On Generalization Bounds of a Family of Recurrent Neural Networks

Minshuo Chen[*] **Xingguo Li**[†] **Tuo Zhao**[*]
[*] Georgia Tech [†] Princeton University

Abstract

Recurrent Neural Networks (RNNs) have been widely applied to sequential data analysis. Due to their complicated modeling structures, however, the theory behind is still largely missing. To connect theory and practice, we study the generalization properties of vanilla RNNs as well as their variants, including Minimal Gated Unit (MGU), Long Short Term Memory (LSTM), and Convolutional (Conv) RNNs. Specifically, our theory is established under the PAC-Learning framework. The generalization bound is presented in terms of the spectral norms of the weight matrices and the total number of parameters. We also establish refined generalization bounds with additional norm assumptions, and draw a comparison among these bounds. We remark: (1) Our generalization bound for vanilla RNNs is significantly tighter than the best of existing results; (2) We are not aware of any other generalization bounds for MGU and LSTM RNNs in the exiting literature; (3) We demonstrate the advantages of these variants in generalization.

1 Introduction

Recurrent Neural Networks (RNNs) have successfully revolutionized sequential data analysis, and been widely applied to many real world problems, such as natural language processing (Cho et al., 2014; Bahdanau et al., 2014; Sutskever et al., 2014), speech recognition (Graves et al., 2006; Mikolov et al., 2010; Graves, 2012; Graves et al., 2013), computer vision (Gregor et al., 2015; Xu et al., 2015; Donahue et al., 2015; Karpathy and Fei-Fei, 2015), healthcare (Lipton et al., 2015; Choi et al., 2016a,b), and robot control (Lee and Teng, 2000; Yoo et al., 2006). Quite a few of these applications can be approached easily in our daily life, such as Google Translate and Apple Siri.

The sequential modeling nature of RNNs is significantly different from feedforward neural networks, though they both have neurons as the basic components. RNNs exploit the internal state (also known as hidden unit) to process the sequence of inputs, which naturally captures the dependence of the sequence. Besides the vanilla version, RNNs have many other variants. A large class of variants incorporate the so-called "gated" units to trim RNNs for different tasks. Typical examples include Long Short-Term Memory (LSTM, Hochreiter and Schmidhuber (1997)), Gated Recurrent Unit (GRU, Jozefowicz et al. (2015)) and Minimal Gated Unit (MGU, Zhou et al. (2016)).

The success of RNNs owes not only to their special network structures and the ability to fit data, but also to their good generalization property: They provide accurate predictions on unseen data. For example, Graves et al. (2013) report that after training with merely 462 speech samples, deep LSTM RNNs achieve a test set error of 17.7% on TIMIT phoneme recognition benchmark, which is the best recorded score. Despite of the popularity of RNNs in applications, their theory is less studied than other feedforward neural networks (Haussler, 1992; Bartlett et al., 2017; Neyshabur et al., 2017; Golowich et al., 2017; Li et al., 2018). There are still several long lasting fundamental questions regarding the approximation, trainability, and generalization of RNNs.

In this paper, we propose to understand the generalization ability of RNNs and their variants. We aim to answer two questions from a theoretical perspective:

Q.1) Do RNNs suffer from significant curse of dimensionality?

Q.2) What are the advantages of MGU and LSTM over vanilla RNNs?

The investigation of generalization properties of RNNs has a long history. Many early works are based on over-simplified assumptions. For example, Dasgupta and Sontag (1996) and Koiran (1998) adopt a VC-

Proceedings of the 23$^{\text{rd}}$International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108. Copyright 2020 by the author(s).

dimension argument to show complexity bounds of RNNs that are polynomial in the size of the network. They, however, either consider linear RNNs for binary classification tasks, or assume RNNs take the first coordinate of their hidden states as outputs. More recently, Bartlett et al. (2017) propose a new technique for developing generalization bounds for feedforward neural networks based on empirical Rademacher complexity under the PAC-Learning framework. Neyshabur et al. (2017) further adapt the technique to establish their generalization bound using the PAC-Bayes approach. The follow-up work Zhang et al. (2018) use the PAC-Bayes approach to establish a generalization bound for vanilla RNNs.

We exploit the compositional nature of RNNs, and decouples the spectral norms of weight matrices and the number of weight parameters. This makes our analysis conceptually much simpler (e.g. avoid layer wise analysis), and also yields better generalization bound than Zhang et al. (2018).

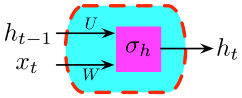

Figure 1: A building block of vanilla RNNs.

Consider vanilla RNNs, we observe m sequences of data points $(x_{i,t}, z_{i,t})_{t=1}^T$, where $x_{i,t} \in \mathbb{R}^{d_x}$ and the response $z_{i,t} \in \mathcal{Z}$ for all $t = 1, ..., T$ and $i = 1, ..., m$. Each sequence is drawn independently from some underlying distribution over $\mathbb{R}^{d_x \times T} \times \mathcal{Z}$. Extensions to dependent sequences are discussed in Section 7, however, note that data points $(x_{i,t}, z_{i,t})$ can be dependent within a sequence, i.e., for a fixed $i \in \{1, \ldots, m\}$. The vanilla RNNs compute $h_{i,t}$ and $y_{i,t}$ iteratively as follows,

$$h_{i,t} = \sigma_h\left(Uh_{i,t-1} + Wx_{i,t}\right), \quad \text{and} \quad y_{i,t} = \sigma_y\left(Vh_{i,t}\right),$$

where σ_y and σ_h are activation operators, $h_{i,t} \in \mathbb{R}^{d_h}$ is the hidden state with $h_{i,0} = 0$, and $U \in \mathbb{R}^{d_h \times d_h}$, $V \in \mathbb{R}^{d_y \times d_h}$, and $W \in \mathbb{R}^{d_h \times d_x}$ are weight matrices. The activation operators σ_h and σ_y are entrywise, i.e., $\sigma_h([v_1, \ldots, v_d]^\top) = [\sigma_h(v_1), \ldots, \sigma_h(v_d)]^\top$, and Lipschitz with parameters ρ_h and ρ_y respectively. We assume $\sigma_h(\cdot) = \tanh(\cdot)$, $\sigma_y(0) = 0$, and $\rho_y = 1$. Extensions to general activations are given in Section 2.

Our Contribution. To establish the generalization bound, we need to define the "model complexity" of vanilla RNNs. In this paper, we adopt the empirical Rademacher complexity (ERC, see more details in Section 2), which has been widely used in the existing literature on PAC-Learning. For many nonparametric function classes, we often need complicated argument to upper bound their ERC. Our analysis, however, shows that we can upper bound the ERC of vanilla RNNs in a very simple manner by exploiting their Lipschitz continuity with respect to the model parameters, since they are essentially in parametric forms. More specifically, denote $\mathcal{F}_t = \{f_t : \{x_1, ..., x_t\} \mapsto y_t\}$ as the class of mappings from the first t inputs to the t-th output computed by vanilla RNNs. For a matrix A, $\|A\|_2$ denotes the spectral norm, and for a vector v, $\|v\|_2$ denotes the Euclidean norm. Define $\frac{a^t-1}{a-1} = t$ for $a = 1$. Then, informally speaking, the "model complexity" of vanilla RNNs satisfies

$$\mathsf{Complexity} = O\Bigg(d\|V\|_2 \min\Big\{\sqrt{d}, \|W\|_2 \frac{\|U\|_2^t - 1}{\|U\|_2 - 1}\Big\} \times \sqrt{\log\left(t\sqrt{d}\frac{\|U\|_2^t - 1}{\|U\|_2 - 1}\right)}\Bigg),$$

where $d = \sqrt{d_x d_h + d_h^2 + d_h d_y}$.

