On the Convergence of Encoder-only Shallow Transformers

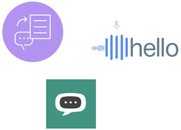
Fanghui Liu

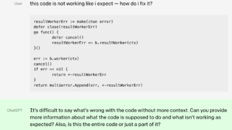
Department of Computer Science, University of Warwick, UK
Centre for Discrete Mathematics and its Applications (DIMAP), Warwick

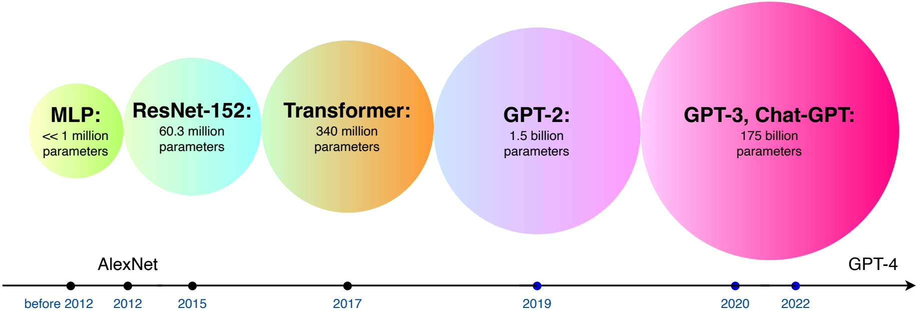
Based on joint work with
[Yongtao Wu (EPFL), Fanghui Liu, Grigorios Chrysos (UW-Madison), Volkan Cevher (EPFL)]

at MILD Seminar, University of British Columbia

Over-parameterization: more parameters than training data







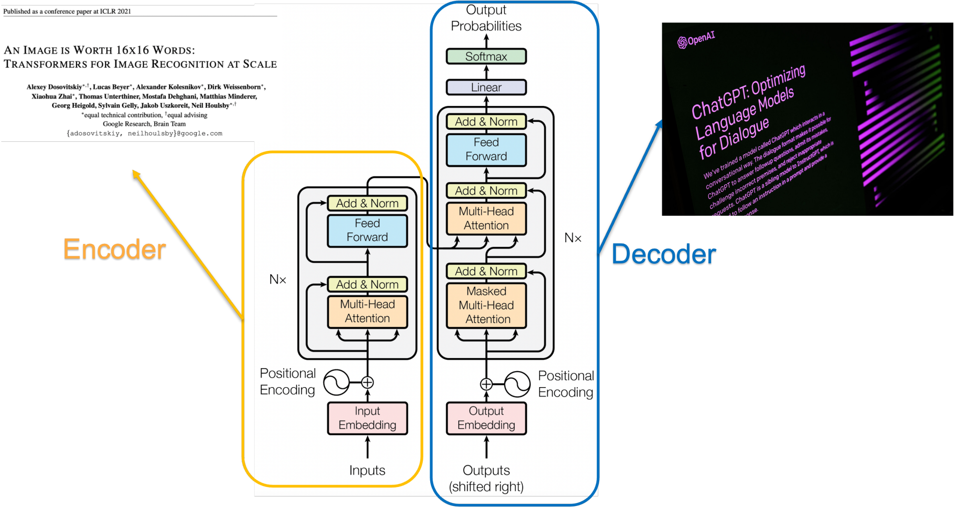


Figure: **Left**: Vision Transformer(ViT) [1], based on the encoder of Transformer. **Middle**: Original Transformer [2], with encoder and decoder. **Right**: ChatGPT, based on the decoder of Transformer.

Self-attention

- input $\boldsymbol{X} \in \mathbb{R}^{d_s \times d}$
- σ_s: soft-max (row-wise)
- $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V \in \mathbb{R}^{d_m \times d}$

- d_s: number of tokens
- d: the feature dimension of each token
- d_m: width

Self-attention

- input $\boldsymbol{X} \in \mathbb{R}^{d_s \times d}$
- σ_s: soft-max (row-wise)
- $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V \in \mathbb{R}^{d_m \times d}$
- d_s: number of tokens
- d: the feature dimension of each token
- d_m: width

$$\text{Self-attention}(\boldsymbol{X}) \triangleq \text{Softmax}\left(\tau_0 (\boldsymbol{X}\boldsymbol{W}_Q^\top)\left(\boldsymbol{X}\boldsymbol{W}_K^\top\right)^\top\right)\left(\boldsymbol{X}\boldsymbol{W}_V^\top\right) = \sigma_s\left(\tau_0 \boldsymbol{X}\boldsymbol{W}_Q^\top \boldsymbol{W}_K \boldsymbol{X}^\top\right)\left(\boldsymbol{X}\boldsymbol{W}_V^\top\right).$$

Self-attention

- input $\boldsymbol{X} \in \mathbb{R}^{d_s \times d}$
- σ_s: soft-max (row-wise)
- $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V \in \mathbb{R}^{d_m \times d}$
- d_s: number of tokens
- d: the feature dimension of each token
- d_m: width

$$\text{Self-attention}(\boldsymbol{X}) \triangleq \text{Softmax}\left(\tau_0 (\boldsymbol{X}\boldsymbol{W}_Q^\top)\left(\boldsymbol{X}\boldsymbol{W}_K^\top\right)^\top\right)\left(\boldsymbol{X}\boldsymbol{W}_V^\top\right) = \sigma_s\left(\tau_0 \boldsymbol{X}\boldsymbol{W}_Q^\top \boldsymbol{W}_K \boldsymbol{X}^\top\right)\left(\boldsymbol{X}\boldsymbol{W}_V^\top\right).$$

$$\text{input of softmax: } [\tau_0 \boldsymbol{W}_Q^\top \boldsymbol{W}_K]_{ij} = \tau_0 \sum_{k=1}^{d_m} [\boldsymbol{W}_Q^\top]_{ik} [\boldsymbol{W}_K]_{kj}$$

