In-Context Deep Learning via Transformer Models

Weimin Wu[*†] **Maojiang Su**[*†] **Jerry Yao-Chieh Hu**[*†] **Zhao Song**[§] **Han Liu**[†]

{wwm,maojiangsu2030,jhu}@u.northwestern.edu; magic.linuxkde@gmail.com; hanliu@northwestern.edu

[†]Northwestern University; [§]University of California Berkeley

Summary: Goals and Contributions

Goals: Investigate the transformer's capability for in-context learning (ICL) to simulate the training process of deep models.

- An explicit construction of a $(2N+4)L$-layer transformer capable of simulating L gradient descents of an N-layer ReLU network by ICL.
- The theoretical guarantees for the approximation within any given error and the convergence of the ICL gradient descent.
- Extend the analysis to $\mathrm{Softmax}$-based transformers.
- Validate the findings for multiple-layer neural networks.

Preliminary: Transformer

Assume the input sequence is $H \in \mathbb{R}^{D\times n}$.

ReLU-Attention Layer: An M-head ReLU-attention layer with parameters $\theta = \{Q_m, K_m, V_m\}_{m\in[M]}$ outputs

$$\mathrm{Attn}_\theta(H) := H + \frac{1}{n}\sum_{m=1}^{M}(V_m H)\cdot\sigma((Q_m H)^\top (K_m H)),$$

where $Q_m, K_m, V_m \in \mathbb{R}^{D\times D}$ and $\sigma(\cdot)$ is ReLU activation function.

MLP Layer: An d'-hidden dimensions MLP layer with parameters $\theta = (W_1, W_2)$ outputs

$$\mathrm{MLP}_\theta(H) := H + W_2\sigma(W_1 H)$$

, where $W_1 \in \mathbb{R}^{d'\times D}$, $W_2 \in \mathbb{R}^{D\times d'}$ and $\sigma(\cdot): \mathbb{R}\to\mathbb{R}$ is element-wise ReLU activation function.

Transformer: An L-layer transformer with parameters $\theta = \{\theta_{\mathrm{Attn}}, \theta_{\mathrm{MLP}}\}$ outputs

$$\mathrm{TF}_\theta^{\mathrm{L}}(H) := \mathrm{MLP}_{\theta^{(L)}_{\mathrm{mlp}}} \circ \mathrm{Attn}_{\theta^{(L)}_{\mathrm{attn}}} \ldots \mathrm{MLP}_{\theta^{(1)}_{\mathrm{mlp}}} \circ \mathrm{Attn}_{\theta^{(1)}_{\mathrm{attn}}}(H),$$

where $\theta = \{\theta_{\mathrm{Attn}}, \theta_{\mathrm{MLP}}\}$ consists of Attention layers $\theta_{\mathrm{Attn}} = \{(Q^l_m, K^l_m, V^l_m)\}_{l\in[L], m\in[M^l]}$ and MLP layers $\theta_{\mathrm{MLP}} = \{(W^l_1, W^l_2)\}_{l\in[L]}$.

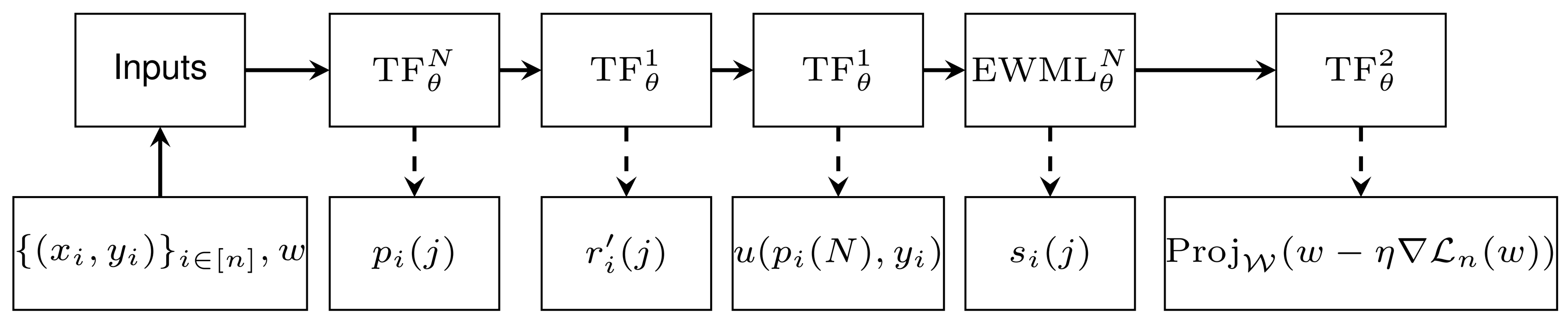
Problem Setting: In-Context Gradient Descent (ICGD)

