When Recurrent Models Don't Need To Be Recurrent

John Miller
University of California, Berkeley
miller_john@berkeley.edu

Moritz Hardt
University of California, Berkeley
hardt@berkeley.edu

May 30, 2018

Abstract

We prove stable recurrent neural networks are well approximated by feed-forward networks for the purpose of both inference and training by gradient descent. Our result applies to a broad range of non-linear recurrent neural networks under a natural stability condition, which we observe is also necessary. Complementing our theoretical findings, we verify the conclusions of our theory on both real and synthetic tasks. Furthermore, we demonstrate recurrent models satisfying the stability assumption of our theory can have excellent performance on real sequence learning tasks.

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 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 synthesis [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 feed-forward models 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 non-linear recurrent models that on some input sequences cannot be approximated by feed-forward models. But would such recurrent models be trainable by gradient descent?

Characterizing exactly which recurrent models are learnable by gradient descent is a delicate task beyond the reach of current theory. We will therefore instead work the fundamental control-theoretic notion of *stability*. This criterion roughly agrees with the requirement that the gradients of the training objective do not *explode* over time.

Loosely speaking, we prove stable recurrent models have good feed-forward approximations. Moreover, we prove that if gradient descent succeeds in training the recurrent model, it will also succeed in training the feed-forward model and vice-versa. This shows not only are the models equivalent for *inference*, they are also equivalent for *training* via gradient descent.

Of course, it is easy to violate the stability assumption; systems used in practice don't necessarily satisfy stability. However, we experimentally show stability can be enforced without loss of performance on a benchmark sequence task. In other cases not captured by our theory, we show competitive sequence models exhibit the same qualitative phenomena– for instance, limited sensitivity to inputs in the distant past– that allow us to approximate stable recurrent models by feed-forward models.

We prove stable recurrent models do not need to be recurrent, and we give experimental evidence suggesting this conclusion extends to a broader class of commonly used models. Taken together, our theory and experiments suggest that recurrent models are not an inevitable choice in sequence learning.

1.1 Contributions

In this work, we make the following contributions.

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

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

3. We empirically validate our results on synthetic data and show the same principles and phenomena underlying our theoretical analysis also appear in competitive models trained on a benchmark language modeling task.

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 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 RNN case, given $W \in \mathbf{R}^{n\times n}, U \in \mathbf{R}^{n\times d}$, and a pointwise non-linearity ρ, the system evolves according to

$$h_t = \rho\left(W h_{t-1} + U x_t\right).$$

We assume the state transition map is smooth in w and h, and the initial state $h_0 = 0$. Without loss of generality, we also assume $\phi_w(0,0) = 0$ for all w. Otherwise, we can reparameterize $\phi_w(h,x) \mapsto \phi_w(h,x) - \phi_w(0,0)$ without affecting expressivity of ϕ_w.

Throughout this paper, we focus on 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 $x \in \mathbf{R}^d$,

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

In the RNN case, stability corresponds to requiring $\|W\| < 1/L_\rho$, where L_ρ is the Lipschitz constant of ρ. More broadly, we assume there some compact convex domain $\Omega \subset \mathbf{R}^m$ so that the map ϕ_w is λ-contractive for all $w \in \Omega$.

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.

Let f denote a prediction function that maps a state h_t to outputs $f(h_t) = y_t$. Let y_t^k denote the predictions from the truncated model. 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 the sequel, we study $\left\|y_t - y_t^k\right\|$ both during and after training.

2.1 Our results

Our first result concerns inference in stable recurrent models. For fixed weights w, the predictions of truncated model well approximate the predictions of the full recurrent model at test-time.

Proposition (Informal version of Proposition 1)**.** *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, $\left\|y_t - y_t^k\right\| \leq \varepsilon$.*

Equipped with our approximation result, we turn towards optimization. We prove if it is possible to train a stable recurrent model via gradient descent to perform well on some task, then it is possible to get equally good performance by instead training an autoregressive feed-forward model.

Concretely, suppose both the full recurrent model and the truncated model are initialized at a common point w^0 and optimized to minimize some loss function on a common sequence of inputs. This results in a weight vector w_{recurr} for the full recurrent model and a 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)**.** *Assume the system ϕ is λ-contractive. Under additional smoothness and Lipschitz assumptions on the system ϕ, the prediction function f, and the loss p, we show if*

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

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

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 but can certainly make optimization harder by reducing the number of trainable model parameters.

2.2 Related work

In the *linear* dynamical system setting, [12] exploit the connection between stability and a truncated system approximation 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, and we extend this result to the non-linear setting. We also analyze the impact of truncation on training with gradient descent. Results of this 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] show gradient descent can efficiently learn stable, 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 principled explanation for this phenomenon.

