Trained Transformers Learn Linear Models In-Context

Haojun Wu

June 2025

Table of Contents

Table of Contents

Background introdution

Garg et al. [Gar+22] showed transformer models trained on prompts from a particular function class (e.g., linear models, neural networks, or decision trees), they succeed at in-context learning, and the behavior of the trained transformers can mimic those of familiar learning algorithms like ordinary least squares.

the model is trained on prompts $(x_1, h(x_1), \dots, x_N, h(x_N), x_{\text{query}})$ where $x_i, x_{\text{query}} \overset{i.i.d}{\sim} \mathcal{D}_x$ and $h \in \mathcal{H} \sim$ *a distribution* Δ. The transformer succeeds at in-context learning when given a new prompt $(x'_1, h'(x'_1), \dots, x'_N, h'(x'_N), x'_{\text{query}})$ where h' may not belong to training function class $\mathcal{H}$. formulate a prediction for x'_{query} that is close to $h'(x'_{\text{query}})$

It leaves open the question of how it is that gradient-based optimization algorithms over transformer architectures produce models which are capable of in-context learning.

In this work, we investigate the learning dynamics of gradient flow in a simplified transformer architecture when the training prompts consists of random instances of linear regression datasets.

Table of Contents

Notation

We write $[n] = 1, 2, ..., n$. We use $\otimes$ to denote the Kronecker product, and Vec the vectorization operator in column-wise order.

Examples

$$Vec(\begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix}) = (1, 2, 3, 4)^T$$

We write the inner product of two matrices $A, B \in R^{m \times n}$ as

$$< A, B >= tr(AB^T)$$

We use 0_n and $0_{m \times n}$ to denote the zero vector and zero matrix of size n and $m \times n$

For a general matrix A, $A_{k:}$ and $A_{:k}$ denote the k-th row and k-th column, respectively. We denote the matrix operator norm and Frobenius norm as $\| \cdot \|_{op}$ and $\| \cdot \|_F$.

Notation

Remark

the $m \times n$ matrix A operator norm and Frobenius norm as $\|\cdot\|_{op}$ and $\|\cdot\|_F$.

$$\|A\|_{op} = \sup_{\|x\| \leq 1, x \in R^n} \|Ax\|$$

$$\|A\|_F = \sqrt{tr(AA^T)}$$

For a positive semi-definite matrix A, we write $\|x\|_A^2 := x^T A x$

framework for in-context learning of function classes

The goal for an in-context learner is to use the prompt to form a prediction $\widehat{y}(x_{query})$ for the query such that $\widehat{y}(x_{query}) \approx h(x_{query})$.

Examples

one can view ordinary least squares as an 'in-context learner' for linear models.

$$\textit{given} \quad (x_1, y_1(= w^T x_1 + \epsilon_1), x_2, y_2(= w^T x_2 + \epsilon_2), ..., x_N, y_N, x_{query})$$

ordinary least squares gives an estimate $\widehat{w}$ of w,and x_{query}'s prediction $\widehat{y}(x_{query}) = \widehat{w}^T x_{query}$

We formalize the training loss and train objective in the following definition

framework for in-context learning of function classes

Definition (Trained on in-context examples)

Let $\mathcal{D}_x$ be a distribution over an input space $\mathcal{X}$, $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ a set of functions $\mathcal{X} \to \mathcal{Y}$, and $\mathcal{D}_{\mathcal{H}}$ a distribution over functions in $\mathcal{H}$. Let $\mathcal{S} = \{(x_1, y_1, \ldots, x_n, y_n) : x_i \in \mathcal{X}, y_i \in \mathcal{Y}\}$ be the set of finite-length sequences of (x, y) pairs and let

$$\mathcal{F}_\Theta = \{f_\theta : \mathcal{S} \times \mathcal{X} \to \mathcal{Y}, \theta \in \Theta\}$$

be a class of functions parameterized by θ (model functions). For $N > 0$, training Goal on the length N prompts:

$$\theta^* \in argmin_{\theta \in \Theta} \mathbb{E}_{P=(x_1, h(x_1), \ldots, x_N, h(x_N), x_{\text{query}})} \left[\ell \left(f_\theta(P), h(x_{\text{query}}) \right) \right], \tag{3.1}$$

where $x_i, x_{\text{query}} \overset{\text{i.i.d.}}{\sim} \mathcal{D}_x$ and $h \sim \mathcal{D}_{\mathcal{H}}$ are independent.

framework for in-context learning of function classes

Remark

[a learning algorithm from data:] Sample independent prompts by sampling a random function $h \sim \mathcal{D}_{\mathcal{H}}$ and feature vectors $x_i, x_{\text{query}} \overset{\text{i.i.d.}}{\sim} \mathcal{D}_x$, and then minimize the objective function appearing in (3.1) using stochastic gradient descent or other stochastic optimization algorithms.

This procedure returns a model that is learned from in-context examples and achieves some degree of generalization.

We quantifies how well such a model performs on in-context examples.

In-context learning of a hypothesis class

Definition (In-context learning of a hypothesis class)

a model $f : \mathcal{S} \times \mathcal{X} \to \mathcal{Y}$ in-context learns a hypothesis class $\mathcal{H}$ on $(\mathcal{D}_{\mathcal{H}}, \mathcal{D}_x)$ up to error $\eta \in \mathbb{R}$ if there exists $M_{\mathcal{D}_{\mathcal{H}},\mathcal{D}_x}(\varepsilon)$ such that for every $\varepsilon \in (0,1)$, and for every prompt P of length $M \geq M_{\mathcal{D}_{\mathcal{H}},\mathcal{D}_x}(\varepsilon)$,

$$\mathbb{E}_{P=(x_1,h(x_1),\ldots,x_M,h(x_M),x_{\text{query}})}\left[\ell\left(f(P), h(x_{\text{query}})\right)\right] \leq \eta + \varepsilon, \tag{3.2}$$

where the expectation taken $x_i, x_{\text{query}} \overset{\text{i.i.d.}}{\sim} \mathcal{D}_x$ and $h \sim \mathcal{D}_{\mathcal{H}}$.

The additive error term η may be noise.

What would we do in this ppt

With these two definitions in hand, we can formulate the following questions.

1. Can a model from $\mathcal{F}_\Theta$ that is trained on in-context examples of functions in $\mathcal{H}$ w.r.t. $(\mathcal{D}_\mathcal{H}, \mathcal{D}_x)$ in-context learn the hypothesis class $\mathcal{H}$ w.r.t. $(\mathcal{D}_\mathcal{H}, \mathcal{D}_x)$ with small prediction error?
2. Do standard gradient-based optimization algorithms suffice for training the model from in-context examples?
3. How long must the contexts be during training and at test time to achieve small prediction error?

In the remaining sections, we shall answer these questions.
for the case of f being one-layer transformers with linear self-attention modules when the hypothesis class is linear models $\mathcal{H}$

Linear self-attention networks

we first recall the definition of the softmax-based single-head self-attention module.

$$f_{\text{Attn}}(E; W^K, W^Q, W^V, W^P) = E + W^P W^V E \cdot \text{softmax}\left(\frac{(W^K E)^\top W^Q E}{\rho}\right)$$

where $\rho > 0$ a normalization factor
In particular, we consider a single-layer linear self-attention (LSA) model, yet it is still capable of in-context learning linear models

$$f_{\text{LSA}}(E; \theta) = E + W^{PV} E \cdot \left(\frac{E^\top W^{KQ} E}{\rho}\right), \theta = (W^{PV}, W^{KQ}) \qquad (3.3)$$

Remark

It is noteworthy that recent empirical work shows that state-of-the-art trained vision transformers with standard softmax-based attention modules are such that $(W^K)^T W^Q$ and $W^P W^V$ are nearly multiples of the identity matrix [TK23], which can be represented under the parameterization we consider.

