reduction to a 2-dimensional system with dependence on initial conditions. We find that there is only one stable fixed point:

$$||dh||^2 = \frac{1}{2}A_{\text{high}} + \sqrt{\frac{1}{4}A_{\text{high}}^2 + A_{\text{low}}^2}, \qquad (10)$$
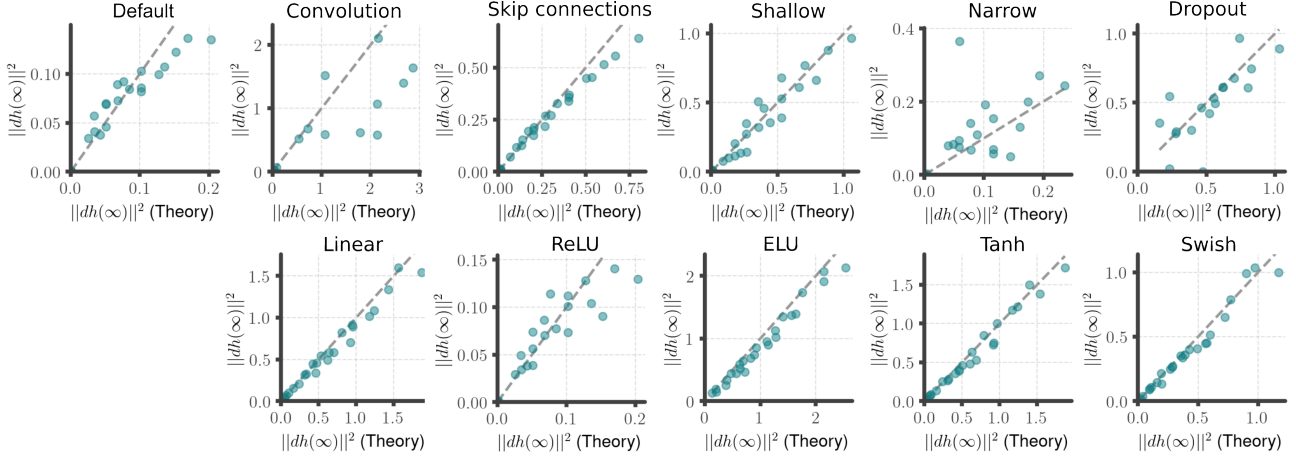
5

Figure 5. Structured learning at small initial weights. The final representational distance of two point experiments with varying input and target output distances compared against the theory (Equation (10)). Each dot represents a single trial, i.e. one training run. The effective learning rates for each architecture have been fit by averaging over 50 trials at *fixed* input and target output distances.

where

$$A_{\text{high}} = \left(\frac{||dh(0)||^2}{||x_2 - x_1||^2} - \frac{\tau_y}{\tau_h} \frac{||dy(0)||^2}{||dh(0)||^2} \right) ||x_2 - x_1||^2$$

$$A_{\text{low}} = \sqrt{\frac{\tau_y}{\tau_h}} ||x_2 - x_1|| \cdot ||y_2 - y_1||,$$

$$\tag{11}$$

hence we expect this to be the final representational distance.

When the initial weights are large, i.e.

$$\frac{||dh(0)||^2}{||x_2 - x_1||^2}, \frac{||dy(0)||^2}{||dh(0)||^2} \gg 1, \tag{12}$$

the final representational distance converges to A_{high}, which depends on the *input* structure of the data and the *random initialization*. In the case that they are small

$$\frac{||dh(0)||^2}{||x_2 - x_1||^2}, \frac{||dy(0)||^2}{||dh(0)||^2} \ll 1, \tag{13}$$

it converges to A_{low}, which depends on the *input* and *output* structure of the data.

This separation between a random regime and a structured regime closely resembles the phenomenon of the "rich" and "lazy" learning that deep neural networks exhibit (Chizat et al., 2019; Flesch et al., 2021; 2022; Atanasov et al., 2022). In particular, a crucial factor determining the regime is the initial weight scale (Woodworth et al., 2020).

4.2. Rich Structure

The representation at small initial weights is of particular interest, as it appears to be based on a structure found in the dataset, rather than formed by the random weight initialization. Note that again the only unknown quantities

here are the two effective learning rates. By fitting these to the learning trajectories at a fixed choice of input and target output distances of the datapoints, we can extrapolate to predictions about the final representational distance for other distances, as shown in Figure 5.

4.3. Intuition

There is an intuition as to why smoothness constraints can result in these two different representational regimes. At large weights, representations are initially far apart, so the flexibility of the network allows the decoder to individually learn the correct output freely for each datapoint, without need of adjusting the representational structure. The dominating structure in the final representation is determined by the structure already present at initialization.

At small weights, representations are nearby and thus, due to smoothness constraints, they can no longer freely learn outputs without influencing each other. To illustrate this, consider the case where we have two datapoints with the same target output but only one already has the correct output prediction. There are two ways to learn the correct output for the other datapoint: (1) the decoder map can adjust at its representation and (2) the encoder map can move the representation towards the representation of the other datapoint, which due to continuity of the decoder map will align its output to that of the other datapoint, which already has the correct output. The second option therefore causes an effect on representations, where they move closer when their outputs agree. The opposite effect also holds: when two datapoints have different target outputs but the same representation, by continuity they cannot both have the correct output prediction. Their representations must move apart to fit the data.
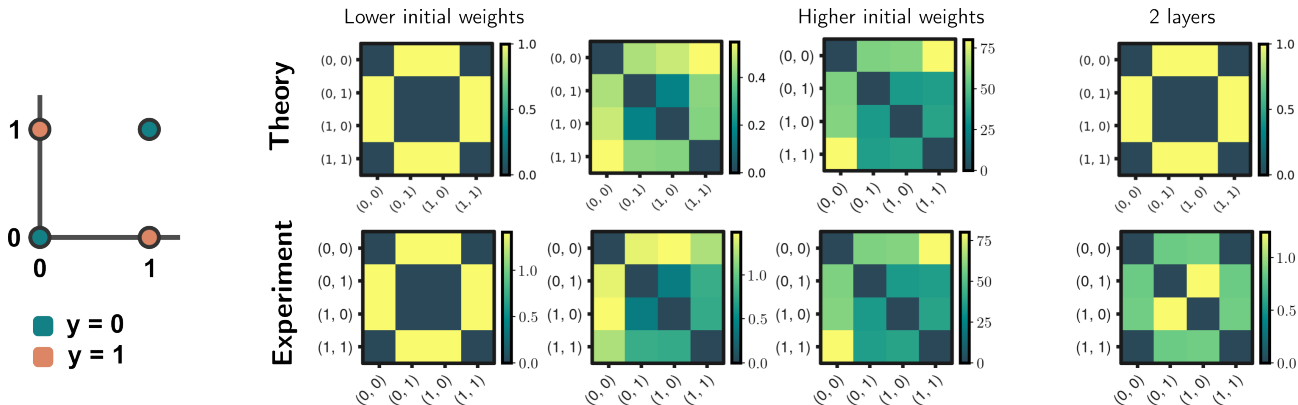
Figure 6. Representational structure in the XOR task. The XOR problem (**left**) consists of 4 datapoints with two input features and one alternating output feature. The learned hidden representational pair-wise distances match the theory prediction at varying initial weight scales for the default 20 layer network (**middle**), but not for a 2 layer network at small weights (**right**).

Therefore, we see that at small weights there is a structured effect on the representation, dependent on the output data. This may help explain why overparameterized deep neural networks sometimes learn a representational structure, even without regularization. With a large dataset the interactions between datapoints may be numerous enough to dominate the representational learning behavior. In this case, *the fastest way to learn the correct output is to learn a structured representation*, and gradient descent looks for the fastest way to minimize the loss. At small weights, the fastest way to learn may not be overfitting but instead structured learning, where representations due to their closeness benefit from each other.

4.4. XOR Task

To investigate the types of structures the theory is able to predict, we first consider the example of the XOR task. Here the aim is to learn a simple binary non-linear computation; there are only four datapoints, with two possible outputs, as illustrated in Figure 6 (left). The theory predicts that at small initial weights, representations of datapoints with the same target output will merge, as $||y_2 - y_1||$ is zero for such pairs. At large initial weights no such merging should occur. As can be seen in Figure 6 (middle), this matches the learned representation of the large 20-layer default network. However, the theory fails to anticipate the merging of only one pair when the network is reduced to a two layer architecture, stressing the importance of the high model expressivity assumption (Figure 6 (right)).

4.5. Feature Collapse

Next we consider an example of feature collapse explored in (Flesch et al., 2022). Here a network is trained on a dataset consisting of images of Gaussian blobs with varying x and

y positions, as shown in Figure 7. The task contains two contexts, one where the aim is to predict the x position of the blobs mean, and one where the aim is to predict its y position.
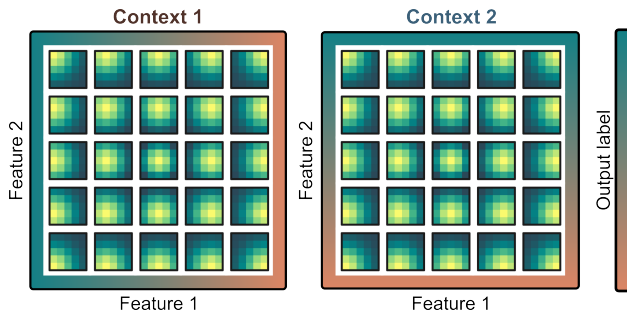

Figure 7. A toy dataset with two orthogonal features. The dataset consists of 2*30*30 datapoints, each 5*5 pixel images of Gaussian blobs, centered at varying x and y positions. A one hot encoded context variable is added, specifying whether the goal is to learn the feature 1 (x-position) or the feature 2 (y-position).

