Stable Recurrent Models

John Miller* Moritz Hardt†

February 26, 2019

Abstract

We prove stable recurrent neural networks are well approximated by feed-forward networks for the purpose of both inference and training. Our result applies to a broad range of non-linear recurrent neural networks under natural stability and smoothness assumptions on the state-transition map. As a corollary, we show that stable recurrent neural networks cannot have long-term memory; the gradients of the training objective must vanish with respect to inputs encountered far enough in the past.

From a rigorous theoretical perspective, our work sheds light on central phenomena in learning artificial neural networks such as the vanishing gradient problem and the power of recurrent models.

1 Introduction

Recurrent neural networks are a popular modeling choice for solving sequence learning problems arising in domains such as speech recognition, and natural language processing. At the outset, recurrent neural networks are non-linear dynamical systems commonly trained to fit sequence data via some variant of gradient descent.

Recurrent models feature flexibility and expressivity that come at a cost. Empirical experience shows that these models are often more delicate to tune and more brittle to train [10] than standard feed-forward architectures. Recurrent architectures can also introduce significant computational burden compared with feed-forward implementations.

In response to these shortcomings, a growing line of empirical research succeeds in replacing recurrent models effectively by feed-forward models in important applications, including translation [5, 14], speech [13], and language modeling [4].

This development raises an intriguing question for theoretical investigation:

Can well-behaved recurrent neural networks in principle always be replaced by a feed-forward model of comparable size without loss in performance?

To answer this question, we need to understand what class of recurrent neural networks we ought to call *well-behaved*. In principle, it easy to contrive a non-linear recurrent models that on some input sequences cannot be approximated by a feed forward model. But would such recurrent models be trainable by gradient descent?

*Email: miller_john@berkeley.edu
†Email: hardt@berkeley.edu

One natural—even if not strictly necessary—requirement for gradient descent to work is that the gradients of the training objective do not *explode* over time. This criterion roughly agrees with the natural control-theoretic requirement of *stability*. The system can always be normalized in such a manner that the gradients do not explode, but then they might *vanish*. Gradients that vanish over time suggest that the system is unable to utilize inputs encountered long ago, suggesting that a feed-forward approximation may be possible.

Roughly speaking, we show that for recurrent neural networks there is no robust sweet spot between exploding and vanishing gradients. If they avoid exploding gradients, they exhibit vanishing gradients. We can avoid vanishing gradients by making the system unstable, but then we face exploding gradients and must resort to various heuristics for coping with them, such as, gradient clipping [10].

Extending on the vanishing gradients property, we prove a general approximation result showing that stable recurrent neural networks can be approximated well by feed-forward models for the purpose of both inference and training by gradient descent. The latter result uses fundamental stability properties of gradient descent.

1.1 Contributions

In this work, we make the following contributions.

1. We identify stability as a natural requirment for the analysis of recurrent models and show, under the stability assumption, feed-forward networks can approximate recurrent networks both for inference and training.

2. To allow for a unified analysis, our results are stated and proved for general non-linear dynamical systems. We complement this analysis with sufficient conditions for several commonly used model classes, including long-short-term memory (LSTM) networks, that imply the assumptions of our theorems.

3. We empirically demonstrate feed-forward networks can well-approximate widely used recurrent networks on a benchmark natural language processing task. Moreover, we show that our assumptions are not overly limiting and produce networks that achieve respectable, though not state-of-the-art, results and satisfy all of the assumptions required for our theorems.

2 Problem Statement and Results

We consider general non-linear dynamical systems given by a differentiable *state-transition map* $\phi_w \colon \mathbf{R}^n \times \mathbf{R}^d \to \mathbf{R}^n$, parameterized by $w \in \mathbf{R}^m$. The hidden state $h_t \in \mathbf{R}^n$ evolves in discrete time steps according to the update rule

$$h_t = \phi_w(h_{t-1}, x_t)\,. \tag{1}$$

Here, the vector $x_t \in \mathbf{R}^d$ is an arbitrary input provided to the system at time t. This formulation is quite general and allows us to unify the analysis for several examples of interest, including linear dynamical systems, recurrent neural networks (RNN), and Long-Short-Term Memory (LSTM)

networks. For instance, in the linear dynamical systems cases, given $W \in \mathbf{R}^{n\times n}, U \in \mathbf{R}^{n\times d}$, the system evolves according to

$$h_t = Wh_{t-1} + Ux_t.$$

Throughout this paper, we focus on the case of stable recurrent models. This corresponds to assuming the state-transition map ϕ is *contractive*, so there exists some $\lambda < 1$ such that, for any weights $w \in \mathbf{R}^m$, states $h, h' \in \mathbf{R}^n$, and input $x \in \mathbf{R}^d$,

$$\left\|\phi_w(h, x) - \phi_w(h', x)\right\| \leq \lambda \left\|h - h'\right\|. \tag{2}$$

For each model class, we provide sufficient conditions that imply contractivity. For instance, in the linear dynamical systems case, this corresponds to requiring $\|W\| < 1$. A similar requirement applies to RNNs, where the norm constraint will depend on the choice of non-linearity. The assumptions required for LSTMs are more subtle; we will discuss them later.

We study when the system (1) can be approximated by a feed-forward model with finite context. While there are many choices for a feed-forward approximation, we consider the simplest one—truncation of the system to some finite context k. In other words, the feed forward approximation moves over the input sequence with a sliding window of length k producing an output every time the sliding window advances by one step. Formally, for context length k chosen in advance, we define the *truncated model* via the update rule

$$h_t^k = \phi_w(h_{t-1}^k, x_t), \quad h_{t-k}^k = 0\,. \tag{3}$$

Note that h_t^k is a function only of the previous k inputs $x_{t-k}, \ldots, x_t$, and can be implemented as an autoregressive, depth-k feed-forward model.

2.1 Our results

Our first result concerns inference in stable recurrent models. Suppose we're given a prediction function f that maps a state h_t to outputs $f(h_t) = y_t$.

Proposition (Informal version of Proposition 4)**.** *Assuming the system ϕ is λ-contractive and under additional Lipschitz assumptions, we show if $k \geq O(\log(1/(1-\lambda)\varepsilon))$, then the difference in predictions between the recurrent and truncated model is negligible, $\left\|y_t - y_t^k\right\| \leq \varepsilon$.*

In other words, for fixed weights w, there exists a feed-forward model that well approximates the full recurrent model at test-time. The dependence on the contractivity parameter λ of the system is the natural one even for linear systems, where λ corresponds to the largest singular value of the state transition matrix.

Equipped with our approximation result, we turn towards optimization. Suppose both the full recurrent model and the truncated model are initialized at a common point w^0, and optimized to minimize some scalar loss function p on a common sequence of inputs. This results in a weight vector w_{recurr} for the full recurrent model and weight vector w_{trunc} for the truncated model. We show that for truncation parameter $k \approx O(\log(N/\varepsilon))$, after N steps of gradient descent, the weights of the recurrent and feed-forward model are ε-close in Euclidean distance.

Theorem (Informal version of Theorem 1)**.** *Assuming the system ϕ is λ-contractive and under additional smoothness and Lipschitz assumptions on the system ϕ and the loss p, if*

$$k \geq O\left(\log(N^{1/(1-\lambda)^3}/(\varepsilon(1-\lambda)^2))\right),$$

then after N steps of gradient descent with decaying step size $\alpha_t = O(1/t)$, $\|w_{\text{recurr}} - w_{\text{trunc}}\| \leq \varepsilon$, which in turn implies $\left\|y_t(w_{\text{recurr}}) - y_t^k(w_{\text{trunc}})\right\| \leq O(\varepsilon)$.

The operational interpretation of this theorem is that if it is possible to train a stable recurrent model via gradient descent to perform well on some task, then it's possible to get equally good performance by instead training an autoregressive feed-forward model. In practice the cost of training a fully recurrent model can be prohibitive, in which case truncation is commonly used for computational reasons. Our theorem gives reassurance that this truncation step does not hurt training performance. Contrast this with operations like compression and weight sparsification of a neural net, which done after training do not hurt inference. However, reducing the number of trainable model parameters can certainly make optimization harder.

The decaying step size in our theorem is consistent with the regime in which gradient descent is known to be stable for non-convex training objectives [7]. While the decay is faster than many learning rates encountered in practice, classical results nonetheless show that with this learning rate gradient descent still converges to a stationary point; see p. 119 in [3] and references there.

Our previous results apply to general non-linear dynamical systems. In Section 4, we give sufficient conditions for various different model classes that imply the assumptions of our theorems. In the case of linear dynamical systems, and simple recurrent neural networks these assumptions are easily stated in terms of the spectral norm of the weight matrices and the non-linearities of the system (if present). We also consider LSTM models. Interestingly, LSTMs are not contractive in any obvious way and are likely not stable in general under the assumptions that make RNNs stable. This is witnessed by the need to use heuristics like gradient clipping when working with LSTMs in practice. Despite this obstacle, we provide non-trivial sufficient conditions on the parameterization of these models that imply the assumptions of our theorems. Whether these conditions are also necessary is an interesting open problem.

2.2 Proof overview

To prove the inference result, we show stability and bounded weights imply the states of the system themselves are bounded. This means the difference between the initial state of the truncated system and the state of the full model is bounded, and contractivity then implies after sufficiently many steps, the state difference becomes negligible.

To prove the optimization result, we initialize both the recurrent and truncated models at the same point and then track the difference in weights during training. To do this, we prove (1) the difference in gradients due to truncation vanishes like $O(k\lambda^k)$ and (2) under regularity, the dynamical system is smooth. The first result relies on the vanishing gradient phenomenon– the "long-term" contributions to the gradient vanish exponentially fast. The second result uses stability and a Lipschitz assumption on the weights to argue the states of systems with similar weights are similar, even for large numbers of iterations. These conditions are sufficient to argue gradient descent itself is stable, and this property makes it possible to bound the divergence of weights as training progresses.

