
Dynamical Isometry and a Mean Field Theory of RNNs: Gating Enables Signal Propagation in Recurrent Neural Networks

Minmin Chen^{*1} Jeffrey Pennington^{*2} Samuel S. Schoenholz^{*2}

Abstract

Recurrent neural networks have gained widespread use in modeling sequence data across various domains. While many successful recurrent architectures employ a notion of gating, the exact mechanism that enables such remarkable performance is not well understood. We develop a theory for signal propagation in recurrent networks after random initialization using a combination of mean field theory and random matrix theory. To simplify our discussion, we introduce a new RNN cell with a simple gating mechanism that we call the minimalRNN and compare it with vanilla RNNs. Our theory allows us to define a maximum timescale over which RNNs can remember an input. We show that this theory predicts trainability for both recurrent architectures. We show that gated recurrent networks feature a much broader, more robust, trainable region than vanilla RNNs, which corroborates recent experimental findings. Finally, we develop a closed-form critical initialization scheme that achieves dynamical isometry in both vanilla RNNs and minimalRNNs. We show that this results in significantly improved training dynamics. Finally, we demonstrate that the minimalRNN achieves comparable performance to its more complex counterparts, such as LSTMs or GRUs, on a language modeling task.

1. Introduction

Recurrent Neural Networks (RNNs) (Rumelhart et al., 1986; Elman, 1990) have found widespread use across a variety of domains from language modeling (Mikolov et al., 2010; Kiros et al., 2015; Jozefowicz et al., 2016) and machine translation (Bahdanau et al., 2014) to speech recogni-

tion (Graves et al., 2013) and recommendation systems (Hidasi et al., 2015; Wu et al., 2017). However, RNNs as originally proposed are difficult to train and are rarely used in practice. Instead, variants of RNNs - such as Long Short-Term Memory (LSTM) networks (Hochreiter & Schmidhuber, 1997) and Gated Recurrent Units (GRU) (Chung et al., 2014) - that feature various forms of “gating” perform significantly better than their vanilla counterparts. Often, these models must be paired with techniques such as normalization layers (Ioffe & Szegedy, 2015b; Ba et al., 2016) and gradient clipping (Pascanu et al., 2013) to achieve good performance.

A rigorous explanation for the remarkable success of gated recurrent networks remains illusive (Jozefowicz et al., 2015; Greff et al., 2017). Recent work (Collins et al., 2016) provides empirical evidence that the benefits of gating are mostly rooted in improved trainability rather than increased capacity or expressivity. The problem of disentangling trainability from expressivity is widespread in machine learning since state-of-the-art architectures are nearly always the result of sparse searches in high dimensional spaces of hyperparameters. As a result, we often mistake trainability for expressivity. Seminal early work (Glorot & Bengio; Bertschinger et al.) showed that a major hindrance to trainability was the vanishing and exploding of gradients.

Recently, progress has been made in the feed-forward setting (Schoenholz et al., 2017; Pennington et al., 2017; Yang & Schoenholz, 2017) by developing a theory of both the forward-propagation of signal and the backward-propagation of gradients. This theory is based on studying neural networks whose weights and biases are randomly distributed. This is equivalent to studying the behavior of neural networks after random initialization or, equivalently, to studying the prior over functions induced by a particular choice of hyperparameters (Lee et al., 2017). It was shown that randomly initialized neural networks are trainable if three conditions are satisfied: (1) the size of the output of the network is finite for finite inputs, (2) the output of the network is sensitive to changes in the input, and (3) gradients neither explode nor vanish. Moreover, neural networks achieving dynamical isometry, i.e. having input-output Jacobian matrices that are well-conditioned, were shown to

^{*}Equal contribution ¹Google ²Google Brain. Correspondence to: Minmin Chen <minminc@google.com>.

train orders of magnitude faster than networks that do not.

In this work, we combine mean field theory and random matrix theory to extend these results to the recurrent setting. We will be particularly focused on understanding the role that gating plays in trainability. As we will see, there are a number of subtleties that must be addressed for (gated) recurrent networks that were not present in the feed-forward setting. To clarify the discussion, we will therefore contrast vanilla RNNs with a gated RNN cell, that we call the minimalRNN, which is significantly simpler than LSTMs and GRUs but implements a similar form of gating. We expect the framework introduced here to be applicable to more complicated gated architectures.

The first main contribution of this paper is the development of a mean field theory for forward propagation of signal through vanilla RNNs and minimalRNNs. In doing so, we identify a theory of the maximum timescale over which signal can propagate in each case. Next, we produce a random matrix theory for the end-to-end Jacobian of the minimal-RNN. As in the feed-forward setting, we establish that the duality between the forward propagation of signal and the backward propagation of gradients persists in the recurrent setting. We then show that our theory is indeed predictive of trainability in recurrent neural networks by comparing the maximum trainable number of steps of RNNs with the timescale predicted by the theory. Overall, we find remarkable alignment between theory and practice. Additionally, we develop a closed-form initialization procedure for both networks and show that on a variety of tasks RNNs initialized to be dynamically isometric are significantly easier to train than those lacking this property.

Corroborating the experimental findings of Collins et al. (2016), we show that both signal propagation and dynamical isometry in vanilla RNNs is far more precarious than in the case of the minimalRNN. Indeed the vanilla RNN achieves dynamical isometry only if the network is initialized with orthogonal weights at the boundary between order-and-chaos, a one-dimensional line in parameter space. Owing to its gating mechanism, the minimalRNN on the other hand enjoys a robust multi-dimensional subspace of good initializations which all enable dynamical isometry. Based on these insights, we conjecture that more complex gated recurrent neural networks also benefit from the similar effects.

2. Related Work

Identity and Orthogonal initialization schemes have been identified as a promising approach to improve trainability of deep neural networks (Le et al., 2015; Mishkin & Matas, 2015). Additionally, Arjovsky et al. (2016); Hyland & Rätsch (2017); Xie et al. (2017) advocate going beyond initialization to constrain the transition matrix to be orthog-

onal throughout the entire learning process either through re-parametrisation or by constraining the optimization to the Stiefel manifold (Wisdom et al., 2016). However, as was pointed out in Vorontsov et al. (2017), strictly enforcing orthogonality during training may hinder training speed and generalization performance. While these contributions are similar to our own, in the sense that they attempt to construct networks that feature dynamical isometry, it is worth noting that orthogonal weight matrices do not guarantee dynamical isometry. This is due to the nonlinear nature of deep neural networks as shown in Pennington et al. (2017). In this paper we continue this trend and show that orthogonality has little impact on the conditioning of the Jacobian (and so trainability) in gated RNNs.

The notion of “edge of chaos” initialization has been explored previously especially in the case of recurrent neural networks. Bertschinger et al.; Glorot & Bengio propose edge-of-chaos initialization schemes that they show leads to improved performance. Additionally, architectural innovations such as batch normalization (Ioffe & Szegedy, 2015a), orthogonal matrix initialization (Saxe et al., 2013), random walk initialization (Sussillo & Abbott, 2014), composition kernels (Daniely et al., 2016), or residual network architectures (He et al., 2015) all share a common goal of stabilizing gradients and improving training dynamics.

