Mean Field Analysis of Recurrent Neural Networks

Sarang Mittal
Marcella Bonsall SURF Fellow
Mentor: Maria Spiropolu
Co-Mentors: Jean-Roch Vlimant, Stephan Zheng
California Institute of Technology

Abstract

We analyze the dynamics of recurrent neural networks whose initial weights and biases are randomly distributed. Utilizing mean field theory and making some mean field approximations, we develop a theoretical basis for length scales which limit the depth of signal propagation in a one layer recurrent network with long input or prediction sequences. For specific hyperparameter choices, these length scales diverge. Building on the conclusions of Schoenholz et al. [2016], we argue that random recurrent networks can only be trained if the input signal can propagate through them. This suggests that recurrent networks can be trained with arbitrarily long sequences provided they are initialized sufficiently close to criticality. We begin searching for empirical evidence in support of our conclusions, and provide some preliminary results.

Background

One of the newest and most excited data analysis tools to be applied to high energy physics (HEP) is machine learning, and specifically artificial neural networks. The applications of neural networks (NNs) outside of HEP are plentiful. Tech giants such as Google, Apple, and Microsoft are using neural networks for speech recognition, image classification, and other artificial intelligence applications (Poole et al. [2016], Schoenholz et al. [2016]). The best results seem to come from a Deep Neural Networks (DNN), which are essentially multiple NNs stacked on top of each other.

Despite the overwhelming success of neural networks, one of the major issues in the field is the interpretability of the models. To the scientist, the model is essentially a black box, with a collection of matrices that one cannot directly interpret. Additionally, the process of selecting the right hyperparameters for training a DNN is to some extent, guess work. There has been some recent work to apply ideas from statistical physics to understand the dynamics of the DNN. Lastly, DNNs are particularly hard to train due to exploding/vanishing gradients. It is difficult to propagate corrections to the model if there are many layers.

To solve these problems, computer scientists have turned to tools used in statistical physics. One intuitive way to consider DNNs is that each layer learns progressively more abstract features. To the theoretical physicists, this is reminiscent of the renormalization group, in which an iterative coarse-graining scheme is applied to deemphasize unimportant microscopic reactions and capture the macroscopic behavior of a collection of interacting particles. In this vein, there was recently a paper that demonstrated an exact mapping between variational renormalization and a specific deep learning model known as stacked restricted Boltzmann machines (Mehta and Schwab [2014]).

Another important paper in the field applied mean field theory to feed-forward, fully connected NNs, finding theoretical and empirical evidence of a phase transition between trainability and untrainability in the hyperparameter space. They claimed that asymptotically deep networks can be trained given that they are initialized sufficiently close to the critical threshold. This behavior is claimed to be independent of data set, and solely a result of the network architecture (Schoenholz et al. [2016]).

This project extends the analysis of that paper to a more complicated network architecture known as a recurrent neural network (RNN). These kinds of networks are built to take inputs over a period of time. For example, one could feed in the track of a particle in the Compact Muon Solenoid (CMS) at the Large

Contact: smittal@caltech.edu. Code and plots can be found at https://github.com/sarangmittal/DLRG

Hadron Collider (LHC), and ask the network to reconstruct the trajectory of the particle before it interacted with the detector. In RNN's, the analog to the network depth is the sequence length of the data. Thus, we will apply a mean field approximation to the signals in RNN's, and look for specific initializations that allow for extremely long sequence lengths.

Forward Signal Propagation

Model Architecture

We will study the dynamics of Elman networks, which are governed by the following equations:

$$\mathbf{z}^t = \mathbf{W}^h \mathbf{h}^{t-1} + \mathbf{W}^x \mathbf{x}^t + \mathbf{b}^h \tag{1}$$

$$\mathbf{h}^t = \phi(\mathbf{z}^t) \tag{2}$$

With h^t the hidden state, z^t the pre-activation, W^h and W^x as matrices with trainable weights and b^h the trainable vector of biases. The matrices are initialized with mean 0 and variance $\frac{\sigma_w^2}{N_h}$ where N_h is the number of hidden features. The bias vector is initialized with mean 0 and variance σ_b^2. The hidden state starts of as a vector full of zeros. The weights and biases are independent random variables, and for now, we will assume that the input vectors at each time step x^t are simply noise.

Variance and Co-variance map

The mean field approximation is to replace z_i^t with a Gaussian whose first two moments match that of the pre-activations. Notice, we have defined the weights and biases to have zero mean, the moments are:

$$\mathbb{E}[z_i^t] = 0$$

$$\mathbb{E}[z_i^t; z_j^t] = q^t \delta_{ij}$$

where δ_{ij} is the Kronecker delta and q^t is defined:

$$q^t = \frac{1}{N_h} \sum_{i=1}^{N_h} (z_i^t)^2$$

or the normalized variance of the pre-activation. We can construct an iterative way to determine q^t:

$$q^t = \langle (z_i^t)^2 \rangle = \langle (\mathbf{w}^{h,i} \cdot \phi(\mathbf{z}^{t-1}) + \mathbf{w}^{x,i} \cdot \mathbf{x}^t + b_i^h)^2 \rangle \tag{3}$$