We then consider a new testing sequence $(x_t, z_t)_{t=1}^T$. The response sequence is computed by $\widetilde{z}_t = \phi(y_t)$, for all $t = 1, \ldots, T$, where ϕ is a function mapping the output of vanilla RNNs to the response of interest. In practice, the function ϕ varies across different data analysis tasks. For example, in sequence to sequence classification, we take $\phi(y_t) = \text{argmax}_j [y_t]_j$; in regression, we take $\phi(y_t) = y_t$; in density estimation, we can take $\phi(y_t) = \text{softmax}(y_t)$.

We further define a risk function that can unify different data analysis tasks. Specifically, let $\mathcal{L}(\mathcal{A}(y, z))$ be a loss function, where $\mathcal{A}(y, z)$ is a function taking the output y_t and the observed response z_t as inputs, and $\mathcal{L}$ is chosen according to different tasks. Then we define the population risk for the t-th output as $\mathcal{R}(f_t) = \mathbb{E}[\mathcal{L}(\mathcal{A}(y_t, z_t))]$. Its empirical counterpart is similarly defined as $\widehat{\mathcal{R}}(f_t) = \frac{1}{m}\sum_{i=1}^m \mathcal{L}(\mathcal{A}(y_{i,t}, z_{i,t}))$. Training RNNs is essentially minimizing the empirical risk $\widehat{\mathcal{R}}(f_t)$. Many applications can be formulated into this framework. For example, in classification, we take $\mathcal{A} = -\mathcal{M}$ as the functional margin operator and $\mathcal{L} = \mathcal{L}_\gamma$ as the ramp loss with γ being the margin value (see detailed definitions in Section 2); in regression, we take $\mathcal{A}(y_t, z_t) = y_t - z_t$ and $\mathcal{L}$ as the ℓ_p loss for $p \in \mathbb{Z}_+$. We then give the generalization bound in the following statement.

Theorem 1 (informal)**.** Assume the input data space is bounded, i.e., $\|x\|_2 \leq 1$ and $z \in \mathcal{Z}$ bounded. Suppose the mapping $\mathcal{A}(y, z)$ is Lipschitz in y, and the loss function $\mathcal{L}$ satisfies $|\mathcal{L}(\mathcal{A}(y, z))| \leq B$ and is L-Lipschitz for any y computed by RNNs and $z \in \mathcal{Z}$. Given a collection of samples $S = \left\{(x_{i,t}, z_{i,t})_{t=1}^T, i = 1, ..., m\right\}$ and a new testing sequence $(x_t, z_t)_{t=1}^T$, with probability at least $1 - \delta$ over S, for any $f_t \in \mathcal{F}_t$ with integer $t \leq T$, we have,

$$\mathcal{R}(f_t) \leq \widehat{\mathcal{R}}(f_t) + \widetilde{O}\left(\frac{L \times \mathsf{Complexity}}{\sqrt{m}} + B\sqrt{\frac{\log(1/\delta)}{m}}\right).$$

Please refer to Section 2 for a complete statement. Most of the aforementioned commonly used $\mathcal{A}$ and $\mathcal{L}$

satisfy the assumptions in Theorem 1. For example, in classification, the functional margin operator $\mathcal{M}(y,z)$ is 2-Lipschitz in $\widetilde{z}$. The ramp loss $\mathcal{L}_\gamma$ is uniformly bounded by 1 and $\frac{1}{\gamma}$-Lipschitz. In regression, $\mathcal{A}(y,z)$ is 1-Lipschitz in y and bounded since the input data are bounded. Then the ℓ_p loss becomes bounded and Lipschitz due to its bounded input.

Comparison with Existing Results. To better understand the obtained generalization bound and draw a comparison among existing literature, we instantiate Theorem 1 for sequence to sequence classification using vanilla RNNs. Recall that for classification tasks, we have $L = 1/\gamma$, $B = 1$ and $\mathcal{M}(y,z)$ is 2-Lipschitz in y. We list the corresponding generalization bounds in Table 1 according to the magnitude of $\|U\|_2$.

Table 1: Generalization bounds for vanilla RNNs in classification tasks (we only list the order of the gap $\mathcal{R}(f_t) - \widehat{\mathcal{R}}(f_t)$). The third column lists the result obtained in Zhang et al. (2018).

	Theorem 1	Zhang et al. (2018)
(I) $\|U\|_2 < 1$	$\widetilde{O}(d/\sqrt{m}\gamma)$	$\widetilde{O}(dt^2/\sqrt{m}\gamma)$
(II) $\|U\|_2 = 1$	$\widetilde{O}(dt/\sqrt{m}\gamma)$	$\widetilde{O}(dt^2/\sqrt{m}\gamma)$
(III) $\|U\|_2 > 1$	$\widetilde{O}(\sqrt{d^3t}/\sqrt{m}\gamma)$	$\widetilde{O}(dt^2\|U\|_2^t/\sqrt{m}\gamma)$

As can be seen, the obtained generalization bound only has a polynomial dependence on the size of vanilla RNNs, i.e., width d and sequence length t. Thus, we theoretically justify that the complexity of vanilla RNNs do not suffer from significant curse of dimensionality. Because they compute outputs y_t recursively using the same weight matrices, and their hidden states h_t are entrywise bounded.

We compare Theorem 1 with the generalization bound obtained in Zhang et al. (2018), which is of the order $\widetilde{O}\left(dt^2\|W\|_2\|V\|_2 \max\{1, \|U\|_2^t\}/\sqrt{m}\gamma\right)$, and we distinguish the same three different scenarios as listed in Table 1. Our bound is tighter by a factor of t^2 for case **(I)**, a factor of t for case **(II)**. Additionally, Zhang et al. (2018) fail to incorporate the boundedness condition of hidden state into their analysis, thus the generalization bound is exponential in t for case **(III)**. Our generalization bound, however, is still polynomial in d and t for case **(III)**.

Moreover, **(II)** is closely related to a few recent results on imposing orthogonal constraints on weight matrices to stabilize the training of RNNs (Saxe et al., 2013; Le et al., 2015; Arjovsky et al., 2016; Vorontsov et al., 2017; Zhang et al., 2018). We remark that from a learning theory perspective, **(II)** implies that orthogonal constraints can potentially help generalization.

We also present refined generalization bounds with additional matrix norm assumptions. These assumptions allow us to derive norm-based generalization bounds. We draw a comparison among these bounds and highlight their advantage under different scenarios.

Our theory can be further extended to several variants, including MGU and LSTM RNNs. Specifically, we show that the gated units in MGU and LSTM RNNs can introduce extra decaying factors to further reduce the dependence on d and t in generalization. Such an advantage in generalization makes these RNNs do not suffer from significant curse of dimensionality. To the best of our knowledge, these are the first results on generalization guarantees for these neural networks.

