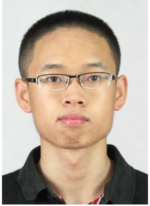
Transformers Meet In-Context Learning: A Universal Approximation Theory

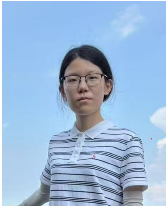
Yuxin Chen

Wharton Statistics & Data Science

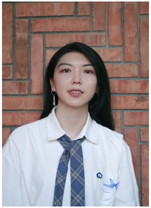
Coauthors



Gen Li
CUHK



Yuchen Jiao
CUHK



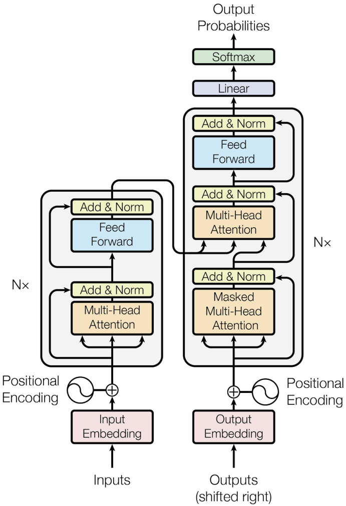
Yu Huang
UPenn

Yuting Wei
UPenn

"Transformers meet in-context learning: A universal approximation theory," G. Li, Y. Jiao, Y. Huang, Y. Wei, Y. Chen, arXiv:2506.05200, 2025

Transformers:

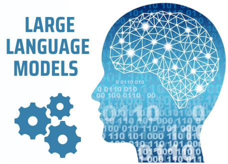
leverage attention mechanism to capture dependencies between tokens in a sequence



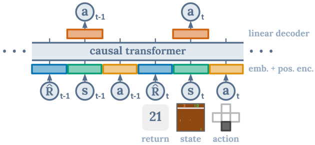
"Attention is all you need", Vaswani, Shazeer, Uszkoreit, Jones, Gomez, Kaiser, Polosukhin '17

Transformer

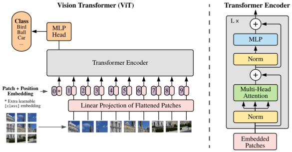
Attention Is All You Need



large language models



reinforcement learning



vision transformer (ViT)

image generation

Emergent ability: in-context learning

In-context learning (ICL): a pretrained LLM can perform a task from a few examples w/o fine-tuning or weight updates

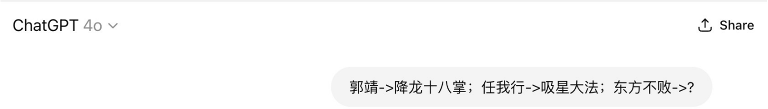
arXiv
https://arxiv.org › cs

[2005.14165] Language Models are Few-Shot Learners

by TB Brown · 2020 · Cited by 31178 — Specifically, we train **GPT-3**, an autoregressive language model with 175 billion parameters, 10x more than any previous non-sparse languag…

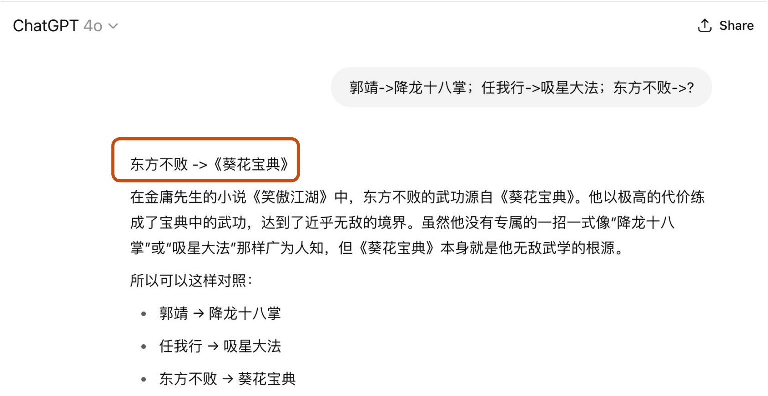
Emergent ability: in-context learning

In-context learning (ICL): a pretrained LLM can perform a task from a few examples w/o fine-tuning or weight updates