Linear self-attention networks

Embedding matrix E used in this work

$$E = E(P) = \begin{pmatrix} x_1 & x_2 & \cdots & x_N & x_{\text{query}} \\ y_1 & y_2 & \cdots & y_N & 0 \end{pmatrix} \in \mathbb{R}^{(d+1)\times(N+1)}. \quad (3.4)$$

The network's prediction for the token x_{query} will be the bottom-right entry of matrix output by f_{LSA}, namely,

$$\widehat{y}_{\text{query}} = \widehat{y}_{\text{query}}(E;\theta) = [f_{\text{LSA}}(E;\theta)]_{(d+1),(N+1)}. \quad (1)$$

with LSA model $f_{\text{LSA}}(E;\theta) = E + W^{PV}E \cdot \left(\frac{E^\top W^{KQ} E}{\rho}\right), \theta = (W^{PV}, W^{KQ})$ we can do training on it.

we only consider the task of in-context learning linear predictors. Training prompts are sampled as follows. Let Λ be a positive definite covariance matrix. Each training prompt, indexed by $\tau \in \mathbb{N}$, takes the form of $P_\tau = (x_{\tau,1}, h_\tau(x_{\tau,1}), \ldots, x_{\tau,N}, h_\tau(x_{\tau,N}), x_{\tau,\text{query}})$, where task weights $w_\tau \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_d)$, inputs $x_{\tau,i}, x_{\tau,\text{query}} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Lambda)$, and labels $h_\tau(x) = \langle w_\tau, x \rangle$.

Each prompt's embedding matrix E_τ:

$$E_\tau := \begin{pmatrix} x_{\tau,1} & x_{\tau,2} & \cdots & x_{\tau,N} & x_{\tau,\text{query}} \\ \langle w_\tau, x_{\tau,1} \rangle & \langle w_\tau, x_{\tau,2} \rangle & \cdots & \langle w_\tau, x_{\tau,N} \rangle & 0 \end{pmatrix} \in \mathbb{R}^{(d+1)\times(N+1)}. \tag{2}$$

We denote the prediction of the LSA model on the query label in the task τ as $\widehat{y}_{\tau,\text{query}} = [f_{\text{LSA}}(E_\tau)]_{(d+1),(N+1)}$. The empirical risk over B independent prompts is defined as

$$\widehat{L}(\theta) = \frac{1}{2B} \sum_{\tau=1}^{B} \left(\widehat{y}_{\tau,\text{query}} - \langle w_\tau, x_{\tau,\text{query}} \rangle \right)^2 . \tag{3.7}$$

LSA training

It is natural to consider taking large B of the training population loss. when $B \to \infty$, define:

$$L(\theta) = \lim_{B\to\infty} \widehat{L}(\theta) = \frac{1}{2}\mathbb{E}_{w_\tau, x_{\tau,1},\dots,x_{\tau,N},x_{\tau,\text{query}}}\left[\left(\widehat{y}_{\tau,\text{query}} - \langle w_\tau, x_{\tau,\text{query}}\rangle\right)^2\right]. \tag{3.8}$$

the expectation is taken over $x_{\tau,i}, x_{\text{query}} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,\Lambda)$ and $w_\tau \sim \mathcal{N}(0, I_d)$. Gradient flow captures the behavior of gradient descent with infinitesimal step size and has dynamics given by the following differential equation:

$$\frac{d\theta}{dt} = -\nabla L(\theta) \tag{3.9}$$

Remark

In our main results, we conclude that the gradient flow when $t \to +\infty$ of $L(\theta)$ led to the success of in-context learning the linear predictor of a wide range of distribution.

What would we do in this paper

With these definitions in mind, we come back to the problems we mentioned above.

1. Can a model from $\mathcal{F}_\Theta$ that is trained on in-context examples of functions in $\mathcal{H}$ w.r.t. $(\mathcal{D}_\mathcal{H}, \mathcal{D}_x)$ in-context learn the hypothesis class $\mathcal{H}$ w.r.t. $(\mathcal{D}_\mathcal{H}, \mathcal{D}_x)$ with small prediction error?
2. Do standard gradient-based optimization algorithms suffice for training the model from in-context examples?
3. How long must the contexts be during training and at test time to achieve small prediction error?

Table of Contents

Theorem 4.1 ($L(\theta)$'s Convergence and limits).

define

$$\Gamma := \left(1 + \frac{1}{N}\right)\Lambda + \frac{1}{N}\operatorname{tr}(\Lambda) I_d \in \mathbb{R}^{d\times d}.$$

Suppose the initialization satisfies Assumption below with initialization scale $\sigma > 0$ satisfying $\sigma^2 \|\Gamma\|_{op}\sqrt{d} < 2$, the gradient flow of linear self-attention network f^*_{LSA}(prove PL inequality holds) converges (exponentially about t) to a global minimum of the population loss $L(\theta)$. Moreover, W^{PV} and W^{KQ} converge respectively to

$$W^{KQ}_* = \left[\operatorname{tr}\left(\Gamma^{-2}\right)\right]^{-\frac{1}{4}} \begin{pmatrix} \Gamma^{-1} & 0_d \\ 0_d^\top & 0 \end{pmatrix}, \qquad W^{PV}_* = \left[\operatorname{tr}\left(\Gamma^{-2}\right)\right]^{\frac{1}{4}} \begin{pmatrix} 0_{d\times d} & 0_d \\ 0_d^\top & 1 \end{pmatrix}.$$

Assumption (Initialization). Let $\sigma > 0$ be a parameter, $\Theta \in \mathbb{R}^{d\times d}$ be any matrix satisfying $\|\Theta\Theta^\top\|_F = 1$ and $\Theta\Lambda \neq 0_{d\times d}$. We assume

$$W^{PV}(0) = \sigma \begin{pmatrix} 0_{d\times d} & 0_d \\ 0_d^\top & 1 \end{pmatrix}, \quad W^{KQ}(0) = \sigma \begin{pmatrix} \Theta\Theta^\top & 0_d \\ 0_d^\top & 0 \end{pmatrix}. \qquad (3.10)$$

Trained transformer indeed in-context learn linear predictor

At the global optimum f^*_{LSA}, input a test prompt $P = (x_1, y_1, \ldots, x_M, y_M, x_{\text{query}}, y_{\text{query}})$, where $(x_i, y_i), (x_{\text{query}}, y_{\text{query}}) \overset{\text{i.i.d.}}{\sim} \mathcal{D}$ with marginal distribution $x_i, x_{query} \sim \mathcal{D}_x = \mathcal{N}(0, \Lambda)$.

