Simons Institute for the Theory of Computing

Berkeley UNIVERSITY OF CALIFORNIA

In-context learning

of linear regression

Jingfeng Wu, 10/2023
w/ Difan Zou, Zixiang Chen, Vladimir Braverman, Quanquan Gu, Peter Bartlett

In-context learning

```
>>> x = np.random.randn(5)
>>> y = x*1.7+np.random.randn(5)*0.01
>>> x
array([-0.6615554 ,  0.58568541, -0.06671855, -0.47585609,  0.74095414])
>>> y
array([-1.12766266,  0.98464498, -0.1269867 , -0.80137645,  1.24708834])
>>>
```

Given a pretrained model,

solving new task by

- fine-tuning (backward passes)
- or **just a forward pass** (ICL)

JW

-0.6615554 => -1.12766266,
0.58568541 => 0.98464498,
-0.06671855 => -0.1269867,
-0.47585609 => -0.80137645,
0.74095414 => ?

It looks like you're trying to find a pattern or rule in a sequence of numbers. Let's analyze the given transformations to see if we can apply the same rule to the number 0.74095414:

Original Number (X) => Transformed Number (Y)
-0.6615554 => -1.12766266
0.58568541 => 0.98464498
-0.06671855 => -0.1269867
-0.47585609 => -0.80137645

It appears that the transformation applied is to multiply the original number by approximately 1.70507279 (rounded to 8 decimal places).

So, to find the transformed number for 0.74095414, you can apply the same rule:

0.74095414 * 1.70507279 ≈ 1.26342623

So, the transformed number for 0.74095414 is approximately 1.26342623.

Linear regression with a Gaussian prior

$$X \in \mathbb{R}^{*\times d}, \quad Y \in \mathbb{R}^{*}, \quad x \in \mathbb{R}^{d}, \quad y \in \mathbb{R}$$

1. draw **task** parameter: $\beta \sim \mathcal{N}(0,\ \psi^2 I_d)$
2. draw covariate-response: $x \sim \mathcal{N}(0,\ H),\ y \sim \mathcal{N}(\beta^\top x,\ \sigma^2)$
3. draw context examples: each row of $(X,\ Y)$ ~iid~ $(x^\top,\ y)$

(ψ^2, σ^2, H) are fixed (determining the meta-task)

context length can vary

Garg, Shivam, Dimitris Tsipras, Percy S. Liang, and Gregory Valiant. "What can transformers learn in-context? a case study of simple function classes." Advances in Neural Information Processing Systems 35 (2022): 30583-30598.

Model for (weak) ICL

$$f : \mathbb{R}^{*\times d} \otimes \mathbb{R}^{*} \otimes \mathbb{R}^{d} \to \mathbb{R}$$