Self-attention

- input $\boldsymbol{X} \in \mathbb{R}^{d_s \times d}$
- σ_s: soft-max (row-wise)
- $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V \in \mathbb{R}^{d_m \times d}$
- d_s: number of tokens
- d: the feature dimension of each token
- d_m: width

$$\text{Self-attention}(\boldsymbol{X}) \triangleq \text{Softmax}\left(\tau_0(\boldsymbol{X}\boldsymbol{W}_Q^\top)\left(\boldsymbol{X}\boldsymbol{W}_K^\top\right)^\top\right)\left(\boldsymbol{X}\boldsymbol{W}_V^\top\right) = \sigma_s\left(\tau_0 \boldsymbol{X}\boldsymbol{W}_Q^\top \boldsymbol{W}_K \boldsymbol{X}^\top\right)\left(\boldsymbol{X}\boldsymbol{W}_V^\top\right).$$

$$\text{input of softmax: } [\tau_0 \boldsymbol{W}_Q^\top \boldsymbol{W}_K]_{ij} = \tau_0 \sum_{k=1}^{d_m} [\boldsymbol{W}_Q^\top]_{ik} [\boldsymbol{W}_K]_{kj}$$

◦ scaling schemes given by $\boldsymbol{W}_Q, \boldsymbol{W}_K$ initialized by standard Gaussian

Self-attention

- input $\boldsymbol{X} \in \mathbb{R}^{d_s \times d}$
- σ_s: soft-max (row-wise)
- $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V \in \mathbb{R}^{d_m \times d}$
- d_s: number of tokens
- d: the feature dimension of each token
- d_m: width

$$\text{Self-attention}(\boldsymbol{X}) \triangleq \text{Softmax}\left(\tau_0 (\boldsymbol{X}\boldsymbol{W}_Q^\top) \left(\boldsymbol{X}\boldsymbol{W}_K^\top\right)^\top\right)\left(\boldsymbol{X}\boldsymbol{W}_V^\top\right) = \sigma_s\left(\tau_0 \boldsymbol{X}\boldsymbol{W}_Q^\top \boldsymbol{W}_K \boldsymbol{X}^\top\right)\left(\boldsymbol{X}\boldsymbol{W}_V^\top\right).$$

$$\text{input of softmax: } [\tau_0 \boldsymbol{W}_Q^\top \boldsymbol{W}_K]_{ij} = \tau_0 \sum_{k=1}^{d_m} [\boldsymbol{W}_Q^\top]_{ik} [\boldsymbol{W}_K]_{kj}$$

◦ scaling schemes given by $\boldsymbol{W}_Q, \boldsymbol{W}_K$ initialized by standard Gaussian

- $\tau_0 = d_m^{-1/2}$ in the original Transformer [2]:

$$[\tau_0 \boldsymbol{W}_Q^\top \boldsymbol{W}_K]_{ij} \text{ has zero-mean and unit variance} \quad \forall i, j \in [d]$$

Self-attention

- input $\boldsymbol{X} \in \mathbb{R}^{d_s \times d}$
- σ_s: soft-max (row-wise)
- $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V \in \mathbb{R}^{d_m \times d}$

- d_s: number of tokens
- d: the feature dimension of each token
- d_m: width

$$\text{Self-attention}(\boldsymbol{X}) \triangleq \text{Softmax}\left(\tau_0 (\boldsymbol{X}\boldsymbol{W}_Q^\top) \left(\boldsymbol{X}\boldsymbol{W}_K^\top\right)^\top\right) \left(\boldsymbol{X}\boldsymbol{W}_V^\top\right) = \sigma_s \left(\tau_0 \boldsymbol{X}\boldsymbol{W}_Q^\top \boldsymbol{W}_K \boldsymbol{X}^\top\right) \left(\boldsymbol{X}\boldsymbol{W}_V^\top\right).$$

$$\text{input of softmax: } [\tau_0 \boldsymbol{W}_Q^\top \boldsymbol{W}_K]_{ij} = \tau_0 \sum_{k=1}^{d_m} [\boldsymbol{W}_Q^\top]_{ik} [\boldsymbol{W}_K]_{kj}$$

○ scaling schemes given by $\boldsymbol{W}_Q, \boldsymbol{W}_K$ initialized by standard Gaussian

- $\tau_0 = d_m^{-1/2}$ in the original Transformer [2]:

$$[\tau_0 \boldsymbol{W}_Q^\top \boldsymbol{W}_K]_{ij} \text{ has zero-mean and unit variance } \quad \forall i, j \in [d]$$

- $\tau_0 = d_m^{-1}$: from the neural tangent kernel (NTK) analysis [3] for $d_m \to \infty$.

$$\lim_{d_m \to \infty} \tau_0 [\boldsymbol{W}_Q^\top \boldsymbol{W}_K]^{(ij)} = \lim_{d_m \to \infty} \frac{1}{d_m} \sum_{k=1}^{d_m} [\boldsymbol{W}_Q^\top]_{ik} [\boldsymbol{W}_K]_{kj} = 0\,.$$

Self-attention

- input $\boldsymbol{X} \in \mathbb{R}^{d_s \times d}$
- σ_s: soft-max (row-wise)
- $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V \in \mathbb{R}^{d_m \times d}$
- d_s: number of tokens
- d: the feature dimension of each token
- d_m: width

$$\text{Self-attention}(\boldsymbol{X}) \triangleq \text{Softmax}\left(\tau_0(\boldsymbol{X}\boldsymbol{W}_Q^\top)\left(\boldsymbol{X}\boldsymbol{W}_K^\top\right)^\top\right)\left(\boldsymbol{X}\boldsymbol{W}_V^\top\right) = \sigma_s\left(\tau_0 \boldsymbol{X}\boldsymbol{W}_Q^\top \boldsymbol{W}_K \boldsymbol{X}^\top\right)\left(\boldsymbol{X}\boldsymbol{W}_V^\top\right).$$

$$\text{input of softmax: } [\tau_0 \boldsymbol{W}_Q^\top \boldsymbol{W}_K]_{ij} = \tau_0 \sum_{k=1}^{d_m} [\boldsymbol{W}_Q^\top]_{ik} [\boldsymbol{W}_K]_{kj}$$