The f^*_{LSA} prediction $\widehat{y}_{\text{query}} = [f^*_{LSA}(E_P; (W^{PV}_*, W^{KQ}_*))]_{(d+1),(M+1)}$ is

$$\begin{pmatrix} 0_d^\top & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{M}\sum_{i=1}^M x_i x_i^\top + \frac{1}{M} x_{\text{query}} x_{\text{query}}^\top & \frac{1}{M}\sum_{i=1}^M x_i y_i \\ \frac{1}{M}\sum_{i=1}^M x_i^\top y_i & \frac{1}{M}\sum_{i=1}^M y_i^2 \end{pmatrix} \begin{pmatrix} \Gamma^{-1} & 0_d \\ 0_d^\top & 0 \end{pmatrix} \begin{pmatrix} x_{\text{query}} \\ 0 \end{pmatrix}$$

$$= x_{\text{query}}^\top \Gamma^{-1} \left(\frac{1}{M} \sum_{i=1}^M y_i x_i \right) . (3)$$

When the length N of training prompts is large, we have $\Gamma^{-1} \approx \Lambda^{-1}$, and when $M \to +\infty$ implies

$$\widehat{y}_{\text{query}} \approx x_{\text{query}}^\top \Lambda^{-1} \mathbb{E}_{(x,y)\sim\mathcal{D}}[yx] = x_{\text{query}}^\top \left(argmin_{w \in \mathbb{R}^d} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[(y - \langle w, x \rangle)^2 \right] \right)$$

for sufficiently large N, the trained transformer indeed in-context learns the class of linear predictors.

Remark

1. f^*_{LSA} can be trained to approximate by training data (takes the form of $P_\tau = (x_{\tau,1}, h_\tau(x_{\tau,1}), \ldots, x_{\tau,N}, h_\tau(x_{\tau,N}), x_{\tau,\text{query}})$, where task weights $w_\tau \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_d)$, inputs $x_{\tau,i}, x_{\tau,\text{query}} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Lambda)$, and labels $h_\tau(x) = \langle w_\tau, x \rangle$.)

Remark

From demonstration $\widehat{y}_{query} = x_{\text{query}}^{\top} \Gamma^{-1} \left(\frac{1}{M} \sum_{i=1}^{M} y_i x_i \right) \approx x_{\text{query}}^{\top} \Lambda^{-1} \mathbb{E}_{(x,y)\sim\mathcal{D}}[yx]$ above, we can know that it still holds for query shifts but covariate shifts not:

Query shifts. Consider $y_i = \langle w, x_i \rangle$, we have

$$\widehat{y}_{\text{query}} \approx x_{\text{query}}^{\top} \Lambda^{-1} \left(\frac{1}{M} \sum_{i=1}^{M} x_i x_i^{\top} \right) w. \quad (4)$$

From this we see that whether query shifts can be tolerated hinges upon the distribution of the x_i's. Since $\mathcal{D}_x^{\text{train}} = \mathcal{D}_x^{\text{test}}$, if M is large then

$$\widehat{y}_{\text{query}} \approx x_{\text{query}}^{\top} \Lambda^{-1} \Lambda w = x_{\text{query}}^{\top} w. \quad (4.8)$$

Thus, very general shifts in the query distribution can be tolerated.

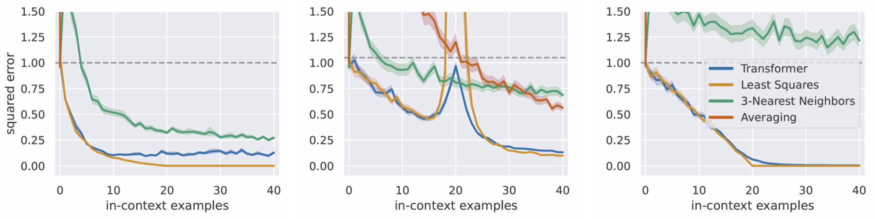
Remark

Covariate shifts. In contrast to query shifts, covariate shifts cannot be fully tolerated. When $\mathcal{D}_x^{\text{train}} \neq \mathcal{D}_x^{\text{test}}$, then the approximation in (4.8) does not hold as $\frac{1}{M}\sum_{i=1}^{M} x_i x_i^\top$ will not cancel Γ^{-1} when M and N are large. For instance, if we consider test prompts where the covariates are scaled by a constant $c \neq 1$, then

$$\widehat{y}_{\text{query}} \approx x_{\text{query}}^\top \Lambda^{-1} \left(\frac{1}{M} \sum_{i=1}^{M} x_i x_i^\top \right) \approx x_{\text{query}}^\top \Lambda^{-1} c^2 \Lambda w = c^2 x_{\text{query}}^\top w \neq x_{\text{query}}^\top w. \tag{5}$$

This failure mode of the trained transformer with linear self-attention was also observed in the trained transformer architectures by Garg et al.[Gar+22]

Behavior of trained transformer under distribution shifts



(a) Skewed covariance (b) Noisy linear regression (c) Different orthants

Figure: In-context learning on out-of-distribution prompts. Garg use isotropic Gaussian while training on standard GPT-2 model using adam optimize. (a) test prompt inputs from a non-isotropic Gaussian (failure), (b) adding label noise to in-context examples, (c) restricting in-context examples to a single (random) orthant.

In all cases, the model error degrades gracefully and remains close to that of the least squares estimator, indicating that its in-context learning ability extrapolates beyond the training distribution.

Thinking

It may seem surprising that a transformer trained on linear regression tasks fails in settings where ordinary least squares performs well.

In the following theorem 4.2, we characterize f^*_{LSA}'s prediction error in theorem 4.1.

Theorem 4.2. transformers in-context learn the best linear predictor

Let $\mathcal{D}$ be a distribution over $(x, y) \in \mathbb{R}^d \times \mathbb{R}$, whose marginal distribution on x is $\mathcal{D}_x = \mathcal{N}(0, \Lambda)$. Assume $\mathbb{E}_{\mathcal{D}}[y]$, $\mathbb{E}_{\mathcal{D}}[xy]$, $\mathbb{E}_{\mathcal{D}}[y^2 x x^\top]$ exist and are finite. If we define $a := \Lambda^{-1}\mathbb{E}_{(x,y)\sim\mathcal{D}}[xy]$, $\quad \Gamma := \Lambda + \frac{1}{N}\Lambda + \frac{1}{N}\operatorname{tr}(\Lambda) I_d$, and $\Sigma := \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[(xy - \mathbb{E}(xy))(xy - \mathbb{E}(xy))^\top\right]$.

f^*_{LSA} be the LSA model in above theorem. Assume the test prompt is of the form $P = (x_1, y_1, \ldots, x_M, y_M, x_{\mathrm{query}})$, where $(x_i, y_i), (x_{\mathrm{query}}, y_{\mathrm{query}}) \overset{\text{i.i.d.}}{\sim} \mathcal{D}$. and $\widehat{y}_{\mathrm{query}} = [f^*_{LSA}(E_P; (W^{PV}_*, W^{KQ}_*))]_{(d+1),(M+1)}$ is the trained LSA model prediction for x_{query} given the prompt. we have:

$$\mathbb{E}\left(\widehat{y}_{\mathrm{query}} - y_{\mathrm{query}}\right)^2 = \underbrace{\min_{w\in\mathbb{R}^d} \mathbb{E}\left(\langle w, x_{\mathrm{query}}\rangle - y_{\mathrm{query}}\right)^2}_{\text{Error of best linear predictor}}$$

$$+ \operatorname{tr}\left[\Sigma\Gamma^{-2}\Lambda\right] + \frac{1}{N^2}\left[\|a\|^2_{\Gamma^{-2}\Lambda^3} + 2\operatorname{tr}(\Lambda)\|a\|^2_{\Gamma^{-2}\Lambda^2} + \operatorname{tr}(\Lambda)^2\|a\|^2_{\Gamma^{-2}\Lambda}\right],$$

where the expectation is over $(x_i, y_i), (x_{\mathrm{query}}, y_{\mathrm{query}}) \overset{\text{i.i.d.}}{\sim} \mathcal{D}$.

