Basics of Transformers, Learning Theory and In-context Learning

Yihong Zuo

June 10, 2025

Overview

Large Language Models (LLMs)

- **LLMs are autoregressive probabilistic models** defined over sequences of tokens.
- **Input:** A sequence prefix $(x_1, x_2, \ldots, x_{t-1})$.
- **Output:** A probability distribution over the vocabulary for the next token x_t.
- **Core mechanism:** The model predicts one token at a time, conditioning on all previous tokens as context.

Decoder-only Transformer

- Let the input sequence be $[w_1, w_2, \ldots, w_n]$, where each token w_i is mapped to an embedding vector $h_i \in \mathbb{R}^d$.

Decoder-only Transformer

- Let the input sequence be $[w_1, w_2, \ldots, w_n]$, where each token w_i is mapped to an embedding vector $h_i \in \mathbb{R}^d$.
- The sequence is represented as a matrix:

$$H = [h_1; h_2; \ldots; h_n] \in \mathbb{R}^{n \times d}$$

  where H is the input to the first decoder layer.
- Add positional encodings:

$$\tilde{H} = H + P$$

  where $P \in \mathbb{R}^{n \times d}$ is the positional encoding matrix.

Decoder-only Transformer

- Let the input sequence be $[w_1, w_2, \ldots, w_n]$, where each token w_i is mapped to an embedding vector $h_i \in \mathbb{R}^d$.
- The sequence is represented as a matrix:

$$H = [h_1; h_2; \ldots; h_n] \in \mathbb{R}^{n \times d}$$

  where H is the input to the first decoder layer.
- Add positional encodings:

$$\tilde{H} = H + P$$

  where $P \in \mathbb{R}^{n \times d}$ is the positional encoding matrix.
- Each decoder layer applies masked self-attention and feed-forward networks to generate contextualized representations:

$$\tilde{H} \to \text{Attention}_1 \to \text{FFN}_1 \to \cdots \to \text{Attention}_L \to \text{FFN}_L$$

- The output at each position i is used to predict the next token w_{i+1}:

$$\hat{y}_i = \text{Softmax}(W h_i^{(L)} + b)$$

Attention Layer

Definition 1

(Self-Attention layer with Softmax). A multi-head self-attention layer with M heads is denoted as $\text{Attn}_{\boldsymbol{\theta}}(\cdot)$, where

$$\boldsymbol{\theta} = \left\{(\boldsymbol{Q}_m, \boldsymbol{K}_m, \boldsymbol{V}_m) \in \mathbb{R}^{D\times D}\right\}_{m=1}^{M}$$

Given input $\boldsymbol{H} = [h_1, \ldots, h_N] \in \mathbb{R}^{D\times N}$, the layer outputs:

$$\text{Attn}_{\boldsymbol{\theta}}(\boldsymbol{H}) := \boldsymbol{H} + \sum_{m=1}^{M} \boldsymbol{V}_m \boldsymbol{H} \cdot \mathrm{Softmax}\left((\boldsymbol{Q}_m \boldsymbol{H})^\top (\boldsymbol{K}_m \boldsymbol{H})\right),$$

where Softmax is applied column-wise (across keys for each query).

Token-wise View

For each token $h_i \in \mathbb{R}^D$, the updated representation is:

$$\tilde{h}_i := h_i + \sum_{m=1}^{M} \sum_{j=1}^{N} \alpha_{ij}^{(m)} \cdot \boldsymbol{V}_m h_j,$$

where the attention weights $\alpha_{ij}^{(m)}$ are computed by

$$\alpha_{ij}^{(m)} = \frac{\exp\left(\langle \boldsymbol{Q}_m h_i, \, \boldsymbol{K}_m h_j \rangle\right)}{\sum_{k=1}^{N} \exp\left(\langle \boldsymbol{Q}_m h_i, \, \boldsymbol{K}_m h_k \rangle\right)}$$

- $\boldsymbol{Q}_m h_i$: query — what token i wants to know
- $\boldsymbol{K}_m h_j$: key — what token j is about
- $\boldsymbol{V}_m h_j$: value — information token j provides

Transformer: Pros and Cons

Advantages

- Captures long-range dependencies via self-attention
- Highly parallelizable (unlike RNNs)
- Scales efficiently with data and compute
- Achieves strong empirical performance on a wide range of tasks (NLP, vision, etc.)

Transformer: Pros and Cons

Advantages

- Captures long-range dependencies via self-attention
- Highly parallelizable (unlike RNNs)
- Scales efficiently with data and compute
- Achieves strong empirical performance on a wide range of tasks (NLP, vision, etc.)

Limitations

- Quadratic time and memory complexity in sequence length
- Each new token requires re-evaluating the full Transformer over all previous tokens

  Solutions: **Key-Value (KV) cache**, **Speculative Sampling**

Overview

The In-Context Learning (ICL) Capability

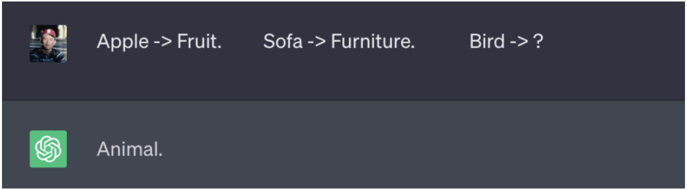


Figure: Enter Caption

The In-Context Learning (ICL) Capability

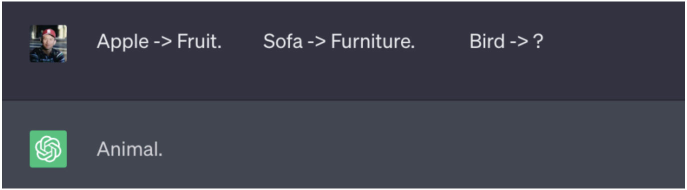


Figure: Enter Caption

- A Transformer is meta-trained on diverse tasks.
- At inference, given a prompt with a few input-output pairs (x_i, y_i), and a new input x_{N+1}, the model predicts y_{N+1}.
- No explicit parameter update—learning happens "in context"!