◦ scaling schemes given by $\boldsymbol{W}_Q, \boldsymbol{W}_K$ initialized by standard Gaussian

- $\tau_0 = d_m^{-1/2}$ in the original Transformer [2]:

$$[\tau_0 \boldsymbol{W}_Q^\top \boldsymbol{W}_K]_{ij} \text{ has zero-mean and unit variance } \quad \forall i, j \in [d]$$

- $\tau_0 = d_m^{-1}$: from the neural tangent kernel (NTK) analysis [3] for $d_m \to \infty$.

$$\lim_{d_m \to \infty} \tau_0 [\boldsymbol{W}_Q^\top \boldsymbol{W}_K]^{(ij)} = \lim_{d_m \to \infty} \frac{1}{d_m} \sum_{k=1}^{d_m} [\boldsymbol{W}_Q^\top]_{ik} [\boldsymbol{W}_K]_{kj} = 0\,.$$

Softmax becomes a pooling layer!

Previous attempts on scaling in theory

$$[\tau_0 \boldsymbol{W}_Q^\top \boldsymbol{W}_K]_{ij}$$

- scaling schemes given by $\boldsymbol{W}_Q, \boldsymbol{W}_K$ initialized by standard Gaussian
  - $\tau_0 = d_m^{-1/2}$ in the original Transformer [2]:

$$[\tau_0 \boldsymbol{W}_Q^\top \boldsymbol{W}_K]_{ij} \text{ has zero-mean and unit variance} \quad \forall i, j \in [d]$$

Previous attempts on scaling in theory

$$[\tau_0 \boldsymbol{W}_Q^\top \boldsymbol{W}_K]_{ij}$$

- ∘ scaling schemes given by $\boldsymbol{W}_Q, \boldsymbol{W}_K$ initialized by standard Gaussian
  - $\tau_0 = d_m^{-1/2}$ in the original Transformer [2]:

$$[\tau_0 \boldsymbol{W}_Q^\top \boldsymbol{W}_K]_{ij} \text{ has zero-mean and unit variance } \quad \forall i,j \in [d]$$

$$[4]: \underbrace{\cancel{\text{Softmax}}}_{\text{ReLU}}\left(\tau_0 (\boldsymbol{X}\boldsymbol{W}_Q^\top)\left(\boldsymbol{X}\boldsymbol{W}_K^\top\right)^\top\right)\left(\boldsymbol{X}\boldsymbol{W}_V^\top\right)$$

Previous attempts on scaling in theory

$$[\tau_0 \boldsymbol{W}_Q^\top \boldsymbol{W}_K]_{ij}$$

○ scaling schemes given by $\boldsymbol{W}_Q, \boldsymbol{W}_K$ initialized by standard Gaussian

- $\tau_0 = d_m^{-1/2}$ in the original Transformer [2]:

$$[\tau_0 \boldsymbol{W}_Q^\top \boldsymbol{W}_K]_{ij} \text{ has zero-mean and unit variance } \quad \forall i, j \in [d]$$

$$[4]: \quad \underbrace{\cancel{\text{Softmax}}}_{\text{ReLU}}\left(\tau_0 (\boldsymbol{X}\boldsymbol{W}_Q^\top)\left(\boldsymbol{X}\boldsymbol{W}_K^\top\right)^\top\right)\left(\boldsymbol{X}\boldsymbol{W}_V^\top\right)$$

- $\tau_0 = d_m^{-1}$: from the neural tangent kernel (NTK) analysis [3] for $d_m \to \infty$.

$$\lim_{d_m \to \infty} \tau_0 [\boldsymbol{W}_Q^\top \boldsymbol{W}_K]^{(ij)} = \lim_{d_m \to \infty} \frac{1}{d_m} \sum_{k=1}^{d_m} [\boldsymbol{W}_Q^\top]_{ik} [\boldsymbol{W}_K]_{kj} = 0\,.$$

[5] : setting $\boldsymbol{W}_Q = \boldsymbol{W}_K$

Question

How can we do analysis of Transformers under a realistic setting?

Question

How can we do analysis of Transformers under a realistic setting?

even though

- a shallow Transformer
- an encoder-only shallow Transformer
- global convergence
- under the lazy training regime

Question

How can we do analysis of Transformers under a realistic setting?

even though

- a shallow Transformer
- an encoder-only shallow Transformer
- global convergence
- under the lazy training regime

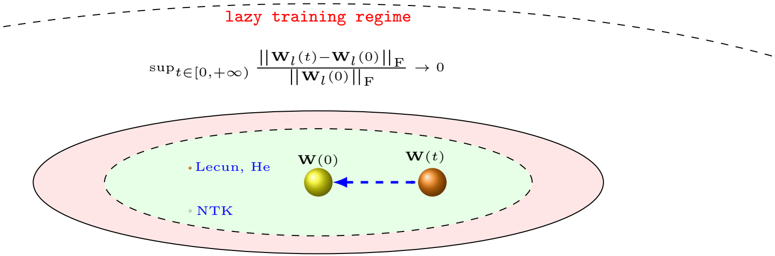


Figure: Training dynamics of two-layer ReLU NNs with infinite width under different initializations [3, 6, 7].