Our training time analysis builds on the stability analysis of gradient descent in [7], but interestingly uses it for an entirely different purpose.

3 Approximation during inference

Suppose we train a full recurrent model ϕ_w and obtain a prediction y_t. For an appropriate choice of context k, the truncated model makes essentially the same prediction y_t^k as the full recurrent model. To show this result, we first control the difference between the hidden states of both models.

Lemma 1. *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 $k \geq O\left(\log\left(\frac{L_x B_x}{(1-\lambda)\varepsilon}\right)\right)$, then the difference in hidden states $\left\|h_t - h_t^k\right\| \leq \varepsilon$.*

Lemma 1 effectively says stable models do not have long-term memory– distant inputs do not change the states of the system. For this reason, it is a key building block for both our inference and our subsequent training-time analysis. A full proof is given in the appendix. If the prediction function is Lipschitz, Lemma 1 immediately implies that the predictions between the recurrent and truncated model are nearly identical. This leads us to the following proposition.

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

4 Approximation during training

In this section, we show gradient descent for stable recurrent models finds essentially the same solutions as gradient descent for truncated models. Consequently, both the recurrent and truncated models found by gradient descent make essentially the same predictions.

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 is 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.

The gradient descent result requires two essential lemmas. The first bounds the difference in gradient between the full and the truncated model. The second establishes the gradient map of both the full and truncated models is Lipschitz. We defer proofs of both lemmas to the appendix.

Let p_T denote the loss function evaluated on recurrent model after T time steps, and define p_T^k similarly for the truncated model.

Lemma 2. *Assume p (and therefore p^k) is Lipschitz and smooth. Assume ϕ_w is smooth, λ-contractive, and Lipschitz in x and w. Assume the inputs satisfy $\|x_t\| \leq B_x$, then*

$$\left\|\nabla_w p_T - \nabla_w p_T^k\right\| = \gamma k \lambda^k,$$

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

Lemma 3. *For any $w, w' \in \Omega$, suppose ϕ_w is smooth, λ-contractive, and Lipschitz in w. If p is Lipschitz and smooth, then*

$$\left\|\nabla_w p_T(w) - \nabla_w p_T(w')\right\| \leq \beta \left\|w - w'\right\|,$$

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

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, $w_{\text{recurr}}^0 = w_{\text{trunc}}^0$. For k sufficiently large, Lemma 2 guarantees the difference between the gradient of the recurrent and truncated models is negligible. Therefore, after a gradient update, $\left\|w^1_{\text{recurr}} - w^1_{\text{trunc}}\right\|$ is small. Lemma 3 then guarantees that this small difference in weights does not lead to large differences in the gradient on the subsequent time step. For an appropriate choice of learning rate, formalizing this argument leads to the following proposition.

Proposition 2. *Under the assumptions of Lemmas 2 and 3, for compact, convex Ω, after N steps of projected gradient descent with step size $\alpha_t = \alpha/t$, $\left\|w^N_{\text{recurr}} - w^N_{\text{trunc}}\right\| \leq \alpha\gamma k \lambda^k N^{\alpha\beta+1}$.*

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. In section 7, we give empirical evidence the $O(1/t)$ rate is necessary for our theorem and show examples of stable systems trained with constant or $O(1/\sqrt{t})$ rates that do not satisfy our bound.

Critically, the bound in Proposition 2 goes to 0 as $k \to \infty$. In particular, if we take $\alpha = 1$ and $k = O\left(\log(\gamma N^\beta/\varepsilon)\right)$, then after N steps of projected gradient descent, $\left\|w^N_{\text{recurr}} - w^N_{\text{trunc}}\right\| \leq \varepsilon$. For this choice of k, we obtain the main theorem. The proof is left to the appendix.

Theorem 1. *Let p be Lipschitz and smooth. Assume ϕ_w is smooth, λ-contractive, Lipschitz in x and w. Assume the inputs are bounded, and the prediction function f is L_f-Lipschitz. If $k = O(\log(\gamma N^\beta/\varepsilon))$, then after N steps of projected gradient descent with step size $\alpha_t = 1/t$, $\left\|y_T - y^k_T\right\| \leq \varepsilon$.*

5 Counterexamples without stability

While the stability assumption on the state-transition map ϕ_w might seem limiting, it is in fact necessary on two counts. First, without stability, there are trivial 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, without stability, it is difficult to show gradient descent converges to a stationary point, even in the linear dynamical system case. Indeed, there exist simple 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 3. *There exists an unstable system ϕ such that, for any finite truncation length k, $\left\|y_t - y^k_t\right\| \to \infty$ as $t \to \infty$.*

Proposition 4. *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 $N \to \infty$.*

6 Examples of stable models

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. 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.