There is a long history of applying mean field-like approaches to understand the behavior of neural networks. Indeed several pieces of seminal work used statistical physics (Derrida & Pomeau; Sompolinsky et al., 1988) and Gaussian Processes (Neal, 2012) to show that neural networks exhibit remarkable regularity as the width of the network gets large. Mean field theory also has long been used to study Boltzmann machines (Ackley et al.) and sigmoid belief networks (Saul et al., 1996). More recently, there has been a revitalization of mean field theory to explore questions of trainability and expressivity in fully-connected networks and residual networks (Poole et al., 2016; Schoenholz et al., 2017; Yang & Schoenholz, 2017; Schoenholz et al., 2017; Karakida et al., 2018; Hayou et al., 2018; Hanin & Rolnick, 2018; Yang & Schoenholz, 2018). Our approach will closely follow these later contributions and extend many of their techniques to the case of recurrent networks with gating. Beyond mean field theory, there have been several attempts in understanding signal propagation in RNNs, e.g., using Geršgorin circle theorem (Zilly et al., 2016) or time invariance (Tallec & Ollivier, 2018).

3. Theory and Critical Initialization

We begin by developing a mean field theory for vanilla RNNs and discuss the notion of dynamical isometry. Afterwards, we move on to a simple gated architecture to explain the role of gating in facilitating signal propagation in RNNs.

3.1. Vanilla RNN

Vanilla RNNs are described by the recurrence relation,

$$\mathbf{e}^t = \mathbf{W}\mathbf{h}^{t-1} + \mathbf{V}\mathbf{x}^t + \mathbf{b} \quad \mathbf{h}^t = \phi(\mathbf{e}^t). \quad (1)$$

Here $\mathbf{x}^t \in \mathbb{R}^M$ is the input, $\mathbf{e}^t \in \mathbb{R}^N$ is the pre-activation, and $\mathbf{h}^t \in \mathbb{R}^N$ is the hidden state after applying an arbitrary activation function $\phi : \mathbb{R} \rightarrow \mathbb{R}$. For the purposes of this discussion we set $\phi = \tanh$. Furthermore, $\mathbf{W} \in \mathbb{R}^{N \times N}$ and $\mathbf{V} \in \mathbb{R}^{N \times M}$ are weight matrices that multiply the hidden state and inputs respectively and $\mathbf{b} \in \mathbb{R}^N$ is a bias.

Next, we apply mean-field theory to vanilla RNNs following a similar strategy introduced in (Poole et al., 2016; Schoenholz et al., 2017). At the level of mean-field theory, vanilla RNNs will prove to be intimately related to feed-forward networks and so this discussion proceeds analogously. For a more detailed discussion, see these earlier studies.

Consider two sequences of inputs $\{\mathbf{x}_1^t\}$ and $\{\mathbf{x}_2^t\}$, described by the covariance matrix $\mathbf{R}^t \in \mathbb{R}^{2 \times 2}$ with $\mathbf{R}_{ab}^t = \frac{1}{M}\mathbb{E}[\mathbf{x}_a \cdot \mathbf{x}_b]$, $a, b \in \{1, 2\}$. To simplify notation, we assume the input sequences have been standardized so that $R_{11}^t = R_{22}^t = R$ independent of time. This allows us to write $\mathbf{R}^t = R\boldsymbol{\Sigma}^t$, where $\boldsymbol{\Sigma}^t$ is a matrix whose diagonal terms are 1 and whose off-diagonal terms are the cosine similarity between the inputs at time t . These sequences are then passed into two identical copies of an RNN to produce two corresponding pre-activation sequences $\{e_1^t\}$ and $\{e_2^t\}$. As in Poole et al. (2016) we let the weights and biases be Gaussian distributed so that $W_{ij} \sim \mathcal{N}(0, \sigma_w^2/N)$, $V_{ij} \sim \mathcal{N}(0, \sigma_v^2/M)$, and $b_i \sim \mathcal{N}(\mu_b, \sigma_b^2)$ ¹, and we consider the wide network limit, $N \rightarrow \infty$. As in the fully-connected setting, we would like to invoke the Central Limit Theorem (CLT) to conclude that the pre-activations of hidden states are jointly Gaussian distributed. Unfortunately, the CLT is violated in the recurrent setting as \mathbf{h}^{t-1} is correlated with \mathbf{W} due to weight sharing between steps of the RNN.

To make progress, we proceed by developing the theory of signal propagation for RNNs with *untied* weights. This allows for several simplifications, including the application of the CLT to conclude that e_{ia}^t are jointly Gaussian distributed,

$$[e_{i1}^t, e_{i2}^t]^T \sim \mathcal{N}(\mu_b \mathbf{1}, \mathbf{q}^t \delta_{ij}), \quad i, j \in \{1, \dots, N\}$$

where the covariance matrix $\mathbf{q}^t \in \mathbb{R}^{2 \times 2}$ is independent of neuron index, i . We explore the ramifications of this approximation by comparing simulations of RNNs with tied and untied weights. Overall, we will see that while ignoring weight tying leads to quantitative differences between theory and experiment, it does not change the qualitative picture that emerges. See figs. 1 and 2 for verification.

With this approximation in mind, we will now quantify how the pre-activation hidden states $\{e_1^t\}$ and $\{e_2^t\}$ evolve by

deriving the recurrence relation of the covariance matrix \mathbf{q}^t from the recurrence on e^t in eq. (1). Using identical arguments to Poole et al. (2016) one can show that,

$$\mathbf{q}^t = \sigma_w^2 \int D_{\mathbf{q}^{t-1}} \mathbf{z} \phi(\mathbf{z}) \phi(\mathbf{z})^\top + \sigma_v^2 R \boldsymbol{\Sigma}^t + \sigma_b^2 \mathbf{I}. \quad (2)$$

where $\mathbf{z} = [z_1, z_2]^\top$, and

$$\int D_{\mathbf{q}} \mathbf{z} = \frac{1}{2\pi\sqrt{|\mathbf{q}|}} \int d\mathbf{z} e^{-(\mathbf{z}-\mu_b \mathbf{1})^\top \mathbf{q}^{-1} (\mathbf{z}-\mu_b \mathbf{1})} \quad (3)$$

is a Gaussian measure with covariance matrix \mathbf{q} . By symmetry, our normalization allows us to define $q_{11}^t = q_{22}^t = q^t$ to be the magnitude of the pre-activation hidden state and $c^t = q_{12}^t/q^t$ to be the cosine similarity between the hidden states. We will be particularly concerned with understanding the dynamics of the cosine similarity, c^t .

