THE CLOSENESS OF IN-CONTEXT LEARNING AND WEIGHT SHIFTING FOR SOFTMAX REGRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) are known for their exceptional performance in natural language processing, making them highly effective in many human life-related tasks. The attention mechanism in the Transformer architecture is a critical component of LLMs, as it allows the model to selectively focus on specific input parts. The softmax unit, which is a key part of the attention mechanism, normalizes the attention scores. Hence, the performance of LLMs in various NLP tasks depends significantly on the crucial role played by the attention mechanism with the softmax unit.

In-context learning is one of the celebrated abilities of recent LLMs. Without further parameter updates, Transformers can learn to predict based on few in-context examples. However, the reason why Transformers becomes in-context learners is not well understood. Recently, in-context learning has been studied from a mathematical perspective with simplified linear self-attention without softmax unit. Based on a linear regression formulation $\min_x \|Ax - b\|_2$, existing works show linear Transformers' capability of learning linear functions in context. The capability of Transformers with softmax unit approaching full Transformers, however, remains unexplored.

In this work, we study the in-context learning based on a softmax regression formulation $\min_x \|\langle \exp(Ax), \mathbf{1}_n \rangle^{-1} \exp(Ax) - b\|_2$. We show the upper bounds of the data transformations induced by a single self-attention layer with softmax unit and by gradient-descent on a ℓ_2 regression loss for softmax prediction function. Our theoretical results imply that when training self-attention-only Transformers for fundamental regression tasks, the models learned by gradient-descent and Transformers show great similarity.

1 INTRODUCTION

In recent years, there has been a significant increase in research and development in the field of Artificial Intelligence, with large language models (LLMs) emerging as an effective way to tackle complex tasks. Transformers have achieved state-of-the-art results in various NLP tasks, such as machine translation (Prato et al., 2019; Gao et al., 2020) and text generation (Luo et al., 2022). As a result, they have become the preferred architecture for NLP, where BERT (Devlin et al., 2018), GPT-3 (Brown et al., 2020), PaLM (Chowdhery et al., 2022) were proposed. They have demonstrated remarkable learning and reasoning capabilities and have proven to be more efficient than traditional models when processing natural language.

Additionally, LLMs can be fine-tuned for multiple purposes without requiring a new build from scratch, making them a versatile tool for AI applications. Moreover, recent studies on the in-context learning abilities of LLMs have demonstrated that even without further fine-tuning, LLMs can generalize to new tasks with only a few demonstration examples in the prompt. To understand how LLMs become in-context learners, recent works have studied the problem and provided mathematical explanations from the Transformer architecture perspective, showing a simplified linear self-attention layer of Transformer can learn linear functions similarly as a step of gradient descent (Oswald et al., 2022; Akyürek et al., 2022; Garg et al., 2022). While such linear approximation of full Transformers is overly simplistic, studies on more complex Transformer architecture are needed to further explain the in-context learning phenomenon.

Transformers have a specific type of sequence-to-sequence neural network architecture. They utilize the attention mechanism (Vaswani et al., 2017; Radford et al., 2018; Devlin et al., 2018; Brown et al., 2020) that allows them to capture long-range dependencies and context from input data effectively. The core of the attention mechanism is the attention matrix which is comprised of rows and columns, corresponding to individual words or "tokens". The attention matrix represents the relationships within the given text. It measures the importance of each token in a sequence as it relates to the desired output. During the training process, the attention matrix is learned and optimized to improve the accuracy of the model's predictions. Through the attention mechanism, each input token is evaluated based on its relevance to the desired output by assigning a token score. This score is determined by a similarity function that compares the current output state with input states.

Theoretically, the attention matrix is comprised of the query matrix $Q \in \mathbb{R}^{n \times d}$, the key matrix $K \in \mathbb{R}^{n \times d}$ and the value matrix $V \in \mathbb{R}^{n \times d}$. Following Zandieh et al. (2023); Alman & Song (2023); Brand et al. (2023), the computation of the normalized attention function is defined as $D^{-1} \exp(QK^\top)V$. Following the transformer literature, we apply $\exp$ to a matrix entry-wise way. Here $D \in \mathbb{R}^{n \times n}$ is diagonal matrix that defined as $D = \mathrm{diag}(\exp(QK^\top)\mathbf{1}_n)$. Intuitively, D denotes the softmax normalization matrix. A more general computation formulation can be written as

$$\underbrace{D^{-1}}_{n\times n \text{ diagonal matrix}} \underbrace{\exp(XQK^\top X^\top)}_{n\times n} \underbrace{X}_{n\times d} \underbrace{V}_{d\times d}, \quad D := \mathrm{diag}(\exp(XQK^\top X^\top)\mathbf{1}_n)$$

In the above setting, we treat $Q, K, V \in \mathbb{R}^{d \times d}$ as weights and X is the input sentence data that has length n and each word embedding size is d. In the remaining of the part, we will switch X to notation A and use A to denote sentence. Mathematically, the attention computation problem can be formulated as a regression problem in the following sense

Definition 1.1. *We consider the following problem*

$$\min_{X \in \mathbb{R}^{d \times d}} \|D^{-1} \exp(AXA^\top) - B\|_F$$

where $A \in \mathbb{R}^{n \times d}$ can be treated as a length-n document and each word has length-d embedding size. Here $D = \mathrm{diag}(AXA^\top \mathbf{1}_n)$. For any given $A \in \mathbb{R}^{n \times d}$ and $B \in \mathbb{R}^{n \times n}$, the goal is to find some weight X to optimize the above objective function.

In contrast to the formulation in Zandieh et al. (2023); Alman & Song (2023); Brand et al. (2023), the parameter X in Definition 1.1 is equivalent to the $QK^\top \in \mathbb{R}^{d \times d}$ in the generalized version of Zandieh et al. (2023); Alman & Song (2023); Brand et al. (2023) (e.g. replacing $Q \in \mathbb{R}^{n \times d}$ by XQ where $X \in \mathbb{R}^{n \times d}$ and $Q \in \mathbb{R}^{d \times d}$. Similarly for K and V. In such scenario, X can be viewed as a matrix representation of a length-n sentence.).

A number of work (Akyürek et al., 2022; Garg et al., 2022; Oswald et al., 2022) study the in-context learning from mathematical perspective in a much simplified setting than Definition 1.1, which is linear regression formulation as in Definition 1.2. They show linear Transformer without softmax unit in its attention layer can mimic the ability of gradient descent in learning linear functions in context. While the softmax unit plays an important role in attention computations of full Transformers, their simplification of the softmax unit leaves a gap in explaining LLMs' in-context learning abilities.

Definition 1.2. *Given a matrix $A \in \mathbb{R}^{n \times d}$ and $b \in \mathbb{R}^n$, the goal is to solve*

$$\min_x \|Ax - b\|_2$$

Several theoretical transformer work have studied either exponential regression (Gao et al., 2023; Li et al., 2023) or softmax regression problem (Deng et al., 2023b). In this work, to take a step forward to understand the softmax unit in the attention scheme in LLMs. We consider the following softmax regression and study the in-context learning phenomena and its closeness to gradient descent from the data transformation perspective.

Definition 1.3 (Softmax Regression)**.** *Given a $A \in \mathbb{R}^{n \times d}$ and a vector $b \in \mathbb{R}^n$, the goal is to solve*

$$\min_{x \in \mathbb{R}^d} \|\langle \exp(Ax), \mathbf{1}_n \rangle^{-1} \exp(Ax) - b\|_2$$

We remark that the Definition 1.3 of Softmax Regression is a formulation in between Definition 1.2 and Definition 1.1.

We state our major result as follows:

Theorem 1.4 (Bounded shift for in-context learning, informal version of the combination of Theorem 5.1 and Theorem 5.2)**.** *If the following conditions hold*

- *Let* $A \in \mathbb{R}^{n \times d}$.
- *Let* $b \in \mathbb{R}^n$.
- $\|A\| \leq R$.
- *Let* $\|x\|_2 \leq R$.
- $\|A(x_{t+1} - x_t)\|_\infty < 0.01$.
- $\|(A_{t+1} - A_t)x\|_\infty < 0.01$.
- *Let* $R \geq 4$.
- *Let* $M := n^{1.5} \exp(10R^2)$.

We consider the softmax regression (Definition 1.3) problem

$$\min_x \|\langle \exp(Ax), \mathbf{1}_n \rangle^{-1} \exp(Ax) - b\|_2.$$

- **Part 1.** *If we move the x_t to x_{t+1}, then we're solving a new softmax regression with*
  $$\min_x \|\langle \exp(Ax), \mathbf{1}_n \rangle^{-1} \exp(Ax) - \widetilde{b}\|_2$$
  where
  $$\|\widetilde{b} - b\|_2 \leq M \cdot \|x_{t+1} - x_t\|_2$$
- **Part 2.** *If we move the A_t to A_{t+1}, then we're solving a new softmax regression with*
  $$\min_x \|\langle \exp(Ax), \mathbf{1}_n \rangle^{-1} \exp(Ax) - \widehat{b}\|_2$$
  where
  $$\|\widehat{b} - b\|_2 \leq M \cdot \|A_{t+1} - A_t\|$$

Recall that $A \in \mathbb{R}^{n \times d}$ denotes a length-n document and each word has the length-d embedding size and x denotes the simplified version of $QK^\top$. One-step gradient descent can be treated as an update to the model's weight x. Thus, part 1 of our result (Theorem 1.4) implies that the data transformation of b induced by gradient-descent on the ℓ_2 regression loss is bounded by $M \cdot \|x_{t+1} - x_t\|_2$.

According to Oswald et al. (2022), to do in-context learning, a self-attention layer update can be treated as an update to the tokenized document A. For detailed derivation, please refer to Oswald et al. (2022). Thus, part 2 of our result (Theorem 1.4) implies that the data transformation of b induced by a single self-attention layer is bounded by $M \cdot \|A_{t+1} - A_t\|$.

We remark that the data transformation of b induced by 1) a single self-attention layer and by 2) gradient-descent on the ℓ_2 regression loss are both bounded. The bounded transformation of b implies that when training self-attention-only Transformers for fundamental regression tasks, the models learned by gradient-descent and Transformers show great similarity.