$$(X, Y, x) \mapsto \hat{y}$$

example 2: single layer attention

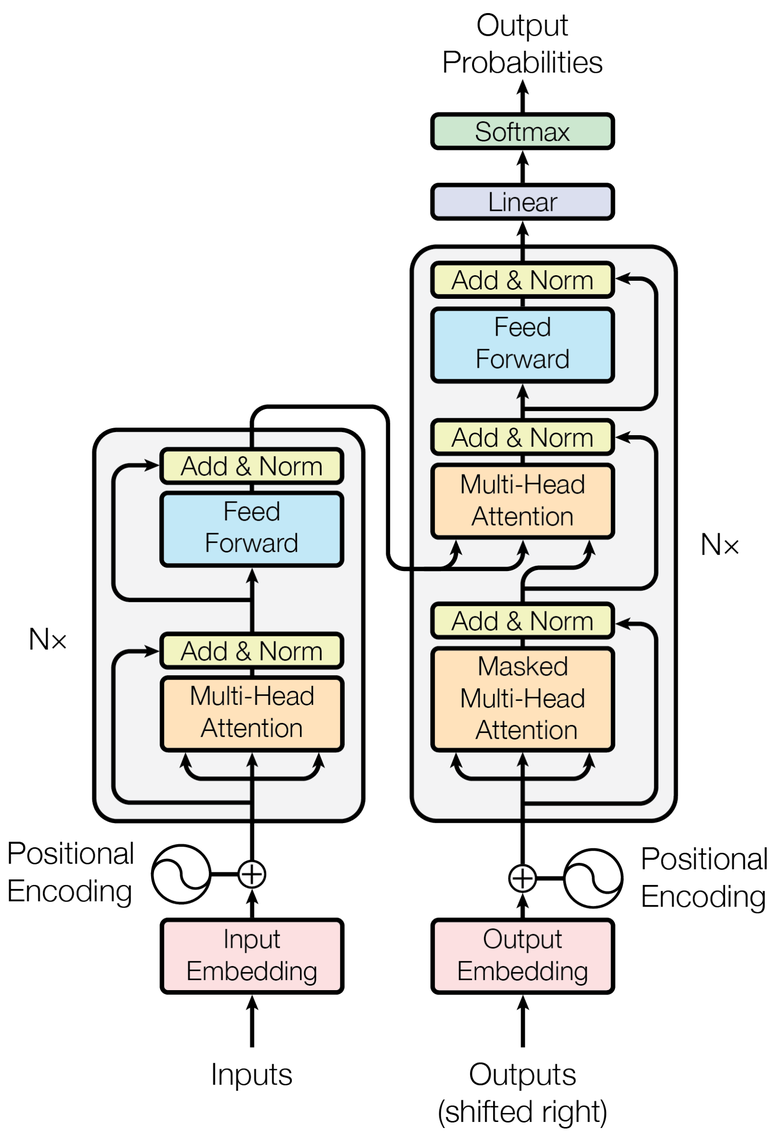
$$Z = \begin{pmatrix} X^\top & x \\ Y^\top & 0 \end{pmatrix} \in \mathbb{R}^{(d+1)\times(n+1)},$$

$$\hat{y} = \left(Z + VZ \cdot \mathtt{sfmx}\left((QZ)^\top (KZ) \right) \right)_{d+1,n+1}$$

V, Q, K are trainable matrix params

example 3: ERM (no param to pretrain)

example 1: transformer



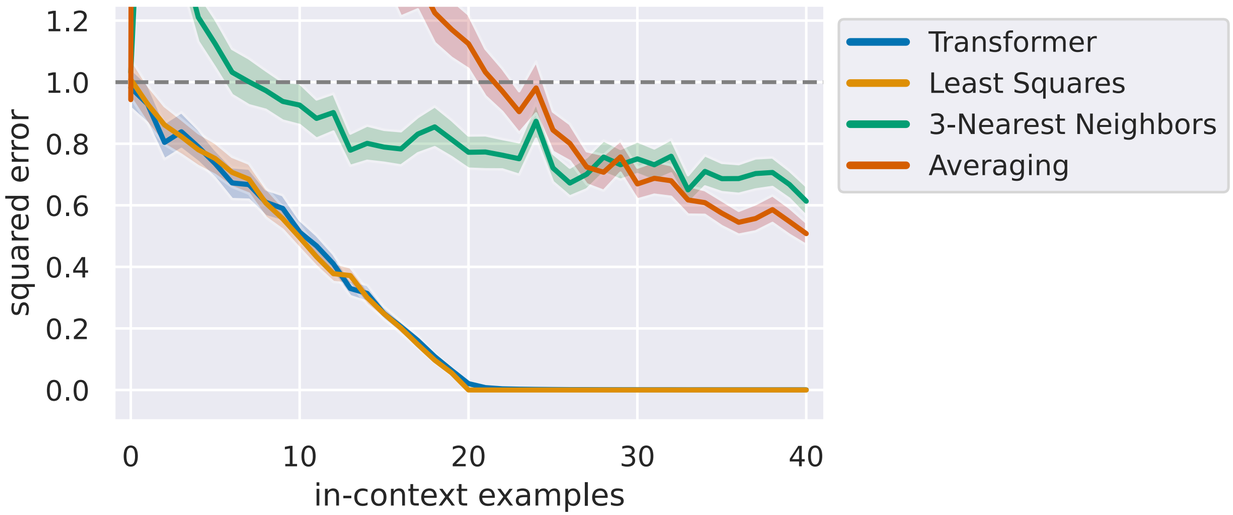
Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