In feed-forward networks, the inputs dictate the initial value of the cosine similarity, c^0 and then the evolution of c^t is determined solely by the network architecture. By contrast in recurrent networks, inputs perturb c^t at each timestep. Analyzing the dynamics of c^t for arbitrary $\boldsymbol{\Sigma}^t$ is therefore challenging, however significant insight can be gained by studying the off-diagonal entries of eq. (2) for $\boldsymbol{\Sigma}^t = \boldsymbol{\Sigma}$ independent of time. In the case of time-independent $\boldsymbol{\Sigma}^t$, as $t \rightarrow \infty$ both $q^t \rightarrow q^*$ and $c^t \rightarrow c^*$ where q^* and c^* are fixed points of the variance of the pre-activation hidden state and the cosine-similarity between pre-activation hidden states respectively. As was discussed previously (Poole et al., 2016; Schoenholz et al., 2017), the dynamics of q^t are generally uninteresting provided q^* is finite. We therefore choose to normalize the hidden state such that $q^0 = q^*$ which implies that $q^t = q^*$ independent of time.

In this setting it was shown in Schoenholz et al. (2017) that in the vicinity of a fixed point, the off-diagonal term in eq. (2) can be expanded to lowest order in $\epsilon^t = c^t - c^*$ to give the linearized dynamics, $\epsilon^t = \chi_{c^*} \epsilon^{t-1}$ where

$$\chi_{c^*} = \sigma_w^2 \int D_{\mathbf{q}^*} \mathbf{z} \phi'(z_1) \phi'(z_2). \quad (4)$$

These dynamics have the solution $\epsilon^t = \chi_{c^*}^{t-t_0} \epsilon^{t_0}$ where t_0 is the time when c^t is close enough to c^* for the linear approximation to be valid. If $\chi_{c^*} < 1$ it follows that c^t approaches c^* exponentially quickly over a timescale $\tau = -1/\log \chi_{c^*}$ and c^* is called a stable fixed point. When c^t gets too close to c^* to be distinguished from it to within numerical precision, information about the initial inputs has been lost. Thus, τ sets the maximum timescale over which we expect the RNN to be able to remember information. If $\chi_{c^*} > 1$ then c^t gets exponentially farther from c^* over time and c^* is an unstable fixed point. In this case, for the activation function considered here, another fixed point that

¹in practice we will set $\mu_b = 0$ for vanillaRNN.

is stable will emerge. Note that χ_{c^*} is independent of Σ and so the dynamics of c^t near c^* do not depend on Σ .

In vanilla fully-connected networks $c^* = 1$ is always a fixed point of c^t , but it is not always stable. Indeed, it was shown that these networks exhibit a phase transition where $c^* = 1$ goes from being a stable fixed point to an unstable one as a function of the network’s hyperparameters. This is known as the order-to-chaos transition and it occurs exactly when $\chi_1 = 1$. Since $\tau = -1/\log(\chi_1)$, signal can propagate infinitely far at the boundary between order and chaos. Comparing the diagonal and off-diagonal entries of eq. (2), we see that in recurrent networks, $c^* = 1$ is a fixed point only when $\Sigma_{12} = 1$, and in this case the discussion is identical to the feed-forward setting. When $\Sigma_{12} < 1$, it is easy to see that $c^* < 1$ since if $c^t = 1$ at some time t then $c^{t+1} = 1 - \sigma_v^2 R(1 - \Sigma_{12})/q^* < 1$. We see that in recurrent networks noise from the inputs destroys the ordered phase and there is no ordered-to-chaos critical point. As a result we should expect the maximum timescale over which memory may be stored in vanilla RNNs to be fundamentally limited by noise from the inputs.

The end-to-end Jacobian of a vanilla RNN with untied weights is in fact formally identical to the input-output Jacobian of a feedforward network, and thus the results from (Pennington et al., 2017) regarding conditions for dynamical isometry apply directly. In particular, dynamical isometry is achieved with orthogonal state-to-state transition matrices \mathbf{W} , tanh non-linearities, and small values of q^* . Perhaps surprisingly, these conclusions continue to be valid if the assumption of untied weights is relaxed. To understand why this is the case, consider the example of a linear network. For untied weights, the end-to-end Jacobian is given by $\hat{\mathbf{J}} = \prod_{t=1}^T \mathbf{W}_t$, while for tied weights the Jacobian is given by $\mathbf{J} = \mathbf{W}^T$. It turns out that as $N \rightarrow \infty$ there is sufficient self-averaging to overcome the dependencies induced by weight tying and the asymptotic singular value distributions of $\hat{\mathbf{J}}$ and \mathbf{J} are actually identical (Haagerup & Larsen, 2000).

3.2. MinimalRNN

3.2.1. MEAN-FIELD THEORY

To study the role of gating, we introduce the minimalRNN which is simpler than other gated RNN architectures but nonetheless features the same gating mechanism. A sequence of inputs $\mathbf{x}^t \in \mathbb{R}^M$, is first mapped to the hidden space through $\tilde{\mathbf{x}}^t = \Phi(\mathbf{x}^t)^2$. From here on, we refer to $\tilde{\mathbf{x}}^t \in \mathbb{R}^N$ as the inputs to minimalRNN. The minimalRNN

² $\Phi(\cdot)$ here can be any highly flexible functions such as a feed-forward network. In our experiments, we take $\Phi(\cdot)$ to be a fully connected layer with tanh activation, that is, $\Phi(\mathbf{x}^t) = \tanh(\mathbf{W}_x \mathbf{x}^t)$.

is then described by the recurrence relation,

$$\begin{aligned} \mathbf{e}^t &= \mathbf{W} \mathbf{h}^{t-1} + \mathbf{V} \tilde{\mathbf{x}}^t + \mathbf{b} & \mathbf{u}^t &= \sigma(\mathbf{e}^t) \\ \mathbf{h}^t &= \mathbf{u}^t \odot \mathbf{h}^{t-1} + (\mathbf{1} - \mathbf{u}^t) \odot \tilde{\mathbf{x}}^t \end{aligned} \quad (5)$$

where $\mathbf{e}^t \in \mathbb{R}^N$ is the pre-activation to the gating function, $\mathbf{u}^t \in \mathbb{R}^N$ the update gate and $\mathbf{h}^t \in \mathbb{R}^N$ the hidden state. The minimalRNN retains the most essential gate in LSTMs (Jozefowicz et al., 2015; Greff et al., 2017) and achieves competitive performance. The simplified update of this cell on the other hand, enables us to pinpoint the role of gating in a more controlled setting.

As in the previous case we consider two sequences of inputs to the network, $\{\tilde{\mathbf{x}}_1^t\}$ and $\{\tilde{\mathbf{x}}_2^t\}$. We take $W_{ij} \sim \mathcal{N}(0, \sigma_w^2/N)$, $V_{ij} \sim \mathcal{N}(0, \sigma_v^2/N)$ and $b_i \sim \mathcal{N}(\mu_b, \sigma_b^2)$. By analogy to the vanilla case, we can make the mean field approximation that the e_{ia}^t are jointly Gaussian distributed with covariance matrix $q^t \delta_{ij} \in \mathbb{R}^{2 \times 2}$. Here,

$$\mathbf{q}^t = \sigma_w^2 \mathbf{Q}^{t-1} + \sigma_v^2 \mathbf{R}^t + \sigma_b^2 \mathbf{I} \quad (6)$$

where we have defined \mathbf{Q}^t as the second-moment matrix with $Q_{ab}^t = \mathbb{E}[h_{ia}^t h_{ib}^t]$.³ As in the vanilla case, \mathbf{R}^t is the covariance between inputs so that $R_{ab}^t = \frac{1}{N} \mathbb{E}[\tilde{\mathbf{x}}_a \cdot \tilde{\mathbf{x}}_b]$.

