Learning two-layer networks, one (giant) step at a time

Ludovic Stephan

École Polytechnique Fédérale de Lausanne

Yatin Dandi

Florent Krzakala

Bruno Loureiro

Luca Pesce

Introduction: approximating real-world networks

Modern AI models...

Modern AI models...

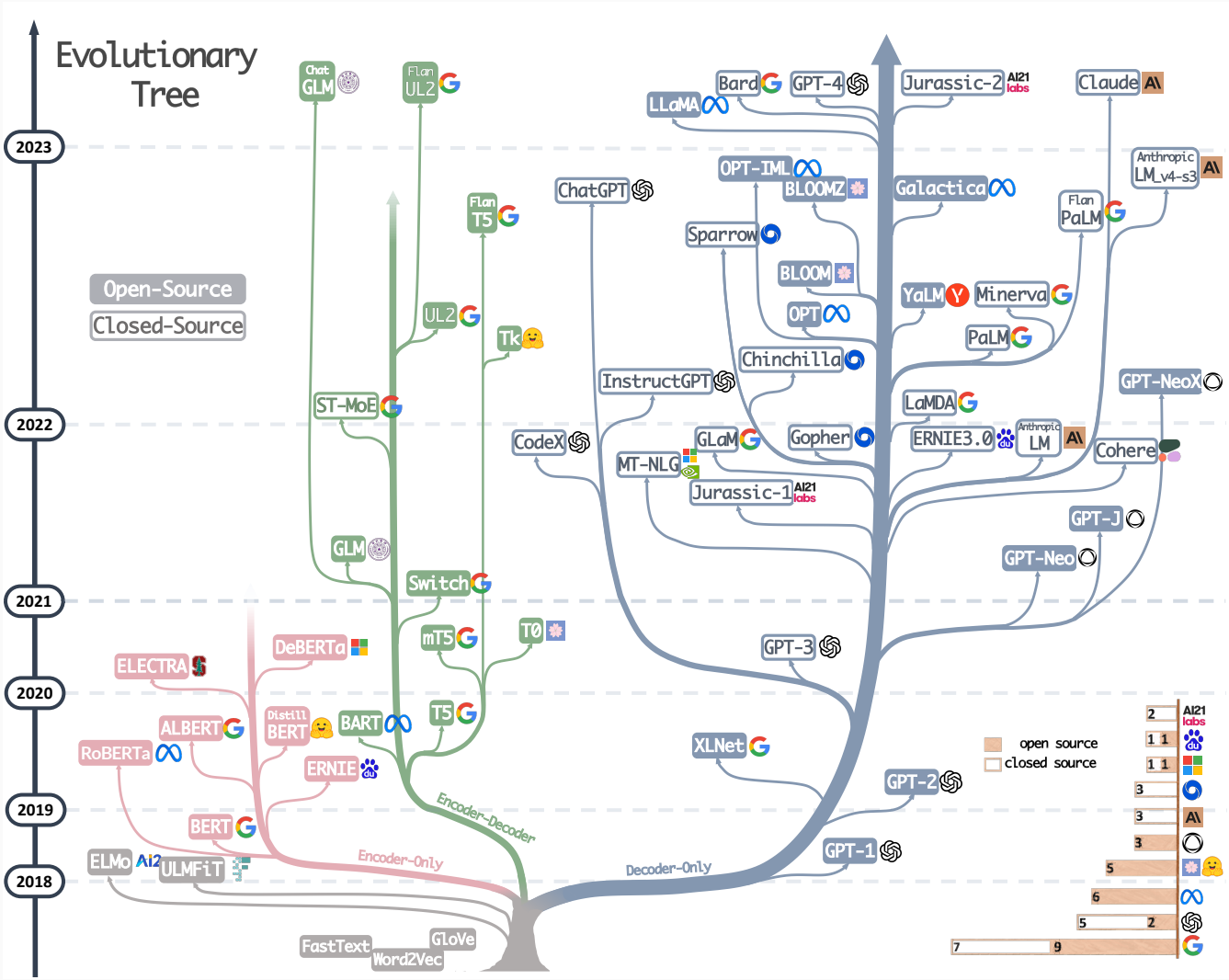


figure from Yang et al '23

… and their understanding

… and their understanding

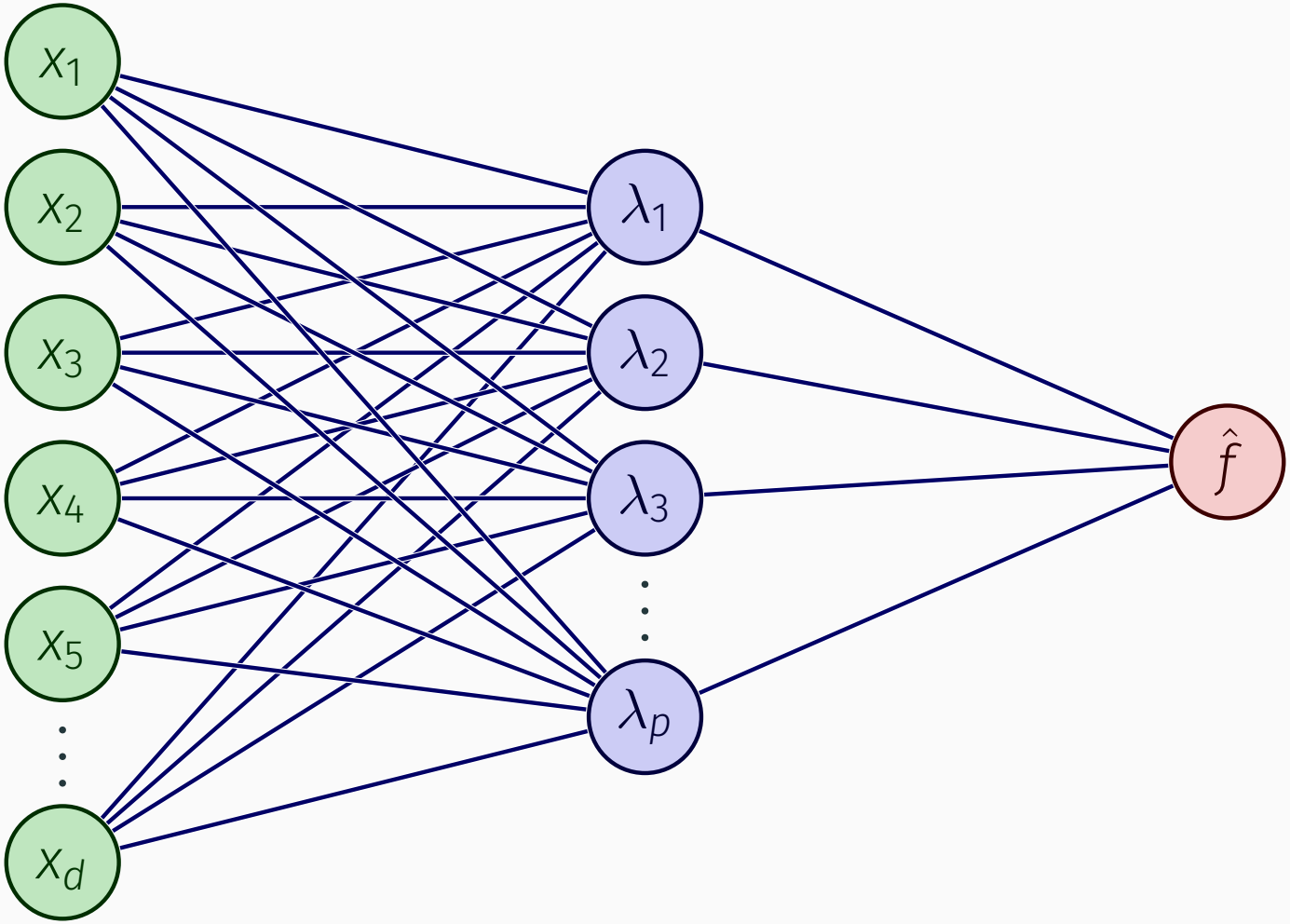
[Slide intentionally left blank]

