Learning in Gated Neural Networks

Ashok Vardhan Makkuva* **Sreeram Kannan†** **Sewoong Oh†** **Pramod Viswanath***
*University of Illinois at Urbana-Champaign
†University of Washington

Abstract

Gating is a key feature in modern neural networks including LSTMs, GRUs and sparsely-gated deep neural networks. The backbone of such gated networks is a mixture-of-experts layer, where several experts make regression decisions and gating controls how to weigh the decisions in an input-dependent manner. Despite having such a prominent role in both modern and classical machine learning, very little is understood about parameter recovery of mixture-of-experts since gradient descent and EM algorithms are known to be stuck in local optima in such models.

In this paper, we perform a careful analysis of the optimization landscape and show that with appropriately designed loss functions, gradient descent can indeed learn the parameters of a MoE accurately. A key idea underpinning our results is the design of two *distinct* loss functions, one for recovering the expert parameters and another for recovering the gating parameters. We demonstrate the first sample complexity results for parameter recovery in this model for any algorithm and demonstrate significant performance gains over standard loss functions in numerical experiments.

1 Introduction

In recent years, *gated recurrent neural networks (RNNs)* such as LSTMs and GRUs have shown remarkable successes in a variety of challenging machine learning tasks such as machine translation, image captioning, image generation, hand writing generation, and speech recognition (Sutskever et al., 2014; Vinyals et al., 2014; Graves et al., 2013; Gregor et al., 2015; Graves, 2013).

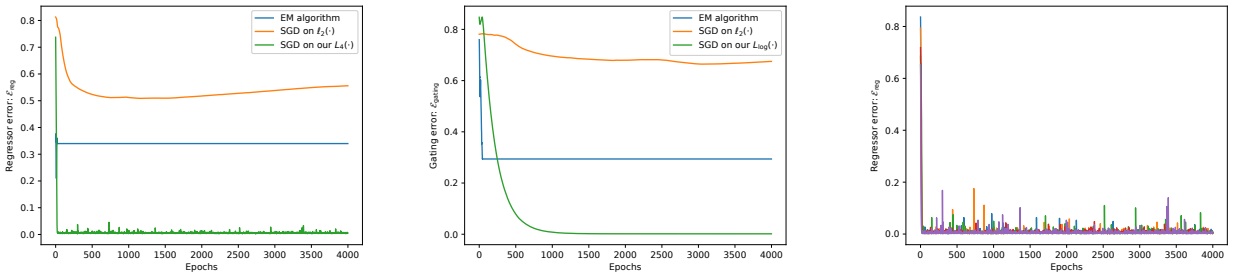
Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

A key interesting aspect and an important reason behind the success of these architectures is the presence of a *gating mechanism* that dynamically controls the flow of the past information to the current state at each time instant. In addition, it is also well known that these gates prevent the vanishing (and exploding) gradient problem inherent to traditional RNNs (Hochreiter and Schmidhuber, 1997).

Surprisingly, despite their widespread popularity, there is very little theoretical understanding of these gated models. In fact, basic questions such as learnability of the parameters still remain open. Even for the simplest vanilla RNN architecture, this question was open until the very recent works of Allen-Zhu et al. (2018) and Allen-Zhu and Li (2019), which provided the first theoretical guarantees of SGD for vanilla RNN models in the presence of non-linear activations. While this demonstrates that the theoretical analysis of these simpler models has itself been a challenging task, gated RNNs have an additional level of complexity in the form of gating mechanisms, which further enhances the difficulty of the problem. This motivates us to ask the following question:

Question 1. Given the complicated architectures of LSTMs/GRUs, can we find analytically tractable substructures of these models?

We believe that addressing the above question can provide new insights into a principled understanding of gated RNNs. In this paper, we make progress towards this and provide a positive answer to the question. In particular, we make a non-trivial connection that a GRU (gated recurrent unit) can be viewed as a time-series extension of a basic building block, known as *Mixture-of-Experts (MoE)* (Jacobs et al., 1991; Jordan and Jacobs, 1994). In fact, much alike LSTMs/GRUs, MoE is itself a widely popular gated neural network architecture and has found success in a wide range of applications (Tresp, 2001; Collobert et al., 2002; Rasmussen and Ghahramani, 2002; Yuksel et al., 2012; Masoudnia and Ebrahimpour, 2014; Ng and Deisenroth, 2014; Eigen et al., 2014). In recent years, there is also a growing interest in the fiels of natural language processing and computer vision to build complex



(a) Regressor error (b) Gating error (c) $L_4(\cdot)$ over different initializations

Figure 1: Our proposed losses L_4 (defined in Eq. (6)) and $L_{\log}$ (defined in Eq. (8)) to learn the respective regressor and gating parameters of a MoE model in Eq. (1) achieve much better empirical results than the standard methods.

neural networks incorporating MoE models to address challenging tasks such as machine translation (Gross et al., 2017; Shazeer et al., 2017). Hence the *main goal* of this paper is to study MoE in close detail, especially with regards to learnability of its parameters.

The canonical MoE model is the following: Let $k \in \mathbb{N}$ denote the number of mixture components (or equivalently neurons). Let $x \in \mathbb{R}^d$ be the input vector and $y \in \mathbb{R}$ be the corresponding output. Then the relationship between x and y is given by:

$$y = \sum_{i=1}^{k} z_i \cdot g(\langle a_i^*, x \rangle) + \xi, \quad \xi \sim \mathcal{N}(0, \sigma^2), \qquad (1)$$

where $g : \mathbb{R} \to \mathbb{R}$ is a non-linear activation function, ξ is a Gaussian noise independent of x and the latent Bernoulli random variable $z_i \in \{0, 1\}$ indicates which expert has been chosen. In particular, only a single expert is active at any time, i.e. $\sum_{i=1}^{k} z_i = 1$, and their probabilities are modeled by a soft-max function:

$$\mathbb{P}\left[z_i = 1 | x\right] = \frac{e^{\langle w_i^*, x \rangle}}{\sum_{j=1}^{k} e^{\langle w_j^*, x \rangle}}.$$

Following the standard convention (Makkuva et al., 2019; Jacobs et al., 1991), we refer to the vectors a_i^* as *regressors*, the vectors w_i^* as either *classifiers* or *gating parameters*, and without loss of generality, we assume that $w_k^* = 0$.