2.3 Assumptions

In this section, we collect several assumptions on the system ϕ, the inputs $\{x_t\}$, the prediction function f, and the loss function p that we repeatedly use throughout our analysis.

Our primary assumption is there some compact convex domain $\Omega \subset \mathbf{R}^m$ so that the map ϕ_w is λ-contractive with $\lambda < 1$ for all $w \in \Omega$. Without loss of generality, we also assume $\phi_w(0,0) = 0$ for all w. Otherwise, we can reparameterize $(h, x) \mapsto \phi_w(h, x) - \phi_w(0, 0)$ without affecting expressivity of ϕ_w.

We further assume the map ϕ_w is L_x-Lipschitz in the Euclidean norm with respect to the input x, and, for all reachable states h, the system ϕ_w is L_w-Lipschitz in w. Further, every sequence of inputs $\{x_t\}_{t=1}^T$ is uniformly bounded with $\|x_t\| \leq B_x$. To study optimization, we additionally assume that the map ϕ_w satisfies four smoothness conditions: for any reachable states h, h', and any weights $w, w' \in \Omega$,

1. $\left\| \frac{\partial \phi_w(h,x)}{\partial w} - \frac{\partial \phi_{w'}(h,x)}{\partial w} \right\| \leq \beta_{ww} \|w - w'\|$.
2. $\left\| \frac{\partial \phi_w(h,x)}{\partial w} - \frac{\partial \phi_w(h',x)}{\partial w} \right\| \leq \beta_{wh} \|h - h'\|$.
3. $\left\| \frac{\partial \phi_w(h,x)}{\partial h} - \frac{\partial \phi_{w'}(h,x)}{\partial h} \right\| \leq \beta_{hw} \|w - w'\|$.
4. $\left\| \frac{\partial \phi_w(h,x)}{\partial h} - \frac{\partial \phi_w(h',x)}{\partial h} \right\| \leq \beta_{hh} \|h - h'\|$.

We assume the prediction function f is L_f Lipschitz, and the loss function p is L_p Lipschitz and β_p smooth. To simplify the presentation, the prediction function f is not parameterized. This is without loss of generality because it's always possible to fold the parameters into the system ϕ_w itself. In addition, we assume the initial state of the recurrent model $h_0 = 0$.

2.4 Related work

The connection between stability and a truncated system approximation was exploited in [12] to prove bounds on the number of samples needed to learn a truncated approximation to the full stable system. Their approximation result is the same as our inference result in the *linear* dynamical system case. We extend this result to the non-linear setting and, moreover, exploit the associated vanishing gradient phenomenon to analyze the impact of truncation on training with gradient descent. Results of the latter kind are completely new to our knowledge.

Learning dynamical systems with gradient descent has been a recent topic of interest in the machine learning community. For instance, [6] showed gradient descent can efficiently learn linear dynamical systems. In contrast, our analysis controls the difference between the truncated and full-system solutions obtained by gradient descent. Roughly speaking, these results can be combined with ours to show, when gradient descent succeeds for a class of stable dynamical systems, it succeeds for the truncated systems as well. Work by [11] gives a moment-based approach for learning some classes of non-linear recurrent neural networks.

The vanishing gradient problem was first introduced in [2] and further explored in [10]. Our work is complementary to both of these papers; while they view the vanishing gradient problem primarily as an optimization issue to be overcome, we interpret vanishing gradients as a representational limitation that restricts the power of recurrent architectures. In particular, recurrent models with

vanishing gradients can be well approximated by feed-forward models with limited context. Further, this result applies not just at inference time, but throughout training via gradient descent.

From an empirical perspective, [1] conducted a detailed evaluation of recurrent and convolutional, feed-forward models on a variety of sequence modeling tasks. In diverse settings, they reliably find feed-forward models outperform their recurrent counterparts. However, their work does not offer an explanation for this phenomenon.

We built on the stability analysis of [7], but interestingly use it for an entirely different purpose.

3 Stability

This entire work is conducted under the assumption of stability. While this assumption might seem limiting, it is in fact necessary on two counts. First, without stability, it is easy to construct counterexamples where finite-length truncation can be arbitrarily bad, even for large values of k. This alone rules out both the inference and optimization results without additional assumptions. Second, even in the linear dynamical system case, without stability it is difficult to show gradient descent converges to a stationary point. Indeed, there exists trivial counterexamples where gradient descent fails to converge. Both points are made precise in the propositions below, and the proofs are deferred to the appendix.

Proposition 1. *There exists an unstable system ϕ such that, for any finite truncation length k, $\left\|y_t - y_t^k\right\| \to \infty$ as $t \to \infty$.*

Proposition 2. *There exists a system ϕ_w such that, if w is not constrained to the set Ω where ϕ_w is stable, then gradient descent does not converge to a stationary point, and $\|\nabla_w p_T\| \to \infty$ as the number of iterations $N \to \infty$.*

4 Examples

Our results are stated in the language of general non-linear dynamical systems, and our assumptions are given in terms of a generic state-to-state transition map ϕ. This level of abstraction allows us to separate the core phenomenon that makes approximation possible from the specific parameterization of particular model classes. However, the utility of the theory is in it's application to specific models. In this section, we show how linear dynamical systems, recurrent neural networks, and LSTMs fit into this general framework and give non-trivial sufficient conditions to ensure stability for each class.

4.1 Linear Dynamical Systems

Given matrices $W \in \mathbf{R}^{n \times n}, U \in \mathbf{R}^{n \times d}$, the state-transition map for a linear dynamical system is

$$h_t = W h_{t-1} + U x_t. \tag{4}$$

Using the linear structure of the updates, both the state and the gradients have a particularly simple form:

$$h_t = \sum_{i=0}^{t} W^i U x_{t-i} \quad \text{and} \quad \nabla_{h_t} p_T = W^{T-t} \nabla_{h_T} p_T. \tag{5}$$

As these expressions suggest, the model is stable provided $\|W\| < 1$, and Lipschitz in x provided $\|U\|$ is bounded. Using Lemma (2) below, the model is $O(1/(1 - \|W\|))$ Lipschitz in W, and it's a simple exercise to check that such a system satisfies the remaining Lipschitz and smoothness assumptions.

4.2 Recurrent Neural Networks

Given a Lipschitz, point-wise non-linearity ρ and matrices $W \in \mathbf{R}^{n \times n}$ and $U \in \mathbf{R}^{n \times d}$, the state-transition map for a recurrent neural network (RNN) is

$$h_t = \rho(Wh_{t-1} + Ux_t). \tag{6}$$

This model is stable provided ρ is L_ρ Lipschitz and $\|W\| < \frac{1}{L_\rho}$, and it satisfies our Lipschitz and smoothness assumptions if ρ is smooth and $\|U\| \leq B_U$. Our results do not apply for the non-smooth ReLu non-linearity. For concreteness, let ρ be tanh, which is 1-smooth and 1-Lipschitz.

To see the system is stable for $\|W\| < 1$, for any states h, h'

$$\left\|\tanh(Wh + Ux) - \tanh(Wh' + Ux)\right\| \leq \left\|Wh + Ux - Wh' - Ux\right\| \leq \|W\| \left\|h - h'\right\|.$$

Lemma (2) ensures $\|h_t\| \leq \frac{B_U B_x}{(1-\lambda)}$, so the model is $\frac{B_U B_x}{(1-\lambda)}$ Lipschitz in W over reachable states h. Smoothness follows from the smoothness of ρ. These properties are checked in the appendix.

4.3 Long Short-Term Memory Networks

Long Short-Term Memory (LSTM) networks are another commonly used class of sequence models [8]. In an LSTM, the state is a pair of vectors $s = (c, h) \in \mathbf{R}^{d \times d}$, and the model is parameterized by eight matrices, $W_i \in \mathbf{R}^{d \times d}$ and $U_i \in \mathbf{R}^{d \times n}$ for $i \in \{i, f, o, z\}$. The state-transition map ϕ_{LSTM} is given by

$$\begin{aligned}
f_t &= \sigma(W_f h_{t-1} + U_f x_t) \\
i_t &= \sigma(W_i h_{t-1} + U_i x_t) \\
o_t &= \sigma(W_o h_{t-1} + U_o x_t) \\
z_t &= \tanh(W_z h_{t-1} + U_z x_t) \\
c_t &= i_t \circ z_t + f_t \circ c_t \\
h_t &= o_t \cdot \tanh(c_t),
\end{aligned}$$

where $\circ$ denotes elementwise multiplication and σ is the logistic function. Without loss of generality, assume $s = 0$.

The state-transition map is not Lipschitz in s, much less stable, unless $\|c\|$ is bounded. However, assuming the weights are bounded, we first prove $\|c\|$ is always bounded. Below we denote by $\|W\|_\infty$ the induced ℓ_∞ matrix norm, which corresponds to the maximum absolute row sum $\max_i \sum_j |W_{ij}|$.

Lemma 1. *Let $\|f\|_\infty = \sup_t \|f_t\|_\infty$. If $\|W_f\|_\infty < \infty$, $\|U_f\|_\infty < \infty$, and $\|x_t\| \leq B_x$, then $\|f\|_\infty < 1$ and $\|c_t\|_\infty \leq \frac{1}{(1-\|f\|_\infty)}$ for all t.*

We provide conditions under which the iterated system ϕ^r_{LSTM} is stable. We leave it as an open problem to find different parameter regimes where the system is stable, as well as resolve whether the original system ϕ_{LSTM} is stable.

Proposition 3. *Under the conditions of Lemma* (1)*, if we further assume* $\|W_i\|_\infty, \|W_o\|_\infty < 1 - \|f\|_\infty$*,* $\|W_z\|_\infty \leq (1/4)(1 - \|f\|_\infty)$*,* $\|W_f\|_\infty < (1 - \|f\|_\infty)^2$*, and* $r = O(\log(d))$*, then the iterated system* ϕ^r_{LSTM} *is stable on the set of reachable states.*

The proofs of both Lemma (1) and Proposition (3) are deferred to the appendix. Equipped with several examples of stable recurrent models, we now turn to the proving the main results of the paper.