Mathematical Definition of In-context Learning

In-context learning capability

Let $\{(x_i, y_i)\}_{i=1}^{N+1} \sim \mathbb{P}$, with $\mathbb{P}$ unknown to the model.

Form the context input $H = [x_1, y_1, x_2, y_2, \ldots, x_N, y_N, x_{N+1}]$.

Output a good estimate $\hat{y}_{N+1} = \mathrm{TF}_{\hat{\theta}}(H) \approx y_{N+1}$.

Formulating the Problem

For a loss function $\ell(\cdot, \cdot)$, our goal is to minimize the population risk:

$$\mathcal{R}(\theta) = \mathbb{E}_{\{(x_i, y_i)\} \sim \mathbb{P}} \left[\ell(\mathrm{TF}_\theta(H), y_{N+1}) \right].$$

Given n samples $H^{(j)} = [x_1^{(j)}, y_1^{(j)}, \ldots, x_{N+1}^{(j)}]$ and $y_{N+1}^{(j)}$, we perform empirical risk minimization (ERM):

$$\min_\theta \hat{\mathcal{R}}(\theta) = \frac{1}{n} \sum_{j=1}^{n} \ell(\mathrm{TF}_\theta(H^{(j)}), y_{N+1}^{(j)})$$

Formulating the Problem

For a loss function $\ell(\cdot, \cdot)$, our goal is to minimize the population risk:

$$\mathcal{R}(\theta) = \mathbb{E}_{\{(x_i, y_i)\} \sim \mathbb{P}} \left[\ell(\mathrm{TF}_\theta(H), y_{N+1}) \right].$$

Given n samples $H^{(j)} = [x_1^{(j)}, y_1^{(j)}, \ldots, x_{N+1}^{(j)}]$ and $y_{N+1}^{(j)}$, we perform empirical risk minimization (ERM):

$$\min_\theta \hat{\mathcal{R}}(\theta) = \frac{1}{n} \sum_{j=1}^{n} \ell(\mathrm{TF}_\theta(H^{(j)}), y_{N+1}^{(j)})$$

This leads to a random, non-convex optimization problem, typically solved via stochastic gradient descent (SGD). Suppose SGD returns an approximate solution:

$$\hat{\theta} \in \left\{ \theta \,\middle|\, \hat{\mathcal{R}}(\theta) \leq \inf_{\theta \in \Theta} \hat{\mathcal{R}}(\theta) + \Delta_{\mathrm{opt}} \right\}$$

Our goal: to upper bound the population risk $\mathcal{R}(\hat{\theta})$.

Risk Decomposition

$$
\begin{aligned}
\mathcal{R}(\hat{\theta}) &= \mathcal{R}(\hat{\theta}) - \hat{\mathcal{R}}(\hat{\theta}) + \hat{\mathcal{R}}(\hat{\theta}) \\
&\leq \sup_{\theta\in\Theta} \left(\mathcal{R}(\theta) - \hat{\mathcal{R}}(\theta)\right) + \inf_{\theta\in\Theta} \hat{\mathcal{R}}(\theta) + \Delta_{\text{opt}} \\
&\leq \underbrace{2\sup_{\theta\in\Theta} \left|\mathcal{R}(\theta) - \hat{\mathcal{R}}(\theta)\right|}_{\text{generalization error}} + \underbrace{\inf_{\theta\in\Theta} \mathcal{R}(\theta)}_{\text{approximation error}} + \underbrace{\Delta_{\text{opt}}}_{\text{optimization error}}
\end{aligned}
$$

Risk Decomposition

$$
\begin{aligned}
\mathcal{R}(\hat{\theta}) &= \mathcal{R}(\hat{\theta}) - \hat{\mathcal{R}}(\hat{\theta}) + \hat{\mathcal{R}}(\hat{\theta}) \\
&\leq \sup_{\theta\in\Theta} \Big(\mathcal{R}(\theta) - \hat{\mathcal{R}}(\theta)\Big) + \inf_{\theta\in\Theta} \hat{\mathcal{R}}(\theta) + \Delta_{\text{opt}} \\
&\leq \underbrace{2\sup_{\theta\in\Theta} \Big|\mathcal{R}(\theta) - \hat{\mathcal{R}}(\theta)\Big|}_{\text{generalization error}} + \underbrace{\inf_{\theta\in\Theta} \mathcal{R}(\theta)}_{\text{approximation error}} + \underbrace{\Delta_{\text{opt}}}_{\text{optimization error}}
\end{aligned}
$$

- **Generalization error:** Often bounded using the chaining method or covering arguments.

Risk Decomposition

$$
\begin{aligned}
\mathcal{R}(\hat{\theta}) &= \mathcal{R}(\hat{\theta}) - \hat{\mathcal{R}}(\hat{\theta}) + \hat{\mathcal{R}}(\hat{\theta}) \\
&\leq \sup_{\theta \in \Theta} \left(\mathcal{R}(\theta) - \hat{\mathcal{R}}(\theta) \right) + \inf_{\theta \in \Theta} \hat{\mathcal{R}}(\theta) + \Delta_{\text{opt}} \\
&\leq \underbrace{2 \sup_{\theta \in \Theta} \left| \mathcal{R}(\theta) - \hat{\mathcal{R}}(\theta) \right|}_{\text{generalization error}} + \underbrace{\inf_{\theta \in \Theta} \mathcal{R}(\theta)}_{\text{approximation error}} + \underbrace{\Delta_{\text{opt}}}_{\text{optimization error}}
\end{aligned}
$$

- **Generalization error:** Often bounded using the chaining method or covering arguments.
- **Approximation error:** Case-dependent; typically bounded by explicitly constructing a suitable θ.

