INITIALIZING RELU NETWORKS IN AN EXPRESSIVE SUBSPACE OF WEIGHTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Using a mean-field theory of signal propagation, we analyze the evolution of correlations between two signals propagating forward through a deep ReLU network with correlated weights. Signals become highly correlated in deep ReLU networks with uncorrelated weights. We show that ReLU networks with anti-correlated weights can avoid this fate and have a chaotic phase where the signal correlations saturate below unity. Consistent with this analysis, we find that networks initialized with anti-correlated weights can train faster by taking advantage of the increased expressivity in the chaotic phase. An intiialization scheme combining this with a previously proposed strategy of using an asymmetric initialization to reduce dead node probability shows consistently lower training times compared to various other initializations on synthetic and real-world datasets. Our study suggests that use of initial distributions with correlations in them can help in reducing training time.

1 INTRODUCTION

Rectified Linear Unit (ReLU) Fukushima (1969); Fukushima & Miyake (1982) is the most widely used non-linear activation function in Deep Neural Networks (DNNs) LeCun et al. (2015); Ramachandran et al. (2018); Nair & Hinton (2010), applied to various tasks like computer vision Glorot et al. (2011b); Krizhevsky et al. (2012); He et al. (2015), speech recognition Maas et al. (2013); Tóth (2013); Hinton et al. (2012), intelligent gaming Silver et al. (2016), and solving scientific problems Seif et al. (2019). ReLU, $\phi(x) = \max(0, x)$, outperforms most of the other activation functions proposed Glorot et al. (2011a). It has several advantages over other activations. ReLU activation function is computationally simple as it essentially involves only a comparison operation. ReLU suffers less from the vanishing gradients, a major problem in training networks with sigmoid-type activations that saturate at both ends Glorot et al. (2011b). They generalize well even in the overly parameterized regime Maennel et al. (2018).

Despite its success, ReLU also has a few drawbacks, one of which is the dying ReLU problem He et al. (2015); Trottier et al. (2017). The dying ReLU is a type of vanishing gradient problem in which the network outputs zero for all inputs and is dead. There is no gradient flow in this state. ReLU also suffers from exploding gradient problem, which occurs when backpropagating gradients become large Hanin (2018).

Several methods are proposed to overcome the vanishing/exploding gradient problem; these can be classified into three categories Lu et al. (2020). The first approach modifies the architecture, which includes using modified activation functions Ramachandran et al. (2018); He et al. (2015); Trottier et al. (2017); Clevert et al. (2016); Klambauer et al. (2017); Hendrycks & Gimpel (2016), adding connections between non-consecutive layers (residual connections)He et al. (2016), and optimizing network depth and width. The proposed activations are often computationally less efficient and require a fine-tuned parameter Lu et al. (2020). The second approach relies on normalization techniques Ba et al. (2016); Ioffe & Szegedy (2015); Salimans & Kingma (2016); Ulyanov et al. (2016); Wu & He (2018), the most popular one being batch normalization Ioffe & Szegedy (2015). Batch normalization prevents the vanishing and exploding gradients by normalizing the output at each layer but with an additional computational cost of up to 30% Mishkin & Matas (2016). A related strategy involves using the self-normalizing activation (SeLU), which by construction ensures output with zero mean and unit variance Klambauer et al. (2017). The third approach focuses on the

initialization of the weights and biases. As local (gradient-based) algorithms are used for optimization Kingma & Ba (2015); Zeiler (2012); Duchi et al. (2011), it is challenging to train deep networks with millions of parameters Du et al. (2019); Srivastava et al. (2015), and optimal initialization is essential for efficient training Nesterov (2014). He-initialization He et al. (2015) is a commonly used strategy that uses uncorrelated Gaussian weights with variance $\frac{2}{N}$, where N is the width of the network. Recently, Lu et al. (2020) proposed Random asymmetric initialization (RAI), which reduces the probability of dead ReLU at the initialization. In this paper, we aim to further improve the initialization scheme for ReLU networks.

A growing body of work has analyzed signal propagation in infinitely wide networks to understand the phase diagram of forward-propagation in DNNs Saxe et al. (2014); Poole et al. (2016); Raghu et al. (2017); Schoenholz et al. (2017); Lee et al. (2018); Hayou et al. (2019a); Li & Saad (2018; 2020); Bahri et al. (2020). We mention a few results for ReLU networks. Hayou et al. (2019a) showed that correlations in input signals propagating through a ReLU network always converge to one. Many other works found that ReLU networks are in general biased towards computing simpler functions De Palma et al. (2019); Rahaman et al. (2019); He et al. (2020); Valle-Perez et al. (2019); Hanin & Rolnick (2019), which may account for their better generalization properties even in the overly parameterized regime. However, from their successful application in different domains, one may guess that they should be capable of computing more complex functions. There might be a subspace of the parameters where the network can represent complex functions.

Li & Saad (2018; 2020) applied weight and input perturbations to analyze the function space of ReLU networks. They found that ReLU networks with anti-correlated weights compute richer functions than uncorrelated/positively correlated weights. Consistent with this, Shang et al. (2016) found that ReLU CNN's produce anti-correlated feature matrices after training. These studies motivated us to analyze the phase diagram of signal propagation in ReLU networks with anti-correlated weights.

Following the mean-field theory of signal propagation proposed by Poole et al. (2016), we found that ReLU networks with anti-correlated weights have a chaotic phase, which implies higher expressivity. In contrast, ReLU networks with uncorrelated weights do not have a chaotic phase. Furthermore, we find that initializing ReLU networks with anti-correlated weights results in faster training. We call it Anti-correlated initialization (ACI). Additional improvement in performance is achieved by incorporating RAI, which reduces the dead node probability. This combined scheme, which we call Random asymmetric anti-correlated initialization (RAAI), is the main result of this work and is defined as follows. We pick weights and bias incoming to each node from anti-correlated Gaussian distribution and replace one randomly picked weight/bias with a random variable drawn from a beta distribution. The code to generate weights drawn from the RAAI distribution is given in Appendix G. We analyze the correlation properties of RAAI and show that it performs better than the best-known initialization schemes on tasks of varying complexity. It may be of concern that initialization in an expressive space may lead to overfitting, and we do observe the same for ACI for deeper networks and complex tasks. In contrast, RAAI shows no signs of overfitting and performs consistently better than all other initialization schemes.

We organize the article as follows. First, we contrast the mean-field analysis of ReLU networks with correlated weights with uncorrelated in Section 2. Next, Section 3 analyzes the critical properties of correlations in input signals for RAI and RAAI. Then, in Section 4, we describe the various tasks used to validate the performance of different initialization schemes in Section 5. Lastly, Section 6 concludes the article.

2 Mean-field analysis of signal propagation with correlated weights

This section presents the mean-field theory of signal propagation (proposed by Ref. Poole et al. (2016)) in ReLU networks with correlated weights and compares it with uncorrelated weights. Unlike Ref. Li & Saad (2018; 2020), which study perturbation to a ReLU network, we aim to understand the phase diagram of the signal propagation. Furthermore, we provide numerical results to corroborate the mean-field results.

Consider a fully connected neural network with L layers (in addition to the input layer) and N_l nodes in layer l. The layer index ranges between 0 and L. For an input signal $\boldsymbol{s}^0 = x$, we denote

the pre-activation at node i in layer l by $h_i^l(x)$ and activation by $s_i^l(x)$. A signal $(s_1^{l-1}, \ldots, s_{N_l}^{l-1})$ at layer $l-1$ propagates to layer l by the rule

$$h_i^l(x) = \sum_{j=1}^{N_{l-1}} w_{ij}^l s_j^{l-1}(x) + b_i^l \qquad \text{where } l \in \{1, L\}$$
$$s_i^l(x) = \phi(h_i^l(x)),$$

where ϕ is the non-linear activation function and w_{ij}^l, b_i^l are the weights and biases. We consider correlations within the set of weights $(\boldsymbol{w}_i^l)$ incoming to each node i. The correlated Gaussian distribution is

$$P(\boldsymbol{w}_1^l, \boldsymbol{w}_2^l, \boldsymbol{w}_3^l \ldots) = \prod_i^{N_l} \frac{e^{\left(-\frac{1}{2}(\boldsymbol{w}_i^l)^T A^{-1} \boldsymbol{w}_i^l\right)}}{\sqrt{(2\pi)^{N_{l-1}}|A|}} \quad \text{with } A = \frac{\sigma_w^2}{N_{l-1}}\left(\mathbb{I} - \frac{k}{1+k}\frac{J}{N_{l-1}}\right), \tag{1}$$

where $\mathbb{I}$ is the identity matrix, J is an all-ones matrix, and k parameterizes the correlation strength. Positively correlated and anti-correlated regimes correspond to the regions $-1 < k < 0$ and $k > 0$, respectively, whereas $k = 0$ generates uncorrelated weights. The overall scaling by $1/N_{l-1}$ in the covariance matrix ensures that the input contribution from the last layer to each node is $\mathcal{O}(1)$. The bias is drawn from a Gaussian distribution $b_i^l \sim \mathcal{N}(0, \sigma_b^2)$. Note that weights reaching two different nodes are uncorrelated, and also the bias is uncorrelated with the weights.