Belying the canonical nature, and significant research effort, of the MoE model, the topic of learning MoE parameters is very poorly theoretically understood. In fact, the task of learning the parameters of a MoE, i.e. a_i^* and w_i^*, with provable guarantees is a long standing open problem for more than two decades (Sedghi et al., 2014). One of the key technical difficulties is that in a MoE, there is an inherent *coupling* between the regressors a_i^* and the gating parameters w_i^*, as can be seen from Eq. (1), which makes the problem challenging (Ho et al., 2019). In a recent work (Makkuva et al., 2019), the authors provided the first consistent algorithms for learning MoE parameters with theoretical guarantees. In order to tackle the aforementioned coupling issue, they proposed a clever scheme to first estimate the regressor parameters a_i^* and then estimating the gating parameters w_i^* using a combination of spectral methods and the EM algorithm. However, a major draw back is that this approach requires specially crafted algorithms for learning each of these two sets of parameters. In addition, they lack finite sample guarantees. Since SGD and its variants remain the de facto algorithms for training neural networks because of their practical advantages, and inspired by the successes of these gradient-descent based algorithms in finding global minima in a variety of non-convex problems, we ask the following question:

Question 2. How do we design objective functions amenable to efficient optimization techniques, such as SGD, with provable learning guarantees for MoE?

In this paper, we address this question in a principled manner and propose two non-trivial non-convex loss functions $L_4(\cdot)$ and $L_{\log}(\cdot)$ to learn the regressors and the gating parameters respectively. In particular, our loss functions possess nice landscape properties such as local minima being global and the global minima corresponding to the ground truth parameters. We also show that gradient descent on our losses can recover the true parameters with *global/random initializations*. To the best of our knowledge, ours is the first GD based approach with finite sample guarantees to learn the parameters of MoE. While our procedure to learn $\{a_i^*\}$ and $\{w_i^*\}$ separately and the technical assumptions are similar in spirit to Makkuva et al. (2019), our loss function based approach with provable guarantees for SGD is significantly different from that of Makkuva et al. (2019). We summarize our main contributions

below:

- **MoE as a building block for GRU:** We provide the first connection that the well-known GRU models are composed of basic building blocks, known as MoE. This link provides important insights into theoretical understanding of GRUs and further highlights the importance of MoE.
- **Optimization landscape design with desirable properties:** We design *two non-trivial* loss functions $L_4(\cdot)$ and $L_{\log}(\cdot)$ to learn the regressors and the gating parameters of a MoE separately. We show that our loss functions have nice landscape properties and are amenable to simple local-search algorithms. In particular, we show that SGD on our novel loss functions recovers the parameters with *global/random* initializations.
- **First sample complexity results:** We also provide the first sample complexity results for MoE. We show that our algorithms can recover the true parameters with accuracy ε and with high probability, when provided with samples *polynomial* in the dimension d and $1/\varepsilon$.

Related work. Linear dynamical systems can be thought of as the linear version of RNNs. There is a huge literature on the topic of learning these linear systems Alaeddini et al. (2018); Arora et al. (2018); Dean et al. (2017, 2018); Marecek and Tchrakian (2018); Oymak and Ozay (2018); Simchowitz et al. (2018); Hardt et al. (2018). However these works are very specific to the linear setting and do not extend to non-linear RNNs. Allen-Zhu et al. (2018) and Allen-Zhu and Li (2019) are two recent works to provide first theoretical guarantees for learning RNNs with ReLU activation function. However, it is unclear how these techniques generalize to the gated architectures. In this paper, we focus on the learnability of MoE, which are the building blocks for these gated models.

While there is a huge body of work on MoEs (see Yuksel et al. (2012); Masoudnia and Ebrahimpour (2014) for a detailed survey), the topic of learning MoE parameters is theoretically less understood with very few works on it. Jordan and Xu (1995) is one of the early works that showed the local convergence of EM. In a recent work, Makkuva et al. (2019) provided the first consistent algorithms for MoE in the population setting using a combination of spectral methods and EM algorithm. However, they do not provide any finite sample complexity bounds. In this work, we provide a unified approach using GD to learn the parameters with finite sample guarantees. To the best of our knowledge, we give the first gradient-descent based method with consistent learning guarantees, as well as the first finite-sample guarantee for any algorithm. The topic of designing the loss functions and analyzing their landscapes is a hot research topic in a wide variety of machine learning problems: neural networks (Hardt and Ma, 2017; Kawaguchi, 2016; Li and Yuan, 2017; Panigrahy et al., 2017; Zhong et al., 2017; Ge et al., 2018; Gao et al., 2019), matrix completion (Bhojanapalli et al., 2016), community detection (Bandeira et al., 2016), orthogonal tensor decomposition (Ge et al., 2015). In this work, we present the first objective function design and the landscape analysis for MoE.

Notation. We denote ℓ_2-Euclidean norm by $\|\cdot\|$. $[d] \triangleq \{1, 2, \ldots, d\}$. $\{e_i\}_{i=1}^d$ denotes the standard basis vectors in $\mathbb{R}^d$. We denote matrices by capital letters like A, W, etc. For any two vectors $x, y \in \mathbb{R}^d$, we denote their Hadamard product by $x \odot y$. $\sigma(\cdot)$ denotes the sigmoid function $\sigma(z) = 1/(1+e^{-z}), z \in \mathbb{R}$. For any $z = (z_1, \ldots, z_k) \in \mathbb{R}^k$, $\text{softmax}_i(z) = \exp(z_i)/(\sum_j \exp(z_j))$. $\mathcal{N}(mu, \Sigma)$ denotes the Gaussian distribution with mean $\mu \in \mathbb{R}^d$ and covariance $\Sigma \in \mathbb{R}^{d\times d}$. Through out the paper, we interchangeably denote regressors as $\{a_i\}$ or A, and gating parameters as $\{w_i\}$ or W.