a variant of training on in-context examples

We now consider the distribution $\mathcal{D}_x$ is sampled randomly from a distribution Δ.

$$\theta^* \in argmin_{\theta \in \Theta} \mathbb{E}_{P=(x_1, h(x_1), \ldots, x_N, h(x_N), x_{\text{query}})} \left[\ell \left(f_\theta(P), h(x_{\text{query}}) \right) \right], \quad (4.9)$$

where $\mathcal{D}_x \sim \Delta$, $x_i, x_{\text{query}} \overset{\text{i.i.d.}}{\sim} \mathcal{D}_x$ and $h \sim \mathcal{D}_{\mathcal{H}}$.
The population loss now includes an expectation over the distribution of the covariance matrices Λ_τ (random matrices):

$$L(\theta) = \frac{1}{2} \mathbb{E}_{w_\tau, \Lambda_\tau, x_{\tau,1}, \ldots, x_{\tau,N}, x_{\tau,\text{query}}} \left[\left(\widehat{y}_{\tau,\text{query}} - \langle w_\tau, x_{\tau,\text{query}} \rangle \right)^2 \right]. \quad (4.10)$$

the previous definition of training on in-context examples by taking $supp(\Delta) = \{\Lambda\}$. Similarly to Theorem 4.1, we have

Theorem 4.5 (Global convergence in random covariance case). Consider gradient flow over the general population loss (4.10), where Λ_τ are diagonal (convenient for analysis) with independent diagonal entries (random variables) which are strictly positive a.s. and have finite third moments. Suppose the initialization satisfies Assumption, $\|\mathbb{E}\Lambda_\tau\Theta\|_F \neq 0$, with initialization scale $\sigma > 0$ satisfying

$$\sigma^2 < \frac{2\|\mathbb{E}\Lambda_\tau\Theta\|_F^2}{\sqrt{d}\left[\mathbb{E}\|\Gamma_\tau\|_{op}\|\Lambda_\tau\|_F^2\right]}. \tag{4.11}$$

Then gradient flow converges to a global minimum of the population loss. Moreover, W^{PV} and W^{KQ} converge to W_*^{PV} and W_*^{KQ}, where

$$\begin{aligned} W_*^{KQ} &= \left\| \left[\mathbb{E}\Gamma_\tau\Lambda_\tau^2\right]^{-1} \mathbb{E}\left[\Lambda_\tau^2\right] \right\|_F^{-\frac{1}{2}} \cdot \begin{pmatrix} \left[\mathbb{E}\Gamma_\tau\Lambda_\tau^2\right]^{-1}\left[\mathbb{E}\Lambda_\tau^2\right] & 0_d \\ 0_d^\top & 0 \end{pmatrix}, \\ W_*^{PV} &= \left\| \left[\mathbb{E}\Gamma_\tau\Lambda_\tau^2\right]^{-1} \mathbb{E}\left[\Lambda_\tau^2\right] \right\|_F^{\frac{1}{2}} \cdot \begin{pmatrix} 0_{d\times d} & 0_d \\ 0_d^\top & 1 \end{pmatrix}, \end{aligned} \tag{4.12}$$

where $\Gamma_\tau = \frac{N+1}{N}\Lambda_\tau + \frac{1}{N}\operatorname{tr}(\Lambda_\tau) I_d$ and the expectations above are over the distribution of Λ_τ.

From this result, we can see why the trained transformer fails in the random covariance case.

Suppose we have a test prompt corresponding to a weight matrix $w \in \mathbb{R}^d$ and covariance matrix Λ_{new}, and set $\Lambda_\tau \overset{d}{\equiv} \Lambda_{new}$, $x_i, x_{\text{query}} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Lambda_{\text{new}})$, $y_i = \langle w, x_i \rangle$, $i \in [M]$ and $y_{\text{query}} = \langle w, x_{\text{query}} \rangle$. At convergence, the prediction $\widehat{y}_{\text{query}}$ by the trained transformer on the new task will be

$$\begin{pmatrix} 0_d^\top & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{M}\sum_{i=1}^M x_i x_i^\top + \frac{1}{M} x_{\text{query}} x_{\text{query}}^\top & \frac{1}{M}\sum_{i=1}^M x_i y_i \\ \frac{1}{M}\sum_{i=1}^M x_i^\top y_i & \frac{1}{M}\sum_{i=1}^M y_i^2 \end{pmatrix} \begin{pmatrix} \left[\mathbb{E}\Gamma_\tau \Lambda_\tau^2\right]^{-1} \left[\mathbb{E}\Lambda_\tau^2\right] \\ 0_d^\top \end{pmatrix}$$

$$= x_{\text{query}}^\top \cdot \left[\mathbb{E}\Lambda_\tau^2\right] \left[\mathbb{E}\Gamma_\tau \Lambda_\tau^2\right]^{-1} \cdot \left[\frac{1}{M}\sum_{i=1}^M x_i x_i^\top\right] w$$

$$\to x_{\text{query}}^\top \cdot \left[\mathbb{E}\Lambda_\tau^2\right] \left[\mathbb{E}\Gamma_\tau \Lambda_\tau^2\right]^{-1} \cdot \Lambda_{\text{new}} w \quad \text{almost surely when } M \to \infty. \quad (6)$$

When $M, N \to \infty$ so that $\Gamma_\tau \to \Lambda_\tau$. taking expectation over Λ_{new}:

$$\mathbb{E}\left[\widehat{y}_{\text{query}} \mid x_{\text{query}}, w\right] \to x_{\text{query}}^\top \cdot \left[\mathbb{E}\Lambda_\tau^2\right] \left[\mathbb{E}\Lambda_\tau^3\right]^{-1} \cdot \left[\mathbb{E}\Lambda_\tau\right] w. \quad (7)$$

If we consider the case $\lambda_{\tau,i} \overset{\text{i.i.d.}}{\sim}$ Exponential(1), so that $\mathbb{E}[\Lambda_\tau] = I_d$, $\mathbb{E}[\Lambda_\tau^2] = 2I_d$, and $\mathbb{E}[\Lambda_\tau^3] = 6I_d$, we get

$$\mathbb{E}\widehat{y}_{\text{query}} \to \frac{1}{3}\langle w, x_{\text{query}}\rangle. \tag{8}$$

This shows that training on in-context examples with random covariate distributions does not allow for in-context learning of a hypothesis class with varying covariate distributions.

the behavior of more complex transformer architectures

Experiments with large, nonlinear transformers. GPT-2: a large, nonlinear transformer

trained on in-context examples of linear models, both in the fixed-covariance case and in the random-covariance case.

training prompts sample from random independent covariance matrices: $\Lambda_\tau = diag(\lambda_{\tau,1}, ..., \lambda_{\tau,d})$, where $\lambda_{\tau,i} \overset{i.i.d}{\sim} exp(1)$ or fixed matrices: the covariance matrix is fixed to the identity matrix.

test prompts sample from random covariance matrices: $c\Lambda = diag(c\lambda_1, ..., c\lambda_d)$, where $\lambda_i \overset{i.i.d}{\sim} exp(1)$, and $c > 0$ is a scaling factor or fixed matrices: the covariance matrix is fixed to the identity matrix.