By linearity of expected value and the independence of the weights and biases:

$$q^t = \langle (\mathbf{w}^{h,i} \cdot \phi(\mathbf{z}^{t-1}))^2 \rangle + \langle (\mathbf{w}^{x,i} \cdot \mathbf{x}^t)^2 \rangle + \langle (b_i^h)^2 \rangle \tag{4}$$

$$= \frac{\sigma_w^2}{N_h} \sum_{i=1}^{N_h} \phi(z_i^{t-1})^2 + \frac{\sigma_w^2}{N_h} \sum_{i=1}^{N_d} (x_i^t)^2 + \sigma_b^2 \tag{5}$$

Because the previous hidden state is also distributed like a Gaussian with zero mean and variance q^{t-1}:

$$q^t = \sigma_w^2 \int \mathcal{D}_z \phi(\sqrt{q^{t-1}} z)^2 + \frac{\sigma_w^2}{N_h} \sum_{i=1}^{N_d} (x_i^t)^2 + \sigma_b^2 \tag{6}$$

let's assume that the x_i^t are i.i.d according to some probability density function $f(x)$.

$$q^t = \sigma_w^2 \int \mathcal{D}_z \phi(\sqrt{q^{t-1}} z)^2 + \sigma_w^2 \frac{N_d}{N_h} \int dx \; x^2 f(x) + \sigma_b^2 \tag{7}$$

With $\mathcal{D}_z$ the standard Gaussian measure. If we further approximate the input as a Gaussian with mean μ_x and variance σ_x^2:

$$\int dx\ x^2 f(x) = \int dx\ x^2 \frac{1}{\sqrt{2\pi\sigma_x^2}}\ e^{\frac{-(x-\mu_x)^2}{2\sigma_x^2}} = \sigma_x^2 + \mu_x^2 \tag{8}$$

Thus, our iterative map for the variance is:

$$q^t = \sigma_w^2 \int \mathcal{D}_z \phi(\sqrt{q^{t-1}}z)^2 + \sigma_w^2 \frac{N_d}{N_h}(\sigma_x^2 + \mu_x^2) + \sigma_b^2 \tag{9}$$

with initial condition $q^0 = 0$. If we reference Schoenholder (reference here), we see that this is similar to the feed forward networks (FFNs), except there is a new term that represent how the data is distributed. If the number of hidden features N_h is much larger than the number of input dimensions, $N_d/N_h << 1$, then equation 9 reduces to the same equation from the FFN case. It is also worthwhile to note that in most cases, one can rescale the data such that is has 0 mean, further reducing the magnitude of the second term. By running this map with several different parameter values, we can observe these effects in Figure 1.