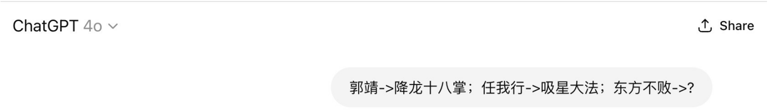


Emergent ability: in-context learning

In-context learning (ICL): a pretrained LLM can perform a task from a few examples w/o fine-tuning or weight updates



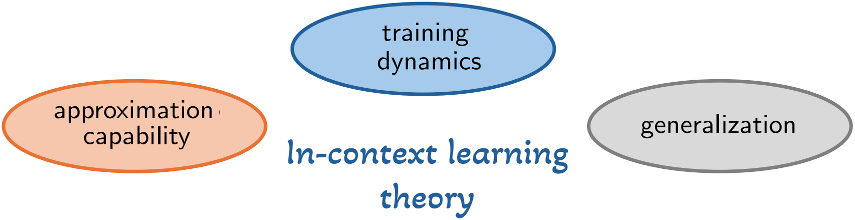
In-context learning/inference

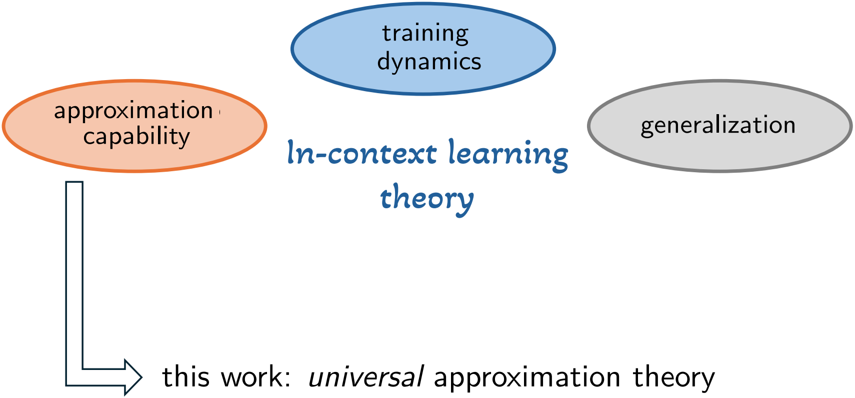


- given $\underbrace{\text{any function } f \text{ of interest}}_{\text{specifies a task}}$ and the prompt below

$$\text{prompt}: \quad \begin{array}{ccccc} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_N & \color{red}{\boldsymbol{x}_{N+1}} \\ \downarrow & \downarrow & \vdots & \downarrow & \color{red}{\downarrow} \\ f(\boldsymbol{x}_1) & f(\boldsymbol{x}_2) & \cdots & f(\boldsymbol{x}_N) & \color{red}{?} \end{array}$$

- predict $f(\boldsymbol{x}_{N+1})$ on the fly (w/o weight updates)


training
dynamics
approximation
capability
generalization
In-context learning
theory


training
dynamics
approximation
capability
generalization
In-context learning
theory
this work: *universal* approximation theory

Transformers as algorithm approximators

A dominant approach in prior approximation theory

— construct transformers to mimic iterations of optimization algs.

Transformers as algorithm approximators

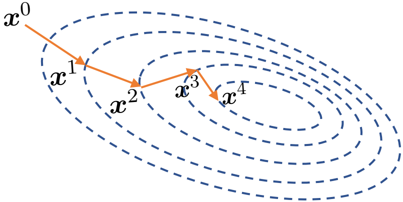
A dominant approach in prior approximation theory

— construct transformers to mimic iterations of optimization algs.

[Submitted on 15 Dec 2022 (v1), last revised 31 May 2023 (this version, v2)]

Transformers learn in-context by gradient descent

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, Max Vladymyrov



Transformers as algorithm approximators

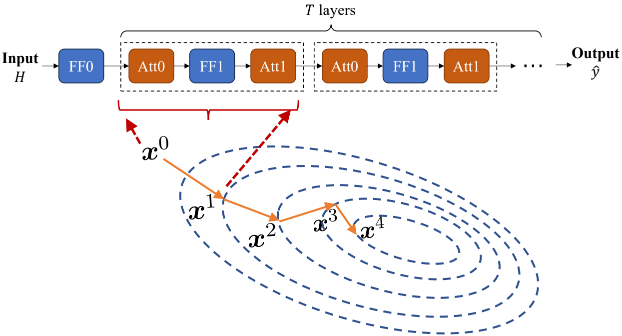
A dominant approach in prior approximation theory

— construct transformers to mimic iterations of optimization algs.