Risk Decomposition

$$
\begin{aligned}
\mathcal{R}(\hat{\theta}) &= \mathcal{R}(\hat{\theta}) - \hat{\mathcal{R}}(\hat{\theta}) + \hat{\mathcal{R}}(\hat{\theta}) \\
&\leq \sup_{\theta\in\Theta}\Big(\mathcal{R}(\theta) - \hat{\mathcal{R}}(\theta)\Big) + \inf_{\theta\in\Theta}\hat{\mathcal{R}}(\theta) + \Delta_{\mathrm{opt}} \\
&\leq \underbrace{2\sup_{\theta\in\Theta}\Big|\mathcal{R}(\theta) - \hat{\mathcal{R}}(\theta)\Big|}_{\text{generalization error}} + \underbrace{\inf_{\theta\in\Theta}\mathcal{R}(\theta)}_{\text{approximation error}} + \underbrace{\Delta_{\mathrm{opt}}}_{\text{optimization error}}
\end{aligned}
$$

- **Generalization error:** Often bounded using the chaining method or covering arguments.
- **Approximation error:** Case-dependent; typically bounded by explicitly constructing a suitable θ.
- **Optimization error:** Hard to analyze, especially for deep networks. Most works assume it is negligible or zero; we make the same assumption here. For shallow (1–2 layer) networks, some results are known, but we defer their discussion.

Overview

In the following, we use a simplified Transformer to simplify the theoretical analysis.

Definition 2

(Attention layer). A *(self-)attention layer with M heads is denoted as* $\mathrm{Attn}_{\boldsymbol{\theta}}(\cdot)$ *with parameters* $\boldsymbol{\theta} = \{(\mathbf{V}_m, \mathbf{Q}_m, \mathbf{K}_m)\}_{m\in[M]} \subset \mathbb{R}^{D\times D}$. On any input sequence $\mathbf{H} \in \mathbb{R}^{D\times N}$,

$$\widetilde{\mathbf{H}} = \mathrm{Attn}_{\boldsymbol{\theta}}(\mathbf{H}) := \mathbf{H} + \frac{1}{N}\sum_{m=1}^{M}\left(\mathbf{V}_m\mathbf{H}\cdot\sigma\left((\mathbf{Q}_m\mathbf{H})^\top(\mathbf{K}_m\mathbf{H})\right)\right) \in \mathbb{R}^{D\times N},$$

where $\sigma : \mathbb{R} \to \mathbb{R}$ is the ReLU function. In vector form,

$$\widetilde{\mathbf{h}}_i = [\mathrm{Attn}_{\boldsymbol{\theta}}(\mathbf{H})]_i = \mathbf{h}_i + \sum_{m=1}^{M}\frac{1}{N}\sum_{j=1}^{N}\sigma\left(\langle\mathbf{Q}_m\mathbf{h}_i, \mathbf{K}_m\mathbf{h}_j\rangle\right)\cdot\mathbf{V}_m\mathbf{h}_j.$$

MLP Layer

Definition 3

(MLP layer). *A (token-wise) MLP layer with hidden dimension D' is denoted as*

$$\mathrm{MLP}_{\boldsymbol{\theta}}(\cdot) \text{ with parameters } \boldsymbol{\theta} = (\mathbf{W}_1, \mathbf{W}_2) \in \mathbb{R}^{D' \times D} \times \mathbb{R}^{D \times D'}.$$

On any input sequence $\mathbf{H} \in \mathbb{R}^{D \times N}$,

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

where $\sigma : \mathbb{R} \to \mathbb{R}$ is the ReLU function. In vector form, we have

$$\widetilde{\mathbf{h}}_i = \mathbf{h}_i + \mathbf{W}_2 \sigma(\mathbf{W}_1 \mathbf{h}_i).$$

Definition 4

(Transformer). *An L-layer transformer, denoted as $\mathrm{TF}_{\boldsymbol{\theta}}(\cdot)$, is a composition of L self-attention layers each followed by an MLP layer:*

$$\mathbf{H}^{(L)} = \mathrm{TF}_{\boldsymbol{\theta}}(\mathbf{H}^{(0)}), \quad \text{where } \mathbf{H}^{(0)} \in \mathbb{R}^{D \times N}$$

is the input sequence, and

$$\mathbf{H}^{(\ell)} = \mathrm{MLP}_{\boldsymbol{\theta}_{\mathrm{mlp}}^{(\ell)}}\left(\mathrm{Attn}_{\boldsymbol{\theta}_{\mathrm{attn}}^{(\ell)}}(\mathbf{H}^{(\ell-1)})\right), \quad \ell \in \{1, \ldots, L\}.$$

Above, the parameter $\boldsymbol{\theta} = (\boldsymbol{\theta}_{\mathrm{attn}}^{(1:L)}, \boldsymbol{\theta}_{\mathrm{mlp}}^{(1:L)})$ consists of the attention layers $\boldsymbol{\theta}_{\mathrm{attn}}^{(\ell)} = \{(\mathbf{V}_m^{(\ell)}, \mathbf{Q}_m^{(\ell)}, \mathbf{K}_m^{(\ell)})\}_{m \in [M]} \subset \mathbb{R}^{D \times D}$ and the MLP layers $\boldsymbol{\theta}_{\mathrm{mlp}}^{(\ell)} = (\mathbf{W}_1^{(\ell)}, \mathbf{W}_2^{(\ell)}) \in \mathbb{R}^{D \times D'} \times \mathbb{R}^{D' \times D}$.

Norm of Parameters

We additionally define the following norm of a transformer $\mathrm{TF}_{\boldsymbol{\theta}}$:

$$\|\boldsymbol{\theta}\|_{op} := \max_{\ell \in [L]} \left\{ \max_{m \in [M]} \left\{ \|\mathbf{Q}_m^{(\ell)}\|_{\mathrm{op}}, \|\mathbf{K}_m^{(\ell)}\|_{\mathrm{op}} \right\} + \sum_{m=1}^{M} \|\mathbf{V}_m^{(\ell)}\|_{\mathrm{op}} + \|\mathbf{W}_1^{(\ell)}\|_{\mathrm{op}} + \|\mathbf{W}_2^{(\ell)}\|_{\mathrm{op}} \right\}$$

We can prove that Transformer is Lipschitz continous to this norm when inputs are bounded.

