In-Context Learning behaves as a greedy layer-wise gradient descent algorithm

Brian K Chen
National University of Singapore
e0694208@u.nus.edu

Tianyang Hu
Huawei Noah's Ark Lab
hutianyang1@huawei.com

Hui Jin
Huawei Noah's Ark Lab
jinhui34@huawei.com

Lee Hwee Kuan
Agency for Science, Technology and Research
leehk@bii.a-star.edu.sg

Kenji Kawaguchi
National University of Singapore
kenji@comp.nus.edu.sg

Abstract

In-context learning (ICL) is a powerful capability of large language models that have shown up in past years. Despite its impact, the exact mechanism behind ICL is still only understood to a very limited capacity. In this paper, we suggest that ICL on a single linearized self-attention layer is equivalent to a single step of gradient descent with a specific dataset. This property is shown without additional assumptions on the model parameters which is required in other work in the field. We then extend our setting to a more realistic multi-layer framework and observe that in-context learning resembles using a greedy-layer-wise algorithm to update the weights within a large language model with multiple layers. Last but not least, we extend our theoretical conclusions to the autoregressive setting. We notice that many other works comparing ICL to gradient descent are restricted to very specific settings that do not contain a causal mask.

1 Introduction

In-context learning presents a whole new world of possibilities compared to traditional gradient-based fine-tuning. It allows us to update large language models (LLMs) without the increasingly costly training process. The regular fine-tuning process is often prohibitively expensive, with state-of-the-art models such as Llama-3 surpassing 70 billion parameters. ICL potentially democratizes the updating of large language models, allowing users other than large companies with copious resources to tune their own models. ICL is also highly interpretable when including demonstration examples in the prompts. Demonstration examples typically take the form of natural language prompts which are readily understandable by humans.

Although In-context learning exhibits many promising qualities, how it works is still poorly understood. In the past few years, there have been many attempts, both theoretical and empirical, to understand why exactly ICL occurs and how it ties into the larger transformer architecture. However, the conclusions are still very limited and many questions have yet to be answered.

This study proposes that, for a single layer, ICL in linearized transformers is equivalent to conducting gradient descent with a specific training set determined by the input prompt. For multiple layers, ICL constitutes a greedy layer-wise gradient descent update. We do so by highlighting the dual relationship between the linear self-attention mechanism and gradient descent of linear models in l_2 regression. This results in the following key contributions:

Preprint. Under review.

- Demonstrate the equivalence between in-context prompts and a meta-gradient descent update upon the query of the linear self-attention mechanism to provide intuition on the ICL mechanism.
- Observe how in-context learning on a multilayer transformer model constitutes a form of the greedy layer-wise training algorithm.
- Analyse how ICL can be viewed through the prism of greedy layer-wise training algorithms. The properties of greedy layer-wise training algorithms help elucidate many current observations regarding ICL.
- Extend our findings in a theoretical capacity to the autoregressive transformers including a causal mask.

Most established work in the field is limited to the regression setting. Furthermore, there are strong assumptions placed upon the Key and Value matrices. In our work, we make no assumptions and expand the discussion to include all linearized transformers. We hope that our comparison of ICL to a greedy layer-wise learning algorithm can help us better understand the nature of ICL and hopefully apply it more effectively in the future.

2 Preliminaries

In this paper, we focus on transformers consisting of attention modules and feed-forward networks. More specifically, we analyze the attention module of the transformer layer. This is because other components of transformers such as Layer-normalisations and feed-forward neural networks are typically token-wise operators. The input of each layer consists of a sequence of mathematical tokens $X = [p_n, ..., p_1]^T$. We assume each token has dimensionality d_{in} such that $X \in \mathbb{R}^{N \times d_{in}}$.

2.1 The attention mechanism

Regarding attention, we discard the scaling factor $\sqrt{d_k}$ and approximate the softmax by a kernel. This results in a linearized attention function. Studies have shown that models built on linearized attention can still provide reasonable results [Katharopoulos et al., 2020, Lu et al., 2021] and have an inherent ICL ability.