6.1 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

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

The model is stable provided $\|W\| \leq \lambda < 1$, and Lipschitz in x provided $\|U\|$ is bounded. For a stable linear dynamical system, it is easy show $\max_t \|h_t\| < O((1-\lambda)^{-1})$, and consequently the model is $O((1-\lambda)^{-1})$ Lipschitz in W. It's a simple exercise to check that such a system satisfies the remaining Lipschitz and smoothness assumptions.

6.2 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

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

If ρ is L_ρ-Lipschitz, the model is stable provided $\|W\| < \frac{1}{L_\rho}$. Indeed, for any states h, h', and any x,

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

Our remaining Lipschitz and smoothness assumptions are satisfied if ρ is smooth and $\|U\| \leq B_U$. For concreteness, in the appendix, we show each of the assumptions holds when ρ is tanh, which is 1-smooth and 1-Lipschitz. On the other hand, our results do not apply for the non-smooth ReLu.

6.3 Long short-term memory networks

Long Short-Term Memory (LSTM) networks are another commonly used class of sequence models [8]. The state is a pair of vectors $s = (c, h) \in \mathbf{R}^{2d}$, and the model is parameterized by eight matrices, $W_\square \in \mathbf{R}^{d \times d}$ and $U_\square \in \mathbf{R}^{d \times n}$, for $\square \in \{i, f, o, z\}$. The state-transition map ϕ_{LSTM} is given by

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

where $\circ$ denotes elementwise multiplication, and σ is the logistic function.

We provide conditions under which the iterated system $\phi_{\text{LSTM}}^r = \phi_{\text{LSTM}} \circ \cdots \circ \phi_{\text{LSTM}}$ is stable. Let $\|f\|_\infty = \sup_t \|f_t\|_\infty$. If the weights W_f, U_f and inputs x_t are bounded, then $\|f\|_\infty < 1$ since $|\sigma| < 1$ for any finite input. This means the next state c_t must "forget" a non-trivial portion of c_{t-1}. We leverage this phenomenon to give sufficient conditions for ϕ_{LSTM} to be contractive in the ℓ_∞ norm, which in turn implies the iterated system ϕ_{LSTM}^r is contractive in the ℓ_2 norm for $r = O(\log(d))$. Let $\|W\|_\infty$ denote the induced ℓ_∞ matrix norm, which corresponds to the maximum absolute row sum $\max_i \sum_j |W_{ij}|$.

Proposition 5. *If* $\|W_i\|_\infty, \|W_o\|_\infty < (1 - \|f\|_\infty)$*,* $\|W_z\|_\infty \leq (1/4)(1 - \|f\|_\infty)$*,* $\|W_f\|_\infty < (1 - \|f\|_\infty)^2$*, and* $r = O(\log(d))$*, then the iterated system* ϕ^r_{LSTM} *is stable on the set of reachable states.*

The proof is given in the appendix. As a consequence of Proposition 5, both Proposition 1 and Theorem 1 apply to ϕ_{LSTM} at the cost of an additional $\log(d)$ factor in the choice of truncation length k. 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.

7 Experiments

In the experiments, we verify the conclusions of our theoretical investigations using synthetic data and present evidence our results hold beyond settings captured by our theory using a benchmark language modeling task, WikiText-2 [9]. All of our language modeling experiments use publicly available code,[1] and details of the hyperparameters for all experiments are given in the appendix.

7.1 Understanding the main gradient descent bound

The key result underlying Theorem 1 is the bound on the parameter difference $\|w_{\text{trunc}} - w_{\text{recurr}}\|$ while running gradient descent obtained in Proposition 2. We show this bound has the correct qualitative scaling using random instances.

Concretely, we sample random Gaussian input sequences $x_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 4I_{32})$ and $y_T \sim \text{Unif}[-2, 2]$ for $T = 200$. We fix $\lambda = 0.75$ and randomly initialize a stable linear dynamical system or tanh-RNN (details in appendix). 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 $\|w_{\text{trunc}} - w_{\text{recurr}}\| \leq 2\lambda$.