The theory's prediction at small initial weights is that for each context the representation will collapse along the feature direction which is irrelevant to the task, as the output label does not change along this direction. At large initial weights the representation will retain the input data structure, i.e. one 2-d square for each context. These two structures at different weight regimes were indeed observed for a two layer ReLU network in (Flesch et al., 2022). Because the prediction here is universal we reran the experiment for a different architecture (5-layer tanh), and found similar results, as shown in Figure 8.
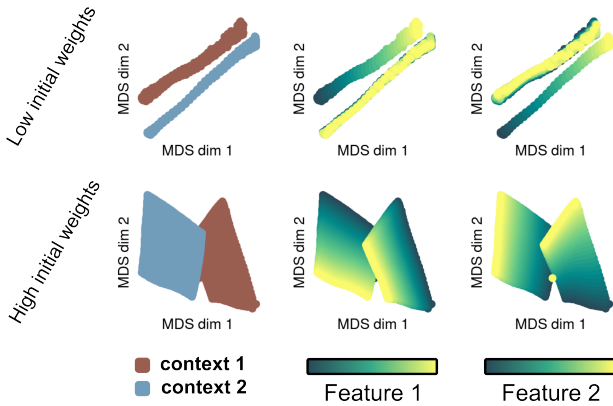
Figure 8. Representational structure in the orthogonal feature task. A multidimensional scaling projection of the representational structure of a 5-layer, tanh, fully connected network with 100 hidden units per layer, trained on the orthogonal feature dataset. At small initial weights the task-irrelevant direction collapses in each context, but remains at large initial weights.

4.6. MNIST

When dealing with more complex datasets we cannot always apply Equation (10) directly. This is because it only predicts an interaction between two datapoints. Even simple geometric constraints such as the triangle inequality are not expected. Nevertheless, in Figure 9 we see that part of the representational structure still resembles the theory when a network is trained on the MNIST dataset. Here we applied a simple method to reduce violations of the triangle inequality, as detailed in Appendix B.4.
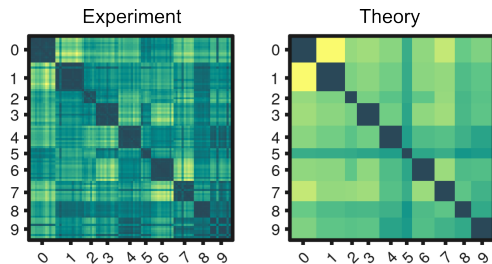

Figure 9. Representational structure on the MNIST dataset at small initial weights. Pairwise distances of the first 100 datapoints in MNIST after averaging over 50 trials of a 4-layer, leaky-ReLU, fully connected network trained on the full dataset (**left**) compared to the theory after exponential weighing (**right**). The Pearson correlation coefficient between the theory and experiment is 0.703 and when not factoring in the diagonal blocks it is 0.512.

5. Depth

In the theory we consider the representation dynamics at "some intermediate layer H". Deep neural networks have many layers at which the representations can be observed. This raises the question: how do the dynamics depend on the depth of the chosen intermediate layer?

5.1. Predictivity of the effective theory

Let us begin by determining at which layers the effective theory is still valid. For the linear approximation to be accurate, we need the representations to start off near each other. Suppose that at small initial weights the average activational gain factor of each layer is some constant $G < 1$. The initial representational distance as a function of the depth n scales roughly as

$$||dh(0)||^2 \sim G^n ||x_2 - x_1||^2. \tag{14}$$

This is a decreasing function, so we expect the theory to be more accurate in the later layers of the network. Indeed, as can be seen in Figure 10 (red) the effective theory models the learning behavior more accurately at the later layers.
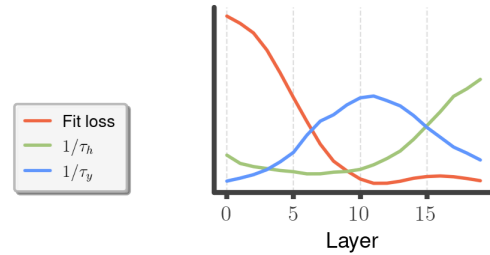

Figure 10. Fit loss and fitted effective learning rates of the theory as a function of the hidden layer depth. The default model (20 fully connected layers, 500 units per layer, leaky ReLU) is trained on the two point dataset, and the considered hidden layer is varied. The y-axis is not to scale. We can see that the theory is accurate only in the second half of the network. In this regime the effective learning rate of the encoder increases with depth, whereas the effective learning rate of the decoder decreases.

5.2. Effective Learning Rates

How do we expect the effective learning rates to change at different hidden layers? The standard gradient descent update contains a sum over the parameters, so it makes changes to a map proportional to the number of parameters it contains. As we consider later hidden layers, the number of parameters in the encoder map increases, whereas the number of parameters in the decoder map decreases. This should result in the effective learning rate for the encoder increasing with the depth and the effective learning rate for

the decoder decreasing with the depth. We can see that in Figure 10 this relationship holds for the later layers of the network, where the theory is accurate, although at earlier layers the effective learning rate for the decoder seems to increase. Notably, these results suggest that representational dynamics in the later layers behave in a similar manner, up to a rescaling of effective learning rates.

6. Conclusion

We have introduced an effective theory for the universal part of the learning dynamics common amongst different neural network architectures, and shown it already exhibits structured representational learning, in particular at small initial weights. The aim here is not so much to point to this specific theory as an accurate universal model, rather it is to suggest that some of the essentials required for representation learning may already be contained within gradient descent, as opposed to solely coming from inductive biases contained in the choice of architecture. Therefore, perhaps a more universal perspective to learning dynamics is possible. Additionally, the effective theory underlines the initial weight scale as a crucial factor for the formation of the final representational structure.

A significant limitation to this specific approach for modelling universality is that there is no clear path to extending it to larger datasets. More work is needed to find a satisfying resolution to dealing with more complex data. Additionally, many architectures do have inductive biases that affect the learned representation, and the theory here does not take those into account. The representational effect modelled here likely interacts with these inductive biases, causing potentially interesting effects on the representation that are worth exploring.

Impact Statement

This paper presents work whose goal is to increase understanding of deep learning, which may lead to advancements in the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgements

References

Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R., and Wang, R. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 8139–8148, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/dbc4d84bfcfe2284ba11beffb853a8c4-Abstract.html.

Atanasov, A., Bordelon, B., Sainathan, S., and Pehlevan, C. The Onset of Variance-Limited Behavior for Networks in the Lazy and Rich Regimes, 2022. URL https://arxiv.org/abs/2212.12147.

Baek, D. D., Liu, Z., and Tegmark, M. GenEFT: Understanding Statics and Dynamics of Model Generalization via Effective Theory, 2024. URL https://arxiv.org/abs/2402.05916.

Baldi, P. and Hornik, K. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2(1):53–58, 1989. ISSN 08936080. doi: 10.1016/0893-6080(89)90014-2. URL https://linkinghub.elsevier.com/retrieve/pii/0893608089900142.

Barron, A. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, 1993. ISSN 0018-9448, 1557-9654. doi: 10.1109/18.256500. URL https://ieeexplore.ieee.org/document/256500/.

Bouchacourt, D., Ibrahim, M., and Morcos, A. S. Grounding inductive biases in natural images: invariance stems from variations in data. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 19566–19579, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/a2fe8c05877ec786290dd1450c3385cd-Abstract.html.

Bracci, S., Mraz, J., Zeman, A., Leys, G., and Op de Beeck, H. The representational hierarchy in human and artificial visual systems in the presence of object-scene regularities. *PLoS computational biology*, 19(4):e1011086, 2023. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1011086.

Braun, L., Dominé, C. C. J., Fitzgerald, J. E., and Saxe, A. M. Exact learning dynamics of deep linear networks with prior knowledge. 2022. URL https://openreview.net/forum?id=lJx2vng-KiC.

Chizat, L. and Bach, F. R. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Conference on Learning Theory, COLT 2020, 9-12 July 2020, Virtual Event [Graz, Austria]*, volume 125 of *Proceedings of Machine Learning Research*, pp. 1305–1338. PMLR, 2020. URL http://proceedings.mlr.press/v125/chizat20a.html.

Chizat, L., Oyallon, E., and Bach, F. R. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 2933–2943, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/ae614c557843b1df326cb29c57225459-Abstract.html.

Conwell, C., Prince, J. S., Kay, K. N., Alvarez, G. A., and Konkle, T. What can 1.8 billion regressions tell us about the pressures shaping high-level visual representation in brains and machines?, 2023. URL https://www.biorxiv.org/content/10.1101/2022.03.28.485868v2. Pages: 2022.03.28.485868 Section: New Results.

Csáji, B. *Approximation with Artificial Neural Networks*. PhD thesis, 2001.

Flesch, T., Juechems, K., Dumbalska, T., Saxe, A., and Summerfield, C. Rich and lazy learning of task representations in brains and neural networks, 2021. URL https://www.biorxiv.org/content/10.1101/2021.04.23.441128v1. Pages: 2021.04.23.441128 Section: New Results.

Flesch, T., Juechems, K., Dumbalska, T., Saxe, A., and Summerfield, C. Orthogonal representations for robust context-dependent task performance in brains and neural networks. *Neuron*, 110(7):1258–1270.e11, 2022. ISSN 08966273. doi: 10.1016/j.neuron.2022.01.005. URL https://pubmed.ncbi.nlm.nih.gov/36549271/.