Overview. The rest of the paper is organized as follows: In Section 2, we establish the precise mathematical connection between the well known GRU model and the MoE model. Building upon this correspondence, which highlights the importance of MoE, in Section 3 we design two novel loss functions to learn the respective regressors and gating parameters of a MoE and present our theoretical guarantees. In Section 4, we empirically validate that our proposed losses perform much better than the current approaches on a variety of settings.

2 GRU as a hierarchical MoE

In this section, we show that the recurrent update equations for GRU can be obtained from that of MoE, described in Eq. (1). In particular, we show that GRU can be viewed as a hierarchical MoE with depth-2. To see this, we restrict to the setting of a 2-MoE, i.e. let $k = 2$ and $(a_1^*, a_2^*) = (a_1, a_2)$, and $(w_1^*, w_2^*) = (w, 0)$ in Eq. (1). Then we obtain that

$$y = (1-z)\ g(a_1^\top x) + z\ g(a_2^\top x) + \xi, \tag{2}$$

where $z \in \{0, 1\}$ and $\mathbb{P}[z = 0|x] = \sigma(w^\top x)$. Since ξ is a zero mean random variable independent of x, taking conditional expectation on both sides of Eq. (2) yields that

$$\begin{aligned} y(x) &\triangleq \mathbb{E}[y|x] \\ &= \sigma(w^\top x) g(a_1^\top x) + (1 - \sigma(w^\top x)) g(a_2^\top x) \in \mathbb{R}. \end{aligned}$$

Now letting the output $y(x) \in \mathbb{R}^m$ to be a vector and allowing for different gating parameters $\{w_i\}$ and regressors $\{(a_{1i}, a_{2i})\}$ along each dimension $i = 1, \ldots, m$, we obtain

$$y(x) = (1 - z(x)) \odot g(A_1 x) + z(x) \odot g(A_2 x), \quad (3)$$

where $z(x) = (z_1(x), \ldots, z_m(x))^\top$ with $z_i(x) = \sigma(w_i^\top x)$, and $A_1, A_2 \in \mathbb{R}^{m \times d}$ denote the matrix of regressors corresponding to first and second experts respectively.

We now show that Eq. (3) is the basic equation behind the updates in GRU. Recall that in a GRU, given a time series $\{(x_t, y_t)\}_{t=1}^T$ of sequence length T, the goal is to produce a sequence of hidden states $\{h_t\}$ such that the output time series $\hat{y}_t = f(Ch_t)$ is close to $\{y_t\}$ in some well-defined loss metric, where f denotes the non-linear activation of the last layer. The equations governing the transition dynamics between $\{x_t\}$ and $\{h_t\}$ at any time $t \in [T]$ are given by (Cho et al., 2014):

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t,$$
$$\tilde{h}_t = g(U_h x_t + W_h(r_t \odot h_{t-1})),$$

where z_t and r_t denote the update and reset gates, which are given by

$$z_t = \sigma(U_z x_t + W_z h_{t-1}), \quad r_t = \sigma(U_r x_t + W_r h_{t-1}),$$

where the matrices U and W with appropriate subscripts are parameters to be learnt. While the gating activation function σ is modeled as sigmoid for the ease of obtaining gradients while training, their intended purpose was to operate as binary valued gates taking values in $\{0, 1\}$. Indeed, in a recent work Li et al. (2018), the authors show that binary valued gates enhance robustness with more interpretability and also give better performance compared to their continuous valued counterparts. In view of this, letting σ to be the binary threshold function $\mathbb{1}\{x \geq 0\}$, we obtain that

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot ((1 - r_t) \odot g(U_h x_t) + r_t \odot g(U_h x_t + W_h h_{t-1})). \quad (4)$$

Letting $x = (x_t, h_{t-1})$ and $y(x) = h_t$ in Eq. (3) with second expert $g(A_2 x)$ replaced by a 2-MoE, we can see from Eq. (4) that GRU is a depth-2 hierarchical MoE. This is also illustrated in Figure 2.

Note that in Figure 2, NN-1 models the mapping $(x_t, h_{t-1}) \mapsto h_{t-1}$, NN-2 represents $(x_t, h_{t-1}) \mapsto g(U_h x_t)$, and NN-3 models $(x_t, h_{t-1}) \mapsto g(U_h x_t + W_h h_{t-1})$. Hence, this is slightly different from the traditional MoE setting in Eq. (1) where the same activation $g(\cdot)$ is used for all the nodes. Nonetheless, we believe that studying this canonical model is a crucial first step which can shed important insights for a general setting.

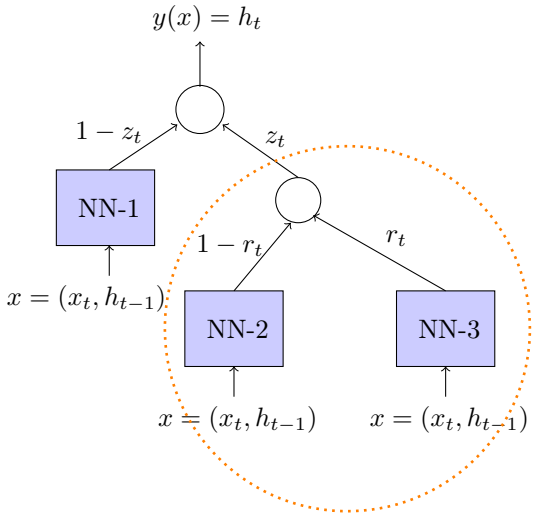


Figure 2: GRU as a hierarchical 2-MoE. The dotted circled portion indicates the canonical 2-MoE in Eq. (2). NN-1, NN-2 and NN-3 denote specific input-output mappings obtained from Eq. (4).

3 Optimization landscape design for MoE

In the previous section, we presented the mathematical connection between the GRU and the MoE. In this section, we focus on the learnability of the MoE model and design two novel loss functions for learning the regressors and the gating parameters separately.