Why ?

Main challenges:

- network architectures are getting more and more complex
- real-world data distribution is hard to approximate
- the labelling function can be arbitrarily complicated
- the choice of training algorithm matters

Our architecture: two-layer neural networks



Our architecture: two-layer neural networks

Estimator of the form

$$\hat{f}(x; W, a) = \frac{1}{p} \sum_{i=1}^{n} a_i \sigma(\langle w_i, x \rangle)$$

Parameters:

- $W \in \mathbb{R}^{p \times d}$: *first layer* weights
- $a \in \mathbb{R}^{p}$: *second layer* weights
- $\sigma : \mathbb{R} \to \mathbb{R}$: activation function

Why 2LNNs ?

Simple enough to be tractable, but...

- non-linear function $\rightarrow$ non-convex problem
- rich optimization landscape
- algorithm-dependent weight choice

Our data model: Gaussians

Why Gaussians

Two main reasons:

- simplest distribution possible

figure from Loureiro et al ’21

Why Gaussians

Two main reasons:

- simplest distribution possible
- Gaussian Universality

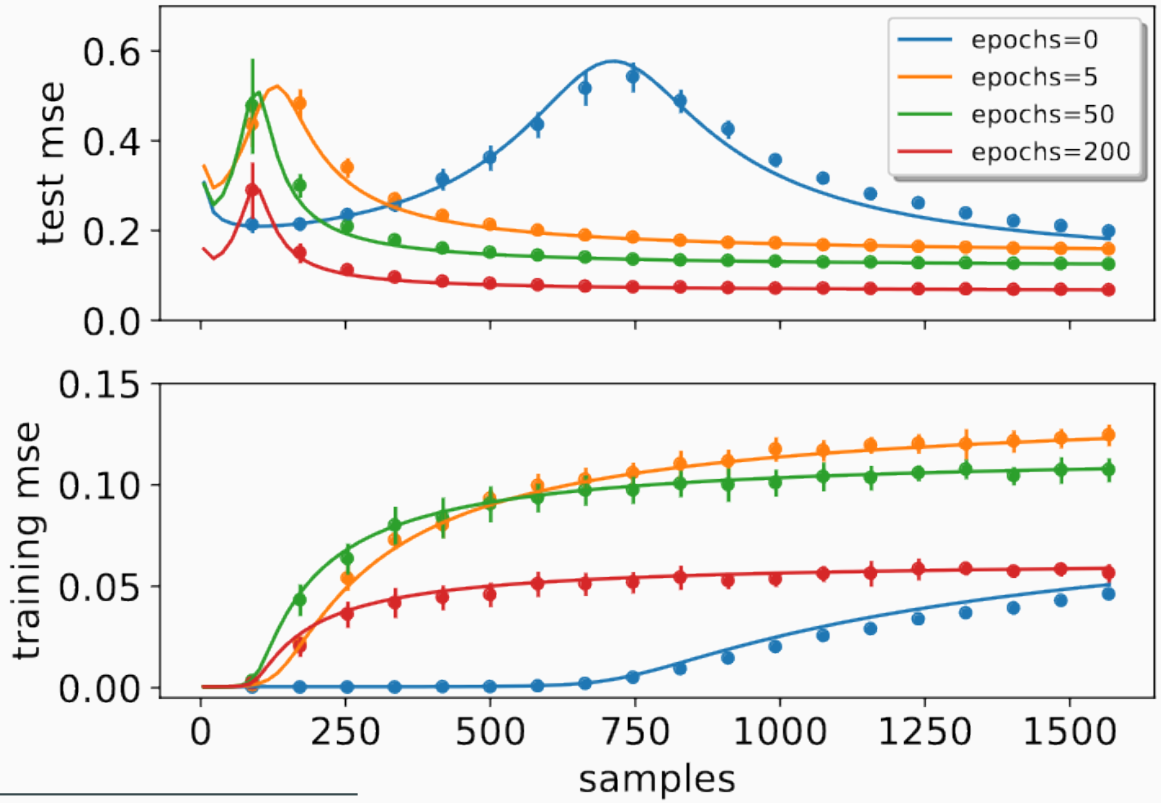


figure from Loureiro et al '21

Labeling function

Common assumption: y depends only on a few privileged directions

$$y = f^\star(\langle w_1^\star, x\rangle, \dots, \langle w_r^\star, x\rangle)$$

- proxy for more complicated distributions, e.g. power laws
- useful to study feature learning