Notations: Given a vector $v \in \mathbb{R}^d$, we denote its Euclidean norm by $\|v\|_2^2 = \sum_{i=1}^d |v_i|^2$, and the infinity norm by $\|v\|_\infty = \max_j |v_j|$. Given a matrix $M \in \mathbb{R}^{m\times n}$, we denote the spectral norm by $\|M\|_2$ as the largest singular value of M, the Frobenius norm by $\|M\|_{\mathrm{F}}^2 = \mathrm{trace}(MM^\top)$, and the $(2,1)$ norm by $\|M\|_{2,1} = \sum_{i=1}^n \|M_{:,i}\|_2$. Given a function f, we denote the function infinity norm by $\|f\|_\infty = \sup |f|$. We use $\widetilde{O}(\cdot)$ to denote $O(\cdot)$ with hidden log factors. We denote $a \wedge b = \min\{a,b\}$.

2 Generalization of Vanilla RNNs

To establish the generalization bound, we start with imposing some mild assumptions.

Assumption 1. Input data are bounded, i.e., $\|x_{i,t}\|_2 \leq B_x$ for all $i = 1,\ldots,m$ and $t = 1,\ldots,T$.

Assumption 2. The spectral norms of weight matrices are bounded respectively, i.e., $\|U\|_2 \leq B_U$, $\|V\|_2 \leq B_V$, and $\|W\|_2 \leq B_W$.

Assumption 3. Activation operators σ_h and σ_y are Lipschitz with parameters ρ_h and ρ_y respectively, and $\sigma_h(0) = \sigma_y(0) = 0$. Additionally, σ_h is entrywise bounded by b.

Assumptions 1 and 2 are moderate assumptions. Moreover, Assumption 3 holds for most commonly used activation operators, such as $\sigma_h(\cdot) = \tanh(\cdot)$ and $\sigma_y(\cdot) = \mathrm{ReLU}(\cdot) = \max\{\cdot, 0\}$ (1-Lipschitz).

Recall vanilla RNNs compute $h_{i,t}$ and $y_{i,t}$ as follows,

$$h_{i,t} = \sigma_h\left(Uh_{i,t-1} + Wx_{i,t}\right) \quad \text{and} \quad y_{i,t} = \sigma_y\left(Vh_{i,t}\right),$$

where $U \in \mathbb{R}^{d_h\times d_h}$, $V \in \mathbb{R}^{d_y\times d_h}$, and $W \in \mathbb{R}^{d_h\times d_x}$. We consider multiclass classification tasks with the label $z \in \mathcal{Z} = \{1,\ldots,K\}$. Given a sequence $(x_t, z_t)_{t=1}^T$, we define $X_t \in \mathbb{R}^{d_x\times t}$ by concatenating $x_1,\ldots,x_t$ as columns of X_t. Recall that we denote $\mathcal{F}_t = \{f_t : X_t \mapsto y_t\}$ as the class of mappings from the first t inputs to the t-th output computed by vanilla RNNs.

As previously mentioned, we define the functional margin for the t-th output in vanilla RNNs as

$$\mathcal{M}(f_t(X_t), z_t) = [f_t(X_t)]_{z_t} - \max_{j\neq z_t}[f_t(X_t)]_j.$$

We further define a ramp loss $\mathcal{L}_\gamma(-\mathcal{M}(f_t(X_t), z_t)) : \mathbb{R} \mapsto \mathbb{R}^+$ to each margin, where $\mathcal{L}_\gamma$ is a piecewise linear function defined as

$$\mathcal{L}_\gamma(a) = \mathbb{1}\{a > 0\} + (1 + a/\gamma)\mathbb{1}\{-\gamma \leq a \leq 0\},$$

where $\mathbb{1}\{A\}$ denotes the indicator function of a set A. Accordingly, the ramp risk is defined as $\mathcal{R}_\gamma(f_t) = \mathbb{E}\left[\mathcal{L}_\gamma(-\mathcal{M}(f_t(X_t), z_t))\right]$, and its empirical counterpart is defined as $\widehat{\mathcal{R}}_\gamma(f_t) = \frac{1}{m}\sum_{i=1}^m \mathcal{L}_\gamma(-\mathcal{M}(f_t(X_{i,t}), z_{i,t}))$. We then present the formal statement of Theorem 1.

Theorem 2. Let activation operators σ_h and σ_y be given, and Assumptions 1–3 hold. Then for $(x_t, z_t)_{t=1}^T$ and $S = \{(x_{i,t}, z_{i,t})_{t=1}^T, i = 1, \ldots, m\}$ drawn i.i.d. from any underlying distribution over $\mathbb{R}^{d_x \times T} \times \{1, \ldots, K\}$, with probability at least $1-\delta$ over S, for every margin value $\gamma > 0$ and every $f_t \in \mathcal{F}_t$ for integer $t \leq T$, we have

$$\mathbb{P}(\widetilde{z}_t \neq z_t) \leq \widehat{\mathcal{R}}_\gamma(f_t) + 3\sqrt{\frac{\log\frac{2}{\delta}}{2m}} + O\left(\frac{d\rho_y B_V \lambda_t \sqrt{\log\left(t\sqrt{dm}\frac{\beta^t - 1}{\beta - 1}\right)}}{\sqrt{m}\gamma}\right), \quad (1)$$

where $d = \sqrt{d_x d_h + d_h^2 + d_h d_y}$, $\beta = \rho_h B_U$, and $\lambda_t = \min\left\{b\sqrt{d}, \rho_h B_x B_W \frac{\beta^t - 1}{\beta - 1}\right\}$.

Remark 1. To ease the presentation, we only provide the generalization bound for the classification task. Extensions to general tasks are straightforward by replacing functions $\mathcal{A}$ and $\mathcal{L}$ and substituting suitable values of L and B.

The generalization bound depends on the total number of weights, and the range of $\rho_h B_U$ in three cases as indicated in Section 1. More precisely, if $\rho_h B_U \lesssim (1 + \frac{1}{t^\alpha})$ for constant $\alpha > 0$ bounded away from zero, the generalization bound is of the order $\widetilde{O}\left(\frac{dt^\alpha}{\sqrt{m}\gamma}\right)$, which has a polynomial dependence on d and t. As can be seen, with proper normalization on model parameters, the model complexity of vanilla RNNs do not suffer from significant curse of dimensionality.

We also highlight a tradeoff between generalization and representation of vanilla RNNs. As can be seen, when $\rho_h B_U$ is strictly smaller than 1, the generalization bound is nearly independent on t. The hidden state, however, only has limited representation ability, since its magnitude diminishes as t grows large. On the contrary, when $\rho_h B_U$ is strictly greater than 1, the representation ability is amplified but the generalization becomes worse. As a consequence, recent empirical results show that imposing extra constraints or regularization, such as $U^\top U = I$ or $\|U\|_2 \leq 1$ (Saxe et al., 2013; Le et al., 2015; Arjovsky et al., 2016; Vorontsov et al., 2017; Zhang et al., 2018), helps balance the generalization and representation of RNNs.

3 Proof of Main Results

Our analysis is based on the PAC-learning framework. Due to space limit, we only present an outline of our proof. More technical details are deferred to Appendix A. Before we proceed, we first define the empirical Rademacher complexity as follows.

Definition 1 (Empirical Rademacher Complexity)**.** Let $\mathcal{H}$ be a function class and $S = \{s_1, \ldots, s_m\}$ be a collection of samples. The empirical Rademacher complexity of $\mathcal{H}$ given S is defined as

$$\mathfrak{R}_S(\mathcal{H}) = \mathbb{E}_\epsilon\left[\sup_{h \in \mathcal{H}} \frac{1}{m}\sum_{i=1}^m \epsilon_i h(s_i)\right],$$

where ϵ_i's are i.i.d. Rademacher random variables, i.e., $\mathbb{P}(\epsilon_i = 1) = \mathbb{P}(\epsilon_i = -1) = 0.5$.