Goldt, S., Mézard, M., Krzakala, F., and Zdeborová, L. Modelling the influence of data structure on learning in neural networks: the hidden manifold model. *ArXiv preprint*, abs/1909.11500, 2019. URL https://arxiv.org/abs/1909.11500.

Golikov, E., Pokonechnyy, E., and Korviakov, V. Neural Tangent Kernel: A Survey, 2022. URL https://arxiv.org/abs/2208.13614.

Gunasekar, S., Lee, J. D., Soudry, D., and Srebro, N. Implicit bias of gradient descent on linear convolutional networks. In *Advances in Neural Information*

Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 9482–9491, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/0e98aeeb54acf612b9eb4e48a269814c-Abstract.html.

He, H. and Su, W. J. The local elasticity of neural networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=HJxMYANtPH.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. d. L., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., Driessche, G. v. d., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Rae, J. W., Vinyals, O., and Sifre, L. Training Compute-Optimal Large Language Models, 2022. URL https://arxiv.org/abs/2203.15556.

Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989. ISSN 08936080. doi: 10.1016/0893-6080(89)90020-8. URL https://linkinghub.elsevier.com/retrieve/pii/0893608089900208.

Huang, T., Zhen, Z., and Liu, J. Semantic Relatedness Emerges in Deep Convolutional Neural Networks Designed for Object Recognition. *Frontiers in Computational Neuroscience*, 15, 2021. ISSN 1662-5188. URL https://www.frontiersin.org/articles/10.3389/fncom.2021.625804.

Jacot, A., Hongler, C., and Gabriel, F. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 8580–8589, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/5a4be1fa34e62bb8a6ec6b91d2462f5a-Abstract.html.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling Laws for Neural Language Models, 2020. URL https://arxiv.org/abs/2001.08361.

Kawaguchi, K. Deep learning without poor local minima. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing*

Systems 2016, December 5-10, 2016, Barcelona, Spain, pp. 586–594, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/f2fc990265c712c49d51a18a32b39f0c-Abstract.html.

Lampinen, A. K. and Ganguli, S. An analytic theory of generalization dynamics and transfer learning in deep linear networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=ryfMLoCqtQ.

Lee, J., Xiao, L., Schoenholz, S. S., Bahri, Y., Novak, R., Sohl-Dickstein, J., and Pennington, J. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 8570–8581, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/0d1a9651497a38d8b1c3871c84528bd4-Abstract.html.

Maheswaranathan, N., Williams, A. H., Golub, M. D., Ganguli, S., and Sussillo, D. Universality and individuality in neural dynamics across large populations of recurrent networks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 15603–15615, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/5f5d472067f77b5c88f69f1bcfda1e08-Abstract.html.

McMahan, B., Kleinman, M., and Kao, J. C. Learning rule influences recurrent network representations but not attractor structure in decision-making tasks. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 21972–21983, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/b87039703fe79778e9f140b78621d7fb-Abstract.html.

Mei, S., Montanari, A., and Nguyen, P.-M. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018. doi: 10.1073/pnas.1806579115. URL https://www.pnas.org/doi/10.1073/pnas.1806579115. Publisher: Proceedings of the National Academy of Sciences.

Mei, S., Misiakiewicz, T., and Montanari, A. Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit. In *Conference on Learning Theory, COLT 2019, 25-28 June 2019, Phoenix, AZ, USA*, volume 99 of *Proceedings of Machine Learning Research*, pp. 2388–2464. PMLR, 2019. URL http://proceedings.mlr.press/v99/mei19a.html.

Neyshabur, B., Tomioka, R., and Srebro, N. In Search of the Real Inductive Bias: On the Role of Implicit Regularization in Deep Learning, 2015. URL http://arxiv.org/abs/1412.6614. arXiv:1412.6614 [cs, stat].

Pandey, M. and Xu, M. Mean Field Theory and its application to deep learning. 2019.

Pesme, S. and Flammarion, N. Saddle-to-Saddle Dynamics in Diagonal Linear Networks, 2023. URL https://arxiv.org/abs/2304.00488.

Poole, B., Lahiri, S., Raghu, M., Sohl-Dickstein, J., and Ganguli, S. Exponential expressivity in deep neural networks through transient chaos. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 3360–3368, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/148510031349642de5ca0c544f31b2ef-Abstract.html.

Rotskoff, G. M. and Vanden-Eijnden, E. Neural Networks as Interacting Particle Systems: Asymptotic Convexity of the Loss Landscape and Universal Scaling of the Approximation Error. *ArXiv*, 2018a. URL https://www.semanticscholar.org/paper/Neural-Networks-as-Interacting-Particle-Systems%3A-of-Rotskoff-Vanden-Eijnden/50ad17c11eeae5982f81e90385a2182f30330afa.

Rotskoff, G. M. and Vanden-Eijnden, E. Trainability and Accuracy of Neural Networks: An Interacting Particle System Approach. *ArXiv preprint*, abs/1805.00915, 2018b. URL https://arxiv.org/abs/1805.00915.

Saad, D. and Solla, S. A. Exact Solution for On-Line Learning in Multilayer Neural Networks. *Physical Review Letters*, 74(21):4337–4340, 1995. ISSN 0031-9007, 1079-7114. doi: 10.1103/PhysRevLett.74.4337. URL https://link.aps.org/doi/10.1103/PhysRevLett.74.4337.

Saul, L. K., Jaakkola, T., and Jordan, M. I. Mean Field Theory for Sigmoid Belief Networks. *Journal of Artificial Intelligence Research*, 4:61–76, 1996. ISSN 1076-9757.

doi: 10.1613/jair.251. URL https://www.jair.org/index.php/jair/article/view/10156.

Saxe, A., Nelli, S., and Summerfield, C. If deep learning is the answer, what is the question? *Nature Reviews Neuroscience*, 22(1):55–67, 2021. ISSN 1471-0048. doi: 10.1038/s41583-020-00395-8. URL https://www.nature.com/articles/s41583-020-00395-8. Number: 1 Publisher: Nature Publishing Group.

Saxe, A. M., McClelland, J. L., and Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6120.

Saxe, A. M., Sodhani, S., and Lewallen, S. J. The neural race reduction: Dynamics of abstraction in gated networks. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 19287–19309. PMLR, 2022. URL https://proceedings.mlr.press/v162/saxe22a.html.

Seung, H. S., Sompolinsky, H., and Tishby, N. Statistical mechanics of learning from examples. *Physical Review A*, 45(8):6056–6091, 1992. ISSN 1050-2947, 1094-1622. doi: 10.1103/PhysRevA.45.6056. URL https://link.aps.org/doi/10.1103/PhysRevA.45.6056.

Soudry, D., Hoffer, E., Nacson, M. S., and Srebro, N. The implicit bias of gradient descent on separable data. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=r1q7n9gAb.

Wang, J., Yan, J. N., Gu, A., and Rush, A. M. Pretraining Without Attention, 2022. URL https://arxiv.org/abs/2212.10544.

Woodworth, B. E., Gunasekar, S., Lee, J. D., Moroshko, E., Savarese, P., Golan, I., Soudry, D., and Srebro, N. Kernel and rich regimes in overparametrized models. In *Conference on Learning Theory, COLT 2020, 9-12 July 2020, Virtual Event [Graz, Austria]*, volume 125 of *Proceedings of Machine Learning Research*, pp. 3635–3673. PMLR, 2020. URL http://proceedings.mlr.press/v125/woodworth20a.html.

Xiao, L., Bahri, Y., Sohl-Dickstein, J., Schoenholz, S. S., and Pennington, J. Dynamical isometry and a mean field theory of cnns: How to train 10, 000-layer vanilla convolutional neural networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5389–5398. PMLR, 2018. URL http://proceedings.mlr.press/v80/xiao18a.html.

Yamins, D. L. K. and DiCarlo, J. J. Using goal-driven deep learning models to understand sensory cortex. *Nature Neuroscience*, 19(3):356–365, 2016. ISSN 1546-1726. doi: 10.1038/nn.4244. URL https://www.nature.com/articles/nn.4244. Number: 3 Publisher: Nature Publishing Group.

Yamins, D. L. K., Hong, H., Cadieu, C. F., Solomon, E. A., Seibert, D., and DiCarlo, J. J. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, 111(23):8619–8624, 2014. doi: 10.1073/pnas.1403112111. URL https://www.pnas.org/doi/10.1073/pnas.1403112111. Publisher: Proceedings of the National Academy of Sciences.

Zhang, J., Wang, H., and Su, W. J. Imitating deep learning dynamics via locally elastic stochastic differential equations. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 6392–6403, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/327af0f71f7acdfd882774225f04775f-Abstract.html.

A. Details on the Theoretical Analysis

Here we provide computational details to the derivation of the effective theory, along with some additional results on the behavior the theory exhibits.