To track the layer-wise information flow, consider the squared length and overlap of the pre-activations for two input signals, $\boldsymbol{s}^0 = x_1$ and $\boldsymbol{s}^0 = x_2$, after propagating to layer l

$$q_h^l(x_a) = \frac{1}{N_l}\sum_{i=1}^{N_l}\left(h_i^l(x_a)\right)^2 \qquad \text{where } a \in \{1,2\} \text{ and } 1 \le l \le L$$
$$q_h^l(x_1, x_2) = \frac{1}{N_l}\sum_{i=1}^{N_l} h_i^l(x_1) h_i^l(x_2).$$

Assuming self averaging, consider an average over the weights and the bias incoming to layer l. For simplicity of notations later, we use the same symbol for averaged q_h^l.

$$q_h^l(x_a) = \frac{\sigma_w^2}{N_{l-1}}\sum_{j,m=1}^{N_{l-1}}\left(\delta_{j,m} - \frac{k}{1+k}\frac{1}{N_{l-1}}\right)\phi(h_j^{l-1}(x_a))\phi(h_m^{l-1}(x_a)) + \sigma_b^2$$
$$q_h^l(x_1, x_2) = \frac{\sigma_w^2}{N_{l-1}}\sum_{j,m=1}^{N_{l-1}}\left(\delta_{j,m} - \frac{k}{1+k}\frac{1}{N_{l-1}}\right)\phi(h_j^{l-1}(x_1))\phi(h_m^{l-1}(x_2)) + \sigma_b^2,$$

For large width, each h_i^{l-1} is a weighted sum of a large number of zero-mean random variables. Thus, we expect the joint distribution of $h_i^{l-1}(x_1)$ and $h_i^{l-1}(x_2)$ to converge to a zero-mean Gaussian with a covariance matrix with diagonal entries $q_h^{l-1}(x_1), q_h^{l-1}(x_2)$ and off-diagonal entries $q_h^{l-1}(x_1, x_2)$. On replacing the average over h_i^{l-1} (this is equivalent to considering an average over all previous layers) in the last layer with an average over this Gaussian distribution, we obtain iterative maps for the length and overlap. Specializing to ReLU activation yeilds the equations

$$q_h^l(x) = \frac{\sigma_w^2}{2}\left(1 - \frac{k}{1+k}\frac{1}{\pi}\right)q_h^{l-1}(x) + \sigma_b^2$$
$$q_h^l(x_1, x_2) = \frac{\sigma_w^2}{2}\left(f(c_h^{l-1}) - \frac{k}{1+k}\frac{1}{\pi}\right)\sqrt{q_h^{l-1}(x_1) q_h^{l-1}(x_2)} + \sigma_b^2 \tag{2}$$
$$f(c_h^{l-1}) = \frac{c_h^{l-1}}{2} + \frac{c_h^{l-1}}{\pi}\sin^{-1}(c_h^{l-1}) + \frac{1}{\pi}\sqrt{1 - (c_h^{l-1})^2},$$

where $c_h^l = \frac{q_h^l(x_1,x_2)}{\sqrt{q_h^l(x_1)q_h^l(x_2)}}$ is the correlation coefficient between the two signals reaching layer l (for details of the derivation, see Appendix A).

Poole et al. (2016) found that the signal's length reaches its fixed point within a few layers, and the fixed point of the correlation coefficient, c_h^*, can be estimated with the assumption that $q_h^l(x)$ has reached its fixed point q_h^*. We can check that $c_h^* = 1$ is always a fixed point of the recursive map (Eqn. 2) under this assumption. The stability of the fixed point $c_h^* = 1$ is determined by

$$\chi_1 \equiv \frac{\partial c_h^l}{\partial c_h^{l-1}}\Big|_{c_h^{l-1}=1},$$

which evaluates to $\frac{\sigma_w^2}{2}$. χ_1 separates the parameter space into two phases —first, an ordered phase with $\chi_1 < 1$, where the $c_h^* = 1$ fixed point is stable; and second, a chaotic phase with $\chi_1 > 1$, where the $c_h^* = 1$ fixed point is unstable. $\chi_1 = 1$ defines the phase boundary line. In the ordered phase, two distinct signals will become perfectly correlated asymptotically. In the chaotic phase, the correlations converge to a stable fixed point below unity. Two closely related signals will eventually lose correlations in this phase. This suggests that initializing the network with parameters (σ_w^2, σ_b^2) at the phase transition boundary (corresponding to an infinite depth of correlations) allows for an optimal information flow through the network Poole et al. (2016); Schoenholz et al. (2017)

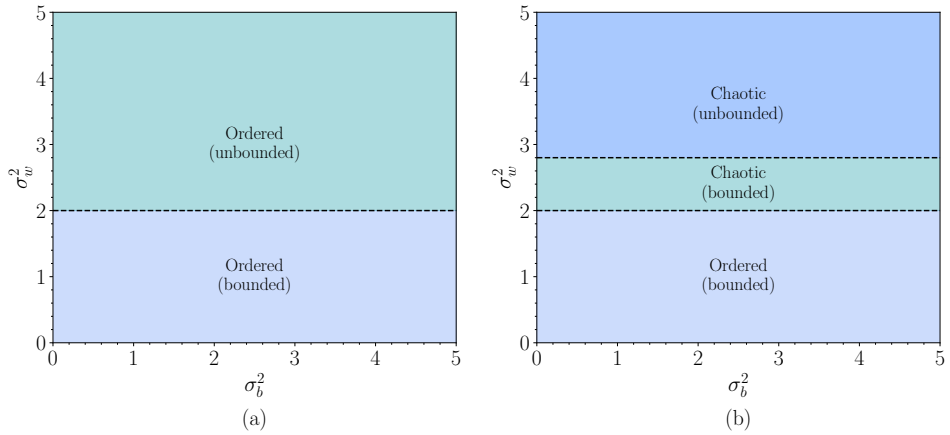

Figure 1: Phase diagram for ReLU networks with uncorrelated and anti-correlated Gaussian distributed weights. (a) ReLU networks with uncorrelated weights have two phases. First, a bounded phase where q_h^* is finite and second in which it diverges. The two phases are separated by $\sigma_w^2 = 2$. In both phases, any two signals will eventually become correlated. (b) ReLU networks with anti-correlated weights have three phases. In addition to the transition between the bounded and unbounded phases (at $\sigma_w^2 = g_k$) there is an order to chaos transition at $\sigma_w^2 = 2$. The results are shown for $k = 100$.

ReLU networks with uncorrelated weights (k = 0) The above analysis is applied assuming q_h^* is finite shows that ReLU networks with uncorrelated weights do not have a chaotic phase, and any two signals propagating through a ReLU network become asymptotically correlated for all values of (σ_w^2, σ_b^2). In other words, $c_h^* = 1$ is always a stable fixed point. However, the parameter space can be classified into two phases based on the boundedness of the fixed point q_h^* of the length map (Eqn. 2) - first, a bounded phase where q_h^* is finite and non-zero; second, an unbounded phase, where q_h^* is either zero or infinite Lee et al. (2018); Hayou et al. (2019b). The two phases are separated by the boundary $\sigma_w^2 = 2$. Figure 1a depicts the phase diagram for ReLU networks with uncorrelated weights. Note that the analysis of the stability of $c_h^* = 1$ fixed point in ReLU networks is valid only in the bounded phase. However, numerical results presented in Fig. 2 indicate that the fixed point remains stable even in the unbounded phase.

ReLU networks with correlated weights The phase diagram for ReLU networks with correlated weights can be analyzed similarly. The length is bounded if $\sigma_w^2 < g_k = \frac{2}{\left(1-\frac{k}{1+k}\frac{1}{\pi}\right)}$. Thus, for anti-

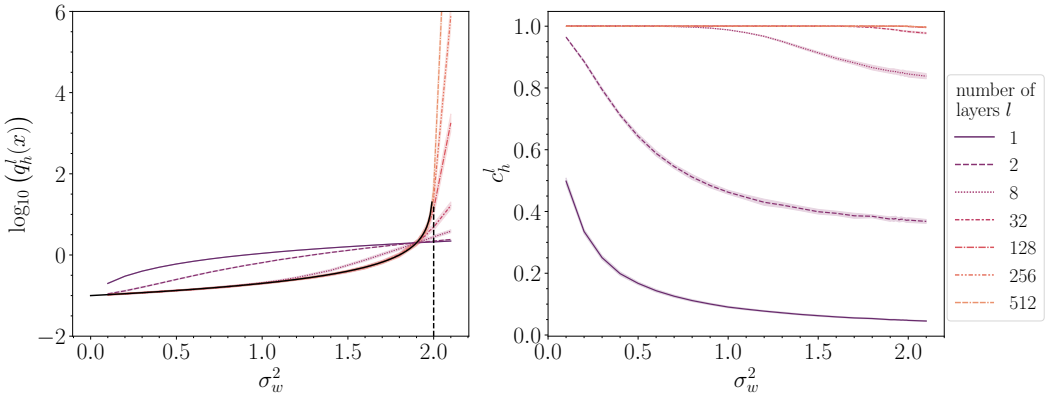

Figure 2: The above plots show the signal's length and correlation coefficient after propagating through l layers in a ReLU network with uncorrelated weights. We estimate the length and correlation coefficient averaged over $M = 1024$ input signals, and 40 networks with a constant width $N = 2048$. The shaded regions denotes the standard deviation. In the first panel, the vertical dashed line indicates the theoretical phase boundary $\sigma_w^2 = 2$, and the solid black line denotes the theoretical prediction for the length's fixed point. As the critical boundaries do not depend on the variance of the bias, we show results for $\sigma_b^2 = 0.1$ only. We find that $c_h^l \to 1$ for all values of σ_w^2 and σ_b^2.

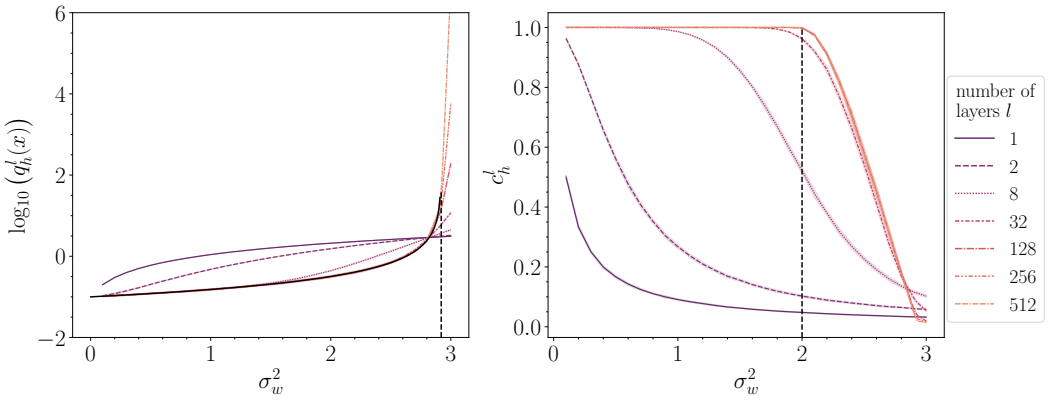

Figure 3: The above plots show the signal's length and correlation coefficient after propagating through l layers in a ReLU network with anti-correlated weights with a correlation strength $k = 100$. We estimate the length and correlation coefficient averaged over $M = 1024$ input signals, and 40 networks with a constant width $N = 2048$. The shaded regions denotes the standard deviation. The vertical dashed lines indicate the theoretical phase boundaries at $\sigma_w^2 = 2.92$ and $\sigma_w^2 = 2.0$ for $q_h^l(x)$ and $c_h^l(x)$. The solid black line in the first panel denotes the theoretical prediction for the length's fixed point. As the critical boundaries do not depend on the variance of bias, we show results for $\sigma_b^2 = 0.1$. Unlike the case of uncorrelated weights, we find a chaotic region.

correlated weights ($k > 0$), the boundary g_k moves upwards relative to the $k = 0$ case (see Fig. 1a). The $c_h^* = 1$ fixed point of the correlations is unstable in this region of the bounded phase.