3.1 Loss function for regressors: L_4

To motivate the need for loss function design in a MoE, first we take a moment to highlight the issues with the traditional approach of using the mean square loss ℓ_2. If (x, y) are generated according to the ground-truth MoE model in Eq. (1), $\ell_2(\cdot)$ computes the quadratic cost between the expected predictions $\hat{y}$ and the ground-truth y, i.e.

$$\ell_2(\{a_i\}, \{w_i\}) = \mathbb{E}_{(x,y)} \|\hat{y}(x) - y\|^2,$$

where $\hat{y}(x) = \sum_i \text{softmax}_i(w_1^\top x, \ldots, w_{k-1}^\top x, 0)\ g(a_i^\top x)$ is the predicted output, and $\{a_i\}, \{w_i\}$ denote the respective regressors and gating parameters. It is well-known that this mean square loss is prone to bad local minima as demonstrated empirically in the earliest work of Jacobs et al. (1991) (we verify this in Section 4 too), which also emphasized the importance of the right objective function to learn the parameters. Note that the bad landscape of ℓ_2 is not just unique to MoE, but also widely observed in the context of training neural network parameters (Livni et al., 2014). In the one-hidden-layer NN setting, some recent works (Ge et al.,

2018; Gao et al., 2019) addressed this issue by designing new loss functions with good landscape properties so that standard algorithms like SGD can provably learn the parameters. However these methods do not generalize to the MoE setting since they crucially rely on the fact that the coefficients z_i appearing in front of the activation terms $g(\langle a_i^*, x\rangle)$ in Eq. (1), which correspond to the linear layer weights in NN, are constant. Such an assumption does not hold in the context of MoEs because the gating probabilities depend on x in a parametric way through the softmax function and hence introducing the coupling between w_i^* and a_i^* (a similar observation was noted in Makkuva et al. (2019) in the context of spectral methods).

In order to address the aforementioned issues, inspired by the works of Ge et al. (2018) and Gao et al. (2019), we design a novel loss function $L_4(\cdot)$ to learn the regressors first. Our loss function depends on two distinct special transformations on both the input $x \in \mathbb{R}^d$ and the output $y \in \mathbb{R}$. For the output, we consider the following transformations:

$$\mathcal{Q}_4(y) \triangleq y^4 + \alpha y^3 + \beta y^2 + \gamma y, \quad \mathcal{Q}_2(y) \triangleq y^2 + \delta y, \quad (5)$$

where the set of coefficients $(\alpha, \beta, \gamma, \delta)$ are dependent on the choice of non-linearity g and noise variance σ^2. These are obtained by solving a simple linear system (see Appendix B). For the special case $g = \mathrm{Id}$, which corresponds to linear activations, the Quartic transform is $\mathcal{Q}_4(y) = y^4 - 6y^2(1+\sigma^2) + 3 + 3\sigma^4 - 6\sigma^2$ and the Quadratic transform is $\mathcal{Q}_2(y) = y^2 - (1+\sigma^2)$. For the input x, we assume that $x \sim \mathcal{N}(0, I_d)$, and for any two fixed $u, v \in \mathbb{R}^d$, we consider the projections of multivariate-Hermite polynomials Grad (1949); Holmquist (1996); Janzamin et al. (2014) along these two vectors, i.e.

$$\begin{aligned}
t_3(u,x) &= \frac{(u^\top x)^2 - \|u\|^2}{c'_{g,\sigma}}, \\
t_2(u,x) &= \frac{(u^\top x)^4 - 6\|u\|^2 (u^\top x)^2 + 3\|u\|^4}{c_{g,\sigma}}, \\
t_1(u,v,x) &= ((u^\top x)^2 (v^\top x)^2 - \|u\|^2 (v^\top x)^2 \\
&\quad - 4(u^\top x)(v^\top x)(u^\top v) - \|v\|^2 (u^\top x)^2 \\
&\quad + \|u\|^2 \|v\|^2 + 2(u^\top v)^2)/c_{g,\sigma},
\end{aligned}$$

where $c_{g,\sigma}$ and $c'_{g,\sigma}$ are two non-zero constants depending on g and σ. These transformations (t_1, t_2, t_3) on the input x and $(\mathcal{Q}_4, \mathcal{Q}_2)$ on the output y can be viewed as extractors of higher order information from the data. The utility of these transformations is concretized in Theorem 1 through the loss function defined below. Denoting the set of our regression parameters by the matrix $A^\top = [a_1|a_2|\dots|a_k] \in \mathbb{R}^{d\times k}$, we now define our objective function $L_4(A)$ as

$$\begin{aligned}
&L_4(A) \\
&\triangleq \sum_{\substack{i,j\in[k]\\ i\neq j}} \mathbb{E}[\mathcal{Q}_4(y) t_1(a_i, a_j, x)] - \mu \sum_{i\in[k]} \mathbb{E}[\mathcal{Q}_4(y) t_2(a_i, x)] \\
&+ \lambda \sum_{i\in[k]} (\mathbb{E}[\mathcal{Q}_2(y) t_3(a_i, x)] - 1)^2 + \frac{\delta}{2}\|A\|_F^2, \qquad (6)
\end{aligned}$$

where $\mu, \lambda, \delta > 0$ are some positive regularization constants. Notice that L_4 is defined as an expectation of terms involving the data transformations: $\mathcal{Q}_4, \mathcal{Q}_2, t_1, t_2$, and t_3. Hence its gradients can be readily computed from finite samples and is amenable to standard optimization methods such as SGD for learning the parameters. Moreover, the following theorem highlights that the landscape of L_4 does not have any spurious local minima.