Learning algorithm

Adam: A method for stochastic optimization

Authors	Diederik Kingma, Jimmy Ba
Publication date	2015
Conference	International Conference on Learning Representations
Description	We introduce Adam, an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments. The method is straightforward to implement, is computationally efficient, has little memory requirements, is invariant to diagonal rescaling of the gradients, and is well suited for problems that are large in terms of data and/or parameters. The method is also appropriate for non-stationary objectives and problems with very noisy and/or sparse gradients. The hyper-parameters have intuitive interpretations and typically require little tuning. Some connections to related algorithms, on which Adam was inspired, are discussed. We also analyze the theoretical convergence properties of the algorithm and provide a regret bound on the convergence rate that is comparable to the best known results under the online convex optimization framework. Empirical results demonstrate that Adam works well in practice and compares favorably to other stochastic optimization methods. Finally, we discuss AdaMax, a variant of Adam based on the infinity norm.
Total citations	Cited by 153449

Apologies to DORA

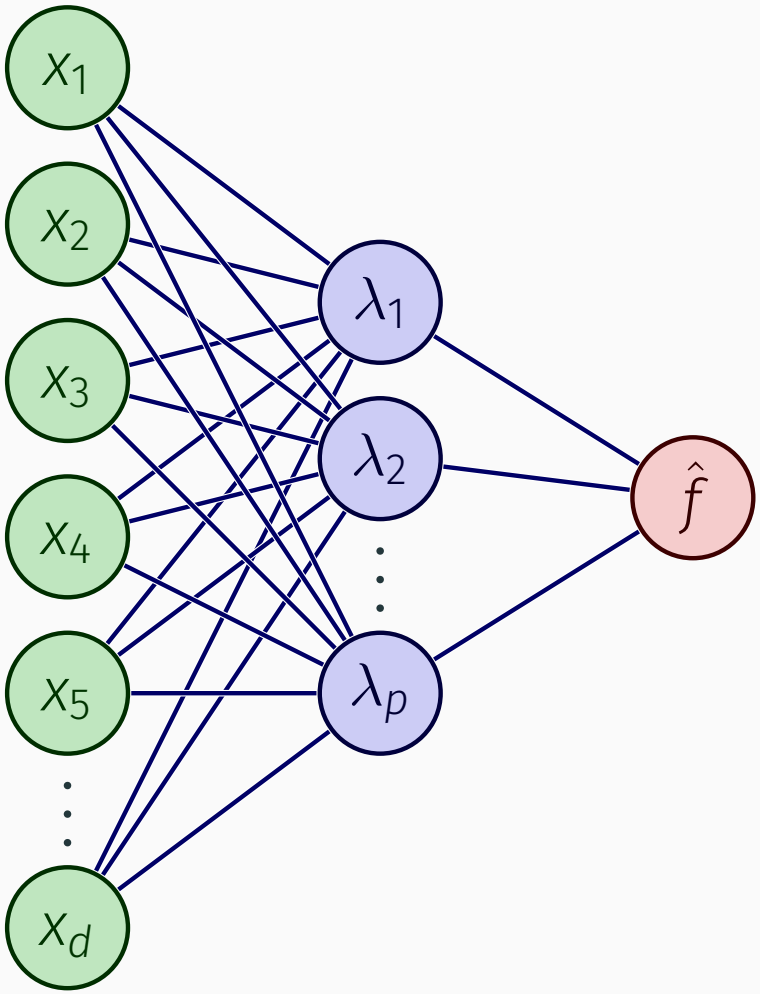
Learning algorithm

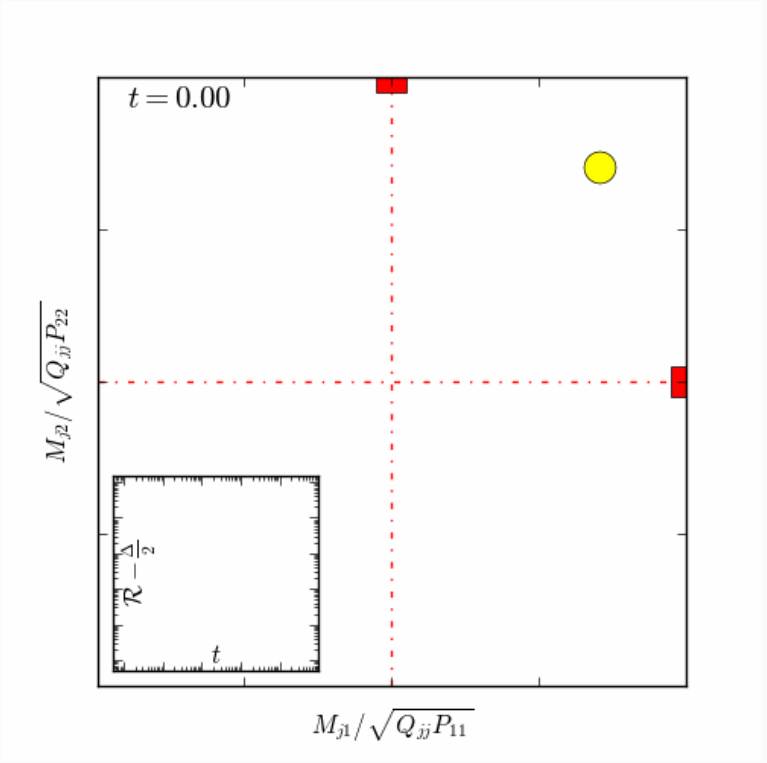
In most research works: one-pass SGD

$$w_i^{t+1} = w_i^t - \eta \nabla \ell(y_t, \hat{f}(x_t)), \quad (x_t, y_t) \text{ i.i.d}$$