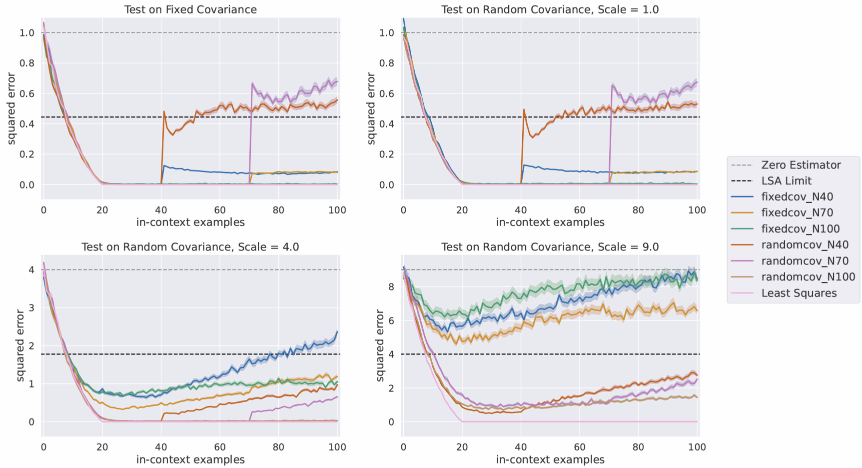
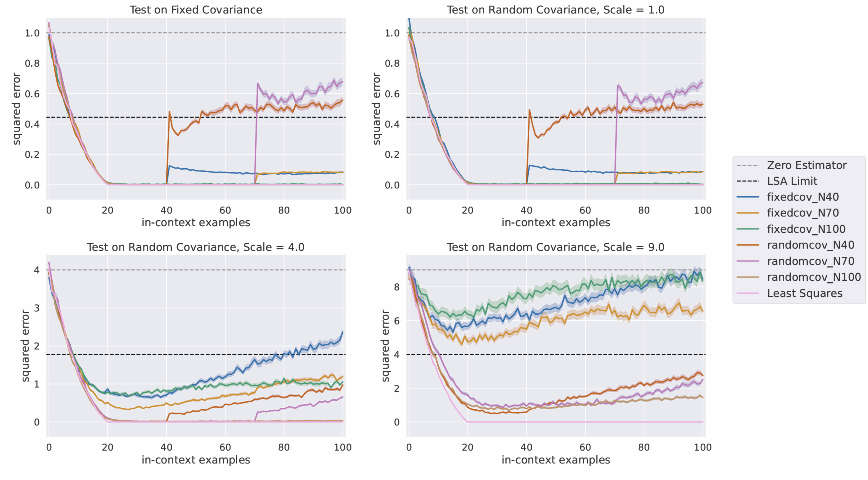
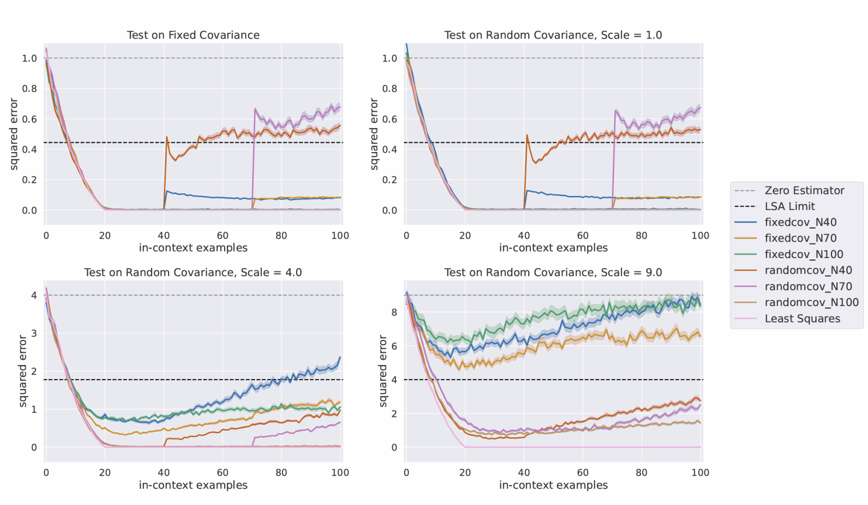


Figure: take N=40,70,100 when train and test six of them(fixed and random matrices case $2 \times 3 = 6$) for each small figure corresponding to four test include fixed matrices test prompts and random matrices with scaling factors c=1,4,9



The black dash line is LSA limit.
It is noteworthy that train and test c=1 on random matrices, GPT-2 performs well while we analyze failure in LSA model (linear architecture).

When the test prompt length M exceeds the training prompt length N: there is an evident spike in prediction error, regardless of fixed or random covariance case, and the spike appears to decrease when evaluated on prompts with higher variance.



Explanation: The positional encodings are randomly initialized and are learnable parameters but the encoding for position i is only updated if the transformer encounters a prompt which has a context of length i. Thus, when evaluating on prompts of length $M > N$, the model is relying upon random positional encodings for $M - N$ samples.

A concurrent work found that removing positional encoders improves performance when evaluating on larger contexts [APG23].

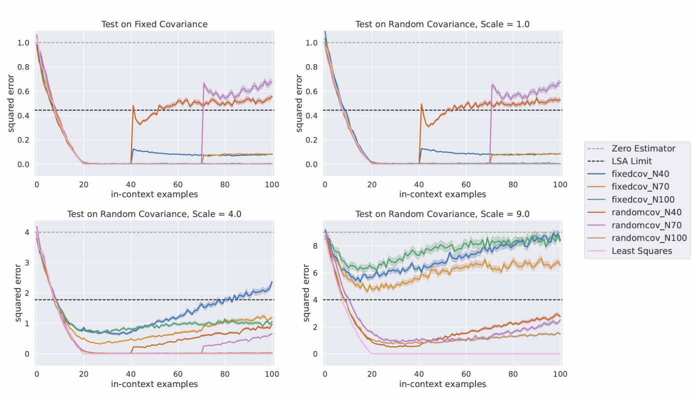


Table of Contents

1. recognize that the prediction $\widehat{y}_{\text{query}}(E_\tau;\theta)$ can be written as the output of a quadratic function $u^\top H_\tau u$ for a matrix H_τ depending on the token embedding matrix E_τ and for the vector u depending on $\theta = (W^{KQ}, W^{PV})$.
2. We then see that the dynamics are governed by a complex system of d^2+1 coupled differential equations.
3. the set of global minima for the d^2+1 coupled differential equations satisfies the condition $u^{-1}U_{11} = \Gamma^{-1}$. And get Minimum of Loss Function:

$$\tilde{\ell}(U_{11}, u_{-1}) - \min_{U_{11}\in\mathbb{R}^{d\times d}, u_{-1}\in\mathbb{R}} \tilde{\ell}(U_{11}, u_{-1}) = \frac{1}{2}\left\| \Gamma^{\frac{1}{2}} \left(u_{-1}\Lambda^{\frac{1}{2}} U_{11} \Lambda^{\frac{1}{2}} - \Lambda\Gamma^{-1}\right.\right.$$

4. Finally, we show that although the optimization problem is non-convex, a Polyak-Łojasiewicz (PL) inequality holds, which implies that gradient flow converges to a global minimum.