We then proceed with our analysis. Recall that Mohri et al. (2012) give an empirical Rademacher complexity (ERC)-based generalization bound, which is restated in the following lemma with $\mathcal{F}_{\gamma,t} = \{(X_t, z_t) \mapsto \ell_\gamma(-\mathcal{M}(f_t(X_t), z_t)) : f_t \in \mathcal{F}_t\}$.

Lemma 1. Given a testing sequence $(x_t, z_t)_{t=1}^T$, with probability at least $1 - \delta$ over samples $S = \{(x_{i,t}, z_{i,t})_{t=1}^T, i = 1, \ldots, m\}$, for every margin value $\gamma > 0$ and any $f_t \in \mathcal{F}_t$, we have

$$\mathcal{R}_\gamma(f_t) \leq \widehat{\mathcal{R}}_\gamma(f_t) + 2\mathfrak{R}_S(\mathcal{F}_{\gamma,t}) + 3\sqrt{\frac{\log(2/\delta)}{2m}}.$$

Note that Lemma 1 adapts the original version (Theorem 3.1, Chapter 3.1, Mohri et al. (2012)) for the multiclass ramp loss, and we have $\mathbb{P}(\widetilde{z}_t \neq z_t) \leq \mathcal{R}_\gamma(f_t)$ by definition.

Now we only need to bound the ERC $\mathfrak{R}_S(\mathcal{F}_{\gamma,t})$. Our analysis consists of three steps. First, we characterize the Lipschitz continuity of vanilla RNNs w.r.t model parameters. Next, we bound the covering number of function class $\mathcal{F}_t$. At last, we derive an upper bound on $\mathfrak{R}_S(\mathcal{F}_{\gamma,t})$ via the standard machinery in the PAC-learning framework. Specifically, consider two different sets of weight matrices (U, V, W) and (U', V', W'). Given the same activation operators and input data, denote the t-th output as y_t and y_t' respectively. We characterize the Lipschitz property of $\|y_t\|_2$ w.r.t model parameters in the following lemma.

Lemma 2. Under Assumptions 1–3, given input $(x_t)_{t=1}^T$ and for any integer $t \leq T$, $\|y_t\|_2$ is Lipschitz in U, V and W, i.e.,

$$\|y_t - y_t'\|_2 \leq L_{U,t}\|U - U'\|_{\mathrm{F}} + L_{V,t}\|V - V'\|_{\mathrm{F}} + L_{W,t}\|W - W'\|_{\mathrm{F}},$$

where $L_{U,t} = \rho_h B_V B_W t a_t$, $L_{V,t} = B_W a_t$, and $L_{W,t} = B_V a_t$ with $a_t = \rho_y \rho_h B_x \frac{(\rho_h B_U)^t - 1}{\rho_h B_U - 1}$.

The detailed proof is provided in Appendix A.2. We give a simple example to illustrate the proof technique. Specifically, we consider a single layer network that outputs $y = \sigma(Wx)$, where x is the input, σ is an activation operator with Lipschitz parameter ρ, and W is a weight matrix. Such a network is Lipschitz in both x and W as follows. Given weight matrices W and W', we have

$$\|y - y'\|_2 = \|\sigma(Wx) - \sigma(W'x)\|_2 \leq \rho\|x\|_2\|W - W'\|_{\mathrm{F}}.$$

Additionally, given inputs x and x', we have

$$\|y - y'\|_2 = \|\sigma(Wx) - \sigma(Wx')\|_2 \leq \rho\|W\|_2\|x - x'\|_2.$$

Since vanilla RNNs are multilayer networks, Lemma 2 can be obtained by telescoping.

We remark that Lemma 2 is the key to the proof of our generalization bound, which separates the spectral norms of weight matrices and the total number of parameters.

Next, we bound the covering number of $\mathcal{F}_t$. Denote by $\mathcal{N}(\mathcal{F}_t, \epsilon, \mathrm{dist}(\cdot,\cdot))$ the minimal cardinality of a subset $\mathcal{C} \subset \mathcal{F}_t$ that covers $\mathcal{F}_t$ at scale ϵ w.r.t the metric $\mathrm{dist}(\cdot,\cdot)$, such that for any $f_t \in \mathcal{F}_t$, there exists $\widehat{f}_t \in \mathcal{C}$ satisfying $\mathrm{dist}(f_t, \widehat{f}_t) = \sup_{X_t}\|f_t(X_t) - \widehat{f}_t(X_t)\|_2 \leq \epsilon$. The following lemma gives an upper bound on $\mathcal{N}(\mathcal{F}_t, \epsilon, \mathrm{dist}(\cdot,\cdot))$.

Lemma 3. Under Assumptions 1–3, given any $\epsilon > 0$, the covering number of $\mathcal{F}_t$ satisfies

$$\mathcal{N}(\mathcal{F}_t, \epsilon, \mathrm{dist}(\cdot,\cdot)) \leq \left(1 + \frac{6c\sqrt{d}t\left((\rho_h B_U)^t - 1\right)}{\epsilon\left(\rho_h B_U - 1\right)}\right)^{3d^2},$$

where $c = \rho_y \rho_h B_V B_W B_x \max\{1, \rho_h B_U\}$.

The detailed proof is provided in Appendix A.3. We briefly explain the proof technique. Given activation operators, since vanilla RNNs are in parametric forms, f_t has a one-to-one correspondence to its weight matrices U, V, and W. Lemma 2 implies that $\mathrm{dist}(\cdot,\cdot)$ is controlled by the Frobenius norms of the differences of weight matrices. Thus, it suffices to bound the covering numbers of three weight matrices. The product of covering numbers of three weight matrices gives us Lemma 3.

Lastly, we give an upper bound on $\mathfrak{R}_S(\mathcal{F}_{\gamma,t})$ in the following lemma.

Lemma 4. Under Assumptions 1–3, given activation operators and samples $S = \{(x_{i,t}, z_{i,t})_{t=1}^T, i = 1, \ldots, m\}$, the empirical Rademacher complexity $\mathfrak{R}_S(\mathcal{F}_{\gamma,t})$ satisfies

$$\mathfrak{R}_S(\mathcal{F}_{\gamma,t}) = O\Bigg(d \min\left\{b\sqrt{d}, \rho_h B_x B_W \frac{(\rho_h B_U)^t - 1}{\rho_h B_U - 1}\right\} \times \frac{\rho_y B_V \sqrt{\log\left(t\sqrt{dm}\frac{(\rho_h B_U)^t - 1}{\rho_h B_U - 1}\right)}}{\sqrt{m}\gamma}\Bigg).$$

The detailed proof is provided in Appendix A.4. Our proof exploits the Lipschitz continuity of $\mathcal{M}$ and ℓ_γ, and uses Dudley's entropy integral as the standard machinery to establish Lemma 4. Combining Lemma 1 and Lemma 4, we complete the proof.

4 Refined Generalization Bounds

When additional norm constraints on weight matrices U, V and W are available, we can further refine generalization bounds. Specifically, we consider assumptions as follows.