Input:

$$\mathbf{H} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_N & \mathbf{x}_{N+1} \\ y_1 & y_2 & \cdots & y_N & 0 \\ \mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_N & \mathbf{p}_{N+1} \end{bmatrix} \in \mathbb{R}^{D\times(N+1)}, \quad \mathbf{p}_i := \begin{bmatrix} \mathbf{0}_{D-(d+3)} \\ 1 \\ 1\{i < N+1\} \end{bmatrix} \in$$

We assume

$$\|\mathbf{x}_i\|_2 \leq B_x, |y_i| \leq B_y, a.s.$$

Input:

$$\mathbf{H} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_N & \mathbf{x}_{N+1} \\ y_1 & y_2 & \cdots & y_N & 0 \\ \mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_N & \mathbf{p}_{N+1} \end{bmatrix} \in \mathbb{R}^{D\times(N+1)}, \quad \mathbf{p}_i := \begin{bmatrix} \mathbf{0}_{D-(d+3)} \\ 1 \\ 1\{i < N+1\} \end{bmatrix} \in$$

We assume

$$\|\mathbf{x}_i\|_2 \leq B_x, |y_i| \leq B_y, a.s.$$

Output:

$$\widetilde{\mathbf{H}} = \mathsf{TF}_\theta(\mathbf{H})$$

$$\hat{y}_{N+1} = \widetilde{\mathrm{read}}_y(\widetilde{\mathbf{H}}) := \mathrm{clip}_R\left(\left(\widetilde{\mathbf{h}}_{N+1}\right)_{d+1}\right)$$

Ridge Regression Estimation

$$\mathbf{w}^{\lambda}_{\text{ridge}} := \arg\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{2N}\sum_{i=1}^{N}\left(\langle \mathbf{w}, \mathbf{x}_i\rangle - y_i\right)^2 + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

Main Approximation Theorem

Theorem 5 (Implementing in-context ridge regression)

For any $\lambda \geq 0$, $0 \leq \alpha \leq \beta$ *with* $\kappa := \frac{\beta+\lambda}{\alpha+\lambda}$, $B_w > 0$, *and* $\varepsilon < B_x B_w/2$, *there exists an* L*-layer attention-only transformer* TF_θ^0 *with*

$$L = \lceil 2\kappa \log(B_x B_w/(2\varepsilon)) \rceil + 1, \qquad \max_{\ell \in [L]} M^{(\ell)} \leq 3, \qquad \|\theta\|_{op} \leq 4R + 8(\beta+\lambda)$$

(with $R := \max\{B_x B_w, B_y, 1\}$*) such that the following holds. On any input data* $(\mathcal{D}, \mathbf{x}_{N+1})$ *such that the problem (ICRidge) is well-conditioned and has a bounded solution:*

$$\alpha \leq \lambda_{\min}(\mathbf{X}^\top \mathbf{X}/N) \leq \lambda_{\max}(\mathbf{X}^\top \mathbf{X}/N) \leq \beta, \qquad \|\mathbf{w}_{\mathrm{ridge}}^\lambda\|_2 \leq B_w/2,$$

TF_θ^0 *approximately implements (ICRidge): The prediction* $\hat{y}_{N+1} = \mathrm{read}_y(\mathrm{TF}_\theta^0(\mathcal{H}))$ *satisfies*

$$\left| \hat{y}_{N+1} - \langle \mathbf{w}_{\mathrm{ridge}}^\lambda, \mathbf{x}_{N+1} \rangle \right| \leq \varepsilon.$$

Main Approximation Theorem

Theorem 6 (Implementing in-context ridge regression)

Further, the second-to-last layer approximates $\mathbf{w}_{\mathrm{ridge}}^{\lambda}$*: we have* $\|\mathrm{read}_w(\mathbf{h}_i^{(L-1)}) - \mathbf{w}_{\mathrm{ridge}}^{\lambda}\|_2 \leq \varepsilon / B_x$ *for all* $i \in [N+1]$ *.*

Machanism: Approximating Gradient Descent

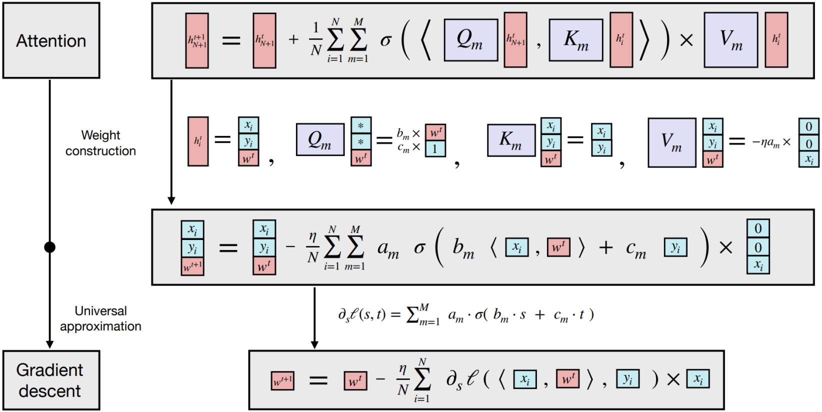
$$\min_{\mathbf{w}\in\mathbb{R}^d} L(\mathbf{w}) = \frac{1}{2N}\sum_{i=1}^{N}(\langle \mathbf{w}, \mathbf{x}_i\rangle - y_i)^2 + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

Gradient Descent Iteration

$$\begin{aligned}
\mathbf{w}^k &= \mathbf{w}^{k-1} - \eta\nabla_{\mathbf{w}} L(\mathbf{w}) \\
&= \mathbf{w}^{k-1} - \eta\left(\frac{1}{N}\sum_{i=1}^{N}\Big(\langle \mathbf{w}^{k-1}, \mathbf{x}_i\rangle - y_i\Big)\mathbf{x}_i + \lambda\mathbf{w}^{k-1}\right) \\
&= \mathbf{w}^{k-1} - \left(\frac{\eta}{N}\sum_{i=1}^{N}\Big(\langle (\mathbf{w}^{k-1}, -1), (\mathbf{x}_i, y_i)\rangle\Big)\mathbf{x}_i\right) - \frac{\eta\lambda}{N}\sum_{i=1}^{N}\mathbf{w}^{k-1}
\end{aligned}$$

Note that $x = \sigma(x) - \sigma(-x)$ for any $x \in \mathbb{R}$, The above iteration happens to be an attention!