Definition: (Linearized attention) The input consists of the query (Q), key (K) and value (V) matrices which have dimensions d_k, d_k and d_v respectively. With kernel representation function $\phi(\cdot)$, the linear attention is computed as:

$$LinAttn(V, \phi(K), \phi(Q)) = V\phi(K)^T\phi(Q). \quad (1)$$

We explicitly focus on single-headed attention. This is due to the simplicity of notation. All results can be extended to multi-headed attention which is used in most real-world settings.

Definition: (Single-headed attention layer) Given input data $X \in \mathbb{R}^{d_{in} \times N}$ a single attention layer is characterized by trainable matrices W_Q, W_K, W_V which are in $\mathbb{R}^{d_K \times d_{in}}$, $\mathbb{R}^{d_K \times d_{in}}$ and $\mathbb{R}^{d_V \times d_{in}}$ respectively. The Single-headed attention layer takes the form:

$$Attn(W_V X, W_K X, W_Q X). \quad (2)$$

In our case, we will be looking at:

$$LinAttn(W_V X, \phi(W_K X), \phi(W_Q X)). \quad (3)$$

2.2 In-context learning

In this paper, we choose to consider the most general case of ICL. The ICL prompt is treated as a sequence of tokens without added requirements on the structure. We have an initial prompt $X = [p_N, ..., p_1]$ of length N and a Demonstration ICL prompt $X' = [p'_M, ..., p'_1]$ of length M. The final input concatenates X' and X to form a sequence $[X'; X] = [p'_M, ..., p'_1, p_N, ..., p_1] \in \mathbb{R}^{d_{in} \times (M+N)}$.

2.3 Dual form between linear attention and gradient descent on a linear function

This study seeks to take advantage of the duality between the linear attention operator and gradient descent on a linear function. Based on the work of Irie et al. [2022]:

Proposition 2.1 (**Dual form of a linear function trained by gradient descent).** *Let $f(x) = Wx$ be a linear function $f : \mathbb{R}^{d_{in}} \to \mathbb{R}^{d_{out}}$ with parameters $W \in \mathbb{R}^{d_{out} \times d_{in}}$. Given gradient descent with l_2 loss, T training samples $\{x_i, y_i\}_{i=1}^{T}$ and learning rate η, we have the identity:*

$$W_1 x = (W_0 - \eta \nabla \frac{1}{T} \sum_{i=1}^{T} l_2(f_W(x_i), y_i))|_{W=W_0} x = W_0 x + LinAttn(\frac{\eta}{T} E, X, x). \tag{4}$$

X is the matrix of inputs $X = [x_1; ...; x_T]$ and $E = Y - W_0 X$ is the error matrix where $Y = [y_1, ..., y_T]$.

3 Viewing in-context learning with linear attention as a gradient descent step

First, we examine a single-layer self-attention mechanism. We consider how the ICL prompt affects each singular query token $\mathbf{q} = W_Q \mathbf{x}$ where $\mathbf{x} \in \mathbb{R}^{in}$. Given ICL prompt $X' = [p'_M, ..., p'_1]$ and initial prompt $X = [p_N, ..., p_1]$, the attention result of the linear attention head can be expressed as:

$$\begin{aligned}\mathcal{F}([X'; X], \mathbf{q}) &= LinAttn(W_V[X'; X], \phi(W_K[X'; X]), \mathbf{q}) \\ &= LinAttn(W_V[X], \phi(W_K[X]), \mathbf{q}) + LinAttn(W_V[X'], \phi(W_K[X']), \mathbf{q}) \\ &= \mathcal{F}([X], \mathbf{q}) + LinAttn(W_V[X'], \phi(W_K[X']), \mathbf{q})\end{aligned} \tag{5}$$

where $\mathcal{F}([X], \mathbf{q}) = LinAttn(W_V[X], \phi(W_K[X]), \mathbf{q})$. N.B. $\mathcal{F}$ can take inputs $[X]$ of differing dimensions. There is a clear similarity between the form of equation (4) and equation (5). This leads to the main theorem:

Theorem 3.1 (**Dual form between in-context learning and gradient descent).** [1] *For an initial self-attention mechanism with matrices W_V, W_K, and prompt $X = [p_N, ..., p_1]$, we have the operator $\mathcal{F}_0([X], \boldsymbol{q})$. The following systems are equivalent (i.e. $S_1 = S_2$ for all $\boldsymbol{q}$):*

$$S_1 = \mathcal{F}_0([X'; X], \boldsymbol{q}) \tag{6}$$

and

$$S_2 = \mathcal{F}_1([X], \boldsymbol{q}), \tag{7}$$

where $\mathcal{F}_1([X], .)$ is the linear function $\mathcal{F}_0([X], .) = W_{[X]}(.) := W_V[X](\phi(W_K[X]))^T(.)$ after one step of gradient descent with learning rate η and training set $\{x_i, y_i\}_{i=1}^{M}$. For every $i \in \{1, ..., M\}$, $x_i = \phi(W_K p'_i)$ and $y_i = \frac{M}{\eta} W_V p'_i + \mathcal{F}_0([X], \phi(W_K p'_i))$.

This allows us to arrive at a few interesting observations:

1. Theorem 3.1 demonstrates that in-context learning forms a type of meta-optimizer on the query resembling gradient descent with very specific training data for linearized transformers. Unlike past conclusions, our statement isn't constrained to specific regression settings and values for W_Q, W_K and W_V.
2. If one takes $y_i = \mathcal{F}_0([X], W_k p'_i)$ and ignores the other half, the loss is 0 which means that the gradient descent has no effect at all. We can consider this as a baseline. The significant part is: $\frac{M}{\eta} W_v p'_i$.
3. $W_V p'_i$ is intuitively the "value" which we place upon a token $W_Q p'_i$. Here we are placing emphasis of $W_K p'_i$ instead when applied to the function based on $[p_N, ..., p_1]$

4 Extension to multiple layers

This section extends Theorem 3.1 to a more general setting beyond the single linearized attention layer. Consider a more realistic model architecture with L layers stacked upon each other: $f(x) =$

[1] During writing, we found concurrent work with a similar result by Ren and Liu [2023]

$(T_L + I) \circ (T_{L-1} + I) \circ ... \circ (T_1 + I)(x)$ where for each $i \in \{1, ..., L\}$, T_i is either an FFN layer with a residual connection or a linear self-attention layer $T_i = LSA_i(x)$ with corresponding weight matrices $W_{K_i}, W_{Q_i}, W_{V_i}$, and I is the identity function to capture the residual connection. Given a prompt $X = [p_n, ..., p_1]$, we define

$$LSA_i([p_n, ..., p_1]) = W_{base,i}([p_n, ..., p_1])W_{Q_i}[p_n, ..., p_1] \quad (8)$$

$$W_{base,i}(X) = W_{V_i}X\phi(W_{K_i}X)^T \quad (9)$$

Algorithm 1: ICL imitation algorithm

1: input: f_1 and $[p'_m, ..., p'_1, p_n, ..., p_1]$
2: for $i \in \{1, ..., L\}$
IF T_i is a FFN with residual connection, return
$[p'_m, ..., p'_1, p_n, ..., p_1] = (T_i + I)([p'_m, ..., p'_1, p_n, ..., p_1])$

ELSE $T_i = LSA_i$
(a) construct matrix $W_0 = W_{base,i}([p_n, ..., p_1])$
(b) Update the linear functional $f(x) = W_0 x$ with a single step of gradient descent with learning rate m and training set $\{\phi(), W_{V_i}p'_j + W_0\phi(W_{K_i}p'_j))\}_{j=1}^m$ such that the updated weights are W_1
(c) $[p'_m, ..., p'_1, p_n, ..., p_1] = W_1\phi(W_{Q_i}[p'_m, ..., p'_1, p_n, ..., p_1]) + [p'_m, ..., p'_1, p_n, ..., p_1]$

Theorem 4.1 (**Dual form of the transformer algorithm**). *For a model f_1 described above and a prompt $[p'_m, ..., p'_1, p_n, ..., p_1]$, the ICL imitation algorithm on f_1 and $[p'_m, ..., p'_1, p_n, ..., p_1]$ (Algorithm 1) produces the same output as $f_1([p'_m, ..., p'_1, p_n, ..., p_1])$*

Proof. The proof of equivalence is trivial through repeated applications of Theorem 3.1.