- fairly fast
- good convergence guarantees

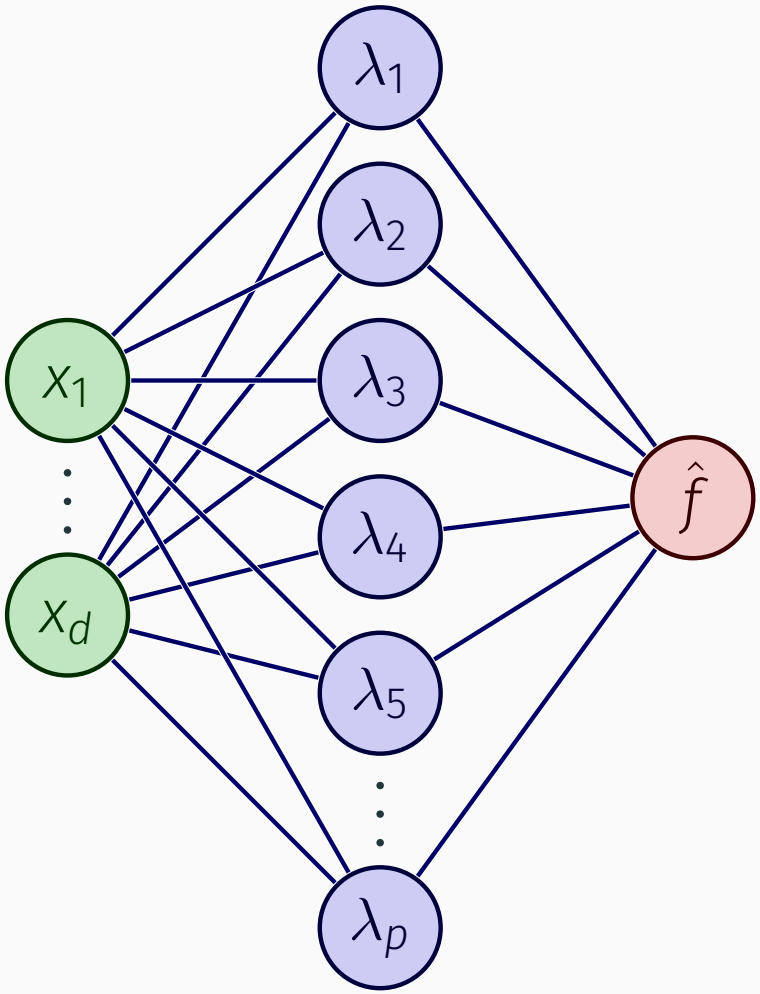
Study of SGD



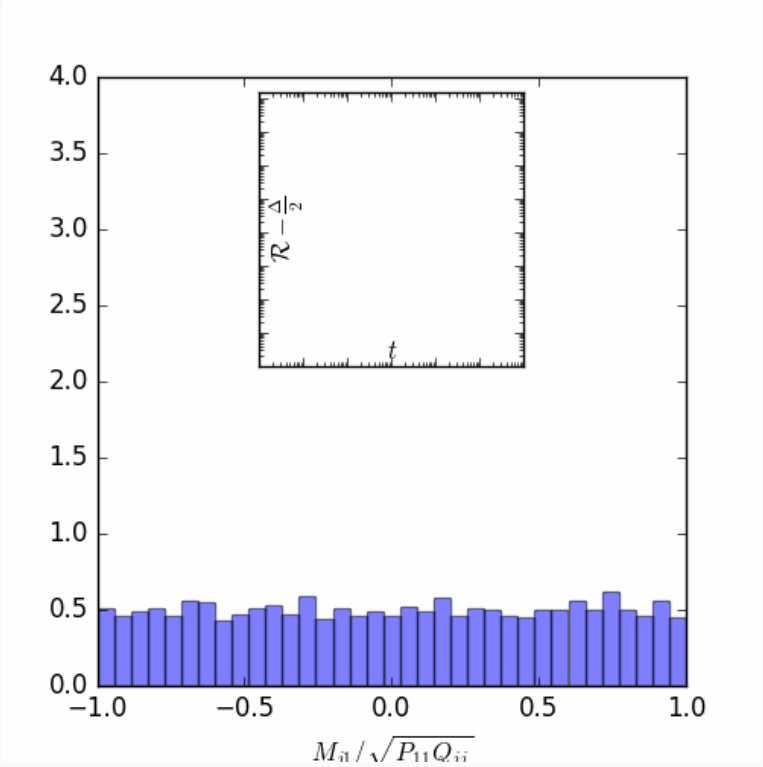


Saad & Solla '95; animation by Luca A.

Study of SGD

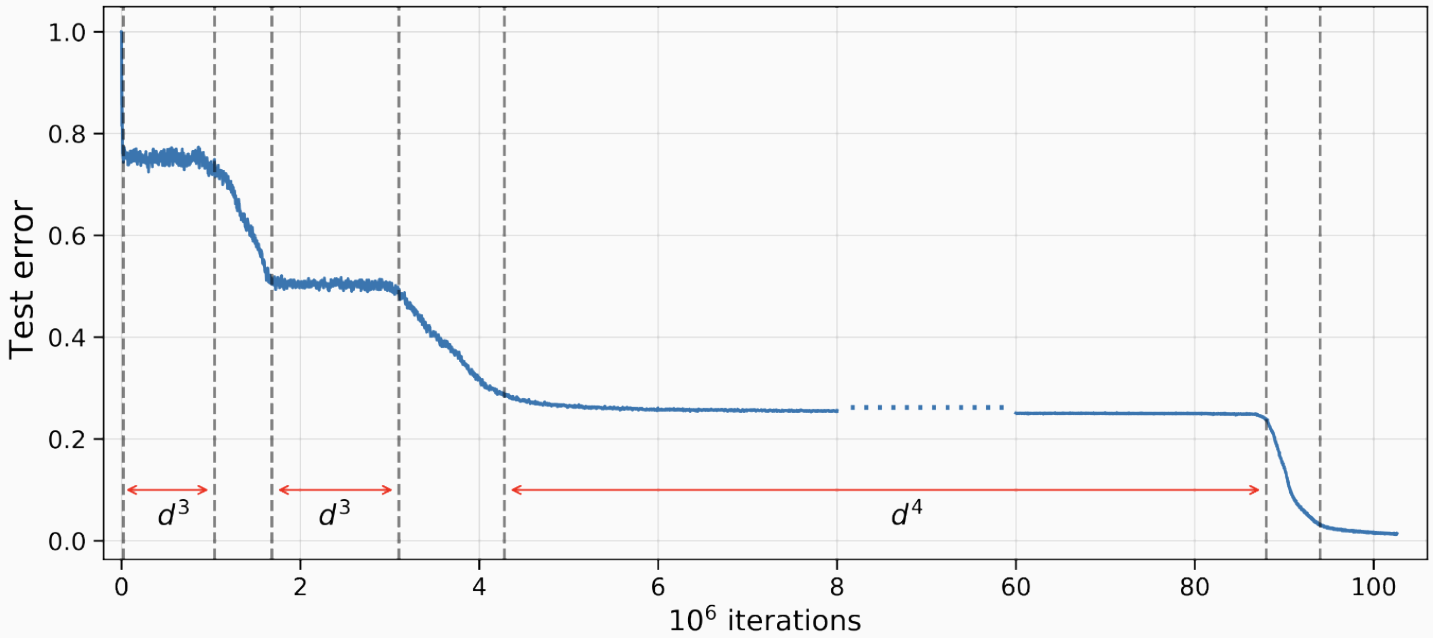


$\rightarrow$



Rotskoff & Vanden-Eijnden, Mei et al. '18; animation by Luca A.