Diagram


Attention
Weight construction
Universal approximation
Gradient descent
$h_{N+1}^{t+1} = h_{N+1}^{t} + \frac{1}{N}\sum_{i=1}^{N}\sum_{m=1}^{M} \sigma\left(\left\langle Q_m h_{N+1}^t, K_m h_i^t \right\rangle\right) \times V_m h_i^t$
$h_i^t = [x_i; y_i; w^t]$, $Q_m [*; *; w^t] = [b_m \times w^t; c_m \times 1]$, $K_m [x_i; y_i; w^t] = [x_i; y_i]$, $V_m [x_i; y_i; w^t] = -\eta a_m \times [0; 0; x_i]$
$[x_i; y_i; w^{t+1}] = [x_i; y_i; w^t] - \frac{\eta}{N}\sum_{i=1}^{N}\sum_{m=1}^{M} a_m \, \sigma\left(b_m \langle x_i, w^t \rangle + c_m \, y_i \right) \times [0; 0; x_i]$
$\partial_s \ell(s,t) = \sum_{m=1}^{M} a_m \cdot \sigma(b_m \cdot s + c_m \cdot t)$
$w^{t+1} = w^t - \frac{\eta}{N}\sum_{i=1}^{N} \partial_s \ell(\langle x_i, w^t \rangle, y_i) \times x_i$

Extension: Lasso

$$\min_{\mathbf{w}\in\mathbb{R}^d} \hat{L}_{lasso}(\mathbf{w}) = \frac{1}{2N}\sum_{i=1}^{N}\left(\langle \mathbf{w}, \mathbf{x}_i\rangle - y_i\right)^2 + \lambda_N \|\mathbf{w}\|_1$$

$$\mathbf{w}_{\text{lasso}} = \arg\min_{\mathbf{w}\in\mathbb{R}^d} = \frac{1}{2N}\sum_{i=1}^{N}\left(\langle \mathbf{w}, \mathbf{x}_i\rangle - y_i\right)^2 + \lambda_N \|\mathbf{w}\|_1$$

Implementing in-context Lasso

Theorem 7 (Implementing in-context Lasso)

For any $\lambda_N \geq 0$, $\beta > 0$, $B_w > 0$, *and* $\varepsilon > 0$, *there exists a* L*-layer transformer* TF_θ *with*

$$L = \left\lceil \frac{\beta B_w^2}{\varepsilon} \right\rceil + 1, \quad \max_{\ell \in [L]} M^{(\ell)} \leq 2, \quad \max_{\ell \in [L]} D^{(\ell)} \leq 2d, \quad \|\theta\| \leq \mathcal{O}\left(R + (1 + \lambda\right.$$

(where $R := \max\{B_x B_w, B_y, 1\}$*) such that the following holds. On any input data* $(\mathcal{D}, \mathbf{x}_{N+1})$ *such that*

$$\lambda_{\max}(\mathbf{X}^\top \mathbf{X}/N) \leq \beta \quad \textit{and} \quad \|\mathbf{w}_{\textit{lasso}}\|_2 \leq B_w/2,$$

$\mathrm{TF}_\theta(\mathbf{H}^{(0)})$ *approximately implements (ICLasso), in that it outputs* $\hat{y}_{N+1} = \langle \mathbf{x}_{N+1}, \hat{\mathbf{w}} \rangle$ *with*

$$\hat{L}_{\textit{lasso}}(\hat{\mathbf{w}}) - \hat{L}_{\textit{lasso}}(\mathbf{w}_{\textit{lasso}}) \leq \varepsilon.$$

Gradient step:

$$\mathbf{w}^{(t+\frac{1}{2})} = \mathbf{w}^{(t)} - \eta \cdot \frac{1}{N} \sum_{i=1}^{N} \left(\left\langle \mathbf{w}^{(t)}, \mathbf{x}_i \right\rangle - y_i \right) \mathbf{x}_i$$

This is the same as the ridge regression which can be achived by an Attention.

Proximal step (soft-thresholding):

$$\begin{aligned} \mathbf{w}^{(t+1)} &= \operatorname{sign}\left(\mathbf{w}^{(t+\frac{1}{2})}\right) \cdot \max\left(\left|\mathbf{w}^{(t+\frac{1}{2})}\right| - \eta\lambda_N,\, 0\right) \\ &= \sigma(\mathbf{w}^{(t+\frac{1}{2})} - \eta\lambda_N) - \sigma(-\mathbf{w}^{(t+\frac{1}{2})} - \eta\lambda_N) \end{aligned}$$

This can be achived by the MLP layer.

Extension: Regulized M-Estimator

$$\hat{L}_N(\mathbf{w}) := \frac{1}{N}\sum_{i=1}^{N} \ell(\mathbf{w}^\top \mathbf{x}_i, y_i) + \mathcal{R}(\mathbf{w})$$

Proximal Graient Descent:

$$\mathbf{w}_{\mathrm{PGD}}^{t+1} := \underbrace{\mathrm{prox}_{\eta\mathcal{R}}}_{\text{MLP}}\left(\underbrace{\mathbf{w}_{\mathrm{PGD}}^{t} - \eta\nabla\hat{L}_N^0(\mathbf{w}_{\mathrm{PGD}}^{t})}_{\text{Attention}}\right),$$

where we denote $\quad \hat{L}_N^0(\mathbf{w}) := \dfrac{1}{N}\displaystyle\sum_{i=1}^{N} \ell(\mathbf{w}^\top \mathbf{x}_i, y_i).$

Implementing In-Context M-estimation

Definition 8 (Approximability by sum of relus)

A function $g : \mathbb{R}^k \to \mathbb{R}$ is $(\varepsilon_{\text{approx}}, R, M, C)$-approximable by sum of relus, if there exists a '(M, C)-sum of relus" function

$$f_{M,C}(\mathbf{z}) = \sum_{m=1}^{M} c_m \sigma(\mathbf{a}_m^\top [\mathbf{z}; 1])$$

with

$$\sum_{m=1}^{M} |c_m| \leq C, \quad \max_{m \in [M]} \|\mathbf{a}_m\|_1 \leq 1, \quad \mathbf{a}_m \in \mathbb{R}^{k+1}, \ c_m \in \mathbb{R},$$

such that

$$\sup_{\mathbf{z} \in [-R,R]^k} |g(\mathbf{z}) - f_{M,C}(\mathbf{z})| \leq \varepsilon_{\text{approx}}.$$