Roadmap. In Section 2, we introduce some related works. In Section 3, we give some preliminaries. In Section 4, we compute the gradient of the loss function with softmax function with respect to x. Those functions include $\alpha(x)^{-1}$, $\alpha(x)$ and $f(x)$. In Section 5, we give our formal theoretical results, validated by preliminary experiments present in Appendix D. In Section 6, we conclude our paper.

2 Related Work

2.1 In-context Learning

Akyürek et al. (2022) indicate that Transformer-based in-context learners are able to perform traditional learning algorithms implicitly. This is achieved by encoding smaller models within their internal activations. These smaller models are updated by the given context. They theoretically investigate the learning algorithms that Transformer decoders can implement. They demonstrate that Transformers need only a limited number of layers and hidden units to implement various linear regression algorithms. For d-dimensional regression problems, a $O(d)$-hidden-size Transformer can perform a single step of gradient descent. They also demonstrate its ability to update a ridge regression problem. The study reveals that Transformers theoretically have the ability to perform multiple linear regression algorithms.

Garg et al. (2022) concentrate on training Transformer to learn certain functions, under in-context conditions. The goal is to have a more comprehensive understanding of in-context learning and determine if Transformers can learn the majority of functions within a given class after training. They found that in-context learning is possible even when there is a distribution shift between training and inference data or between in-context examples and query inputs. In addition, they find out that Transformers can learn more complex function classes such as sparse linear functions, two-layer neural networks, and decision trees. These trained Transformers have comparable performance to task-specific learning algorithms.

Oswald et al. (2022) demonstrate the similarity between the training process of Transformers in in-context tasks and some meta-learning formulations based on gradient descent. During training Transformers for auto-regressive tasks, the implementation of in-context learning in the Transformer forward pass is carried out through gradient-based optimization of an implicit auto-regressive inner loss that is constructed from the in-context data.

Formally speaking, they consider the following problem $\min_x \|Ax - b\|_2$ defined in Definition 1.2. They show that one step of gradient descent carries out data transformation as follows:

$$\begin{aligned}\|A(x + \delta_x) - b\|_2 &= \|Ax - (b - \delta_b)\|_2 \\ &= \|Ax - \widetilde{b}\|_2\end{aligned}$$

where δ_x denotes the one-step gradient descent on x and δ_b denotes the corresponding data transformation on b. They also show that a self-attention layer is in principle capable of exploiting statistics in the current training data samples. Concretely, let $Q, K, V \in \mathbb{R}^{d \times d}$ denotes the weights for the query matrix, key matrix, and value matrix respectively. The linear self-attention layer updates an input sample by doing the following data transformation:

$$\widehat{b}_j = b_j + PVK^\top Q_j$$

where $\widehat{b}$ denotes the updated b and P denotes the projection matrix such that a Transformer step $\widehat{b}_j$ on every j is identical to the gradient-induced dynamics $\widetilde{b}_j$. This equivalence implies that when training linear-self-attention-only Transformers for fundamental regression tasks, the models learned by GD and Transformers show great similarity.

Xie et al. (2021) explores the occurrence of in-context learning during pre-training when documents exhibit long-range coherence. The Language Model (LLM) develops the ability to generate coherent next tokens by deducing a latent document-level concept. During testing, in-context learning is observed when the LLM deduces a shared latent concept between examples in a prompt. They demonstrate that in-context learning happens even when there is a distribution mismatch between prompts and pretraining data, especially when the pretraining distribution is a mixture of Hidden Markov Models (Baum & Petrie, 1966). Theoretically, they show that the error of the in-context predictor is optimal when a distinguishability condition holds. In cases where this condition does not hold, the expected error still reduces as the length of each example increases. This finding highlights the importance of both input and input-output mapping for in-context learning.

2.2 Transformer Theory

The advancements of Transformers have been noteworthy, however, their learning mechanisms are not completely comprehensible yet. Although these models have performed remarkably well in structured and reasoning activities, our comprehension of their mathematical foundations lags significantly behind. Past research has indicated that the outstanding performance of Transformer-based models can be attributed to the information within their components, such as multi-head attention. Various studies (Tenney et al., 2019; Vig & Belinkov, 2019; Hewitt & Liang, 2019; Belinkov, 2022) have presented empirical proof that these components carry a substantial amount of information, which can help resolve different probing tasks.

Recent research has investigated the potential of Transformers through both theoretical and experimental methods, including Turing completeness (Bhattamishra et al., 2020b), function approximation (Yun et al., 2020; Chen et al., 2021), formal language representation (Bhattamishra et al., 2020a; Ebrahimi et al., 2020; Yao et al., 2021), and abstract algebraic operation learning (Zhang et al., 2022). Some of these studies have indicated that Transformers may act as universal approximators for sequence-to-sequence operations and emulate Turing machines (Pérez et al., 2019; Bhattamishra et al., 2020b). Liu et al. (2023) demonstrate the existence of contextual sparsity in LLM, which can be accurately predicted. They exploit the sparsity to speed up LLM inference without degrading the performance from both a theoretical perspective and an empirical perspective. Dao et al. (2021) proposed the Pixelated Butterfly model that uses a simple fixed sparsity pattern to speed up the training of Transformer. Other studies have focused on the expressiveness of attention within Transformers (Dehghani et al., 2018; Vuckovic et al., 2020; Zhang et al., 2020; Edelman et al., 2021; Snell et al., 2021; Wei et al., 2021).

Furthermore, Zhao et al. (2023) has demonstrated that moderately sized masked language models may effectively parse and recognize syntactic information that helps in the partial reconstruction of a parse tree. Inspired by the language grammar model studied by Zhao et al. (2023), Deng et al. (2023a) consider the tensor cycle rank approximation problem. Gao et al. (2023) consider the exponential regression in neural tangent kernel over-parameterization setting. Li et al. (2023) studied the computation of regularized version of the exponential regression problem but they ignore the normalization factor. Deng et al. (2023b) consider the softmax regression which considers the normalization factor compared to exponential regression problems Gao et al. (2023); Li et al. (2023). The majority of LLMs can perform attention computations in an approximate manner during inference, as long as there are sufficient guarantees of precision. This perspective has been studied by various research, including Child et al. (2019); Kitaev et al. (2020); Wang et al. (2020); Daras et al. (2020); Katharopoulos et al. (2020); Chen et al. (2021; 2022). With this in mind, Zandieh et al. (2023); Alman & Song (2023); Brand et al. (2023); Deng et al. (2023c) have studied the attention matrix computation from the hardness perspective and developed faster algorithms.

3 Preliminary

In Section 3.1, we introduce the notations used in this paper. In Section 3.2, we give some facts about the basic algebra. In Section 3.3, we propose the lower bound on $\langle \exp(Ax), \mathbf{1}_n \rangle$.

3.1 Notations

For a positive integer n, we use $[n]$ to denote $\{1, 2, \cdots, n\}$, for any positive integer n. We use $\mathbb{E}[\cdot]$ to denote expectation. We use $\Pr[\cdot]$ to denote probability. We use $\mathbf{1}_n$ to denote the vector where all entries are one. We use $\mathbf{0}_0$ to denote the vector where all entries are zero. The identity matrix of size $n \times n$ is represented by I_n for a positive integer n. The symbol $\mathbb{R}$ refers to real numbers and $\mathbb{R}_{\geq 0}$ represents non-negative real numbers. For any vector $x \in \mathbb{R}^n$, $\exp(x) \in \mathbb{R}^n$ denotes a vector where the i-th entry is $\exp(x_i)$ and $\|x\|_2$ represents its ℓ_2 norm, that is, $\|x\|_2 := (\sum_{i=1}^n x_i^2)^{1/2}$. We use $\|x\|_\infty$ to denote $\max_{i \in [n]} |x_i|$. For any vector $x \in \mathbb{R}^n$ and vector $y \in \mathbb{R}^d$, we use $\langle x, y \rangle$ to denote the inner product of vector x and y. The notation B_i is used to indicate the i-th row of matrix B. If a and b are two column vectors in $\mathbb{R}^n$, then $a \circ b$ denotes a column vector where $(a \circ b)_i = a_i b_i$. For a square and full rank matrix B, we use B^{-1} to denote the true inverse of B.

3.2 Basic Algebras

Fact 3.1. *For vectors $x, y \in \mathbb{R}^n$, we have*

- $\|x \circ y\|_2 \leq \|x\|_\infty \cdot \|y\|_2$
- $\|x\|_\infty \leq \|x\|_2 \leq \sqrt{n}\|x\|_\infty$
- $\|\exp(x)\|_\infty \leq \exp(\|x\|_2)$

- *For any* $\|x-y\|_\infty \leq 0.01$*, we have* $\|\exp(x)-\exp(y)\|_2 \leq \|\exp(x)\|_2 \cdot 2\|x-y\|_\infty$

Fact 3.2. *For matrices* X, Y*, we have*

- $\|X^\top\| = \|X\|$
- $\|X\| \geq \|Y\| - \|X-Y\|$
- $\|X+Y\| \leq \|X\| + \|Y\|$
- $\|X \cdot Y\| \leq \|X\| \cdot \|Y\|$
- *If* $X \preceq \alpha \cdot Y$*, then* $\|X\| \leq \alpha \cdot \|Y\|$

3.3 Lower bound on β

Lemma 3.3. *If the following conditions holds*

- $\|A\| \leq R$
- $\|x\|_2 \leq R$
- *Let* β *be lower bound on* $\langle \exp(Ax), \mathbf{1}_n \rangle$

Then we have

$$\beta \geq \exp(-R^2)$$

Proof. We have

$$\begin{aligned}
\langle \exp(Ax), \mathbf{1}_n \rangle = & ~ \sum_{i=1}^n \exp((Ax)_i) \\
\geq & ~ \min_{i \in [n]} \exp((Ax)_i) \\
\geq & ~ \min_{i \in [n]} \exp(-|(Ax)_i|) \\
= & ~ \exp(-\max_{i \in [n]} |(Ax)_i|) \\
= & ~ \exp(-\|Ax\|_\infty) \\
\geq & ~ \exp(-\|Ax\|_2) \\
\geq & ~ \exp(-R^2)
\end{aligned}$$

the 1st step follows from simple algebra, the 2nd step comes from simple algebra, the 3rd step follows from the fact that $\exp(x) \geq \exp(-|x|)$, the 4th step follows from the fact that $\exp(-x)$ is monotonically decreasing, the 5th step comes from definition of ℓ_∞ norm, the 6th step follows from Fact 3.1, the 7th step follows from the assumption on A and x. □

4 Softmax Function with respect to x

In Section 4.1, we give the definitions used in the computation. In Section 4.2, we compute the gradient of the loss function with softmax function with respect to x. Those functions includes $\alpha(x)^{-1}$, $\alpha(x)$ and $f(x)$.