Let $\epsilon > 0$ and $L \geq 1$. Consider a model $f(w,x): \mathbb{R}^{D_w}\times\mathbb{R}^d \to \mathbb{R}^d$ parameterized by $w \in \mathbb{R}^{D_w}$. Given a dataset $\mathcal{D}_n := \{(x_i, y_i)\}_{i\in[n]} \overset{\text{iid}}{\sim} \mathbb{P}$ with $(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R}^d$, define the empirical risk function:

$$\mathcal{L}_n(w) := \frac{1}{2n}\sum_{i=1}^{n}\ell(f(w,x_i), y_i),$$

where $\ell: \mathbb{R}^d\times\mathbb{R}^d \to \mathbb{R}$ is a loss function. Let $\mathcal{W} \subseteq \mathbb{R}^{D_w}$ be a closed domain, and $\mathrm{Proj}_{\mathcal{W}}$ denote the projection onto $\mathcal{W}$. The problem of "ICGD on model $f(w,\cdot)$" is to find a transformer $\mathcal{T}$ with L blocks, each approximating one step of gradient descent using T layers. For any input $H^{(0)} \in \mathbb{R}^{D\times(n+1)}$, the transformer approximates L gradient descents. Specifically, we consider $f(w,x)$ as **N-layer neural networks**.

One Step ICGD with $(2N+4)$-layer Transformer



This illustration presents the backpropagation process within an ICGD in a transformer model with $2N+4$ layers. It simulates a single gradient descent step for an N-layer neural network, trained with loss $\mathcal{L}_n$ and datasets $\{(x_i, y_i)\}_{i\in[n]}$. The term $p_i(j)$ denotes the output after the j-th layer for input x_i. The terms $r'_i(j)$, $u(p_i(N), y_i)$, and $s_i(j)$ are intermediate gradient terms of gradient $\nabla\mathcal{L}_n(w)$ from the chain rule. The expression $\mathrm{Proj}_{\mathcal{W}}(w - \eta\nabla\mathcal{L}_n(w))$ shows one gradient descent step. Here, η is the learning rate, and $\mathcal{W}$ is the bounded domain for the N-layer NN.

Results 1: ICGD on NN with ReLU-Transformer

$(2N+4)L$-layer ReLU-transformer is capable of simulating L gradient descents of an N-layer NN by ICL under any given approximation error.

Theorem 1. Fix any $B_v, \eta, \epsilon > 0, L \geq 1$. For any input sequences, there exist upper bounds B_x, B_y such that for any $i \in [n]$, $\|y_i\|_2 \leq B_y$, $\|x_i\|_2 \leq B_x$. Assume functions $r(t)$, $r'(t)$ and $u(t,y)[k]$ are $L_r, L_{r'}, L_l$-Lipschitz continuous. Suppose $\mathcal{W}$ is a closed domain such that for any $j \in [N-1]$ and $k \in [K]$,

$$\mathcal{W} \subset \{w = [v_{j_k}] \in \mathbb{R}^{D_N} : \|v_{j_k}\|_2 \leq B_v\},$$

and $\mathrm{Proj}_{\mathcal{W}}$ project w into bounded domain $\mathcal{W}$. Assume $\mathrm{Proj}_{\mathcal{W}} = \mathrm{MLP}_\theta$ for some MLP layer with hidden dimension D_w parameters $\|\theta\| \leq C_w$. If functions $r(t)$, $r'(t)$ and $u(t,y)[k]$ are C^4-smoothness, then for any $\epsilon > 0$, there exists a transformer model NN_θ with $(2N+4)L$ hidden layers consists of L neural network blocks $\mathrm{TF}_\theta^{N+2} \circ \mathrm{EWML}_\theta^N \circ \mathrm{TF}_\theta^2$,