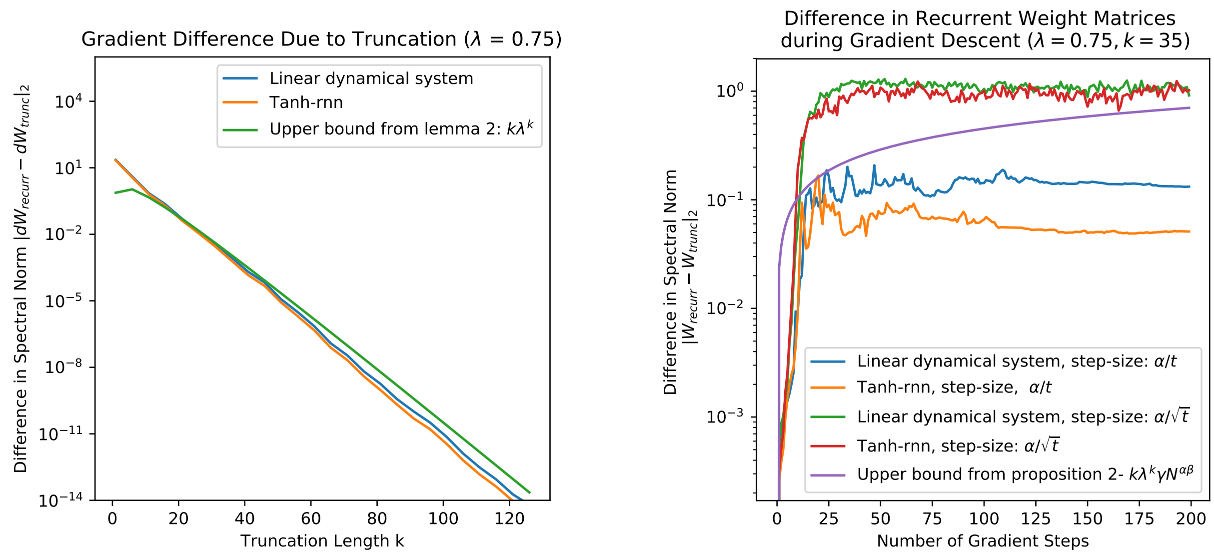
In Figure 1(a), we plot $\left\|\nabla_w p_T - \nabla_w p_T^k\right\|$ as k varies (averaged over 10 runs) for both a linear dynamical system and a recurrent neural network. The error closely matches the $k\lambda^k$ scaling predicted by Lemma 2.

In Figure 1(b), we plot the parameter error $\left\|w^t_{\text{trunc}} - w^t_{\text{recurr}}\right\|$ as training progresses for both a linear dynamical system and a recurrent neural network with tanh non-linearities (averaged over 10 runs). The error scales comparably with the bound given in Proposition 2. We also find for larger step-sizes like $\alpha/\sqrt{t}$ or constant α (omitted from the plot to reduce clutter), the bound fails to hold, suggesting the $O(1/t)$ condition is necessary.

7.2 Moving beyond stability

Our theoretical results require stability, and Section 5 shows stability is necessary without further assumptions. On the other hand, recurrent models trained in practice are not a-priori stable. We first show imposing the stability constraint need not significantly change the performance of benchmark models. We then provide evidence phenomena like vanishing gradients and truncated system approximation hold outside of settings captured by our theory, posing challenges for future work.

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



(a) Truncation Bounds

(b) Gradient Descent Bounds

Figure 1: Empirical validation of Lemma 2 and Proposition 2 on random Gaussian instances. Without the $1/t$ rate, the gradient descent bound no longer appears qualitatively correct, suggesting the $O(1/t)$ rate is necessary.

At least in one very sp

Enforcing stability doesn't hurt performance. Stable models can achieve good performance on Wikitext-2. To demonstrate this, we trained two single layer tanh recurrent neural networks. The first model is unconstrained, and the second model is constrained to $\|W\| \leq 1$, which ensures stability by Section 6. 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 $[0, 1)$. All of the hyperparameters were chosen via grid-search to maximize the performance of the unconstrained model. At convergence, there is little difference between the two models. The unconstrained model achieves a final test perplexity of **146.7**, whereas the stable, constrained model achieves a final test perplexity of **143.5**.

The importance of this result is two-fold. First, our theory applies to models that achieve reasonable performance on a benchmark task. Second, it suggests other conditions on the data distribution or weight matrices combine so models trained in practice are effectively stable and thus permit approximation by feed-forward networks.

Vanishing gradients. The central phenomena that makes feed-forward approximation during training possible is vanishing gradients. Indeed, vanishing gradients are a key ingredient in our proof of Lemma 2. LSTMs and recurrent neural networks trained in practice exhibit vanishing gradients and limited sensitivity to past inputs beyond what's guaranteed by our theory. In Figure (2), we train an LSTM and RNN on Wikitext-2 and plot $\|\nabla_{x_t} p_{t+i}\|$ for $i = 1, \ldots, 50$ for t ranging over the validation set at the end of each epoch. We do not enforce the spectral norm constraint in either case. 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.