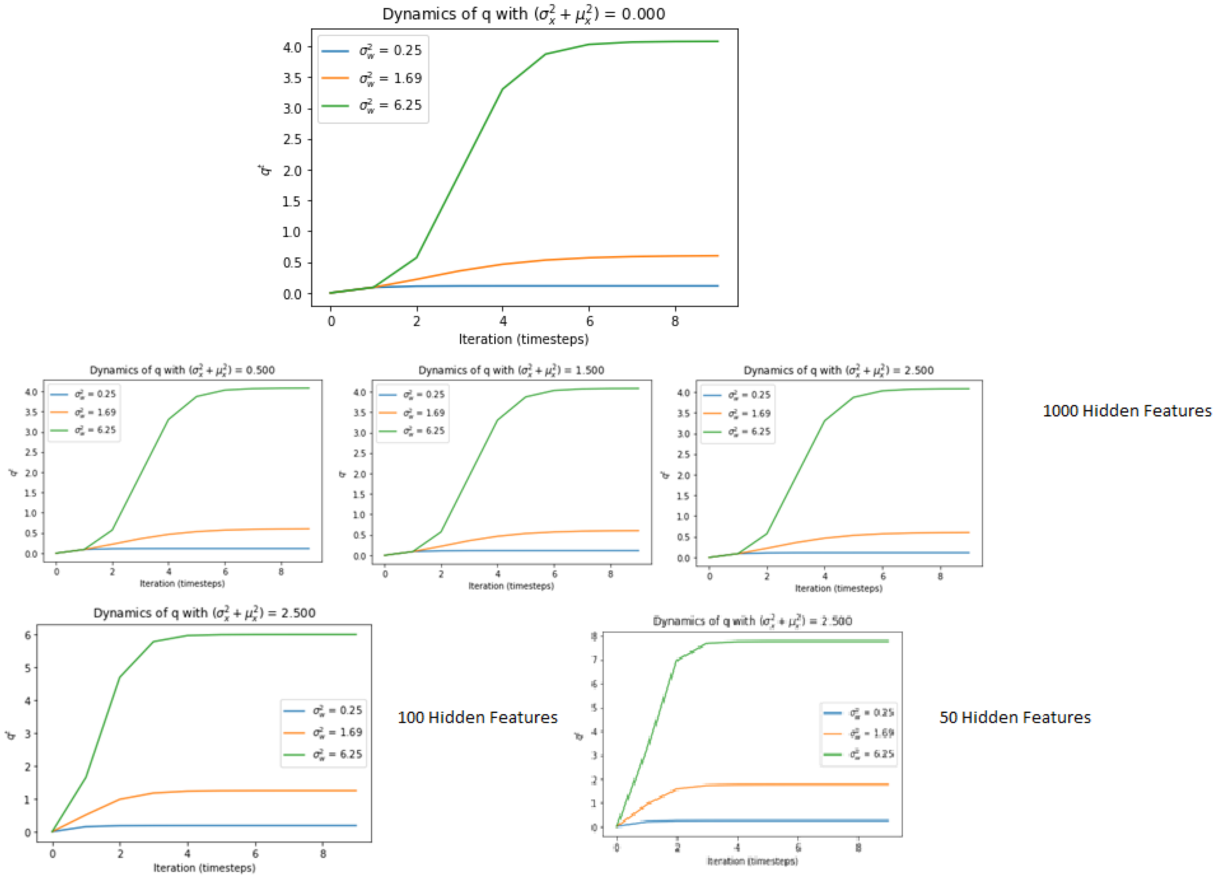

Figure 1: Variance Dynamics of Elman RNN. One sees that all the maps converge to the fixed point fairly quickly. Also, as the magnitude of the second term of equation 9 grows, so does $q*$, the fixed point. All plots have $\sigma_b = 0.3$ and $N_d = 10$. From the plots with fewer hidden features, we can also see the increasing divergence from the FFN equation.

We can use similar logic to construct an iterative map for the covarience between the pre-activation states of two different input sequences:

$$q_{ab}^t = \frac{1}{N_h}\sum_{i=1}^{N_h} z_i^t(\mathbf{x}^a) z_i^t(\mathbf{x}^b) \tag{10}$$

where $\mathbf{x}^a$ is the series of points $\{\mathbf{x}^{0,a}, \mathbf{x}^{1,a}, ..., \mathbf{x}^{t,a}\}$. Inputting the dynamics at one time-step,

$$q^t_{ab} = \langle (\mathbf{w}^{h,i} \cdot \phi(\mathbf{z}^{t-1}(\mathbf{x}^a)) + \mathbf{w}^{x,i} \cdot \mathbf{x}^{t,a} + b^h_i)(\mathbf{w}^{h,i} \cdot \phi(\mathbf{z}^{t-1}(\mathbf{x}^b)) + \mathbf{w}^{x,i} \cdot \mathbf{x}^{t,b} + b^h_i)\rangle \tag{11}$$

Because the weights and biases are independently distributed, each with 0 mean:

$$\begin{aligned} q^t_{ab} &= \langle (\mathbf{w}^{h,i} \cdot \phi(\mathbf{z}^{t-1}(\mathbf{x}^a)))\ (\mathbf{w}^{h,i} \cdot \phi(\mathbf{z}^{t-1}(\mathbf{x}^b)))\rangle + \langle (\mathbf{w}^{x,i} \cdot \mathbf{x}^{t,a})\ (\mathbf{w}^{x,i} \cdot \mathbf{x}^{t,b})\rangle + \langle (b^h_i)^2\rangle & (12) \\ &= \frac{\sigma_w^2}{N_h} \sum_{i=1}^{N_h} \phi(z_i^{t-1}(\mathbf{x}^a))\ \phi(z_i^{t-1}(\mathbf{x}^b)) + \frac{\sigma_w^2}{N_h} \sum_i^{N_d} x_i^{t,a} x_i^{t,b} + \sigma_b^2 & (13) \end{aligned}$$

If we assume that the joint empirical distribution of $\mathbf{z}^t(\mathbf{x}^a))$ and $\mathbf{z}^t(\mathbf{x}^b))$ converges to a two-dimensional Gaussian with covariance q^t_{ab}, then:

$$q^t_{ab} = \sigma_w^2 \int \mathcal{D}_{z_1}\mathcal{D}_{z_2}\ \phi(u_1)\phi(u_2)\ \ + \frac{\sigma_w^2}{N_h} \sum_i^{N_d} x_i^{t,a} x_i^{t,b} + \sigma_b^2 \tag{14}$$

with:

$$u_1 = \sqrt{q_{aa}^{t-1}} z_1\ , u_2 = \sqrt{q_{bb}^{t-1}}[c_{ab}^{t-1} z_1 + \sqrt{1-(c_{ab}^{t-1})^2} z_2]\ , c^t_{ab} = \frac{q^t_{ab}}{\sqrt{q^t_{aa}}\sqrt{q^t_{bb}}} \tag{15}$$

Making a similar approximation for the data as we did in the earlier derivation (Equation 7-9):

$$q^t_{ab} = \mathcal{C}(c_{ab}^{t-1}, q_{aa}^{t-1}, q_{bb}^{t-1} | \sigma_w, \sigma_b) = \sigma_w^2 \int \mathcal{D}_{z_1}\mathcal{D}_{z_2}\ \phi(u_1)\phi(u_2)\ \ + \sigma_w^2 \frac{N_d}{N_h}(\sigma_x^2 + \mu_x^2) + \sigma_b^2 \tag{16}$$

The dynamics for the Pearson correlation c^t_{ab} when both inputs are at the fixed point q^*,can be obtained by setting $q^t_{aa} = q^t_{bb} = q^*$ and dividing the above equation by q^*:

$$c^t_{ab} = \frac{1}{q^*}\mathcal{C}(c_{ab}^{t-1}, q^*, q^* | \sigma_w, \sigma_b) \tag{17}$$

Simple calculations show that $c^t_{ab}(1) = 1$, meaning that $c^t_{ab} = 1$ is a fixed point of the dynamics. We are allowed to make the assignment $q^t_{aa} = q^t_{bb} = q^*$, because the variance map converges to the fixed point within a few iterations.