In summary, anti-correlations induce a bounded chaotic phase in $2 < \sigma_w^2 < g_k$ (see Fig. 1b). We demonstrate these results numerically in Fig. 3 for a correlation strength of $k = 100$. As predicted by the above equations, the stability of the fixed point $c_h^* = 1$ changes at $\sigma_w^2 = 2$, and the length diverges at $g_{k=100} = 2.92$. In contrast, for positively correlated weights, the length's fixed point boundary shifts downward resulting in a similar phase diagram as uncorrelated weights.

As a result, a ReLU network with anti-correlated weights can be more expressive by taking advantage of a chaotic phase, and it may be beneficial for a ReLU network to remain in this subspace. Thus, we propose initializing ReLU networks with anti-correlated weights at the order to

chaos boundary $(\sigma_w^2, \sigma_b^2) = (2, 0)$. We call it Anti-Correlated Initialization (ACI). Appendix B demonstrates that ReLU networks initialized with anti-correlated weights give an advantage over He initializations for a range of tasks.

Many alternatives are proposed to improve ReLU networks Trottier et al. (2017); Lu et al. (2020); Clevert et al. (2016). Of particular interest is Random asymmetric initialization (RAI), which aims to increase expressivity through an independent strategy of reducing the dead node probability. In the next section, we analyze the correlation properties of RAI and then combine it with ACI to propose a new initialization scheme RAAI, which has both a chaotic phase and low dead node probability.

3 Random Asymmetric Anti-correlated Initialization

We begin by analyzing critical properties of Random asymmetric initialization (RAI) proposed in Lu et al. (2020) to reduce the dead node probability. For ReLU networks with symmetric distributions for weights and biases, the dead node probability is half. RAI reduces it by initializing one of the weights/the bias incoming to each node from a distribution with positive support (like the beta distribution), resulting in a positive mean for the pre-activations. Lu et al. (2020) proposes initializing RAI with a variance $\sigma_w^2 = 0.36$ to ensure that the signal's length is bounded. We analyzed the correlation properties of RAI and found that $c_h^* = 1$ is always a fixed point of the recursive maps (see Appendix C). Deriving the stability condition for the fixed point $c_h^* = 1$ even with the mean-field assumptions is difficult. However, numerical results presented in Figure 4 show that $c_h^* = 1$ is always a stable fixed point (right panel), and the length remains finite for σ_w^2 up to 0.72 (left panel). A qualitative picture of the phase diagram can be captured with additional assumptions over the mean-field approximation (see Appendix D).

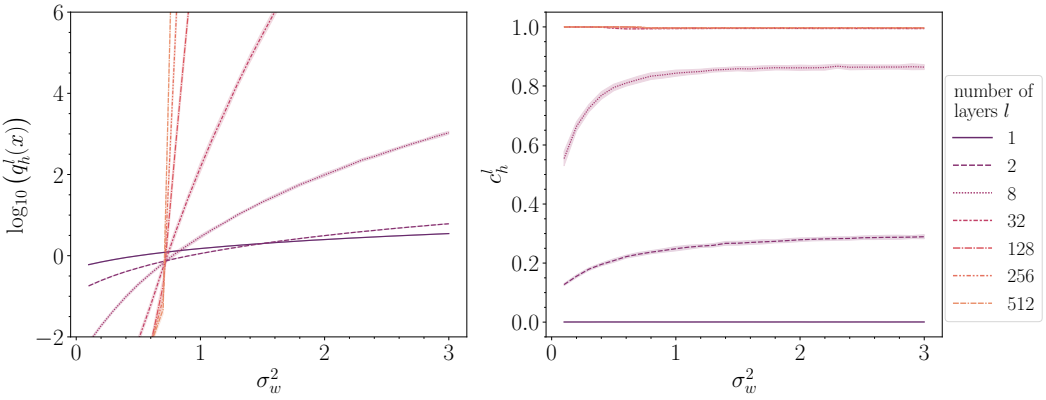

Figure 4: The above plots show the signal's length and correlation coefficient after propagating through l layers in a ReLU network with RAI. We estimate the length and correlation coefficient averaged over $M = 1024$ input signals, and 40 networks with a constant width $N = 2048$. The shaded regions denotes the standard deviation. Similar to Fig. 2, the chaotic region is absent.

RAI focuses on decreasing the dead node probability to increase expressive power, whereas ACI uses anti-correlated weights to improve the expressivity. As RAI and ACI increase the expressivity using different mechanisms, we explore the possibility of combining the two. We call it Random asymmetric anti-correlated initialization (RAAI). To prepare weights drawn from RAAI, we consider anti-correlated Gaussian weights and bias incoming to each node (like Eqn. 3) and replace one randomly picked weight/bias with a random number drawn from a beta distribution. Note that the weights and biases reaching different nodes are uncorrelated. Like ACI, we observe three phases for RAAI. Numerical results presented in Fig. 5 suggest that the order to chaos boundary is around $\sigma_w^2 = 0.9$, and the length diverges for $\sigma_w^2 > 1.2$. Again, a qualitative picture of the phase diagram can be captured with additional assumptions over the mean-field approximation (see Appendix E).

In summary, RAAI has a chaotic phase like ACI, and a lower dead node probability like RAI, as can be checked numerically. As RAAI inherits the advantages of both strategies, we expect it to be a strong candidate for initializing ReLU networks. Table 1 summarizes and compares different

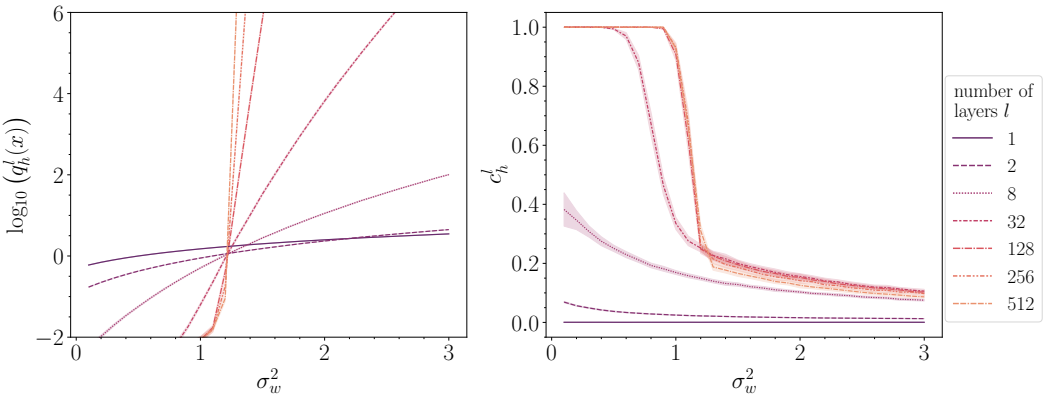

Figure 5: The above plots show the signal's length and correlation coefficient after propagating through l layers in a ReLU network with RAAI. We estimate the length and correlation coefficient averaged over $M = 1024$ input signals, and 40 networks with a constant width $N = 2048$. The shaded regions denotes the standard deviation. Similar to ACI, we find a chaotic region. However, the correlations do not converge to zero even for large σ_w^2.

Table 1: A comparison of different initialization schemes for ReLU networks. The dead node probabilities are calculated numerically for input signals drawn from the standard normal distribution.

Initialization scheme	σ_w^2	k	Chaotic phase	Dead node probability
He	2.0	0.0	No	0.5
ACI	2.0	100.0	Yes	0.5
RAI	0.36	0.0	No	0.36
RAAI	0.92	100.0	Yes	0.36

initialization schemes. In the following sections, we analyze the training dynamics and performance of RAAI and compare it with other initialization schemes.

4 Training tasks

This section describes various tasks used to analyze the dynamics and performance of different initialization schemes. We consider a variant of teacher-student setup Seung et al. (1992), in which a student ReLU network is trained with examples generated by an untrained teacher network. We consider three different tasks with varying complexities.

1. First, a standard teacher task, in which the training data is generated by a ReLU network of the same size as the student network, initialized with He initialization.

2. Next, we consider a simple teacher task in which the capacity of the teacher network is much lower than the student network. In many real data sets, the high-dimensional inputs lie in a low-dimensional manifold Goldt et al. (2020), which motivates us to consider a simple teacher task. We consider a single-layer ReLU network with $N = 10$ nodes, initialized with He initialization.

3. Lastly, we consider a complex teacher task, in which the complexity of the teacher network is more than the student network. We consider a teacher network with tanh activation of the same size as the student network initialized in the chaotic regime, $(\sigma_w^2, \sigma_b^2) = (1.5, 0)$ Poole et al. (2016). A ReLU network initialized with symmetric distributions has half of the nodes dead. Therefore, it has a lower capacity than a tanh network of the same size.

We consider an $L = 10$ layered (in addition to input layer) student network with a constant width $N = 100$, trained using SGD and Adam algorithms (for further details, see 6).

5 COMPARISON OF LEARNING DYNAMICS FOR DIFFERENT INITIALIZATION SCHEMES

This section compares the performance of RAAI with other initialization schemes listed in Table 1 on tasks described in Section 4.

Standard teacher task Figure 6 shows the average validation loss for the standard tasks trained with SGD and Adam algorithms. We observe that RAAI performs better than all other schemes.