Implementing In-Context M-estimation

Definition 9 (Approximability by MLP)

An operator $P : \mathbb{R}^d \to \mathbb{R}^d$ is (ε, R, D, C)-approximable by MLP, if there exists an MLP $\theta_{\text{mlp}} = (\mathbf{W}_1, \mathbf{W}_2) \in \mathbb{R}^{D\times d} \times \mathbb{R}^{d\times D}$ with hidden dimension D, $\|\mathbf{W}_1\|_{\text{op}} + \|\mathbf{W}_2\|_{\text{op}} \leq C$, such that

$$\sup_{\|\mathbf{w}\|_2 \leq R} \|P(\mathbf{w}) - \text{MLP}_{\theta_{\text{mlp}}}(\mathbf{w})\|_2 \leq \varepsilon.$$

Theorem 10 (Convex ICGPD)

Fix any $B_w > 0$, $L > 1$, $\eta > 0$, and $\varepsilon + \varepsilon' \leq B_w/(2L)$. Suppose that

1. *The loss $\ell(\cdot, \cdot)$ is convex in the first argument;*
2. *$\partial_s \ell$ is (ε, R, M, C)-approximable by sum of relus with $R = \max\{B_x B_w, B_y, 1\}$.*
3. *$\mathcal{R}$ convex, and the proximal operator $\mathrm{prox}_{\eta\mathcal{R}}(\mathbf{w})$ is $(\eta\varepsilon', R', D', C')$-approximable by MLP with $R' = \sup_{\|\mathbf{w}\|_2 \leq B_w} \|\mathbf{w}_\eta^+\|_2 + \eta\varepsilon$.*

Then there exists a transformer TF_θ with $(L+1)$ layers, $\max_{\ell \in [L]} M^{(\ell)} \leq M$ heads within the first L layers, $M^{(L+1)} = 2$, and hidden dimension D' such that, for any input data $(\mathcal{D}, \mathbf{x}_{N+1})$ such that

$$\sup_{\|\mathbf{w}\|_2 \leq B_w} \lambda_{\max}\left(\nabla^2 \widehat{L}_N(\mathbf{w})\right) \leq 2/\eta,$$

$$\exists \mathbf{w}^* \in \arg\min_{\mathbf{w} \in \mathbb{R}^d} \widehat{L}_N(\mathbf{w}) \text{ such that } \|\mathbf{w}^*\|_2 \leq B_w/2,$$

Theorem 11

the transformer output $\mathrm{TF}_\theta(\mathbf{H}^{(0)})$ *approximately implements (ICGD):*

1. *(Parameter space) For every* $\ell \in [L]$*, the* ℓ*-th layer's output* $\mathbf{H}^{(\ell)} = \mathrm{TF}_\theta^{(1:\ell)}(\mathbf{H}^{(0)})$ *approximates* ℓ *steps of (ICGD): We have* $\mathbf{h}_i^{(\ell)} = [\mathbf{x}_i; y_i'; \widehat{\mathbf{w}}^\ell; 0_{D-2d-3}; 1; t_i]$ *for every* $i \in [N+1]$*, where*

$$\|\widehat{\mathbf{w}}^\ell - \mathbf{w}_{\mathrm{PGD}}^\ell\|_2 \leq (\varepsilon + \varepsilon') \cdot (L\eta B_x).$$

2. *(Prediction space) The final output* $\mathbf{H}^{(L+1)} = \mathrm{TF}_\theta(\mathbf{H}^{(0)})$ *approximates the prediction of* L *steps of (ICGD):*
We have

$$\mathbf{h}_{N+1}^{(L+1)} = [\mathbf{x}_{N+1};\, \hat{y}_{N+1};\, \widehat{\mathbf{w}}^L;\, 0_{D-2d-3};\, 1;\, t_i],$$

where $\hat{y}_{N+1} = \langle \widehat{\mathbf{w}}^L, \mathbf{x}_{N+1} \rangle$ *so that*

$$\left| \hat{y}_{N+1} - \langle \mathbf{w}_{\mathrm{PGD}}^L, \mathbf{x}_{N+1} \rangle \right| \leq (\varepsilon + \varepsilon') \cdot (2L\eta B_x^2).$$

Further, the weight matrices have norm bounds $\|\theta\| \leq 3 + R + 2\eta C + C'$.

Overview

Question

Background

In the previous subsection, we have seen that Transformers can simulate a variety of algorithms.

New Question

Can Transformers not only simulate algorithms, but also *adaptively select* the most suitable algorithm based on the input data?

- Is it possible for a Transformer to learn to choose between multiple algorithms depending on the characteristics of the data?
- Is it possible for a Transformer to choose optimal parameters for one algorithm?

Train-Validation Split

$t_i := 1$ for $i \in \mathcal{D}_{\text{train}}$, $t_i := -1$ for $i \in \mathcal{D}_{\text{val}}$, and $t_{N+1} := 0$.

In-context algorithm selection via train-validation split

Suppose that $\ell(\cdot, \cdot)$ is approximable by sum of relus (*Definition 12*, which includes all C^3-smooth bivariate functions). Then there exists a 3-layer transformer TF_θ that maps (recalling $y_i' = y_i \mathbf{1}\{i < N+1\}$)

$$\begin{aligned} \mathbf{h}_i &= [\mathbf{x}_i; y_i'; *; f_1(\mathbf{x}_i); \cdots; f_K(\mathbf{x}_i); \mathbf{0}_{K+1}; 1; t_i] \\ \longrightarrow \quad \mathbf{h}_i' &= [\mathbf{x}_i; y_i'; *; \hat{f}(\mathbf{x}_i); 1; t_i], \quad i \in [N+1], \end{aligned}$$

where the predictor $\hat{f} : \mathbb{R}^d \to \mathbb{R}$ is a convex combination of $\{f_k : \hat{L}_{\text{val}}(f_k) \le \min_{k_\star \in [K]} \hat{L}_{\text{val}}(f_{k_\star}) + \gamma\}$. As a corollary, for any convex risk $L : (\mathbb{R}^d \to \mathbb{R}) \to \mathbb{R}$, $\hat{f}$ satisfies

$$L(\hat{f}) \le \min_{k_\star \in [K]} L(f_{k_\star}) + \max_{k \in [K]} \left| \hat{L}_{\text{val}}(f_k) - L(f_k) \right| + \gamma.$$

Adaptive regression

Adaptive regression or classification; Informal version

There exists a transformer with $\mathcal{O}(\log(1/\epsilon))$ layers such that the following holds: On any $\mathcal{D}$ such that $y_i \in \{0,1\}$, it outputs $\hat{y}_{N+1}$ that ϵ-approximates the prediction of *in-context logistic regression*.