A.1. Reduction to a 3-dimensional System

We begin by modelling our neural network as being represented by a smooth encoder map $h : X \to H$ assigning representations to the inputs and a smooth decoder map $y : H \to Y$ assigning outputs to the representations. To model the two point interaction we restrict to a two datapoint dataset $\mathcal{D} = \{(x_1, y_1), (x_2, y_2)\}$ and take a linear approximation around their representational and output mean:

$$h(x_i) = h(\frac{x_2 + x_1}{2}) + \frac{1}{2}D_h(x_i - x_{\neg i})$$

$$
\begin{aligned}
y(x_i) &= y(h(\frac{x_2 + x_1}{2})) + \frac{1}{2}D_y(h(x_i) - h(x_{\neg i})) \\
&= y(h(\frac{x_2 + x_1}{2})) + \frac{1}{2}D_y(D_h(x_i - x_{\neg i})).
\end{aligned}
\tag{15}
$$

The mean squared loss in this approximation has the form:

$$L = \frac{1}{2}\langle||y(h(\frac{x_2 + x_1}{2})) + \frac{1}{2}D_y(D_h(x_i - x_{\neg i})) - y_i||^2\rangle_{\mathcal{D}}. \tag{16}$$

Motivated by the assumption of high model expressivity, we apply gradient decent optimization directly with respect to D_h, D_y and $\bar{y} := y(h(\frac{x_2+x_1}{2}))$, resulting in the dynamics:

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\bar{y} &= -\frac{1}{\tau_{\bar{y}}}\frac{\partial L}{\partial \bar{y}} \\
&= -\frac{1}{\tau_{\bar{y}}}\langle\bar{y} + \frac{1}{2}D_y(D_h(x_i - x_{\neg i})) - y_i\rangle_{\mathcal{D}} \\
&= -\frac{1}{\tau_{\bar{y}}}(\bar{y} - \frac{y_2 + y_1}{2})
\end{aligned}
$$

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}D_h &= -\frac{1}{\tau_h}\frac{\partial L}{\partial D_h} \\
&= -\frac{1}{\tau_h}\frac{1}{2}\langle(\frac{1}{2}D_y^\top D_y D_h(x_i - x_{\neg i})(x_i - x_{\neg i})^\top + D_y^\top(\bar{y} - y_i)(x_i - x_{\neg i})^\top)\rangle_{\mathcal{D}} \\
&= -\frac{1}{\tau_h}\frac{1}{2}D_y^\top(D_y D_h(x_2 - x_1) - (y_2 - y_1))(x_2 - x_1)^\top
\end{aligned}
\tag{17}
$$

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}D_y &= -\frac{1}{\tau_y}\frac{\partial L}{\partial D_y} \\
&= -\frac{1}{\tau_y}\frac{1}{2}\langle(\frac{1}{2}D_y D_h(x_i - x_{\neg i})(D_h(x_i - x_{\neg i}))^\top + (\bar{y} - y_i)(D_h(x_i - x_{\neg i}))^\top)\rangle_{\mathcal{D}} \\
&= -\frac{1}{\tau_y}\frac{1}{2}(D_y D_h(x_2 - x_1) - (y_2 - y_1))(D_h(x_2 - x_1))^\top,
\end{aligned}
$$

where we used the matrix differentiation identities $\frac{\partial a^\top X b}{\partial X} = ab^\top$, $\frac{\partial a^\top X^\top C X a}{\partial X} = (C + C^\top)Xaa^\top$.

The $\bar{y}$ dynamics are decoupled and can be solved directly:

$$\bar{y}(t) = \frac{y_2 + y_1}{2} + (y(0) - \frac{y_2 + y_1}{2})e^{-\frac{t}{\tau_{\bar{y}}}}, \tag{18}$$

the solution of which takes the form of exponential decay towards the target output mean.

Define $dh := D_h(x_2 - x_1), dy := D_y D_h(x_2 - x_1), w := ||dy||^2 - dy^\top(y_2 - y_1)$. To simplify the dynamics, we restrict to solutions where the representations only move towards each other or away from each other, i.e.

$$\frac{\mathrm{d}}{\mathrm{d}t} dh \propto dh. \tag{19}$$

We find that the derivatives

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} ||dh||^2 &= 2dh^\top \frac{\mathrm{d}}{\mathrm{d}t} dh \\
&= 2dh^\top \dot{D}_h(x_2 - x_1) \\
&= -\frac{1}{\tau_h}(||dy||^2 - dy^\top(y_2 - y_1))||x_2 - x_1||^2 \\
&= -\frac{1}{\tau_h}||x_2 - x_1||^2 w
\end{aligned}
$$

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} ||dy||^2 &= 2dy^\top \frac{\mathrm{d}}{\mathrm{d}t} dy \\
&= 2dy^\top(\dot{D}_y D_h(x_2 - x_1) + D_y \frac{\mathrm{d}}{\mathrm{d}t} dh) \\
&= 2dy^\top(\dot{D}_y D_h(x_2 - x_1) + \frac{dh^\top \frac{\mathrm{d}}{\mathrm{d}t} dh}{||dh||^2} D_y dh) \\
&= 2dy^\top(\dot{D}_y D_h(x_2 - x_1) + \frac{1}{2}\frac{\frac{\mathrm{d}}{\mathrm{d}t}||dh||^2}{||dh||^2} D_y dh) \\
&= 2dy^\top(-\frac{1}{\tau_y}\frac{1}{2}(dy - (y_2 - y_1))(dh)^\top dh - \frac{1}{\tau_h}\frac{1}{2}||x_2 - x_1||^2 \frac{w}{||dh||^2} dy) \\
&= -w(\frac{1}{\tau_y}||dh||^2 + \frac{1}{\tau_h}||x_2 - x_1||^2 \frac{||dy||^2}{||dh||^2})
\end{aligned}
\tag{20}
$$

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} w &= (2dy - (y_2 - y_1))^\top \frac{\mathrm{d}}{\mathrm{d}t} dy \\
&= (2dy - (y_2 - y_1))^\top(-\frac{1}{\tau_y}\frac{1}{2}(dy - (y_2 - y_1))||dh||^2 - \frac{1}{\tau_h}\frac{1}{2}||x_2 - x_1||^2 \frac{w}{||dh||^2} dy) \\
&= -\frac{1}{\tau_y}\frac{1}{2}(3w - ||dy||^2 + ||y_2 - y_1||^2)||dh||^2 - \frac{1}{\tau_h}\frac{1}{2}||x_2 - x_1||^2 \frac{(||dy||^2 + w)w}{||dh||^2},
\end{aligned}
$$

form a self-contained 3 dimensional system:

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} ||dh||^2 &= -\frac{1}{\tau_h}||x_2 - x_1||^2 w \\
\frac{\mathrm{d}}{\mathrm{d}t} ||dy||^2 &= -w(\frac{1}{\tau_y}||dh||^2 + \frac{1}{\tau_h}||x_2 - x_1||^2 \frac{||dy||^2}{||dh||^2}) \\
\frac{\mathrm{d}}{\mathrm{d}t} w &= -\frac{1}{\tau_y}\frac{1}{2}(3w - ||dy||^2 + ||y_2 - y_1||^2)||dh||^2 - \frac{1}{\tau_h}\frac{1}{2}||x_2 - x_1||^2 \frac{(||dy||^2 + w)w}{||dh||^2}.
\end{aligned}
\tag{21}
$$

Because this system is self-contained and consists of only 3 scalar variables, it can easily be solved numerically given initial conditions and values for the effective learning rates $1/\tau_h$ and $1/\tau_y$.

A.2. Expression for the Loss Curve

The loss can be expressed in terms of variables in the 3-dimensional system:

$$
\begin{aligned}
L &= \frac{1}{2}\langle ||\bar{y} + \frac{1}{2}D_y D_h(x_i - x_{\neg i}) - y_i||^2\rangle_{\mathcal{D}} \\
&= \frac{1}{2}\langle ||\bar{y} - y_i||^2 + (D_y D_h(x_i - x_{\neg i}))^\top(\bar{y} - y_i) + \frac{1}{4}||dy||^2\rangle_{\mathcal{D}} \\
&= \frac{1}{4}(||\bar{y} - y_1||^2 + ||\bar{y} - y_2||^2 - (dy)^\top(y_2 - y_1) + \frac{1}{2}||dy||^2) \\
&= \frac{1}{4}(||\bar{y} - y_1||^2 + ||\bar{y} - y_2||^2 + w - \frac{1}{2}||dy||^2) \\
&= \frac{1}{4}(||(y(0) - \frac{y_2 + y_1}{2})e^{-\frac{t}{\tau_{\bar{y}}}} + \frac{y_2 - y_1}{2}||^2 + ||(y(0) - \frac{y_2 + y_1}{2})e^{-\frac{t}{\tau_{\bar{y}}}} - \frac{y_2 - y_1}{2}||^2 + w - \frac{1}{2}||dy||^2) \\
&= \frac{1}{4}(2||(y(0) - \frac{y_2 + y_1}{2})e^{-\frac{t}{\tau_{\bar{y}}}}||^2 + w + \frac{1}{2}(||y_2 - y_1||^2 - ||dy||^2)).
\end{aligned}
\tag{22}
$$

A.3. Final Representational Distance

The system in Equation (21) cannot be solved analytically in the general case, however, we can still gain some analytical insights. For instance, we can determine the final representational distance, which may provide insight into the kinds of representational structures formed. To find this, we exploit the following relationship in the dynamics:

$$
\frac{\mathrm{d}}{\mathrm{d}t}\frac{||dy||^2}{||dh||^2} = \frac{||dh||^2\frac{\mathrm{d}}{\mathrm{d}t}||dy||^2 - ||dy||^2\frac{\mathrm{d}}{\mathrm{d}t}||dh||^2}{||dh||^4} = -\frac{1}{\tau_y}w = \frac{\tau_h}{\tau_y}\frac{1}{||x_2 - x_1||^2}\frac{\mathrm{d}}{\mathrm{d}t}||dh||^2.
\tag{23}
$$

