In-context Learning on Function Classes Unveiled for Transformers

Zhijie Wang **Bo Jiang** **Shuai Li**

July 6, 2024

Transformers

A transformer [Vaswani et al. 2017] layer contains two sub-layers: **Attention layer** and **MLP layer**.

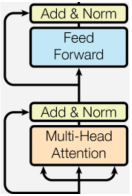

Figure. A transformer layer

Definition 0.1

Attention layer *with parameters* $\boldsymbol{\theta} = \{\mathbf{V}_m, \mathbf{Q}_m, \mathbf{K}_m\}_{m\in[M]}$ *and input matrix* $\mathbf{H}$*:*

$$\mathrm{Attn}_{\boldsymbol{\theta}}(\mathbf{H}) = \mathbf{H} + \frac{1}{\mathrm{N}}\sum_{\mathrm{m}=1}^{\mathrm{M}}(\mathbf{V}_{\mathrm{m}}\mathbf{H}) \times \sigma((\mathbf{Q}_{\mathrm{m}}\mathbf{H})^{\top}(\mathbf{K}_{\mathrm{m}}\mathbf{H})).$$

Definition 0.2

MLP layer *with parameters* $\boldsymbol{\theta} = (\mathbf{W}_1, \mathbf{W}_2)$ *with input* $\mathbf{H}$ *:*

$$\mathrm{MLP}_{\boldsymbol{\theta}}(\mathbf{H}) = \mathbf{H} + \mathbf{W}_2\sigma(\mathbf{W}_1\mathbf{H}).$$

Neural Networks

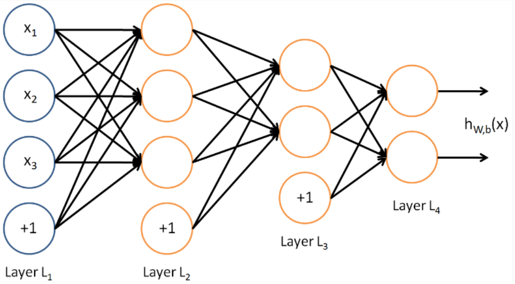

Figure. Neural Networks

Definition 0.3

*The output of an n-**layer neural network** on the input $x \in \mathbb{R}^d$:*

$$\mathrm{pred}_n(\mathbf{w}, \mathbf{x}) \triangleq h_{\mathbf{w}}(\mathbf{x}) = \mathbf{W}^{(n)}(r(\mathbf{W}^{(n-1)}(r(\cdots r(\mathbf{W}^{(1)}\mathbf{x})))))$$

In-context Learning(ICL)

- Pre-train a model $h = f(H; \hat{\theta})$
- Take $H = [x_1, y_1, x_2, y_2, \cdots, x_N, y_N, x_{N+1}]$ as the input
- Prediction $\hat{y}_{N+1} = f(H; \hat{\theta}) \approx y_{N+1}$

apple->fruit, carpet->furniture, bird->

bird -> animal

Review: This movie sucks. Sentiment: negative. Review: I love this movie. Sentiment:

positive.

IN-CONTEXT GRADIENT DESCENT ON NEURAL NETWORKS (NNS)

Transformers in-context learn NNs $\longleftrightarrow$ Gradient Descent on NN parameters

Theorem 1

For a_n-layer transformers: input data $(\mathcal{D}, \mathbf{x}_{N+1})$ and NN (width K, depth n) parameters $\mathbf{w}$:

$$\mathbf{w}_\eta^+ = \mathrm{Proj}_{\mathcal{W}}\left(\mathbf{w} - \eta(\nabla \mathrm{L_N}(\mathbf{w}) + \epsilon(\mathbf{w}))\right), \quad \|\epsilon(\mathbf{w})\|_2 \leq \eta\epsilon.$$

Here $\mathcal{O}(a_n) = \mathcal{O}(n) + \mathcal{O}(a_{n-1})$. Number of heads and hidden dimension:

$$\max_{l \in [a_n]} M^{(l)} \leq \mathcal{O}(nK^2\epsilon^{-2}), \quad \max_{l \in [a_n]} D^{(l)} \leq \mathcal{O}(nK^2\epsilon^{-2}).$$

Remark: $a_n = \mathcal{O}(n^2)$

ICL on Function Classes

Natural to connect transformer with neural networks approximation.

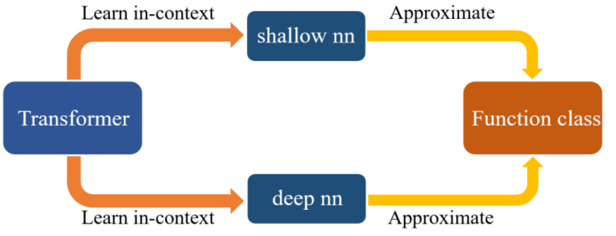

Figure. Bridging Transformers & Function Classes

- classification function: $f(\mathbf{x}) = \mathbf{1}(\|A\mathbf{x} + \mathbf{b}\| \leq r)$
- linear function: $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle$
- nonlinear smooth function: $f \in C^2$

Classification

Transformer learns $f(\mathbf{x}) = \mathbf{1}(\|A\mathbf{x} + \mathbf{b}\| \leq r)$ in-context.