Study of SGD



Abbe et al. ’22

Our work: large gradient steps

More precisely, large-batch gradient steps:

$$w_i^{t+1} = w_i^t - \frac{\eta}{n} \sum_{\mu=1}^{n} \nabla \ell(y^\mu, \hat{f}(x^\mu)), \quad (x^\mu, y^\mu) \text{ i.i.d}$$

Large-batch $\quad \Leftrightarrow \quad n \to \infty,\ n = \Theta(d^\kappa)$

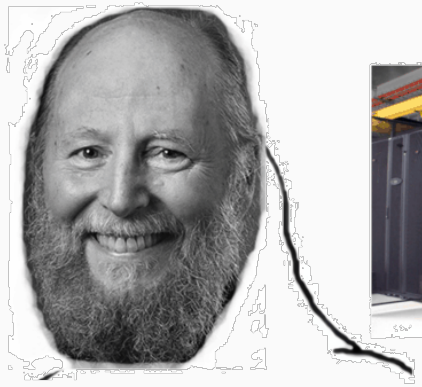
Why big gradient steps ?

Works well with:

Why big gradient steps ?

Works well with: parallel programming,

nooooo you can't just scale up pure connectionist models on Internet data without inductive biases and modularization and expect them to learn real-world knowledge and grammar from form, or arithmetic and logical reasoning and causal inference—that's just memorization and superficial pattern-matching like Eliza,you need grounding in real-world communication with intent and social dynamics and multimodal robotic embodiment which can foster disentangled learning from guided exploration and self-directed goals expressed in Bayesian programs and probabilistic graphical models which are interpretable and pin down a unique semantics which can be debiased and expressed with uncertainty, and learned efficiently on tiny academic budgets, the cost only shows how this is a dead-end, we need to stop chasing SOTAs and model the complexity of the brain and consider the social context to decolonize AI's structural biases for Third World researchers...



haha gpus go brrr

Why big gradient steps ?

Works well with: parallel programming, decentralized/distributed/federated learning