We note that \mathbf{R}^t is fixed by the input, but it remains for us to work out \mathbf{Q}^t . We find that (see SI section B),

$$\begin{aligned} \mathbf{Q}^t &= \mathbf{Q}^{t-1} \int D_{q^t} \mathbf{z} \sigma(\mathbf{z}) \sigma(\mathbf{z})^\top \\ &+ \mathbf{R}^t \int D_{q^t} \mathbf{z} (1 - \sigma(\mathbf{z})) (1 - \sigma(\mathbf{z}))^\top \end{aligned} \quad (7)$$

Here we assume that the expectation factorizes so that \mathbf{h}^{t-1} and \mathbf{u}^t are approximately independent. We believe this approximation becomes exact in the $N \rightarrow \infty$ limit.

We choose to normalize the data in a similar manner to the vanilla case so that $R_{11}^t = R_{22}^t = R$ independent of time. An immediate consequence of this normalization is that $Q_{11}^t = Q_{22}^t = Q^t$ and $q_{11}^t = q_{22}^t = q^t$. We then write $C^t = Q_{12}^t/Q^t$ and $c^t = q_{12}^t/q^t$ as the cosine similarities between the hidden states and the pre-activations respectively. With this normalization, we can work out the mean-field recurrence relation characterizing the covariance matrix for the minimalRNN. This analysis can be done by deriving the recurrence relation for either \mathbf{Q}^t or q^t . We will choose to study the dynamics of q^t , however the two are trivially related by eq. (6). In SI section C, we analyze the dynamics of the diagonal term in the recurrence relation and prove that there is always a fixed point at some q^* . In SI section D, we compute the depth scale over which q^t approaches q^* . However, as in the case of the vanillaRNN, the dynamics of q^* are generally uninteresting.

³ \mathbf{h}^t will be centered under mean field approximation if \mathbf{h}^0 is initialized with mean zero.

We now turn our attention to the dynamics of the cosine similarity between the pre-activations, c^t . As in the case of vanilla RNNs, we note that q^t approaches q^* quickly relative to the dynamics of c^t . We therefore choose to normalize the hidden state of the RNN so that $Q^0 = Q^*$ in which case both $Q^t = Q^*$ and $q^t = q^*$ independent of time. From eq. (6) and (7) it follows that the cosine similarity of the pre-activation evolves as,

$$c^t = [c^{t-1} + (\sigma_w^2 - \sigma_v^2)\rho^{t-1}] \int D_{q^{t-1}} z \sigma(z_1)\sigma(z_2) - 2\sigma_w^2\rho^{t-1} \int D_{q^{t-1}} z \sigma(z_1) + \sigma_w^2\rho^{t-1} + \sigma_v^2\rho^t \quad (8)$$

where we have defined $\rho^t = R\Sigma_{12}^t/q^*$. As in the case of the vanilla RNN, we can study the behavior of c^t in the vicinity of a fixed point, c^* . By expanding eq. (8) to lowest order in $\epsilon^t = c^* - c^t$ we arrive at a linearized recurrence relation that has an exponential solution $\epsilon^{t+1} = \chi_{c^*}\epsilon^t$ where here,

$$\chi_{c^*} = \int D_{q^*} z \sigma(z_1)\sigma(z_2) + (q^*c^* + (\sigma_w^2 - \sigma_v^2)R\Sigma_{12}) \int D_{q^*} z \sigma'(z_1)\sigma'(z_2). \quad (9)$$

The discussion above in the vanilla case carries over directly to the minimalRNN with the appropriate replacement of χ_{c^*} . Unlike in the case of the vanilla RNN, here we see that χ_{c^*} itself depends on Σ_{12} .

Again $c^* = 1$ is a fixed point of the dynamics only when $\Sigma_{12} = 1$. In this case, the minimalRNN experiences an order-to-chaos phase transition when $\chi_1 = 1$ at which point the maximum timescale over which signal can propagate goes to infinity. Similar to the vanilla RNN, when $\Sigma_{12} < 1$, we expect that the phase transition will be destroyed and the maximum duration of signal propagation will be severely limited. However, in a significant departure from the vanilla case, when $\mu_b \rightarrow \infty$ we notice that $\sigma(z + \mu_b) \rightarrow 1$, and $\sigma'(z + \mu_b) \rightarrow 0$ for all z . Considering eq. (9) we notice that in this regime $\chi_{c^*} \rightarrow 1$ independent of Σ_{12} . In other words, gating allows for arbitrarily long term signal propagation in recurrent neural networks independent of Σ_{12} .

We explore agreement between our theory and MC simulations of the minimalRNN in fig. 1. In this set of experiments, we consider inputs such that $\Sigma_{12}^t = 0$ for $t < 10$ and $\Sigma_{12}^t = 1$ for $t \geq 10$. Fig. 1 (a,c,d) show excellent quantitative agreement between our theory and MC simulations. In fig. 1 (a,b) we compare the MC simulations of the minimalRNN with and without weight tying. While we observe that for many choices of hyperparameters the untied weight approximation is quite good (particularly when $c^* \approx 1$), deeper into the chaotic phase the quantitative agreement between breaks down. Nonetheless, we observe that the untied approximation describes the qualitative behavior of the real

minimalRNN overall. In fig. 1 (e) we plot the timescale for signal propagation for $\Sigma_{12} = 1, 0.99$, and 0 for the minimalRNN with identical choices of hyperparameters. We see that while $\tau \rightarrow \infty$ as μ_b gets large independent of Σ_{12} , a critical point at $\mu_b = 0$ is only observed when $\Sigma_{12} = 1$.