By contrast, for any distribution $\mathbb{P}$ whose marginal distribution of y is not concentrated around $\{0,1\}$, with high probability (over $\mathcal{D}$), $\hat{y}_{N+1}$ ϵ-approximates the prediction of *in-context least squares*.

Pre Distribution Test

$$\Psi^{\text{binary}}(\mathcal{D}) = \frac{1}{N}\sum_{i=1}^{N}\psi(y_i),$$

$$\psi(y) := \begin{cases} 1, & y \in \{0,1\}, \\ 0, & y \notin [-\varepsilon,\varepsilon] \cup [1-\varepsilon, 1+\varepsilon], \\ \text{linear interpolation}, & \text{otherwise.} \end{cases}$$

Lemma 18. *There exists a single attention layer with 6 heads that implements* Ψ^{binary} *exactly.*

$$\psi(y) = \sigma\left(\frac{y+\varepsilon}{\varepsilon}\right) - 2\sigma\left(\frac{y}{\varepsilon}\right) + \sigma\left(\frac{y-\varepsilon}{\varepsilon}\right)$$
$$+\sigma\left(\frac{y-(1-\varepsilon)}{\varepsilon}\right) - 2\sigma\left(\frac{y-1}{\varepsilon}\right) + \sigma\left(\frac{y-(1+\varepsilon)}{\varepsilon}\right)$$

Overview

Generalization Theory

Denote

$$\ell_{\text{icl}}(\boldsymbol{\theta}, \boldsymbol{Z}^{(i)}) = \ell\left(\text{read}\left(\text{TF}_{\boldsymbol{\theta}}(\boldsymbol{H}^{(i)})\right), y_{N+1}^{(i)}\right)$$

Recall the generalization error:

$$\sup_{\boldsymbol{\theta}\in\Theta}\left|\mathcal{R}(\boldsymbol{\theta}) - \widehat{\mathcal{R}}(\boldsymbol{\theta})\right| = \sup_{\boldsymbol{\theta}\in\Theta}\left|\frac{1}{n}\sum_{i=1}^{n}\left(\ell_{\text{icl}}(\boldsymbol{\theta}, \boldsymbol{Z}^{(i)}) - \mathbb{E}\ell_{\text{icl}}(\boldsymbol{\theta}, \boldsymbol{Z}^{(i)})\right)\right|$$

Denote

$$Y_{\boldsymbol{\theta}}^{(i)} = \ell_{\text{icl}}(\boldsymbol{\theta}, \boldsymbol{Z}^{(i)}) - \mathbb{E}\ell_{\text{icl}}(\boldsymbol{\theta}, \boldsymbol{Z}^{(i)})$$

$$X_{\boldsymbol{\theta}} = \frac{1}{n}\sum_{i=1}^{n} Y_{\boldsymbol{\theta}}^{(i)}$$

One-step Chaining

We have

$$\sup_{\boldsymbol{\theta}\in\Theta} |X_{\boldsymbol{\theta}}| \leq \sup_{\boldsymbol{\theta}\in\Theta_\varepsilon} \left| \frac{1}{n}\sum_{i=1}^{n} Y_{\boldsymbol{\theta}}^{(i)} \right| + \sup_{\boldsymbol{\theta}\in\Theta} \inf_{\phi\in\Theta_\varepsilon} |X_{\boldsymbol{\theta}} - X_\phi|$$

where Θ_ε is a minimal finite ε-covering of the parameter space Θ. Suppose the following conditions hold:

(a) *The index set Θ is equipped with a distance ρ and has diameter D. For some constant A, for any ball Θ' of radius r in Θ, the covering number satisfies $\log N(\delta; \Theta', \rho) \leq d\log(2Ar/\delta)$ for all $0 < \delta \leq 2r$.*

(b) *For any fixed $\boldsymbol{\theta} \in \Theta$ and z sampled from $\mathbb{P}_z$, the random variable $Y_{\boldsymbol{\theta}}(z)$ is sub-Gaussian: $Y_{\boldsymbol{\theta}}(z) \sim \mathrm{SG}(B^0)$.*

(c) *For any $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta$ and $z \sim \mathbb{P}_z$, the difference $Y_{\boldsymbol{\theta}}(z) - Y_{\boldsymbol{\theta}'}(z)$ is sub-Gaussian: $Y_{\boldsymbol{\theta}}(z) - Y_{\boldsymbol{\theta}'}(z) \sim \mathrm{SG}(B^1\rho(\boldsymbol{\theta}, \boldsymbol{\theta}'))$.*

Bounding Term 1

Fix $D_0 \in (0, D]$ to be specified later. Take a $(D_0/2)$-covering Θ_0 of Θ so that $\log|\Theta_0| \le d\log(2AD/D_0)$. By standard results on covering numbers of independent sub-Gaussian random variables, with probability at least $1 - \delta/2$,

$$\sup_{\boldsymbol{\theta}\in\Theta_0} |X_{\boldsymbol{\theta}}| \le CB^0\sqrt{\frac{\log(2AD/D_0) + \log(2/\delta)}{n}}$$

where C is a universal constant.