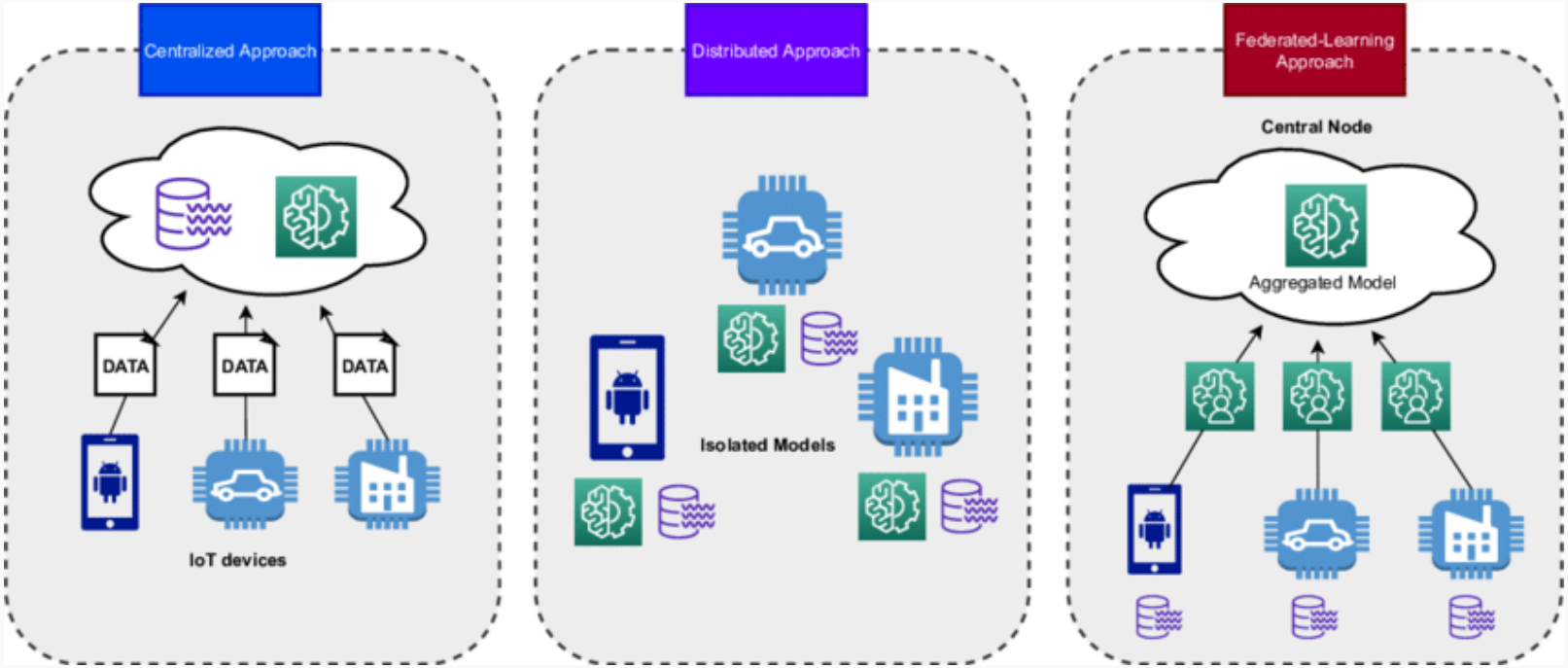
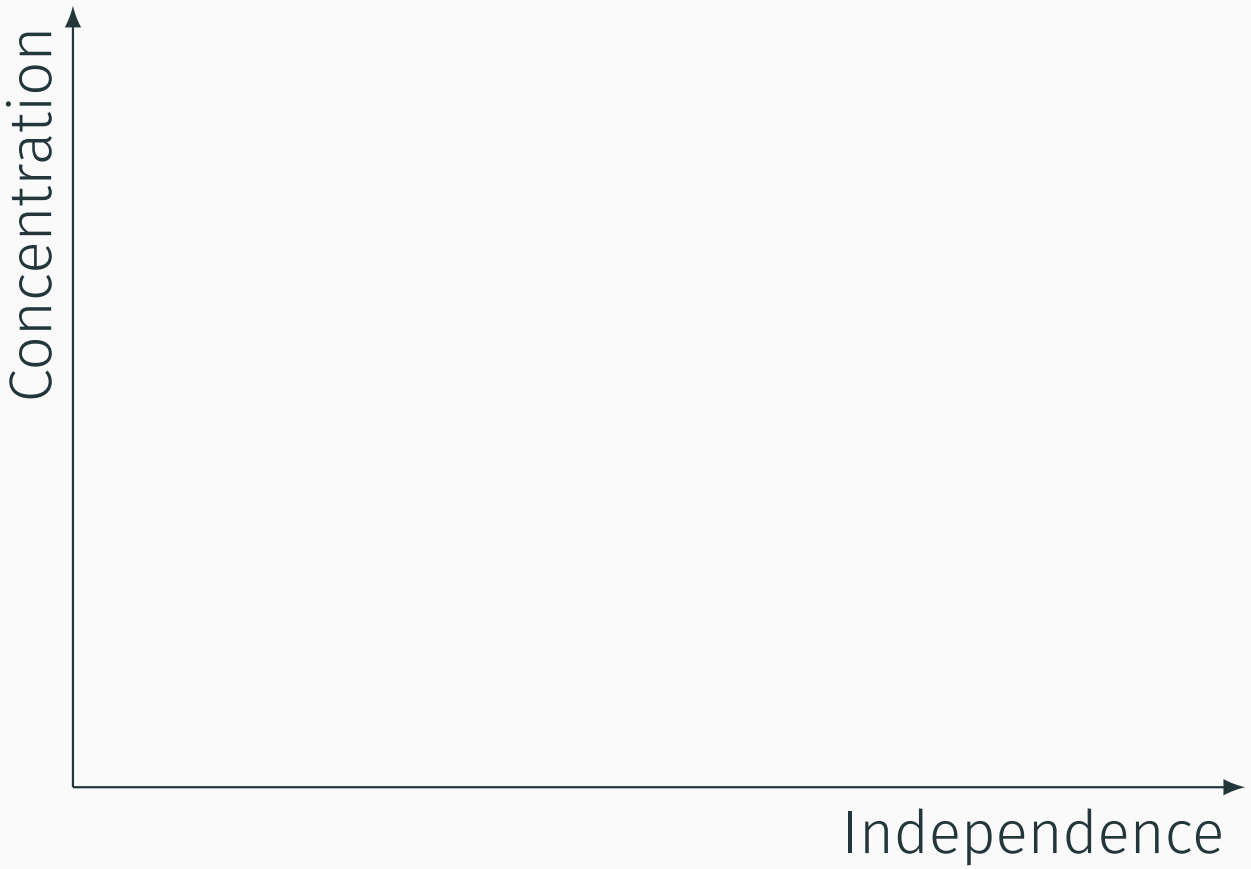
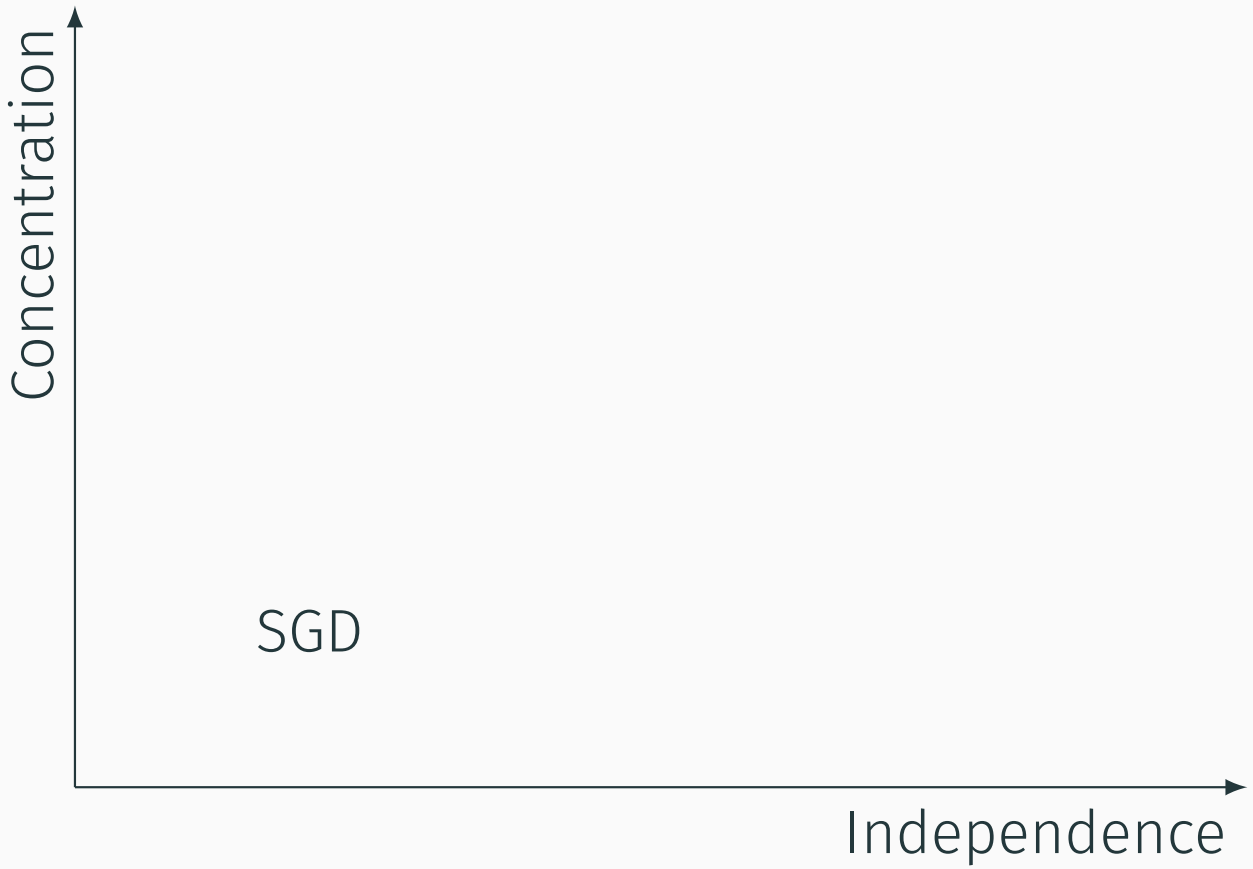