Evidence of ICL by a pretrained transformer

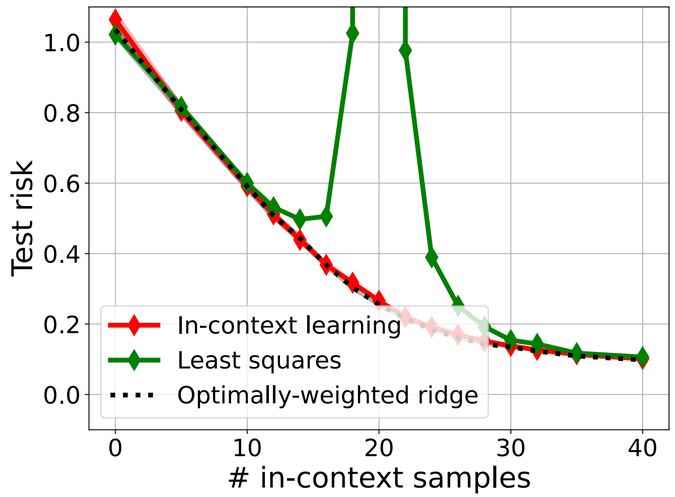
Fixing model f, its **ICL risk at context length** n is

$$f : \mathbb{R}^{*\times d} \otimes \mathbb{R}^{*} \otimes \mathbb{R}^{d} \to \mathbb{R}$$
$$(X, Y, x) \mapsto \hat{y}$$

$$R_n(f) = \mathbb{E}_{X,Y,x,y}\big(f(X, Y, x) - y\big)^2, \text{ where } \mathtt{dim}(Y) = n$$



$H = I_{20},\ \sigma^2 = 0,\ \psi^2 = 1$



$H = I_{20},\ \sigma^2 = 1,\ \psi^2 = 1$

$$\hat{y}_{\mathtt{ols}} = \langle (X^\top X)^{-1} X^\top Y,\ x \rangle$$

$$\hat{y}_{\mathtt{ridge}} = \langle (X^\top X + \sigma^2/\psi^2 \cdot I)^{-1} X^\top Y,\ x \rangle$$

Garg, Shivam, Dimitris Tsipras, Percy S. Liang, and Gregory Valiant. "What can transformers learn in-context? a case study of simple function classes." Advances in Neural Information Processing Systems 35 (2022): 30583-30598.

Akyürek, Ekin, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. "What learning algorithm is in-context learning? investigations with linear models." *arXiv preprint arXiv:2211.15661* (2022).

Li, Yingcong, Muhammed Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. "Transformers as algorithms: Generalization and stability in in-context learning." (2023).

Online pretraining

$$f : \mathbb{R}^{*\times d} \otimes \mathbb{R}^* \otimes \mathbb{R}^d \to \mathbb{R}$$
$$(X, Y, x) \mapsto \hat{y}$$

repeat:

1. draw context length $n \in \{1, \ldots, N\}$
2. draw new dataset of size $n+1$: $X \in \mathbb{R}^{n\times d}, Y \in \mathbb{R}^n, x \in \mathbb{R}^d, y \in \mathbb{R}$
3. update: $f \leftarrow f - \gamma \cdot \nabla_f \left(f(X, Y, x) - y\right)^2$

in their experiments: autoregressive loss, ADAM instead of SGD

Garg, Shivam, Dimitris Tsipras, Percy S. Liang, and Gregory Valiant. "What can transformers learn in-context? a case study of simple function classes." Advances in Neural Information Processing Systems 35 (2022): 30583-30598.
Akyürek, Ekin, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. "What learning algorithm is in-context learning? investigations with linear models." *arXiv preprint arXiv:2211.15661* (2022).
Li, Yingcong, Muhammed Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. "Transformers as algorithms: Generalization and stability in in-context learning." (2023).