5 Feed-Forward Approximation

In this section, we demonstrate, for an appropriate choice of context k, the truncated model makes essentially the same predictions as the full recurrent model. We first show the maximum size of the hidden state is bounded, and then we argue contractivity implies the difference between truncated and recurrent hidden states becomes negligible after k steps.

Lemma 2 (No-Blow-Up)**.** *If* ϕ_w *is* λ*-contractive,* L_x *Lipschitz in* x*, and* $\|x_t\| \leq B_x$*, then for all* t*,* $\|h_t\| \leq \frac{L_x B_x}{(1-\lambda)}$*.*

Proof. For any $t \geq 1$, we apply the contracitivity and Lipschitz assumptions and then sum a geometric series,

$$\begin{aligned}
\|h_t\| &= \|\phi_w(h_{t-1}, x_t) - \phi_w(0,0)\| \\
&\leq \|\phi_w(h_{t-1}, x_t) - \phi_w(0, x_t)\| + \|\phi_w(0, x_t) - \phi_w(0,0)\| \\
&\leq \lambda \|h_{t-1}\| + L_x \|x_t\| \\
&\leq \sum_{i=0}^{t} \lambda^i L_x B_x \\
&\leq \frac{L_x B_x}{(1-\lambda)}\,.
\end{aligned}$$

□

The No-Blow-Up lemma guarantees the difference between the recurrent state and the initial state of the truncated model is bounded. Contractivity then implies after sufficiently many steps of both models, this difference becomes very small, as the following lemma makes precise.

Lemma 3. *Assume* ϕ_w *is* λ*-contractive and* L_x *Lipschitz in* x*. Assume the input sequence* $\|x_t\| \leq B_x$ *for all* t*. If* $k \geq \log_{1/\lambda}\left(\frac{L_x B_x}{(1-\lambda)\varepsilon}\right)$*, then the difference in hidden states* $\left\|h_t - h_t^k\right\| \leq \varepsilon$*.*

Proof. Consider the difference between hidden states at time step t. Unrolling the iterates k steps and invoking the no-blow-up property yields

$$\left\|h_t - h_t^k\right\| = \left\|\phi_w(h_{t-1}, x_t) - \phi_w(h_{t-1}^k, x_t)\right\| \leq \lambda \left\|h_{t-1} - h_{t-1}^k\right\| \leq \lambda^k \|h_{t-k}\| \leq \frac{\lambda^k L_x B_x}{(1-\lambda)},$$

and solving for k gives the result. □

If the prediction function is Lipschitz, then this result immediately implies that the predictions between the recurrent and truncated model are nearly identical.

Proposition 4. *If* ϕ_w *is a* L_x*-Lipschitz and* λ*-contractive map, and* f *is* L_f *Lipschitz and* $k \geq \log_{1/\lambda}\left(\frac{L_f L_x B_x}{(1-\lambda)\varepsilon}\right)$*, then* $\left\|y_t - y_t^k\right\| \leq \varepsilon$*.*

6 Vanishing Gradients

One way to interpret the results of the previous section is the dependence on distant inputs is limited in stable recurrent models. To make this connection precise, we show the gradient of a scalar loss at step T with respect to x_0 vanishes exponentially fast as T becomes large. These results are similar to those derived in [10].

Lemma 4. *Let p_T be an L_p Lipschitz loss evaluated at step T, and assume ϕ_w is λ-contractive and L_x Lipschitz in x, then $\|\nabla_{x_0} p_T\| \leq L_p L_x \lambda^T$.*

Proof. Writing out the gradient and using sub-multiplicativity of the spectral norms,

$$\|\nabla_{x_0} p_T\| = \left\| \frac{\partial h_0}{\partial x_0}^\top \frac{\partial h_T}{\partial h_0}^\top \nabla_{h_T} p_T \right\| \leq \left\| \frac{\partial h_0}{\partial x_0} \right\| \left\| \frac{\partial h_T}{\partial h_0} \right\| \|\nabla_{h_T} p_T\| \ .$$

By the Lipschitz assumptions, $\left\| \frac{\partial h_0}{\partial x_0} \right\| \leq L_x$ and $\|\nabla_{h_T} p_T\| \leq L_p$. Since ϕ_w is λ-contractive, $\left\| \frac{\partial h_t}{\partial h_{t-1}} \right\| \leq \lambda$. Putting this together,

$$\|\nabla_{x_0} p_T\| \leq L_p L_x \left\| \frac{\partial h_T}{\partial h_0} \right\| = L_p L_x \left\| \prod_{0<i\leq T} \frac{\partial h_t}{\partial h_{t-1}} \right\| \leq L_p L_x \prod_{0<i\leq T} \left\| \frac{\partial h_t}{\partial h_{t-1}} \right\| \leq L_p L_x \lambda^T.$$

□

We can similarly show that the gradient with respect to the weights at distant-time steps is small. The Jacobian of the loss with respect to the weights is

$$\frac{\partial p_T}{\partial w} = \frac{\partial p_T}{\partial h_T} \left(\sum_{t=0}^{T} \frac{\partial h_T}{\partial h_t} \frac{\partial h_t}{\partial w} \right), \tag{7}$$

where $\frac{\partial h_t}{\partial w}$ is the partial derivative of h_t with respect to w, assuming h_{t-1} is constant with respect to w. We call the terms in (7) for steps $t = 0$ to $t = T - k$ the "long-term components" of the gradient.

Proposition 5. *If p_T is an L_p Lipschitz loss, ϕ_w is λ-contractive and L_w Lipschitz in w, then the long term components of the gradient vanish as $k \to \infty$, namely $\left\| \sum_{t=0}^{T-k} \frac{\partial p_T}{\partial h_t} \frac{\partial h_t}{\partial w} \right\| \leq \lambda^k \frac{L_p L_w}{(1-\lambda)}$.*

Proof. The contribution to the gradient for steps $t = 0$ to $t = T - k$ is

$$\frac{\partial p_T}{\partial h_T} \left(\sum_{t=0}^{T-k} \frac{\partial h_T}{\partial h_t} \frac{\partial h_t}{\partial w} \right).$$

Taking norms and using the Lipschitz and contractive assumptions,

$$\|\nabla_{h_T} p_T\| \left\| \sum_{t=0}^{T-k} \frac{\partial h_T}{\partial h_t} \frac{\partial h_t}{\partial w} \right\| \leq \|\nabla_{h_T} p_T\| \, \|\nabla_w h_t\| \sum_{t=0}^{T-k} \left\| \frac{\partial h_T}{\partial h_t} \right\| \leq L_p L_w \sum_{t=0}^{T-k} \lambda^{T-t} \leq \lambda^k \frac{L_p L_w}{(1-\lambda)}.$$

□

7 Gradient Descent Analysis

In this section, we study the gradient descent in stable recurrent models. Our goal is to show the recurrent model and truncated models found by running gradient descent make essentially the same predictions.

At a high-level, our proof technique is to initialize both the recurrent and truncated models at the same point and track the divergence in weights throughout the course of gradient descent. Roughly, we show if $k \approx O(\log(N/\varepsilon))$, then after N steps of gradient descent, the difference in the weights between the recurrent and truncated models is at most ε.

Even if the gradients are similar for both models at the same point, it's a priori possible that slight differences in the gradients accumulate over time and lead to divergent weights where no meaningful comparison is possible. Building on similar techniques as [7], we show that gradient descent itself is stable and this type of divergence cannot occur.

We begin by stating two essential lemmas. The first bounds the difference in gradient as a result of running the truncated model rather than full model. The second establishes the gradient map of the full and recurrent models is Lipschitz.

Lemma 5. *Let p_T^k denote the loss function evaluated on truncated model. Assume p (and therefore p_T^k) are Lipschitz and smooth. Assume ϕ_w is λ-contractive, Lipschitz in x and w, and satisfies smoothness conditions 1-4. Assume the inputs satisfy $\|x_t\| \leq B_x$, then*

$$\left\|\nabla_w p_T - \nabla_w p_T^k\right\| = \gamma k \lambda^k, \tag{8}$$

where $\gamma = O\left(\frac{B_x}{(1-\lambda)^2}\right)$, suppressing dependence on the Lipschitz and smoothness parameters.

Lemma 6. *For any $w, w' \in \Omega$, suppose ϕ_w is λ-contractive, Lipschitz in w, and satisfies smoothness conditions 1-4. If p is Lipschitz and smooth, then*

$$\left\|\nabla_w p_T(w) - \nabla_w p_T(w')\right\| \leq \beta \left\|w - w'\right\|, \tag{9}$$

where $\beta = O\left(\frac{1}{(1-\lambda)^3}\right)$, suppressing dependence on the Lipschitz and smoothness parameters.

Lemma (5) is proved in the next subsection, and Lemma (6) is proved in the appendix. We now proceed prove our main gradient descent result. Let w_{recurr}^i be the weights of the recurrent model on step i and define w_{trunc}^i similarly for the truncated model. At initialization, $w_{\text{recurr}}^0 = w_{\text{trunc}}^0$.