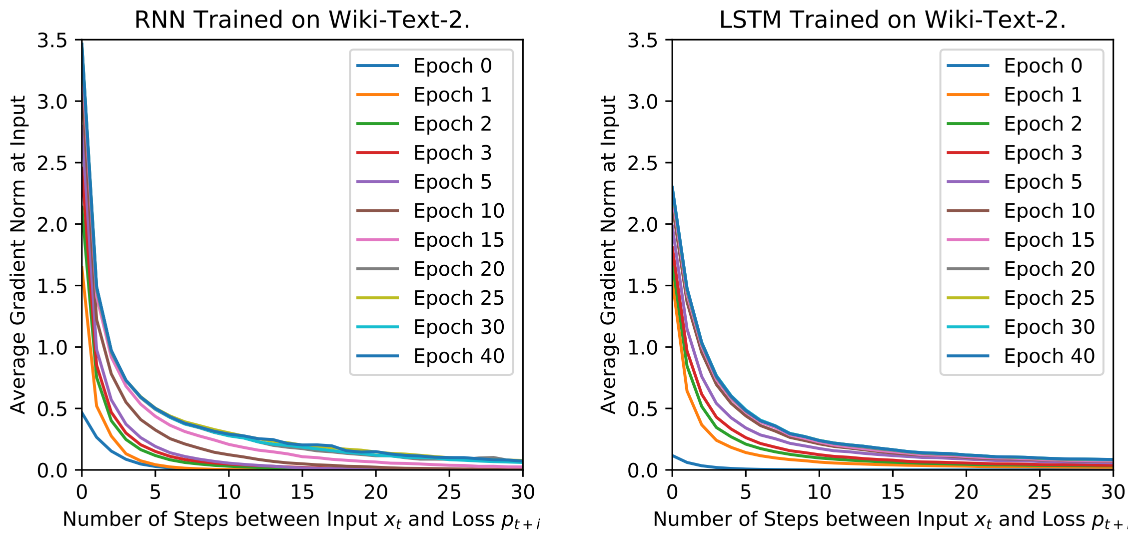
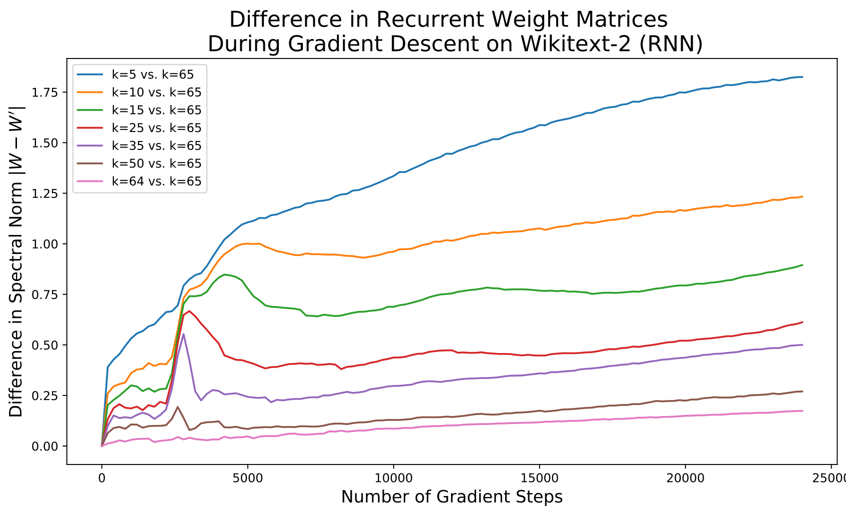


Figure 2: Norm of the gradient with respect to inputs, $\|\nabla_{x_t} p_{t+i}\|$, as the distance between the input and the loss grows, averaged over the entire held-out set. The gradient vanishes for moderate values of i in both cases. The RNN has test perplexity 146.7 and the LSTM has test perplexity of 92.3.

Approximation during training. In settings not captured by our existing results, a phenomenon similar to Proposition 2 holds empirically. In particular, $\left\|w^t_{\text{trunc}} - w^t_{\text{recurr}}\right\|$ grows slowly with t, and the rate of this growth decreases as the value of the truncation parameter k increases. As a representative example, we trained truncated RNN and LSTM models on Wikitext-2 for various values of k. Training the full recurrent model is impractical, and hence we assume $k = 65$ well captures the full-recurrent model. All of the models are initialized at the same point, and we track the distance between the hidden-to-hidden matrices W as training progresses.