Some theory about ICL

Simplification 1: linear attention

$$f : \mathbb{R}^{*\times d} \otimes \mathbb{R}^{*} \otimes \mathbb{R}^{d} \to \mathbb{R}$$
$$(X, Y, x) \mapsto \hat{y}$$

single layer attention

$$\hat{y} = \left(Z + VZ \cdot \mathtt{sfmx}\left((QZ)^\top (KZ) \right) \right)_{d+1,n+1} \qquad Z = \begin{pmatrix} X^\top & x \\ Y^\top & 0 \end{pmatrix} \in \mathbb{R}^{(d+1)\times(n+1)}$$

single layer **linear** attention

$$\hat{y} = \left(Z + VZ \cdot \frac{(QZ)^\top (KZ)}{n} \right)_{d+1,n+1}$$

Ahn, Kwangjun, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. "Transformers learn to implement preconditioned gradient descent for in-context learning." *arXiv preprint arXiv:2306.00297* (2023).
Zhang, Ruiqi, Spencer Frei, and Peter L. Bartlett. "Trained Transformers Learn Linear Models In-Context." *arXiv preprint arXiv:2306.09927* (2023).

Simplification 2: reparameterization

$$f : \mathbb{R}^{*\times d} \otimes \mathbb{R}^{*} \otimes \mathbb{R}^{d} \to \mathbb{R}$$
$$(X, Y, x) \mapsto \hat{y}$$

single layer linear attention

$$\hat{y} = \left(Z + VZ \cdot \frac{(QZ)^\top (KZ)}{n} \right)_{d+1,n+1} \qquad Z = \begin{pmatrix} X^\top & x \\ Y^\top & 0 \end{pmatrix} \in \mathbb{R}^{(d+1)\times(n+1)}$$

equals to $\hat{y} = \left\langle \boxed{\left(vW^\top\right)} \cdot \frac{X^\top Y}{n}, x \right\rangle$

replace by $\Gamma \in \mathbb{R}^{d\times d}$

if the bottom left $1 \times d$ blocks in V and $QK^\top$ are fixed to zeros:

$$V = \begin{pmatrix} * & * \\ 0 & v \end{pmatrix} \text{ and } QK^\top = \begin{pmatrix} W & * \\ 0 & * \end{pmatrix}$$

Ahn, Kwangjun, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. "Transformers learn to implement preconditioned gradient descent for in-context learning." *arXiv preprint arXiv:2306.00297* (2023).
Zhang, Ruiqi, Spencer Frei, and Peter L. Bartlett. "Trained Transformers Learn Linear Models In-Context." *arXiv preprint arXiv:2306.09927* (2023).

~= one-step GD

$$f : \mathbb{R}^{*\times d} \otimes \mathbb{R}^{*} \otimes \mathbb{R}^{d} \to \mathbb{R}$$
$$(X, Y, x) \mapsto \hat{y}$$

single layer linear attention with re-parameterization

$$\hat{y} = \left\langle \Gamma \cdot \frac{X^\top Y}{n}, x \right\rangle, \Gamma \in \mathbb{R}^{d\times d}$$

one-step GD with $w_0 = 0$ and trainable matrix stepsize

$$\hat{y} = \langle \hat{w}, x \rangle, \hat{w} = w_0 - \Gamma \cdot \frac{1}{n} X^\top (X w_0 - Y)$$

intuition: attention ~= kernel ~= Gram matrix

Ahn, Kwangjun, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. "Transformers learn to implement preconditioned gradient descent for in-context learning." *arXiv preprint arXiv:2306.00297* (2023).
Zhang, Ruiqi, Spencer Frei, and Peter L. Bartlett. "Trained Transformers Learn Linear Models In-Context." *arXiv preprint arXiv:2306.09927* (2023).

Simplification 3: fixed context length

repeat:

1. **fix context length** $n = N$
2. draw new dataset of size $N + 1$: $X \in \mathbb{R}^{N \times d}, Y \in \mathbb{R}^N, x \in \mathbb{R}^d, y \in \mathbb{R}$
3. update: $\Gamma \leftarrow \Gamma - \gamma \cdot \nabla_\Gamma \left(\hat{y} - y\right)^2$

$$\hat{y} = \left\langle \Gamma \cdot \frac{X^\top Y}{N}, x \right\rangle$$

pretraining risk (i.e., ICL risk **at context length** N)

$$R_N(\Gamma) = \mathbb{E}_{X,Y,x,y} \left(\hat{y} - y\right)^2, \text{ where } \mathtt{dim}(Y) = N$$