Proposition 6. *Under the assumptions of Lemmas (5) and (6), for compact, convex Ω, after N steps of projected gradient descent with step size $\alpha_t = \alpha/t$, $\left\|w_{\text{recurr}}^N - w_{\text{trunc}}^N\right\| \leq \alpha\gamma k \lambda^k N^{\alpha\beta+1}$.*

Proof. Let Π_Ω denote the Euclidean projection onto Ω, which we assume can be efficiently evaluated. Let $\delta_i = \left\|w_{\text{recurr}}^i - w_{\text{trunc}}^i\right\|$. Initially $\delta_0 = 0$, and on step $i+1$, we have the following recurrence

relation for δ_{i+1},

$$\begin{aligned}
\delta_{i+1} &= \left\|w^{i+1}_{\text{recurr}} - w^{i+1}_{\text{trunc}}\right\| \\
&= \left\|\Pi_\Omega(w^i_{\text{recurr}} - \alpha_i \nabla p_T(w^i)) - \Pi_\Omega(w^i_{\text{trunc}} - \alpha_i \nabla p^k_T(w^i_{\text{trunc}}))\right\| \\
&= \left\|w^i_{\text{recurr}} - \alpha_i \nabla p_T(w^i)) - w^i_{\text{trunc}} - \alpha_i \nabla p^k_T(w^i_{\text{trunc}})\right\| \\
&\leq \left\|w^i_{\text{recurr}} - w^i_{\text{trunc}}\right\| + \alpha_i \left\|\nabla p_T(w^i_{\text{recurr}}) - \nabla p^k_T(w^i_{\text{trunc}})\right\| \\
&\leq \delta_i + \alpha_i \left\|\nabla p_T(w^i_{\text{recurr}}) - \nabla p_T(w^i_{\text{trunc}})\right\| \\
&\quad + \alpha_i \left\|\nabla p_T(w^i_{\text{trunc}}) - \nabla p^k_T(w^i_{\text{trunc}})\right\| \\
&\leq \delta_i + \alpha_i \left(\beta \delta_i + \gamma k \lambda^k\right) \\
&\leq \exp\left(\alpha_i \beta\right) \delta_i + \alpha_i \gamma k \lambda^k,
\end{aligned}$$

the penultimate line applied lemmas (5) and (6), and the last line used $1 + x \leq e^x$ for all x. Unwinding the recurrence relation at step N,

$$\begin{aligned}
\delta_N &\leq \sum_{i=1}^N \left\{ \prod_{j=i+1}^N \exp(\alpha_j \beta) \right\} \alpha_i \gamma k \lambda^k \\
&\leq \sum_{i=1}^N \left\{ \prod_{j=i+1}^N \exp\left(\frac{\alpha\beta}{j}\right) \right\} \frac{\alpha \gamma k \lambda^k}{i} \\
&= \sum_{i=1}^N \left\{ \exp\left(\alpha\beta \sum_{j=i+1}^N \frac{1}{j}\right) \right\} \frac{\alpha \gamma k \lambda^k}{i} \\
&\leq \sum_{i=1}^N \exp(\alpha\beta \log(N/i)) \frac{\alpha \gamma k \lambda^k}{i} \\
&= \alpha \gamma k \lambda^k N^{\alpha\beta} \sum_{i=1}^N \frac{1}{i^{\alpha\beta+1}} \\
&\leq \alpha \gamma k \lambda^k N^{\alpha\beta+1}.
\end{aligned}$$

□

Given the previous result, if we take $\alpha = 1$ and $k = O\left(\log(\gamma N^\beta/\varepsilon)\right)$, then after N steps of projected gradient descent, $\left\|w^N_{\text{recurr}} - w^N_{\text{trunc}}\right\| \leq \varepsilon$.

To translate control over the weight difference into control over the predictions, we first show small differences in weights don't significantly change the trajectory of the recurrent model.

Lemma 7. *For some w, w', suppose $\phi_w, \phi_{w'}$ are λ-contractive and L_w Lipschitz in w. Let $h_t(w), h_t(w')$ be the hidden state at time t obtain from running the model with weights w, w' on common inputs $\{x_t\}$. If $h_0(w) = h_0(w')$, then*

$$\left\|h_t(w) - h_t(w')\right\| \leq \frac{L_w \left\|w - w'\right\|}{(1-\lambda)}. \tag{10}$$

Proof. Since both models are initialized at a common point, we can repeatedly apply Lipschitz and contractivity to obtain a geometric series in λ

$$\begin{aligned}
\left\|h_t(w) - h_t(w')\right\| &= \left\|\phi_w(h_{t-1}(w), x_t) - \phi_{w'}(h_{t-1}(w'), x_t)\right\| \\
&\leq \|\phi_w(h_{t-1}(w), x_t) - \phi_{w'}(h_{t-1}(w), x_t)\| + \left\|\phi_{w'}(h_{t-1}(w), x_t) - \phi_{w'}(h_{t-1}(w'), x_t)\right\| \\
&\leq L_w \left\|w - w'\right\| + \lambda \left\|h_{t-1}(w) - h_{t-1}(w')\right\| \\
&\leq \sum_{i=0}^{t} L_w \left\|w - w'\right\| \lambda^i \\
&\leq \frac{L_w \|w - w'\|}{(1-\lambda)}.
\end{aligned}$$

□

Putting each of the previous pieces together, we obtain the main theorem.

Theorem 1. *Let p be Lipschitz and smooth. Assume ϕ_w is λ-contractive, Lipschitz in x and w, and satisfies smoothness conditions 1-4. Assume the inputs are bounded, and the prediction function f is L_f-Lipschitz. If $k = O(\log(\gamma N^\beta/\varepsilon))$, then $\left\|y_T - y_T^k\right\| \leq \varepsilon$.*

Proof. Combining Lemmas (3) and (7) via the triangle inequality, at step T,

$$\left\|h_T(w) - h_T^k(w')\right\| \leq \left\|h_T(w) - h_T(w')\right\| + \left\|h_T(w') - h_T^k(w')\right\| \leq \frac{L_w \|w - w'\|}{(1-\lambda)} + \lambda^k \frac{L_x B_x}{(1-\lambda)}.$$

Since f is L_f-Lipschitz assumption, the prediction error

$$\begin{aligned}
\left\|y_T - y_T^k\right\| &\leq L_f \left\|h_T(w_{\text{recurr}}^N) - h_T^k(w_{\text{trunc}}^N)\right\| \\
&\leq \frac{L_f L_w \left\|w_{\text{recurr}}^N - w_{\text{trunc}}^N\right\|}{(1-\lambda)} + \lambda^k \frac{L_f L_x B_x}{(1-\lambda)} \\
&\leq \frac{L_f L_w \alpha\lambda k \lambda^k N^{\alpha\beta+1}}{(1-\lambda)} + \lambda^k \frac{L_f L_x B_x}{(1-\lambda)},
\end{aligned}$$

and solving for k such that both terms are less than $\varepsilon/2$ gives the result. □

7.1 Truncation

In the section, we argue the difference in gradient with respect to the weights between the recurrent and truncated models is $O(k\lambda^k)$. Hence, for k sufficiently large (independent of the sequence length), the impact of truncation is negligible.

At a high level, the long-term gradient components quickly vanish, so the main challenge is to show the short term gradient components are similar.

Proof of Lemma 5. Expanding the expression for the gradient, we wish to bound

$$\begin{aligned}
&\left\|\nabla_w p_T(w) - \nabla_w p_T^k(w)\right\| \\
&= \left\|\sum_{t=1}^{T} \left(\frac{\partial h_T}{\partial h_t}\frac{\partial h_t}{\partial w}\right)^\top \nabla_{h_T} p_T - \sum_{t=T-k+1}^{T} \left(\frac{\partial h_T^k}{\partial h_t^k}\frac{\partial h_t^k}{\partial w}\right)^\top \nabla_{h_T^k} p_T^k\right\| \\
&\leq \left\|\sum_{t=1}^{T-k} \left(\frac{\partial h_T}{\partial h_t}\frac{\partial h_t}{\partial w}\right)^\top \nabla_{h_T} p_T\right\| + \sum_{t=T-k+1}^{T} \left\|\left(\frac{\partial h_T}{\partial h_t}\frac{\partial h_t}{\partial w}\right)^\top \nabla_{h_T} p_T - \left(\frac{\partial h_T^k}{\partial h_t^k}\frac{\partial h_t^k}{\partial w}\right)^\top \nabla_{h_T^k} p_T\right\|.
\end{aligned}$$

By Proposition (5), the first term is bounded by $\lambda^k \frac{L_p L_w}{(1-\lambda)}$. Focusing on the second term,

$$\begin{aligned}
&\sum_{t=T-k+1}^{T} \left\| \left(\frac{\partial h_T}{\partial h_t} \frac{\partial h_t}{\partial w} \right)^\top \nabla_{h_T} p_T - \left(\frac{\partial h_T^k}{\partial h_t^k} \frac{\partial h_t^k}{\partial w} \right)^\top \nabla_{h_T^k} p_T \right\| \\
&\leq \sum_{t=T-k+1}^{T} \left\| \nabla_{h_T} p_T - \nabla_{h_T^k} p_T^k \right\| \left\| \frac{\partial h_T^k}{\partial h_t^k} \frac{\partial h_t^k}{\partial w} \right\| + \left\| \nabla_{h_T} p_T \right\| \left\| \frac{\partial h_T}{\partial h_t} \frac{\partial h_t}{\partial w} - \frac{\partial h_T^k}{\partial h_t^k} \frac{\partial h_t^k}{\partial w} \right\| \\
&\leq \sum_{t=T-k+1}^{T} \underbrace{\beta_p \left\| h_T - h_T^k \right\| \lambda^{T-t} L_w}_{(a)} + \underbrace{L_p \left\| \frac{\partial h_T}{\partial h_t} \frac{\partial h_t}{\partial w} - \frac{\partial h_T^k}{\partial h_t^k} \frac{\partial h_t^k}{\partial w} \right\|}_{(b)}.
\end{aligned}$$

Using Lemma (3) to upper bound (a),

$$\sum_{t=T-k}^{T} \beta_p \left\| h_T - h_T^k \right\| \lambda^{T-t} L_w \leq \sum_{t=T-k}^{T} \lambda^{T-t} \frac{\lambda^k \beta_p L_w L_x B_x}{(1-\lambda)} \leq \frac{\lambda^k \beta_p L_w L_x B_x}{(1-\lambda)^2}.$$