Theorem 1 (Landscape analysis for learning regressors)**.** *Under the mild technical assumptions of Makkuva et al. (2019), the loss function L_4 does not have any spurious local minima. More concretely, let $\varepsilon > 0$ be a given error tolerance. Then we can choose the regularization constants μ, λ and the parameters ε, τ such that if A satisfies*

$$\|\nabla L_4(A)\|_2 \le \varepsilon, \quad \nabla^2 L_4(A) \succcurlyeq -\tau/2,$$

then $(A^\dagger)^\top = PD\Gamma A^* + E$, where D is a diagonal matrix with entries close to 1, Γ is a diagonal matrix with $\Gamma_{ii} = \sqrt{\mathbb{E}[p_i^*(x)]}$, P is a permutation matrix and $\|E\| \le \varepsilon_0$. Hence every approximate local minimum is ε-close to the global minimum.

Intuitions behind the theorem and the special transforms: While the transformations and the loss L_4 defined above may appear non-intuitive at first, the key observation is that L_4 can be viewed as a fourth-order polynomial loss in the parameter space, i.e.

$$\begin{aligned}
&L_4(A) \\
&= \sum_{m\in[k]} \mathbb{E}[p_m^*(x)] \sum_{\substack{i\neq j\\ i,j\in[k]}} \langle a_m^*, a_i\rangle^2 \langle a_m^*, a_j\rangle^2 \\
&- \mu \sum_{m,i\in[k]} \mathbb{E}[p_m^*(x)] \langle a_m^*, a_i\rangle^4 \qquad (7) \\
&+ \lambda \sum_{i\in[k]} \Big(\sum_{m\in[k]} \mathbb{E}[p_m^*(x)] \langle a_m^*, a_i\rangle^2 - 1\Big)^2 + \frac{\delta}{2}\|A\|_F^2,
\end{aligned}$$

where p_i^* refers to the softmax probability for the i^{th} label with true gating parameters, i.e. $p_i^*(x) = \mathrm{softmax}_i(\langle w_1^*, x\rangle, \dots, \langle w_{k-1}^*, x\rangle, 0)$. This alternate characterization of $L_4(\cdot)$ in Eq. (7) is the crucial step towards proving Theorem 1. Hence these specially designed transformations on the data (x, y) help us to

achieve this objective. Given this viewpoint, we utilize tools from Ge et al. (2018), where a similar loss involving fourth-order polynomials were analyzed in the context of 1-layer ReLU network, to prove the desired landscape properties for L_4. The full details behind the proof are provided in Appendix C. Moreover, in Section 4 we empirically verify that the technical assumptions are only needed for the theoretical results and that our algorithms are robust to these assumptions, and work equally well even when we relax them.

In the finite sample regime, we replace the population expectations in Eq. (6) with sample average to obtain the empirical loss $\hat{L}$. The following theorem establishes that $\hat{L}$ too inherits the same landscape properties of L when provided enough samples.

Theorem 2 (Finite sample landscape)**.** *There exists a polynomial* $\mathrm{poly}(d, 1/\varepsilon)$ *such that whenever* $n \geq \mathrm{poly}(d, 1/\varepsilon)$*,* $\hat{L}$ *inherits the same landscape properties as that of* L *established in Theorem 1 with high probability. Hence stochastic gradient descent on* $\hat{L}$ *converges to an approximate local minima which is also close to a global minimum in time polynomial in* $d, 1/\varepsilon$*.*

Remark 1. Notice that the parameters $\{a_i\}$ learnt through SGD are some permutation of the true parameters a_i^* upto sign flips. This sign ambiguity can be resolved using existing standard procedures such as Algorithm 1 in Ge et al. (2018). In the remainder of the paper, we assume that we know the regressors upto some error $\varepsilon_{\mathrm{reg}} > 0$ in the following sense: $\max_{i\in[k]} \|a_i - a_i^*\| = \sigma^2 \varepsilon_{\mathrm{reg}}$.

3.2 Loss function for gating parameters: $L_{\log}$

In the previous section, we have established that we can learn the regressors a_i^* upto small error using SGD on the loss function L_4. Now we are interested in answering the following question: Can we design a loss function amenable to efficient optimization algorithms such as SGD with recoverable guarantees to learn the gating parameters?

In order to gain some intuition towards addressing this question, consider the simplified setting of $\sigma = 0$ and $A = A^*$. In this setting, we can see from Eq. (1) that the output y equals one of the activation values $g(\langle a_i^*, x\rangle)$, for $i \in [k]$, with probability 1. Since we already have access to the true parameters, i.e. $A = A^*$, we can see that we can exactly recover the hidden latent variable z, which corresponds to the chosen hidden expert for each sample (x, y). Thus the problem of learning the *classifiers* $w_i^*, \ldots, w_{k-1}^*$ reduces to a multiclass classification problem with label z for each input x and hence can be efficiently solved by traditional methods such as logistic regression. It turns out that these observations can be formalized to deal with more general settings (where we only know the regressors approximately and the noise variance is not zero) and that the gradient descent on the log-likelihood loss achieves the same objective. Hence we use the negative log-likelihood function to learn the classifiers, i.e.

$$\begin{aligned} &L_{\log}(W, A) \\ &\triangleq -\mathbb{E}_{(x,y)}[\log P_{y|x}] \qquad (8) \\ &= -\mathbb{E} \log \left(\sum_{i\in[k]} \frac{e^{\langle w_i, x\rangle}}{\sum_{j\in[k]} e^{\langle w_j, x\rangle}} \cdot \mathcal{N}(y | g(\langle a_i, x\rangle), \sigma^2) \right), \end{aligned}$$

where $W^\top = [w_1 | w_2 | \ldots | w_{k-1}]$. Note that the objective Eq. (8) in not convex in the gating parameters W whenever $\sigma \neq 0$. We omit the input distribution P_x from the above negative log-likelihood since it does not depend on any of the parameters. We now define the domain of the gating parameters Ω as

$$W \in \Omega \triangleq \{W \in \mathbb{R}^{(k-1)\times d} : \|w_i\|_2 \leq R, \forall i \in [k-1]\},$$

for some fixed $R > 0$. Without loss of generality, we assume that $w_k = 0$. Since we know the regressors approximately from the previous stage, i.e. $A \approx A^*$, we run gradient descent only for the classifier parameters keeping the regressors fixed, i.e.