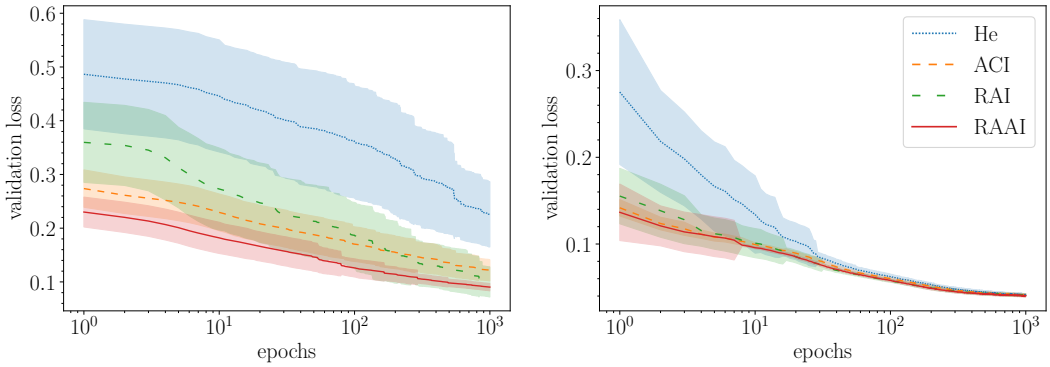

Figure 6: Average validation loss for ReLU networks trained on the standard teacher task with SGD (left) and Adam optimizer (right) for different initialization schemes. The shaded region shows the standard deviation around the average loss.

Simple teacher task Figure 7 shows the average validation loss for the simple teacher task. Similar to the standard teacher task, RAAI performs better than or on par with other initialization schemes.

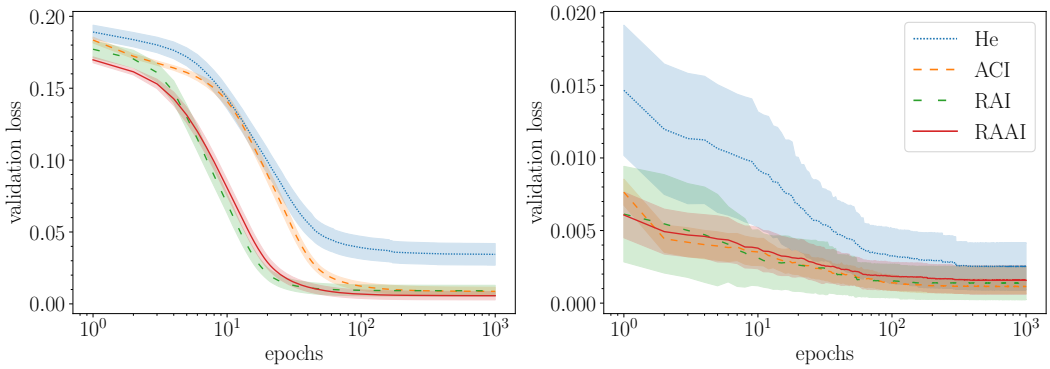

Figure 7: Average validation loss for ReLU networks trained on the simple teacher task with SGD (left) and Adam optimizer (right) for different initialization schemes. The shaded region shows the standard deviation around the average loss.

Complex teacher task Figure 8 shows the average validation loss for the complex teacher task. We observe that for a complex teacher, ACI starts to perform worse when trained with SGD algorithm, whereas, RAAI faces no such problem and performs better or on par with other initializations.

In various scenarios, RAI performs comparable to RAAI, however, we find that RAAI performs better RAI on real-world datasets. We present different intitialization schemes on three different real-world datasets —MNIST, Fashion-MNIST and CIFAR-10. We find that RAAI outperforms all other intialization schemes at early training steps and all initialization schemes perform equally well after a few epochs. For further details, see Appendix F.

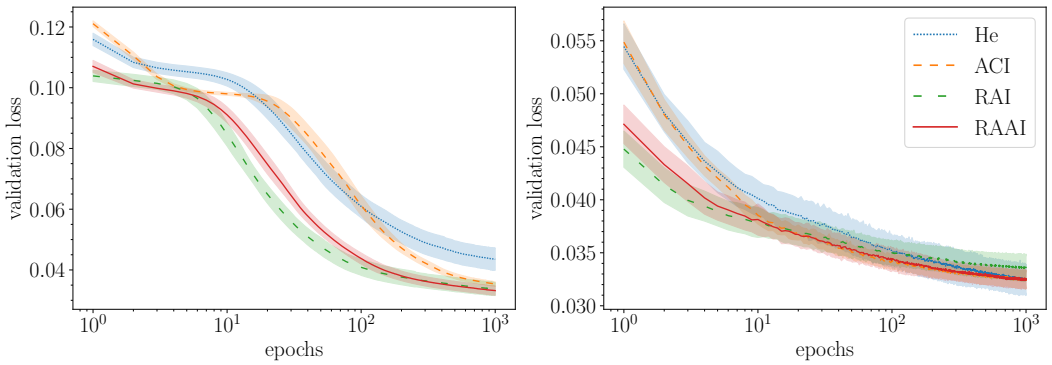

Figure 8: Average validation loss for ReLU networks trained on the complex teacher task with SGD (left) and Adam optimizer (right) for different initialization schemes. The shaded region shows the standard deviation around the average loss.

6 Discussion and Conclusion

In this article, we analyzed the evolution of correlation between signals propagating through a ReLU network with correlated weights using the mean-field theory of signal propagation. Multiple studies show that ReLU networks with uncorrelated weights are biased towards computing simpler functions, but ReLU networks do perform complex tasks in practice. Unlike ReLU networks with uncorrelated weights, ReLU networks with anti-correlated weights reaching a node have a chaotic phase where correlation saturates below unity. This suggests that such networks can exhibit higher expressivity. Although we have focused on the ReLU networks in this study, anti-correlation in weights may be useful in general. Networks with other non-linear activation functions like tanh, SELU, and sigmoid have a chaotic phase even with uncorrelated weights. In these cases, the weight correlations may still help to tune the phase boundaries and expressivity of the networks.

We further investigated the possibility that ReLU networks with the enhanced expressivity may prove beneficial in faster learning. Comparison of training and test performance of networks in a range of teacher-student setups clearly showed that networks with anti-correlated weights learn faster. While ACI shows better learning performance in general, it shows poor performance with SGD during an intermediate learning stage when the teacher network has a relatively higher capacity. We believe that this may be due to the system getting stuck in local minima. This is consistent with the absence of a similar regime on training with Adam optimizer. On training deeper networks with ACI, we found that it overfits, but this can be avoided by fine-tuning correlation strength k. We also investigated a possible improvement in training time from adding a regularization term in the loss function that favors anti-correlated weights, but our attempts did not show any systematic results.

We compared ACI with a recently proposed initialization scheme called RAI, which introduces a systematic asymmetry (around 0) in the weights to decrease dead node probability. We find that the relative performance between RAI and ACI depends on the task and the optimization algorithm. RAI improves expressivity by reducing the dead node probability, whereas ACI achieves the same by inducing a chaotic phase. As RAI and ACI rely on different mechanisms, we explored a strategy of combining the two initialization schemes. We analyzed the correlation properties of the combined scheme, which we call RAAI and found that it has a chaotic phase like ACI. We demonstrated that RAAI leads to faster training and learning than commonly-used methods on various teacher tasks of a range of complexity. For different initialization schemes, the behavior of the training dynamics at large epochs may depend on the optimizer and training data, however RAAI shows a definite advantage over other schemes when using the SGD optimizer, especially in early training epochs. In addition to faster training, RAAI also shows no sign of overfitting and thus improves on the simpler strategy that relies only on anti-correlations. Our study has focused on adding simple two point correlation in the initial distributions motivated by a richer phase space for ReLU networks with anti-correlated weights. This simplest deviation from the uncorrelated Gaussian distribution showed a consistent advantage in terms of training time, suggesting that initialization with more complex and tailored correlations may lead to better performance.

REFERENCES

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: Large-scale machine learning on heterogeneous distributed systems, 2015. URL http://download.tensorflow.org/paper/whitepaper2015.pdf.

Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016. URL http://arxiv.org/abs/1607.06450.

Yasaman Bahri, Jonathan Kadmon, Jeffrey Pennington, Sam S. Schoenholz, Jascha Sohl-Dickstein, and Surya Ganguli. Statistical mechanics of deep learning. *Annual Review of Condensed Matter Physics*, 11(1):501–528, 2020. doi: 10.1146/annurev-conmatphys-031119-050745. URL https://doi.org/10.1146/annurev-conmatphys-031119-050745.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). In Yoshua Bengio and Yann LeCun (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1511.07289.

Giacomo De Palma, Bobak Kiani, and Seth Lloyd. Random deep neural networks are biased towards simple functions. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32, pp. 1964–1976. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/feab05aa91085b7a8012516bc3533958-Paper.pdf.

Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1675–1685. PMLR, 09–15 Jun 2019. URL http://proceedings.mlr.press/v97/du19c.html.

John C. Duchi, Elad Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. In *J. Mach. Learn. Res.*, 2011.

K. Fukushima. Visual feature extraction by a multilayered network of analog threshold elements. *IEEE Transactions on Systems Science and Cybernetics*, 5(4):322–333, 1969. doi: 10.1109/TSSC.1969.300225.

Kunihiko Fukushima and Sei Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In Shun-ichi Amari and Michael A. Arbib (eds.), *Competition and Cooperation in Neural Nets*, pp. 267–285, Berlin, Heidelberg, 1982. Springer Berlin Heidelberg. ISBN 978-3-642-46466-9.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In Geoffrey Gordon, David Dunson, and Miroslav Dudík (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 315–323, Fort Lauderdale, FL, USA, 11–13 Apr 2011a. JMLR Workshop and Conference Proceedings. URL http://proceedings.mlr.press/v15/glorot11a.html.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In Geoffrey Gordon, David Dunson, and Miroslav Dudík (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 315–323, Fort Lauderdale, FL, USA, 11–13 Apr 2011b. JMLR Workshop and Conference Proceedings. URL http://proceedings.mlr.press/v15/glorot11a.html.

Sebastian Goldt, Marc Mezard, Florent Krzakala, and Lenka Zdeborova. Modeling the influence of data structure on learning in neural networks: The hidden manifold model. *Physical Review X*, 10, 12 2020. doi: 10.1103/PhysRevX.10.041044.

B. Hanin and D. Rolnick. Deep relu networks have surprisingly few activation patterns. In *NeurIPS*, 2019.

Boris Hanin. Which neural net architectures give rise to exploding and vanishing gradients? In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31, pp. 582–591. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/13f9896df61279c928f19721878fac41-Paper.pdf.