Assumption 4. The weight matrices satisfy $\|U\|_{2,1} \leq M_U$, $\|V\|_{2,1} \leq M_V$, and $\|W\|_{2,1} \leq M_W$.

Assumption 5. The weight matrices satisfy $\|U\|_{\mathrm{F}} \leq B_{U,\mathrm{F}}$, $\|V\|_{\mathrm{F}} \leq B_{V,\mathrm{F}}$, and $\|W\|_{\mathrm{F}} \leq B_{W,\mathrm{F}}$.

Note that Assumption 4 appears in Bartlett et al. (2017) and Assumption 5 appears in Neyshabur et al. (2017). We have an equivalent relation between matrix norms, i.e., $\|\cdot\|_2 \leq \|\cdot\|_{2,1} \leq \sqrt{d}\|\cdot\|_{\mathrm{F}} \leq d\|\cdot\|_2$. Comparing to Assumption 2, Assumptions 4 and 5 further restrict the model class. We then establish refined empirical Rademacher complexities for vanilla RNNs, the corresponding generalization bounds follows immediately.

Theorem 3. Let activation operators σ_h and σ_y be given, and Assumptions 1-3 hold. Then for $(x_t, z_t)_{t=1}^T$ and $S = \{(x_{i,t}, z_{i,t})_{t=1}^T, i = 1, \ldots, m\}$ drawn i.i.d. from any underlying distribution over $\mathbb{R}^{d_x \times T} \times \{1, \ldots, K\}$, with probability at least $1-\delta$ over S, for every margin value $\gamma > 0$ and every $f_t \in \mathcal{F}_t$ for integer $t \leq T$, the following two bounds hold:

- Suppose Assumption 4 also holds. We have
$$\mathfrak{R}_S(\mathcal{F}_{\gamma,t}) = O\left(\frac{t\alpha S_{2,1}\frac{(\rho_h B_U)^t - 1}{\rho_h B_U - 1}\sqrt{\log d}\log(\sqrt{d}m)}{\sqrt{m}\gamma}\right), \tag{2}$$
where $\alpha = \rho_h^2 \rho_y B_V B_W B_x$, $S_{2,1} = M_U + M_V + M_W$, and $d = \sqrt{d_x d_h + d_h^2 + d_h d_y}$.
- Suppose Assumption 5 also holds. We have
$$\mathfrak{R}_S(\mathcal{F}_{\gamma,t}) = O\left(\frac{\alpha' B_U \lambda_t S_{\mathrm{F}}\frac{(\rho_h B_U)^t - 1}{\rho_h B_U - 1}\sqrt{d\log d}}{\sqrt{m}\gamma}\right), \tag{3}$$
where $\alpha' = \rho_h \rho_y B_W B_x$, $S_{\mathrm{F}} = B_{U,\mathrm{F}} + B_{W,\mathrm{F}} + B_{V,\mathrm{F}}$, $\lambda_t = \min\left\{b\sqrt{d}, \rho_h B_x B_W \frac{(\rho_h B_U)^t - 1}{\rho_h B_U - 1}\right\}$, and $d = \sqrt{d_x d_h + d_h^2 + d_h d_y}$.

The detailed proof is provided in Appendix B.1. The first bound (2) adapts the matrix covering lemma in Bartlett et al. (2017). The second bound (3) adapts the PAC-Bayes approach (Neyshabur et al., 2017) by analyzing the divergence when imposing small perturbations on the weight matrices.

We highlight the improvements of the obtained refined generalization bounds: When the weight matrices are approximately low rank, that is, $\|\cdot\|_{2,1} \ll d\|\cdot\|_2$ and $\|\cdot\|_F \ll \sqrt{d}\|\cdot\|_2$, for $\beta \leq 1$, bound (3) improves bound (1) by reducing dependence on d. Additionally, if $t\left(M_U + M_V + M_W\right) < d$, bound (2) also tightens bound (1). Note that $t\left(M_U + M_V + M_W\right) < d$ implies that the input sequence is relatively short.

5 Extensions to MGU, LSTM, and Conv RNNs

We extend our analysis to Minimal Gated Unit (MRU), Long Short-Term Memory (LSTM) RNNs, and convolutional RNNs (Conv RNNs).

The MGU RNNs compute

$$\begin{aligned} r_t &= \sigma(W_r x_t + U_r h_{t-1}), \\ \widetilde{h}_t &= \sigma_h\left(W_h x_t + U_h(r_t \odot h_{t-1})\right), \\ h_t &= (1 - r_t) \odot h_{t-1} + r_t \odot \widetilde{h}_t, \end{aligned}$$

where $W_r, W_h \in \mathbb{R}^{d_h \times d_x}$, $U_r, U_h \in \mathbb{R}^{d_h \times d_h}$, $V \in \mathbb{R}^{d_y \times d_h}$, and $r_t \in \mathbb{R}^{d_h}$. The notation $\odot$ denotes the Hadamard product (entrywise product) of vectors. Denote by $\mathcal{F}_{g,t}$ the class of mappings from the first t inputs to the t-th output computed by gated (MGU or LSTM) RNNs. For simplicity, we consider σ being the sigmoid function, i.e., $\sigma(x) = (1+\exp(-x))^{-1}$, $\sigma_h(\cdot) = \tanh(\cdot)$, and σ_y being ρ_y-Lipschitz with $\sigma_y(0) = 0$. Extensions to general Lipschitz activation operators as in Assumption 3 are straightforward. Suppose we have $h_0 = 0$ and the following assumption.

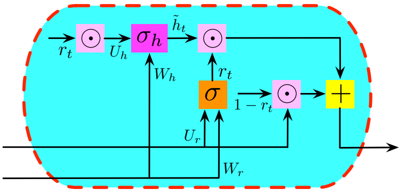

Figure 2: A building block of MGU RNNs.

Assumption 6. All the weight matrices have bounded spectral norms respectively, i.e. $\|W_r\|_2 \leq B_{W_r}, \|W_h\|_2 \leq B_{W_h}, \|U_r\|_2 \leq B_{U_r}, \|U_h\|_2 \leq B_{U_h}$, and $\|V\|_2 \leq B_V$.

A similar argument for vanilla RNNs yields a generalization bound of MGU RNNs as follows.

Theorem 4. Let the activation operator σ_y be given and Assumptions 1 and 6 hold. Then for $(x_t, z_t)_{t=1}^T$ and $S = \left\{(x_{i,t}, z_{i,t})_{t=1}^T, i = 1, \dots, m\right\}$ drawn i.i.d. from any underlying distribution over $\mathbb{R}^{d_x \times T} \times \{1, \dots, K\}$, with probability at least $1 - \delta$ over S, for every margin value $\gamma > 0$ and every $f_t \in \mathcal{F}_{g,t}$ for integer $t \leq T$, we have

$$\mathbb{P}\left(\widetilde{z}_t \neq z_t\right) \leq \widehat{\mathcal{R}}_\gamma(f_t) + 3\sqrt{\frac{\log \frac{2}{\delta}}{2m}}$$
$$+ O\left(\frac{d\rho_y B_V\left(\sqrt{d} \wedge B_{W_h} B_x \frac{\beta^t - 1}{\beta - 1}\right)\sqrt{\log\left(\frac{\theta^t - 1}{\theta - 1} d\sqrt{m}\right)}}{\sqrt{m}\gamma}\right),$$

where $\beta = \max_j \left\{\, \|1 - r_j\|_\infty + B_{U_h} \|r_j\|_\infty^2 \,\right\}$, $\theta = \beta + 2B_{U_r} + B_{U_r} B_{U_h}$, $d = \max\{d_x, d_y, d_h\}$.