$$W_{t+1} = \Pi_\Omega(W_t - \alpha \nabla_W L_{\log}(W_t, A)),$$

where $\alpha > 0$ is a suitably chosen learning-rate, $\Pi_\Omega(W)$ denotes the projection operator which maps each row of its input matrix onto the ball of radius R, and $t > 0$ denotes the iteration step. In a more succinct way, we write

$$\begin{aligned} W_{t+1} &= G(W_t, A), \\ G(W, A) &\triangleq \Pi_\Omega(W - \alpha \nabla_W L_{\log}(W, A)). \end{aligned}$$

Note that $G(W, A)$ denotes the projected gradient descent operator on W for fixed A. In the finite sample regime, we define our loss $L_{\log}^{(n)}(W, A)$ as the finite sample counterpart of Eq. (8) by taking empirical expectations. Accordingly, we define the gradient operator $G_n(W, A)$ as

$$G_n(W, A) \triangleq \Pi_\Omega(W - \alpha \nabla_W L_{\log}^{(n)}(W, A)).$$

In this paper, we analyze a sample-splitting version of the gradient descent, where given the number of samples n and the iterations T, we first split the data into T subsets of size $\lfloor n/T \rfloor$, and perform iterations on fresh batch of samples, i.e. $W_{t+1} = G_{n/T}(W_t, A)$. We use the norm $\|W - W^*\| = \max_{i\in[k-1]} \|w_i - w_i^*\|_2$ for our theoretical results. The following theorem establishes the almost geometric convergence of the population-gradient iterates under some high SNR conditions. The following results are stated for $R = 1$ for simplicity and also hold for any general $R > 0$.

Theorem 3 (GD convergence for classifiers). *Assume that* $\max_{i\in[k]} \|a_i - a_i^*\|_2 = \sigma^2\varepsilon_{\text{reg}}$. *Then there exists two positive constants* α_0 *and* σ_0 *such that for any step size* $0 < \alpha \le \alpha_0$ *and noise variance* $\sigma^2 < \sigma_0^2$, *the population gradient descent iterates* $\{W\}_{t\ge 0}$ *converge almost geometrically to the true parameter* W^* *for any randomly initialized* $W_0 \in \Omega$, *i.e.*

$$\|W_t - W^*\| \le (\rho_\sigma)^t \|W_0 - W^*\| + \kappa\varepsilon_{\text{reg}} \sum_{\tau=0}^{t-1} (\rho_\sigma)^\tau,$$

where $(\rho_\sigma, \kappa) \in (0,1) \times (0,\infty)$ *are dimension-independent constants depending on* g, k *and* σ *such that* $\rho_\sigma = o_\sigma(1)$ *and* $\kappa = O_{k,\sigma}(1)$.

Proof. (Sketch) For simplicity, let $\varepsilon_{\text{reg}} = 0$. Then we can show that $G(W^*, A^*) = W^*$ since $\nabla_W L_{\log}(W = W^*, A^*) = 0$. Then we capitalize on the fact that $G(\cdot, A^*)$ is strongly convex with minimizer at $W = W^*$ to show the geometric convergence rate. The more general case of $\varepsilon_{\text{reg}} > 0$ is handled through perturbation analysis. □

We conclude our theoretical discussion on MoE by providing the following finite sample complexity guarantees for learning the classifiers using the gradient descent in the following theorem, which can be viewed as a finite sample version of Theorem 3.

Theorem 4 (Finite sample complexity and convergence rates for GD). *In addition to the assumptions of Theorem 3, assume that the sample size* n *is lower bounded as* $n \ge c_1 T d \log(\frac{T}{\delta})$. *Then the sample-gradient iterates* $\{W^t\}_{t=1}^T$ *based on* n/T *samples per iteration satisfy the bound*

$$\|W^t - W^*\| \le (\rho_\sigma)^t \|W_0 - W^*\| + \frac{1}{1-\rho_\sigma}\left(\kappa\varepsilon_{\text{reg}} + c_2\sqrt{\frac{dT\log(Tk/\delta)}{n}}\right)$$

with probability at least $1 - \delta$.

4 Experiments

In this section, we empirically validate the fact that running SGD on our novel loss functions L_4 and $L_{\log}$ achieves superior performance compared to the existing approaches. Moreover, we empirically show that our algorithms are robust to the technical assumptions made in Theorem 1 and that they achieve equally good results even when the assumptions are relaxed.

Data generation. For our experiments, we choose $d = 10$, $k \in \{2, 3\}$, $a_i^* = e_i$ for $i \in [k]$ and $w_i^* = e_{k+i}$ for $i \in [k-1]$, and $g = \text{Id}$. We generate the data $\{(x_i, y_i)_{i=1}^n\}$ according to Eq. (1) and using these ground-truth parameters. We chose $\sigma = 0.05$ for all of our experiments.

Error metric. If $A \in \mathbb{R}^{k\times d}$ denotes the matrix of regressors where each row is of norm 1, we use the error metric $\mathcal{E}_{\text{reg}}$ to gauge the closeness of A to the ground-truth A^*:

$$\mathcal{E}_{\text{reg}} \triangleq 1 - \max_{\pi\in S_k} \min_{i\in[k]} |\langle a_i, a^*_{\pi(i)}\rangle|,$$

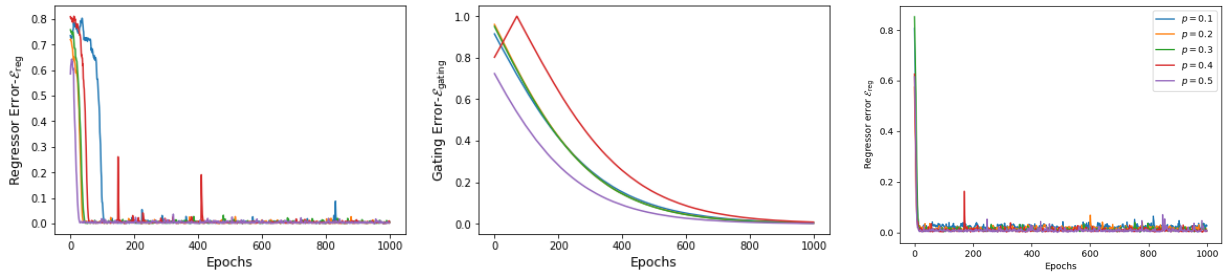
where S_k denotes the set of all permutations on $[k]$. Note that $\mathcal{E}_{\text{reg}} \le \varepsilon$ if and only if the learnt regressors have a minimum correlation of $1 - \varepsilon$ with the ground-truth parameters, upto a permutation. The error metric $\mathcal{E}_{\text{gating}}$ is defined similarly.