3.2.2. DYNAMICAL ISOMETRY

In the previous subsection, we derived a quantity χ_1 that defines the boundary between the ordered and the chaotic phases of forward propagation. Here we show that it also defines the boundary between exploding and vanishing gradients. To see this, consider the Jacobian of the state-to-state transition operator,

$$\mathbf{J}_t = \frac{\partial \mathbf{h}^{t+1}}{\partial \mathbf{h}^t} = \mathbf{D}_{\mathbf{u}^t} + \mathbf{D}_{\sigma'(e^t) \odot (\mathbf{h}^{t-1} - \mathbf{z}^t)} \mathbf{W}, \quad (10)$$

where $\mathbf{D}_{\mathbf{x}}$ denotes a diagonal matrix with \mathbf{x} along its diagonal. We can compute the expected norm-squared of back-propagated error signals, which measures the growth or shrinkage of gradients. It is equal to the mean-squared singular value of the Jacobian (Poole et al., 2016; Schoenholz et al., 2017) or the first moment of $\mathbf{J}_t \mathbf{J}_t^T$,

$$\frac{1}{N} \mathbb{E}[\text{tr}(\mathbf{J}_t \mathbf{J}_t^T)] = \mathbb{E}[(u_1^t)^2] + \sigma_w^2 \mathbb{E}[\sigma'(e_1^t)^2 (h_1^{t-1} - z_1^t)^2], \quad (11)$$

where we have used the fact that the elements of \mathbf{u}^t , \mathbf{h}^t and \mathbf{z}^t are i.i.d. Since we assume convergence to the fixed point, these distributions are independent of t and it is easy to see that $\frac{1}{N} \mathbb{E}[\text{tr}(\mathbf{J}_t \mathbf{J}_t^T)] = \chi_1$. The variance of back-propagated error signals through T time steps is therefore $\frac{1}{N} \mathbb{E}[\text{tr}(\mathbf{J} \mathbf{J}^T)] = \chi_1^T$. As such, the constraint $\chi_1 = 1$ defines the boundary between phases of exponentially exploding and exponentially vanishing gradient norm (variance). Note that unlike in the case of forward signal propagation, in the case of backpropagation this is independent of Σ .

As argued in (Pennington et al., 2017; 2018), controlling the variance of back-propagated gradients is necessary but not sufficient to guarantee trainability, especially for very deep networks. Beyond the first moment, the entire distribution of eigenvalues of $\mathbf{J} \mathbf{J}^T$ (or of singular values of \mathbf{J}) is relevant. Indeed, it was found in (Pennington et al., 2017; 2018) that enabling dynamical isometry, namely the condition that all singular values of \mathbf{J} are close to unity, can drastically improve training speed for very deep feed-forward networks.

Following (Pennington et al., 2017; 2018), we use tools from free probability theory to compute the variance $\sigma_{\mathbf{J} \mathbf{J}^T}^2$ of the limiting spectral density of $\mathbf{J} \mathbf{J}^T$; however, unlike previous work, in our case the relevant matrices are not symmetric and therefore we must invoke tools from non-Hermitian free probability, see (Cakmak, 2012) for a review. As in previous section, we make the simplifying assumption that the weights are untied, relying on the same motivations

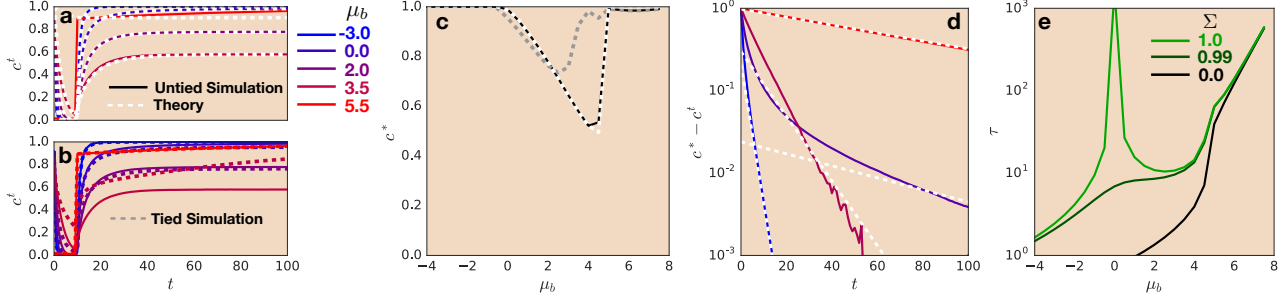


Figure 1. Numerical verification of mean field results on the minimalRNN. All experiments were done with fixed $\sigma_w = 6.88$, $\sigma_v = 1.39$, $R = 0.46$, and $\mu_b \in [-4, 8]$. These hyperparameters are chosen so that the minimalRNN has an order-to-chaos critical point at $\mu_b = 0$. MC simulations were averaged over 1000 minimalRNNs with hidden dimension 8192. (a) Cosine similarity for theory (white dashed) compared with MC simulations of a minimalRNN with untied weights (solid). (b) Cosine similarity for tied weights (dashed) compared with untied weights (solid). (c) Comparison of c^* for theory (white dashed), MC with untied weights (solid black), tied weights (dashed grey). (d) Comparison of linearized dynamics. Dashed lines show $Ae^{-t/\tau}$ and solid lines show the result of simulations with untied weights. (e) Comparison of the timescale for signal propagation for different values of Σ_{12} .

given in section 3.1. Using these tools, an un-illuminating calculation reveals that,

$$\sigma_{JJ^T}^2 = \chi_1^{2T} \left(1 + T \frac{2(\mu_1 - s_1)\mu_2 + \sigma_1^2 + \sigma_2^2}{\chi_1^2} \right), \quad (12)$$

where,

$$\chi_1 = \mu_1 + \mu_2 \quad (13)$$

$$\mu_1 = \int \mathcal{D}z \sigma^2(\sqrt{q^*}z + \mu_b)$$

$$\sigma_1^2 = -\mu_1^2 + \int \mathcal{D}z \sigma^4(\sqrt{q^*}z + \mu_b)$$

$$\mu_2 = \sigma_w^2(Q^* + R) \int \mathcal{D}z [\sigma'(\sqrt{q^*}z + \mu_b)]^2$$

$$\sigma_2^2 = -\mu_2^2 + \sigma_w^4((Q^*)^2 + R^2) \int \mathcal{D}z [\sigma'(\sqrt{q^*}z + \mu_b)]^4$$

and s_1 is the first term in the Taylor expansion of the S-transform of the eigenvalue distribution of WW^T (Pennington et al., 2018). For example, for Gaussian matrices, $s_1 = -1$ and for orthogonal matrices $s_1 = 0$.

Some remarks are in order about eq. (12). First, we note the duality between the forward and backward signal propagation (eq. (9) and eq. (13)). For critical initializations, $\chi_1 = 1$, so $\sigma_{JJ^T}^2$ does not grow exponentially, but it still grows linearly with T . This situation is entirely analogous to the feed-forward analysis of (Pennington et al., 2017; 2018). In the case of the vanilla RNN, the coefficient of the linear term is proportion to q^* , and can only be reduced by taking the weight and bias variances $(\sigma_w^2, \sigma_b^2) \rightarrow (1, 0)$. A crucial difference in the minimalRNN is that the coefficient of the linear term can be made arbitrarily small by simply adjusting the bias mean μ_b to be positive, which will send $\mu_2 \rightarrow 0$ and $\mu_1 \rightarrow 1$ independent of Σ . Therefore the conditions for dynamical isometry decouple from the weight

and bias variances, implying that trainability can occur for a higher-dimensional, more robust, slice of parameter space. Moreover, the value of s_1 has no effect on the capacity of the minimalRNN to achieve dynamical isometry. We believe these are fundamental reasons why gated cells such as the minimalRNN perform well in practice.

Algorithm 1 describes the procedure to find σ_w^2 , σ_v^2 and σ_b^2 to achieve χ_1 condition for minimalRNN. Given σ_w^2 , σ_v^2 , σ_b^2 , we then construct the weight matrices and biases accordingly. Q^* is used to initialize the h^0 to avoid transient phase.