Integrating both sides, we can write $||dy||^2$ as a function of $||dh||^2$

$$
||dy||^2 = \frac{\tau_h}{\tau_y}\frac{1}{||x_2 - x_1||^2}||dh||^4 + \left(\frac{||dy(0)||^2}{||dh(0)||^2} - \frac{\tau_h}{\tau_y}\frac{||dh(0)||^2}{||x_2 - x_1||^2}\right)||dh||^2,
\tag{24}
$$

reducing the dynamics to a 2-dimensional system:

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}||dh||^2 &= -\frac{1}{\tau_h}||x_2 - x_1||^2 w \\
\frac{\mathrm{d}}{\mathrm{d}t}w &= -\frac{1}{2}(-\frac{\tau_h}{\tau_y{}^2}\frac{1}{||x_2 - x_1||^2}||dh||^6 + \frac{1}{\tau_y}||y_2 - y_1||^2||dh||^2 + \frac{4}{\tau_y}||dh||^2 w + \frac{1}{\tau_h}||x_2 - x_1||^2\frac{w^2}{||dh||^2} \\
&\quad + \left(\frac{||dy(0)||^2}{||dh(0)||^2} - \frac{\tau_h}{\tau_y}\frac{||dh(0)||^2}{||x_2 - x_1||^2}\right)(\frac{1}{\tau_h}||x_2 - x_1||^2 w - \frac{1}{\tau_y}||dh||^4)).
\end{aligned}
\tag{25}
$$

This system explicitly contains the initial conditions, making it less intuitive to work with. However, it is useful for finding fixed points, as any fixed point of this system must also be one in the original system. To determine what the fixed points are, we set the derivatives to zero.

From the $||dh||^2$ dynamics we immediately find that w is zero at the fixed points:

$$
\frac{\mathrm{d}}{\mathrm{d}t}||dh||^2 = 0 \implies w = 0,
\tag{26}
$$

and from the w dynamics we find three potential values for $||dh||^2$:

$$
\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}t}w = 0 \\
&\implies 0 = -\frac{\tau_h}{\tau_y}\frac{1}{||x_2 - x_1||^2}||dh||^6 - \left(\frac{||dy(0)||^2}{||dh(0)||^2} - \frac{\tau_h}{\tau_y}\frac{||dh(0)||^2}{||x_2 - x_1||^2}\right)||dh||^4 + ||y_2 - y_1||^2||dh||^2 \\
&\implies ||dh||^2 = 0 \text{ or } ||dh||^2 = \frac{1}{2}A_{\text{high}} \pm \sqrt{\frac{1}{4}A_{\text{high}}^2 + A_{\text{low}}^2},
\end{aligned}
\tag{27}
$$

where

$$A_{\text{high}} = \left(\frac{||dh(0)||^2}{||x_2 - x_1||^2} - \frac{\tau_y}{\tau_h} \frac{||dy(0)||^2}{||dh(0)||^2} \right) ||x_2 - x_1||^2$$

$$A_{\text{low}} = \sqrt{\frac{\tau_y}{\tau_h}} ||x_2 - x_1|| \cdot ||y_2 - y_1||. \tag{28}$$

The fixed point at $||dh||^2 = \frac{1}{2}A_{\text{high}} - \sqrt{\frac{1}{4}A_{\text{high}}^2 + A_{\text{low}}^2}$ is negative and is thus not a valid solution. The dynamics will never be able to reach it as $||dh||^2$ cannot go below zero.

The fixed point at $||dh||^2 = \frac{1}{2}A_{\text{high}} + \sqrt{\frac{1}{4}A_{\text{high}}^2 + A_{\text{low}}^2}$ has Jacobian

$$J = \begin{bmatrix} 0 & -\frac{1}{\tau_h}||x_2 - x_1||^2 \\ \frac{1}{4}\frac{\tau_h}{\tau_y^2}\frac{1}{||x_2 - x_1||^2}(2A_{\text{low}}^2 + \frac{1}{2}(A_{\text{high}} + \sqrt{A_{\text{high}}^2 + 4A_{\text{low}}^2})^2) & -\frac{1}{\tau_y}(\frac{1}{2}A_{\text{high}} + \sqrt{A_{\text{high}}^2 + 4A_{\text{low}}^2}) \end{bmatrix}, \tag{29}$$

The trace

$$\tau = -\frac{1}{\tau_y}(\frac{1}{2}A_{\text{high}} + \sqrt{A_{\text{high}}^2 + 4A_{\text{low}}^2}) \tag{30}$$

is negative and the determinant

$$D = \frac{1}{4}\frac{1}{\tau_y^2}(2A_{\text{low}}^2 + \frac{1}{2}(A_{\text{high}} + \sqrt{A_{\text{high}}^2 + 4A_{\text{low}}^2})^2) \tag{31}$$

is positive. This fixed point is therefore always stable.

The fixed point at $||dh||^2 = 0$ has Jacobian

$$J = \begin{bmatrix} 0 & -\frac{1}{\tau_h}||x_2 - x_1||^2 \\ \frac{1}{2}(\frac{1}{\tau_h}||x_2 - x_1||^2\frac{w^2}{||dh||^4} - \frac{1}{\tau_y}||y_2 - y_1||^2) & (\frac{1}{2}\frac{1}{\tau_y}A_{\text{high}} - \frac{1}{\tau_h}||x_2 - x_1||^2\frac{w}{||dh||^2}) \end{bmatrix}. \tag{32}$$

This Jacobian cannot be directly evaluated at $w = 0, ||dh||^2 = 0$ because of the undetermined term $\frac{w}{||dh||^2}$. By replacing $\frac{w}{||dh||^2}$ with the direction of approach $\frac{b}{a}$ we can solve for eigenvectors

$$\begin{bmatrix} 0 & -\frac{1}{\tau_h}||x_2 - x_1||^2 \\ \frac{1}{2}(\frac{1}{\tau_h}||x_2 - x_1||^2\frac{b^2}{a^2} - \frac{1}{\tau_y}||y_2 - y_1||^2) & (\frac{1}{2}\frac{1}{\tau_y}A_{\text{high}} - \frac{1}{\tau_h}||x_2 - x_1||^2\frac{b}{a}) \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \lambda \begin{bmatrix} a \\ b \end{bmatrix} \tag{33}$$

to find

$$v_{\pm} = \begin{bmatrix} 1 \\ -\frac{\tau_h}{\tau_y}\frac{A_{\text{high}} \pm \sqrt{A_{\text{high}}^2 + 4A_{\text{low}}^2}}{2||x_2 - x_1||^2} \end{bmatrix} \tag{34}$$

with one positive and one negative eigenvalue

$$\lambda_{\pm} = \frac{1}{\tau_y}\frac{A_{\text{high}} \pm \sqrt{A_{\text{high}}^2 + 4A_{\text{low}}^2}}{2}. \tag{35}$$

There is always one direction along which perturbations increase, so this fixed point is not stable. In the limit of infinite time we expect the final representational distance to go to the only stable fixed point:

$$||dh(\infty)||^2 = \frac{1}{2}A_{\text{high}} + \sqrt{\frac{1}{4}A_{\text{high}}^2 + A_{\text{low}}^2}. \tag{36}$$

A.4. Solution for Identical Outputs

In the case that the target outputs of the two datapoints are identical, i.e. $y_1 = y_2$, an exact solution can be found. In this case we have by definition $w = ||dy||^2$, reducing the system to:

$$\frac{\mathrm{d}}{\mathrm{d}t}||dh||^2 = -\frac{1}{\tau_h}||x_2 - x_1||^2||dy||^2$$

$$\frac{\mathrm{d}}{\mathrm{d}t}||dy||^2 = -||dy||^2(\frac{1}{\tau_y}||dh||^2 + \frac{1}{\tau_h}||x_2 - x_1||^2\frac{||dy||^2}{||dh||^2})$$

(37)

Using Equation (24) we can write self-contained $||dh||^2$ dynamics:

$$\frac{\mathrm{d}}{\mathrm{d}t}||dh||^2 = -\frac{1}{\tau_y}||dh||^4 + \frac{1}{\tau_y}A_{\text{high}}||dh||^2,$$

(38)

which is a Bernoulli equation and has solution

$$||dh(t)||^2 = \frac{A_{\text{high}}}{1 + (\frac{A_{\text{high}}}{||dh(0)||^2} - 1)e^{-A_{\text{high}}t}}.$$

(39)

Plugging this in Equation (24) gives us the $||dy||^2$ solution:

$$||dy||^2 = \frac{\tau_h}{\tau_y}\frac{A_{\text{high}}^2}{||x_2 - x_1||^2}\frac{(1 - \frac{A_{\text{high}}}{||dh(0)||^2})e^{-A_{\text{high}}t}}{(1 + (\frac{A_{\text{high}}}{||dh(0)||^2} - 1)e^{-A_{\text{high}}t})^2}.$$

(40)

A.5. Lazy Learning Dynamics

At large weights $w \approx ||dy||^2$, so increasing the weights scales the initial values as

$$||dh||^2 \rightarrow G||dh||^2$$
$$||dy||^2 \rightarrow G^2||dy||^2.$$
$$w \rightarrow G^2w$$

(41)

Under the variable change

$$||dh||^{2'} = G||dh||^2$$
$$||dy||^{2'} = G^2||dy||^2,$$
$$w' = G^2w$$
$$t' = G^{-1}t$$

(42)

the dynamics change to

$$\frac{\mathrm{d}}{\mathrm{d}t'}||dh||^{2'} = -\frac{1}{\tau_h}||x_2 - x_1||^2w'$$

$$\frac{\mathrm{d}}{\mathrm{d}t'}||dy||^{2'} = -w'(\frac{1}{\tau_y}||dh||^{2'} + \frac{1}{\tau_h}||x_2 - x_1||^2\frac{||dy||^{2'}}{||dh||^{2'}})$$

$$\frac{\mathrm{d}}{\mathrm{d}t'}w' = -\frac{1}{\tau_y}\frac{1}{2}(3w' - ||dy||^{2'} + G^{-2}||y_2 - y_1||^2)||dh||^{2'} - \frac{1}{\tau_h}\frac{1}{2}||x_2 - x_1||^2\frac{(||dy||^{2'} + w')w'}{||dh||^{2'}}.$$

(43)

Taking the large initial weight limit $G \rightarrow \infty$, these dynamics reduce to the dynamics for identical outputs. We can thus expect a similar solution for large initial weights, as the one derived in Appendix A.4, i.e. smooth exponential decay. The training loss will therefore also exponentially decay to zero without a plateau at large initial weights, which is similar what is observed in different neural networks.