The detailed proof is provided in Appendix C.1. As can be seen, r_t shrinks the magnitude of hidden state to reduce the dependence on d and t in generalization. As a result, with proper normalization of weight matrices, the generalization bound of MGU RNNs is less dependent on d, t.

The LSTM RNNs are more complicated than MGU RNNs, which introduce more gates to control the information flow in RNNs. LSTM RNNs have two hidden states, and compute them as,

$$\begin{aligned} g_t &= \sigma(W_g x_t + U_g h_{t-1}), & r_t &= \sigma(W_r x_t + U_r h_{t-1}), \\ o_t &= \sigma(W_o x_t + U_o h_{t-1}), & \widetilde{c}_t &= \sigma_c\left(W_c x_t + U_c h_{t-1}\right), \\ c_t &= g_t \odot c_{t-1} + r_t \odot \widetilde{c}_t, & h_t &= o_t \odot \tanh(c_t), \end{aligned}$$

where $W_g, W_r, W_o, W_c \in \mathbb{R}^{d_h \times d_x}$, $U_g, U_r, U_o, U_c \in \mathbb{R}^{d_h \times d_h}$, and $g_t, r_t, o_t \in \mathbb{R}^{d_h}$. For simplicity, we also consider σ being the sigmoid function, and $\sigma_c(\cdot) = \tanh(\cdot)$. The t-th output is $y_t = \sigma_y(V h_t)$, where $V \in \mathbb{R}^{d_y \times d_h}$, and σ_y is ρ_y-Lipschitz with $\sigma_y(0) = 0$. Suppose we have $h_0 = c_0 = 0$ and the following assumption.

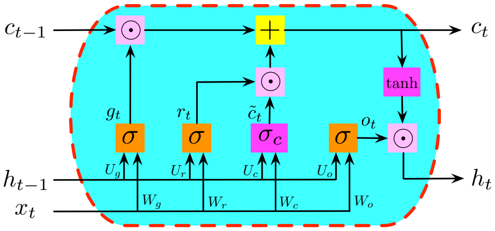

Figure 3: A building block of LSTM RNNs.

Assumption 7. The spectral norms of weight matrices are bounded respectively, i.e. $\|W_g\|_2 \leq B_{W_g}, \|W_r\|_2 \leq B_{W_r}, \|W_o\|_2 \leq B_{W_o}, \|W_c\|_2 \leq B_{W_c}, \|U_g\|_2 \leq B_{U_g}, \|U_r\|_2 \leq B_{U_r}, \|U_o\|_2 \leq B_{U_o}, \|U_h\|_2 \leq B_{U_h}$, and $\|V\|_2 \leq B_V$.

For properly normalized weight matrices W_o and U_o, the generalization bound of LSTM RNNs is given in the following theorem.

Theorem 5. Let the activation operator σ_y be given and Assumptions 1 and 7 hold. Then for $(x_t, z_t)_{t=1}^T$ and $S = \left\{(x_{i,t}, z_{i,t})_{t=1}^T, i = 1, \dots, m\right\}$ drawn i.i.d.

from any underlying distribution over $\mathbb{R}^{d_x \times T} \times \{1, \ldots, K\}$, with probability at least $1 - \delta$ over S, for every margin value $\gamma > 0$ and every $f_t \in \mathcal{F}_{g,t}$ for integer $t \leq T$, we have

$$\mathbb{P}(\widetilde{z}_t \neq z_t) \leq \widehat{\mathcal{R}}_\gamma(f_t) + 3\sqrt{\frac{\log \frac{2}{\delta}}{2m}}$$

$$+ O\left(\frac{d\rho_y B_V \left(\sqrt{d} \wedge B_{W_c} B_x \frac{\beta^t - 1}{\beta - 1}\right) \sqrt{\log\left(\frac{\theta^t - 1}{\theta - 1} d\sqrt{m}\right)}}{\sqrt{m}\gamma}\right),$$

where $\beta = \max\left\{\|g_j\|_\infty + B_{U_c}\|r_j\|_\infty \|o_j\|_\infty\right\}$, $\theta = \beta + B_{U_g} + B_{U_r} + B_{U_o}$, and $d = \max\{d_x, d_y, d_h\}$.

The detailed proof is provided in Appendix C.2. Similar to MGU RNNs, LSTM RNNs also introduce extra decaying factors to reduce the dependence on d and t in generalization. However, LSTM RNNs are more complicated, but more flexible than MGU RNNs, since three factors, r_t, o_t and g_t are used to jointly control the spectrum of U_c. We further remark that LSTM RNNs need spectral norms of weight matrices, W_g, W_r, W_o, U_g, U_r and U_o, to be properly controlled for obtaining better generalization bounds.

We further extend our analysis to Convolutional RNNs (Conv RNNs). Conv RNNs integrate convolutional filters and recurrent neural networks. Specifically, we consider input $x \in \mathbb{R}^d$ and k-channel k-dimensional convolutional filters $\mathcal{I}_1, \ldots, \mathcal{I}_k \in \mathbb{R}^k$ followed by an average pooling layer over the k channels for reducing dimensionality. Extensions to convolution with strides and other kinds of average pooling layers (e.g., blockwise pooling) are straightforward.

Here we denote the circulant-like matrix generated by $\mathcal{I}_i$ as

$$C_i = \begin{bmatrix} \mathcal{I}_i^\top & & \underbrace{0 \ldots\ldots\ldots 0}_{d-k} \\ 0 & \mathcal{I}_i^\top & \underbrace{0 \ldots\ldots 0}_{d-k-1} \\ & \ddots & \\ \underbrace{0 \ldots\ldots\ldots 0}_{d-k} & & \mathcal{I}_i^\top \end{bmatrix} \in \mathbb{R}^{(d-k+1) \times d},$$

and write $W_\mathcal{I} = [C_1^\top, \ldots, C_k^\top]^\top$. We further denote $P = \frac{1}{k}\underbrace{[I_{d-k+1} \; I_{d-k+1} \; \cdots \; I_{d-k+1}]}_{\text{totally } k \text{ identity matrices}}$, where I_d denotes the d-dimensional identity matrix. Define $\mathcal{I} = [\mathcal{I}_1, \ldots, \mathcal{I}_k]$, and $\mathcal{I} * x = PW_\mathcal{I} x$. Given a sample $(x_t, z_t)_{t=1}^T$, the Conv RNNs compute h_t and y_t as follows,

$$h_t = \sigma_h(\mathcal{U} * h_{t-1} + \mathcal{W} * x_t), \quad \text{and} \quad y_t = \sigma_y(\mathcal{V} * h_t),$$

where $h_t, x_t \in \mathbb{R}^d$, and $\mathcal{U}, \mathcal{V}, \mathcal{W} \in \mathbb{R}^{k \times k}$ are matrices with column vectors being k-dimensional convolutional filters. We use zero-padding to ensure the output dimension of convolutional filters matches the input (Krizhevsky et al., 2012). To get y_t, we convolve h_t with $\mathcal{V}$ followed by an average pooling to reduce the dimension to K. Since we aim to show that Conv RNNs reduce the dependence on d in generalization through parameter sharing, we simplify the notations to assume $h_0 = 0$, and impose the following assumption. Extensions to general settings are straightforward.