Using the triangle inequality, Lipschitz and smoothness, (b) is bounded by

$$\begin{aligned}
&\sum_{t=T-k+1}^{T} L_p \left\| \frac{\partial h_T}{\partial h_t} \frac{\partial h_t}{\partial w} - \frac{\partial h_T^k}{\partial h_t^k} \frac{\partial h_t^k}{\partial w} \right\| \\
&\leq \sum_{t=T-k+1}^{T} L_p \left\| \frac{\partial h_T}{\partial h_t} \right\| \left\| \frac{\partial h_t}{\partial w} - \frac{\partial h_t^k}{\partial w} \right\| + L_p \left\| \frac{\partial h_t^k}{\partial w} \right\| \left\| \frac{\partial h_T}{\partial h_t} - \frac{\partial h_T^k}{\partial h_t^k} \right\| \\
&\leq \sum_{t=T-k+1}^{T} L_p \lambda^{T-t} \beta_{wh} \left\| h_t - h_t^k \right\| + L_p L_w \left\| \frac{\partial h_T}{\partial h_t} - \frac{\partial h_T^k}{\partial h_t^k} \right\| \\
&\leq k \lambda^k \frac{L_p \beta_{wh} L_x B_x}{(1-\lambda)} + \underbrace{L_p L_w \sum_{t=T-k+1}^{T} \left\| \frac{\partial h_T}{\partial h_t} - \frac{\partial h_T^k}{\partial h_t^k} \right\|}_{(c)},
\end{aligned}$$

where the last line used $\left\| h_t - h_t^k \right\| \leq \lambda^{t-(T-k)} \frac{L_x B_x}{(1-\lambda)}$ for $t \geq T-k$. It remains to bound (c), the difference of the hidden-to-hidden Jacobians. Peeling off one term at a time and applying triangle inequality, for any $t \geq T-k+1$,

$$\begin{aligned}
\left\| \frac{\partial h_T}{\partial h_t} - \frac{\partial h_T^k}{\partial h_t^k} \right\| &\leq \left\| \frac{\partial h_T}{\partial h_{T-1}} - \frac{\partial h_T^k}{\partial h_{T-1}^k} \right\| \left\| \frac{\partial h_{T-1}}{\partial h_t} \right\| + \left\| \frac{\partial h_T^k}{\partial h_{T-1}^k} \right\| \left\| \frac{\partial h_{T-1}}{\partial h_t} - \frac{\partial h_{T-1}^k}{\partial h_t^k} \right\| \\
&\leq \beta_{hh} \left\| h_{T-1} - h_{T-1} \right\| \lambda^{T-t-1} + \lambda \left\| \frac{\partial h_{T-1}}{\partial h_t} - \frac{\partial h_{T-1}^k}{\partial h_t^k} \right\| \\
&\leq \sum_{i=t}^{T-1} \beta_{hh} \lambda^{T-t-1} \left\| h_i - h_i^k \right\| \\
&\leq \lambda^k \frac{\beta_{hh} L_x B_x}{(1-\lambda)} \sum_{i=t}^{T-1} \lambda^{i-t} \\
&\leq \lambda^k \frac{\beta_{hh} L_x B_x}{(1-\lambda)^2},
\end{aligned}$$

so (c) is bounded by $k\lambda^k \frac{L_p L_w \beta_{hh} L_x B_x}{(1-\lambda)^2}$. Ignoring Lipschitz and smoothness constants, we've shown the entire sum is $O\left(\frac{k\lambda^k}{(1-\lambda)^2}\right)$. □

8 Experiments

In this section, we first verify the conclusions of our theoretical investigations on synthetic data, and then we demonstrate the same phenomenon hold for models trained on a benchmark language modeling task.

8.1 Synthetic Data

Our goal in this section is to empirically verify the bounds obtained in Section (7). Using random instances, we first check the conclusion of Lemma (5) and show the difference in gradients due to truncation at length k indeed scales as $k\lambda^k$ for both linear dynamical systems and recurrent neural networks. Then we show the bound on parameter error from running gradient descent given in Proposition (6) also has the correct scaling.

Truncation. To test the conclusions of our truncation lemma, we generate random instances as follows. Fix a sequence length $T = 1000$, generate random Gaussian input data $x_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 0.1 \cdot I_{32})$, and generate a random target $y_T \sim \text{Unif}[-1, 1]$. Then, sample parameters $W, U \in \mathbf{R}^{32\times 32}$ for a linear dynamical system or a recurrent neural networks with tanh non-linearity using $U_{ij}, W_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1)$. Finally, set $\lambda = 0.75$ and threshold the singular values of W so $\|W\| \leq \lambda$. We use the squared loss, and take $f(h_t, x_t) = Ch_t + Dx_t$ as our prediction function, where $C, D \in \mathbf{R}^{32\times 1}$ are sampled $C, D \sim \mathcal{N}(0, I_{32})$. In Figure (1), we plot $\left\|\nabla_W p_T - \nabla_W p_T^k\right\|$ as k varies (averaged over 10 runs) for both a linear dynamical system and a recurrent neural network, and we find the error closely matches the $k\lambda^k$ scaling predicted by our bound.

Gradient Descent. To check the conclusions of the gradient descent bound, we randomly generate instances and initialize model parameters as in the truncation experiment. We fix the truncation length to $k = 35$, set the learning rate to $\alpha_t = \alpha/t$ for $\alpha = 0.01$, and take $N = 200$ gradient steps. (These parameters are chosen so that the $\gamma k \lambda^k N^{\alpha\beta+1}$ bound does not become vacuous – by triangle inequality, we always have $\|W_{\text{recurr}} - W_{\text{trunc}}\| \leq 2\lambda$). In Figure (2), we plot the parameter error $\|W_{\text{recurr}} - W_{\text{trunc}}\|$ as training progresses for both a linear dynamical system and a recurrent neural network with tanh non-linearities, and we find the error scales comparably with the bound given in Proposition (6).

8.2 Language Modeling

In this section, we investigate our theoretical conclusions on real data and systems of practical relevance. We train both LSTM and tanh recurrent neural network language models on the WikiText-2 dataset [9] using publicly available code.[1] First, we show there exist models with respectable performance on the language modeling task that satisfy the sufficient conditions for stability given in Section (4) along with the other assumptions of our theorems. Second, we show the same

[1] `https://github.com/pytorch/examples/tree/master/word_language_model`

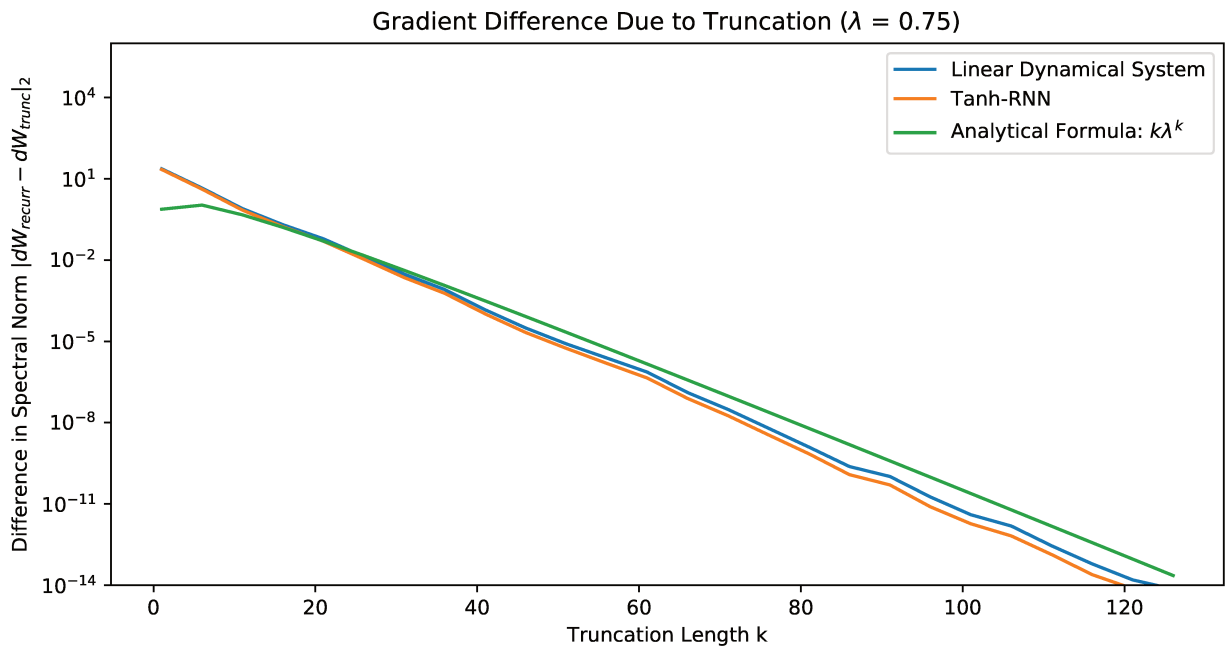

Figure 1: Empirical comparison of the gradient error caused by truncation as k varies. On random Gaussian instances, the observed difference in gradients closely matches the $k\lambda^k$ rate predicted by Lemma (5).