Problem setting: encoder-only shallow Transformer

$$\boldsymbol{A}_1 = \text{Self-attention}(\boldsymbol{X}) \triangleq \sigma_s\left(\tau_0(\boldsymbol{X}\boldsymbol{W}_Q^\top)\left(\boldsymbol{X}\boldsymbol{W}_K^\top\right)^\top\right)\left(\boldsymbol{X}\boldsymbol{W}_V^\top\right),$$

$$\boldsymbol{A}_2 = \tau_1\sigma_r(\boldsymbol{A}_1\boldsymbol{W}_H), \qquad \boldsymbol{a}_3 = \varphi(\boldsymbol{A}_2), \qquad f(\boldsymbol{X};\boldsymbol{\theta}) = \boldsymbol{a}_3^\top \boldsymbol{w}_O .$$

- Input: $\boldsymbol{X} \in \mathbb{R}^{d_s \times d}$ (d_s is the number of tokens and d is the feature dimension of each token)

Problem setting: encoder-only shallow Transformer

$$
\begin{aligned}
\boldsymbol{A}_1 &= \text{Self-attention}(\boldsymbol{X}) \triangleq \sigma_s\left(\tau_0(\boldsymbol{X}\boldsymbol{W}_Q^\top)\left(\boldsymbol{X}\boldsymbol{W}_K^\top\right)^\top\right)\left(\boldsymbol{X}\boldsymbol{W}_V^\top\right), \\
\boldsymbol{A}_2 &= \tau_1\sigma_r(\boldsymbol{A}_1\boldsymbol{W}_H), \qquad \boldsymbol{a}_3 = \varphi(\boldsymbol{A}_2), \qquad f(\boldsymbol{X};\boldsymbol{\theta}) = \boldsymbol{a}_3^\top \boldsymbol{w}_O\,.
\end{aligned}
$$

- Input: $\boldsymbol{X} \in \mathbb{R}^{d_s \times d}$ (d_s is the number of tokens and d is the feature dimension of each token)
- A *self-attention layer*: σ_s is the row-wise softmax function and the learnable parameters are $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V \in \mathbb{R}^{d_m \times d}$.

Problem setting: encoder-only shallow Transformer

$$
\begin{aligned}
\boldsymbol{A}_1 &= \text{Self-attention}(\boldsymbol{X}) \triangleq \sigma_s\left(\tau_0(\boldsymbol{X}\boldsymbol{W}_Q^\top)\left(\boldsymbol{X}\boldsymbol{W}_K^\top\right)^\top\right)\left(\boldsymbol{X}\boldsymbol{W}_V^\top\right), \\
\boldsymbol{A}_2 &= \tau_1\sigma_r(\boldsymbol{A}_1\boldsymbol{W}_H), \qquad \boldsymbol{a}_3 = \varphi(\boldsymbol{A}_2), \qquad f(\boldsymbol{X};\boldsymbol{\theta}) = \boldsymbol{a}_3^\top \boldsymbol{w}_O.
\end{aligned}
$$

- Input: $\boldsymbol{X} \in \mathbb{R}^{d_s \times d}$ (d_s is the number of tokens and d is the feature dimension of each token)
- A *self-attention layer*: σ_s is the row-wise softmax function and the learnable parameters are $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V \in \mathbb{R}^{d_m \times d}$.
- A *feed-forward ReLU* layer: σ_r is the ReLU activation function; the learnable parameter is $\boldsymbol{W}_H \in \mathbb{R}^{d_m \times d_m}$. We assume $\boldsymbol{W}_H = \boldsymbol{I}$.
- An *average pooling* layer: φ indicates the column-wise average pooling.
- An *output* layer with learnable parameter $\boldsymbol{w}_O \in \mathbb{R}^{d_m}$.

Problem setting: encoder-only shallow Transformer

$$\boldsymbol{A}_1 = \text{Self-attention}(\boldsymbol{X}) \triangleq \sigma_s \left(\tau_0 (\boldsymbol{X}\boldsymbol{W}_Q^\top) \left(\boldsymbol{X}\boldsymbol{W}_K^\top\right)^\top \right) \left(\boldsymbol{X}\boldsymbol{W}_V^\top\right),$$

$$\boldsymbol{A}_2 = \tau_1 \sigma_r (\boldsymbol{A}_1 \boldsymbol{W}_H), \qquad \boldsymbol{a}_3 = \varphi(\boldsymbol{A}_2), \qquad f(\boldsymbol{X};\boldsymbol{\theta}) = \boldsymbol{a}_3^\top \boldsymbol{w}_O .$$

- Input: $\boldsymbol{X} \in \mathbb{R}^{d_s \times d}$ (d_s is the number of tokens and d is the feature dimension of each token)
- A *self-attention layer*: σ_s is the row-wise softmax function and the learnable parameters are $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V \in \mathbb{R}^{d_m \times d}$.
- A *feed-forward ReLU* layer: σ_r is the ReLU activation function; the learnable parameter is $\boldsymbol{W}_H \in \mathbb{R}^{d_m \times d_m}$. We assume $\boldsymbol{W}_H = \boldsymbol{I}$.
- An *average pooling* layer: φ indicates the column-wise average pooling.
- An *output* layer with learnable parameter $\boldsymbol{w}_O \in \mathbb{R}^{d_m}$.