Results. In Figure 1, we choose $k = 3$ and compare the performance of our algorithm against existing approaches. In particular, we consider three methods: 1) EM algorithm, 2) SGD on the the classical ℓ_2-loss from Eq. (5), and 3) SGD on our losses L_4 and $L_{\log}$. For all the methods, we ran 5 independent trials and plotted the mean error. Figure 1a highlights the fact that minimizing our loss function L_4 by SGD recovers the ground-truth regressors, whereas SGD on ℓ_2-loss as well as EM get stuck in local optima. For learning the gating parameters W using our approach, we first fix the regressors A at the values learnt using L_4, i.e. $A = \hat{A}$, where $\hat{A}$ is the converged solution for L_4. For ℓ_2 and the EM algorithm, the gating parameters W are learnt jointly with regressors A. Figure 1b illustrates the phenomenon that our loss $L_{\log}$ for learning the gating parameters performs considerably better than the standard approaches, as indicated in significant gaps between the respective error values. Finally, in Figure 1c we plot the regressor error for L_4 over 5 random initializations. We can see that we recover the ground truth parameters in all the trials, thus empirically corroborating our technical results in Section 3.

4.1 Robustness to technical assumptions

In this section, we verify numerically the fact that our algorithms work equally well in the absence of technical assumptions made in Section 3.

Relaxing orthogonality in Theorem 1. A key assumption in proving Theorem 1, adapted from Makkuva et al. (2019), is that the set of regressors $\{a_i^*\}$ and set of gating parameters $\{w_i^*\}$ are orthogonal to each other. While this assumption is needed for the technical proofs, we now empirically verify that our conclusions still hold when we relax this. For this experiment, we choose $k = 2$ and let $(a_1^*, a_2^*) = (e_1, e_2)$. For the gating parameter $w^* \triangleq w_1^*$, we randomly generate it from uniform distribution on the d-dimensional unit sphere. In Figure 3a and Figure 3b, we plotted the individual parameter estimation error for 5 different runs



(a) (b) (c)

Figure 3: (a), (b): Robustness to parameter orthogonality: Plots show performance over 5 different trials for our losses L_4 and $L_{\log}$ respectively. (c) Robustness to Gaussianity of input: Performance over various mixing probabilities p.

for both of our losses L_4 and $L_{\log}$ for learning the regressors and the gating parameter respectively. We can see that our algorithms are still able to learn the true parameters even when the orthogonality assumption is relaxed.

Relaxing Gaussianity of the input. To demonstrate the robustness of our approach to the assumption that the input x is standard Gaussian, i.e. $x \sim \mathcal{N}(0, I_d)$, we generated x according to a mixture of two symmetric Gaussians each with identity covariances, i.e. $x \sim p\mathcal{N}(\mu, I_d) + (1-p)\mathcal{N}(-\mu, I_d)$, where $p \in [0, 1]$ is the mixing probability and $\mu \in \mathbb{R}^d$ is a fixed but randomly chosen vector. For various mixing proportions $p \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$, we ran SGD on our loss L_4 to learn the regressors. Figure 3c highlights that we learn these ground truth parameters in all the settings.

Finally we note that in all our experiments, the loss L_4 seems to require a larger batch size (1024) for its gradient estimation while running SGD. However, with smaller batch sizes such as 128 we are still able to achieve similar performance but with more variance. (see Appendix E).

5 Discussion

In this paper we established the first mathematical connection between two popular gated neural networks: GRU and MoE. Inspired by this connection and the success of SGD based algorithms in finding global minima in a variety of non-convex problems in deep learning, we provided the first gradient descent based approach for learning the parameters in a MoE. While the canoncial MoE does not involve any time series, extension of our methods for the recurrent setting is an important future direction. Similarly, extensions to deep MoE comprised of multiple gated as well as non-gated layers is also a fruitful direction of further research. We believe that the theme of using different loss functions for distinct parameters in NN models can potentially enlighten some new theoretical insights as well as practical methodologies for complex neural models.

Acknowledgements

We would like to thank the anonymous reviewers for their suggestions. This work is supported by NSF grants 1927712 and 1929955.

References

Alaeddini, A., Alemzadeh, S., Mesbahit, A., and Mesbahi, M. (2018). Linear model regression on time-series data: Non-asymptotic error bounds and applications. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 2259–2264. IEEE.

Allen-Zhu, Z. and Li, Y. (2019). Can sgd learn recurrent neural networks with provable generalization? *arXiv preprint arXiv:1902.01028*.

Allen-Zhu, Z., Li, Y., and Song, Z. (2018). On the convergence rate of training recurrent neural networks. *arXiv preprint arXiv:1810.12065*.

Arora, S., Hazan, E., Lee, H., Singh, K., Zhang, C., and Zhang, Y. (2018). Towards provable control for unknown linear dynamical systems.

Balakrishnan, S., Wainwright, M. J., and Yu, B. (2017). Statistical guarantees for the EM algorithm: From population to sample-based analysis. *The Annals of Statistics*, 45(1):77–120.

Bandeira, A. S., Boumal, N., and Voroninski, V. (2016). On the low-rank approach for semidefinite programs arising in synchronization and community detection. *arXiv preprint arXiv:1602.04426*.

Bhojanapalli, S., Neyshabur, B., and Srebro, N. (2016). Global optimality of local search for low rank matrix recovery. *arXiv preprint arXiv:1605.07221*.

Cho, K., van Merrienboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. abs/1409.1259.