Algorithm 1 Critical initialization for minimalRNNs

Require: q^*, μ_b, R

- 1: $\mathbb{E}[u^2] \leftarrow \int \mathcal{D}z \sigma^2(\sqrt{q^*}z + \mu_b)$
 - 2: $\mathbb{E}[(1-u)^2] \leftarrow \int \mathcal{D}z (1 - \sigma(\sqrt{q^*}z + \mu_b))^2$
 - 3: $Q^* \leftarrow R \cdot \mathbb{E}[(1-u)^2] / (1 - \mathbb{E}[u^2])$ (eq.(7))
 - 4: $\mathbb{E}[u'^2] \leftarrow \int \mathcal{D}z [\sigma'(\sqrt{q^*}z + \mu_b)]^2$
 - 5: $\sigma_w^2 \leftarrow (1 - \mathbb{E}[u^2]) / (Q^* + R) / \mathbb{E}[u'^2]$ (eq.(13))
 - 6: $\sigma_b^2 \leftarrow 0$
 - 7: $\sigma_v^2 \leftarrow (q^* - Q^* \sigma_w^2 - \sigma_b^2) / R$
-

4. Experiments

Having established a theory for the behavior of random vanilla RNNs and minimalRNNs, we now discuss the connection between our theory and trainability in practice. We begin by corroborating the claim that the maximum timescale over which memory can be stored in a RNN is controlled by the timescale τ identified in the previous section. We will then investigate the role of dynamical isometry in speeding up learning.

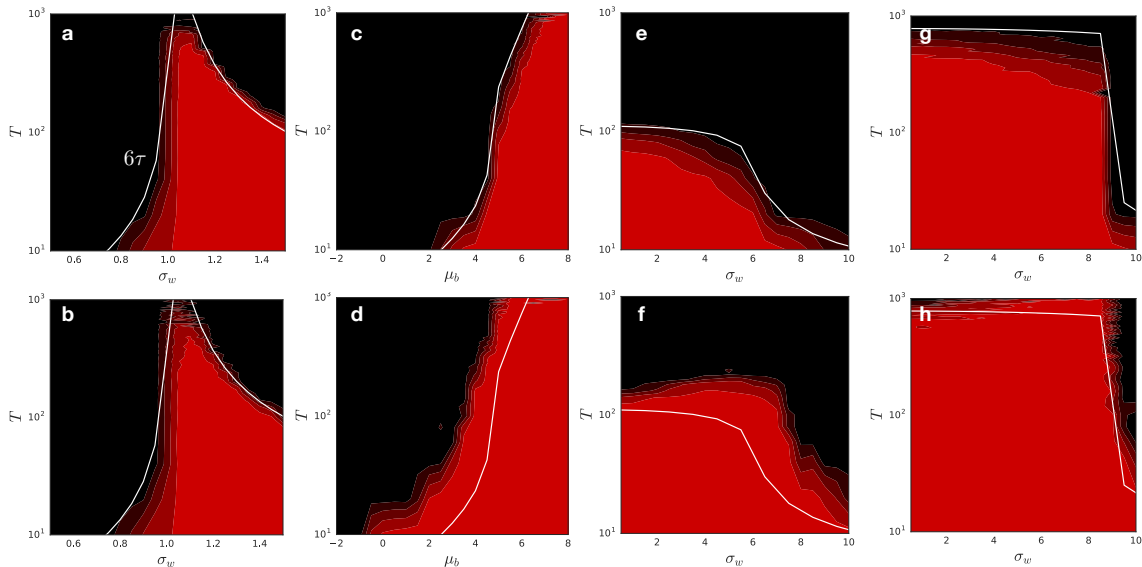


Figure 2. Relationship between theory and trainability. We plot the training accuracy (higher accuracies in red) overlaid with theoretical timescale, τ (shown in white). The top row of figures shows the results with untied weights and the bottom row shows the results with weight tying. (a-b) Vanilla RNN with $\sigma_w \in [0.5, 1.5]$. (c-d) MinimalRNN with $\mu_b \in [-4, 8]$. (e-f) MinimalRNN with $\sigma_w \in [0.5, 10]$ and $\mu_b = 4$. (g-h) MinimalRNN with $\sigma_w \in [0.5, 10]$ and $\mu_b = 6$.

4.1. Trainability

Dataset. To verify the results of our theoretical calculation, we consider a task that is reflective of the theory above. To that end, we constructed a sequence dataset for training RNNs from MNIST (LeCun et al., 1998). Each of the 28×28 digit image is flattened into a vector of 784 pixels and sent as the first input to a RNN. We then send T random inputs $x^t \sim \mathcal{N}(0, \sigma_x^2)$, $0 < t < T$ into the RNN varying T between 10 and 1000 steps. As the only salient information about the digit is in the first layer, the network will need to propagate information through T layers to accurately identify the MNIST digit. The random inputs are drawn independently for each example and so this is a regime where $\Sigma^t = 0$ for all $t > 0$.

We then performed a series of experiments on this task to make connection with our theory. In each case we experimented with both tied and untied weights. The result are shown in fig. 2. In the case of untied weights, we observe strong quantitative agreement between our theoretical prediction for τ and the maximum depth T where the network is still trainable. When the weights of the network are tied, we observe quantitative deviations between our theory and experiments, but the overall qualitative picture remains.

We train vanilla RNNs for 10^3 steps (around 10 epochs) varying $\sigma_w \in [0.5, 1.5]$ while fixing $\sigma_v = 0.025$. The results of this experiment are shown in fig. 2 (a-b). We train minimalRNNs for 10^2 steps (around 1 epoch) fixing $\sigma_v = 1.39$. We perform three different experiments here: 1)

varying $\mu_b \in [-4, 8]$ with $\sigma_w = 6.88$ shown in fig. 2 (c-d), 2) varying $\sigma_w \in [0.5, 10]$ with $\mu_b = 4$ shown in fig. 2 (e-f), 3) varying $\sigma_w \in [0.5, 10]$ with $\mu_b = 6$ shown in fig. 2 (g-h). Comparing fig. 2(a,b) with fig. 2(c,d, g,h), the minimalRNN with large depth T is trainable over a much wider range of hyperparameters than the vanillaRNN despite the fact that the network was trained for an order of magnitude less time.

4.2. Critical initialization

Dataset. To study the impact of critical initialization on training speed, we constructed a more realistic sequence dataset from MNIST. We unroll the pixels into a sequence of T inputs, each containing $784/T$ pixels. We tested $T = 196$ and $T = 784$ to vary the difficulty of the tasks.

Note that we are more interested in the training speed of these networks under different initialization conditions than the test accuracy. We compare the convergence speed of vanilla RNN and minimalRNN under four initialization conditions: 1) critical initialization with orthogonal weights (solid blue); 2) critical initialization with Gaussian distributed weights (solid red); 3) off-critical initialization with orthogonal weights (dotted green); 4) off-critical initialization with Gaussian distributed weights (dotted black).