Preparation

By simple calculation, actually only part of W^{PV} and W^{KQ} affect the prediction $\widehat{y}$:
denote $W^{PV} \in \mathbb{R}^{(d+1)\times(d+1)}$ and $W^{KQ} \in \mathbb{R}^{(d+1)\times(d+1)}$

$$W^{PV} = \begin{pmatrix} W_{11}^{PV} & w_{12}^{PV} \\ (w_{21}^{PV})^\top & w_{22}^{PV} \end{pmatrix}, \quad W^{KQ} = \begin{pmatrix} W_{11}^{KQ} & w_{12}^{KQ} \\ (w_{21}^{KQ})^\top & w_{22}^{KQ} \end{pmatrix}, \tag{3.5}$$

where $W_{11}^{PV} \in \mathbb{R}^{d\times d}$; $w_{12}^{PV}, w_{21}^{PV} \in \mathbb{R}^d$; $w_{22}^{PV} \in \mathbb{R}$; and $W_{11}^{KQ} \in \mathbb{R}^{d\times d}$; $w_{12}^{KQ}, w_{21}^{KQ} \in \mathbb{R}^d$; $w_{22}^{KQ} \in \mathbb{R}$.
Then, the prediction $\widehat{y}_{\text{query}}$ is

$$\widehat{y}_{\text{query}} = \begin{pmatrix} (w_{21}^{PV})^\top & w_{22}^{PV} \end{pmatrix} \cdot \left(\frac{EE^\top}{N}\right) \begin{pmatrix} W_{11}^{KQ} \\ (w_{21}^{KQ})^\top \end{pmatrix} x_{\text{query}}, \tag{3.6}$$

we can set all other entries zero.

Step1: Lemma 5.1.

$$E_\tau := \begin{pmatrix} x_{\tau,1} & x_{\tau,2} & \cdots & x_{\tau,N} & x_{\tau,\text{query}} \\ \langle w_\tau, x_{\tau,1}\rangle & \langle w_\tau, x_{\tau,2}\rangle & \cdots & \langle w_\tau, x_{\tau,N}\rangle & 0 \end{pmatrix} \in \mathbb{R}^{(d+1)\times(N+1)}. \tag{9}$$

. Then the prediction $\widehat{y}_{\text{query}}(E_\tau;\theta)$ for the query covariate can be written as the output of a quadratic function,$\widehat{y}_{\text{query}}(E_\tau;\theta) = u^\top H_\tau u$, where the matrix H_τ is defined as,

$$H_\tau = \frac{1}{2} X_\tau \otimes \left(\frac{E_\tau E_\tau^\top}{N}\right) \in \mathbb{R}^{(d+1)^2\times(d+1)^2}, \quad X_\tau = \begin{pmatrix} 0_{d\times d} & x_{\tau,\text{query}} \\ (x_{\tau,\text{query}})^\top & 0 \end{pmatrix} \tag{5.1}$$

$$u = \text{Vec}(U) \in \mathbb{R}^{(d+1)^2}, \quad U = \begin{pmatrix} U_{11} & u_{12} \\ (u_{21})^\top & u_{-1} \end{pmatrix} \in \mathbb{R}^{(d+1)\times(d+1)},$$

where $U_{11} = W_{11}^{KQ} \in \mathbb{R}^{d\times d}$, $u_{12} = w_{21}^{PV} \in \mathbb{R}^{d\times 1}$, $u_{21} = w_{21}^{KQ} \in \mathbb{R}^{d\times 1}$, $u_{-1} = w_{22}^{PV} \in \mathbb{R}$ correspond to particular components of W^{PV} and W^{KQ} This implies that we can write the original loss function (3.7) as

$$\widehat{L} = \frac{1}{2B}\sum_{\tau=1}^{B}\left(u^\top H_\tau u - w_\tau^\top x_{\tau,\text{query}}\right)^2.$$

Basic knowledge used in proof

Lemma D.1 (Matrix Derivatives, Kronecker Product and Vectorization, [PP+08]). We denote A, B, X as matrices and $\mathbf{x}$ as vectors. Then, we have

- $\dfrac{\partial \mathbf{x}^\top B \mathbf{x}}{\partial \mathbf{x}} = (B + B^\top)\mathbf{x}$.
- $\mathrm{Vec}(AXB) = (B^\top \otimes A)\,\mathrm{Vec}(X)$.
- $\mathrm{tr}(A^\top B) = \mathrm{Vec}(A)^\top \mathrm{Vec}(B)$.
- $\dfrac{\partial}{\partial X}\mathrm{tr}(XBX^\top) = XB^\top + XB$.
- $\dfrac{\partial}{\partial X}\mathrm{tr}(AX^\top) = A$.
- $\dfrac{\partial}{\partial X}\mathrm{tr}(AXBX^\top C) = A^\top C^\top X B^\top + CAXB$.

Step1: Lemma 5.1.

$$E_\tau := \begin{pmatrix} x_{\tau,1} & x_{\tau,2} & \cdots & x_{\tau,N} & x_{\tau,\text{query}} \\ \langle w_\tau, x_{\tau,1}\rangle & \langle w_\tau, x_{\tau,2}\rangle & \cdots & \langle w_\tau, x_{\tau,N}\rangle & 0 \end{pmatrix} \in \mathbb{R}^{(d+1)\times(N+1)}. \tag{10}$$

. Then the prediction $\widehat{y}_{\text{query}}(E_\tau;\theta)$ for the query covariate can be written as the output of a quadratic function,$\widehat{y}_{\text{query}}(E_\tau;\theta) = u^\top H_\tau u$, where the matrix H_τ is defined as,

$$H_\tau = \frac{1}{2} X_\tau \otimes \left(\frac{E_\tau E_\tau^\top}{N}\right) \in \mathbb{R}^{(d+1)^2\times(d+1)^2}, \quad X_\tau = \begin{pmatrix} 0_{d\times d} & x_{\tau,\text{query}} \\ (x_{\tau,\text{query}})^\top & 0 \end{pmatrix} \tag{5.1}$$

$$u = \text{Vec}(U) \in \mathbb{R}^{(d+1)^2}, \quad U = \begin{pmatrix} U_{11} & u_{12} \\ (u_{21})^\top & u_{-1} \end{pmatrix} \in \mathbb{R}^{(d+1)\times(d+1)},$$

where $U_{11} = W_{11}^{KQ} \in \mathbb{R}^{d\times d}$, $u_{12} = w_{21}^{PV} \in \mathbb{R}^{d\times 1}$, $u_{21} = w_{21}^{KQ} \in \mathbb{R}^{d\times 1}$, $u_{-1} = w_{22}^{PV} \in \mathbb{R}$ correspond to particular components of W^{PV} and W^{KQ}

quadratic function is non-convex

Remark

Prove the matrix

$$H_\tau = \frac{1}{2} X_\tau \otimes \left(\frac{E_\tau E_\tau^\top}{N} \right)$$