Initialization	η_O	η_V	η_Q	η_K	τ_1
LeCun	d_m^{-1}	d^{-1}	d^{-1}	d^{-1}	1
He	$2d_m^{-1}$	$2d^{-1}$	$2d^{-1}$	$2d^{-1}$	1
NTK	1	1	1	1	$d_m^{-1/2}$

Training by gradient descent

- data $\{(\boldsymbol{X}_i, y_i)\}_{i=1}^n$ with $\boldsymbol{y} = [y_1, y_2, \cdots, y_n]^\top$
- model output $\boldsymbol{f}(\boldsymbol{\theta}) := [f(\boldsymbol{X}_1; \boldsymbol{\theta}), f(\boldsymbol{X}_2; \boldsymbol{\theta}), \cdots, f(\boldsymbol{X}_n; \boldsymbol{\theta})]^\top$

$$\ell(\boldsymbol{\theta}) = \frac{1}{2} \|\boldsymbol{f}(\boldsymbol{\theta}) - \boldsymbol{y}\|_2^2$$

Training by gradient descent

- data $\{(\boldsymbol{X}_i, y_i)\}_{i=1}^n$ with $\boldsymbol{y} = [y_1, y_2, \cdots, y_n]^\top$
- model output $\boldsymbol{f}(\boldsymbol{\theta}) := [f(\boldsymbol{X}_1; \boldsymbol{\theta}), f(\boldsymbol{X}_2; \boldsymbol{\theta}), \cdots, f(\boldsymbol{X}_n; \boldsymbol{\theta})]^\top$

$$\ell(\boldsymbol{\theta}) = \frac{1}{2} \|\boldsymbol{f}(\boldsymbol{\theta}) - \boldsymbol{y}\|_2^2$$

Algorithm 2: Gradient descent training

Input: data $(\boldsymbol{X}_i, y_i)_{i=1}^n$, step size γ.
Initialize weights as follows:
$\boldsymbol{\theta}^0 := \{\boldsymbol{W}_Q^0, \boldsymbol{W}_K^0, \boldsymbol{W}_V^0, \boldsymbol{W}_O^0\}$.
for $t = 0$ **to** $t' - 1$ **do**
$\boldsymbol{W}_Q^{t+1} = \boldsymbol{W}_Q^t - \gamma \cdot \nabla_{\boldsymbol{W}_Q} \ell(\boldsymbol{\theta}^t)$, $\boldsymbol{W}_K^{t+1} = \boldsymbol{W}_K^t - \gamma \cdot \nabla_{\boldsymbol{W}_K} \ell(\boldsymbol{\theta}^t)$,
$\boldsymbol{W}_V^{t+1} = \boldsymbol{W}_V^t - \gamma \cdot \nabla_{\boldsymbol{W}_V} \ell(\boldsymbol{\theta}^t)$, $\boldsymbol{W}_O^{t+1} = \boldsymbol{W}_O^t - \gamma \cdot \nabla_{\boldsymbol{W}_O} \ell(\boldsymbol{\theta}^t)$.
end for
Output: the model based on $\boldsymbol{\theta}^{t'}$.

Assumptions on data

Assumption (Bounded data)

The input data is bounded $\|\boldsymbol{X}\|_{\mathrm{F}} \leq \sqrt{d_s} C_x$ *with some positive constant* C_x.

- The embedding of token can have a unit norm [8] independent of d.

Assumptions on data

Assumption (Bounded data)

The input data is bounded $\|\boldsymbol{X}\|_{\mathrm{F}} \leq \sqrt{d_s} C_x$ *with some positive constant* C_x*.*

- The embedding of token can have a unit norm [8] independent of d.

Assumption

The input data $\boldsymbol{X}$ *has full row rank.*

- language tasks: added with positional embedding
- ViT: image grouped by patch and can be uncorrelated

Assumptions on data

Assumption (Bounded data)

The input data is bounded $\|\boldsymbol{X}\|_{\mathrm{F}} \leq \sqrt{d_s} C_x$ *with some positive constant* C_x*.*

- ∘ The embedding of token can have a unit norm [8] independent of d.

Assumption

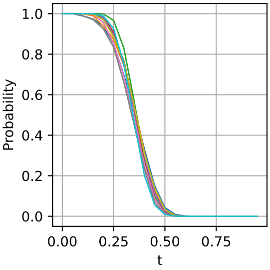
The input data $\boldsymbol{X}$ *has full row rank.*

- ∘ language tasks: added with positional embedding
- ∘ ViT: image grouped by patch and can be uncorrelated

Assumption (different data have smaller similarity)

For any data pair $(\boldsymbol{X}_i, \boldsymbol{X}_j)$*, with* $i \neq j$ *and* $i, j \in [n]$*, then we assume that* $\mathbb{P}\left(\left|\left\langle \boldsymbol{X}_i^\top \boldsymbol{X}_i, \boldsymbol{X}_j^\top \boldsymbol{X}_j \right\rangle\right| \geq t\right) \leq \exp(-t^{\hat{c}})$ *with some constant* $\hat{c} > 0$*.*

- ∘ larger $\hat{c} \Rightarrow$ more diverse data $\Rightarrow$ more separable
- ∘ validated on a language IMDB dataset (sampling with 100 normalized sentences with embedding)



Main results: Global convergence

Theorem (Under $\tau_0 = d_m^{-1/2}$)

Assume $d_m \geq d$, under LeCun/He (NTK) initialization and $d_m = \tilde{\Omega}(n^3)$ ($d_m = \tilde{\Omega}(n^2)$), with probability at least $1 - 8e^{-\frac{d_m}{2}} - \delta - \exp(-\Omega(n-1)^{-\hat{c}} d_s^{-1})$ for proper δ, choosing the step-size $\gamma \leq \mathcal{O}(n^{-\frac{1}{2}})$, then the GD training of the Transformer converges to a global minimum as follows:

$$\ell(\boldsymbol{\theta}^t) \leq \left(1 - \gamma \frac{\alpha^2}{2}\right)^t \ell(\boldsymbol{\theta}^0), \quad \textit{for } t \geq 0\,. \tag{2}$$

Main results: Global convergence

Theorem (Under $\tau_0 = d_m^{-1/2}$)