4.1 Definitions

We define function softmax f as follows

Definition 4.1 (Function f, Definition 5.1 in Deng et al. (2023b))**.** *Given a matrix* $A \in \mathbb{R}^{n \times d}$*. Let* $\mathbf{1}_n$ *denote a length-*n *vector that all entries are ones. We define prediction function* $f : \mathbb{R}^d \to \mathbb{R}^n$ *as follows*

$$f(x) := \langle \exp(Ax), \mathbf{1}_n \rangle^{-1} \cdot \exp(Ax).$$

Definition 4.2 (Loss function $L_{\exp}$, Definition 5.3 in Deng et al. (2023b))**.** *Given a matrix $A \in \mathbb{R}^{n \times d}$ and a vector $b \in \mathbb{R}^n$. We define loss function $L_{\exp} : \mathbb{R}^d \to \mathbb{R}$ as follows*

$$L_{\exp}(x) := 0.5 \cdot \| \langle \exp(Ax), \mathbf{1}_n \rangle^{-1} \exp(Ax) - b \|_2^2.$$

For convenient, we define two helpful notations α and c

Definition 4.3 (Normalized coefficients, Definition 5.4 in Deng et al. (2023b))**.** *We define $\alpha : \mathbb{R}^d \to \mathbb{R}$ as follows*

$$\alpha(x) := \langle \exp(Ax), \mathbf{1}_n \rangle.$$

Then, we can rewrite $f(x)$ (see Definition 4.1) and $L_{\exp}(x)$ (see Definition 4.2) as follows

- $f(x) = \alpha(x)^{-1} \cdot \exp(Ax)$.
- $L_{\exp}(x) = 0.5 \cdot \| \alpha(x)^{-1} \cdot \exp(Ax) - b \|_2^2$.
- $L_{\exp}(x) = 0.5 \cdot \| f(x) - b \|_2^2$.

Definition 4.4 (Definition 5.5 in Deng et al. (2023b))**.** *We define function $c : \mathbb{R}^d \in \mathbb{R}^n$ as follows*

$$c(x) := f(x) - b.$$

Then we can rewrite $L_{\exp}(x)$ (see Definition 4.2) as follows

- $L_{\exp}(x) = 0.5 \cdot \| c(x) \|_2^2$.

4.2 Gradient Computations

We state a lemma from previous work,

Lemma 4.5 (Gradient, Lemma 5.6 in Deng et al. (2023b))**.** *If the following conditions hold*

- *Given matrix $A \in \mathbb{R}^{n \times d}$ and a vector $b \in \mathbb{R}^n$.*
- *Let $\alpha(x)$ be defined in Definition 4.3.*
- *Let $f(x)$ be defined in Definition 4.1.*
- *Let $c(x)$ be defined in Definition 4.4.*
- *Let $L_{\exp}(x)$ be defined in Definition 4.2.*

For each $i \in [d]$, we have

- *Part 1.*
$$\frac{\mathrm{d} \exp(Ax)}{\mathrm{d} x_i} = \exp(Ax) \circ A_{*,i}$$
- *Part 2.*
$$\frac{\mathrm{d} \langle \exp(Ax), \mathbf{1}_n \rangle}{\mathrm{d} x_i} = \langle \exp(Ax), A_{*,i} \rangle$$
- *Part 3.*
$$\frac{\mathrm{d} \alpha(x)^{-1}}{\mathrm{d} x_i} = -\alpha(x)^{-1} \cdot \langle f(x), A_{*,i} \rangle$$
- *Part 4.*
$$\frac{\mathrm{d} f(x)}{\mathrm{d} x_i} = \frac{\mathrm{d} c(x)}{\mathrm{d} x_i} = -\langle f(x), A_{*,i} \rangle \cdot f(x) + f(x) \circ A_{*,i}$$
- *Part 5.*
$$\frac{\mathrm{d} L_{\exp}(x)}{\mathrm{d} x_i} = \underbrace{A_{*,i}^\top}_{1 \times n} \cdot \Big(\underbrace{f(x)}_{n \times 1} \underbrace{\langle c(x), f(x) \rangle}_{\text{scalar}} + \underbrace{\mathrm{diag}(f(x))}_{n \times n} \underbrace{c(x)}_{n \times 1} \Big)$$

5 Main Results

In Section 5.1, we show the lipschitz bound of function f. In Section 5.2, we show our upper bound result of δ_b with respect to x. In Section 5.3, we show our upper bound result of δ_b with respect to A. We also conduct preliminary experiments in Appendix D that provide empirical validation for our theoretical results.

5.1 Lipschtiz Bound

To bound the shift of b, we first show the Lipschitz property for the basic functions:

- $\| \exp(Ax) - \exp(Ay) \|_2 \leq 2\sqrt{n} R \exp(R^2) \cdot \|x - y\|_2$
- $|\alpha(x) - \alpha(y)| \leq \| \exp(Ax) - \exp(Ay) \|_2 \cdot \sqrt{n}$
- $|\alpha(x)^{-1} - \alpha(y)^{-1}| \leq \beta^{-2} \cdot |\alpha(x) - \alpha(y)|$

Then, following the above Lipschitz property, we have

$$\|f(x_{t+1}) - f(x_t)\|_2 \leq 4\beta^{-2} n^{1.5} R \exp(2R^2) \cdot \|x_{t+1} - x_t\|_2$$

Similarly, we can show the Lipschtiz property of function f with respect to A as the following

$$\|f(A_{t+1}) - f(A_t)\|_2 \leq 4\beta^{-2} n^{1.5} R \exp(2R^2) \cdot \|A_{t+1} - A_t\|_2$$

Due to space limitation, we defer formal lemma and proof to Appendix A.2 and C.2.

5.2 Shifting Weight Parameter x

Theorem 5.1 (Bounded shift for shifting the weight parameter, formal of Theorem 1.4)**.** *If the following conditions hold*

- *Let* $A \in \mathbb{R}^{n \times d}$
- $\|A\| \leq R$
- $\|A(x_{t+1} - x_t)\|_\infty < 0.01$
- *Let* $R \geq 4$
- *Let* $M := n^{1.5} \exp(10R^2)$.

We consider the softmax regression problem

$$\min_x \| \langle \exp(Ax), \mathbf{1}_n \rangle^{-1} \exp(Ax) - b \|_2$$

If we move the x_t to x_{t+1}, then we're solving a new softmax regression problem with

$$\min_x \| \langle \exp(Ax), \mathbf{1}_n \rangle^{-1} \exp(Ax) - \widetilde{b} \|_2$$

where

$$\|\widetilde{b} - b\|_2 \leq M \cdot \|x_{t+1} - x_t\|_2$$

Proof. We have

$$\begin{aligned}
\|\widetilde{b} - b\|_2 &\leq 4\beta^{-2} n^{1.5} R \exp(2R^2) \cdot \|x_{t+1} - x_t\|_2 \\
&\leq 4 n^{1.5} R \exp(2R^2) \exp(2R^2) \cdot \|x_{t+1} - x_t\|_2 \\
&\leq n^{1.5} (4R) \exp(4R^2) \cdot \|x_{t+1} - x_t\|_2 \\
&\leq n^{1.5} \exp(6R^2) \exp(4R^2) \cdot \|x_{t+1} - x_t\|_2 \\
&\leq n^{1.5} \exp(10R^2) \cdot \|x_{t+1} - x_t\|_2 \\
&\leq M \cdot \|x_{t+1} - x_t\|_2
\end{aligned}$$

where the 1st step follows from Lemma A.5, the 2nd step comes from Lemma 3.3, the 3rd step comes from simple algebra, the 4th step follows from simple algebra, the 5th step follows from simple algebra and the 6th step follows from the definition of M. □

5.3 SHIFTING SENTENCE DATA A

Theorem 5.2 (Bounded shift for in-context learning, formal of Theorem 1.4)**.** *If the following conditions hold*

- *Let* $A \in \mathbb{R}^{n \times d}$
- $\|A\| \leq R$
- $\|(A_{t+1} - A_t)x\|_\infty < 0.01$
- *Let* $R \geq 4$
- *Let* $M := n^{1.5} \exp(10R^2)$.