[Submitted on 15 Dec 2022 (v1), last revised 31 May 2023 (this version, v2)]

Transformers learn in-context by gradient descent

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, Max Vladymyrov



Transformers as algorithm approximators

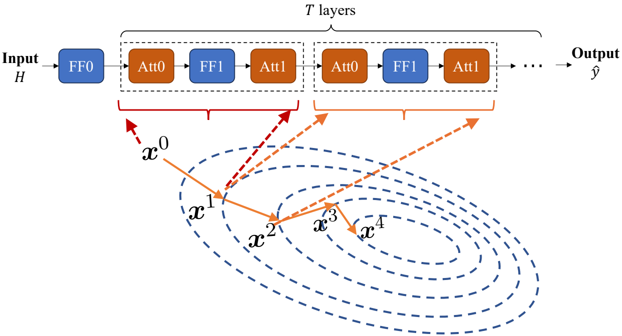
A dominant approach in prior approximation theory

— construct transformers to mimic iterations of optimization algs.

[Submitted on 15 Dec 2022 (v1), last revised 31 May 2023 (this version, v2)]

Transformers learn in-context by gradient descent

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, Max Vladymyrov



Transformers as algorithm approximators

A dominant approach in prior approximation theory

— construct transformers to mimic iterations of optimization algs.

- gradient descent (Von Oswald et al '23)
- preconditioned GD (Ahn et al '23)
- Newton method (Gianno et al '23; Fu et al '24)
- ...
- *algorithm selection* (Bai et al '23)

Transformers as algorithm approximators

A dominant approach in prior approximation theory

— construct transformers to mimic iterations of optimization algs.

- gradient descent (Von Oswald et al '23)
- preconditioned GD (Ahn et al '23)
- Newton method (Gianno et al '23; Fu et al '24)
- ...
- *algorithm selection* (Bai et al '23)

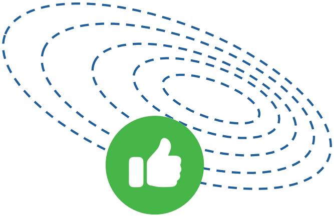
key takeaway: transformers can implement *generic* optimization algs. during inference $\longrightarrow$ in-context inference

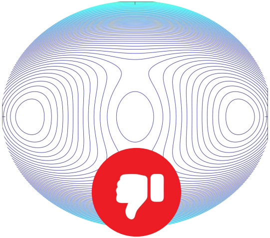
Inadequacy of prior approximation theory

algorithm approximator perspective $\longrightarrow$ constrained by effectiveness of optimization algs (e.g., GD) being approximated

Inadequacy of prior approximation theory

algorithm approximator perspective $\longrightarrow$ constrained by effectiveness of optimization algs (e.g., GD) being approximated

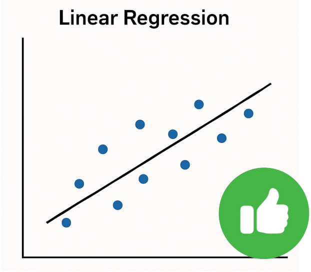


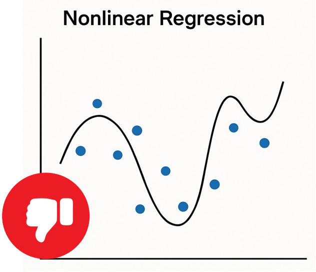


- GD works for convex problems; fails for nonconvex ones

Inadequacy of prior approximation theory

algorithm approximator perspective $\longrightarrow$ constrained by effectiveness of optimization algs (e.g., GD) being approximated





- restricted to $\underbrace{\text{learning linear functions}}_{\text{e.g. linear regression}}$

Can we develop a universal approximation theory that accommodates $\underbrace{\textit{general function classes}}_{\textit{nonconvex problems; beyond linear regression}}$*?*

Formulation: in-context learning

- **function class $\mathcal{F}$:** a set of functions ($\mathbb{R}^d \to \mathbb{R}$)
  - each function $f \in \mathcal{F}$ describes a task