figure from Campos et al. '21

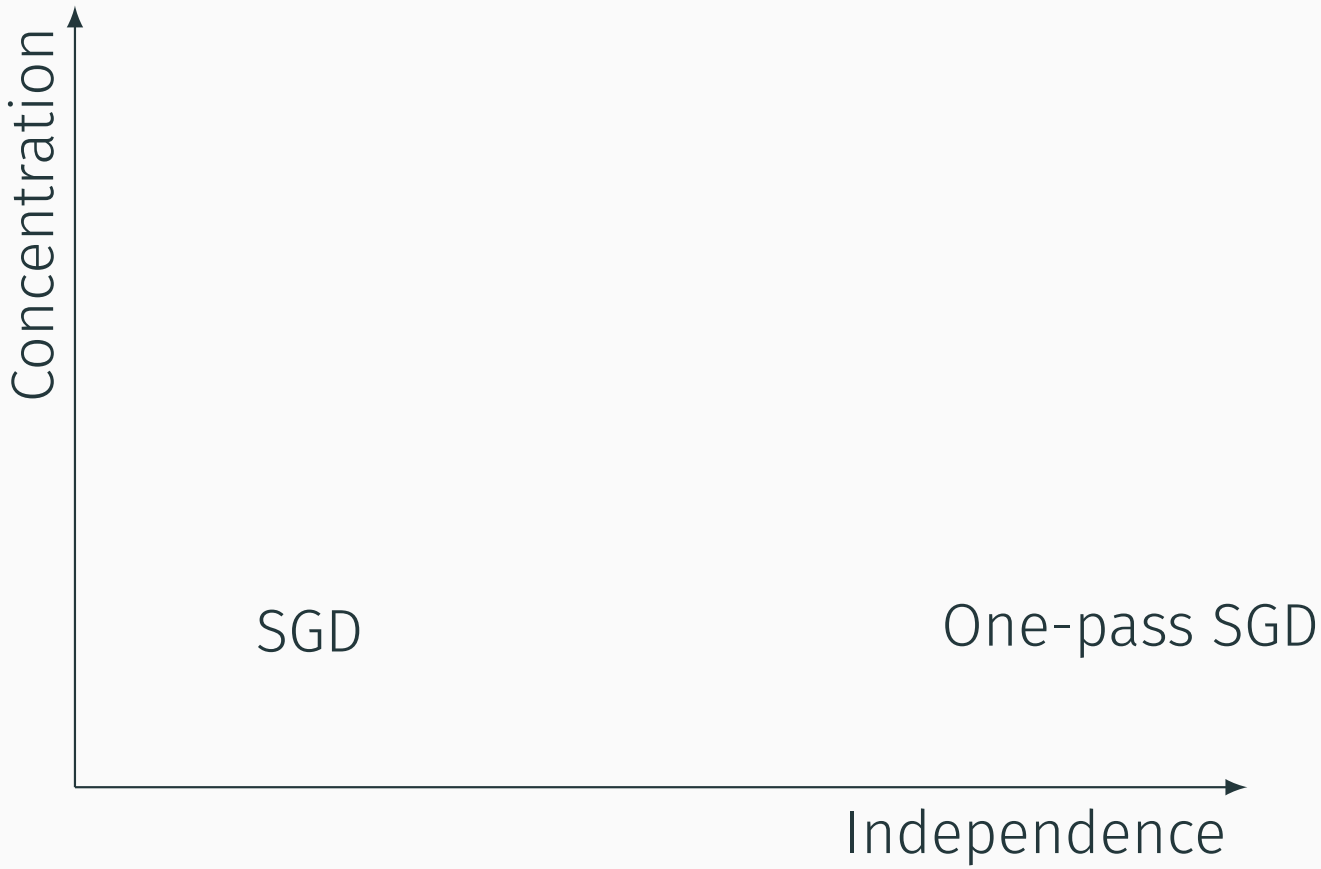
Why big gradient steps ?