There are a few key features of algorithm 1. First of all, algorithm 1 is a recursive algorithm that updates the layers one after another. When applying the model, the i-th layer is updated and the newly updated weights are used to generate the input which will be used to update the $i + 1$-st layer. A second key feature of algorithm 1 is that it is a form of unsupervised learning. This may seem contradictory since we are conducting gradient descent descent on each layer. However, a key observation is that the labels are actually generated from the inputs themselves.

5 Connection to the greedy-layer-wise algorithms

Upon closer inspection, these features of Algorithm 1 take the form of a greedy layer-wise unsupervised pretraining algorithm proposed by Bengio et al. [2006]. In that paper, they propose a greedy layer-wise unsupervised training algorithm to train both deep belief networks (DBN) and auto-encoders in an unsupervised regime. Their experiments show that the general principles of the greedy layer-wise training algorithms can be extended past DBNs and applied to other unsupervised GLT algorithms. We can consider the inclusion of the ICL prompts in 1 as a single pass of a greedy layer-wise training algorithm trained on unsupervised data (The transformed in-context prompts). We argue that there is evidence to show that those general principles may apply to ICL and transformers as well.

First of all, greedy layer-wise training (GLT) is observed to achieve very quick convergence to a local solution [Hinton et al., 2006]. This may explain why only a single step is required within the ICL case. There has been research regarding ICL which has indicated that one step of gradient descent is provably the optimal ICL learner for a single layer of linearized self attention[Mahankali et al., 2023]. This would seem to demonstrate how ICL has displayed such effectiveness despite only resembling a single step of a greedy layer-wise training algorithm.

Secondly, the work by [Bengio et al., 2006] shows that greedy layer-wise training algorithms help learn internal representations that represent higher-level abstractions. Several empirical works studying ICL have shown that, regarding ICL demonstrations, the model learns the format that we are studying rather than the exact detailed labels. They suggest that the model is learning higher-level abstractions rather than specific values. This would align with what we expect from GLT algorithms.

This observation perfectly ties into another property of GLT algorithms. Work by [Bengio et al., 2006] states that by learning high-level internal representations, GLT algorithms are best served for quickly initializing the parameters of a model before other fine-tuning steps. This could possibly motivate future fine-tuning attempts that have a fixed ICL prompt included to provide initialization. We find that such attempts do already exist, in the form of instruction tuning[Zhang et al., 2024, Wei et al., 2022]. Instruction tuning involves fine-tuning LLMs with datasets with form (*INSTRUCTION, OUTPUT*). The model learns to adapt to new tasks which can also be given instructions. In this case, we consider the (*INSTRUCTION*) as a form of context itself. This suggests the need for a unified framework for ICL, regular fine-tuning, and instruction tuning altogether. Perhaps they truly are not substitutes for one another but rather complements.

5.1 Instruction learning with a single instruction

From the comparison between regular ICL and instruction-tuning, we propose a specific variant of instruction-tuning that combines ICL and fine-tuning. In this regime, given a specific purpose, we first determine the appropriate ICL prompt X' for future prompts X. However, in this case we consider that ICL may not be sufficient, so we treat it purely as a form of projection or initialization for fine-tuning.

Given fixed ICL prompt X', we want the fine-tuning set to have form $\{[X'; X_i]\}_{i=1}^{N}$. All inputs in the training set will carry the form $[X'; X_i]$. This incorporates the initialization we obtain from ICL into the fine-tuning process and should allow the fine-tuning process to be faster and less costly. Future inputs should then take the form $[X', X]$ as well so the initialization is included permanently. We expect this to be an effective way to combine ICL and fine-tuning for the best of both worlds, both increasing the effectiveness of ICL and reducing the training cost of existing fine-tuning methods.

6 Extension to the autoregressive case

In previous sections, we limit our analysis to linearized attention without a causal mask. Similar limitations are present in other works studying the relationship between gradient descent and ICL. In this section, we attempt to extend it to the autoregressive setting. To do so we first write equation 1 as follows:

$$O_i = \sum_{j=1}^{N} sim(Q_i, K_j)V_j = \sum_{j=1}^{N} \phi(Q_i)^T \phi(K_j)V_j \tag{10}$$

This is drawn from the work by [Katharopoulos et al., 2020] and is equivalent to the equation 1. To construct the autoregressive form, we convert the equation to:

$$O_i = \sum_{j=1}^{i} \phi(Q_i)\phi(K_j)V_j = \phi(Q_i)^T \sum_{j=1}^{i} \phi(K_j)V_j \tag{11}$$

This means that for each token Q_i, there is a separate corresponding reference equation $\mathcal{F}_0^{(i)}([X], \mathbf{q})$. Through the application of Theorem 4.1, we can arrive at the statement.