Soufiane Hayou, Arnaud Doucet, and Judith Rousseau. On the selection of initialization and activation function for deep neural networks, 2019a. URL https://openreview.net/forum?id=H1lJws05K7.

Soufiane Hayou, Arnaud Doucet, and Judith Rousseau. On the impact of the activation function on deep neural networks training. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2672–2680. PMLR, 09–15 Jun 2019b. URL http://proceedings.mlr.press/v97/hayou19a.html.

Juncai He, Lin Li, Jinchao Xu, and Chunyue Zheng. Relu deep neural networks and linear finite elements. *Journal of Computational Mathematics*, 38(3):502–527, 2020. ISSN 1991-7139. doi: https://doi.org/10.4208/jcm.1901-m2018-0160. URL http://global-sci.org/intro/article_detail/jcm/15798.html.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *IEEE International Conference on Computer Vision (ICCV 2015)*, 1502, 02 2015. doi: 10.1109/ICCV.2015.123.

Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv: Learning*, 2016.

G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012. doi: 10.1109/MSP.2012.2205597.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 448–456, Lille, France, 07–09 Jul 2015. PMLR. URL http://proceedings.mlr.press/v37/ioffe15.html.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30, pp. 971–980. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/5d44ee6f2c3f71b73125876103c8f6c4-Paper.pdf.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 25, pp. 1097–1105. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Yann LeCun, Y. Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–44, 05 2015. doi: 10.1038/nature14539.

Jaehoon Lee, Jascha Sohl-dickstein, Jeffrey Pennington, Roman Novak, Sam Schoenholz, and Yasaman Bahri. Deep neural networks as gaussian processes. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1EA-M-0Z.

Bo Li and David Saad. Exploring the function space of deep-learning machines. *Physical Review Letters*, 120:248301, Jun 2018. doi: 10.1103/PhysRevLett.120.248301. URL https://link.aps.org/doi/10.1103/PhysRevLett.120.248301.

Bo Li and David Saad. Large deviation analysis of function sensitivity in random deep neural networks. *Journal of Physics A: Mathematical and Theoretical*, 53(10):104002, feb 2020. doi: 10.1088/1751-8121/ab6a6f. URL https://doi.org/10.1088/1751-8121/ab6a6f.

Lu Lu, Yeonjong Shin, Yanhui Su, and George Karniadakis. Dying relu and initialization: Theory and numerical examples. *Communications in Computational Physics*, 28:1671–1706, 11 2020. doi: 10.4208/cicp.OA-2020-0165.

Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

Hartmut Maennel, O. Bousquet, and S. Gelly. Gradient descent quantizes relu network features. *ArXiv*, abs/1803.08367, 2018.

Dmytro Mishkin and Jiri Matas. All you need is a good init. *CoRR*, abs/1511.06422, 2016.

Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pp. 807–814, Madison, WI, USA, 2010. Omnipress. ISBN 9781605589077.

Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer Publishing Company, Incorporated, 1 edition, 2014. ISBN 1461346916.

Ben Poole, Subhaneil Lahiri, Maithra Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. Exponential expressivity in deep neural networks through transient chaos. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29, pp. 3360–3368. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/file/148510031349642de5ca0c544f31b2ef-Paper.pdf.

Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl Dickstein. On the expressive power of deep neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 2847–2854. JMLR.org, 2017.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks, 2019. URL https://openreview.net/forum?id=r1gR2sC9FX.

Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions, 2018. URL https://openreview.net/forum?id=SkBYYyZRZ.

Tim Salimans and Diederik P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *NIPS*, 2016.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In Yoshua Bengio and Yann LeCun (eds.), *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.

Samuel S. Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. Deep information propagation. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=H1W1UN9gg.

Alireza Seif, M. Hafezi, and C. Jarzynski. Machine learning the thermodynamic arrow of time. *Nature Physics*, 17:105–113, 2019.

H. S. Seung, H. Sompolinsky, and N. Tishby. Statistical mechanics of learning from examples. *Physical Review A*, 45:6056–6091, Apr 1992. doi: 10.1103/PhysRevA.45.6056. URL https://link.aps.org/doi/10.1103/PhysRevA.45.6056.

Wenling Shang, Kihyuk Sohn, Diogo Almeida, and Honglak Lee. Understanding and improving convolutional neural networks via concatenated rectified linear units. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pp. 2217–2225. JMLR.org, 2016.

David Silver, Aja Huang, Christopher Maddison, Arthur Guez, Laurent Sifre, George Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–489, 01 2016. doi: 10.1038/nature16961.

Rupesh Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. *2015 Neural Information Processing Systems (NIPS 2015 Spotlight)*, 07 2015.

L. Tóth. Phone recognition with deep sparse rectifier neural networks. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6985–6989, 2013.

Ludovic Trottier, P. Giguère, and B. Chaib-draa. Parametric exponential linear unit for deep convolutional neural networks. *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 207–214, 2017.

D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *ArXiv*, abs/1607.08022, 2016.

Guillermo Valle-Perez, Chico Q. Camargo, and Ard A. Louis. Deep learning generalizes because the parameter-function map is biased towards simple functions. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rye4g3AqFm.

Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, 2018.

Matthew D. Zeiler. Adadelta: An adaptive learning rate method. *ArXiv*, abs/1212.5701, 2012.

ETHICS STATEMENT

We do not think that there are any direct negative impacts of this work, however, faster training using RAAI may lead to easier training of algorithms with negative societal impact.

REPRODUCIBILITY STATEMENT

We implemented feedforward networks in Tensorflow (version 2.2.0) Abadi et al. (2015) and train them using 10^5 training examples with a mean squared error loss, a mini-batch size of 32, and default parameters for the optimizers. The validation set contains 10^3 examples.

A DERIVATION OF THE LENGTH AND CORRELATION MAPS FOR CORRELATED WEIGHTS

This section derives the length and covariance maps for ReLU networks with correlated weights given by

$$P(\boldsymbol{w}_1^l, \boldsymbol{w}_2^l, \boldsymbol{w}_3^l \dots) = \prod_i^{N_l} \frac{\exp\left(-\frac{1}{2}(\boldsymbol{w}_i^l)^T A^{-1} \boldsymbol{w}_i^l\right)}{\sqrt{(2\pi)^{N_{l-1}}|A|}}, \tag{3}$$

where the covariance matrix given by

$$A = \frac{\sigma_w^2}{N_{l-1}}\left(\mathbb{I} - \frac{k}{1+k}\frac{J}{N_{l-1}}\right).$$

Here $\mathbb{I}$ is the identity matrix, J is an all-ones matrix, and k parameterizes the correlation strength. Positively correlated and anti-correlated regimes correspond to the regions $-1 < k < 0$ and $k > 0$, respectively, whereas, $k = 0$ generates uncorrelated weights. For simplicity, we consider $N_l = N$ in all layers, but the results hold for all N_l, as long as it is large.

A.1 DERIVATION OF LENGTH MAP

To derive the length map, we follow the approach introduced by 36. Assuming self-averaging, we obtain the average value of the squared length of a signal, $\boldsymbol{s}^0 = x$, after propagating to layer l by considering an average over weights and biases between layer l and $l-1$

$$\begin{aligned} q_h^l(x) &= \frac{1}{N}\left\langle \sum_{i=1}^N (h_i^l(x))^2 \right\rangle = \frac{1}{N}\sum_{i=1}^N \sum_{j,m=1}^N \left\langle w_{ij}^l w_{im}^l \right\rangle \phi(h_j^{l-1}(x))\phi(h_m^{l-1}(x)) + \left\langle (b_i^l)^2 \right\rangle \\ q_h^l(x) &= \frac{\sigma_w^2}{N}\sum_{j,m=1}^N \left(\delta_{j,m} - \frac{k}{1+k}\frac{1}{N}\right)\phi(h_j^{l-1}(x))\phi(h_m^{l-1}(x)) + \sigma_b^2, \end{aligned} \tag{4}$$

where we have used $\left\langle w_{ij}^l w_{im}^l \right\rangle = \frac{\sigma_w^2}{N}\left(\delta_{j,m} - \frac{k}{1+k}\frac{1}{N}\right)$ and $\left\langle (b_i^l)^2 \right\rangle = \sigma_b^2$. For large N, each $h_i^{l-1}(x)$ is a weighted sum of a large number of correlated random variables, which converges to a zero-mean Gaussian with a variance $q_h^{l-1}(x)$. Replacing the average over h_i^l at layer $l-1$ by a Gaussian distribution to get the general form of the recursive map. This average corresponds to averaging over all the weights and biases upto layer $l-1$.

$$q_h^l(x) = \sigma_w^2 \int Dz\ \phi\left(\sqrt{q_h^{l-1}(x)}\ z\right)^2 - \sigma_w^2 \frac{k}{1+k}\left[\int Dz\ \phi\left(\sqrt{q_h^{l-1}(x)}\ z\right)\right]^2 + \sigma_b^2,$$

where Dz is the standard normal distribution. For the second term in the Eqn. 4, we have used the fact that for different nodes $m \neq j$, $h_j^l(x)$ and $h_m^l(x)$ are uncorrelated random variables and ignored $\mathcal{O}(1/N)$ terms. Note that the weights and biases reaching two different nodes are uncorrelated. For a ReLU activation, we can perform the integrals to get the exact form of the recursive relation between $q_h^l(x)$ and $q_h^{l-1}(x)$

$$q_h^l(x) = \frac{\sigma_w^2}{2}\left(1 - \frac{k}{1+k}\frac{1}{\pi}\right) q_h^{l-1}(x) + \sigma_b^2. \tag{5}$$

.