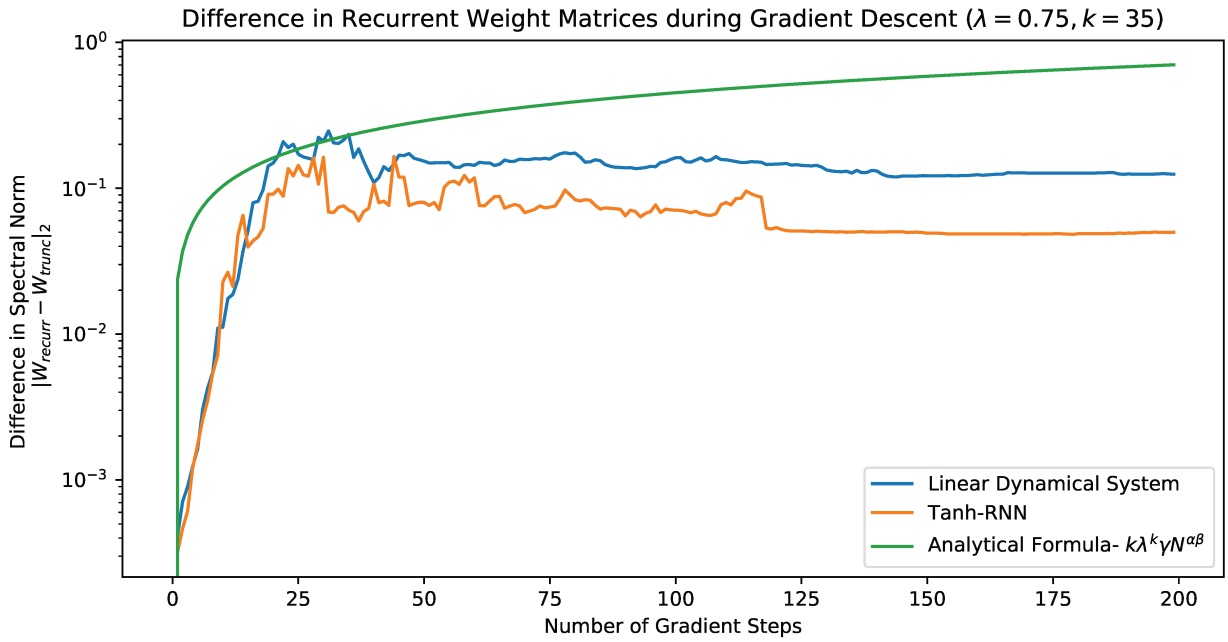

Figure 2: Empirical comparison of the parameter error $\|w_{\text{recurr}} - w_{\text{trunc}}\|$ during gradient descent. On random Gaussian instances, the observed parameter error scales similarly with the $k\lambda^k\gamma N^{\alpha\beta+1}$ rate predicted by Proposition (6).

phenomenon of vanishing gradients and truncated approximation appear in settings not directly captured by our theoretical results.

Stable Recurrent Models. Recurrent models trained in practice are not a-priori stable, so it is natural to consider the impact of imposing the stability assumption during training. To understand these effects, we trained two recurrent neural networks. Both models consist of a single single recurrent layer with tanh nonlinearity, are trained for 40 epochs using a sequence length of 50, use embedding and hidden state dimensions of 1500, dropout 0.65, and an initial learning rate of 5. The first model is otherwise unconstrained, and the second model is constrained to $\|W\| \leq 1$, which assures stability by Section (4). Concretely, after each gradient update, we project the hidden-to-hidden matrix W onto the spectral norm ball by computing the SVD and thresholding the singular values to lie in $[0, 1)$. The projection step is computationally expensive, and the constrained model takes an order of magnitude longer to train. All of the hyperparameters were chosen via grid-search to maximize the performance of the unconstrained model. However, at convergence, there appears to be little difference between the two models. The unconstrained model achieves a final test perplexity of **168.7**, whereas the stable, constrained model achieves a final test perplexity of **149.6**. This result suggests the stability assumption is not overly restrictive, and our theory applies to models that achieve respectable performance on a common benchmark task.

Vanishing Gradients. LSTMs and recurrent neural networks trained in practice exhibit vanishing gradients and limited sensitivity to past inputs well-beyond what's guaranteed by our theorems. Vanishing gradients are not merely encountered in pathological models that fail to train. Even models that achieve competitive performance on benchmark tasks exhibit this phenomenon.

In this experiment, we trained LSTM and recurrent neural networks models. The recurrent model hyperparameters are the same as the previous experiment, and the LSTM model consists of a single layer, uses embedding and hidden state dimensions of 1500, is trained for 40 epochs, and otherwise uses the default hyperparameters. The stability constraints on the weight matrices were not enforced in either case.

At the end of every epoch, we computed $\|\nabla_{x_t} p_{t+i}\|$ for $i = 1, \ldots, 50$ for t ranging over the entire validation set. The results are shown in Figure (3).

The LSTM and the RNN both suffer from limited sensitivity to distant inputs at initialization and throughout training. The gradients of the LSTM vanish more slowly than those of the RNN, but both models exhibit the same qualitative behavior. Intriguingly, as training progresses the rate of decay decreases, which is a not a phenomenon captured by our theory. Moreover, in neither case does the spectral norm assumption obtain– in the RNN case, $\|W\| \approx 7.9$ at convergence. Better understanding this training phenomenon and finding more general conditions either on the weight matrices or the data distribution that lead to vanishing gradients and approximation by feed-forward models is a challenge for future work.

Truncated Approximation. Even in settings not captured by our existing results, recurrent models can be approximated by feed-forward networks for inference and training. To demonstrate this phenomenon, we took the RNN described in the preceeding section and studied the impact of truncating the model for $k = k, 10, 15, 25, 35, 50, 65$. We assume $k = 65$ well captures the full-recurrent model. All of the models were initialized at the same point, and we tracked the distance between the hidden-to-hidden matrices W as training progress. In Figure (4), we plot $\|W_k - W_{65}\|$

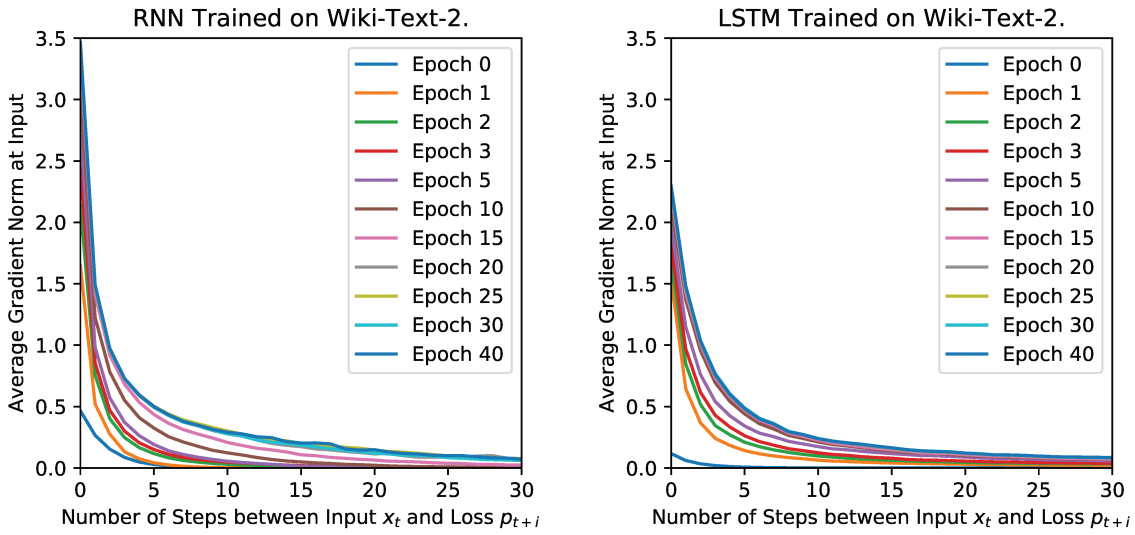

Figure 3: Norm of the loss gradient with respect to inputs, $\|\nabla_{x_t} p_{t+i}\|$, as the distance between the input and the loss grows, averaged over the entire held-out set. The gradient vanishes for moderate values of i in both the RNN and the LSTM case. The LSTM achieves a perplexity of 92.3, and the RNN achieves a perplexity of 168.7.

for $k = 5, 10, 15, 25, 35, 50, 64$ as training proceeds. After an initial rapid increase in distance, $\|W_k - W_{65}\|$ grows slowly as training continues. As our theory suggests, there is a diminishing return to choosing larger values of the truncation parameter k.

References

[1] S. Bai, J. Z. Kolter, and V. Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[2] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[3] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

[4] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier. Language modeling with gated convolutional networks. In *International Conference on Machine Learning*, pages 933–941, 2017.

[5] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin. Convolutional sequence to sequence learning. In *International Conference on Machine Learning*, pages 1243–1252, 2017.

[6] M. Hardt, T. Ma, and B. Recht. Gradient descent learns linear dynamical systems. *arXiv preprint arXiv:1609.05191*, 2016.

[7] M. Hardt, B. Recht, and Y. Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, pages 1225–1234, 2016.

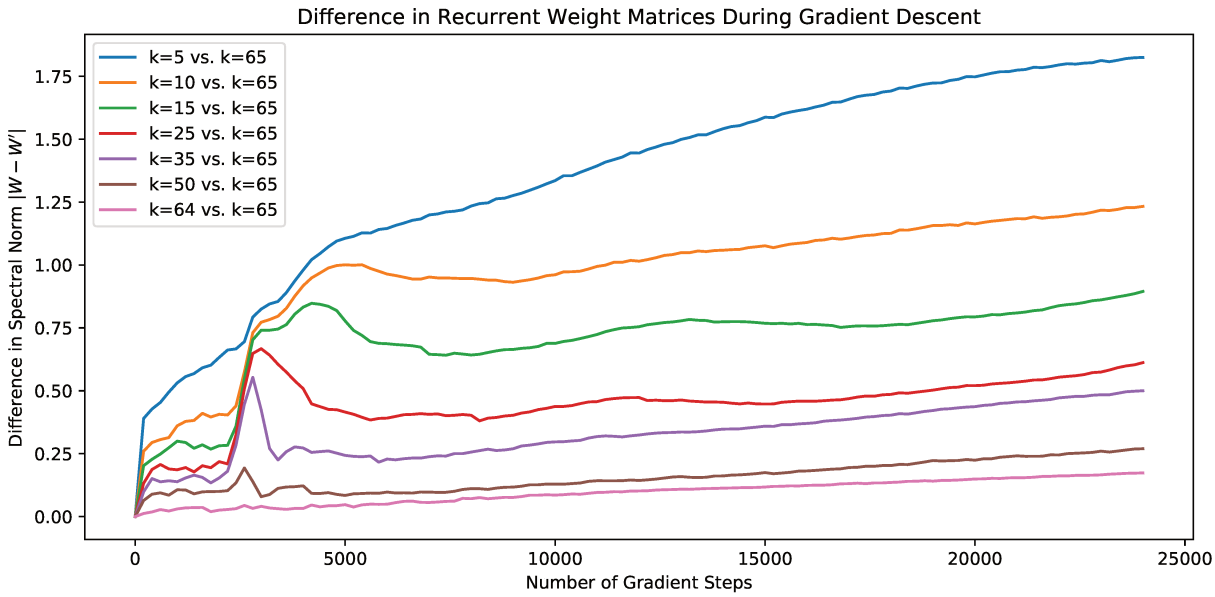

Figure 4: Distance in spectral norm between the hidden-to-hidden matrices of truncated RNNs trained with different values of the truncation parameter k during gradient descent. All models were initialized at the same point, and all other hyperparameters and random seeds are shared between the models. For reference, $\|W_{65}\| = 2.95$ at convergence.