In Figure (3), we plot $\|W_k - W_{65}\|$ for $k \in \{5, 10, 15, 25, 35, 50, 64\}$ as training proceeds. In the RNN case, W_k denotes the recurrent matrix trained with truncation length k, and in the LSTM case, W_k denotes the concatenation of $[W_i; W_f; W_o; W_z]$ trained with truncation length k.

After an initial rapid increase in distance, $\|W_k - W_{65}\|$ grows slowly, similar to the results obtained in the random Gaussian case and in Proposition 2. Moreover, as suggested by our theory, there is a diminishing return to choosing larger values of the truncation parameter k in terms of the accuracy of the approximation.



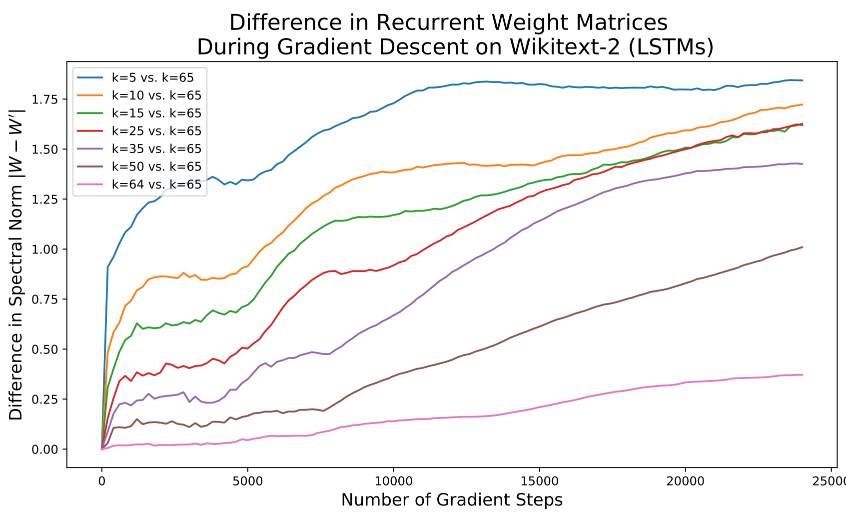


Figure 3: Distance in spectral norm between the hidden-to-hidden matrices of truncated RNNs and LSTMs trained on Wikitext-2 with different values of the truncation parameter k during gradient descent. All models were initialized at the same point, and share all other hyperparameters and random seeds. At convergence, $\left\|W_{65}^{\mathrm{RNN}}\right\| = 2.95$ and $\left\|W_{65}^{\mathrm{LSTM}}\right\| = 5.4$.

References

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

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

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

[4] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *International Conference on Machine Learning*, pages 933–941, 2017.

[5] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *International Conference on Machine Learning*, pages 1243–1252, 2017.

[6] Moritz Hardt, Tengyu Ma, and Benjamin Recht. Gradient descent learns linear dynamical systems. *arXiv preprint arXiv:1609.05191*, 2016.

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

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

[9] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

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

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

[12] Stephen Tu, Ross Boczar, Andrew Packard, and Benjamin Recht. Non-asymptotic analysis of robust control from coarse-grained identification. *arXiv preprint arXiv:1707.04791*, 2017.

[13] Aaron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010, 2017.

A Deferred proofs

A.1 Proofs from section 3

Proof of Lemma 1. For any $t \geq 1$, by triangle inequality,

$$\|h_t\| = \|\phi_w(h_{t-1}, x_t) - \phi_w(0,0)\| \leq \|\phi_w(h_{t-1}, x_t) - \phi_w(0, x_t)\| + \|\phi_w(0, x_t) - \phi_w(0,0)\| .$$

Applying the stability and Lipschitz assumptions and then summing a geometric series,

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

Now, consider the difference between hidden states at time step t. Unrolling the iterates k steps and then using the previous display 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. □

A.2 Proofs from section 4

Before proceeding, we introduce notation for our smoothness assumption. We assume the map ϕ_w satisfies four smoothness conditions: for any reachable states h, h', and any weights $w, w' \in \Omega$, there are some scalars $\beta_{ww}, \beta_{wh}, \beta_{hw}, \beta_{hh}$ such that

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. $\left\|\frac{\partial \phi_w(h,x)}{\partial h} - \frac{\partial \phi_w(h',x)}{\partial h}\right\| \leq \beta_{hh} \|h - h'\|$.