has at least $d+1$ negative eigenvalues

Step 2: Lemma 5.2. Let $u = \text{Vec}(U) := \text{Vec}\left(\begin{pmatrix} U_{11} & u_{12} \\ (u_{21})^\top & u_{-1} \end{pmatrix}\right)$ as in Lemma 5.1. Consider gradient flow over $L := \frac{1}{2}\mathbb{E}\left(u^\top H_\tau u - w_\tau^\top x_{\tau,\text{query}}\right)^2$ the expectation is taken over $x_{\tau,i}, x_{\text{query}} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,\Lambda)$ and $w_\tau \sim \mathcal{N}(0, I_d)$. with respect to u starting from an initial value satisfying Assumption. Then the dynamics of U follows

$$\begin{aligned} \frac{d}{dt}U_{11}(t) &= -u_{-1}^2\Gamma\Lambda U_{11}\Lambda + u_{-1}\Lambda^2 \\ \frac{d}{dt}u_{-1}(t) &= -\operatorname{tr}\left[u_{-1}\Gamma\Lambda U_{11}\Lambda(U_{11})^\top - \Lambda^2(U_{11})^\top\right], \end{aligned} \tag{5.4}$$

and $u_{12}(t) = 0_d$, $u_{21}(t) = 0_d$ for all $t \geq 0$, where $\Gamma = \left(1 + \frac{1}{N}\right)\Lambda + \frac{1}{N}\operatorname{tr}(\Lambda) I_d \in \mathbb{R}^{d\times d}$.

So the dynamics are governed by a complex system of d^2+1 coupled differential equations. We can shows that these dynamics are the same as those of gradient flow on the following objective function:

$$\tilde{\ell} : \mathbb{R}^{d\times d} \times \mathbb{R} \to \mathbb{R}, \quad \tilde{\ell}(U_{11}, u_{-1}) = \operatorname{tr}\left[\frac{1}{2}u_{-1}^2\Gamma\Lambda U_{11}\Lambda(U_{11})^\top - u_{-1}\Lambda^2(U_{11})^\top\right]$$

We will use the following lemma in proof **Lemma D.2. (Isserlis' Theorem)** If X is Gaussian random vector of d dimension, mean zero and covariance matrix Λ, and $A \in \mathbb{R}^{d\times d}$ is a fixed matrix. Then

$$\mathbb{E}\left[XX^\top AXX^\top\right] = \Lambda\left(A + A^\top\right)\Lambda + \mathrm{tr}(A\Lambda)\Lambda. \tag{11}$$

1. Calculate the Second Term
2. Calculate the First Term
3. u_{12} and u_{21} Vanish
4. Dynamics of U_{11}
5. Dynamics of u_{-1}

Corollary A.2 (Minimum of Loss Function). The loss function $\tilde{\ell}$ in Lemma A.1 satisfies

$$\min_{U_{11}\in\mathbb{R}^{d\times d}, u_{-1}\in\mathbb{R}} \tilde{\ell}(U_{11}, u_{-1}) = -\frac{1}{2}\operatorname{tr}\left[\Lambda^2\Gamma^{-1}\right] \tag{12}$$

and

$$\tilde{\ell}(U_{11}, u_{-1}) - \min_{U_{11}\in\mathbb{R}^{d\times d}, u_{-1}\in\mathbb{R}} \tilde{\ell}(U_{11}, u_{-1}) = \frac{1}{2}\left\|\Gamma^{\frac{1}{2}}\left(u_{-1}\Lambda^{\frac{1}{2}}U_{11}\Lambda^{\frac{1}{2}} - \Lambda\Gamma^{-1}\right)\right\|_F^2. \tag{13}$$

Equality holds when

$$U_{11} = c\Gamma^{-1}, \quad u_{-1} = c^{-1}$$

Lemma D.4 ([MR99]). For any two positive semi-definite matrices $A, B \in \mathbb{R}^{d\times d}$, we have

- $\mathrm{tr}[AB] \geq 0$.
- $AB \succeq 0$ if and only if A and B commute.

We now show that PL inequality holds, which implies that gradient flow converges to a global minimum:

Lemma 5.4. Suppose the initialization of gradient flow satisfies Assumption with initialization scale satisfying $\sigma^2 < \frac{2}{\sqrt{d}\|\Gamma\|_{op}}$, define:

$$\mu := \frac{\sigma^2}{\sqrt{d}\|\Lambda\|_{op}^2 \operatorname{tr}(\Gamma^{-1}\Lambda^{-1}) \operatorname{tr}(\Lambda^{-1})} \|\Lambda\Theta\|_F^2 \left[2 - \sqrt{d}\sigma^2 \|\Gamma\|_{op}\right] > 0, \quad (5.7)$$

gradient flow on $\tilde{\ell}$ with respect to U_{11} and u_{-1} satisfies, for any $t \geq 0$,

$$\begin{aligned}\left\|\nabla \tilde{\ell}(U_{11}(t), u_{-1}(t))\right\|_2^2 &:= \left\|\frac{\partial \tilde{\ell}}{\partial U_{11}}\right\|_F^2 + \left|\frac{\partial \tilde{\ell}}{\partial u_{-1}}\right|^2 \\ &\geq \mu \left(\tilde{\ell}(U_{11}(t), u_{-1}(t)) - \min_{U_{11}\in\mathbb{R}^{d\times d}, u_{-1}\in\mathbb{R}} \tilde{\ell}(U_{11}, u_{-1})\right)\end{aligned} \quad (5.8)$$

Moreover, gradient flow converges to the global minimum of $\tilde{\ell}$, and we find U_{11} and u_{-1} exactly converge to the following,

$\lim_{t\to\infty} u_{-1}(t) = \|\Gamma^{-1}\|_F^{\frac{1}{2}}$ and $\lim_{t\to\infty} U_{11}(t) = \|\Gamma^{-1}\|_F^{-\frac{1}{2}} \Gamma^{-1}$.

We will use the following lemma in proof:

Lemma D.3 (Von-Neumann's Trace Inequality). Let $U, V \in \mathbb{R}^{d \times n}$ with $d \leq n$. We have

$$\operatorname{tr}\left(U^\top V\right) \leq \sum_{i=1}^{d} \sigma_i(U)\sigma_i(V) \leq \|U\|_{op} \times \sum_{i=1}^{d} \sigma_i(V) \leq \sqrt{d} \cdot \|U\|_{op}\|V\|_F, \tag{14}$$

where $\sigma_1(X) \geq \sigma_2(X) \geq \cdots \geq \sigma_d(X)$ are the ordered singular values of $X \in \mathbb{R}^{d \times n}$.

lemma A.3 says the parameters in the LSA model will keep 'balanced' in the whole trajectory. From the proof of this lemma, we can understand why we assume a balanced parameter Assumption at the initial time.