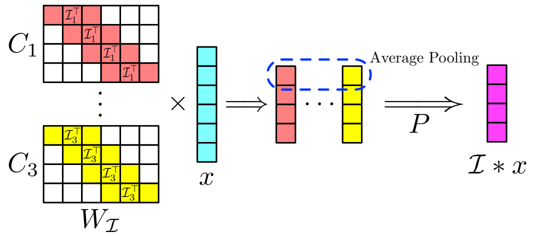

Figure 4: *Illustration of input $x \in \mathbb{R}^6$ convolving with 3-channel 3-dimensional convolutional filters $\mathcal{I}_1$,$\mathcal{I}_2$, and $\mathcal{I}_3$, followed by an average pooling.*

Assumption 8. The activation operators σ_h and σ_y are 1-Lipschitz with $\sigma_h(0) = \sigma_y(0) = 0$. σ_h is entrywise bounded by 1. The convolutional filters $\mathcal{U}$, $\mathcal{V}$, and $\mathcal{W}$ are orthogonal with normalized columns, i.e., $\mathcal{U}^\top\mathcal{U} = \mathcal{U}\mathcal{U}^\top = \frac{1}{k}I_k, \mathcal{V}^\top\mathcal{V} = \mathcal{V}\mathcal{V}^\top = \frac{1}{k}I_k$, and $\mathcal{W}^\top\mathcal{W} = \mathcal{W}\mathcal{W}^\top = \frac{1}{k}I_k$.

We remark that the orthogonality constraints enhance the diversity among convolutional filters (Xie et al., 2017; Huang et al., 2017). Additionally, the normalization factor $\frac{1}{k}$ is to control the spectral norms of $W_\mathcal{U}$, $W_\mathcal{V}$, and $W_\mathcal{W}$, which prevents the blowup of hidden state. Denote by $\mathcal{F}_{c,t}$ the class of mappings from the first t inputs to the t-th output computed by Conv RNNs. Then the generalization bound is given in the following theorem.

Theorem 6. Let activation operators σ_h and σ_y be given, and Assumptions 1 and 8 hold. Then for $(x_t, z_t)_{t=1}^T$ and $S = \left\{(x_{i,t}, z_{i,t})_{t=1}^T, i = 1, \ldots, m\right\}$ drawn i.i.d. from any underlying distribution over $\mathbb{R}^{d \times T} \times \{1, \ldots, K\}$, with probability at least $1 - \delta$ over S, for every margin value $\gamma > 0$ and every $f_t \in \mathcal{F}_{c,t}$ for integer $t \leq T$, we have

$$\mathbb{P}(\widetilde{z}_t \neq z_t) \leq \widehat{\mathcal{R}}_\gamma(f_t) + O\left(\frac{kt\sqrt{\log(dtm)}}{\sqrt{m}\gamma} + \sqrt{\frac{\log\frac{1}{\delta}}{m}}\right).$$

The detailed proof is provided in C.3. Similar to the analysis of vanilla RNNs, our proof is based on the Lipschitz continuity of Conv RNNs with respect to its model parameters in the convolutional filters. Specifically, by Assumption 8, the spectral norms of $W_\mathcal{U}$, $W_\mathcal{V}$, and $W_\mathcal{W}$ are all bounded by 1. Combining with the inequality, $\|W_\mathcal{U}\|_\mathrm{F} \leq \sqrt{d}\|\mathcal{U}\|_\mathrm{F}$, we have $\|y_t - y_t'\|_2 \leq L_{\mathcal{V},t}\|\mathcal{V} - \mathcal{V}'\|_\mathrm{F} + L_{\mathcal{U},t}\|\mathcal{U} - \mathcal{U}'\|_\mathrm{F} + L_{\mathcal{W},t}\|\mathcal{W} - \mathcal{W}'\|_\mathrm{F}$, where $L_{\mathcal{U},t}$, $L_{\mathcal{V},t}$, and $L_{\mathcal{W},t}$ are polynomials in d and t. Additionally, observe that the total number of parameters in a Conv RNN is at most $3k^2$, which is independent of input dimension d. As a consequence, the generalization bound of Conv RNNs only has a lieanr dependence on k and t.

6 Numerical Evaluation

We demonstrate a comparison among our obtained generalization bound with Bartlett et al. (2017), Neyshabur et al. (2017), and Zhang et al. (2018). Specifically, we train[1] a vanilla RNN on the wikitext language modeling dataset (Merity et al., 2016). We take $\sigma_h = \tanh$ and set the hidden state $h \in \mathbb{R}^{128}$ and the input $x \in \mathbb{R}^{14}$ with $\|x\|_2 \leq 1$. Accordingly, we have $d = 128$ and take the sequence length $t = 56$. We list the complexity bounds for vanilla RNNs in Theorem 2 (Ours), Zhang et al. (2018) (Bound 1), (2) of Theorem 3 (Bound 2), and (3) of Theorem 3 (Bound 3) neglecting common log factors in d and t:

- Ours: $dB_V \min\left\{\sqrt{d}, B_W \frac{B_U^t - 1}{B_U - 1}\right\} \sqrt{\log\left(\frac{B_U^t - 1}{B_U - 1}\right)}$;
- Bound 1: $dt^2 B_V B_W \max\{1, B_U^t\}$;
- Bound 2: $B_V B_W \left(M_U + M_V + M_W\right) t \frac{B_U^t - 1}{B_U - 1}$;
- Bound 3: $\left(\min\{\sqrt{d}, B_W \frac{B_U^t - 1}{B_U - 1}\} B_U + B_W\right) \frac{B_U^t - 1}{B_U - 1}$ $\times \sqrt{d(B_{U,\mathrm{F}}^2 + B_{W,\mathrm{F}}^2 + B_{V,\mathrm{F}}^2)}$.

The corresponding complexity bounds are shown in Figure 5. As can be seen, our complexity bound in Theorem 2 is much smaller than Bounds 1-3. In more detail, the trained vanilla RNN has $B_U = 2.6801 > 1$. As discussed earlier, for $B_U > 1$, only our bound in Theorem 2 is polynomial in the size of the network, while Bounds 1-3 are all exponential in t. The resulting complexity bounds corroborate such a conclusion.

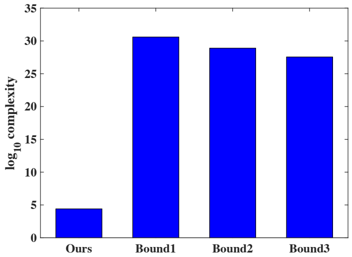

Figure 5: Complexity bounds on wikitext dataset.

We also observe that Bound 3 is smaller than Bound 2. The reason behind is that the weight matrices in the trained vanilla RNN have relatively small Frobenius norms but large $(2,1)$ norms. Taking matrix U as an example, we have $B_{U,\mathrm{F}} = 13.6823$ and $M_U = 154.5439$. Then, we can calculate the stable rank $\frac{B_{U,\mathrm{F}}}{B_U} = 5.1 < \sqrt{d}/2$, and the ratio $\frac{M_U}{B_{U,\mathrm{F}}} = 11.3 \approx \sqrt{d}$. This implies that the singular values of U are not evenly distributed, while the norms of row vectors in U are approximately equal.