A.2.1 Gradient difference due to truncation is negligible

In the section, we argue the difference in gradient with respect to the weights between the recurrent and truncated models is $O(k\lambda^k)$. For sufficiently large k (independent of the sequence length), the impact of truncation is therefore negligible. The proof leverages the "vanishing-gradient" phenomenon– the long-term components of the gradient of the full recurrent model quickly vanish. The remaining challenge is to show the short-term components of the gradient are similar for the full and recurrent models.

Proof of Lemma 2. The Jacobian of the loss with respect to the weights is

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

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. Expanding the expression for the gradient, we wish to bound

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

The first term consists of the "long-term components" of the gradient for the recurrent model. The second term is the difference in the "short-term components" of the gradients between the recurrent and truncated models. We bound each of these terms separately.

For the first term, by the Lipschitz assumptions, $\|\nabla_{h_T} p_T\| \le L_p$ and $\|\nabla_w h_t\| \le L_w$. Since ϕ_w is λ-contractive, so $\left\|\frac{\partial h_t}{\partial h_{t-1}}\right\| \le \lambda$. Using submultiplicavity of the spectral norm,

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

Focusing on the second term, by triangle inequality and smoothness,

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

Using Lemma 1 to upper bound (a),

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

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 $O\left(\frac{k\lambda^k}{(1-\lambda)^2}\right)$. □

A.2.2 Stable recurrent models are smooth

In this section, we prove that the gradient map $\nabla_w p_T$ is Lipschitz. First, we show on the forward pass, the difference between hidden states $h_t(w)$ and $h_t'(w')$ obtained by running the model with weights w and w', respectively, is bounded in terms of $\|w - w'\|$. Using smoothness of ϕ, the difference in gradients can be written in terms of $\|h_t(w) - h_t'(w')\|$, which in turn can be bounded in terms of $\|w - w'\|$. We repeatedly leverage this fact to conclude the total difference in gradients must be similarly bounded.

We first show small differences in weights don't significantly change the trajectory of the recurrent model.

Lemma 4. *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*

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

Proof. By triangle inequality, followed by the Lipschitz and contractivity assumptions,

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

Iterating this argument and then using $h_0(w) = h_0(w')$, we obtain a geometric series in λ.

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

□

The proof of Lemma 3 is similar in structure to Lemma 2, and follows from repeatedly using smoothness of ϕ and Lemma 4.

Proof of Lemma 3. 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 4.

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

Focusing on term (a),

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

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,

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

Supressing Lipschitz and smoothness constants, we've shown the entire sum is $O(1/(1-\lambda)^3)$, as required. $\square$

A.2.3 Gradient descent analysis

Equipped with the smoothness and truncation lemmas (Lemmas 2 and 3), we turn towards proving the main gradient descent result.

Proof of Proposition 2. Let Π_Ω denote the Euclidean projection onto Ω, and let $\delta_i = \left\| w^i_{\text{recurr}} - w^i_{\text{trunc}} \right\|$. Initially $\delta_0 = 0$, and on step $i+1$, we have the following recurrence relation for δ_{i+1},

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

the penultimate line applied lemmas 2 and 3, and the last line used $1 + x \leq e^x$ for all x. Unwinding the recurrence relation at step N,

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

Bounding the inner summation via an integral, $\sum_{j=i+1}^{N} \frac{1}{j} \leq \log(N/i)$ and simplifying the resulting expression,

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

□

A.2.4 Proof of theorem 1

Proof of Theorem 1. Using f is L_f-Lipschitz and the triangle inequality,

$$
\begin{aligned}
\left\| y_T - y^k_T \right\| &\leq L_f \left\| h_T(w^N_{\text{recurr}}) - h^k_T(w^N_{\text{trunc}}) \right\| \\
&\leq L_f \left\| h_T(w^N_{\text{recurr}}) - h_T(w^N_{\text{trunc}}) \right\| + L_f \left\| h_T(w^N_{\text{trunc}}) - h^k_T(w^N_{\text{trunc}}) \right\|.
\end{aligned}
$$

By Lemma 4, the first term is bounded by $\frac{L_w \left\| w^N_{\text{recurr}} - w^N_{\text{trunc}} \right\|}{(1-\lambda)}$, and by Lemma 1, the second term is bounded by $\lambda^k \frac{L_x B_x}{(1-\lambda)}$. Using Proposition 2, after N steps of gradient descent, we have

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

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

A.3 Proofs from section 6

A.3.1 Recurrent neural networks

Details of Section 6.2. 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',

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

Using Lemma 1 with $k = 0$, $\|h_t\| \leq \frac{B_U B_x}{(1-\lambda)}$ 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,

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

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

A.3.2 LSTMs

Similar to the previous sections, we assume $s_0 = 0$.