A.6. Random Feature Learning

A characteristic of the lazy learning regime is that it learns random features, as opposed to the more consistent structured features learned at small initialization. We can explicitly demonstrate the vanishing of randomness in the representation at small initial weights, by assuming the initial weights are Gaussian distributed, such that $\frac{||dh(0)||^2}{||x_i - x_{\neg i}||^2} \sim \mathcal{N}(G, G^2)$, $\frac{||dy(0)||^2}{||dh(0)||^2} \sim \mathcal{N}(G, G^2)$, where G is some overall gain factor at initialization. The exact scaling of these distributions with the initial gain is not important here, this is merely done for illustrative purposes. We can write

$$
\begin{aligned}
P(||dh(\infty)||^2 < h) &= P(\frac{1}{2}A_{\text{high}} + \sqrt{\frac{1}{4}A_{\text{high}}^2 + A_{\text{low}}^2} < h) \\
&= P(\sqrt{A_{\text{high}}^2 + 4A_{\text{low}}^2} < 2h - A_{\text{high}}) \\
&= P(A_{\text{high}}^2 + 4A_{\text{low}}^2 < (2h - A_{\text{high}})^2 \text{ and } 2h - A_{\text{high}} > 0) \\
&= P(4A_{\text{low}}^2 < 4h^2 - 4hA_{\text{high}} \text{ and } 2h - A_{\text{high}} > 0) \\
&= P(A_{\text{high}}h < h^2 - A_{\text{low}}^2 \text{ and } A_{\text{high}} < 2h)
\end{aligned}
\tag{44}
$$

for positive h. Since $-A_{\text{low}}^2 < h^2$, we have $h^2 - A_{\text{low}}^2 < 2h^2$, so in the event that $A_{\text{high}}h < h^2 - A_{\text{low}}^2$, the equality $A_{\text{high}} < 2h$ is automatically satisfied. Thus,

$$
P(||dh(\infty)||^2 < h) = P(A_{\text{high}}h < h^2 - A_{\text{low}}^2).
\tag{45}
$$

The distribution on A_{high} is Gaussian:

$$
\begin{aligned}
A_{\text{high}} &= ||x_2 - x_1||^2 (\frac{||dh(0)||^2}{||x_2 - x_1||^2} - \frac{||dy(0)||^2}{||dh(0)||^2}) \\
&\sim \mathcal{N}(||x_2 - x_1||^2 G - ||x_2 - x_1||^2 G, ||x_2 - x_1||^4 G^2 + ||x_2 - x_1||^4 G^2) \\
&= \mathcal{N}(0, 2||x_2 - x_1||^4 G^2)
\end{aligned}
\tag{46}
$$

such that

$$
P(||dh(\infty)||^2 < h) = \Phi(\frac{h - \frac{A_{\text{low}}^2}{h}}{\sqrt{2}||x_2 - x_1||^2 G}),
\tag{47}
$$

giving us a final representational distribution:

$$
\begin{aligned}
f(h) &= \frac{\mathrm{d}}{\mathrm{d}h} \Phi(\frac{h - \frac{A_{\text{low}}^2}{h}}{\sqrt{2}||x_2 - x_1||^2 G}) \\
&= \frac{1 + \frac{A_{\text{low}}^2}{h^2}}{\sqrt{2}||x_2 - x_1||^2 G} \phi(\frac{h - \frac{A_{\text{low}}^2}{h}}{\sqrt{2}||x_2 - x_1||^2 G}), \\
&= \frac{1 + \frac{A_{\text{low}}^2}{h^2}}{2\sqrt{\pi}||x_2 - x_1||^2 G} e^{-\frac{1}{2}(\frac{h - \frac{A_{\text{low}}^2}{h}}{\sqrt{2}||x_2 - x_1||^2 G})^2}
\end{aligned}
\tag{48}
$$

We can see that as we take the limit of small initial weights $G \to 0$, the distribution converges, removing any randomness in the final structure.

B. Experimental Details

Here we provide additional details and methods used in the experiments. For all experiments we used the open-source library PyTorch. We chose stochastic gradient descent as an optimizer, as it was used for the theory derivation. The hidden layer H considered for all models is always the layer exactly halfway through the network.

B.1. Two Point Experiments

Dataset. The dataset used here is composed of two datapoints with 1-dimensional inputs and outputs. The inputs are -1 and $-1 + dx$, the outputs are 0.6 and $0.6 + dy$ respectively, where dx has been set to 0.5 and dy set to 1.

Table 1. Hyperparameters of the different architectures used in 2 point experiments.

EXPERIMENT	NONLINEARITY	INIT GAIN	LEARNING RATE	EPOCHS	UNITS	HIDDEN LAYERS
DEFAULT	LEAKY RELU	1.0	0.005	6000	500	20
CONVOLUTION	LEAKY RELU	0.95	0.002	6000	500	10
SKIP CONNECTIONS	LEAKY RELU	0.1	0.01	6000	500	20
SHALLOW	LEAKY RELU	0.35	0.01	6000	500	2
NARROW	LEAKY RELU	1.05	0.005	6000	50	20
DROPOUT	LEAKY RELU	1.0	0.005	6000	500	10
LINEAR	NONE	0.85	0.00125	6000	500	20
RELU	RELU	1.0	0.005	6000	500	20
ELU	ELU	0.95	0.00045	6000	500	20
TANH	TANH	0.85	0.00125	6000	500	20
SWISH	SWISH	1.7	0.0025	6000	500	20
LOW INITIAL WEIGHTS	LEAKY RELU	0.9	0.015	6000	500	20
INTERMEDIATE INITIAL WEIGHTS	LEAKY RELU	1.1	0.0025	6000	500	20
HIGH INITIAL WEIGHTS	LEAKY RELU	1.3	0.00075	6000	500	20
VERY HIGH INITIAL WEIGHTS	LEAKY RELU	1.4	0.00025	6000	500	20

Architectures. The architectures used are all variants of deep neural networks where each layer has the same hidden dimension, all layers fully connected unless specified otherwise. All models are initialized using the Xavier normal initialization with gain parameter chosen to display rich learning behavior. Each layer has biases and these are initialized at zero. Learning rates are chosen to produce smooth loss curves while still converging within the 6000 epochs. The different hyperparameters can be found in Table 1. For the convolution architecture the connections between hidden layers are replaced with convolution layers, each with 20 channels and kernel size 11. The skip connection architecture has a skip connection on each layer jumping over the next two layers. For the dropout network a dropout probability of 0.1 was used on all units.

Fitting Procedure. To determine the values of $1/\tau_h$ and $1/\tau_y$ we optimize the squared fit loss:

$$\int_{\text{epochs}} (||dh(t)||^2 - ||dh(t)||^2_{\text{theory}})^2 dt + \int_{\text{epochs}} (||dy(t)||^2 - ||dy(t)||^2_{\text{theory}})^2 dt + \int_{\text{epochs}} (w(t) - w(t)_{\text{theory}})^2 dt, \quad (49)$$

where $||dh(t)||^2_{\text{theory}}$, $||dy(t)||^2_{\text{theory}}$ and $w(t)_{\text{theory}}$ are the numerical solutions to Equation (7) for the choice of $1/\tau_h$ and $1/\tau_y$. Additionally, for the loss curves, the parameter $1/\tau_{\bar{y}}$ was fit in the same manner.

Final Representational Distance. For the comparison of the final representational distance to the theory the same hyperparameter settings were used. First, for each architecture, effective learning rates were fit and averaged over 50 trials at fixed input and output distances $dx = 0.5, dy = 1$. Next, we ran 5*5 trials for each architecture with varying input distance dx between 0.5 and 1.5, and varying output distance dy between 0 and 1, comparing them to the theory prediction from the averaged effective learning rates. Trials which did not converge were discarded.

B.2. XOR Task

Dataset. The dataset used has 4 datapoints, 2 input dimensions and 1 output dimension. Explicitly, the 4 datapoints are $\{x : (0,0), y : 0\}$, $\{x : (1,0), y : 1\}$, $\{x : (0,1), y : 1\}$ and $\{x : (1,1), y : 0\}$.

Architecture. The architecture used is the same as the default architecture in the two point experiments. The hyperparameters used can be found in Table 2 (top). For the theory prediction, the effective learning rates from the two point experiments fitted on the default architecture were used, as it also has 20 layers, 500 units and leaky ReLU.