We fix σ_b^2 to zero in all settings. Under critical initialization, σ_w^2 and σ_v^2 are carefully chosen to achieve $\chi_1 = 1$ as defined in eqn.(4) for vanilla RNN and eqn.(13) (detailed in algorithm 1) for minimalRNN respectively. When testing networks off criticality, we employ a common initialization

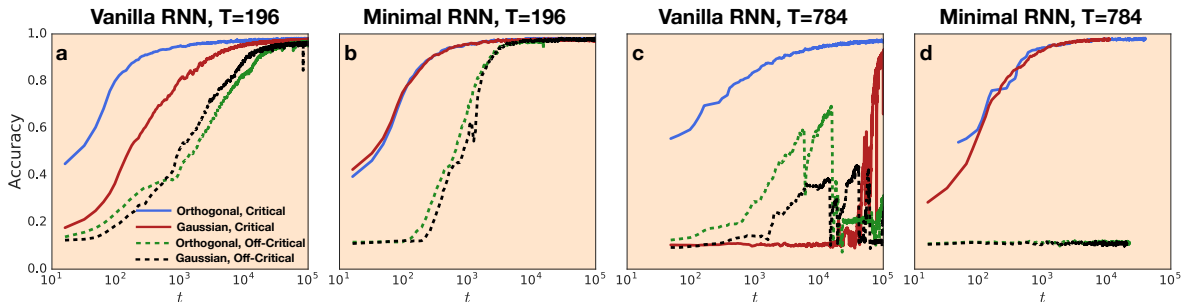


Figure 3. Learning dynamics, measured by accuracy on the test set, for vanillaRNNs and minimalRNNs trained with depth 196 (a, b) and 784 (c, d) under four different initialization conditions. Drastic difference in terms of convergence speed was observed between critical and off-critical initialization in both models. Well-trained models reach an accuracy of 0.98 on the test set.

procedure in which, $\sigma_w^2 = 1.0$ and $\sigma_v^2 = 1.0$.

Figure 3 summarizes our findings: there is a clear difference in training speed between models trained with critical initialization compared with models initialized far from criticality. We observe two orders of magnitude difference in training speed between a critical and off-critical initialization for vanilla RNNs. While a critically initialized model reaches a test accuracy of 90% after 750 optimization steps, the off-critical network takes over 16,000 updates. A similar trend was observed for the minimalRNN. This difference is even more pronounced in the case of the longer sequence with $T = 784$. Both vanilla RNNs and minimalRNNs initialized off-criticality failed at task. The well-conditioned minimal-RNN trains a factor of three faster than the vanilla RNN. As predicted above, the difference in training speed between orthogonal and Gaussian initialization schemes is significant for vanilla RNNs but is insignificant for the minimalRNN. This is corroborated in fig. 3 (b,d) where the distribution of the weights has no impact on the training speed.

5. Language modeling

We compare the minimalRNN against more complex gated RNNs such as LSTM and GRU on the Penn Tree-Bank corpus (Marcus et al., 1993). Language modeling is a difficult task, and competitive performance is often achieved by more complicated RNN cells. We show that the minimalRNN achieves competitive performance despite its simplicity.

We follow the precise setup of (Mikolov et al., 2010; Zaremba et al., 2014), and train RNNs of two sizes: a small configuration with 5M parameters and a medium-sized configuration with 20M parameters⁴. We report the perplexity on the validation and test sets. We focus our comparison on single layer RNNs, however we also report perplexities for multi-layer RNNs from the literature for reference. We

⁴The hidden layer size of these networks are adjusted accordingly to reach the target model size.

	5M-t	20M-v	20M-t
VanillaRNN (Jozefowicz et al., 2015)	122.8	103.0	97.7
GRU (Jozefowicz et al., 2015)	108.2	95.5	91.7
LSTM (Jozefowicz et al., 2015)	109.7	83.3	78.8
LSTM	95.4	87.5	83.8
GRU	99.5	93.9	89.8
minimalRNN	101.4	94.4	89.9

Table 1. Perplexities on the PTB. minimalRNN achieves comparable performance to the more complex gated RNN architectures despite its simplicity.

follow the learning schedule of Zaremba et al. (2014) and (Jozefowicz et al., 2015). We review additional hyperparameter ranges in section F of the supplementary material.

Table 1 summarizes our results. We find that single layer RNNs perform on par with their multi-layer counterparts. Despite being a significantly simpler model, the minimal-RNN performs comparably to GRUs. Given the closed-form critical initialization developed here that significantly boosts convergence speed, the minimalRNN might be a favorable alternative to GRUs. There is a gap in perplexity between the performance of LSTMs and minimalRNNs. We hypothesize that this is due to the removal of an independent gate on the input. The same strategy is employed in GRUs and may cause a conflict between keeping longer-range memory and updating new information as was originally pointed out by Hochreiter & Schmidhuber (1997).

6. Discussion

We have developed a theory of signal propagation for random vanilla RNNs and a simple gated RNNs. We demonstrate rigorously that the theory predicts trainability of these networks and gating mechanisms allow for a significantly larger trainable region. We are planning to extend the theory to more complicated RNN cells as well as RNNs with multiple layers.

Acknowledgements

We thank Jascha Sohl-Dickstein and Greg Yang for helpful discussions and Ashish Bora for many contributions to early stages of this project.