We consider the softmax regression problem

$$\min_x \|\langle \exp(Ax), \mathbf{1}_n \rangle^{-1} \exp(Ax) - b\|_2$$

If we move the A_t *to* A_{t+1} *then we're solving a new softmax regression problem with*

$$\min_x \|\langle \exp(Ax), \mathbf{1}_n \rangle^{-1} \exp(Ax) - \widetilde{b}\|_2$$

where

$$\|\widetilde{b} - b\|_2 \leq M \cdot \|A_{t+1} - A_t\|.$$

Proof. We have

$$\begin{aligned}
\|\widetilde{b} - b\|_2 \leq & ~ 4\beta^{-2} n^{1.5} R \exp(2R^2) \cdot \|A_{t+1} - A_t\| \\
\leq & ~ 4n^{1.5} R \exp(2R^2) \exp(2R^2) \cdot \|A_{t+1} - A_t\| \\
\leq & ~ n^{1.5}(4R) \exp(4R^2) \cdot \|A_{t+1} - A_t\| \\
\leq & ~ n^{1.5} \exp(6R^2) \exp(4R^2) \cdot \|A_{t+1} - A_t\| \\
\leq & ~ n^{1.5} \exp(10R^2) \cdot \|A_{t+1} - A_t\| \\
\leq & ~ M \cdot \|A_{t+1} - A_t\|
\end{aligned}$$

where the 1st step follows from Lemma C.5, the 2nd step follows from Lemma 3.3, the 3rd step follows from simple algebra, the 4th step comes from simple algebra, the 5th step comes from simple algebra and the 6th step follows from the definition of M. □

6 CONCLUSION

The attention mechanism that incorporates the softmax unit is a crucial aspect of Large Language Models (LLMs) and significantly contributes to their extraordinary performance in various Natural Language Processing (NLP) tasks. The ability to learn in-context is highly valued in recent LLMs, and comprehending this concept is vital when querying LLMs. In this study, taking a step further from prior works' studies on linear Transformer's ability of learning linear functions, we examined the in-context learning process from a softmax regression perspective of Transformer's attention mechanism. We established the bound on the data transformations brought about by a single self-attention layer with softmax unit and gradient descent on an L2 regression loss. Our findings suggest that the update acquired through gradient descent and in-context learning are highly similar when training self-attention-only Transformers for softmax regression tasks, which is also validated through our preliminary experimental results. These results offer insights into the theoretical underpinnings of in-context learning in Transformers and can aid in improving the understanding and performance of LLMs in various NLP tasks. Our findings are restricted to small Transformer and simple regression problems. One interesting direction for further investigation is to acquire a comprehensive perception of in-context learning in larger models. To the best of our knowledge, we believe this work does not have any negative societal impact.

References

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*, 2022.

Josh Alman and Zhao Song. Fast attention requires bounded entries. *arXiv preprint arXiv:2302.13214*, 2023.

Leonard E Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, 37(6):1554–1563, 1966.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, March 2022. doi: 10.1162/coli_a_00422. URL https://aclanthology.org/2022.cl-1.7.

Satwik Bhattamishra, Kabir Ahuja, and Navin Goyal. On the Ability and Limitations of Transformers to Recognize Formal Languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7096–7116, Online, November 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.576. URL https://aclanthology.org/2020.emnlp-main.576.

Satwik Bhattamishra, Arkil Patel, and Navin Goyal. On the computational power of transformers and its implications in sequence modeling. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pp. 455–475, Online, November 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.conll-1.37. URL https://aclanthology.org/2020.conll-1.37.

Jan van den Brand, Zhao Song, and Tianyi Zhou. Algorithm and hardness for dynamic attention maintenance in large language models. *arXiv preprint arXiv:2304.02207*, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Beidi Chen, Tri Dao, Eric Winsor, Zhao Song, Atri Rudra, and Christopher Ré. Scatterbrain: Unifying sparse and low-rank attention. *Advances in Neural Information Processing Systems (NeurIPS)*, 34: 17413–17426, 2021.

Beidi Chen, Tri Dao, Kaizhao Liang, Jiaming Yang, Zhao Song, Atri Rudra, and Christopher Re. Pixelated butterfly: Simple and efficient sparse training for neural network models. In *International Conference on Learning Representations (ICLR)*, 2022.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

Tri Dao, Beidi Chen, Kaizhao Liang, Jiaming Yang, Zhao Song, Atri Rudra, and Christopher Re. Pixelated butterfly: Simple and efficient sparse training for neural network models. *arXiv preprint arXiv:2112.00029*, 2021.

Giannis Daras, Nikita Kitaev, Augustus Odena, and Alexandros G Dimakis. Smyrf-efficient attention using asymmetric clustering. *Advances in Neural Information Processing Systems (NeurIPS)*, 33: 6476–6489, 2020.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. *arXiv preprint arXiv:1807.03819*, 2018.

Yichuan Deng, Yeqi Gao, and Zhao Song. Solving tensor low cycle rank approximation. *arXiv preprint arXiv:2304.06594*, 2023a.

Yichuan Deng, Zhihang Li, and Zhao Song. Attention scheme inspired softmax regression. *arXiv preprint arXiv:2304.10411*, 2023b.

Yichuan Deng, Sridhar Mahadevan, and Zhao Song. Randomized and deterministic attention sparsification algorithms for over-parameterized feature dimension. *arxiv preprint: arxiv 2304.03426*, 2023c.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Javid Ebrahimi, Dhruv Gelda, and Wei Zhang. How can self-attention networks recognize Dyck-n languages? In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4301–4306, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.384. URL `https://aclanthology.org/2020.findings-emnlp.384`.

Benjamin L Edelman, Surbhi Goel, Sham Kakade, and Cyril Zhang. Inductive biases and variable creation in self-attention mechanisms. *arXiv preprint arXiv:2110.10090*, 2021.

Peng Gao, Chiori Hori, Shijie Geng, Takaaki Hori, and Jonathan Le Roux. Multi-pass transformer for machine translation. *arXiv preprint arXiv:2009.11382*, 2020.

Yeqi Gao, Sridhar Mahadevan, and Zhao Song. An over-parameterized exponential regression. *arXiv preprint arXiv:2303.16504*, 2023.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *arXiv preprint arXiv:2208.01066*, 2022.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1275. URL `https://aclanthology.org/D19-1275`.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pp. 5156–5165. PMLR, 2020.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.

Zhihang Li, Zhao Song, and Tianyi Zhou. Solving regularized exp, cosh and sinh regression problems. *arXiv preprint, 2303.15725*, 2023.

Zichang Liu, Jue Wang, Tri Dao, Tianyi Zhou, Binhang Yuan, Zhao Song, Anshumali Shrivastava, Ce Zhang, Yuandong Tian, Christopher Re, and Beidi Chen. Deja vu: Contextual sparsity for efficient llms at inference time. In *Manuscript*, 2023.

Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, 23(6), 2022.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. *arXiv preprint arXiv:2212.07677*, 2022.

Jorge Pérez, Javier Marinković, and Pablo Barceló. On the turing completeness of modern neural network architectures. *arXiv preprint arXiv:1901.03429*, 2019.

Gabriele Prato, Ella Charlaix, and Mehdi Rezagholizadeh. Fully quantized transformer for machine translation. *arXiv preprint arXiv:1910.10485*, 2019.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Charlie Snell, Ruiqi Zhong, Dan Klein, and Jacob Steinhardt. Approximating how single head attention learns. *arXiv preprint arXiv:2103.07601*, 2021.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4593–4601, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1452. URL https://aclanthology.org/P19-1452.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Jesse Vig and Yonatan Belinkov. Analyzing the structure of attention in a transformer language model. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 63–76, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4808. URL https://aclanthology.org/W19-4808.

James Vuckovic, Aristide Baratin, and Remi Tachet des Combes. A mathematical theory of attention. *arXiv preprint arXiv:2007.02876*, 2020.

Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

Colin Wei, Yining Chen, and Tengyu Ma. Statistically meaningful approximation: a case study on approximating turing machines with transformers. *arXiv preprint arXiv:2107.13163*, 2021.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 2021.

Shunyu Yao, Binghui Peng, Christos Papadimitriou, and Karthik Narasimhan. Self-attention networks can process bounded hierarchical languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 3770–3785, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.292. URL https://aclanthology.org/2021.acl-long.292.

Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank Reddi, and Sanjiv Kumar. Are transformers universal approximators of sequence-to-sequence functions? In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ByxRM0Ntvr.

Amir Zandieh, Insu Han, Majid Daliri, and Amin Karbasi. Kdeformer: Accelerating transformers via kernel density estimation. *arXiv preprint arXiv:2302.02451*, 2023.

Jingzhao Zhang, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank Reddi, Sanjiv Kumar, and Suvrit Sra. Why are adaptive methods good for attention models? *Advances in Neural Information Processing Systems*, 33:15383–15393, 2020.

Yi Zhang, Arturs Backurs, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, and Tal Wagner. Unveiling transformers with lego: a synthetic reasoning task, 2022. URL https://arxiv.org/abs/2206.04301.

Haoyu Zhao, Abhishek Panigrahi, Rong Ge, and Sanjeev Arora. Do transformers parse while predicting the masked word? *arXiv preprint arXiv:2303.08117*, 2023.

APPENDIX

Roadmap.

In Section A, we compute the Lipschitz with respect to x. In Section B, we give some definitions related to the softmax function of A. In Section C, we compute the Lipschitz with respect to A. In Section D, we show our preliminary experimental findings that support our theoretical results.

A LIPSCHITZ WITH RESPECT TO x

In Section A.1, we give the preliminary to compute the Lipschitz. In Section A.2, we show the upper bound of δ_b. In Section A.3, we compute the Lipschitiz of function $\exp(Ax)$ with respect to x. In Section A.4, we compute the Lipschitiz of the function α with respect to x. In Section A.5, we compute the Lipschitiz of function α^{-1} with respect to x.

A.1 PRELIMINARY

We can compute

$$\frac{\mathrm{d}L}{\mathrm{d}x} = g(x)$$

Let $\eta > 0$ denote the learning rate.

We update

$$x_{t+1} = x_t + \eta \cdot g(x_t)$$

Definition A.1. *We define $\delta_b \in \mathbb{R}^n$ to be the vector that satisfies the following conditions*

$$\|\langle \exp(Ax_{t+1}), \mathbf{1}_n \rangle^{-1} \exp(Ax_{t+1}) - b\|_2^2 = \|\langle \exp(Ax_t), \mathbf{1}_n \rangle^{-1} \exp(Ax_t) - b + \delta_b\|_2^2$$

Let $\{-1, +1\}^n$ denote a vector that each entry can be either -1 or $+1$. In the worst case, there are 2^n possible solutions, e.g.,

$$(\langle \exp(Ax_{t+1}), \mathbf{1}_n \rangle^{-1} \exp(Ax_{t+1}) - \langle \exp(Ax_t), \mathbf{1}_n \rangle^{-1} \exp(Ax_t)) \circ \{-1, +1\}^n$$

The norm of all the choices are the same. Thus, it is sufficient to only consider one solution as follows.

