Provable Gradient-Descent-Based Learning of Decision Lists by Transformers

Anonymous authors

I. Introduction

Despite the incredible success of Transformers [5], there is much that remains to be understood about the attention mechanism and its learning via gradient-based algorithms. The theoretical studies of gradient-based optimization in Transformers are rather limited and paint a far-from-complete picture [1], [3], [4].

In this work, we present convergence proofs for gradient flow and gradient descent on the parameters of a one-layer Transformer in the "beyond-NTK" regime [2], on a simple data distribution where the Transformer's attention plays a *key role* in solving the task. Specifically, we introduce and formalize a simple decision-list-like data distribution, serving as a theoretically simple, near-minimal example of a data distribution for which learning the attention head is crucial in solving the task. We study our simple Transformer's ability to learn this data distribution both empirically and theoretically, and we prove efficient gradient-descent-based learning under the population loss.

We first focus on just training the parameters of the Transformer *inside* the softmax-based-attention and prove that gradient descent on the population loss can efficiently learn these parameters. We then present a generalization of our data distribution and prove a corresponding *local* convergence result, in which the Transformer's value matrix is trained too, beginning from a "good enough" initialization. In this extended abstract, we briefly introduce our simple Transformer model, our data distribution, and our key results.

A. Simplified Transformer model

Let L denote the vocabulary size, n denote the sequence length, and d denote the embedding dimension. For a sequence of tokens of length n, let $X \in R^{d \times n}$ be the matrix formed by concatenating the embeddings of the n tokens. A *Transformer* model is composed of attention layers which have $d \times d$ parameter matrices W_Q, W_K, and W_V and computes new sequence embeddings via the function:

$$X \mapsto \sigma((W_V X)\sigma((W_K X)^\top (W_Q X))),$$

where for any matrix M, $\sigma(M)$ is the softmax function applied to the columns of M.

In the following, we consider a simplified Transformer model where the product $W_K^\top W_Q$ is replaced by a single $d \times d$ parameter matrix W, and for simplicity we use V to denote the matrix W_V:

$$f_{W,V}(X) := \sigma(VX\sigma(X^\top WX)), \tag{1}$$

Further, we assume that the embedding dimension d equals the vocabulary size L, and the input embeddings are simple one-hot embeddings, i.e. token i is embedded as the standard basis vector e_i, which is 0 in all coordinates except i, where it is 1. Thus, a sequence $(i_1, i_2, \ldots, i_n) \in [L]^n$ is represented by $X = [e_{i_1} \mid e_{i_2} \mid \ldots | e_{i_n}]$.

We assume also that the output vocabulary size is L. For a given input sequence represented by X, let $Y = (y_1, y_2, \ldots, y_n) \in [L]^n$ be the label sequence. The loss of the Transformer model $f_{W,V}$ on a single sequence (X, Y) is now defined as

$$\mathcal{L}((W,V);(X,Y)) = \frac{1}{n}\sum_{j=1}^{n} -\log(f_{W,V}(X)_{y_j,j}). \tag{2}$$

We analyze the dynamics of gradient flow and gradient descent on the above loss function on the sequence-to-sequence mapping described in the next section. We let $\mathcal{L}_t$ denote the population loss at time t, i.e. $\mathcal{L}_t := \mathcal{L}(W_t, V_t)$, where (W_t, V_t) are the parameters at time t.

B. Sequence-to-sequence mapping

We consider an arbitrary *permutation function* $\pi : [L] \to [L]$, and we let P denote the $L \times L$ permutation *matrix* associated with the permutation *function* π. The sequence-to-sequence mapping of interest is computed as follows. Let M denote a positive integer assumed to be much less than L. For every token $a \in [L]$, we associate a list of tokens $L_a = (a_1, a_2, \ldots, a_m)$ of length $m \leq M$, such that $a_m = a$. Given a sequence $s \in [L]^n$, the output label y for any token $a \in s$ is equal to $\pi(a_j)$, where j is the smallest index in L_a such that $a_j \in s$. We call a_j the *label-determining token* for a in s, since it *determines* the label: $\pi(a_j)$. Note that since $a_m = a \in s$, j always exists. Such a mapping can be realized by a *decision list* associated with each token a.

C. Convergence analysis

Our main results are as follows[1]:

1) Training only W: Consider $c = \frac{4n\log(L)}{\delta}$. Then gradient flow on the population loss started from $W_0 = 0$ and $V = cP$ finds a matrix W achieving perfect classification within time $\frac{4M}{\log(L)\gamma}$.

2) Training W and V: Consider $c = \frac{10n^2\mathcal{L}_0}{\delta} + 4\epsilon n$ and some $\epsilon > 0$ such that $\|V_0 - cP\|_\infty \leq \epsilon/2$. Suppose L and n are sufficiently large such that $\mathcal{L}_0 - \frac{6\epsilon}{n} > A$ for some $A > 0$ satisfying $A^2 > \frac{4m\mathcal{L}_0}{\epsilon}$. Then, starting from $W_0 = 0$ and V_0, with λ_{max} denoting the maximum eigenvalue of the Hessian of $\mathcal{L}(W,V)$, running gradient descent on the population loss with learning rate $\eta \leq \frac{1}{\lambda_{max}}$ for $T \in \left[\frac{2m\mathcal{L}_0}{A^2\eta}, \frac{\epsilon}{2\eta}\right]$ steps encounters a $t \in \{1, \ldots, T\}$ such that W_t, V_t achieve perfect classification on all sequences.

II. Conclusion

We have presented convergence results for a one-layer Transformer on a simple decision-list-inspired data distribution, which crucially requires the Transformer's token-to-token attention in order to solve the task. Although our local convergence requires the value matrix to be initialized within a "good enough" region in parameter space, we supplement our theorems with empirical results indicating that uniform attention (corresponding to $W = 0$) is sufficient to push V in the direction of P. Therefore, there is perhaps a tantalizing opportunity in future work to extend these results into a full convergence result beginning from small, random initialization (which is notoriously difficult to prove in such a non-overparameterized setting).

[1] The parameters $\delta, \gamma > 0$ depend on the minimum probability mass that the data distribution assigns to certain quantities.

References

[1] Alberto Bietti, Vivien Cabannes, Diane Bouchacourt, Hervé Jegou, and Léon Bottou. Birth of a Transformer: A Memory Viewpoint. 2023.

[2] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. Advances in Neural Information Processing Systems, 2018.

[3] Yuchen Li, Yuanzhi Li, and Andrej Risteski. How Do Transformers Learn Topic Structure: Towards a Mechanistic Understanding. International Conference on Machine Learning, 2023.

[4] Yuandong Tian, Yiping Wang, Beidi Chen, and Simon Du. Scan and Snap: Understanding Training Dynamics and Token Composition in 1-layer Transformer. 2023.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All you Need. Advances in Neural Information Processing Systems, 2017.