Pretraining => d^2-dim linear fitting

$$R_N(\Gamma) = \mathbb{E}_{X,Y,x,y}\left(\hat{y} - y\right)^2, \text{ where } \mathtt{dim}(Y) = N \qquad \hat{y} = \left\langle \Gamma \cdot \frac{X^\top Y}{N}, x \right\rangle$$

linearly fit $\left(\frac{X^\top Y}{N} \otimes x,\ y\right)$ with a **matrix parameter** $\Gamma \in \mathbb{R}^{d \times d}$

can we use less than d^2 observations (pretraining tasks)?

Quick facts

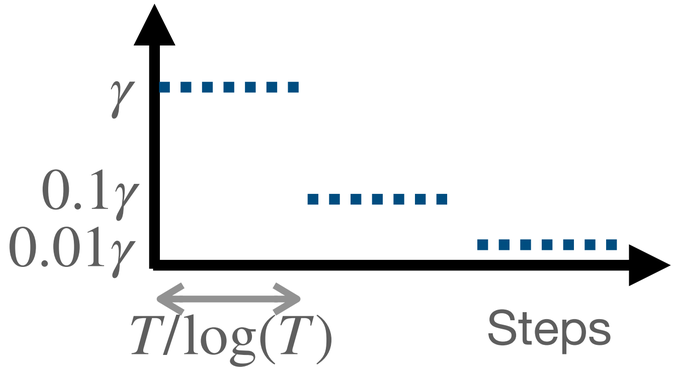
$$\hat{y} = \left\langle \Gamma \cdot \frac{X^\top Y}{N}, x \right\rangle$$

$$R_N(\Gamma) = \mathbb{E}_{X,Y,x,y}\left(\hat{y} - y\right)^2, \text{ where } \mathtt{dim}(Y) = N$$

- $\Gamma^* = \left(\frac{N+1}{N}H + \frac{\mathtt{tr}(H) + \sigma^2/\psi^2}{N}I\right)^{-1} \approx \left(H + \frac{1}{N}I\right)^{-1}$ $\qquad H = \mathbb{E}[xx^\top]$
- $R(\Gamma) - \min R = \left\langle H, (\Gamma - \Gamma^*)\tilde{H}(\Gamma - \Gamma^*)^\top \right\rangle$
- $\tilde{H} = \psi^2 \cdot H\left(\frac{N+1}{N}H + \frac{\mathtt{tr}(H) + \sigma^2/\psi^2}{N}I\right) \approx \psi^2 H\left(H + \frac{1}{N}I\right)$
- $\min R - \sigma^2 = \psi^2 \cdot \mathtt{tr}\left(\Gamma^* H\left(\frac{\mathtt{tr}(H) + \sigma^2/\psi^2}{N} \cdot I + \frac{1}{N}H\right)\right) \lesssim \frac{\psi^2 \mathtt{tr}(H) + \sigma^2}{N}$

$\|\Gamma^*\|_F^2$ could be $\propto N^2 d$ if H has many small eigenvalues (e.g., zeros)

Main result 1



Theorem. For T steps of pretraining, we have

$$\mathbb{E}R_N(\Gamma_T) - \min R_N \lesssim \left\langle H\tilde{H}, \left(\prod_{t=1}^{T}(I - \gamma_t H\tilde{H})\Gamma^*\right)^2 \right\rangle + \left(\psi^2 \mathtt{tr}(H) + \sigma^2\right) \cdot \frac{D_{\mathtt{eff}}}{T_{\mathtt{eff}}}$$

$$T_{\mathtt{eff}} = T/\log(T), \quad D_{\mathtt{eff}} = \sum_{1 \le i,j \le d} \min\left\{1, \gamma^2 T_{\mathtt{eff}}^2 \lambda_i^2 \tilde{\lambda}_j^2\right\}$$

λ_i and $\tilde{\lambda}_j$ are eigenvalues of H and $\tilde{H}$

Corollary. Pretraining enables weak ICL

$$\mathbb{E}R_N(\Gamma_T) = \sigma^2 + \boxed{\min R_N - \sigma^2} + \boxed{\mathbb{E}R_N(\Gamma_T) - \min R_N}$$