Assume $d_m \geq d$, under LeCun/He (NTK) initialization and $d_m = \tilde{\Omega}(n^3)$ ($d_m = \tilde{\Omega}(n^2)$), with probability at least $1 - 8e^{-\frac{d_m}{2}} - \delta - \exp(-\Omega(n-1)^{-\hat{c}} d_s^{-1})$ for proper δ, choosing the step-size $\gamma \leq \mathcal{O}(n^{-\frac{1}{2}})$, then the GD training of the Transformer converges to a global minimum as follows:

$$\ell(\boldsymbol{\theta}^t) \leq \left(1 - \gamma \frac{\alpha^2}{2}\right)^t \ell(\boldsymbol{\theta}^0), \quad \text{for } t \geq 0\,. \tag{2}$$

Theorem (Under $\tau_0 = d_m^{-1}$)

Under the NTK initialization, denote $\boldsymbol{\Phi}^* := \frac{1}{d_s}[\boldsymbol{X}_1^\top \mathbf{1}_{d_s}, \cdots, \boldsymbol{X}_n^\top \mathbf{1}_{d_s}]^\top \in \mathbb{R}^{n \times d}$, the limiting kernel matrix will depend on $\boldsymbol{\Phi}^*$, and with $d_m = \Omega(n)$, the GD training of Transformer converges as Eq. (2).

Main results: Global convergence

Theorem (Under $\tau_0 = d_m^{-1/2}$)

Assume $d_m \geq d$, under LeCun/He (NTK) initialization and $d_m = \tilde{\Omega}(n^3)$ ($d_m = \tilde{\Omega}(n^2)$), with probability at least $1 - 8e^{-\frac{d_m}{2}} - \delta - \exp(-\Omega(n-1)^{-\hat{c}} d_s^{-1})$ for proper δ, choosing the step-size $\gamma \leq \mathcal{O}(n^{-\frac{1}{2}})$, then the GD training of the Transformer converges to a global minimum as follows:

$$\ell(\boldsymbol{\theta}^t) \leq \left(1 - \gamma \frac{\alpha^2}{2}\right)^t \ell(\boldsymbol{\theta}^0), \quad \text{for } t \geq 0\,. \tag{2}$$

Theorem (Under $\tau_0 = d_m^{-1}$)

Under the NTK initialization, denote $\boldsymbol{\Phi}^* := \frac{1}{d_s}[\boldsymbol{X}_1^\top \mathbf{1}_{d_s}, \cdots, \boldsymbol{X}_n^\top \mathbf{1}_{d_s}]^\top \in \mathbb{R}^{n \times d}$, the limiting kernel matrix will depend on $\boldsymbol{\Phi}^*$, and with $d_m = \Omega(n)$, the GD training of Transformer converges as Eq. (2).

Remark: 1) dimension missing: self-attention layer becomes $\boldsymbol{X}\boldsymbol{W}_V^\top$
2) $\tau_0 = d_m^{-1}$ and NTK initialization make Transformer

- enter into the lazy training regime easier
- require less over-parameterization requirement

Proof framework

Polyak-Lojasiewicz (PL) inequality + Lipchitz continuous of gradient, defining $\boldsymbol{F}_{\mathrm{pre}} := \frac{\partial \boldsymbol{f}(\boldsymbol{X})}{\partial \boldsymbol{w}_o} \in \mathbb{R}^{n \times d_m}$

$$||\nabla \ell(\boldsymbol{\theta})||_2^2 \geq 2\lambda_{\min}(\boldsymbol{F}_{\mathrm{pre}}\boldsymbol{F}_{\mathrm{pre}}^\top)\ell(\boldsymbol{\theta})$$

Proof framework

Polyak-Lojasiewicz (PL) inequality + Lipchitz continuous of gradient, defining $\boldsymbol{F}_{\mathrm{pre}} := \frac{\partial \boldsymbol{f}(\boldsymbol{X})}{\partial \boldsymbol{w}_o} \in \mathbb{R}^{n \times d_m}$

$$||\nabla \ell(\boldsymbol{\theta})||_2^2 \geq 2\lambda_{\min}(\boldsymbol{F}_{\mathrm{pre}}\boldsymbol{F}_{\mathrm{pre}}^\top)\ell(\boldsymbol{\theta})$$

$$\ell(\boldsymbol{\theta}^{t+1}) \leq \ell(\boldsymbol{\theta}^t) - \frac{\gamma}{2}\lambda_{\min}(\boldsymbol{F}_{\mathrm{pre}}\boldsymbol{F}_{\mathrm{pre}}^\top)\|\boldsymbol{f}^t - \boldsymbol{y}\|_2^2 \leq (1 - \frac{\gamma\alpha}{2})\ell(\boldsymbol{\theta}^t)$$

Proof framework

Polyak-Lojasiewicz (PL) inequality + Lipchitz continuous of gradient, defining $\boldsymbol{F}_{\text{pre}} := \frac{\partial f(\boldsymbol{X})}{\partial \boldsymbol{w}_o} \in \mathbb{R}^{n \times d_m}$

$$||\nabla \ell(\boldsymbol{\theta})||_2^2 \geq 2\lambda_{\min}(\boldsymbol{F}_{\text{pre}}\boldsymbol{F}_{\text{pre}}^\top)\ell(\boldsymbol{\theta})$$

$$\ell(\boldsymbol{\theta}^{t+1}) \leq \ell(\boldsymbol{\theta}^t) - \frac{\gamma}{2}\lambda_{\min}(\boldsymbol{F}_{\text{pre}}\boldsymbol{F}_{\text{pre}}^\top)\|\boldsymbol{f}^t - \boldsymbol{y}\|_2^2 \leq (1 - \frac{\gamma\alpha}{2})\ell(\boldsymbol{\theta}^t)$$

Lemma (minimum eigenvalue estimation)