7 Discussions and Open Problems

(I) Tighter bounds: Our obtained generalization bounds depend on the spectral norms of weight matrices and the network size. Can we exploit other modeling structures to further reduce the dependence on the network size? Or can we find better choices of norms of weight matrices that yield better bounds?

(II) Margin value: Our generalization bounds depend on the margin value of the predictors. As can be seen, a larger margin value yields a better generalization bound. However, establishing a sharp characterization of the margin value is technically very challenging, because of its complicated dependence on the underlying data distribution and the training algorithm.

(III) Implicit bias of SGD: Numerous empirical evidences have already shown that RNNs trained by stochastic gradient descent (SGD) algorithms have superior generalization performance. There have been a few theoretical results showing that SGD tends to yield low complexity models, which can generalize (Neyshabur et al., 2014, 2015; Zhang et al., 2016; Soudry et al., 2017). Can we extend this argument to RNNs? For example, can SGD always yield weight matrices with well controlled spectra? This is crucial to the generalization of MGU and LSTM RNNs.

(IV) Adaptivity to the underlying distribution: The current PAC-Learning framework focuses on the worst case. Taking classification as an example, the theoretical analysis holds even when the input features and labels are completely independent. Therefore, this often yields very pessimistic results. For many real applications, however, data are not obtained adversarially. Some recent empirical evidences suggest that the generalization of neural networks seems very adaptive to the underlying distribution: Easier tasks lead to low complexity neural networks, while harder ones lead to highly complex neural networks. Unfortunately, none of the existing analysis can take the underlying distribution into consideration.

(V) Sequentially dependent data: To extend the analysis to scenarios where input sequences are dependent is quite challenging and largely open. Rakhlin et al. (2015) propose a so-called "Sequential Rademacher Complexity" to quantify the model complexity with dependent data. Their bound however, is exponential in the depth of a neural network, even with proper normalization on the weight matrices. Kuznetsov and Mohri (2017) also derive generalization bounds for dependent data under mixing conditions. They assume block independence for a subsample selection trick. The extension to fully dependent data is beyond the scope of this paper. We leave it for future investigation.

[1] We adopt the code: https://github.com/pytorch/examples/tree/master/word_language_model.

References

Arjovsky, M., Shah, A. and Bengio, Y. (2016). Unitary evolution recurrent neural networks. In *International Conference on Machine Learning*.

Bahdanau, D., Cho, K. and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bartlett, P. L., Foster, D. J. and Telgarsky, M. J. (2017). Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Choi, E., Bahadori, M. T., Schuetz, A., Stewart, W. F. and Sun, J. (2016a). Doctor ai: Predicting clinical events via recurrent neural networks. In *Machine Learning for Healthcare Conference*.

Choi, E., Schuetz, A., Stewart, W. F. and Sun, J. (2016b). Using recurrent neural network models for early detection of heart failure onset. *Journal of the American Medical Informatics Association*, **24** 361–370.

Dasgupta, B. and Sontag, E. D. (1996). Sample complexity for learning recurrent perceptron mappings. In *Advances in Neural Information Processing Systems*.

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K. and Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Golowich, N., Rakhlin, A. and Shamir, O. (2017). Size-independent sample complexity of neural networks. *arXiv preprint arXiv:1712.06541*.

Graves, A. (2012). Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*.

Graves, A., Fernández, S., Gomez, F. and Schmidhuber, J. (2006). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*. ACM.

Graves, A., Mohamed, A.-r. and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE.

Gregor, K., Danihelka, I., Graves, A., Rezende, D. J. and Wierstra, D. (2015). Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*.

Haussler, D. (1992). Decision theoretic generalizations of the pac model for neural net and other learning applications. *Information and Computation*, **100** 78–150.

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

Huang, L., Liu, X., Lang, B., Yu, A. W. and Li, B. (2017). Orthogonal weight normalization: Solution to optimization over multiple dependent stiefel manifolds in deep neural networks. *arXiv preprint arXiv:1709.06079*.

Jozefowicz, R., Zaremba, W. and Sutskever, I. (2015). An empirical exploration of recurrent network architectures. In *International Conference on Machine Learning*.

Karpathy, A. and Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Koiran, P. (1998). Vapnik-chervonenkis dimension of recurrent neural networks. *Discrete Applied Mathematics*, **86** 63–79.

Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*.

Kuznetsov, V. and Mohri, M. (2017). Generalization bounds for non-stationary mixing processes. *Machine Learning*, **106** 93–117.

Le, Q. V., Jaitly, N. and Hinton, G. E. (2015). A simple way to initialize recurrent networks of rectified linear units. *arXiv preprint arXiv:1504.00941*.

Lee, C.-H. and Teng, C.-C. (2000). Identification and control of dynamic systems using recurrent fuzzy neural networks. *IEEE Transactions on fuzzy systems*, **8** 349–366.

Li, X., Lu, J., Wang, Z., Haupt, J. and Zhao, T. (2018). On tighter generalization bound for deep neural networks: Cnns, resnets, and beyond. *arXiv preprint arXiv:1806.05159*.

Lipton, Z. C., Kale, D. C., Elkan, C. and Wetzel, R. (2015). Learning to diagnose with lstm recurrent neural networks. *arXiv preprint arXiv:1511.03677*.

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

Mikolov, T., Karafiát, M., Burget, L., Černockỳ, J. and Khudanpur, S. (2010). Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*.

Mohri, M., Rostamizadeh, A. and Talwalkar, A. (2012). *Foundations of machine learning*. MIT press.

Neyshabur, B., Bhojanapalli, S., McAllester, D. and Srebro, N. (2017). A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*.

Neyshabur, B., Salakhutdinov, R. R. and Srebro, N. (2015). Path-sgd: Path-normalized optimization in deep neural networks. In *Advances in Neural Information Processing Systems*.

Neyshabur, B., Tomioka, R. and Srebro, N. (2014). In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*.

Rakhlin, A., Sridharan, K. and Tewari, A. (2015). Online learning via sequential complexities. *The Journal of Machine Learning Research*, **16** 155–186.

Saxe, A. M., McClelland, J. L. and Ganguli, S. (2013). Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*.

Soudry, D., Hoffer, E. and Srebro, N. (2017). The implicit bias of gradient descent on separable data. *arXiv preprint arXiv:1710.10345*.

Sutskever, I., Vinyals, O. and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*.

Vorontsov, E., Trabelsi, C., Kadoury, S. and Pal, C. (2017). On orthogonality and learning recurrent networks with long term dependencies. *arXiv preprint arXiv:1702.00071*.

Xie, D., Xiong, J. and Pu, S. (2017). All you need is beyond a good init: Exploring better solution for training extremely deep convolutional neural networks with orthonormality and modulation. *arXiv preprint arXiv:1703.01827*.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R. and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*.

Yoo, S. J., Park, J. B. and Choi, Y. H. (2006). Adaptive dynamic surface control of flexible-joint robots using self-recurrent wavelet neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, **36** 1342–1355.

Zhang, C., Bengio, S., Hardt, M., Recht, B. and Vinyals, O. (2016). Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*.

Zhang, J., Lei, Q. and Dhillon, I. S. (2018). Stabilizing gradients for deep neural networks via efficient svd parameterization. *arXiv preprint arXiv:1803.09327*.

Zhou, G.-B., Wu, J., Zhang, C.-L. and Zhou, Z.-H. (2016). Minimal gated unit for recurrent neural networks. *International Journal of Automation and Computing*, **13** 226–234.