Claim A.2. *We can write δ_b as follows*

$$\delta_b = \underbrace{\langle \exp(Ax_{t+1}), \mathbf{1}_n \rangle^{-1} \exp(Ax_{t+1})}_{f(x_{t+1})} - \underbrace{\langle \exp(Ax_t), \mathbf{1}_n \rangle^{-1} \exp(Ax_t)}_{f(x_t)}.$$

Proof. The proof directly follows from Definition A.1. □

For convenience, we split δ_b into two terms, and provide the following definitions

Definition A.3. *We define*

$$\delta_{b,1} := (\langle \exp(Ax_{t+1}), \mathbf{1}_n \rangle^{-1} - \langle \exp(Ax_t), \mathbf{1}_n \rangle^{-1}) \cdot \exp(Ax_{t+1})$$
$$\delta_{b,2} := \langle \exp(Ax_t), \mathbf{1}_n \rangle^{-1} \cdot (\exp(Ax_{t+1}) - \exp(Ax_t))$$

Thus, we have

Lemma A.4. *We have*

- $$\delta_b = \delta_{b,1} + \delta_{b,2}$$

- *We can rewrite* $\delta_{b,1}$ *as follows*

$$\delta_{b,1} = (\alpha(x_{t+1})^{-1} - \alpha(x_t)^{-1}) \cdot \exp(Ax_{t+1}),$$

- *We can rewrite* $\delta_{b,2}$ *as follows*

$$\delta_{b,2} = \alpha(x_t)^{-1} \cdot (\exp(Ax_{t+1}) - \exp(Ax_t)).$$

Proof. We have

$$\begin{aligned}\delta_b &= \delta_{b,1} + \delta_{b,2} \\ &= \alpha(x_{t+1})^{-1} \exp(Ax_{t+1}) - \alpha(x_t)^{-1} \exp(Ax_{t+1}) + \\ &\quad \alpha(x_t)^{-1} \exp(Ax_{t+1}) - \alpha(x_t)^{-1} \exp(Ax_t) \\ &= \alpha(x_{t+1})^{-1} \exp(Ax_{t+1}) - \alpha(x_t)^{-1} \exp(Ax_t) \\ &= \langle \exp(Ax_{t+1}), \mathbf{1}_n \rangle^{-1} \exp(Ax_{t+1}) - \langle \exp(Ax_t), \mathbf{1}_n \rangle^{-1} \exp(Ax_t),\end{aligned}$$

where the 1st step follows from the definitions of δ_b, the 2nd step follows from the definitions of $\delta_{b,1}$ and $\delta_{b,2}$, the 3rd step follows from simple algebra, the 4th step comes from the definition of α. □

A.2 Upper Bounding δ_b with respect to x

We can show that

Lemma A.5. *If the following conditions hold*

- *Let* $\beta \in (0, 1)$.
- *Let* $\delta_{b,1} \in \mathbb{R}^n$ *be defined as Definition A.3.*
- *Let* $\delta_{b,2} \in \mathbb{R}^n$ *be defined as Definition A.3.*
- *Let* $\delta_b = \delta_{b,1} + \delta_{b,2}$.
- *Let* $R \geq 4$.

We have

- *Part 1.*

$$\|\delta_{b,1}\|_2 \leq 2\beta^{-2} n^{1.5} \exp(2R^2) \cdot \|x_{t+1} - x_t\|_2$$

- *Part 2.*

$$\|\delta_{b,2}\|_2 \leq 2\beta^{-1} \sqrt{n} R \exp(R^2) \cdot \|x_{t+1} - x_t\|_2$$

- *Part 3.*

$$\| \underbrace{f(x_{t+1}) - f(x_t)}_{\delta_b} \|_2 \leq 4\beta^{-2} n^{1.5} R \exp(2R^2) \cdot \|x_{t+1} - x_t\|_2$$

Proof. **Proof of Part 1.** We have

$$\begin{aligned}\|\delta_{b,1}\|_2 &\leq |\alpha(x_{t+1})^{-1} - \alpha(x_t)^{-1}| \cdot \|\exp(Ax_{t+1})\|_2 \\ &\leq |\alpha(x_{t+1})^{-1} - \alpha(x_t)^{-1}| \cdot \sqrt{n} \cdot \exp(R^2) \\ &\leq \beta^{-2} \cdot |\alpha(x_{t+1}) - \alpha(x_t)| \cdot \sqrt{n} \cdot \exp(R^2) \\ &\leq \beta^{-2} \cdot \sqrt{n} \cdot \|\exp(Ax_{t+1}) - \exp(Ax_t)\|_2 \cdot \sqrt{n} \cdot \exp(R^2) \\ &\leq \beta^{-2} \cdot \sqrt{n} \cdot 2\sqrt{n} R \exp(R^2) \|x_{t+1} - x_t\|_2 \cdot \sqrt{n} \cdot \exp(R^2) \\ &= 2\beta^{-2} n^{1.5} R \exp(2R^2) \cdot \|x_{t+1} - x_t\|_2\end{aligned}$$

where the first step follows from definition, the second step follows from assumption on A and x, the third step follows Lemma A.8, the forth step follows from Lemma A.7, the fifth step follows from Lemma A.6.

Proof of Part 2.

We have

$$\begin{aligned}\|\delta_{b,2}\|_2 \leq & ~ |\alpha(x_{t+1})^{-1}| \cdot \| \exp(Ax_{t+1}) - \exp(Ax_t)\|_2 \\ \leq & ~ \beta^{-1} \cdot \| \exp(Ax_{t+1}) - \exp(Ax_t)\|_2 \\ \leq & ~ \beta^{-1} \cdot 2\sqrt{n} R \exp(2R^2) \cdot \|x_{t+1} - x_t\|_2\end{aligned}$$

where the first step follows from definition, the 2nd step comes from Lemma A.6.

Proof of Part 3.

We have

$$\begin{aligned}\|\delta_b\|_2 = & ~ \|\delta_{b,1} + \delta_{b,2}\|_2 \\ \leq & ~ \|\delta_{b,1}\|_2 + \|\delta_{b,2}\|_2 \\ \leq & ~ 2\beta^{-2} n^{1.5} R \exp(2R^2) \cdot \|x_{t+1} - x_t\|_2 + 2\beta^{-1} n^{0.5} R \exp(2R^2) \cdot \|x_{t+1} - x_t\|_2 \\ \leq & ~ 2\beta^{-2} n^{1.5} R \exp(2R^2) \cdot \|x_{t+1} - x_t\|_2 + 2\beta^{-2} n^{1.5} R \exp(2R^2) \cdot \|x_{t+1} - x_t\|_2 \\ \leq & ~ 4\beta^{-2} n^{1.5} R \exp(2R^2) \cdot \|x_{t+1} - x_t\|_2\end{aligned}$$

where the 1st step follows from the definition of δ_b, the 2nd step follows from triangle inequality, the 3rd step follows from the results in Part 1 and Part 2, the 4th step follows from the fact that $n \geq 1$ and $\beta^{-1} \geq 1$, the 5th step follows from simple algebra. □

A.3 Lipschitz for function $\exp(Ax)$ with respect to x

Lemma A.6. *If the following conditions holds*

- *Let $A \in \mathbb{R}^{n \times d}$*
- *Let $\|A(y-x)\|_\infty < 0.01$*
- *Let $\|A\| \leq R$*
- *Let x, y satisfy that $\|x\|_2 \leq R$ and $\|y\|_2 \leq R$*

Then we have

$$\| \exp(Ax) - \exp(Ay)\|_2 \leq 2\sqrt{n} R \exp(R^2) \cdot \|x - y\|_2.$$

Proof. We have

$$\begin{aligned}\| \exp(Ax) - \exp(Ay)\|_2 \leq & ~ \| \exp(Ax)\|_2 \cdot 2\|A(x-y)\|_\infty \\ \leq & ~ \sqrt{n} \cdot \exp(\|Ax\|_2) \cdot 2\|A(x-y)\|_\infty \\ \leq & ~ \sqrt{n} \exp(R^2) \cdot 2\|A(x-y)\|_2 \\ \leq & ~ \sqrt{n} \exp(R^2) \cdot 2\|A\| \cdot \|x-y\|_2 \\ \leq & ~ 2\sqrt{n} R \exp(R^2) \cdot \|x-y\|_2\end{aligned}$$

where the 1st step follows from $\|A(y-x)\|_\infty < 0.01$ and Fact 3.1, the 2nd step comes from Fact 3.1, the 3rd step follows from Fact 3.2, the 4th step follows from Fact 3.2, the last step follows from $\|A\| \leq R$. □

A.4 Lipschitz for function $\alpha(x)$ with respect to x

We state a tool from previous work Deng et al. (2023b).

Lemma A.7 (Lemma 7.2 in Deng et al. (2023b))**.** *If the following conditions hold*

- *Let $\alpha(x)$ be defined as Definition 4.3*

Then we have

$$|\alpha(x) - \alpha(y)| \leq \| \exp(Ax) - \exp(Ay)\|_2 \cdot \sqrt{n}.$$

A.5 Lipschitz for function $\alpha(x)^{-1}$ with respect to x

We state a tool from previous work (Deng et al., 2023b).

Lemma A.8 (Lemma 7.2 in Deng et al. (2023b))**.** *If the following conditions hold*

- *Let $\langle \exp(Ax), \mathbf{1}_n \rangle \geq \beta$*
- *Let $\langle \exp(Ay), \mathbf{1}_n \rangle \geq \beta$*

Then, we have

$$|\alpha(x)^{-1} - \alpha(y)^{-1}| \leq \beta^{-2} \cdot |\alpha(x) - \alpha(y)|.$$

B Softmax Function with respect to A

In this section, we consider the function with respect to A. We define function softmax f as follows

Definition B.1 (Function f, Reparameterized x by A in Definition 4.1)**.** *Given a matrix $A \in \mathbb{R}^{n \times d}$. Let $\mathbf{1}_n$ denote a length-n vector that all entries are ones. We define prediction function $f : \mathbb{R}^{n \times d} \to \mathbb{R}^n$ as follows*

$$f(A) := \langle \exp(Ax), \mathbf{1}_n \rangle^{-1} \cdot \exp(Ax).$$

Similarly, we reparameterized x by A for our loss function L. We define loss function L as follows

Definition B.2 (Loss function $L_{\exp}$, Reparameterized x by A in Definition 4.2)**.** *Given a matrix $A \in \mathbb{R}^{n \times d}$ and a vector $b \in \mathbb{R}^{n \times d}$. We define loss function $L_{\exp} : \mathbb{R}^{n \times d} \to \mathbb{R}$ as follows*

$$L_{\exp}(A) := 0.5 \cdot \| \langle \exp(Ax), \mathbf{1}_n \rangle^{-1} \exp(Ax) - b\|_2^2.$$

For convenience, we define two helpful notations α and c with respect to A as follows:

Definition B.3 (Normalized coefficients, Reparameterized x by A in Definition 4.3)**.** *We define $\alpha : \mathbb{R}^{n \times d} \to \mathbb{R}$ as follows*

$$\alpha(A) := \langle \exp(Ax), \mathbf{1}_n \rangle.$$

Then, we can rewrite $f(A)$ (see Definition B.1) and $L_{\exp}(A)$ (see Definition B.2) as follows

- $f(A) = \alpha(A)^{-1} \cdot \exp(Ax)$.
- $L_{\exp}(A) = 0.5 \cdot \|\alpha(A)^{-1} \cdot \exp(Ax) - b\|_2^2$.
- $L_{\exp}(A) = 0.5 \cdot \|f(A) - b\|_2^2$.