We will also be interested in the slope of the $\mathcal{C}$ map:

$$\chi = \frac{dc^t_{ab}}{dc^{t-1}_{ab}} = \sigma_w^2 \int \mathcal{D}z[\phi'(\sqrt{q^*}z)]^2 \tag{18}$$

Note this is the same as the FFN case because the term added to the $\mathcal{C}$ map was independent of the correlation at the previous timestep. When $\chi \geq 1$, $c^* = 1$ becomes a fixed point because it indicates that any change to the correlations in the previous layer are carried on fully to the next layer, whereas if $\chi < 1$, the changes to the correlation eventually decays to 0 over several layers. This can be seen in Figure 2. Additionally, we observed that the correlation fixed point is independent of the starting correlation, which can be seen in Figure 3.

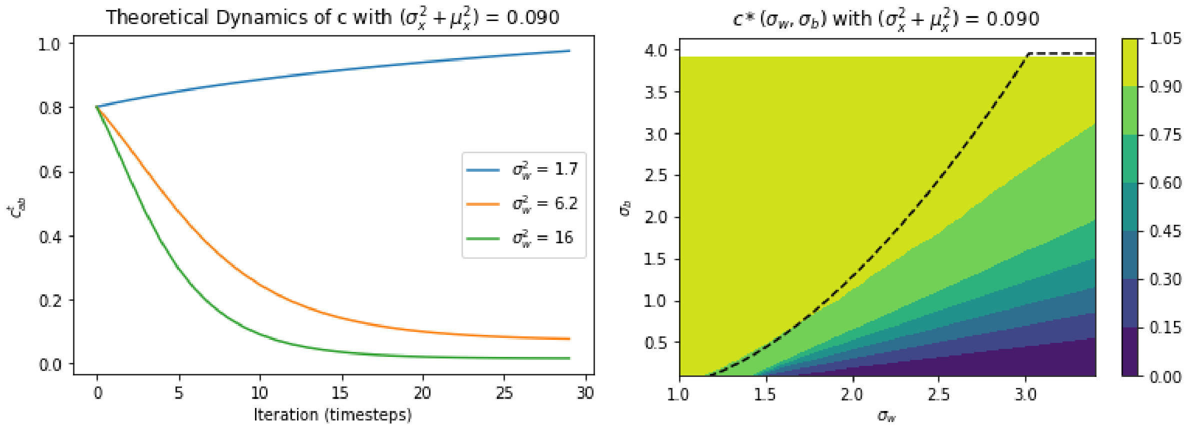

Figure 2: Pearson Correlation of pre-activations between two input sequences. The left plot shows the dynamics of the correlation over several iterations ($N_h = 128, \sigma_b = 0.05$). The right plot shows the fixed point of the correlation map over the initialization space. The dotted line represents $\chi = 1$. As expected, for $\chi \geq 1$, the correlation converges to approximately 1. For $\chi 1 < 1$, the correlation decays to 0

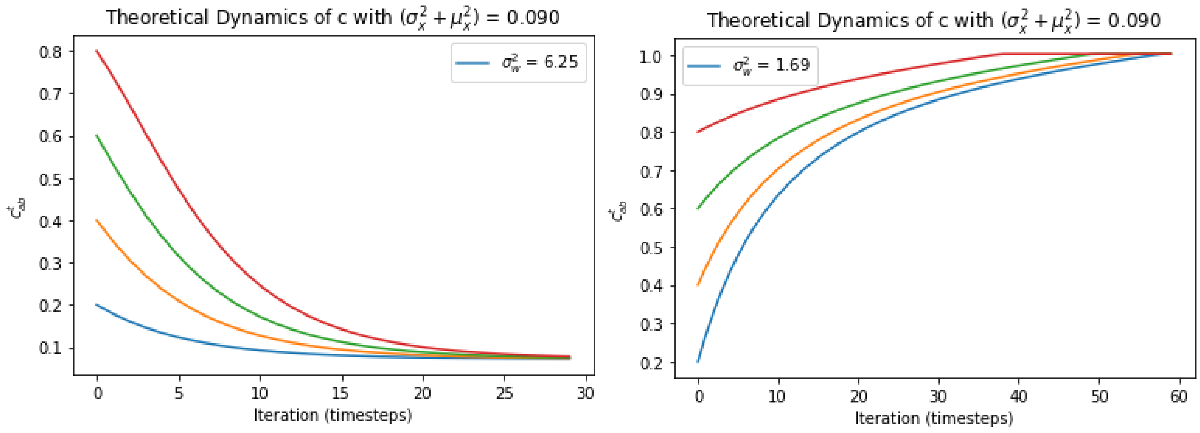

Figure 3: Correlation fixed point independent of starting correlation. ($N_h = 128, \sigma_b = 0.05$)