[8] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[9] S. Merity, C. Xiong, J. Bradbury, and R. Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

[10] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318, 2013.

[11] H. Sedghi and A. Anandkumar. Training input-output recurrent neural networks through spectral methods. *CoRR*, abs/1603.00954, 2016.

[12] S. Tu, R. Boczar, A. Packard, and B. Recht. Non-asymptotic analysis of robust control from coarse-grained identification. *arXiv preprint arXiv:1707.04791*, 2017.

[13] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010, 2017.

A Missing Proofs

A.1 Proofs from Section (3)

In both of the following proofs, we consider the simple example of a scalar linear dynamical system given by

$$h_t = ah_{t-1} + bx_t$$
$$\hat{y}_t = h_t,$$

where $h_0 = 0$, $a, b \in \mathbf{R}$ are parameters, and $x_t, y_t \in \mathbf{R}$ are elements the input-output sequence $\{(x_t, y_t)\}_{t=1}^T$, where L is the sequence length, and $\hat{y}_t$ is the prediction at time t.

Stability of the above system corresponds to $|a| < 1$. If the system is not stable, then finite-length truncation can be arbitrarily bad.

Proof of Proposition (1). Suppose $a = 2$, $b = 1$, and the inputs $x_0 = 1$ and $x_t = 0$ for $i \geq 1$. Fix any truncation length k. At time step $t \geq k + c + 1$ for any $c \geq 0$, $h_t^k = 0$, and the prediction of the truncated model is $y_t^k = 0$. However, for the full model, $\hat{y}_t = h_t = 2^{t-1} = 2^{k+c}$. Sending $c \to \infty$, $\left\|y_t^k - \hat{y}_t\right\| = 2^{k+c} \to \infty$. □

Without stability or further assumptions, "exploding gradients" make analysis of gradient descent untenable.

Proof of Proposition (2). Suppose $(x_t, y_t) = (1, 1)$ for $t = 1, \ldots, L$. Then the desired system (11) simply computes the identity mapping. Suppose we use the squared-loss $\ell(y_t, \hat{y}_t) = (1/2)(y_t - \hat{y}_t)^2$, and suppose further $b = 1$, so the problem reduces to finding $a = 0$. We first compute the gradient. Compactly write

$$h_t = \sum_{i=0}^{t-1} a^t b = \left(\frac{1 - a^t}{1 - a}\right). \tag{11}$$

Let $\delta_t = (\hat{y}_t - y_t)$. The gradient for step T is then

$$\frac{d}{da}\ell(y_T, \hat{y}_T) = \delta_T \frac{d}{da} = \delta_T \sum_{t=0}^{T-1} a^{T-1-t} h_t \tag{12}$$
$$= \delta_T \sum_{t=0}^{T-1} a^{T-1-t} \left(\frac{1 - a^t}{1 - a}\right) \tag{13}$$
$$= \delta_T \left[\frac{1}{(1-a)} \sum_{t=0}^{T-1} a^t - \frac{Ta^{T-1}}{(1-a)}\right] \tag{14}$$
$$= \delta_T \left[\frac{(1 - a^T)}{(1-a)^2} - \frac{Ta^{T-1}}{(1-a)}\right]. \tag{15}$$

Plugging in $y_t = 1$, this becomes

$$\frac{d}{da}\ell(y_T, \hat{y}_T) = \left(\frac{(1 - a^T)}{(1-a)} - 1\right) \left[\frac{(1 - a^T)}{(1-a)^2} - \frac{Ta^{T-1}}{(1-a)}\right]. \tag{16}$$

For large T, if $|a| > 1$, then a^L grows exponentially with T and the gradient is approximately

$$\frac{d}{da}\ell(y_T, \hat{y}_T) \approx \left(a^{T-1} - 1\right) T a^{T-2} \approx T a^{2T-3} \tag{17}$$

Therefore, if a^0 is initialized outside of $[-1, 1]$, the iterates a^i from gradient descent with step size $\alpha_i = (1/i)$ diverge, i.e. $a^i \to \infty$, and from equation (16), it is clear that such a^i are not stationary points. □

A.2 Proof from Section (4)

Details of Section 4.2. Assume $\|W\| \leq \lambda < 1$ and $\|U\| \leq B_U$. Notice $\tanh'(x) = 1 - \tanh(x)^2$, so since $\tanh(x) \in [-1, 1]$, $\tanh(x)$ is 1-Lipschitz and 2-smooth. We previously showed the system is stable since, for any states h, h',

$$\begin{aligned}
&\left\|\tanh(Wh + Ux) - \tanh(Wh' + Ux)\right\| \\
&\leq \left\|Wh + Ux - Wh' - Ux\right\| \\
&\leq \|W\| \left\|h - h'\right\|.
\end{aligned}$$

Lemma (2) ensures $\|h_t\| \leq \frac{B_U B_x}{(1-\lambda)}$ for all t. Therefore, for any W, W', U, U',

$$\begin{aligned}
&\left\|\tanh(Wh_t + Ux) - \tanh(W'h_t + U'x)\right\| \\
&\leq \left\|Wh_t + Ux - W'h_t - U'x\right\| \\
&\leq \sup_t \|h_t\| \left\|W - W'\right\| + B_x \left\|U - U'\right\|. \\
&\leq \frac{B_U B_x}{(1-\lambda)} \left\|W - W'\right\| + B_x \left\|U - U'\right\|,
\end{aligned}$$

so the model is Lipschitz in U, W. We can similarly argue the model is B_U Lipschitz in x. For smoothness, the partial derivative with respect to h is

$$\frac{\partial \phi_w(h, x)}{\partial h} = \mathbf{diag}(\tanh'(Wh + Ux))W,$$

so for any h, h', bounding the ℓ_∞ norm with the ℓ_2 norm,

$$\begin{aligned}
\left\|\frac{\partial \phi_w(h, x)}{\partial h} - \frac{\partial \phi_w(h', x)}{\partial h}\right\| &= \left\|\mathbf{diag}(\tanh'(Wh + Ux))W - \mathbf{diag}(\tanh'(Wh' + Ux))W\right\| \\
&\leq \|W\| \left\|\mathbf{diag}(\tanh'(Wh + Ux) - \tanh'(Wh' + Ux))\right\| \\
&\leq 2\|W\| \left\|Wh + Ux - Wh' - Ux\right\|_\infty \\
&\leq 2\lambda^2 \left\|h - h'\right\|.
\end{aligned}$$

For any W, W', U, U' satisfying our assumptions,

$$\begin{aligned}
\left\|\frac{\partial \phi_w(h, x)}{\partial h} - \frac{\partial \phi_{w'}(h, x)}{\partial h}\right\| &= \left\|\mathbf{diag}(\tanh'(Wh + Ux))W - \mathbf{diag}(\tanh'(W'h + U'x))W'\right\| \\
&\leq \left\|\mathbf{diag}(\tanh'(Wh + Ux) - \tanh'(W'h + U'x))\right\| \|W\| + \left\|\mathbf{diag}(\tanh'(W'h + U'x))\right\| \|W \\
&\leq 2\lambda \left\|(W - W')h + (U - U')x\right\|_\infty + \left\|W - W'\right\| \\
&\leq 2\lambda \left\|(W - W')\right\| \|h\| + 2\lambda \left\|U - U'\right\| \|x\| + \left\|W - W'\right\| \\
&\leq \frac{2\lambda B_U B_x + (1 - \lambda)}{(1 - \lambda)} \left\|W - W'\right\| + 2\lambda B_x \left\|U - U'\right\|.
\end{aligned}$$

Similar manipulations establish $\frac{\partial \phi_w(h,x)}{\partial w}$ is Lipschitz in h and w. □

Proof of Lemma 1. Note $|\tanh(x)|, |\sigma(x)| \leq 1$ for all x. Therefore, for any t, $\|h_t\|_\infty = \|o_t \circ \tanh(c_t)\|_\infty \leq 1$. Since $\sigma(x) < 1$ for $x < \infty$ and σ is monotonically increasing

$$\begin{aligned}\|f_t\|_\infty &\leq \sigma\left(\|W_f h_{t-1} + U_f x_t\|_\infty\right) \\ &\leq \sigma\left(\|W_f\|_\infty \|h_{t-1}\|_\infty + \|U_f\|_\infty \|x_t\|_\infty\right) \\ &\leq \sigma\left(B_W + B_u x\right) \\ &< 1.\end{aligned}$$

Using the trivial bound, $\|i_t\|_\infty \leq 1$ and $\|z_t\|_\infty \leq 1$, so

$$\|c_{t+1}\|_\infty = \|i_t \circ z_t + f_t \circ c_t\|_\infty \leq 1 + \|f_t\|_\infty \|c_t\|_\infty .$$

Unrolling this recursion, we obtain a geometric series

$$\|c_{t+1}\|_\infty \leq \sum_{i=0}^{t} \|f_t\|_\infty^i \leq \frac{1}{(1 - \|f\|_\infty)}.$$

□

Proof of Proposition 3. We show ϕ_{LSTM} is λ-contractive in the ℓ_∞-norm for some $\lambda < 1$. For $r \geq \log_{1/\lambda}(\sqrt{d})$, this in turn implies the iterated system ϕ_{LSTM}^t is contractive is the ℓ_2-norm.