Definition B.4 (Reparameterized x by A in Definition 4.4)**.** *We define function $c : \mathbb{R}^{n \times d} \in \mathbb{R}^n$ as follows*

$$c(A) := f(A) - b.$$

Then we can rewrite $L_{\exp}(A)$ (see Definition B.2) as follows

- $L_{\exp}(A) = 0.5 \cdot \|c(A)\|_2^2$.

C Lipschitz with respect to A

In Section C.1, we give the preliminary to compute the Lipschitz. In Section C.2, we show the upper bound of δ_b with respect to A. In Section C.3, we compute the Lipschitiz of function $\exp(Ax)$ with respect to A. In Section C.4, we compute the Lipschitiz of the function α with respect to A. In Section C.5, we compute the Lipschitiz of function α^{-1} with respect to A.

C.1 PRELIMINARY

We define δ_b as follows

Definition C.1 (Reparameterized x by A in Definition A.1)**.** *We define $\delta_b \in \mathbb{R}^n$ to be the vector that satisfies the following conditions*

$$\| \langle \exp(A_{t+1}x), \mathbf{1}_n \rangle^{-1} \exp(A_{t+1}x) - b \|_2^2 = \| \langle \exp(A_t x), \mathbf{1}_n \rangle^{-1} \exp(A_t x) - b + \delta_b \|_2^2$$

Claim C.2 (Reparameterized x by A in Definition A.2)**.** *We can write δ_b as follows*

$$\delta_b = \underbrace{\langle \exp(A_{t+1}x), \mathbf{1}_n \rangle^{-1} \exp(A_{t+1}x)}_{f(A_{t+1})} - \underbrace{\langle \exp(A_t x), \mathbf{1}_n \rangle^{-1} \exp(A_t x)}_{f(A_t)}.$$

Proof. The proof directly follows from Definition C.1. □

For convenient, we split δ_b into two terms, and provide the following definitions

Definition C.3 (Reparameterized x by A in Definition A.3)**.** *We define*

$$\delta_{b,1} := (\langle \exp(A_{t+1}x), \mathbf{1}_n \rangle^{-1} - \langle \exp(A_t x), \mathbf{1}_n \rangle^{-1}) \cdot \exp(A_{t+1}x)$$
$$\delta_{b,2} := \langle \exp(A_t x), \mathbf{1}_n \rangle^{-1} \cdot (\exp(A_{t+1}x) - \exp(A_t x))$$

Thus, we have

Lemma C.4 (Reparameterized x by A in Lemma A.4)**.** *We have*

- *We can rewrite $\delta_b \in \mathbb{R}^n$ as follows*
$$\delta_b = \delta_{b,1} + \delta_{b,2}$$
- *We can rewrite $\delta_{b,1} \in \mathbb{R}^n$ as follows*
$$\delta_{b,1} = (\alpha(A_{t+1})^{-1} - \alpha(A_t)^{-1}) \cdot \exp(A_{t+1}x),$$
- *We can rewrite $\delta_{b,2} \in \mathbb{R}^n$ as follows*
$$\delta_{b,2} = \alpha(A_t)^{-1} \cdot (\exp(A_{t+1}x) - \exp(A_t x)).$$

Proof. We have

$$\begin{aligned}
\delta_b &= \delta_{b,1} + \delta_{b,2} \\
&= \alpha(A_{t+1})^{-1} \exp(A_{t+1}x) - \alpha(A_t)^{-1} \exp(A_{t+1}x) + \\
&\quad \alpha(A_t)^{-1} \exp(A_{t+1}x) - \alpha(A_t)^{-1} \exp(A_t x) \\
&= \alpha(A_{t+1})^{-1} \exp(A_{t+1}x) - \alpha(A_t)^{-1} \exp(A_t x) \\
&= \langle \exp(A_{t+1}x), \mathbf{1}_n \rangle^{-1} \exp(A_{t+1}x) - \langle \exp(A_t x), \mathbf{1}_n \rangle^{-1} \exp(A_t x),
\end{aligned}$$

where the 1st step follows from the definitions of δ_b, the 2nd step follows from the definitions of $\delta_{b,1}$ and $\delta_{b,2}$, the 3rd step comes from simple algebra, the 4th step comes from the definition of α. □

C.2 UPPER BOUNDING δ_b WITH RESPECT TO A

We can show that

Lemma C.5 (Reparameterized x by A in Lemma A.5)**.** *If the following conditions hold*

- *Let $\beta \in (0, 1)$.*
- *Let $\delta_{b,1} \in \mathbb{R}^n$ be defined as Definition C.3.*
- *Let $\delta_{b,2} \in \mathbb{R}^n$ be defined as Definition C.3.*
- *Let $\delta_b = \delta_{b,1} + \delta_{b,2}$.*

- *Let* $R \geq 4$.

We have

- *Part 1.*

$$\|\delta_{b,1}\|_2 \leq 2\beta^{-2} n^{1.5} \exp(2R^2) \cdot \|A_{t+1} - A_t\|_2$$

- *Part 2.*

$$\|\delta_{b,2}\|_2 \leq 2\beta^{-1} \sqrt{n} R \exp(R^2) \cdot \|A_{t+1} - A_t\|_2$$

- *Part 3.*

$$\| \underbrace{f(A_{t+1}) - f(A_t)}_{\delta_b} \|_2 \leq 4\beta^{-2} n^{1.5} R \exp(2R^2) \cdot \|A_{t+1} - A_t\|_2$$

Proof. **Proof of Part 1.** We have

$$\begin{aligned}
\|\delta_{b,1}\|_2 &\leq |\alpha(A_{t+1})^{-1} - \alpha(A_t)^{-1}| \cdot \|\exp(A_{t+1}x)\|_2 \\
&\leq |\alpha(A_{t+1})^{-1} - \alpha(A_t)^{-1}| \cdot \sqrt{n} \cdot \exp(R^2) \\
&\leq \beta^{-2} \cdot |\alpha(A_{t+1}) - \alpha(A_t)| \cdot \sqrt{n} \cdot \exp(R^2) \\
&\leq \beta^{-2} \cdot \sqrt{n} \cdot \|\exp(A_{t+1}x) - \exp(A_t x)\|_2 \cdot \sqrt{n} \cdot \exp(R^2) \\
&\leq \beta^{-2} \cdot \sqrt{n} \cdot 2\sqrt{n} R \exp(R^2) \|A_{t+1} - A_t\| \cdot \sqrt{n} \cdot \exp(R^2) \\
&= 2\beta^{-2} n^{1.5} R \exp(2R^2) \cdot \|A_{t+1} - A_t\|
\end{aligned}$$

where the first step follows from definition, the second step follows from assumption on A and x, the third step follows Lemma C.8, the forth step follows from Lemma C.7, the fifth step follows from Lemma C.6.

Proof of Part 2.

We have

$$\begin{aligned}
\|\delta_{b,2}\|_2 &\leq |\alpha(A_{t+1})^{-1}| \cdot \|\exp(A_{t+1}x) - \exp(A_t x)\|_2 \\
&\leq \beta^{-1} \cdot \|\exp(A_{t+1}x) - \exp(A_t x)\|_2 \\
&\leq \beta^{-1} \cdot 2\sqrt{n} R \exp(2R^2) \cdot \|A_{t+1} - A_t\|
\end{aligned}$$

Proof of Part 3.

We have

$$\begin{aligned}
\|\delta_b\|_2 &= \|\delta_{b,1} + \delta_{b,2}\|_2 \\
&\leq \|\delta_{b,1}\|_2 + \|\delta_{b,2}\|_2 \\
&\leq 2\beta^{-2} n^{1.5} R \exp(2R^2) \cdot \|A_{t+1} - A_t\| + 2\beta^{-1} n^{0.5} R \exp(2R^2) \cdot \|A_{t+1} - A_t\| \\
&\leq 2\beta^{-2} n^{1.5} R \exp(2R^2) \cdot \|A_{t+1} - A_t\| + 2\beta^{-2} n^{1.5} R \exp(2R^2) \cdot \|A_{t+1} - A_t\| \\
&\leq 4\beta^{-2} n^{1.5} R \exp(2R^2) \cdot \|A_{t+1} - A_t\|
\end{aligned}$$

where the 1st step follows from the definition of δ_b, the 2nd step comes from triangle inequality, the 3rd step comes from the results in Part 1 and Part 2, the 4th step follows from the fact that $n \geq 1$ and $\beta^{-1} \geq 1$, the 5th step follows from simple algebra. □

C.3 Lipschitz for function $\exp(Ax)$ with respect to A

Lemma C.6 (Reparameterized x by A in Lemma A.6)**.** *If the following conditions holds*