Asymptotic Expansion and Characteristic Length Scales

Schoenholz et. al. observed that the residuals of the variance and correlation behaved exponentially over the layers for feed forward networks. We observed a similar behavior in RNN's (Figure 4). The y-axis is on a log scale, so the straight lines indicate exponential behavior of the form $|q - q^*| \sim e^{-t/\xi_q}$ and $|c - c^*| \sim e^{-t/\xi_c}$

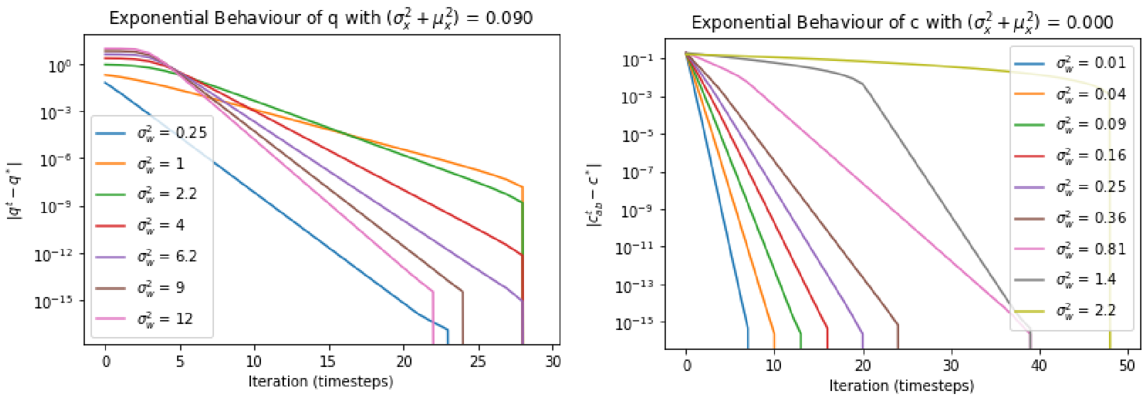

Figure 4: Exponential Behavior of Variance/Correlation in the Approach of the Fixed Point. These were made by pre-calculating q^* and c^*, iterating the variance/correlation map and plotting the residual $|q - q^*|$ or $|c - c^*|$

To determine ξ_q, let $q^t = q^* + \epsilon^t$. Because we know, $\lim_{l \Rightarrow \inf} q^t = q^*$, this construction ensures $\epsilon^t \Rightarrow 0$ as $t \Rightarrow \inf$. Immediately, we observe that the new term included in equation 9 is independent of q^{t-1}, if one follows the calculations of Schoenholz, our equations will look the same. Thus,

$$\xi_q^{-1} = -\log[\chi + \sigma_w^2 \int \mathcal{D}z \phi''(\sqrt{q^* z})\phi(\sqrt{q^* z})] \tag{19}$$

and similarly for ξ_c:

$$\xi_c^{-1} = -\log[\sigma_w^2 \int \mathcal{D}z_1 \mathcal{D}z_2 \phi'(u_1^*)\phi'(u_2^*)] \tag{20}$$

It is important to note that while our equations look the same, they are not equivalent. The equations defining q^*, u_1^*, and u_2^* are different, so these fixed points are also slightly different. In other words, the length scale equation for RNN's is the same as that for FFN's, except the fixed points are shifted. As in the FFN case, these length scales have sharp peaks at specific orientations. In Figure 5, we plot the length scales. We also show the inferred length scale obtained by fitting exponential functions to the curves generated in Figure 4.

The argument made in Schoenholz is that the divergent points in Figure 5 tell us where to initialize the weights and biases in the neural network. In the spirit of that paper, we look for experimental evidence of this behavior in recurrent networks.