Let $\boldsymbol{\Phi} = [\boldsymbol{X}_1\boldsymbol{\beta}_{1,1}, \boldsymbol{X}_2\boldsymbol{\beta}_{1,2}, \cdots, \boldsymbol{X}_n\boldsymbol{\beta}_{1,n}]^\top \in \mathbb{R}^{n\times d}$ where $\boldsymbol{\beta}_{1,i}$ is the i-th output of softmax, then under our assumptions, we have

$$\eta_V/d_s \lesssim \lambda_0 := \lambda_{\min}\left(\mathbb{E}_{\boldsymbol{w}\sim\mathcal{N}(\boldsymbol{0},\eta_V I_d)}[\sigma_r(\boldsymbol{\Phi}\boldsymbol{w})\sigma_r(\boldsymbol{\Phi}\boldsymbol{w})^\top]\right) \lesssim \eta_V d_s \quad w.h.p$$

Proof framework

Polyak-Lojasiewicz (PL) inequality + Lipchitz continuous of gradient, defining $\boldsymbol{F}_{\text{pre}} := \frac{\partial f(\boldsymbol{X})}{\partial \boldsymbol{w}_o} \in \mathbb{R}^{n \times d_m}$

$$||\nabla \ell(\boldsymbol{\theta})||_2^2 \geq 2\lambda_{\min}(\boldsymbol{F}_{\text{pre}}\boldsymbol{F}_{\text{pre}}^\top)\ell(\boldsymbol{\theta})$$

$$\ell(\boldsymbol{\theta}^{t+1}) \leq \ell(\boldsymbol{\theta}^t) - \frac{\gamma}{2}\lambda_{\min}(\boldsymbol{F}_{\text{pre}}\boldsymbol{F}_{\text{pre}}^\top)\|\boldsymbol{f}^t - \boldsymbol{y}\|_2^2 \leq (1 - \frac{\gamma\alpha}{2})\ell(\boldsymbol{\theta}^t)$$

Lemma (minimum eigenvalue estimation)

Let $\boldsymbol{\Phi} = [\boldsymbol{X}_1\boldsymbol{\beta}_{1,1}, \boldsymbol{X}_2\boldsymbol{\beta}_{1,2}, \cdots, \boldsymbol{X}_n\boldsymbol{\beta}_{1,n}]^\top \in \mathbb{R}^{n \times d}$ where $\boldsymbol{\beta}_{1,i}$ is the i-th output of softmax, then under our assumptions, we have

$$\eta_V/d_s \lesssim \lambda_0 := \lambda_{\min}\left(\mathbb{E}_{\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \eta_V I_d)}[\sigma_r(\boldsymbol{\Phi}\boldsymbol{w})\sigma_r(\boldsymbol{\Phi}\boldsymbol{w})^\top]\right) \lesssim \eta_V d_s \quad w.h.p$$

Proof.

- Hermite expansion: $\lambda_0 > \lambda_{\min}[\boldsymbol{\Phi}\boldsymbol{\Phi}^\top]$
- Gershgorin circle theorem: $\lambda_{\min}[\boldsymbol{\Phi}\boldsymbol{\Phi}^\top] \geq \Omega(\|\boldsymbol{\beta}_{1,k}\|_2^2)$

□

Discussion on α

under LeCun initialization, we have $\alpha^2 \geq d_m \lambda_0 / 4 \geq d_m \eta_V \mu(\sigma_r)^2 \Theta(\|\boldsymbol{\beta}_{1,k}\|_2^2)$

- $\tau = d_m^{-1/2}$, we have $\|\boldsymbol{\beta}_{1,k}\|_2^2 \geq 1/d_s$
- $\tau = d_m^{-1}$, we have $\|\boldsymbol{\beta}_{1,k}\|_2^2 \approx 1/d_s$

Discussion on α

under LeCun initialization, we have $\alpha^2 \geq d_m \lambda_0 / 4 \geq d_m \eta_V \mu(\sigma_r)^2 \Theta(\|\boldsymbol{\beta}_{1,k}\|_2^2)$

- $\tau = d_m^{-1/2}$, we have $\|\boldsymbol{\beta}_{1,k}\|_2^2 \geq 1/d_s$
- $\tau = d_m^{-1}$, we have $\|\boldsymbol{\beta}_{1,k}\|_2^2 \approx 1/d_s$

different initializations: $\alpha^2 \geq \tau_1^2 \eta_V d_m \Omega(1/d)$

- LeCun/He initialization: $\alpha^2 \geq \Omega(d_m/d)$
- NTK initialization: $\alpha^2 \geq \Omega(1/d)$

Discussion on α

under LeCun initialization, we have $\alpha^2 \geq d_m \lambda_0/4 \geq d_m \eta_V \mu(\sigma_r)^2 \Theta(\|\boldsymbol{\beta}_{1,k}\|_2^2)$

- $\tau = d_m^{-1/2}$, we have $\|\boldsymbol{\beta}_{1,k}\|_2^2 \geq 1/d_s$
- $\tau = d_m^{-1}$, we have $\|\boldsymbol{\beta}_{1,k}\|_2^2 \approx 1/d_s$

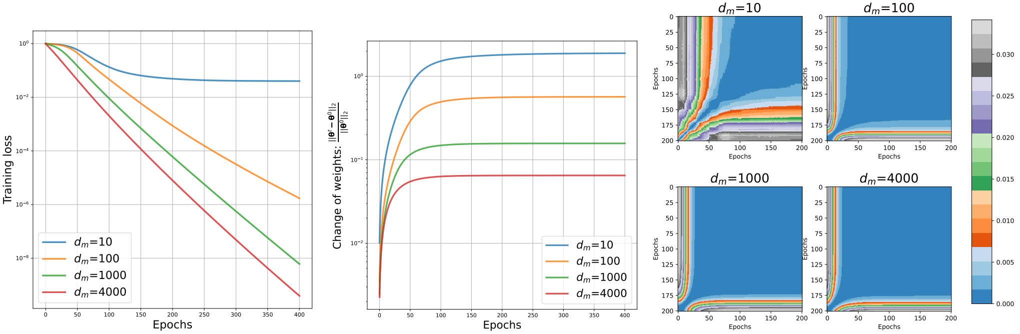
different initializations: $\alpha^2 \geq \tau_1^2 \eta_V d_m \Omega(1/d)$