Collobert, R., Bengio, S., and Bengio, Y. (2002). A parallel mixture of SVMs for very large scale problems. *Neural Computing*.

Dean, S., Mania, H., Matni, N., Recht, B., and Tu, S. (2017). On the sample complexity of the linear quadratic regulator. *arXiv preprint arXiv:1710.01688*.

Dean, S., Tu, S., Matni, N., and Recht, B. (2018). Safely learning to control the constrained linear quadratic regulator. *arXiv preprint arXiv:1809.10121*.

Eigen, D., Ranzato, M., and Sutskever, I. (2014). Learning factored representations in a deep mixture of experts. *arXiv preprint arXiv:1312.4314*.

Gao, W., Makkuva, A. V., Oh, S., and Viswanath, P. (2019). Learning one-hidden-layer neural networks under general input distributions. In Chaudhuri, K. and Sugiyama, M., editors, *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pages 1950–1959. PMLR.

Ge, R., Huang, F., Jin, C., and Yuan, Y. (2015). Escaping from saddle points — online stochastic gradient for tensor decomposition. In *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pages 797–842, Paris, France. PMLR.

Ge, R., Lee, J. D., and Ma, T. (2018). Learning one-hidden-layer neural networks with landscape design. *International Conference on Learning Representations*.

Grad, H. (1949). Note on n-dimensional hermite polynomials. *Communications on Pure and Applied Mathematics*, 2(4):325–330.

Graves, A. (2013). Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Graves, A., rahman Mohamed, A., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. *arXiv preprint arXiv:1303.5778*.

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*.

Gross, S., Ranzato, M., and Szlam, A. (2017). Hard mixtures of experts for large scale weakly supervised vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6865–6873.

Hardt, M. and Ma, T. (2017). Identity matters in deep learning. *arXiv preprint arXiv:1611.04231*.

Hardt, M., Ma, T., and Recht, B. (2018). Gradient descent learns linear dynamical systems. *The Journal of Machine Learning Research*, 19(1):1025–1068.

Ho, N., Yang, C.-Y., and Jordan, M. I. (2019). Convergence rates for gaussian mixtures of experts. *arXiv preprint arXiv:1907.04377*.

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

Holmquist, B. (1996). The d-variate vector hermite polynomial of order k. *Linear algebra and its applications*, 237:155–190.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation*.

Janzamin, M., Sedghi, H., and Anandkumar, A. (2014). Score function features for discriminative learning: Matrix and tensor framework. abs/1412.2863.

Jordan, M. I. and Jacobs, R. A. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Comput.*, 6(2):181–214.

Jordan, M. I. and Xu, L. (1995). Convergence results for the EM approach to mixtures of experts architectures. *Neural Networks*, 8(9):1409–1431.

Kawaguchi, K. (2016). Deep learning without poor local minima. *arXiv preprint arXiv:1605.07110*.

Ledoux, M. and Talagrand, M. (1991). *Probability in Banach Spaces: isoperimetry and processes*. Springer, Berlin.

Li, Y. and Yuan, Y. (2017). Convergence analysis of two-layer neural networks with relu activation. *arXiv preprint arXiv:1705.09886*.

Li, Z., He, D., Tian, F., Chen, W., Qin, T., Wang, L., and Liu, T.-Y. (2018). Towards binary-valued gates for robust lstm training. *arXiv preprint arXiv:1806.02988*.

Livni, R., Shalev-Shwartz, S., and Shamir, O. (2014). On the computational efficiency of training neural networks. In *Advances in neural information processing systems*, pages 855–863.

Makkuva, A. V., Oh, S., Kannan, S., and Viswanath, P. (2019). Breaking the gridlock in mixture-of-experts: Consistent and efficient algorithms. *International Conference on Machine Learning (ICML 2019), arXiv preprint arXiv:1802.07417*.

Marecek, J. and Tchrakian, T. (2018). Robust spectral filtering and anomaly detection. *arXiv preprint arXiv:1808.01181*.

Masoudnia, S. and Ebrahimpour, R. (2014). Mixture of experts: a literature survey. *Artificial Intelligence Review*, 42(2):275.

Ng, J. W. and Deisenroth, M. P. (2014). Hierarchical mixture-of-experts model for large-scale gaussian process regression. *arXiv preprint arXiv:1412.3078*.

Oymak, S. and Ozay, N. (2018). Non-asymptotic identification of lti systems from a single trajectory. *arXiv preprint arXiv:1806.05722*.

Panigrahy, R., Rahimi, A., Sachdeva, S., and Zhang, Q. (2017). Convergence results for neural networks via electrodynamics. *arXiv preprint arXiv:1702.00458*.

Rasmussen, C. E. and Ghahramani, Z. (2002). Infinite mixtures of gaussian process experts. In *Advances in neural information processing systems*, pages 881–888.

Sedghi, H., Janzamin, M., and Anandkumar, A. (2014). Provable tensor methods for learning mixtures of classifiers. *arXiv preprint arXiv:1412.3046*.

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., and Dean, J. (2017). Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

Simchowitz, M., Mania, H., Tu, S., Jordan, M. I., and Recht, B. (2018). Learning without mixing: Towards a sharp analysis of linear system identification. *arXiv preprint arXiv:1802.08334*.

Stein, C. (1972). A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. In *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability*, volume 2, pages 583–602. University of California Press.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14.

Tresp, V. (2001). Mixtures of gaussian processes. NIPS.

Vaart, A. W. and Wellner, J. A. (1996). *Weak convergence and empirical processes: with applications to statistics*. Springer.

Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2014). Show and tell: A neural image caption generator. *arXiv preprint arXiv:1411.4555*.

Yuksel, S. E., Wilson, J. N., and Gader, P. D. (2012). Twenty years of mixture of experts. *IEEE Transactions on Neural Networks and Learning Systems*, 23(8):1177–1193.

Zhong, K., Song, Z., Jain, P., Bartlett, P. L., and Dhillon, I. S. (2017). Recovery guarantees for one-hidden-layer neural networks. *arXiv preprint arXiv:1706.03175*.