Bounding Term 2

Assume $\Theta_0 = \{\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_n\}$. For each $j \in [n]$, let Θ_j be the ball centered at $\boldsymbol{\theta}_j$ with radius D_0 in (Θ, ρ). For $\boldsymbol{\theta} \in \Theta_j$, Θ_j has diameter D_0 and

$$\log \mathcal{N}(\Theta_j, \delta) \leq d \log(AD_0/\delta).$$

Applying Theorem in *High-dimensional Statistics* Section 5.6 to the process $\{X_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta} \in \Theta_j}$, we get

$$\psi = \psi_2, \qquad \|X_{\boldsymbol{\theta}} - X_{\boldsymbol{\theta}'}\|_\psi \leq \frac{B^1}{\sqrt{n}} \rho(\boldsymbol{\theta}, \boldsymbol{\theta}'),$$

and

$$\mathbb{P}\left(\sup_{\boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta_j} |X_{\boldsymbol{\theta}} - X_{\boldsymbol{\theta}'}| \leq C' B^1 D_0 \left(\sqrt{\frac{d \log(2A)}{n}} + t\right)\right) \leq 2\exp(-nt^2)$$

$$, \quad \forall t \geq 0.$$

Generalization Bound

Theorem 12

Suppose (a), (b), (c) hold. Then with probability at least $1-\delta$, we have

$$\sup_{\boldsymbol{\theta}\in\Theta} |X_{\boldsymbol{\theta}}| \leq CB^0\sqrt{\frac{d\log(2A\kappa)+\log(1/\delta)}{n}}$$

where C is a universal constant, and $\kappa = 1 + B^1 D/B^0$.

Verification of the Assumptions

(a)

Lemma 13 (HDS, Example 5.8)

Given any norm $\|\cdot\|'$, *let* $\mathcal{B} = \{\boldsymbol{\theta} \in \mathbb{R}^d : \|\boldsymbol{\theta}\|' \leq 1\}$. *Then*

$$\log N(\delta, \mathcal{B}, \|\cdot\|') \leq d \log\left(1 + \frac{2}{\delta}\right)$$

(b) Since inputs and parameters are bounded, $Y_{\boldsymbol{\theta}}$ is bounded and thus sub-Gaussian.

Proposition 14 (Lipschitzness of transformers)

Fix the number of heads M and hidden dimension D'. Define

$$\Theta_{\mathrm{TF},L,B} = \left\{ \boldsymbol{\theta} = \left(\theta_{\mathrm{attn}}^{(1:L)}, \theta_{\mathrm{mlp}}^{(1:L)} \right) : M^{(\ell)} = M,\ D^{(\ell)} = D',\ \|\boldsymbol{\theta}\| \le B \right\}$$

Then, for any fixed $\boldsymbol{H}$, the function $\mathrm{TF}^{\mathbb{R}}$ is $(LB_H^{L-1}B_\Theta)$-Lipschitz w.r.t. $\boldsymbol{\theta} \in \Theta_{\mathrm{TF},L,B}$.

(c) For all $\boldsymbol{\theta}, \boldsymbol{\theta}'$, $Y_{\boldsymbol{\theta}}(Z) - Y_{\boldsymbol{\theta}'}(Z)$ is $B^1\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_{\mathrm{op}}$-sub-Gaussian for some B^1.

Generalization Bound of Pretraining

Theorem 15 (Generalization for pretraining)

With probability at least $1-\xi$ (over the pretraining instances $\{\mathbf{Z}^j\}_{j\in[n]}$), the solution $\hat{\theta}$ to (TF-ERM) *satisfies*

$$L_{\mathrm{icl}}(\hat{\theta}) \leq \inf_{\theta\in\Theta_{L,M,D',B}} L_{\mathrm{icl}}(\theta) + \mathcal{O}\left(B_y^2\sqrt{\frac{L^2(MD^2+DD')\iota+\log(1/\xi)}{n}}\right),$$

where $\iota = \log(2+\max\{B,R,B_y\})$ is a log factor.

Compose the Bounds

Theorem 16 (Pretraining transformers for in-context linear regression)

Suppose $\mathbf{P} \sim \pi$ *is almost surely well-posed for in-context linear regression (Assumption A) with the canonical parameters. Then, for* $N \geq \widetilde{\mathcal{O}}(d)$*, with probability at least* $1 - \xi$ *(over the training instances* $\mathbf{Z}^{(1:n)}$*), the solution* $\hat{\theta}$ *of* (TF-ERM) *with* $L = \mathcal{O}(\kappa \log(\kappa N/\sigma))$ *layers,* $M = 3$ *heads,* $D' = 0$ *(attention-only), and* $B = \mathcal{O}(\sqrt{\kappa d})$ *achieves small excess ICL risk over* $\mathbf{w}^\star_{\mathbf{P}}$*:*

$$L_{\mathrm{icl}}(\hat{\theta}) - \mathbb{E}_{\mathbf{P}\sim\pi}\mathbb{E}_{(\mathbf{x},y)\sim\mathbf{P}}\left[\frac{1}{2}\left(y - \langle \mathbf{w}^\star_{\mathbf{P}}, \mathbf{x}\rangle\right)^2\right] \leq \widetilde{\mathcal{O}}\left(\sqrt{\frac{\kappa^2 d^2 + \log(1/\xi)}{n}} + \frac{d\sigma^2}{N}\right)$$

where $\widetilde{\mathcal{O}}(\cdot)$ *only hides polylogarithmic factors in* $\kappa, N, 1/\sigma$*.*

When we have sufficient training sample($n \geq \widetilde{\mathcal{O}}(\kappa^2 N/\sigma^2)$,), the above bound achieve Baysian optimal excess risk $\widetilde{O}(\frac{d\sigma^2}{N})$

Overview

Takeaways

- LLMs are **autoregressive probabilistic** models over sequences of tokens.
- Transformers are neural networks with **Attention Layers**.
- In-context learning is the ability to learn tasks and generate corresponding outputs by entering examples **without parameter updates.**
- Approximation Error can be bounded by **constructing** a specific model to **simulate an algorithnm.**
- Generilization Error can be bounded by **chaining methods**.

References

1. Bai, Yu, et al. "Transformers as statisticians: Provable in-context learning with in-context algorithm selection." Advances in neural information processing systems 36 (2023): 57125-57211.
2. Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).
3. Wainwright, Martin J. High-dimensional statistics: A non-asymptotic viewpoint. Vol. 48. Cambridge university press, 2019.