Formulation: in-context learning

- **function class $\mathcal{F}$:** a set of functions ($\mathbb{R}^d \to \mathbb{R}$)
  - each function $f \in \mathcal{F}$ describes a task
- **prompt:** N in-context examples + 1 input for prediction

$$\big(\underbrace{\boldsymbol{x}_1, y_1, \boldsymbol{x}_2, y_2, \ldots, \boldsymbol{x}_N, y_N,}_{N \text{ in-context examples}} \underbrace{\boldsymbol{x}_{N+1}}_{\text{to predict}}\big)$$

  - $y_i \approx f(\boldsymbol{x}_i)$ for some task $f \in \mathcal{F}$

Formulation: in-context learning

- **function class $\mathcal{F}$:** a set of functions ($\mathbb{R}^d \to \mathbb{R}$)
  - each function $f \in \mathcal{F}$ describes a task
- **prompt:** N in-context examples + 1 input for prediction

$$\big(\underbrace{\boldsymbol{x}_1, y_1, \boldsymbol{x}_2, y_2, \ldots, \boldsymbol{x}_N, y_N}_{N \text{ in-context examples}}, \underbrace{\boldsymbol{x}_{N+1}}_{\text{to predict}}\big)$$

  - $y_i \approx f(\boldsymbol{x}_i)$ for some task $f \in \mathcal{F}$
- **goal:** construct a single transformer that works for all tasks: given prompt produced by any $f \in \mathcal{F}$, outputs

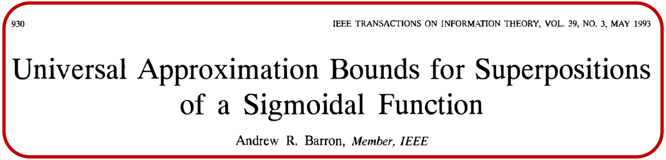
$$\widehat{y}_{N+1} \approx f(\boldsymbol{x}_{N+1})$$

Assumptions: in-context examples

$$y_i \overset{\text{i.i.d.}}{=} f(\boldsymbol{x}_i) + z_i, \qquad 1 \le i \le N$$

- input vector: $\boldsymbol{x}_i \overset{\text{i.i.d.}}{\sim} \mathcal{D}_{\mathcal{X}}$, $\|\boldsymbol{x}_i\|_2 \le 1$,
- sub-Gaussian noise z_i: $\mathbb{E}[z_i] = 0$, sub-Gaussian norm σ

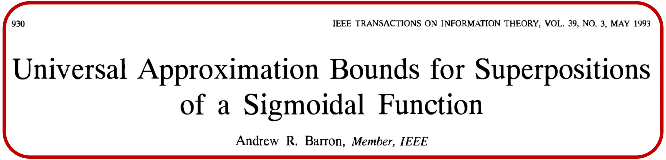
Key Fourier parameter for function class


930 IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 39, NO. 3, MAY 1993

Universal Approximation Bounds for Superpositions of a Sigmoidal Function

Andrew R. Barron, *Member, IEEE*

Key Fourier parameter for function class


930 IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 39, NO. 3, MAY 1993

Universal Approximation Bounds for Superpositions of a Sigmoidal Function

Andrew R. Barron, *Member, IEEE*

Recall: a classical Fourier quantity w.r.t. universal approx for sigmoids

$$\underbrace{C_f := \int_{\boldsymbol{\omega}} \|\boldsymbol{\omega}\|_2 |F_f(\boldsymbol{\omega})| \mathrm{d}\boldsymbol{\omega}}_{\ell_1 \text{ norm of Fourier-transform}(\nabla f)}$$

where $F_f(\boldsymbol{\omega}) = \underbrace{\frac{1}{2\pi} \int_{\boldsymbol{x}} e^{-j\boldsymbol{\omega}^\top \boldsymbol{x}} f(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}}_{\text{Fourier transform of } f}$

Key Fourier parameter for function class

930 IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 39, NO. 3, MAY 1993

Universal Approximation Bounds for Superpositions of a Sigmoidal Function

Andrew R. Barron, *Member, IEEE*

this work: extend C_f to handle a function class

$$C_{\mathcal{F}} := \sup_{f \in \mathcal{F}} |f(\mathbf{0})| + \int_{\boldsymbol{\omega}} \|\boldsymbol{\omega}\|_2 \sup_{f \in \mathcal{F}} |F_f(\boldsymbol{\omega})| \mathrm{d}\boldsymbol{\omega} < \infty,$$