B.3. Feature Collapse Experiment

Dataset. The dataset is similar to one used in (Flesch et al., 2022), but with an increased number datapoints, as this is required to preserve the topology of the representation due to the added depth in the architecture. The 2*30*30 datapoints

Table 2. Hyperparameters used in the experiments on the different toy datasets.

EXPERIMENT	NONLINEARITY	INIT GAIN	LEARNING RATE	EPOCHS	UNITS	HIDDEN LAYERS
XOR Low weights	LEAKY RELU	0.7	0.015	1000	500	20
XOR Intermediate weights	LEAKY RELU	1.25	0.002	1000	500	20
XOR High weights	LEAKY RELU	2.0	0.000000003	1000	500	20
XOR 2 layers	LEAKY RELU	0.0015	0.03	40000	500	1
Blobs Very low weights	TANH	0.275	0.05	2000	100	4
Blobs Low weights	TANH	0.5	0.015	1500	100	4
Blobs High weights	TANH	2.0	0.0015	750	100	4
MINST Trajectories	LEAKY RELU	0.5	0.003	200	200	4
MINST Structure	LEAKY RELU	0.5	0.005	200	200	4

are 5*5 pixel images of Gaussian blobs, with varying x and y position of the mean. The Gaussian variance is kept constant. Two additional input dimensions are added for a one-hot encoding of the context variable, resulting in 27 dimensional input data. The output data has a single dimension and is identical the x position for the half of the dataset in the first context, and the y position for the second half.

Architecture. The architecture used is a 5 layer fully connected deep neural network with tanh nonlinearity. Hyperparameters can be found in Table 2 (middle).

B.4. Experiment on MNIST

Dataset. The full training set of MNIST handwritten digit classification was used during training. The representational structure was plotted for the first 100 digits, sorted by digit. For the learning dynamics, a zero digit and a one digit were selected at random. The digits zero and one were chosen since they look fairly distinct, compared to e.g. three and eight. Trajectories look slightly worse for more similar digits, but still roughly share a similar shape to the effective theory.

Architecture. The architecture used is a fully connected, leaky ReLU network, with only 4 hidden layers instead of the usual 20 to allow for faster convergence in the feature learning regime. The hyperparameters used for both the learning dynamics and representational structure experiments can be found in Table 2 (bottom). Initial weight parameters were chosen to display rich learning behaviour, while still converging within reasonable time.

Representational Structure For comparing the final representational structure to the theory prediction, the model was trained 50 times at varying random seeds. The resulting representational distances between the first 100 digits in the dataset were averaged. The Pearson correlation coefficient is computed between the vector of all pairwise final representational distances and their respective theory predictions from Equation (10).

Exponential Weighing Applying Equation (10) directly makes little sense, as a prediction for pairwise representational distances does not necessarily directly translate into a prediction for the full dataset representation. Even when assuming pairwise distances in the full dataset behave independently, the two datapoint prediction may end up breaking the triangle inequality, i.e. it may predict that representations of datapoints A and B are close, as well as B and C, but still predict A and C to be far away. Thus, we must apply a preprocessing step to take the geometry into account.

One somewhat arbitrary choice of still making predictions is by using exponential weighing, i.e. adjust the representational distance predictions based on nearby datapoints

$$||dh_{i,j}(\infty)||^2_{\text{pred}} = \sum_{k,l} e^{-||dh_{i,k}(\infty)||^2_{\text{theory}}} e^{-||dh_{l,j}(\infty)||^2_{\text{theory}}} ||dh_{k,l}(\infty)||^2_{\text{theory}}. \tag{50}$$

In this case the large distance between A and C would cause the predicted distance between A and B, and B and C to be large too. Thus, the method reduces issues with the triangle inequality. Although simplistic and without theoretical guarantees, the method is easy to vectorize and can thus be used on a large part of the dataset.

C. Supplemental Figures

C.1. Alternative Dynamics

When comparing the effective theory's dynamics to that of the neural networks, the two effective learning rates need to be fit. Given the simplicity of the learning dynamics, one may wonder if fitting these two parameters will lead to spurious conclusions about the accuracy of the effective theory as a model, since a system with enough freedom may be able to fit any simple dynamics.

To demonstrate that the specific form of Equation (7) is important to be able to fit the dynamics well, consider four slightly altered versions of this system, where we added a factor two, a square, removed a minus sign, and remove a term respectively.

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}||dh||^2 &= -\frac{1}{\tau_h}||x_2 - x_1||^2 w^{\mathbf{2}} \\
\frac{\mathrm{d}}{\mathrm{d}t}||dy||^2 &= -\mathbf{2}w(\frac{1}{\tau_y}||dh||^2 + \frac{1}{\tau_{\mathbf{h}}}||\mathbf{x_2} - \mathbf{x_1}||^{\mathbf{2}}\frac{||\mathbf{dy}||^{\mathbf{2}}}{||\mathbf{dh}||^{\mathbf{2}}}) \\
\frac{\mathrm{d}}{\mathrm{d}t}w &= -\frac{1}{\tau_y}\frac{1}{2}(3w - ||dy||^2 + ||y_2 - y_1||^2)||dh||^2 \mathbf{+} \frac{1}{\tau_h}\frac{1}{2}||x_2 - x_1||^2\frac{(||dy||^2 + w)w}{||dh||^2},
\end{aligned}
\tag{51}
$$

When fitting these altered dynamics to the default architecture using the same procedure as before, we find that the squared fit loss (Equation (49)) increases from 0.92 to 15.35, 16.61, 2.99 and 5.64 respectively.
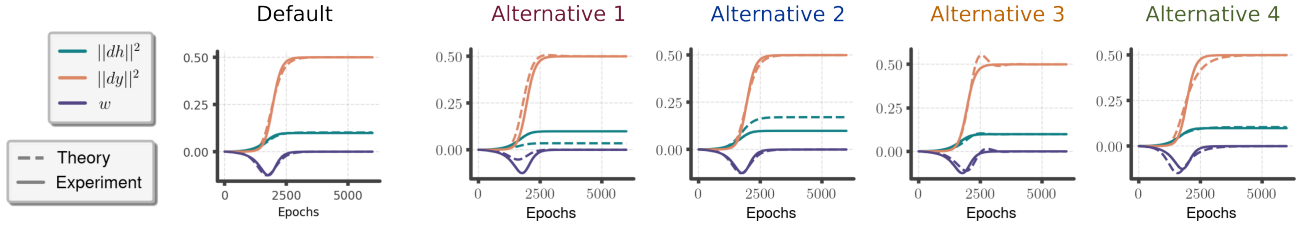

Figure 11. Comparisons of the effective theory (Equation (7)) and four alternative dynamics (Equation (51)) to the training dynamics of the default architecture (20 fully connected layers, 500 units per layer, leaky ReLU) on two datapoints. Trajectories of the alternatives are not as good of a fit compared to the effective theory.

C.2. Loss Curves for Varying Architectures

Just as with the dynamics of the 3-dimensional system, we can fit the loss expression from the theory Equation (9) to the training loss, amongst varying architectures, as is shown in Figure 12. Using the already fitted dynamics of the 3-dimensional system, this requires the fitting of one more unknown parameter, namely $1/\tau_{\bar{y}}$.

C.3. Very Low Weight Limit in the Orthogonal Feature Task

One could argue that for the orthogonal feature task at low initial weights the representation should collapse to a single line, not two disjoint lines. This is because for every point in each context there is a respective point with the same output in the other context, meaning their output difference is zero. Therefore, by Equation (10) their representational distance should also end up at zero. Yet, we still find two separate lines in the representation. This is because the difference in inputs here is much larger than for points along the irrelevant feature, so their representational distance will take longer to go to zero in the small weight limit. Indeed, if we initialize even smaller, we find that the representation collapses to a single line, as can be seen in Figure 13.

C.4. Validation of the Linear Approximation

One may wonder to what extend the linear approximation taken in Equation (5) actually holds during training. For the effective theory to accurately model interactions, we would like the first order term to dominate for pairs of nearby datapoints.
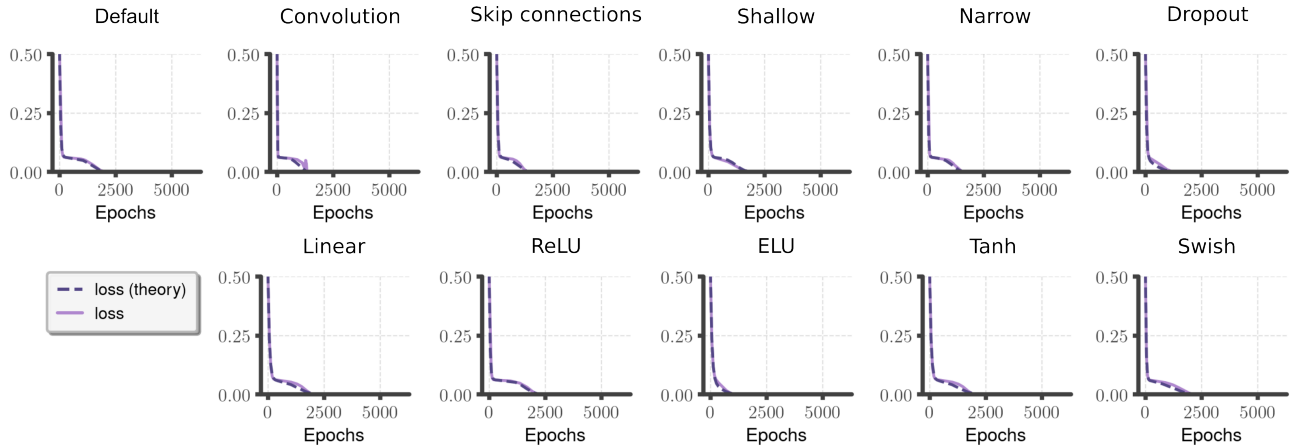
Figure 12. Training loss amongst different architectures. Training loss on the two point experiment compared to the prediction from the effective theory. The first two effective learning rates fitted in the 3-dimensional system were re-used and the final one was fitted using the same procedure.
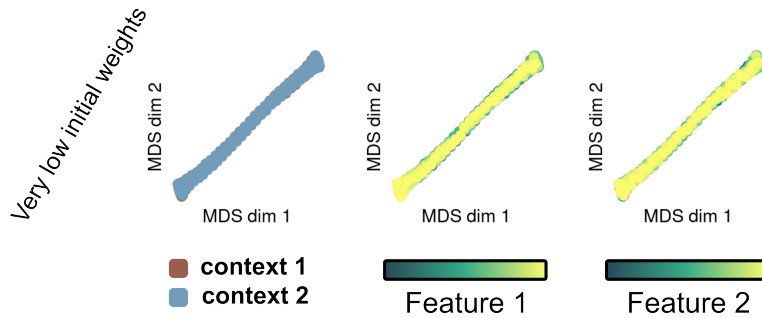