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 this is always the case.

Lemma 5. *Let $\|f\|_\infty = \sup_t \|f_t\|_\infty$. If $\|W_f\|_\infty < \infty$, $\|U_f\|_\infty < \infty$, and $\|x_t\| \le B_x$, then $\|f\|_\infty < 1$ and $\|c_t\|_\infty \le \frac{1}{(1-\|f\|_\infty)}$ for all t.*

Proof of Lemma 5. Note $|\tanh(x)|, |\sigma(x)| \le 1$ for all x. Therefore, for any t, $\|h_t\|_\infty = \|o_t \circ \tanh(c_t)\|_\infty \le 1$. Since $\sigma(x) < 1$ for $x < \infty$ and σ is monotonically increasing

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

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

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

Unrolling this recursion, we obtain a geometric series

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

□

Proof of Proposition 5. We show ϕ_{LSTM} is λ-contractive in the ℓ_∞-norm for some $\lambda < 1$. For $r \ge \log_{1/\lambda}(\sqrt{d})$, this in turn implies the iterated system ϕ_{LSTM}^r is contractive is the ℓ_2-norm.

Consider the pair of reachable hidden states $s = (c, h)$, $s' = (c', h')$. By Lemma 5, c, c' are bounded. Analogous to the recurrent network case above, since σ is (1/4)-Lipschitz and tanh is

1-Lipschitz,

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

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

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

A similar argument shows

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

By assumption,

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

and so

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

as well as

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

which together imply

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

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

A.4 Proofs from section 5

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

$$h_t = a h_{t-1} + b x_t \tag{4}$$

$$\hat{y}_t = h_t, \tag{5}$$

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 3. 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 4. Suppose $(x_t, y_t) = (1, 1)$ for $t = 1, \ldots, L$. Then the desired system (4) 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

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

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

$$\begin{aligned}
\frac{d}{da} \ell(y_T, \hat{y}_T) = \delta_T \frac{d}{da} &= \delta_T \sum_{t=0}^{T-1} a^{T-1-t} h_t \\
&= \delta_T \sum_{t=0}^{T-1} a^{T-1-t} \left(\frac{1 - a^t}{1 - a} \right) \\
&= \delta_T \left[\frac{1}{(1-a)} \sum_{t=0}^{T-1} a^t - \frac{T a^{T-1}}{(1-a)} \right] \\
&= \delta_T \left[\frac{(1 - a^T)}{(1-a)^2} - \frac{T a^{T-1}}{(1-a)} \right].
\end{aligned}$$

Plugging in $y_t = 1$, this becomes

$$\frac{d}{da} \ell(y_T, \hat{y}_T) = \left(\frac{(1 - a^T)}{(1-a)} - 1 \right) \left[\frac{(1 - a^T)}{(1-a)^2} - \frac{T a^{T-1}}{(1-a)} \right]. \tag{6}$$

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

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

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 (6), it is clear that such a^i are not stationary points. □

A.5 Experimental details

Our language modeling experiments use code from `https://github.com/pytorch/examples/tree/master/word_language_model`. Unless otherwise noted, the models and hyperparameters are fixed for all experiments.

Synthetic data. We generate random problem instance by fixing a sequence length $T = 200$, sampling input data $x_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 4 \cdot I_{32})$, and sampling $y_T \sim \text{Unif}[-2, 2]$. Next, we set $\lambda = 0.75$ and randomly initialize a stable linear dynamical system or RNN with tanh non-linearity by sampling $U_{ij}, W_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 0.5)$ and thresholding the singular values of W so $\|W\| \leq \lambda$. We use the squared loss and prediction function $f(h_t, x_t) = Ch_t + Dx_t$, where $C, D \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_{32})$. 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.

Stable models. Both networks consists of a single hidden layer with the tanh-nonlinearity. The embedding dimension is 1024, the hidden state dimension is 256, and dropout is set to 0.25. The initial learning rate is 2, and the model is trained for 40 epochs using a batch size of 20 and a sequence length of 35.

Vanishing gradients and feed-forward approximations The RNN consists of a single hidden layer with the tanh-nonlinearity. The embedding and hidden state dimensions are both 1500, and dropout is set to 0.65. The initial learning rate is 5. The LSTM model consists of a single layer and uses embedding and hidden state dimensions of 1500. Both models are trained for 40 epochs using a batch size of 20 and a sequence length of 65. All other hyperparameters are set to the default for both models.