Lemma A.3 (Balanced Parameters). Consider gradient flow over $L(= \tilde{\ell} + C)$ in with respect to u starting from an initial value satisfying Assumption . For any $t \geq 0$, it holds that

$$u_{-1}^2 = \mathrm{tr}\left[U_{11}(U_{11})^\top\right]. \tag{A.12}$$

We prove A.4 for the following Lemma A.5

Lemma A.4. Consider gradient flow over $L(= \tilde{\ell} + C)$ with respect to u starting from an initial value satisfying Assumption. If the initial scale satisfies

$$0 < \sigma < \sqrt{\frac{2}{\sqrt{d}\|\Gamma\|_{op}}}, \tag{A.13}$$

then, for any $t \geq 0$, it holds that

$$u_{-1} > 0. \tag{15}$$

Lemma A.5. Consider gradient flow over L in with respect to u starting from an initial value satisfying Assumption with initial scale $0 < \sigma < \sqrt{\frac{2}{\sqrt{d}\|\Gamma\|_{op}}}$. For any $t \geq 0$, it holds that

$$u_{-1} \geq \sqrt{\frac{\sigma^2}{2\sqrt{d}\|\Lambda\|_{op}^2}\|\Lambda\Theta\|_F^2\left[2 - \sqrt{d}\sigma^2\|\Gamma\|_{op}\right]} > 0. \qquad \text{(A.14)}$$

Finally, let's prove the PL inequality and further, the global convergence of gradent flow on the loss function $\tilde{\ell}$

Theorem 4.2

Theorem 4.2. Let $\mathcal{D}$ be a distribution over $(x, y) \in \mathbb{R}^d \times \mathbb{R}$, whose marginal distribution on x is $\mathcal{D}_x = \mathcal{N}(0, \Lambda)$. Assume $\mathbb{E}_{\mathcal{D}}[y]$, $\mathbb{E}_{\mathcal{D}}[xy]$, $\mathbb{E}_{\mathcal{D}}[y^2 xx^\top]$ exist and are finite. If we define
$a := \Lambda^{-1}\mathbb{E}_{(x,y)\sim\mathcal{D}}[xy], \quad \Gamma := \Lambda + \frac{1}{N}\Lambda + \frac{1}{N}\operatorname{tr}(\Lambda) I_d$, and
$\Sigma := \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[(xy - \mathbb{E}(xy))(xy - \mathbb{E}(xy))^\top\right]$.
f^*_{LSA} be the LSA model in above theorem. Assume the test prompt is of the form $P = (x_1, y_1, \ldots, x_M, y_M, x_{\mathrm{query}})$, where $(x_i, y_i), (x_{\mathrm{query}}, y_{\mathrm{query}}) \overset{\text{i.i.d.}}{\sim} \mathcal{D}$. and $\widehat{y}_{\mathrm{query}} = [f^*_{LSA}(E_P; (W^{PV}_*, W^{KQ}_*))]_{(d+1),(M+1)}$ is the trained LSA model prediction for x_{query} given the prompt. we have:

$$\mathbb{E}\left(\widehat{y}_{\mathrm{query}} - y_{\mathrm{query}}\right)^2 = \underbrace{\min_{w \in \mathbb{R}^d} \mathbb{E}\left(\langle w, x_{\mathrm{query}}\rangle - y_{\mathrm{query}}\right)^2}_{\text{Error of best linear predictor}}$$

$$+ \operatorname{tr}\left[\Sigma\Gamma^{-2}\Lambda\right] + \frac{1}{N^2}\left[\|a\|^2_{\Gamma^{-2}\Lambda^3} + 2\operatorname{tr}(\Lambda)\|a\|^2_{\Gamma^{-2}\Lambda^2} + \operatorname{tr}(\Lambda)^2\|a\|^2_{\Gamma^{-2}\Lambda}\right],$$

where the expectation is over $(x_i, y_i), (x_{\mathrm{query}}, y_{\mathrm{query}}) \overset{\text{i.i.d.}}{\sim} \mathcal{D}$.

proof of Theorem 4.2.

Theorem 4.2. Let $\mathcal{D}$ be a distribution over $(x, y) \in \mathbb{R}^d \times \mathbb{R}$, whose marginal distribution on x is $\mathcal{D}_x = \mathcal{N}(0, \Lambda)$. Assume $\mathbb{E}_{\mathcal{D}}[y]$, $\mathbb{E}_{\mathcal{D}}[xy]$, $\mathbb{E}_{\mathcal{D}}[y^2 x x^\top]$ exist and are finite. If we define
$a := \Lambda^{-1}\mathbb{E}_{(x,y)\sim\mathcal{D}}[xy], \quad \Gamma := \Lambda + \frac{1}{N}\Lambda + \frac{1}{N}\operatorname{tr}(\Lambda) I_d$, and
$\Sigma := \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[(xy - \mathbb{E}(xy))(xy - \mathbb{E}(xy))^\top\right].$
f^*_{LSA} be the LSA model in above theorem. Assume the test prompt is of the form $P = (x_1, y_1, \ldots, x_M, y_M, x_{\mathrm{query}})$, where $(x_i, y_i), (x_{\mathrm{query}}, y_{\mathrm{query}}) \overset{\text{i.i.d.}}{\sim} \mathcal{D}$. and $\widehat{y}_{\mathrm{query}} = [f^*_{LSA}(E_P; (W^{PV}_*, W^{KQ}_*))]_{(d+1),(M+1)}$ is the trained LSA model prediction for x_{query} given the prompt. we have:

$$\mathbb{E}\left(\widehat{y}_{\mathrm{query}} - y_{\mathrm{query}}\right)^2 = \underbrace{\min_{w\in\mathbb{R}^d} \mathbb{E}\left(\langle w, x_{\mathrm{query}}\rangle - y_{\mathrm{query}}\right)^2}_{\text{Error of best linear predictor}}$$

$$+ \operatorname{tr}\left[\Sigma\Gamma^{-2}\Lambda\right] + \frac{1}{N^2}\left[\|a\|^2_{\Gamma^{-2}\Lambda^3} + 2\operatorname{tr}(\Lambda)\|a\|^2_{\Gamma^{-2}\Lambda^2} + \operatorname{tr}(\Lambda)^2\|a\|^2_{\Gamma^{-2}\Lambda}\right],$$

where the expectation is over $(x_i, y_i), (x_{\mathrm{query}}, y_{\mathrm{query}}) \overset{\text{i.i.d.}}{\sim} \mathcal{D}$.

Table of Contents

Summary

In this work, we investigated the dynamics of in-context learning of transformers with a single linear self attention layer under gradient flow on the population loss.

Summary

There are a number of natural directions for future research.

1. similar results would hold for stochastic gradient descent with finite step sizes?
2. similar results would hold for more general initializations.
3. understanding the dynamics of in-context learning in nonlinear and deep transformers.[1]
4. **covariate shifts** the framework restricted to the fixed marginal distribution over the covariates ($\mathcal{D}_x$) but other learning algorithms (such as ordinary least squares) are able to achieve small prediction error for prompts for very general classes of distributions[2]
5. removing positional encoders in GPT-2 improves performance

1.we refer to Huang et al. [2023](In-context convergence of transformers.), Chen et al. [2024](Training dynamics of multi-head softmax attention...) for linear regression prediction.

Reference

2.we refer to Li et al. [2024](One-Layer Transformer Provably Learns One-Nearest Neighbor In Context)

Other reference mentioned above:

1. Garg et al. [Gar+22](What Can Transformers Learn In-Context? A Case Study of Simple Function Classes)
2. .[PP+08] Kaare Brandt Petersen, Michael Syskind Pedersen, et al. "The matrix cookbook". In: Technical University of Denmark 7.15 (2008), p. 510
3. .[MR99] AR Meenakshi and C Rajian. "On a product of positive semidefinite matrices". In: Linear algebra and its applications 295.1-3 (1999), pp. 3–6
4. .[APG23] Kabir Ahuja, Madhur Panwar, and Navin Goyal. "In-Context Learning through the Bayesian Prism". In: Preprint, arXiv:2306.04891 (2023)