- *Let* $A, B \in \mathbb{R}^{n \times d}$
- *Let* $\|(A - B)x\|_\infty < 0.01$

- *Let* $\|A\| \leq R$
- *Let* x *satisfy that* $\|x\|_2 \leq R$

Then we have

$$\| \exp(Ax) - \exp(Bx)\|_2 \leq 2\sqrt{n} R \exp(R^2) \cdot \|A - B\|.$$

Proof. We have

$$\begin{aligned} \| \exp(Ax) - \exp(Bx)\|_2 &\leq \| \exp(Ax)\|_2 \cdot 2\|(A-B)x\|_\infty \\ &\leq \sqrt{n} \cdot \exp(\|Ax\|_2) \cdot 2\|(A-B)x\|_\infty \\ &\leq \sqrt{n} \exp(R^2) \cdot 2\|(A-B)x\|_2 \\ &\leq \sqrt{n} \exp(R^2) \cdot 2\|A-B\| \cdot \|x\|_2 \\ &\leq 2\sqrt{n} R \exp(R^2) \cdot \|A-B\| \end{aligned}$$

where the 1st step follows from $\|A(y-x)\|_\infty < 0.01$ and Fact 3.1, the 2nd step follows from Fact 3.1, the 3rd step follows from Fact 3.2, the 4th step comes from Fact 3.2, the last step follows from $\|A\| \leq R$. □

C.4 Lipschitz for function $\alpha(A)$ with respect to A

Lemma C.7 (Reparameterized x by A in Lemma A.7)**.** *If the following conditions hold*

- *Let* $\alpha(A)$ *be defined as Definition B.3*

Then we have

$$|\alpha(A) - \alpha(B)| \leq \| \exp(Ax) - \exp(Bx)\|_2 \cdot \sqrt{n}.$$

Proof. We have

$$\begin{aligned} |\alpha(A) - \alpha(B)| &= |\langle \exp(Ax) - \exp(Bx), \mathbf{1}_n \rangle| \\ &\leq \| \exp(Ax) - \exp(Bx)\|_2 \cdot \sqrt{n} \end{aligned}$$

where the 1st step comes from the definition of $\alpha(x)$, the 2nd step follows from Cauchy-Schwarz inequality (Fact 3.1). □

C.5 Lipschitz for function $\alpha(A)^{-1}$ with respect to A

Lemma C.8 (Reparameterized x by A in Lemma A.8)**.** *If the following conditions hold*

- *Let* $\langle \exp(Ax), \mathbf{1}_n \rangle \geq \beta$
- *Let* $\langle \exp(Bx), \mathbf{1}_n \rangle \geq \beta$

Then, we have

$$|\alpha(A)^{-1} - \alpha(B)^{-1}| \leq \beta^{-2} \cdot |\alpha(A) - \alpha(B)|.$$

Proof. We can show that

$$\begin{aligned} |\alpha(A)^{-1} - \alpha(B)^{-1}| &= \alpha(A)^{-1}\alpha(B)^{-1} \cdot |\alpha(A) - \alpha(B)| \\ &\leq \beta^{-2} \cdot |\alpha(A) - \alpha(B)| \end{aligned}$$

where the 1st step follows from simple algebra, the 2nd step follows from $\alpha(A) \geq \beta, \alpha(B) \geq \beta$. □

D Experiments

In this section, we show preliminary experimental findings supporting our theoretical results that when training self-attention-only Transformers for softmax regression tasks, the models learned by gradient-descent and Transformers show great similarity.

Experiments setup. According to Definition 1.3, we construct the synthetic softmax regression tasks consists of randomly sampled length-n documents $A \in \mathbb{R}^{n \times d}$ where each word has the d-dimensional embedding and targets $b \in \mathbb{R}^n$. In our experiments we choose a set of different value of document length $n \in \{25, 50, 100, 200, 400\}$ and a set of different embedding size $d \in \{5, 10, 20, 35, 50\}$. Following Oswald et al. (2022), we compare the two models in our experiment: a trained single self-attention (SA) layer with a softmax unit approximating the full Transformers, and a softmax regression model trained with one-step gradient descent. The training objective for both models is defined as in Definition 1.3. For the single self-attention layer with a softmax unit, we choose the learning rate $\eta_{\mathrm{SA}} = 0.005$. For the softmax regression model, we determine the optimal learning rate η_{GD} by minimizing the ℓ_2 regression loss over a training set of 10^3 tasks through line search.

To compare the trained single self-attention layer with a softmax unit and the softmax regression model trained with one-step gradient descent, we sample 10^3 tasks and record the losses of two models. In addition, we follow Oswald et al. (2022) to record

- **Pred Diff**: the predictions difference measured with the ℓ_2 norm:

$$\|\widehat{y}_{\mathrm{SA}}(A) - \widehat{y}_{\mathrm{GD}}(x)\|_2$$

  where $\widehat{y}_{\mathrm{SA}}(A)$ is corresponding to the $\widetilde{b}$ in Theorem 5.1, and $\widehat{y}_{\mathrm{GD}}(x)$ is corresponding to the $\widetilde{b}$ in Theorem 5.2.
- **Model Cos**: the cosine similarity between the sensitivities of two models:

$$\mathrm{CosSim}(\frac{\partial \widehat{y}_{\mathrm{GD}}(x)}{\partial x}, \frac{\partial \widehat{y}_{\mathrm{SA}}(A)}{\partial A})$$

- **Model Diff**: the model sensitivity difference measured with the ℓ_2 norm:

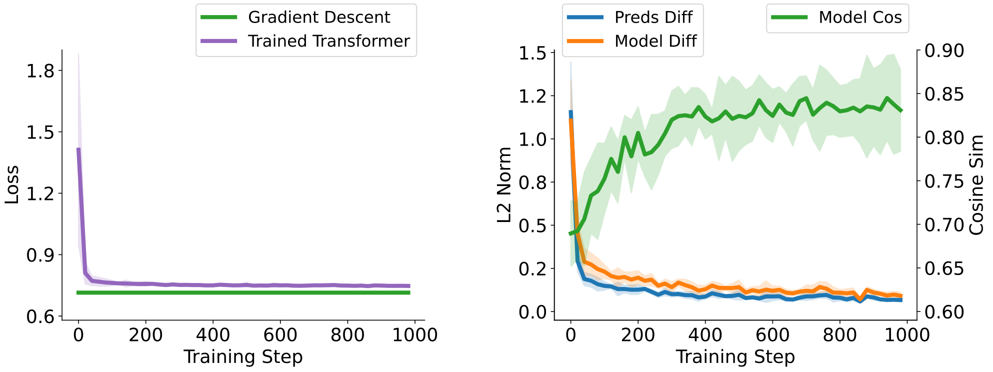
$$\|\frac{\partial \widehat{y}_{\mathrm{GD}}(x)}{\partial x} - \frac{\partial \widehat{y}_{\mathrm{SA}}(A)}{\partial A}\|_2$$

All experiments run on a single NVIDIA RTX2080Ti GPU with 10 independent repetitions.

Results on tasks of different document lengths. The results of the comparisons between a trained single self-attention layer and one-step gradient descent on synthetic softmax regression tasks of document length $n \in \{25, 50, 100, 200, 400\}$ and word embedding size $d = 20$ are shown in Figure 1-5. We measure two models' losses and similarities over the training steps of the SA layer for each set of tasks. From the results, we observe identical performances of the two models measured in losses. We also observe considerable alignment of the two models across tasks of different document lengths, indicated by decreasing prediction and model difference and increasing cosine similarity between models. Besides, comparing all five figures, we observe that with larger document length n, which is common in practical language modeling tasks, the two models show greater similarity, validating the Transformer's in-context learning ability.

Results on tasks of different word embedding sizes. The results of the comparisons between a trained single self-attention layer and one-step gradient descent on synthetic softmax regression tasks of document length $n = 200$ and word embedding size $d \in \{5, 10, 20, 35, 50\}$ are shown in Figure 4 and 6-9. Similarly, we measure two models' losses and similarities over training steps of the SA layer for each set of tasks. We again observe similar performances and close alignment of the two models.

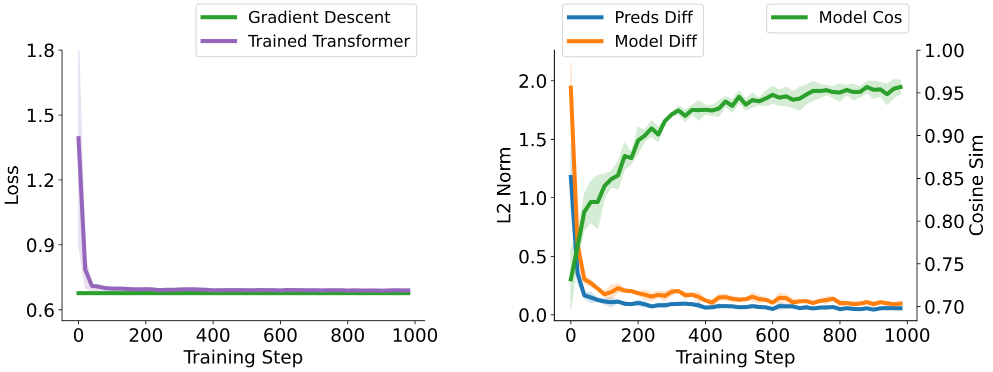
In conclusion, our experimental results empirically validate our theoretical results in Section 5, showing that when training self-attention-only Transformers for softmax regression tasks, the models learned by gradient-descent and Transformers show great similarity. Due to the non-linearity of softmax regression, it is not expected for models to match exactly as implied in our theoretical results in Section 5, which is also observed in our experimental findings.



(a) Losses over training steps of Transformer

(b) Difference and similarity over training steps

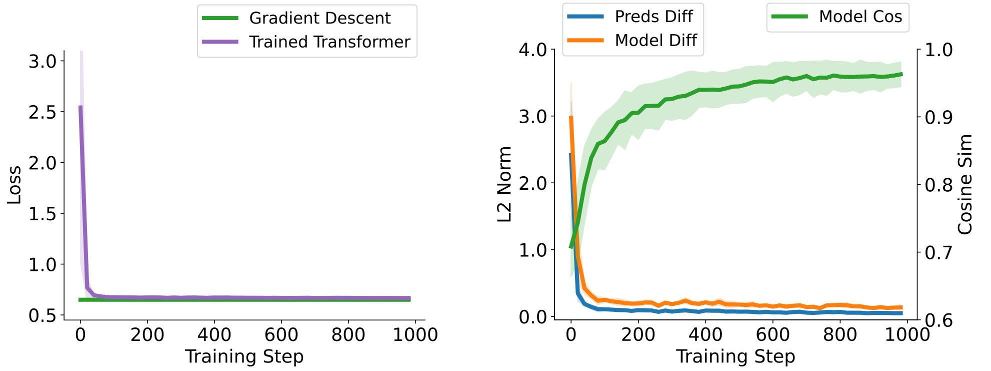
Figure 1: Comparison between trained single-SA-layer Transformer and one-step GD on softmax regression tasks of **document length $\boldsymbol{n = 25}$** and embedding size $d = 20$.