$$\mathrm{NN}_\theta := \mathrm{TF}_\theta^{N+2} \circ \mathrm{EWML}_\theta^N \circ \mathrm{TF}_\theta^2,$$

such that the heads number M^l, parameter dimensions D^l, and the parameter norms B_{θ^l} suffice

$$\max_{l\in[(2N+4)L]} M^l \leq \widetilde{O}(\epsilon^{-2}), \quad \max_{l\in[(2N+4)L]} D^l \leq O(NK^2) + D_w,$$

$$\max_{l\in[(2N+4)L]} B_{\theta^l} \leq O(\eta) + C_w + 1,$$

where $\widetilde{O}(\cdot)$ hides the constants that depend on d, K, N, the radius parameters B_x, B_y, B_v and the smoothness of r and ℓ. And this neural network such that for any input sequences $H^{(0)}$, $\mathrm{NN}_\theta(H^{(0)})$ implements L steps in-context gradient descent: For every $l \in [L]$, the $(2N+4)l$-th layer outputs $h_i^{((2N+4)l)} = [x_i; y_i; \bar{w}^{(l)}; \mathbf{0}; 1; t_i]$ for every $i \in [n+1]$, and approximation gradients $\bar{w}^{(l)}$ such that

$$\bar{w}^{(l)} = \mathrm{Proj}_{\mathcal{W}}(\bar{w}^{(l-1)} - \eta\nabla\mathcal{L}_n(\bar{w}^{(l-1)}) + \epsilon^{(l-1)}),$$

where $\bar{w}^{(0)} = \mathbf{0}$, and $\|\epsilon^{(l-1)}\|_2 \leq \eta\epsilon$ is an error term.

Result 2: ICGD on NN with $\mathrm{Softmax}$-Transformer

There exists a $\mathrm{Softmax}$-transformer to simulate L gradient descents of an N-layer NN by ICL under any given approximation precision.

Theorem 2. Fix any $B_w, \eta, \epsilon > 0, L \geq 1$. For any input sequences, their exist upper bounds B_x, B_y such that for any $i \in [n]$, $\|y_i\|_{\max} \leq B_y$, $\|x_i\|_{\max} \leq B_x$. Suppose $\mathcal{W}$ is a closed domain such that $\|w\|_{\max} \leq B_w$ and $\mathrm{Proj}_{\mathcal{W}}$ project w into bounded domain $\mathcal{W}$. Assume $\mathrm{Proj}_{\mathcal{W}} = \mathrm{MLP}_\theta$ for some MLP layer. Define $l(w, x_i, y_i)$ as a loss function with L-Lipschitz gradient. Let $\mathcal{L}_n(w) = \frac{1}{n}\sum_{i=1}^{n}\ell(w, x_i, y_i)$ denote the empirical loss function, then there exists a $\mathrm{Softmax}$-transformer NN_θ, such that for any input sequences $H^{(0)}$, $\mathrm{NN}_\theta(H^{(0)})$ implements L steps in-context gradient descent on $\mathcal{L}_n(w)$: For every $l \in [L]$, the $4l$-th layer outputs $h_i^{(4l)} = [x_i; y_i; \bar{w}^{(l)}; \mathbf{0}; 1; t_i]$ for every $i \in [n+1]$, and approximation gradients $\bar{w}^{(l)}$ with $\bar{w}^{(0)} = \mathbf{0}$ such that

$$\bar{w}^{(l)} = \mathrm{Proj}_{\mathcal{W}}(\bar{w}^{(l-1)} - \eta\nabla\mathcal{L}_n(\bar{w}^{(l-1)}) + \epsilon^{(l-1)}),$$

where $\|\epsilon^{(l-1)}\|_2 \leq \eta\epsilon$ is an error term.

Numerical Studies

Performance of ICL in ReLU-Transformer and Softmax-Transformer: ICL learns 6-layer NN and achieves R-squared values comparable to those from training with prompt samples.