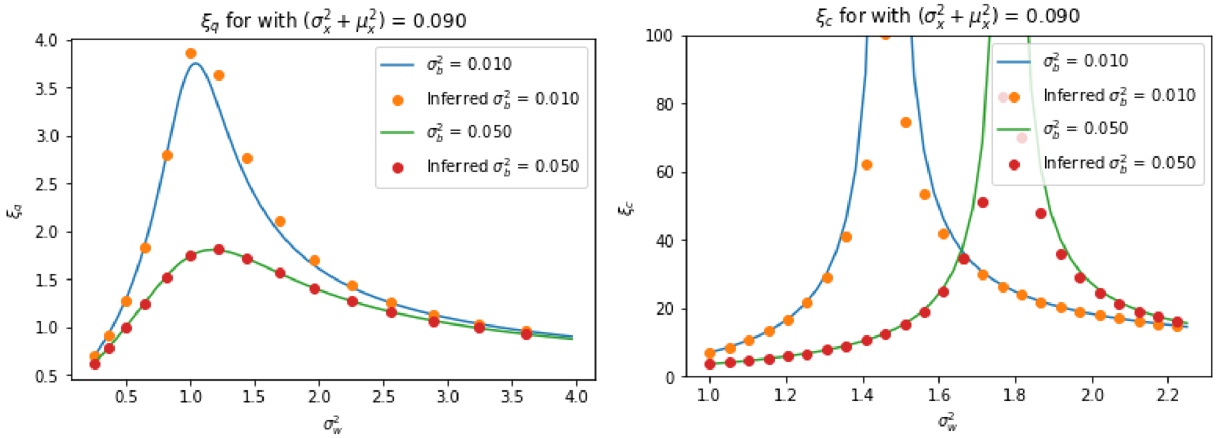

Figure 5: Calculated and inferred length scales for RNNs. One important difference between these two plots is that the length scales for the correlation are much longer than those for the variances.

Experimental Work

In order to observe evidence of the length scales, we had to pick a problem that difficult enough to observe differences in training error among different initializations. We chose to use a sequence to sequence RNN, with an encoder-decoder architecture. This network takes several initial data points into the encoder, and then uses the decoder to predict the remaining points in the sequence.

For our problem, we selected predicting Lorentz Attractor trajectories from the first several points. We had initially used simple circular trajectories, but they proved too easy. The Lorentz attractor is a set of chaotic solutions to the Lorentz system (Yu et al. [2017]). Using Pytorch, a sequence to sequence Elman RNN was implemented that used the first 20% of points as input in order to predict the rest of the points. In the decoder, the network used its own predictions as the input for the next timestep, and the first input to the decoder was the origin. The hidden network is initialized as all zeros.

A learning rate was tuned in between 10^{-2} to 10^{-4}, as the larger networks had smoother training with smaller learning rates. (Klambauer et al. [2017]). The networks were trained for 100 epochs on the computing cluster in the basement of Lauritsen. We also had a chance to run code on the ORNL Titan, which allowed running up to 500 epochs. The results are shown in Figures 6 and 7.

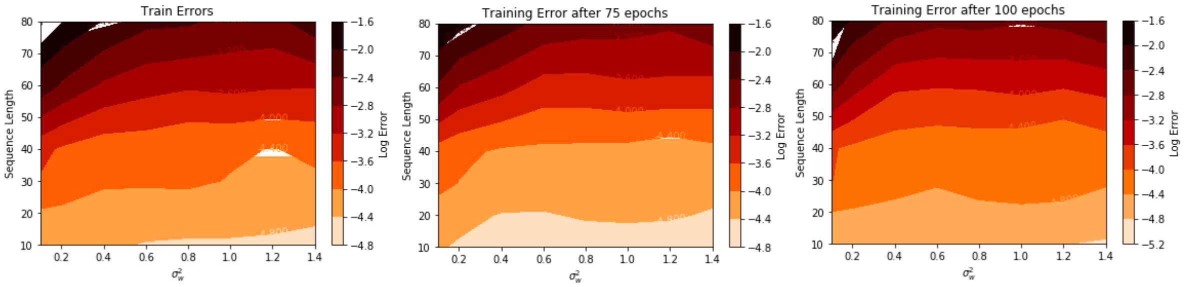

Figure 6: Training Error after 50, 75, and 100 epochs respectively. All plots had $N_h = 128$, $\sigma_b^2 = 0.05$, and $N_d = 3$