Figure 13. Representational structure in the orthogonal feature task at very low initial weights. A multidimensional scaling projection of the representational structure of a 4-layer, tanh, fully connected network with 100 hidden units per layer, trained on the orthogonal feature dataset. At very small initial weights the representational structure collapses to a single line.

One way of validating the linear approximation is to compare the first order Taylor term to higher terms. We see in Figure 14 that it holds up reasonably well for close enough pairs of datapoints.

C.5. Adam Optimizer

The dynamics in Equation (7) were derived on the assumption that the neural network is trained using gradient descent. In practice, other optimization schemes are used to train networks, such as Adam. It is a priori not clear that the derived results translate to different optimizers. Indeed, as can be seen in Figure 15 the effective theory does not fit the dynamics of a neural network trained using Adam very well.

However, the intuitions discussed in Section 4.3 do not explicitly depend on some details of the optimization scheme, such as changes to the learning rate during training. Therefore, we may still expect a similar effect on the final representational structure. As can be seen in Figure 16, Figure 17 and Figure 18 the representational predictions still hold up well.
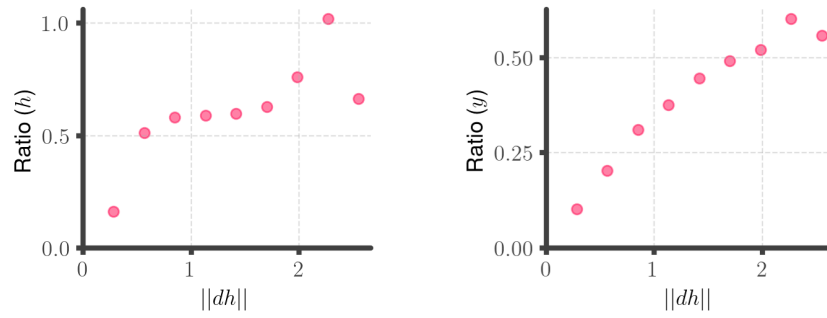
Figure 14. The ratio between the first order term and the second order term in the Taylor expansions of h (**left**) and y (**right**) respectively for pairs of datapoints in MNIST during training. We randomly selected 1000 pairs of datapoints, and then grouped them into 10 bins based on their representational distance. These results are shown after training the MNIST architecture from Table 2 for 30 epochs. We can see a trend for both maps where the linear term dominates for nearby datapoints but gets worse the further they are away. This trend is not present at initialization but does appear very quickly, becoming visible even at a single epoch.
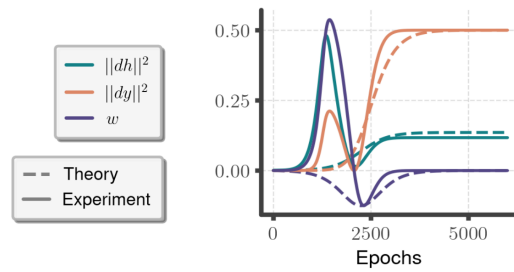

Figure 15. Learning dynamics of the default architecture (20 layers, 500 units, leaky ReLU) trained on two data points using with Adam, compared against the theory after fitting two constants. The fit is not very good, especially in the earlier part of the dynamics.
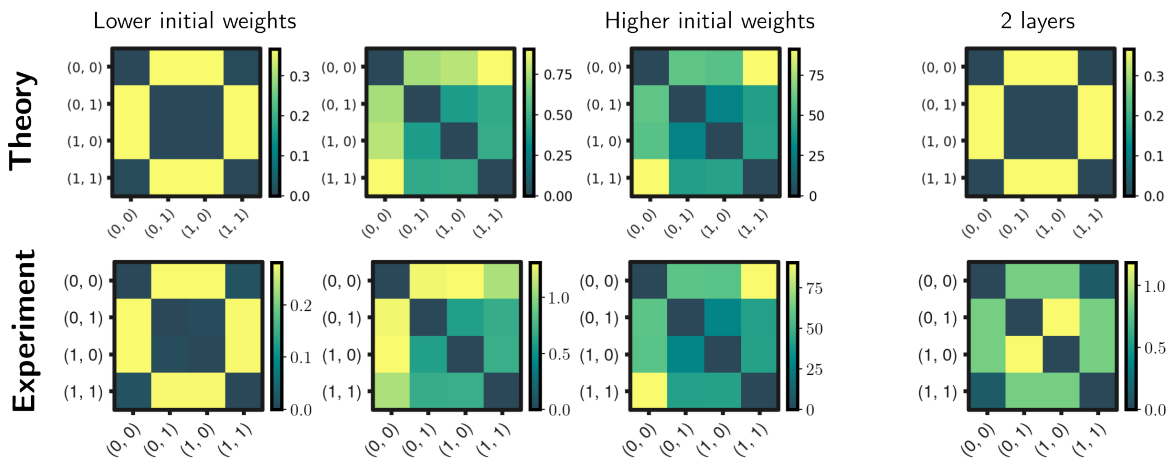

Figure 16. Representational structure in the XOR task, with Adam used as an optimizer. The learned hidden representational pair-wise distances match the theory prediction at varying initial weight scales for the default 20 layer network (**left**), but not for a 2 layer network at small weights (**right**).
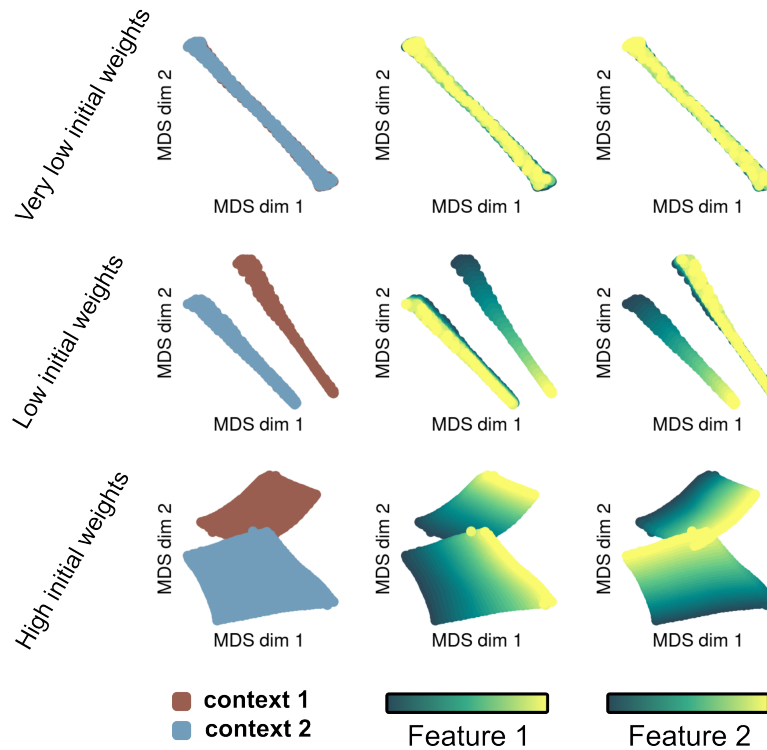
Figure 17. Representational structure in the orthogonal feature task, with Adam used as an optimizer. A multidimensional scaling projection of the representational structure of a 5-layer, tanh, fully connected network with 100 hidden units per layer, trained on the orthogonal feature dataset. At small initial weights the task-irrelevant direction collapses in each context, but remains at large initial weights.
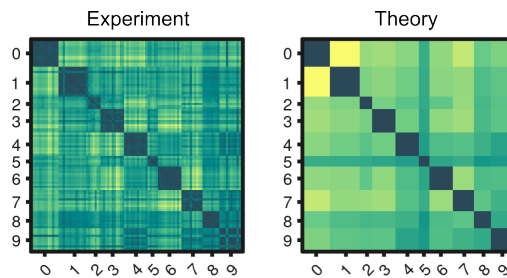

Figure 18. Representational structure on the MNIST dataset at small initial weights, with Adam used as an optimizer. Pairwise distances of the first 100 datapoints in MNIST after averaging over 50 trials of a 4-layer, leaky-ReLU, fully connected network trained on the full dataset (**left**) compared to the theory after exponential weighing (**right**).