Theorem 2

There exists a cL-layer transformer ($L \sim \mathcal{O}(\log(1/\epsilon))$) with

$$\max_{l \in [cL]} M^{(l)} \leq \mathcal{O}(\delta^{-1}\epsilon^{-2}), \quad \max_{l \in [cL]} D^{(l)} \leq \mathcal{O}(\delta^{-1}\epsilon^{-2}),$$

such that the prediction of the transformer $\hat{y}_{N+1}$ satisfies

$$|\hat{y}_{N+1} - y_{N+1}| \leq \mathcal{O}(\epsilon + \delta).$$

Remark: Here we use a **3-layer NN** as a bridge. Using a **2-layer NN** would cause the upper bound to be **exponential** in $\delta^{-1/4}$.

Linear function

Transformer learns $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle$ in-context.

Theorem 3

There exists a $2L$-layer transformer with

$$\max_{l \in [2L]} M^{(l)} \leq \mathcal{O}(\epsilon^{-2}), \quad \max_{l \in [2L]} D^{(l)} \leq \mathcal{O}(\epsilon^{-2}),$$

such that the prediction of the transformer $\hat{y}_{N+1}$ satisfies

$$|\hat{y}_{N+1} - y_{N+1}| \leq \mathcal{O}(\epsilon).$$

Nonlinear smooth function

Transformer learns $f \in \mathcal{W}^{n,\infty}([0,1]^d)$ in-context.

Theorem 4

There exists a $\mathcal{O}(\ln^2(1/\delta)L)$-layer transformer with

$$\max_{l \in [k_\delta L]} M^{(l)} \leq \mathcal{O}(\delta^{-2d/n}\epsilon^{-2}), \quad \max_{l \in [k_\epsilon L]} D^{(l)} \leq \mathcal{O}(\delta^{-2d/n}\epsilon^{-2}),$$

such that the prediction of the transformer $\hat{y}_{N+1}$ satisfies

$$|\hat{y}_{N+1} - y_{N+1}| \leq \mathcal{O}(\epsilon + \delta).$$

ALGORITHM SELECTION

A transformer pretrained on a mixture of **linear regression** and **classification** dataset.

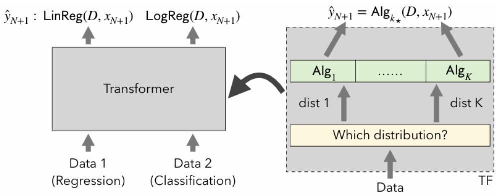

Figure. Algorithm-Selection [Bai et al. 2024]

Theorem 5

There exists a $(c+2)L+1$*-layer transformer with*

$$\max_{l \in [(c+2)L]} M^{(l)} \leq \mathcal{O}(\delta^{-1}\epsilon^{-2}), \quad \max_{l \in [(c+2)L]} D^{(l)} \leq \mathcal{O}(\delta^{-1}\epsilon^{-2}),$$

such that its output satisfies

$$|\hat{y} - y_{N+1}| \leq \mathcal{O}(\epsilon + \delta).$$

Numerical Experiments

- **Quadratic Regression** ($d = 20$): $\mathbf{z} = (\mathbf{x}, y) \in \mathbb{R}^d \times \mathbb{R}$, $\mathbf{x} \sim \mathcal{N}(0, 1)$, $y = \mathbf{w}^\top \mathbf{x}^{\odot 2}$, $\mathbf{w} \sim \mathcal{N}(0, \sigma)$.
- **Three-layer Neural Network** ($K = 50$, $d = 20$): $\mathbf{z} = (\mathbf{x}, y) \in \mathbb{R}^d \times \mathbb{R}$, $\mathbf{x} \sim \mathcal{N}(0, 1)$, $y = \mathbf{W}^{(3)}(r(\mathbf{W}^{(2)}((r(\mathbf{W}^{(1)}\mathbf{x}))))$, $\mathbf{W}_{ij} \sim \mathcal{N}(0, \sigma)$.

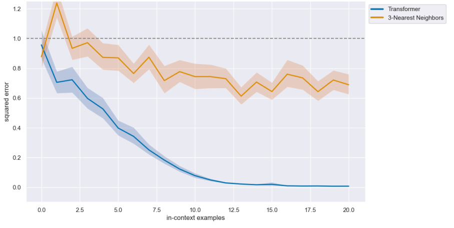

Figure. quadratic function

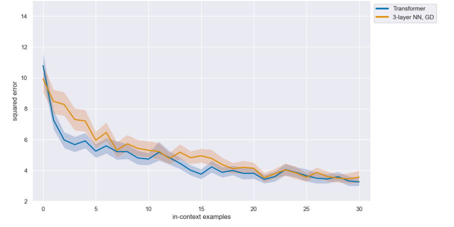

Figure. 3 layer neural network

References

Bai, Yu et al. (2024). **"Transformers as statisticians: Provable in-context learning with in-context algorithm selection".** In: *Advances in neural information processing systems* 36.

Vaswani, Ashish et al. (2017). **"Attention is all you need".** In: *Advances in neural information processing systems* 30.