Lemma 6.1 (lemma)**.** *The inclusion of in-context prompt* $X' = [p'_1, ..., p'_M]$ *is equivalent to updating all reference functions* $\mathcal{F}^{(i)}([X], .) = W_{[}^{(i)}X](.)$ *with a single step of gradient descent with learning rate* η *and training set* $\{x_i, y_i\}_{i=1}^{M}$*. For every* $i \in \{1, ..., M\}$*,* $x_i = \phi(W_K p'_i)$ *and* $y_i = \frac{M}{\eta} W_V p'_i + \mathcal{F}_0^{(i)}([X], \phi(W_K p'_i))$

Note that in this case, the separate functions $\mathcal{F}_0^{(i)}([X], \mathbf{q})$ are not independent but all related to one another. Hence this result is largely theoretical in nature, but does extend our intuitive results from previous sections to the autoregressive setting.

7 Conclusion

In this work, we demonstrate that, for all linearized attention layers, ICL is equivalent to a single step of gradient descent with a specific training set. This is shown for transformers both with and without causal masking. We further extend this statement to multi-layer transformers, showing that they are similar to past greedy layer-wise training algorithms. This explains to an extent some existing characteristics of ICL as well as opening a potential avenue for ICL to be studied in greater theoretical depth in the future. By taking into account past tendencies of greedy layer-wise training algorithms, it could be possible to enhance current ICL methods further opening a whole new dimension of possibilities.

References

Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006. URL `https://proceedings.neurips.cc/paper_files/paper/2006/file/5da713a690c067105aeb2fae32403405-Paper.pdf`.

Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

Kazuki Irie, Róbert Csordás, and Jürgen Schmidhuber. The dual form of neural networks revisited: Connecting test time predictions to training patterns via spotlights of attention. In *International Conference on Machine Learning*, pages 9639–9659. PMLR, 2022.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020.

Jiachen Lu, Jinghan Yao, Junge Zhang, Xiatian Zhu, Hang Xu, Weiguo Gao, Chunjing Xu, Tao Xiang, and Li Zhang. Soft: Softmax-free transformer with linear complexity. *Advances in Neural Information Processing Systems*, 34:21297–21309, 2021.

Arvind Mahankali, Tatsunori B. Hashimoto, and Tengyu Ma. One step of gradient descent is provably the optimal in-context learner with one layer of linear self-attention, 2023. URL `https://arxiv.org/abs/2307.03576`.

Ruifeng Ren and Yong Liu. In-context learning with transformer is really equivalent to a contrastive learning pattern, 2023.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners, 2022. URL `https://arxiv.org/abs/2109.01652`.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. Instruction tuning for large language models: A survey, 2024. URL `https://arxiv.org/abs/2308.10792`.

8 Appendix

8.1 Proof of Theorem 3.1

Proof. Assume such a set $\{x_i, y_i\}$ and corresponding E and X exist such that the two systems are equal. By Proposition 2.1 and equation (6) we have:

$$W[X', X]\mathbf{q} = W_X\mathbf{q} + LinAttn(\frac{\eta}{m}E, X, \mathbf{q})$$

This implies:

$$W_{[X]}\mathbf{q} + LinAttn(W_V X', W_K X', \mathbf{q} = LinAttn(\frac{\eta}{m}E, X, \mathbf{q})$$

To enforce such an equality we need $W_K X' = X$. This shows that for all i:

$$x_i = W_K p_i'$$

Hence substituting in x_i we have:

$$W_V X' = \frac{\eta}{m}E = \frac{\eta}{m}(W_[X]W_K X' - Y)$$

This implies:

$$Y = W_{[X]}W_K X' - \frac{m}{\eta}W_V X'$$

$$y_i = W_{[X]}W_K p_i' + \frac{m}{\eta}W_V p_i'$$

$\square$