Preliminaries: transformer architecture

input matrix: encode inputs as a sequence of $N+1$ tokens

$$\boldsymbol{H} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_N, \boldsymbol{h}_{N+1}]$$

Preliminaries: transformer architecture

input matrix: encode inputs as a sequence of $N+1$ tokens

$$\boldsymbol{H} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_N, \boldsymbol{h}_{N+1}] = \begin{bmatrix} \boldsymbol{x}_1 & \cdots & \boldsymbol{x}_N & \boldsymbol{x}_{N+1} \\ 1 & \cdots & 1 & 1 \\ y_1 & \cdots & y_N & 0 \\ \vdots & \text{auxiliary} & \text{info} & \vdots \\ \widehat{y}_1 & \cdots & \widehat{y}_N & \widehat{y}_{N+1} \end{bmatrix} \in \mathbb{R}^{D\times(N+1)}$$

- unified format after tokenization, suitable for joint processing

Preliminaries: transformer architecture

input matrix: encode inputs as a sequence of $N+1$ tokens

$$\boldsymbol{H} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_N, \boldsymbol{h}_{N+1}] = \begin{bmatrix} \boldsymbol{x}_1 & \cdots & \boldsymbol{x}_N & \boldsymbol{x}_{N+1} \\ 1 & \cdots & 1 & 1 \\ y_1 & \cdots & y_N & 0 \\ \vdots & \text{auxiliary} & \text{info} & \vdots \\ \widehat{y}_1 & \cdots & \widehat{y}_N & \widehat{y}_{N+1} \end{bmatrix} \in \mathbb{R}^{D \times (N+1)}$$

- unified format after tokenization, suitable for joint processing
- auxiliary info expands feature dimension

Preliminaries: transformer architecture

attention mechanism: dynamically attend to different parts of input

- attention operator:

$$\mathsf{attn}(\boldsymbol{H}; \boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) \coloneqq \frac{1}{N} \underbrace{\boldsymbol{V}}_{\text{value}} \boldsymbol{H} \sigma_{\mathsf{attn}}((\underbrace{\boldsymbol{Q}}_{\text{query}} \boldsymbol{H})^\top \underbrace{\boldsymbol{K}}_{\text{key}} \boldsymbol{H})$$

Preliminaries: transformer architecture

attention mechanism: dynamically attend to different parts of input

- attention operator:

$$\mathsf{attn}(\boldsymbol{H};\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) \coloneqq \frac{1}{N}\underbrace{\boldsymbol{V}}_{\text{value}}\boldsymbol{H}\,\sigma_{\mathsf{attn}}\big((\underbrace{\boldsymbol{Q}}_{\text{query}}\boldsymbol{H})^\top \underbrace{\boldsymbol{K}}_{\text{key}}\boldsymbol{H}\big)$$

value   query   key

activation function

$$\sigma_{\mathsf{attn}}(x) = \frac{e^x}{e^x + 1}$$

- multi-head attention layer:

$$\mathsf{Attn}_{\boldsymbol{\Theta}}(\boldsymbol{H}) \coloneqq \boldsymbol{H} + \underbrace{\sum_{m=1}^{M} \mathsf{attn}(\boldsymbol{H};\boldsymbol{Q}_m,\boldsymbol{K}_m,\boldsymbol{V}_m)}_{M \text{ attention heads}}$$

Preliminaries: transformer architecture

feed-forward (a.k.a. MLP) layer: refines feature representation through non-linear transformation

$$\mathsf{FF}_{\boldsymbol{\Theta}}(\boldsymbol{H}) := \boldsymbol{H} + \boldsymbol{U}\sigma_{\mathsf{ff}}(\boldsymbol{W}\boldsymbol{H})$$

parameters

Preliminaries: transformer architecture

feed-forward (a.k.a. MLP) layer: refines feature representation through non-linear transformation

$$\mathsf{FF}_{\mathbf{\Theta}}(\boldsymbol{H}) := \boldsymbol{H} + \boldsymbol{U}\sigma_{\mathsf{ff}}(\boldsymbol{W}\boldsymbol{H})$$

parameters

activation function
$\sigma_{\mathsf{ff}}(x) = x\,\mathbb{1}(x > 0)$

Preliminaries: transformer architecture

multi-layer transformers:

- L attention layers + L feed-forward layers

$$H^{(l)} = \mathsf{FF}_{\Theta_{\mathsf{ff}}^{(l)}}\Big(\mathsf{Attn}_{\Theta_{\mathsf{attn}}^{(l)}}\big(H^{(l-1)}\big)\Big), \qquad l = 1, \ldots, L,$$

- prediction: last entry of $H^{(L)}$

Our universal approximation theory

Theorem 1 (informal; Li, Jiao, Huang, Wei, Chen '25)

Consider a general function class $\mathcal{F}$. One can construct a multi-layer transformer s.t.: for every $f \in \mathcal{F}$,

$$\textit{in-context-prediction-risk} \;\rightarrow\; 0 \qquad \textit{with high prob.}$$

Our universal approximation theory

Theorem 1 (informal; Li, Jiao, Huang, Wei, Chen '25)

Consider a general function class $\mathcal{F}$. One can construct a multi-layer transformer s.t.: for every $f \in \mathcal{F}$,

$$\textit{in-context-prediction-risk} \;\rightarrow\; 0 \qquad \textit{with high prob.}$$

- reliable in-context learning

Our universal approximation theory

Theorem 1 (informal; Li, Jiao, Huang, Wei, Chen '25)

Consider a general function class $\mathcal{F}$. One can construct a multi-layer transformer s.t.: for every $f \in \mathcal{F}$,

$$\textit{in-context-prediction-risk} \ \rightarrow \ 0 \qquad \textit{with high prob.}$$

- reliable in-context learning
- universal design (1 transformer for all tasks)

Our universal approximation theory

Theorem 1 (informal; Li, Jiao, Huang, Wei, Chen '25)

Consider a general function class $\mathcal{F}$. One can construct a multi-layer transformer s.t.: for every $f \in \mathcal{F}$,

$$\textit{in-context-prediction-risk} \;\to\; 0 \qquad \textit{with high prob.}$$

- reliable in-context learning
- universal design (1 transformer for all tasks)
- far beyond linear functions
  - not constrained by effectiveness of GD, Newton's, etc
  - accommodate much broader ICL problems (far beyond convex)

Our universal approximation theory (formal)

Theorem 1 (Li, Jiao, Huang, Wei, Chen '25)

One can construct a transformer s.t.: for every $f \in \mathcal{F}$, with high prob.

$$\underbrace{\mathbb{E}\Big[\big(\widehat{y}_{N+1} - f(\boldsymbol{x}_{N+1})\big)^2\Big]}_{\text{prediction error}} \lesssim \left(\sqrt{\frac{\log N}{N}} + \frac{n}{L}\right) C_{\mathcal{F}}(C_{\mathcal{F}} + \sigma) + C_{\mathcal{F}}^2 \left(\frac{\log|\mathcal{N}_\varepsilon|}{n}\right)^{\frac{2}{3}}$$

as long as $n \gtrsim \log|\mathcal{N}_\varepsilon|$, $\varepsilon \lesssim \sqrt{\frac{\log N}{N}} + \frac{n}{L}$

Our universal approximation theory (formal)

Theorem 1 (Li, Jiao, Huang, Wei, Chen '25)

One can construct a transformer s.t.: for every $f \in \mathcal{F}$, with high prob.

$$\underbrace{\mathbb{E}\Big[\big(\widehat{y}_{N+1} - f(\boldsymbol{x}_{N+1})\big)^2\Big]}_{\text{prediction error}} \lesssim \left(\sqrt{\frac{\log N}{N}} + \frac{n}{L}\right) C_{\mathcal{F}}(C_{\mathcal{F}} + \sigma) + C_{\mathcal{F}}^2 \left(\frac{\log |\mathcal{N}_\varepsilon|}{n}\right)^{\frac{2}{3}}$$

as long as $n \gtrsim \log |\mathcal{N}_\varepsilon|$, $\varepsilon \lesssim \sqrt{\frac{\log N}{N}} + \frac{n}{L}$

- $\mathcal{N}_\varepsilon$: ε-cover of $\mathcal{F} \times$ unit-ball
- L: depth
- N: # input examples
- $C_{\mathcal{F}}$: Fourier quantity of $\mathcal{F}$
- $M \asymp 1$: # attention heads
- n: dimension of aux features
- σ: noise level

Our universal approximation theory (formal)

Theorem 1 (Li, Jiao, Huang, Wei, Chen '25)

One can construct a transformer s.t.: for every $f \in \mathcal{F}$, with high prob.