A.2 DERIVATION OF THE COVARIANCE MAP

The covariance map can be derived similarly by considering an average over the weights and biases

$$q_h^l(x_1,x_2) = \frac{1}{N}\left\langle \sum_{i=1}^{N} h_i^l(x_1)h_i^l(x_2)\right\rangle = \sum_{j,m=1}^{N}\left\langle w_{ij}^l w_{im}^l\right\rangle \phi(h_j^{l-1}(x_1))\phi(h_m^{l-1}(x_2)) + \left\langle (b_i^l)^2\right\rangle$$

$$q_h^l(x_1,x_2) = \frac{\sigma_w^2}{N}\sum_{j,m=1}^{N}\left(\delta_{j,m} - \frac{k}{1+k}\frac{1}{N}\right)\phi(h_j^{l-1}(x_1))\phi(h_m^{l-1}(x_2)) + \sigma_b^2,$$

and then replacing the sum over neurons in the previous layer with an integral with a Gaussian measure. For large N, the joint distribution of $h_j^l(x_1)$ and $h_m^l(x_2)$ will converge to a two-dimensional Gaussian distribution with a covariance matrix

$$\Sigma_{l-1} = \begin{bmatrix} q_h^{l-1}(x_1) & q_h^{l-1}(x_1,x_2) \\ q_h^{l-1}(x_1,x_2) & q_h^{l-1}(x_2)\end{bmatrix}.$$

The correlations among $h_j^l(x_1)$ and $h_m^l(x_2)$ are induced as the two signals are propagating through the same network. Propagating this joint distribution across one layer, we obtain the iterative map

$$q_h^l(x_1,x_2) = \sigma_w^2\int Dz_1Dz_2\phi(u_1)\phi(u_{12}) - \sigma_w^2\frac{k}{k+1}\int Dz_1Dz_2\ \phi(u_1)\phi(u_2) + \sigma_b^2$$

$$u_1 = \sqrt{q_h^{l-1}(x_1)}z_1,\ u_{12} = \sqrt{q_h^{l-1}(x_2)}\left[c_h^{l-1}\ z_1 + \sqrt{1-(c_h^{l-1})^2}\ z_2\right]\ u_2 = \sqrt{q_h^{l-1}(x_2)}z_2,$$

where $c_h^l = \frac{q_h^l(x_1,x_2)}{\sqrt{q_h^l(x_1)q_h^l(x_2)}}$ is the correlation coefficient, and Dz_1, Dz_2 are standard normal Gaussian distributions. Again, in the second part of the above equation, we have used the fact that for $j \neq m$, $h_j^l(x_1)$ and $h_m^l(x_2)$ are uncorrelated random variables and have ignored $\mathcal{O}(1/N)$ terms. Further, we can perform the integrals for ReLU networks to get the exact recursive map

$$q_h^l(x_1,x_2) = \frac{\sigma_w^2}{2}\left(f(c_h^{l-1}) - \frac{k}{1+k}\frac{1}{\pi}\right)\sqrt{q_h^{l-1}(x_1)q_h^{l-1}(x_2)} + \sigma_b^2 \tag{6}$$

$$f(c_h^{l-1}) = \frac{c_h^{l-1}}{2} + \frac{c_h^{l-1}}{\pi}\sin^{-1}(c_h^{l-1}) + \frac{1}{\pi}\sqrt{1-(c_h^{l-1})^2}. \tag{7}$$

B TRAINING WITH ANTI-CORRELATED VS. POSITIVELY CORRELATED INITIALIZATION

This section compares the training dynamics and performance of ReLU networks initialized with different weight correlation strengths on tasks described in Section 4. We utilize the increased expressivity in ReLU networks with anti-correlated weights (ACI) at the initial training phase and compare its training dynamics with He initialization and positively correlated weight initialization. We observe that ACI provides a definite advantage over the other two initialization schemes. We also find that positively correlated weight ReLU networks train slower than He initialization, suggesting that anti-correlation may develop in weights during training. We choose three different correlation strengths; $k = 100$ induces anti-correlated weights, $k = -0.5$ produces positively correlated weights, and lastly, $k = 0$ corresponds to uncorrelated weights (He initialization). We train networks with two different optimization algorithms, SGD and Adam. For SGD, we train for 10^4 epochs, and for Adam, we train for 10^3 epochs.

Standard teacher task Figure 9 shows the average validation loss for different correlation strengths trained using SGD and Adam algorithms. We observe that ReLU networks initialized with ACI train faster than He initialization. In contrast, ReLU networks with positively correlated weights train slower than He initialization.

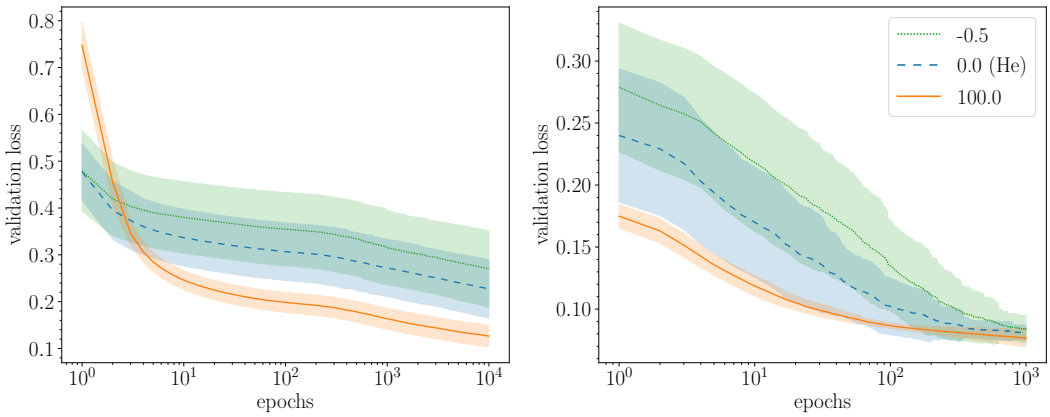

Figure 9: Average validation loss for ReLU networks trained on the standard teacher task with SGD (left) and Adam optimizer (right) for different weight correlations strengths.

Simple teacher task Figure 10 shows the average validation loss for the simple teacher task. We observe similar qualitative results as in the standard teacher task. For SGD, we observe an initial linear region in which all initialization schemes perform equally; however, at large epochs, ACI shows a definite advantage over other initializations.

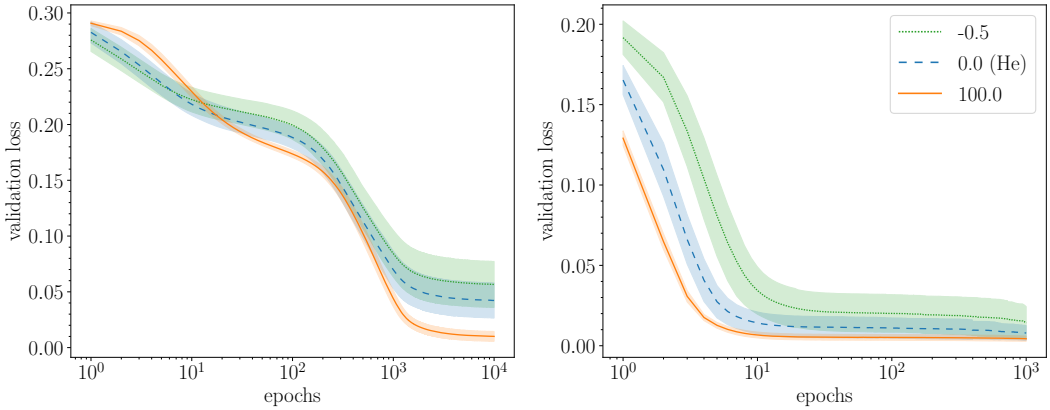

Figure 10: Average validation loss for ReLU networks trained on the simple teacher task with SGD (left) and Adam (right) optimizer for different weight correlations strengths.

Complex teacher task Figure 11 shows the average validation loss for a complex teacher task. For some intermediate regions, ACI performs worse than other initializations on training with SGD. The regions where ACI performs poorly shift depending on the complexity of the task.

C Derivation of the length and covariance map for RAI

To draw weights from the RAI distribution, we first initialize the weights and bias incoming to each node with a Gaussian distribution $\mathcal{N}(0, \frac{\sigma_w^2}{N})$. Next, we replace one weight or the bias incoming to each neuron by a random variable from beta distribution (see 18 for details). The weights and bias are treated on an equal footing. Thus, to simplify the notations, we incorporate the bias in the weight matrix by introducing a fictitious additional node with a constant value of one, i.e.,

$$\mathbf{s}^l(x) = [\phi(\mathbf{h}^l(x)), 1].$$

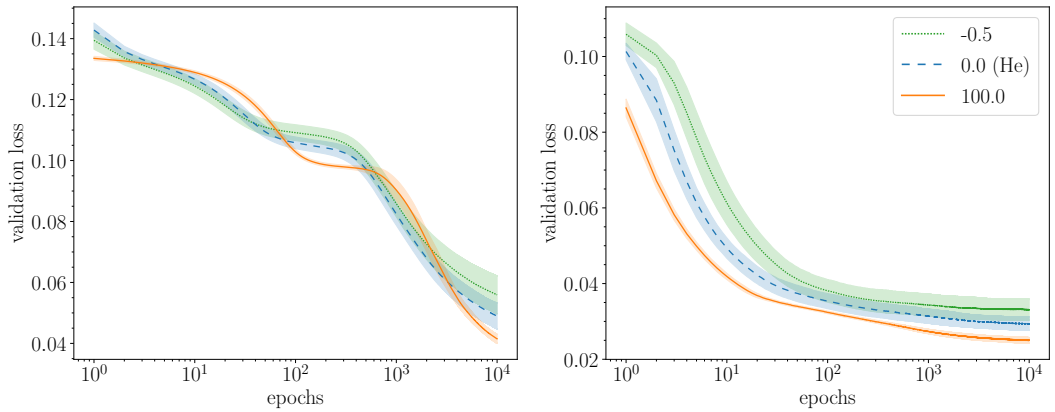

Figure 11: Average validation loss for ReLU networks trained on the complex teacher task with SGD (left) and Adam (right) optimizer for different weight correlations strengths.