- LeCun/He initialization: $\alpha^2 \geq \Omega(d_m/d)$
- NTK initialization: $\alpha^2 \geq \Omega(1/d)$

architectures under LeCun initialization:

- self-attention + two-layer ReLU NN: $\Omega(n^3)$ over-parameterization
- three-layer ReLU NN: $\Omega(n^3)$ over-parameterization

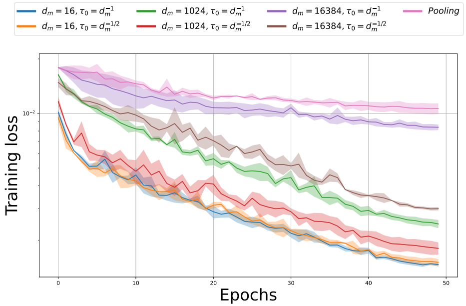
Experimental validations (width matters)



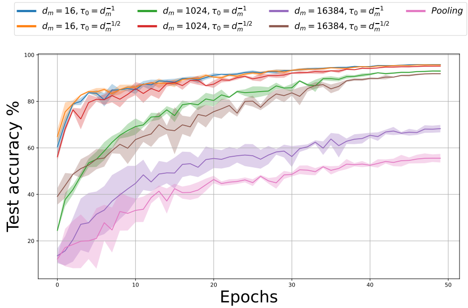
(a) Convergence curve. (b) Weight movement. (c) Kernel distance.

Figure: Visualization of the training process of Transformers with LeCun initialization and $\tau_0 = d_m^{-1/2}$ scaling on synthetic data. (a) Linear convergence. (b) Rate of change of the weights during training. Observe that the weights change very slowly after the 50^{th} epoch. (c) Evolution of the NTK during the training. The result mirrors the plot (b) and demonstrates how the kernel varies significantly at the beginning of the training and remains approximately constant later. As the width increases, the empirical NTK becomes more stable.

Separation between d_m^{-1} and $d_m^{-1/2}$



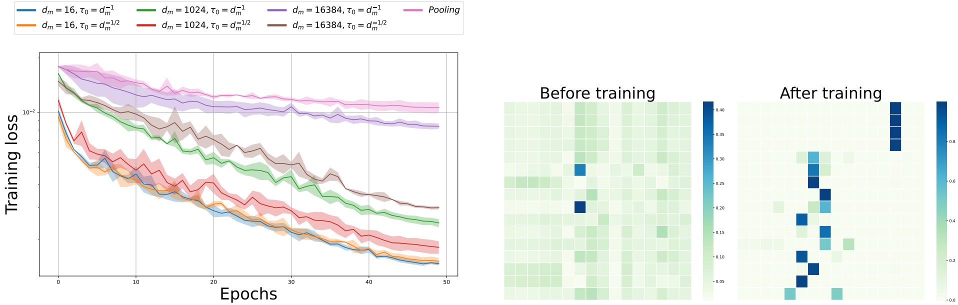
(a) Training loss



(b) Accuracy

Figure: Results on MNIST dataset trained by ViT with different scaling schemes.

Separation between d_m^{-1} and $d_m^{-1/2}$



(a) Training loss

(b) Attention map, $d_m = 16384$.

Figure: Results on MNIST dataset trained by ViT with different scaling schemes.

Conclusion

- scaling factor τ_0: $d_m^{-1/2}$ vs. d_m^{-1}
- initializations: LeCun/He vs. NTK

Conclusion

- scaling factor τ_0: $d_m^{-1/2}$ vs. d_m^{-1}
- initializations: LeCun/He vs. NTK

Future direction

- Architecture: benefits of attention
- Optimization objective: implicit bias
- Application: in-context learning, chain-of-thought reasoning

Thanks for your attention!

Q & A

my homepage www.lfhsgre.org for more information!

References I

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby.
An image is worth 16x16 words: Transformers for image recognition at scale.
2021.
(Cited on page 3.)

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin.
Attention is all you need.
In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.
(Cited on pages 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, and 13.)

[3] Arthur Jacot, Franck Gabriel, and Clément Hongler.
Neural tangent kernel: Convergence and generalization in neural networks.
In *Advances in Neural Information Processing Systems*, pages 8571–8580, 2018.
(Cited on pages 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, and 16.)

[4] Jiri Hron, Yasaman Bahri, Jascha Sohl-Dickstein, and Roman Novak.
Infinite attention: NNGP and NTK for deep attention networks.
2020.
(Cited on pages 11, 12, and 13.)

References II

[5] Greg Yang.
Tensor programs II: Neural tangent kernel for any architecture.
arXiv preprint arXiv:2006.14548, 2020.
(Cited on pages 11, 12, and 13.)

[6] Lenaic Chizat, Edouard Oyallon, and Francis Bach.
On lazy training in differentiable programming.
In *Advances in Neural Information Processing Systems*, pages 2933–2943, 2019.
(Cited on pages 14, 15, and 16.)

[7] Tao Luo, Zhi-Qin John Xu, Zheng Ma, and Yaoyu Zhang.
Phase diagram for two-layer relu neural networks at infinite-width limit.
Journal of Machine Learning Research, 22(71):1–47, 2021.
(Cited on pages 14, 15, and 16.)

[8] Hongkang Li, Meng Wang, Sijia Liu, and Pin-Yu Chen.
A theoretical understanding of shallow vision transformers: Learning, generalization, and sample complexity.
2023.
(Cited on pages 23, 24, and 25.)