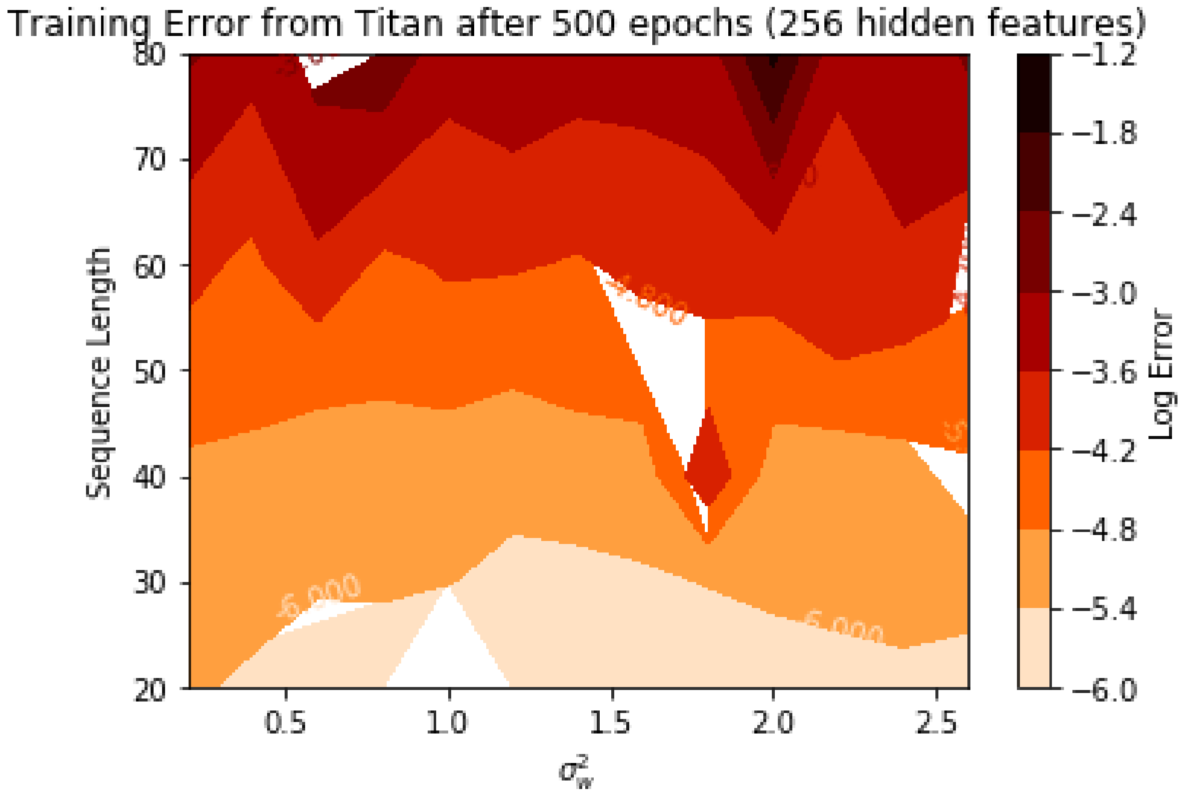

Figure 7: Training Error after 500 epochs. $N_h = 256$, $\sigma_b^2 = 0.05$, and $N_d = 3$

There doesn't appear to be any similarities between these experimental observations and the theory that was previously developed. In order to check the validity of our previous assumptions, we compared the theoretical correlation dynamics to the actual correlation as we passed all our data sequences through the untrained, random network. These experiments showed that our assumptions were accurate for the forward propagation of signals. (See Figure 8).

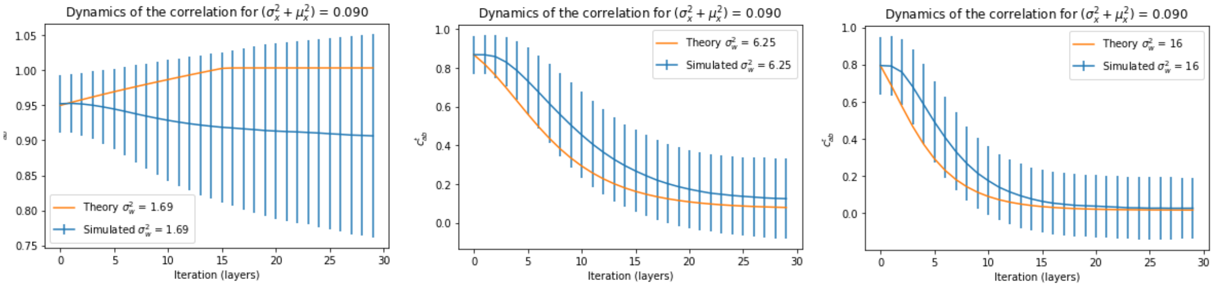

Figure 8: Theory compared to Calculated Correlation.

Discussion

There were several challenges that, if overcome, may result in experimental evidence of the lengths scales we derived. From an implementation standpoint, gradient explosion made training quite difficult as the models got larger. This slowed down experimental progression as hyperparameters, mainly the learning rate, had to be adjusted towards a parameter space that slowed down the training process. Gradient explosion is a well known problem in RNN training, and there have been methods developed to help combat it (Pascanu et al. [2013]). However, when implementing these training tricks, it is important to select those that do not perturb the information propagation.

Perhaps a more fundamental issue arises from the nature of the problem we set out to solve with our RNN implementations. It is an inaccurate to compare training accuracy on models of widely different lengths because the complexity of the problem is not constant. The networks of longer lengths have many more points to predict than do the shorter models. Having the input sequence to encoder scale with the total sequence length is only a partial solution. Further work in the problem should carefully select a problem that nullifies this effect. However, this could prove difficult, as RNNs are specifically designed to solve problems based on sequential data. If an appropriate problem cannot be found, future work must develop a more impartial way of comparing the training accuracy.

Finally, while it important that connections between input data can forward propagate through the network, it is equally important that gradients can back propagate through the network. This is especially crucial for long model lengths. RNN's employ a more complicated back propagation algorithm than FFN's do. Known as Backpropagation through time (BPTT), the algorithm accounts for the fact that the gradient is dependent on the weight matrix at every timestep. We were not able to fully analyze BPTT from a mean-field approach, but some preliminary work is presented below.