$$\underbrace{\mathbb{E}\Big[\big(\widehat{y}_{N+1} - f(\boldsymbol{x}_{N+1})\big)^2\Big]}_{\text{prediction error}} \lesssim \left(\sqrt{\frac{\log N}{N}} + \frac{n}{L}\right) C_{\mathcal{F}}(C_{\mathcal{F}} + \sigma) + C_{\mathcal{F}}^2 \left(\frac{\log |\mathcal{N}_\varepsilon|}{n}\right)^{\frac{2}{3}}$$

as long as $n \gtrsim \log |\mathcal{N}_\varepsilon|$, $\varepsilon \lesssim \sqrt{\frac{\log N}{N}} + \frac{n}{L}$

parameter choice: to yield $\varepsilon_{\mathsf{pred}}$-accuracy, suffices to choose

$$n \asymp C_{\mathcal{F}}^3 \varepsilon_{\mathsf{pred}}^{-3/2} \log |\mathcal{N}_\varepsilon|, \qquad N \gtrsim C_{\mathcal{F}}^2 (C_{\mathcal{F}} + \sigma)^2 \varepsilon_{\mathsf{pred}}^{-2}$$

$$L \gtrsim C_{\mathcal{F}}^4 (C_{\mathcal{F}} + \sigma) \varepsilon_{\mathsf{pred}}^{-5/2} \log |\mathcal{N}_\varepsilon|$$

Our universal approximation theory (formal)

Theorem 1 (Li, Jiao, Huang, Wei, Chen '25)

One can construct a transformer s.t.: for every $f \in \mathcal{F}$, with high prob.

$$\underbrace{\mathbb{E}\Big[\big(\widehat{y}_{N+1} - f(\boldsymbol{x}_{N+1})\big)^2\Big]}_{\text{prediction error}} \lesssim \left(\sqrt{\frac{\log N}{N}} + \frac{n}{L}\right) C_{\mathcal{F}}(C_{\mathcal{F}} + \sigma) + C_{\mathcal{F}}^2 \left(\frac{\log|\mathcal{N}_\varepsilon|}{n}\right)^{\frac{2}{3}}$$

as long as $n \gtrsim \log|\mathcal{N}_\varepsilon|$, $\varepsilon \lesssim \sqrt{\frac{\log N}{N}} + \frac{n}{L}$

prediction risk $\propto 1/\sqrt{N}$ (up to log factor)

Key ideas under our construction

1. **construct universal features:** $\exists\ n$ features $\{\phi_j^{\mathsf{feature}}(\boldsymbol{x})\}_{1\le j\le n}$ s.t.: for every $f \in \mathcal{F}$ and $\boldsymbol{x}$, one can express

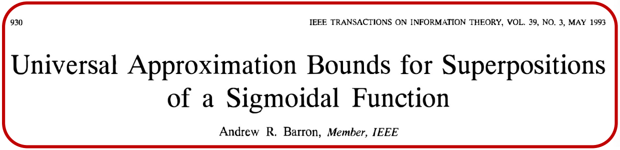
$$f(\boldsymbol{x}) \approx f(\mathbf{0}) + \underbrace{\frac{1}{n}\sum_{j=1}^{n} \rho_{f,j}^{\star}\phi_j^{\mathsf{feature}}(\boldsymbol{x})}_{\text{linear representation over features}} \qquad \text{w/ small } \|\boldsymbol{\rho}_f^{\star}\|_1$$

Key ideas under our construction

1. **construct universal features:** $\exists\, n$ features $\{\phi_j^{\text{feature}}(\boldsymbol{x})\}_{1\le j\le n}$ s.t.: for every $f \in \mathcal{F}$ and $\boldsymbol{x}$, one can express

$$f(\boldsymbol{x}) \approx f(\mathbf{0}) + \underbrace{\frac{1}{n}\sum_{j=1}^{n} \rho^\star_{f,j}\,\phi_j^{\text{feature}}(\boldsymbol{x})}_{\text{linear representation over features}} \qquad \text{w/ small } \|\boldsymbol{\rho}^\star_f\|_1$$