The evolution equation is now given by

$$\mathbf{h}^l(x) = \mathbf{W}^l \cdot \mathbf{s^{l-1}}(x).$$

It is easier to track the evolution using the activation instead of the pre-activations. So we define a few covariance matrices which will come in handy

$$q_s^l(x_1, x_2) = \frac{1}{N+1} \sum_{i=0}^{N} s_i^l(x_1) s_i^l(x_2)$$
$$q_{-k_j^l}^l(x_1, x_2) = \frac{1}{N} \sum_{t \neq k_j^l}^{N} s_t^l(x_1) s_t^l(x_2) \ ,$$

where k_j^l tags variables associated with the special weight. We will use the notations $q_s^l(x) = q_s^l(x, x)$ and $q_{-k}^l(x) = q_{-k}^l(x, x)$. The corresponding correlation coefficients are given by,

$$c_s^l = \frac{q_s^l(x_1, x_2)}{\sqrt{q_s^l(x_1)\ q_s^l(x_2)}}$$
$$c_{-k_j^l}^l = \frac{q_{-k_j^l}^l(x_1, x_2)}{\sqrt{q_{-k_j^l}^l(x_1)\ q_{-k_t^l}^l(x_2)}}$$

C.1 Derivation of the length map for RAI

Given $h^{l-1}(x)$ and considering weights between layers l and $l-1$, we can view $h_j^l(x)$ as a random variable

$$h_j^l(x) = \sigma_w \sqrt{q_{-k_j^{l-1}}^{l-1}(x)}\ z + s_{k_j^{l-1}}^{l-1}(x)\ u,$$

where $z \sim \mathcal{N}(0, 1)$ and $u \sim \beta(2, 1)$. By applying the activation function and squaring it, we obtain

$$\phi(h_j^l(x))^2 = \phi\left(\sigma_w \sqrt{q_{-k_j^{l-1}}^{l-1}(x)}\ z + s_{k_j^{l-1}}^{l-1}(x)\ u\right)^2.$$

Next, we take an average over the weights and the special weight (average denoted by $< . >$) to get

$$\left\langle \phi(h_j^l(x))^2 | h^{l-1}(x) \right\rangle = \sum_{k_j^{l-1}=0}^{N} \frac{1}{N+1} \int dz\ du\ f(z)\ g(u)\ \phi\left(\sigma_w \sqrt{q^{l-1}_{-k_j^{l-1}}(x)}\ z + s^{l-1}_{k_j^{l-1}}(x)\ u\right)^2,$$

where $g(u) \sim \beta(2,1)$ distribution, and $f(z) \sim \mathcal{N}(0,1)$. We can take a sum over all nodes and re-write the equation in terms of the overlap

$$\left\langle q_s^l(x) | h^{l-1}(x) \right\rangle = \frac{1}{N+1}\left[1 + \sum_{j=1}^{N} \sum_{k_j^{l-1}=0}^{N} \frac{1}{N+1} \int dz du\ f(z) g(u)\ \phi\left(\sigma_w \sqrt{q^{l-1}_{-k_j^{l-1}}(x)}\ z + s^{l-1}_{k_j^{l-1}}(x)\ u\right)^2\right]. \tag{8}$$

C.2 Derivation of the covariance map for RAI

The covariance map can be derived similarly, with a key difference of covariance between the pre-activations. For two input signals $\boldsymbol{s}^0 = x_1$ and $\boldsymbol{s}^0 = x_2$, the covariance map reads

$$\left\langle q_s^l(x_1,x_2) | h^{l-1}(x_1), h^{l-1}(x_2) \right\rangle = \frac{1}{N+1}\left[1 + \sum_{j=1}^{N} \sum_{k_j^{l-1}=0}^{N} \frac{1}{N+1} \int dy_1 dy_2 du\ f(y_1,y_2)\ g(u) \times \right.$$
$$\left. \times\ \phi\left(\sigma_w \sqrt{q^{l-1}_{-k_j^{l-1}}(x_1)}\ y_1 + s^{l-1}_{k_j^{l-1}}(x_1)\ u\right) \phi\left(\sigma_w \sqrt{q^{l-1}_{-k_j^{l-1}}(x_2)}\ y_2 + s^{l-1}_{k_j^{l-1}}(x_2)\ u\right)\right], \tag{9}$$

where $f(y_1, y_2)$ is the joint Gaussian distribution of y_1 and y_2, with a covariance matrix given by

$$\Sigma^{l-1}_{k_j^l} = \begin{bmatrix} q^{l-1}_{-k_j^{l-1}}(x_1) & q^{l-1}_{-k_j^{l-1}}(x_1,x_2) \\ q^{l-1}_{-k_j^{l-1}}(x_1,x_2) & q^{l-1}_{-k_j^{l-1}}(x_2) \end{bmatrix}.$$

We can re-write Eqn. 9 in terms of $c^l_{-k_j^l}$

$$\left\langle q_s^l(x_1,x_2) | h^{l-1}(x_1), h^{l-1}(x_2) \right\rangle = \frac{1}{N+1}\left[1 + \sum_{j} \sum_{k_j^{l-1}} \frac{1}{N+1} \int dz_1\ dz_2\ du\ f(z_1) f(z_2)\ g(u) \times \right.$$
$$\times\ \phi\left(\sigma_w \sqrt{q^{l-1}_{-k_j^{l-1}}(x_1)}\ z_1 + s^{l-1}_{k_j^{l-1}}(x_1)\ u\right) \times$$
$$\left. \times\ \phi\left(\sigma_w \sqrt{q^{l-1}_{-k_j^{l-1}}(x_2)} \left[c^l_{-k_j^l} z_1 + \sqrt{1-(c^l_{-k_j^l})^2}\ z_2\right] + s^{l-1}_{k_j^{l-1}}(x_2)\ u\right)\right],$$

where $f(z_1) \sim f(z_2) \sim \mathcal{N}(0,1)$ are standard Gaussian distributions. As suggested by 36, we can find the fixed point of the correlation map under the assumption that the length $q_h^l(x)$ has reached its fixed point. It can be checked that $c_h^l = 1$ is a fixed point of the correlation map.

D STABILITY OF THE FIXED POINTS FOR THE LENGTH AND CORRELATION MAPS FOR RAI

D.1 STABILITY OF THE FIXED POINT FOR THE LENGTH MAP FOR RAI

The derivation of the analytical form of the length map (Eqn. 8) is difficult, and only bounds to the map have been derived (see 18). Inspired by the analytical form of the length map for the anti-correlated initialization and the analysis done by 18, we assume that the length map has a linear dependence on $q_s^{l-1}(x)$. Under this assumption, we can find the stability of the fixed point of the length map by taking a derivative with respect to $q_s^{l-1}(x)$. Yet another problem exists. While taking a derivative, we have to encounter derivatives of the form

$$\frac{\partial s^{l-1}_{-k_j^{l-1}}(x)}{\partial q_h^{l-1}(x)}.$$

To simplify the calculations further, we employ a mean-field type approach by approximating $q^{l-1}_{-k_j^l}(x)$ and $s^{l-1}_{k_j^{l-1}}$ by $q_s^{l-1}(x)$. Note that we can also approximate $s^{l-1}_{k_j^l}$ by its mean value, giving the same qualitative results. This simplifies Eqn. 8 to

$$\left\langle q_s^l(x)|h^{l-1}(x)\right\rangle = \frac{1}{N+1}\left[1+(N+1)\int dz\ du\ f(z)\ g(u)\ \phi\left(\sqrt{q_s^{l-1}(x)}(\sigma_w z+u)\right)^2\right].$$

To find the fixed point of the length map, we take a derivative wrt $q_s^{l-1}(x)$ to get the condition for stability of the fixed point q^*. We denote this derivative by ζ_{q^*}. It separates the dynamics into two phases —a bounded phase when $\zeta_{q^*}<1$, and an unbounded phase when $\zeta_{q^*}>1$.

$$\begin{aligned}
\zeta_{q^*} =& \frac{\partial q_s^l(x)}{\partial q_s^{l-1}(x)}\Big|_{q_s^{l-1}(x)=q^*}\\
\zeta_{q^*} =& \frac{\partial}{\partial q_s^{l-1}(x)}\int dz\ du\ f(z)\ g(u)\ \phi\left(\sqrt{q_s^{l-1}(x)}(\sigma_w z+u)\right)^2\\
\zeta_{q^*} =& \frac{1}{\sqrt{q_s^{l-1}(x)}}\int dz\ du\ f(z)\ g(u)\ (\sigma_w z+u)\phi'\left(\sqrt{q_s^{l-1}(x)}(\sigma_w z+u)\right)\phi\left(\sqrt{q_s^{l-1}(x)}(\sigma_w z+u)\right)
\end{aligned}$$

$$\zeta_{q^*} = \sigma_w^2\int dz\ du\ f(z)\ g(u)\ \left[\phi'\left(\sigma_w z+u\right)\right]^2+\sigma_w\int dz\ du\ f(z)\ g(u)\phi'\left(\sigma_w z+u\right)\phi\left(\sigma_w z+u\right) \tag{10}$$

where we have used the fact that for $a>0$, $\phi(ax)=a\ \phi(x)$. On evaluating the integral, we find that $\zeta_{q^*}=1$ when $\sigma_w^2=0.56$. This critical value underestimates the numerical value obtained in Fig. 4.

D.2 STABILITY OF THE FIXED POINT FOR THE CORRELATION MAP FOR RAI

Under the assumption, $q_s^l(x)\to q^*$, the correlation map has a fixed point $c_s^*=1$, and its stability is given by $\chi_1=\frac{\partial c_s^l}{\partial c_s^{l-1}}$ evaluated at $c_s^{l-1}=1$. But again, we get into the difficulties mentioned in the previous section, and we employ the same assumptions to arrive at a tractable equation for the correlation map

$$\begin{aligned}
\left\langle c_s^l|h^{l-1}(x_1),h^{l-1}(x_2)\right\rangle =& \frac{1}{q_s^*(x)}\frac{1}{N+1}\left[1+N\int dz_1\ dz_2\ du\ f(z_1)f(z_2)\ g(u)\ \times\right.\\
\times\ \ \phi\left(\sqrt{q_s^*(x)}(\sigma_w\ z_1+u)\right)&\left.\phi\left(\sqrt{q_s^*(x)}\left[c_s^{l-1}\ \sigma_w\ z_1+\sqrt{1-(c_s^{l-1})^2}\ \sigma_w\ z_2+u\right]\right)\right],
\end{aligned}$$

Next, we take a derivative to get the condition for the stability of the fixed point $c_h^* = 1$

$$\chi_1 = \frac{\partial c_h^l}{\partial c_h^{l-1}}\Big|_{c_h^{l-1}=1}$$
$$\chi_1 = \frac{1}{q_h^*(x)} \frac{\partial}{\partial c_h^{l-1}} \int dz_1 dz_2 du f(z_1) f(z_2) g(u) \phi\left(\sqrt{q_s^*(x)}(\sigma_w\ z_1 + u)\right) \times$$
$$\times\ \phi\left(\sqrt{q_s^*(x)}\left[c_s^{l-1}\ \sigma_w\ z_1 + \sqrt{1-(c_s^{l-1})^2}\ \sigma_w\ z_2 + u\right]\right)\Big|_{c_h^{l-1}=1}$$

$$\chi_1 = \sigma_w^2 \int dz\ du\ f(z)\ g(u)\ \left[\phi'(\sigma_w z + u)\right]^2. \tag{11}$$

The above equation is the same as the first term we obtained in the condition for the stability of the length map (Eqn. 10). We obtain a critical value of $\sigma_w^2 = 1.41$ by solving for $\chi_1 = 1$. We observe that the critical point for the length is smaller than the critical point of the correlation coefficient, and from our experience with ReLU networks calculations, we expect RAI to have an ordered phase only, which is confirmed by numerical results shown in Fig. 4.