References

- Ackley, David H., Hinton, Geoffrey E., and Sejnowski, Terrence J. A learning algorithm for boltzmann machines*. *Cognitive Science*, 9(1). ISSN 1551-6709.
- Arjovsky, Martin, Shah, Amar, and Bengio, Yoshua. Unitary evolution recurrent neural networks. In *International Conference on Machine Learning*, pp. 1120–1128, 2016.
- Ba, Jimmy Lei, Kiros, Jamie Ryan, and Hinton, Geoffrey E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.
- Bahdanau, Dzmitry, Cho, Kyunghyun, and Bengio, Yoshua. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
- Bertschinger, Nils, Natschläger, Thomas, and Legenstein, Robert A. At the edge of chaos: Real-time computations and self-organized criticality in recurrent neural networks.
- Cakmak, Burak. Non-hermitian random matrix theory for mimo channels. Master’s thesis, Institutt for elektronikk og telekommunikasjon, 2012.
- Chung, Junyoung, Gulcehre, Caglar, Cho, KyungHyun, and Bengio, Yoshua. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.
- Collins, Jasmine, Sohl-Dickstein, Jascha, and Sussillo, David. Capacity and trainability in recurrent neural networks. *ICLR*, 2016.
- Daniely, A., Frostig, R., and Singer, Y. Toward Deeper Understanding of Neural Networks: The Power of Initialization and a Dual View on Expressivity. *arXiv:1602.05897*, 2016.
- Derrida, B. and Pomeau, Y. Random networks of automata: A simple annealed approximation. *EPL (Europhysics Letters)*, 1(2):45.
- Elman, Jeffrey L. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.
- Glorot, Xavier and Bengio, Yoshua. Understanding the difficulty of training deep feedforward neural networks.
- Graves, Alex, Mohamed, Abdel-rahman, and Hinton, Geoffrey. Speech recognition with deep recurrent neural networks. In *ICASSP*, pp. 6645–6649. IEEE, 2013.
- Greff, Klaus, Srivastava, Rupesh K, Koutník, Jan, Steunebrink, Bas R, and Schmidhuber, Jürgen. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232, 2017.
- Haagerup, Uffe and Larsen, Flemming. Brown’s spectral distribution measure for r-diagonal elements in finite von neumann algebras. *Journal of Functional Analysis*, 176(2):331–367, 2000.
- Hanin, Boris and Rolnick, David. How to start training: The effect of initialization and architecture. *arXiv preprint arXiv:1803.01719*, 2018.
- Hayou, Soufiane, Doucet, Arnaud, and Rousseau, Judith. On the selection of initialization and activation function for deep neural networks. *arXiv preprint arXiv:1805.08266*, 2018.
- He, K., Zhang, X., Ren, S., and Sun, J. Deep Residual Learning for Image Recognition. *ArXiv e-prints*, December 2015.
- Hidasi, Balázs, Karatzoglou, Alexandros, Baltrunas, Linas, and Tikk, Domonkos. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.
- Hochreiter, Sepp and Schmidhuber, Jürgen. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- Hyland, Stephanie L and Rätsch, Gunnar. Learning unitary operators with help from u (n). In *AAAI*, pp. 2050–2058, 2017.
- Ioffe, Sergey and Szegedy, Christian. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of The 32nd International Conference on Machine Learning*, pp. 448–456, 2015a.
- Ioffe, Sergey and Szegedy, Christian. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pp. 448–456, 2015b.
- Jozefowicz, Rafal, Zaremba, Wojciech, and Sutskever, Ilya. An empirical exploration of recurrent network architectures. In *ICML*, pp. 2342–2350, 2015.
- Jozefowicz, Rafal, Vinyals, Oriol, Schuster, Mike, Shazeer, Noam, and Wu, Yonghui. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*, 2016.
- Karakida, R., Akaho, S., and Amari, S.-i. Universal Statistics of Fisher Information in Deep Neural Networks: Mean Field Approach. *ArXiv e-prints*, June 2018.
- Kiros, Ryan, Zhu, Yukun, Salakhutdinov, Ruslan R, Zemel, Richard, Urtasun, Raquel, Torralba, Antonio, and Fidler, Sanja. Skip-thought vectors. In *Advances in neural information processing systems*, pp. 3294–3302, 2015.

- Le, Quoc V, Jaitly, Navdeep, and Hinton, Geoffrey E. A simple way to initialize recurrent networks of rectified linear units. *arXiv preprint arXiv:1504.00941*, 2015.
- LeCun, Yann, Bottou, Léon, Bengio, Yoshua, and Haffner, Patrick. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- Lee, Jaehoon, Bahri, Yasaman, Novak, Roman, Schoenholz, Samuel S, Pennington, Jeffrey, and Sohl-Dickstein, Jascha. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017.
- Marcus, Mitchell P, Marcinkiewicz, Mary Ann, and Santorini, Beatrice. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.
- Mikolov, Tomas, Karafiát, Martin, Burget, Lukas, Cernocký, Jan, and Khudanpur, Sanjeev. Recurrent neural network based language model. In *Interspeech*, volume 2, pp. 3, 2010.
- Mishkin, Dmytro and Matas, Jiri. All you need is a good init. *arXiv preprint arXiv:1511.06422*, 2015.
- Neal, Radford M. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.
- Pascanu, Razvan, Mikolov, Tomas, and Bengio, Yoshua. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pp. 1310–1318, 2013.
- Pennington, Jeffrey, Schoenholz, Sam, and Ganguli, Surya. Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice. *NIPS*, 2017.
- Pennington, Jeffrey, Schoenholz, Samuel S., and Ganguli, Surya. The emergence of spectral universality in deep networks. In *AISTATS*, pp. 1924–1932, 2018.
- Poole, B., Lahiri, S., Raghu, M., Sohl-Dickstein, J., and Ganguli, S. Exponential expressivity in deep neural networks through transient chaos. *NIPS*, 2016.
- Rumelhart, David E, Hinton, Geoffrey E, and Williams, Ronald J. Learning representations by back-propagating errors. *nature*, 323(6088):533, 1986.
- Saul, Lawrence K, Jaakkola, Tommi, and Jordan, Michael I. Mean field theory for sigmoid belief networks. *Journal of artificial intelligence research*, 4:61–76, 1996.
- Saxe, Andrew M, McClelland, James L, and Ganguli, Surya. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.
- Schoenholz, S. S., Gilmer, J., Ganguli, S., and Sohl-Dickstein, J. Deep Information Propagation. *ICLR*, 2017.
- Schoenholz, Samuel S, Pennington, Jeffrey, and Sohl-Dickstein, Jascha. A correspondence between random neural networks and statistical field theory. *arXiv preprint arXiv:1710.06570*, 2017.
- Sompolinsky, H., Crisanti, A., and Sommers, H. J. Chaos in random neural networks. *Phys. Rev. Lett.*, 61:259–262, Jul 1988. doi: 10.1103/PhysRevLett.61.259.
- Sussillo, David and Abbott, LF. Random walks: Training very deep nonlinear feed-forward networks with smart initialization. *CoRR*, vol. abs/1412.6558, 2014.
- Tallec, Corentin and Ollivier, Yann. Can recurrent neural networks warp time? *arXiv preprint arXiv:1804.11188*, 2018.
- Vorontsov, Eugene, Trabelsi, Chiheb, Kadoury, Samuel, and Pal, Chris. On orthogonality and learning recurrent networks with long term dependencies. *arXiv preprint arXiv:1702.00071*, 2017.
- Wisdom, Scott, Powers, Thomas, Hershey, John, Le Roux, Jonathan, and Atlas, Les. Full-capacity unitary recurrent neural networks. In *Advances in Neural Information Processing Systems*, pp. 4880–4888, 2016.
- Wu, Chao-Yuan, Ahmed, Amr, Beutel, Alex, Smola, Alexander J, and Jing, How. Recurrent recommender networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pp. 495–503. ACM, 2017.
- Xie, Di, Xiong, Jiang, and Pu, Shiliang. All you need is beyond a good init: Exploring better solution for training extremely deep convolutional neural networks with orthonormality and modulation. *arXiv preprint arXiv:1703.01827*, 2017.
- Yang, Greg and Schoenholz, Sam S. Deep mean field theory: Layerwise variance and width variation as methods to control gradient explosion, 2018.
- Yang, Greg and Schoenholz, Samuel S. Mean field residual networks: On the edge of chaos. *arXiv preprint arXiv:1712.08969*, 2017.
- Zaremba, Wojciech, Sutskever, Ilya, and Vinyals, Oriol. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.
- Zilly, Julian Georg, Srivastava, Rupesh Kumar, Koutník, Jan, and Schmidhuber, Jürgen. Recurrent highway networks. *arXiv preprint arXiv:1607.03474*, 2016.