   - insight borrowed from Barron theory: use sigmoid functions


930 IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 39, NO. 3, MAY 1993

Universal Approximation Bounds for Superpositions of a Sigmoidal Function

Andrew R. Barron, *Member, IEEE*

Key ideas under our construction

1. **construct universal features:** $\exists\ n$ features $\{\phi_j^{\text{feature}}(\boldsymbol{x})\}_{1\le j\le n}$ s.t.: for every $f \in \mathcal{F}$ and $\boldsymbol{x}$, one can express

$$f(\boldsymbol{x}) \approx f(\mathbf{0}) + \underbrace{\frac{1}{n}\sum_{j=1}^{n} \rho^\star_{f,j}\phi_j^{\text{feature}}(\boldsymbol{x})}_{\text{linear representation over features}} \qquad \text{w/ small } \|\boldsymbol{\rho}^\star_f\|_1$$

2. **learn $\rho^\star_f$ by solving Lasso**

$$\underset{\boldsymbol{\rho}\in\mathbb{R}^{n+1}}{\text{minimize}} \quad \frac{1}{N}\sum_{i=1}^{N}(y_i - \boldsymbol{\phi}^{\text{feature}}(\boldsymbol{x}_i)^\top \boldsymbol{\rho})^2 + \lambda\|\boldsymbol{\rho}\|_1$$

Key ideas under our construction

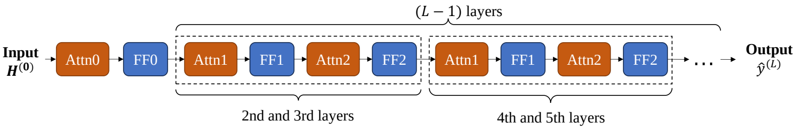
3. **solve Lasso via proximal gradient methods:**

$$\boldsymbol{\rho} \;\leftarrow\; \text{soft-thresh}\Big(\boldsymbol{\rho} + \frac{2\eta}{N}\sum_{i=1}^{N}\big(y_i - \boldsymbol{\phi}^{\text{feature}}(\boldsymbol{x}_i)^\top\boldsymbol{\rho}\big)\boldsymbol{\phi}^{\text{feature}}(\boldsymbol{x}_i)\Big)$$

Key ideas under our construction

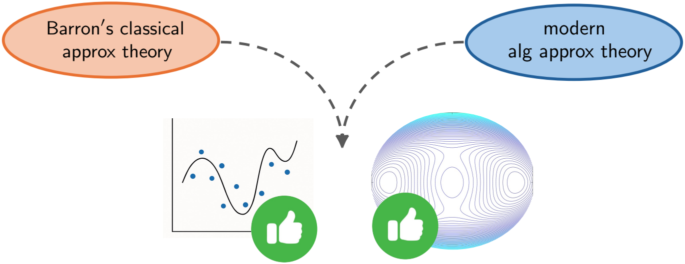
3. **solve Lasso via proximal gradient methods:**

$$\boldsymbol{\rho} \leftarrow \text{soft-thresh}\Big(\boldsymbol{\rho} + \frac{2\eta}{N}\sum_{i=1}^{N}\big(y_i - \boldsymbol{\phi}^{\text{feature}}(\boldsymbol{x}_i)^\top \boldsymbol{\rho}\big)\boldsymbol{\phi}^{\text{feature}}(\boldsymbol{x}_i)\Big)$$



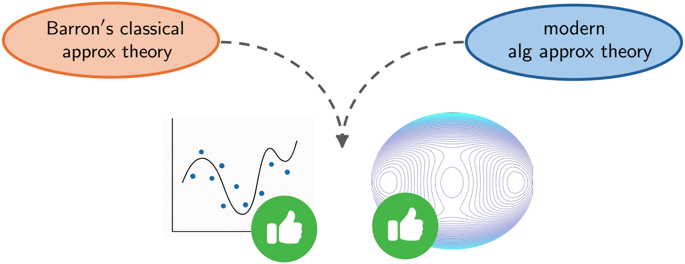
4. **build transformers to approximate prox grad iterations**
   - insight borrowed from prior ICL approximation theory (i.e., transformers as algorithm approximator)

Concluding remarks



- A universal function approximation theory for in-context learning
- Extends far beyond linear functions / convex settings

Concluding remarks



- A universal function approximation theory for in-context learning
- Extends far beyond linear functions / convex settings

future direction: understand training dynamics?

"Transformers Meet In-Context Learning: A Universal Approximation Theory," G. Li, Y. Jiao, Y. Huang, Y. Wei, Y. Chen, `arXiv:2506.05200`, 2025.