E Derivation of length and correlation map for RAAI and stability conditions

E.1 Derivation for the length map for RAAI and the stability condition

Similar to the previous section, we can view $h_j^l(x)$ as a random variable

$$h_j^l(x) = \sigma_w \sqrt{\tilde{q}^{l-1}(x)}\ z + s^{l-1}_{-k_j^{l-1}}(x)\ u,$$

where $\tilde{q}^{l-1}(x) = q^{l-1}_{-k_j^{l-1}}(x)\left(1 - \frac{k}{1+k}\frac{1}{\pi}\right)$. Then, we can re-define σ_w as

$$\tilde{\sigma}^2 = \sigma_w^2\left(1 - \frac{k}{1+k}\frac{1}{\pi}\right),$$

which yields,

$$h_j^l(x) = \tilde{\sigma} \sqrt{q^{l-1}_{-k_j^{l-1}}(x)}\ z + s^{l-1}_{-k_j^{l-1}}(x)\ u,$$

Now, the entire analysis goes through as Appendix C.1, just with a re-definition of the variance. Now, we can read off the stability condition for the fixed point of the length map

$$\zeta_{q^*} = \tilde{\sigma}_w^2 \int dz\ du\ f(z)\ g(u)\ \left[\phi'\left(\tilde{\sigma} z + u\right)\right]^2 + \tilde{\sigma} \int dz\ du\ f(z)\ g(u) \phi'\left(\tilde{\sigma} z + u\right) \phi\left(\tilde{\sigma} z + u\right) \tag{12}$$

On solving the equations numerically, we observe that the length is bounded when $\sigma_w^2 < 1.75$, which overestimates the numerical value observed in Fig. 5.

E.2 Derivation for the correlation map for RAAI and the stability condition

The correlation map for RAAI can be derived similar to RAI (Appendix C.2), with a key difference being the covariance matrix. The covariance matrix, in this case, is

$$\Sigma^{l-1}_{k^l_j}(x_1,x_2) = \begin{bmatrix} q^{l-1}_{-k^{l-1}_j}(x_1) - \frac{k}{1+k}\left(m^l_{-k^{l-1}_j}(x_1)\right)^2 & q^{l-1}_{-k^{l-1}_j}(x_1,x_2) - \frac{k}{1+k} m^l_{-k^{l-1}_j}(x_1) m^l_{-k^{l-1}_j}(x_2) \\ q^{l-1}_{-k^{l-1}_j}(x_1,x_2) - \frac{k}{1+k} m^l_{-k^{l-1}_j}(x_1) m^l_{-k^{l-1}_j}(x_2) & q^{l-1}_{-k^{l-1}_j}(x_2) - \frac{k}{1+k}\left(m^l_{-k^{l-1}_j}(x_2)\right)^2 \end{bmatrix}.$$

Again, we can check that $c^*_s = 1$ is a fixed point of the dynamics. The stability of the fixed point under the assumptions considered in Appendix D.2 determines the order to chaos boundary at $\sigma^2_w = 1.41$. The critical value overestimates the numerical results presented in Fig. 5.

Note that instead of approximating $s^{l-1}_{k^{l-1}_j}$ by its RMS value $q^{l-1}_s(x)$, we can also approximate it by its mean value. In this case, we observe similar qualitative results. In Table 2, we compare the boundaries predicted by the RMS and mean approximations.

Table 2: A comparison between the decision boundaries obtained by approximating $s^{l-1}_{k^{l-1}_j}$ by its RMS and mean value. The RMS approximation underestimates the length boundary for RAI, whereas it overestimates both the phase boundaries for RAAI. On the other hand, the mean approximation overestimates the phase boundaries for RAI and RAAI both.

Approximation	$(\sigma^2_w)_q(RAI)$	$(\sigma^2_w)_c(RAAI)$	$(\sigma^2_w)_q(RAAI)$
RMS	0.57	1.41	1.75
Mean	0.85	1.46	1.89

F Comparison of the performance of different initialization schemes on real-world datasets

In this section, we compare the performance of different initializiation schemes on three different real-world datasets. We consider MNIST, Fashion-MNIST and CIFAR-10 datasets, and train them with feedforward networks with depth $L = 10$ (in addition to input layer) and a constant width of $N = 100$ for all hidden layers. We implemented feedforward networks in Tensorflow and train them using the complete training set with a cross entropy loss, a mini-batch size of 32, and default parameters for the optimizers. As high performance for these datasets is achieved quickly (in terms of epochs), we observe the training accuracy as a function of the number of steps (and not epochs). It is noteworthy that we used training accuracy for demonstration purposes and validation accuracy shows similar behaviour for the trends.

F.1 MNIST task

Figure 12 shows the average training accuracy for the MNIST task trained with SGD. We observe that RAAI performs better than all other schemes, however, the advantage is only observed at early time steps.

F.2 Fashion-MNIST task

Figure 13 shows the average training accuracy for the MNIST task trained with SGD. We observe that RAAI performs better than all other schemes, however, the advantage is only observed at early time steps.

F.3 CIFAR-10

Figure 14 shows the average training accuracy for the CIFAR-10 task trained with SGD. We observe that RAAI performs better than all other schemes, however, the advantage is only observed at early time steps.

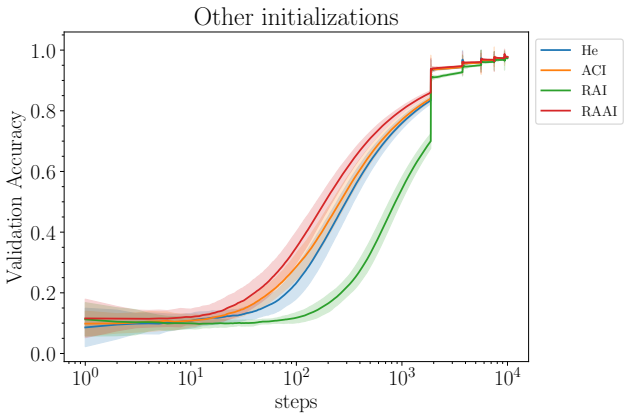

Figure 12: Average training accuracy for ReLU networks trained on the MNIST task with SGD optimizer for different initialization schemes.

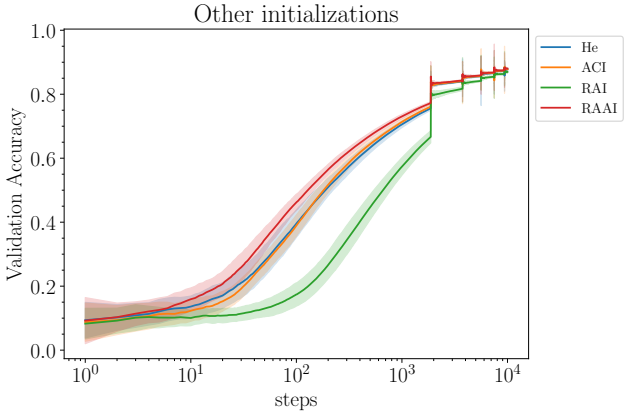

Figure 13: Average training accuracy for ReLU networks trained on the Fashion-MNIST task with SGD optimizer for different initialization schemes.

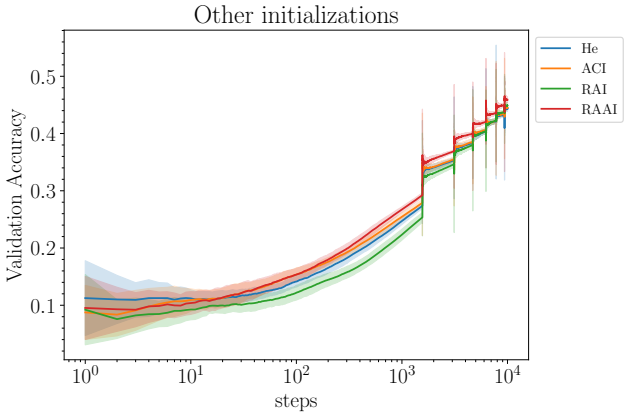

Figure 14: Average training accuracy for ReLU networks trained on the CIFAR-10 task with SGD optimizer for different initialization schemes.

G CODE TO GENERATE WEIGHTS DRAWN FROM RAAI DISTRIBUTION

```
import numpy as np

def RAAI(fan_in, fan_out, k = 100, variance_weights = 0.9):
    """Randomized Asymmetric Anti-correlated Initializer (RAAI)
    Arguments:
    fan_in -- the number of neurons in the previous layer
    fan_out -- the number of neurons in the next layer
    corr -- correlation strength for the Gaussian weights
    variance_weights -- variance of the weights
    Returns:
    W, b -- weight and bias matrices with shape(fan_in, fan_out), and (fan_out, )
    """
    corr = k/(1+k)
    mean = np.zeros(fan_in + 1)
    J = np.ones((fan_in + 1, fan_in + 1))
    cov = (np.identity(fan_in + 1) - J*(corr/(fan_in +1)) )* variance_weights/fan_in
    P = np.random.multivariate_normal(mean = mean, cov = cov, size = (fan_out))
    for j in range(P.shape[0]):
        k = np.random.randint(0, high = fan_in + 1)
        P[j, k] = np.random.beta(2, 1)
    W = P[:, :-1].T
    b = P[:, -1]
    return  W.astype(np.float32), b.astype(np.float32)
```