$\to 0$ as $N \to \infty$ $\quad$ $\to 0$ as $T \to \infty$

yay, we match ERM

Bayesian optimal during inference

$$f : \mathbb{R}^{*\times d} \otimes \mathbb{R}^* \otimes \mathbb{R}^d \to \mathbb{R}$$
$$(X, Y, x) \mapsto \hat{y}$$

Fixing model f, its average risk (conditional on X) **at length** M is

$$R_M(f; X) = \mathbb{E}_{Y,x,y}\big(f(X, Y, x) - y\big)^2, \text{ where } \mathtt{dim}(Y) = M$$

Proposition. Tuned ridge is Bayesian optimal

$$\hat{y} = \left\langle (X^\top X + \sigma^2/\psi^2 I)^{-1} X^\top Y,\ x \right\rangle$$

Moreover, if $\psi^2 \mathtt{tr}(H) \lesssim \sigma^2$, then the average risk (whp) is

$$R_M(\mathtt{ridge}; X) - \sigma^2 \eqsim \psi^2 \cdot \sum_i \min\{\lambda_i,\ \mu_M\}, \text{ where } \mu_M \eqsim \frac{\sigma^2/\psi^2}{M}$$

Tsigler, Alexander, and Peter L. Bartlett. "Benign overfitting in ridge regression." *J. Mach. Learn. Res.* 24 (2023): 123-1.

Main result 2

Pretrain **at length** N

$$\hat{y} = \left\langle \Gamma^* \cdot \frac{X^\top Y}{\mathtt{dim}(Y)}, x \right\rangle \text{ where } \Gamma^* \approx \left(H + \frac{\mathtt{tr}(H) + \sigma^2/\psi^2}{N} I \right)^{-1}$$

Theorem. If $\psi^2 \mathtt{tr}(H) \lesssim \sigma^2$, then the average risk **at length** M is

$$\mathbb{E} R_M(f; X) - \sigma^2 \approx \boxed{\psi^2 \cdot \sum_i \min\{\lambda_i, \mu_M\}}$$

ridge risk

$$+\psi^2 \boxed{(\mu_M - \mu_N)^2} \cdot \sum_i \min\left\{ \frac{\lambda_i}{\mu_N^2}, \frac{1}{\lambda_i} \right\} \cdot \min\left\{ \frac{\lambda_i}{\mu_M}, 1 \right\}$$

where $\mu_M \approx \dfrac{\sigma^2/\psi^2}{M}$, $\mu_N \approx \dfrac{\sigma^2/\psi^2}{N}$.

We match optimal ridge if M=N

$$f : \mathbb{R}^{*\times d} \otimes \mathbb{R}^* \otimes \mathbb{R}^d \to \mathbb{R}$$

$$(X, Y, x) \mapsto \hat{y}$$

Four levels of ICL by pretraining

1. "**consistent**" — **ERM, nothing to learn**
   - error -> min error when $M, N \to \infty$
2. "**one-point optimal**" — **one hyperparameter, one thing to learn**
   - error rate is optimal when $M = N$

supervised learning

real in-context learning

3. "**uniformly optimal**" / algorithm selection — **N hyperparameters log(N) things to learn**
   - error rate is optimal for every $M \leq N$
4. "**generalizable optimal**" / algorithm learning — **a rule to learn**
   - error rate is optimal for both $M \leq N$ and $M > N$

Where are we

$$f : \mathbb{R}^{*\times d} \otimes \mathbb{R}^{*} \otimes \mathbb{R}^{d} \to \mathbb{R}$$
$$(X, Y, x) \mapsto \hat{y}$$

1. "**consistent**"
   - error -> min error when $M, N \to \infty$
2. "**one-point optimal**"
   - error rate is optimal when $M = N$
3. "**uniformly optimal**" / algorithm selection
   - error rate is optimal for every $M \leq N$
4. "**generalizable optimal**" / algorithm learning
   - error rate is optimal for both $M \leq N$ and $M > N$