BPTT

Given an error E,

$$\frac{\partial E}{\partial \theta} = \sum_{1 \leq t \leq T} \frac{\partial E_t}{\partial \theta} \tag{21}$$

$$\frac{\partial E_t}{\partial W^h_{ij}} = \sum_{1 \leq k \leq t} (\frac{\partial E_t}{\partial z^t} \frac{\partial z^t}{\partial z^k} \frac{\partial z^k}{\partial W^h_{ij}}) \tag{22}$$

Since the weights are i.i.d,

$$||\nabla_{W^h_{ij}} E^t||^2 = \sum_{ij} (\frac{\partial E^t}{\partial W^h_{ij}})^2 \tag{23}$$

$$\approx N_h^2 \ \mathrm{E}[(\frac{\partial E^t}{\partial W^h_{ij}})^2] \tag{24}$$

Expanding the expectation value:

$$\mathrm{E}[(\frac{\partial E^t}{\partial W^h_{ij}})^2] = \mathrm{E}[(\frac{\partial E_t}{\partial z^t})^2 (\sum_{1 \leq k \leq t} \frac{\partial z^t}{\partial z^k} \frac{\partial z^k}{\partial W^h_{ij}})^2] \tag{25}$$

Note $\frac{\partial z^k}{\partial W^h_{ij}} = \phi(z^{k-1})$.

Let's try to develop a recursive expression for $\frac{\partial E_t}{\partial W^h_{ij}}$. Starting with equation 22,

$$\frac{\partial E_{t+1}}{\partial W^h_{ij}} = \sum_{1 \leq k \leq t+1} (\frac{\partial E_{t+1}}{\partial z^{t+1}} \frac{\partial z^{t+1}}{\partial z^k} \frac{\partial z^k}{\partial W^h_{ij}}) \tag{26}$$

$$= \frac{\partial E_{t+1}}{\partial z^{t+1}} \sum_{1 \leq k \leq t+1} (\frac{\partial z^{t+1}}{\partial z^k} \frac{\partial z^k}{\partial W^h_{ij}}) \tag{27}$$

$$= \frac{\partial E_{t+1}}{\partial z^{t+1}} \frac{\partial z^{t+1}}{\partial z^t} ((\sum_{1 \leq k \leq t} \frac{\partial z^t}{\partial z^k} \frac{\partial z^k}{\partial W^h_{ij}}) + \frac{\partial z^t}{\partial z^{t+1}} \frac{\partial z^{t+1}}{\partial W^h_{ij}}) \tag{28}$$

$$= \frac{\partial E_{t+1}}{\partial z^{t+1}} \frac{\partial z^{t+1}}{\partial z^t} (\frac{\partial E_t}{\partial W^h_{ij}} \frac{\partial z^t}{\partial E_t} + \frac{\partial z^t}{\partial z^{t+1}} \frac{\partial z^{t+1}}{\partial W^h_{ij}}) \tag{29}$$

We can evaluate some of these terms:

$$\frac{\partial z^t}{\partial W^h_{ij}} = \phi(z^{t-1}) \tag{30}$$

$$\frac{\partial z^{t+1}}{\partial z^t} = W^h \phi'(z^t) \tag{31}$$

And if $E^t = \mathcal{L}(y^t)$ is the loss function:

$$\frac{\partial E^t}{\partial z^t} = \mathcal{L}'(y^t) W^o \phi'(z^t) \tag{32}$$

With this expansion, it should be efficient to calculate the gradient. We have not applied any mean-field approximations, nor have we attempted to look at the variance. These operations will require careful analysis of the dependence of the terms in our recurrence relation.

Conclusion

We developed a mean-field theory approximation of the forward propagation of signals through a randomly initialized Elman RNN. Despite developing theoretical foundations for the existence of divergent length scales, we were not able to observe the corresponding experimental signatures. Several issues with implementation, as well as fundamental considerations in problem selection an backpropagation could be explain the lack of expected results.

However, from the analysis we have performed so far, the mean-field approach has not been ruled out as a valid path of exploration. As pointed out by an astute attendee of the SURF Seminar Day, the exact place where the mean field breaks down is at the diverging length scales. The work on FFN's did not seem to be impacted by this fact. The more complicated nature of RNN's may expose the shortcomings of the mean field approach. Regardless, we believe that there is merit to continue this line of inquiry, with emphasis placed on understanding backpropagation. There have been analysis of different systems that are similar in spirit to what we have attempted, but on different architectures (Bertschinger and Natschl?ger [2004], Klambauer et al. [2017]) With any luck, we will be able to develop training guidelines that improve RNN performance.

References

Nils Bertschinger and Thomas Natschl?ger. Real-time computation at the edge of chaos in recurrent neural networks. *Neural Computation*, 16(7):1413?1436, 2004. doi: 10.1162/089976604323057443.

G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter. Self-Normalizing Neural Networks. *ArXiv e-prints*, June 2017.

P. Mehta and D. J. Schwab. An exact mapping between the Variational Renormalization Group and Deep Learning. *ArXiv e-prints*, October 2014.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. *Proceedings of the 30th International Conference on Machine Learning,*, 28, 2013.

B. Poole, S. Lahiri, M. Raghu, J. Sohl-Dickstein, and S. Ganguli. Exponential expressivity in deep neural networks through transient chaos. *ArXiv e-prints*, June 2016.

S. S. Schoenholz, J. Gilmer, S. Ganguli, and J. Sohl-Dickstein. Deep Information Propagation. *ArXiv e-prints*, November 2016.

Rose Yu, Stephan Zheng, and Yan Liu. Learning chaotic dynamics using tensor recurrent neural networks. *Proceedings of the ICML 17 Workshop on Deep Structured Prediction*, 2017.