(a) Losses over training steps of Transformer

(b) Difference and similarity over training steps

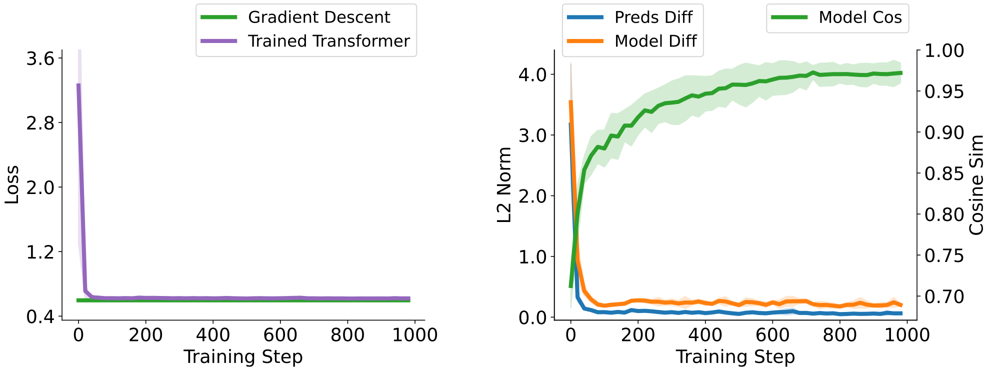
Figure 2: Comparison between trained single-SA-layer Transformer and one-step GD on softmax regression tasks of **document length $\boldsymbol{n = 50}$** and embedding size $d = 20$.



(a) Losses over training steps of Transformer

(b) Difference and similarity over training steps

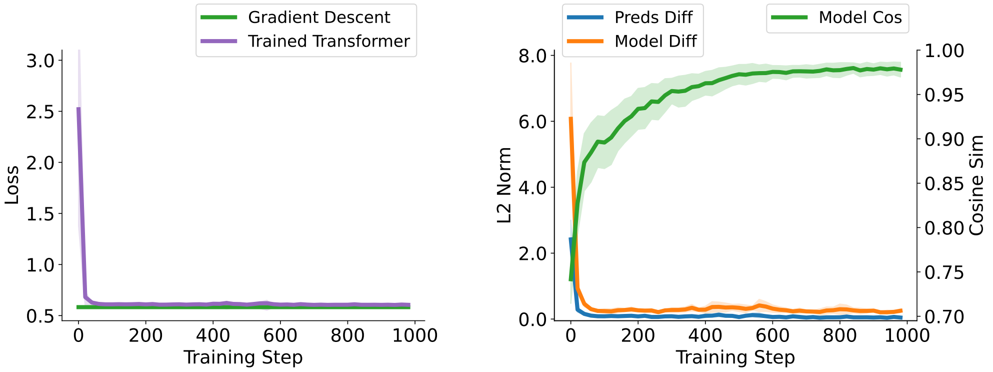
Figure 3: Comparison between trained one-SA-layer Transformer and one-step GD on softmax regression tasks of **document length $\boldsymbol{n = 100}$** and embedding size $d = 20$.



(a) Losses over training steps of Transformer

(b) Difference and similarity over training steps

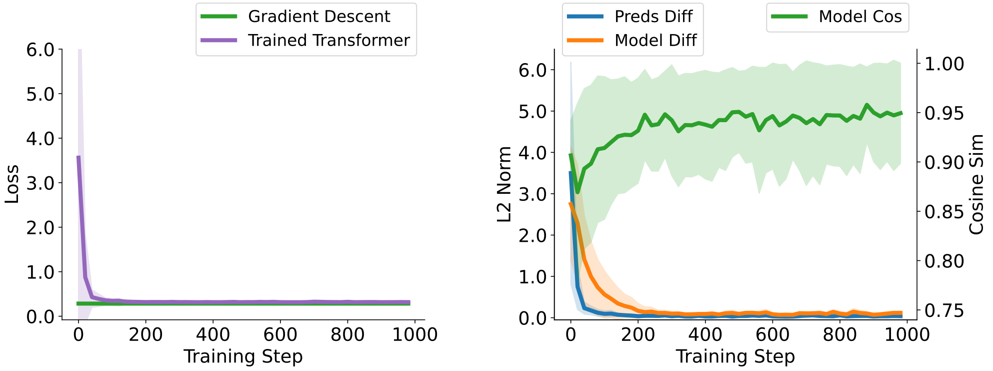
Figure 4: Comparison between trained one-SA-layer Transformer and one-step GD on softmax regression tasks of **document length $n = 200$** and **embedding size $d = 20$**.



(a) Losses over training steps of Transformer

(b) Difference and similarity over training steps

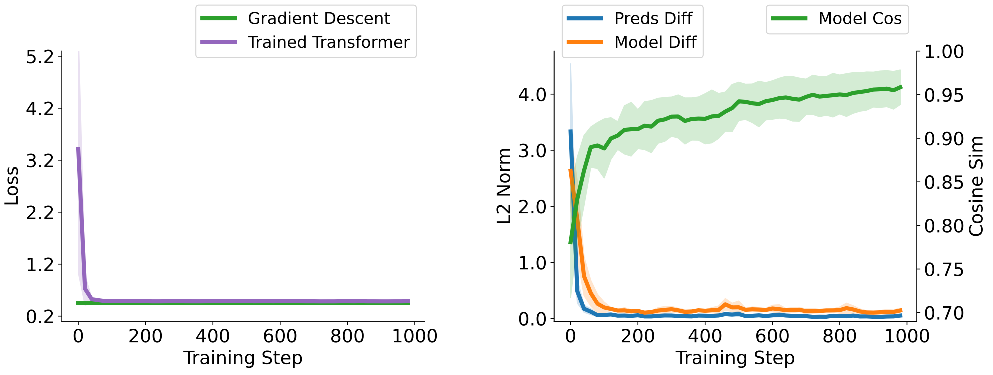
Figure 5: Comparison between trained one-SA-layer Transformer and one-step GD on softmax regression tasks of **document length $n = 400$** and embedding size $d = 20$.



(a) Losses over training steps of Transformer

(b) Difference and similarity over training steps

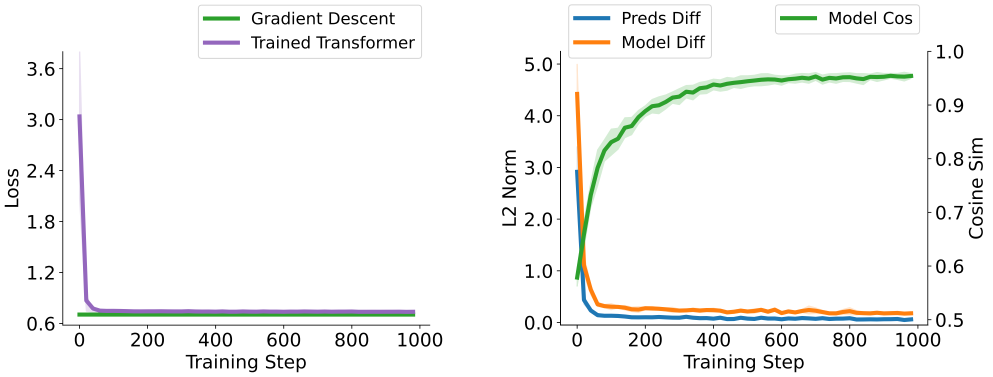
Figure 6: Comparison between trained one-SA-layer Transformer and one-step GD on softmax regression tasks of document length $n = 200$ and **embedding size $d = 5$**.



(a) Losses over training steps of Transformer

(b) Difference and similarity over training steps

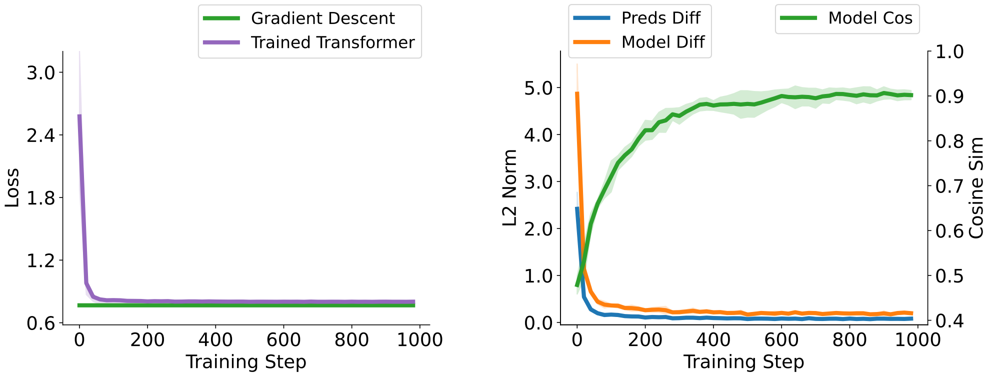
Figure 7: Comparison between trained one-SA-layer Transformer and one-step GD on softmax regression tasks of document length $n = 200$ and **embedding size $d = 10$**.



(a) Losses over training steps of Transformer

(b) Difference and similarity over training steps

Figure 8: Comparison between trained one-SA-layer Transformer and one-step GD on softmax regression tasks of document length $n = 200$ and **embedding size $d = 35$**.



(a) Losses over training steps of Transformer

(b) Difference and similarity over training steps

Figure 9: Comparison between trained one-SA-layer Transformer and one-step GD on softmax regression tasks of document length $n = 200$ and **embedding size $d = 50$**.