Consider the pair of reachable hidden states $s = (c, h)$, $s' = (c', h')$. By Lemma (1), c, c' are bounded. Analogous to the recurrent network case above, since σ is (1/4)-Lipschitz and tanh is 1-Lipschitz,

$$\begin{aligned}\|i - i'\| &\leq \frac{1}{4} \|W_i\|_\infty \|h - h'\|_\infty \\ \|f - f'\| &\leq \frac{1}{4} \|W_f\|_\infty \|h - h'\|_\infty \\ \|o - o'\| &\leq \frac{1}{4} \|W_o\|_\infty \|h - h'\|_\infty \\ \|z - z'\| &\leq \|W_z\|_\infty \|h - h'\|_\infty .\end{aligned}$$

Both $\|z\|_\infty, \|i\|_\infty \leq 1$ since they're the output of a sigmoid. Letting c_+ and c'_+ denote the state on the next time step, applying the triangle inequality,

$$\begin{aligned}\|c_+ - c'_+\|_\infty &\leq \|i \circ z - i' \circ z'\|_\infty + \|f \circ c - f' \circ c'\|_\infty \\ &\leq \|(i - i') \circ z\|_\infty + \|i' \circ (z - z')\|_\infty + \|f \circ (c - c')\|_\infty + \|c \circ (f - f')\|_\infty \\ &\leq \|i - i'\|_\infty \|z\|_\infty + \|z - z'\|_\infty \|i'\|_\infty + \|c - c'\|_\infty \|f\|_\infty + \|f - f'\|_\infty \|c\|_\infty \\ &\leq \left(\frac{\|W_i\|_\infty + \|c\|_\infty \|W_f\|_\infty}{4} + \|W_z\|_\infty\right) \|h - h'\|_\infty + \|f\|_\infty \|c - c'\|_\infty .\end{aligned}$$

A similar argument shows

$$\|h_+ - h'_+\|_\infty \leq \|o - o'\|_\infty + \|c_+ - c'_+\|_\infty \leq \frac{\|W_o\|_\infty}{4} \|h - h'\|_\infty + \|c_+ - c'_+\|_\infty .$$

By assumption,

$$\left(\frac{\|W_i\|_\infty + \|c\|_\infty \|W_f\|_\infty + \|W_o\|_\infty}{4} + \|W_z\|_\infty\right) < 1 - \|f\|_\infty ,$$

and so

$$\left\|h_+ - h'_+\right\|_\infty < (1 - \|f\|_\infty) \left\|h - h'\right\|_\infty + \|f\|_\infty \left\|c - c'\right\|_\infty \leq \left\|s - s'\right\|_\infty ,$$

as well as

$$\left\|c_+ - c'_+\right\|_\infty < (1 - \|f\|_\infty) \left\|h - h'\right\|_\infty + \|f\|_\infty \left\|c - c'\right\|_\infty \leq \left\|s - s'\right\|_\infty ,$$

which together imply

$$\left\|s_+ - s'_+\right\|_\infty < \left\|s - s'\right\|_\infty ,$$

establishing ϕ_{LSTM} is contractive in the ℓ_∞ norm. □

A.3 Section (7)

Proof of Lemma 6. Let $h'_t = h_t(w')$. Expanding the gradients and using $\|h_t(w) - h_t(w')\| \leq \frac{L_w \|w - w'\|}{(1-\lambda)}$ from Lemma (7),

$$\begin{aligned}
\left\|\nabla_w p_T(w) - \nabla_w p_T(w')\right\| &\leq \sum_{t=1}^{T} \left\| \left(\frac{\partial h_T}{\partial h_t}\frac{\partial h_t}{\partial w}\right)^\top \nabla_{h_T} p_T - \left(\frac{\partial h'_T}{\partial h'_t}\frac{\partial h'_t}{\partial w}\right)^\top \nabla_{h'_T} p_T \right\| \\
&\leq \sum_{t=1}^{T} \left\|\nabla_{h_T} p_T - \nabla_{h'_T} p_T\right\| \left\|\frac{\partial h'_T}{\partial h'_t}\frac{\partial h'_t}{\partial w}\right\| + \|\nabla_{h_T} p_T\| \left\|\frac{\partial h_T}{\partial h_t}\frac{\partial h_t}{\partial w} - \frac{\partial h'_T}{\partial h'_t}\frac{\partial h'_t}{\partial w}\right\| \\
&\leq \sum_{t=1}^{T} \beta_p \left\|h_T - h'_T\right\| \lambda^{T-t} L_w + L_p \left\|\frac{\partial h_T}{\partial h_t}\frac{\partial h_t}{\partial w} - \frac{\partial h'_T}{\partial h'_t}\frac{\partial h'_t}{\partial w}\right\| \\
&\leq \frac{\beta_p L_w^2 \|w - w'\|}{(1-\lambda)^2} + \underbrace{L_p \sum_{t=1}^{T} \left\|\frac{\partial h_T}{\partial h_t}\frac{\partial h_t}{\partial w} - \frac{\partial h'_T}{\partial h'_t}\frac{\partial h'_t}{\partial w}\right\|}_{(a)}.
\end{aligned}$$

Focusing on term (a),

$$\begin{aligned}
L_p \sum_{t=1}^{T} \left\|\frac{\partial h_T}{\partial h_t}\frac{\partial h_t}{\partial w} - \frac{\partial h'_T}{\partial h'_t}\frac{\partial h'_t}{\partial w}\right\| &\leq L_p \sum_{t=1}^{T} \left\|\frac{\partial h_T}{\partial h_t} - \frac{\partial h'_T}{\partial h'_t}\right\| \left\|\frac{\partial h_t}{\partial w}\right\| + L_p \left\|\frac{\partial h'_T}{\partial h'_t}\right\| \left\|\frac{\partial h_t}{\partial w} - \frac{\partial h'_t}{\partial w}\right\| \\
&\leq L_p L_w \sum_{t=1}^{T} \left\|\frac{\partial h_T}{\partial h_t} - \frac{\partial h'_T}{\partial h'_t}\right\| + L_p \sum_{t=1}^{T} \lambda^{T-t} \left(\beta_{wh} \left\|h_t - h'_t\right\| + \beta_{ww} \left\|w - w'\right\|\right) \\
&\leq \underbrace{L_p L_w \sum_{t=1}^{T} \left\|\frac{\partial h_T}{\partial h_t} - \frac{\partial h'_T}{\partial h'_t}\right\|}_{(b)} + \frac{L_p \beta_{wh} L_w \|w - w'\|}{(1-\lambda)^2} + \frac{L_p \beta_{ww} \|w - w'\|}{(1-\lambda)},
\end{aligned}$$

where the penultimate line used,

$$\begin{aligned}\left\|\frac{\partial h_t}{\partial w}-\frac{\partial h'_t}{\partial w}\right\| &\leq \left\|\frac{\partial \phi_w(h_{t-1},x_t)}{\partial w}-\frac{\partial \phi_w(h'_{t-1},x_t)}{\partial w}\right\|+\left\|\frac{\partial \phi_w(h'_{t-1},x_t)}{\partial w}-\frac{\partial \phi_{w'}(h'_{t-1},x_t)}{\partial w}\right\| \\ &\leq \beta_{wh}\left\|h-h'\right\|+\beta_{ww}\left\|w-w'\right\|.\end{aligned}$$

To bound (b), we peel off terms one by one using the triangle inequality,

$$\begin{aligned}L_pL_w\sum_{t=1}^{T}\left\|\frac{\partial h_T}{\partial h_t}-\frac{\partial h'_T}{\partial h'_t}\right\| &\leq L_pL_w\sum_{t=1}^{T}\left\|\frac{\partial h_T}{\partial h_{T-1}}-\frac{\partial h'_T}{\partial h'_{T-1}}\right\|\left\|\frac{\partial h_{T-1}}{\partial h_t}\right\|+\left\|\frac{\partial h'_T}{\partial h'_{T-1}}\right\|\left\|\frac{\partial h_{T-1}}{\partial h_t}-\frac{\partial h'_{T-1}}{\partial h'_t}\right\| \\ &\leq L_pL_w\sum_{t=1}^{T}\left[\left(\beta_{hh}\left\|h_{T-1}-h'_{T-1}\right\|+\beta_{hw}\left\|w-w'\right\|\right)\lambda^{T-t-1}+\lambda\left\|\frac{\partial h_{T-1}}{\partial h_t}-\frac{\partial h'_{T-1}}{\partial h'_t}\right\|\right] \\ &\leq L_pL_w\sum_{t=1}^{T}\left[\beta_{hw}(T-t)\lambda^{T-t-1}\left\|w-w'\right\|+\beta_{hh}\sum_{i=1}^{T-t}\left\|h_{T-i}-h'_{T-i}\right\|\lambda^{T-t-1}\right] \\ &\leq L_pL_w\sum_{t=1}^{T}\left[\beta_{hw}(T-t)\lambda^{T-t-1}\left\|w-w'\right\|+\frac{\beta_{hh}L_w\left\|w-w'\right\|}{(1-\lambda)}(T-t)\lambda^{T-t-1}\right] \\ &\leq \frac{L_pL_w\beta_{hw}\left\|w-w'\right\|}{(1-\lambda)^2}+\frac{L_pL_w^2\beta_{hh}\left\|w-w'\right\|}{(1-\lambda)^3}.\end{aligned}$$

□