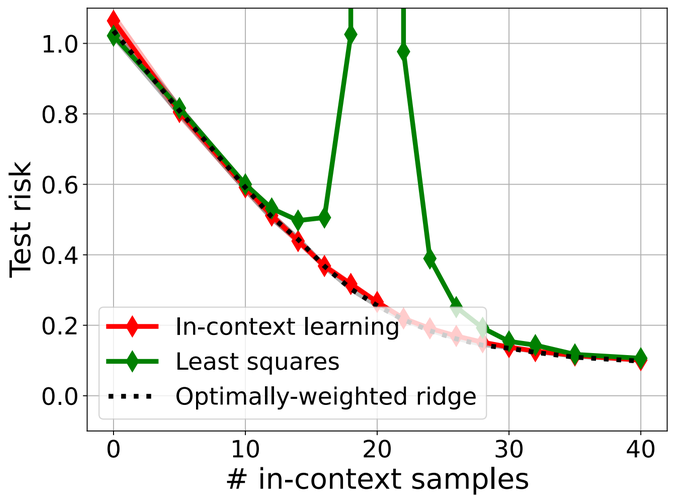
theory <= level 2

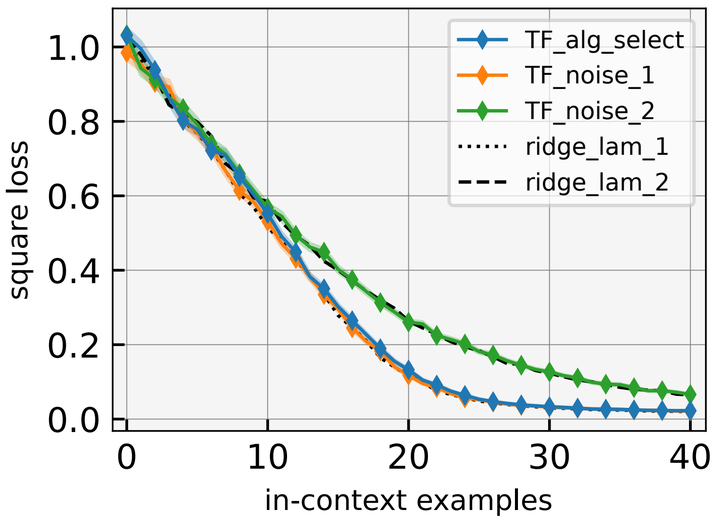
supervised learning

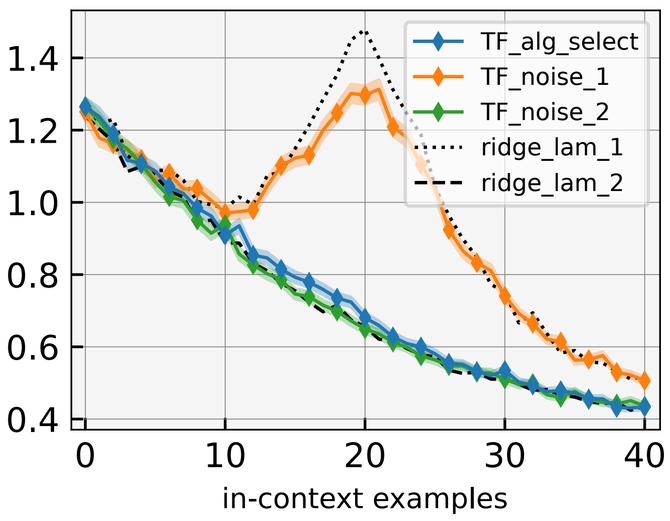
real in-context learning

practice >= level 3

Evidences for “practice >= level 3”







Li, Yingcong, Muhammed Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. "Transformers as algorithms: Generalization and stability in in-context learning." (2023).
Bai, Yu, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. "Transformers as Statisticians: Provable In-Context Learning with In-Context Algorithm Selection." *arXiv preprint arXiv:2306.04637* (2023).

Takes

- task complexity of pretraining
- four levels of ICL
- a theory of ICL at level 2

Open questions

- many “obvious” questions
- two less obvious questions
  - can we do ICL theory at level 3?
  - can transformer do ICL at level 4?