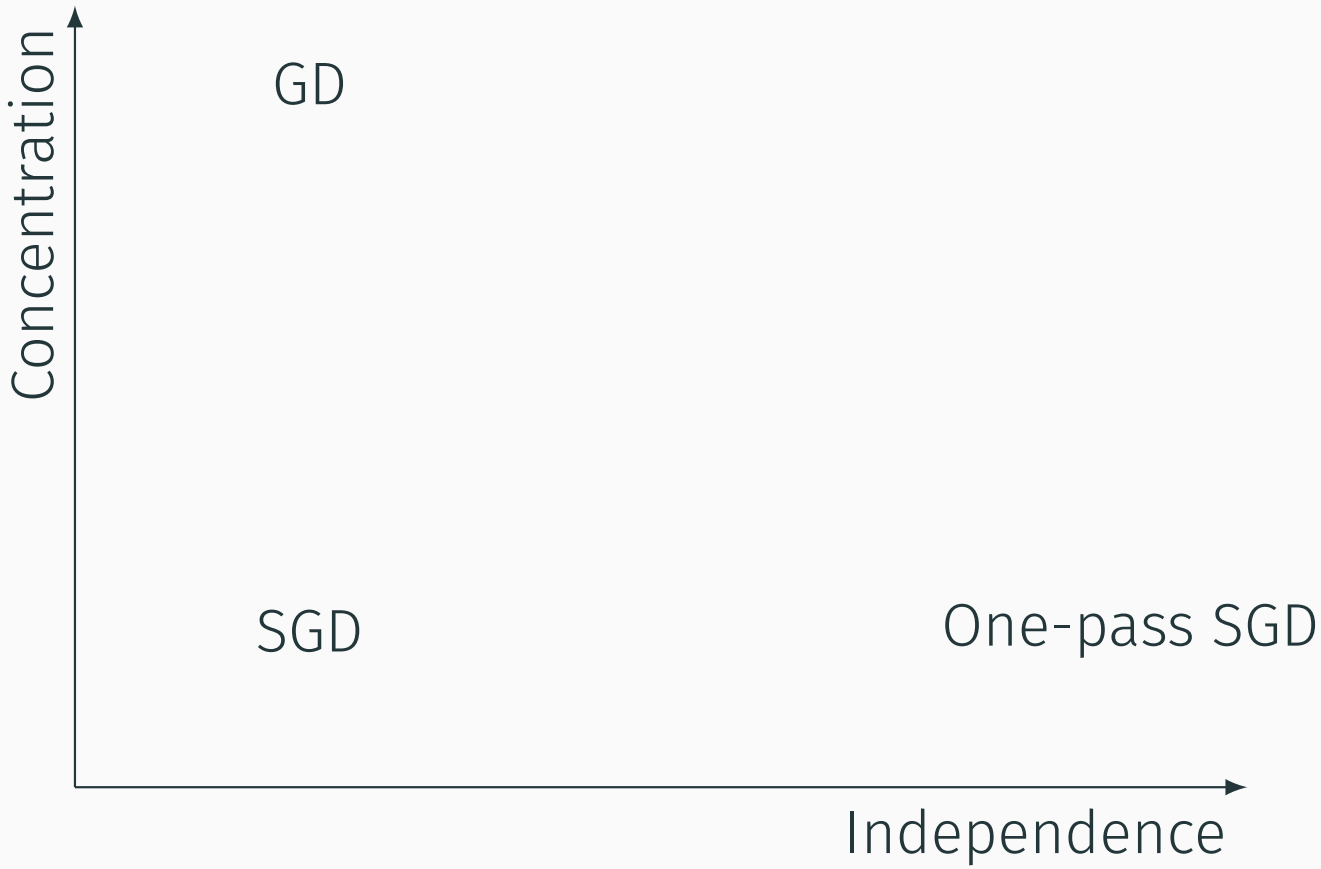
Why big gradient steps ?



Why big gradient steps ?



Why big gradient steps ?



Why big gradient steps ?

Our results: feature learning in the first layer

Setting

Network function:

$$\hat{f}(x; W, a) = \frac{1}{p} \sum_{i=1}^{n} a_i \sigma(\langle w_i, x \rangle)$$

Setting

Network function:

$$\hat{f}(x; W, a) = \frac{1}{p} \sum_{i=1}^{n} a_i \sigma(\langle w_i, x \rangle)$$

Low-dimensional, separable target function:

$$y = f^\star(x) = \sum_{k=1}^{r} \sigma_k^\star(\langle w_k^\star, x \rangle),$$

Setting

Network function:

$$\hat{f}(x; W, a) = \frac{1}{p} \sum_{i=1}^{n} a_i \sigma(\langle w_i, x \rangle)$$

Low-dimensional, separable target function:

$$y = f^\star(x) = \sum_{k=1}^{r} \sigma_k^\star(\langle w_k^\star, x \rangle),$$

Large-step GD on Gaussian data with quadratic loss:

$$w_i^{t+1} = w_i^t - \frac{\eta}{n} \sum_{\mu=1}^{n} \nabla \ell(f^\star(x^\mu), \hat{f}(x^\mu; W^t, a))$$

$$x^\mu \overset{i.i.d}{\sim} \mathcal{N}(0, I_d), \quad \ell(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2$$

Initialization

Fairly standard initialization...

$$w_i^0 \sim \mathrm{Unif}(\mathbb{S}^{d-1}), \quad a_i^0 \sim \mathrm{Unif}(\{-1, 1\})$$

Initialization

Fairly standard initialization...

$$w_i^0 \sim \mathrm{Unif}(\mathbb{S}^{d-1}), \quad a_i^0 \sim \mathrm{Unif}(\{-1, 1\})$$

... with some cheating:

$$w_i^0 = w_{n-i}^0, \quad a_i^0 = -a_{n-i}^0$$

Hermite decomposition and leap index

Hermite decomposition of teacher functions:

$$\sigma_k^\star(z) = \sum_{j=0}^{\infty} \frac{\mu_{k,j}^\star}{j!} H_j(z)$$

Hermite decomposition and leap index

Hermite decomposition of teacher functions:

$$\sigma_k^\star(z) = \sum_{j=1}^{\infty} \frac{\mu_{k,j}^\star}{j!} H_j(z)$$

Hermite decomposition and leap index

Hermite decomposition of teacher functions:

$$\sigma_k^\star(z) = \sum_{j=1}^{\infty} \frac{\mu_{k,j}^\star}{j!} H_j(z)$$

Leap index of $f^\star$:

$$\ell = \min\{j \geq 1 : \mu_{k,j}^\star \neq 0 \text{ for some } k \in [r]\}$$

Ben Arous et al. '21

Results: one gradient step

Main takeaway: phase transition at $n = \Theta(d^{\ell})$

Results: one gradient step

Main takeaway: phase transition at $n = \Theta(d^\ell)$

First case: batch size too small

Theorem (DKLPS '23)

Assume that $n = O(d^{\ell-\varepsilon})$ for sufficiently small ε. Then, whatever the choice of η, the first layer never learns: for all $i \in [p]$,

$$\|P_{V^\star} w_i\| = o(\|w_i\|)$$

Results: one gradient step

Second case: $\ell = 1$

Theorem (DKLPS '23)

Assume that $\ell = 1$, and $n = \Omega(d)$. Then, choosing $\eta = p$, the first layer learns: for all $i \in [p]$,

$$\|P_{V^\star} w_i\| = \Omega(\|w_i\|).$$

However, it only learns a linear subspace:

$$P_{V^\star} w_i \in \mathrm{span}(v^\star), \quad v^\star = \sum_{k=1}^{r} \mu_{k,1}^\star w_k^\star.$$

Ba et al. '22

Results: one gradient step

Third case: $\ell > 1$

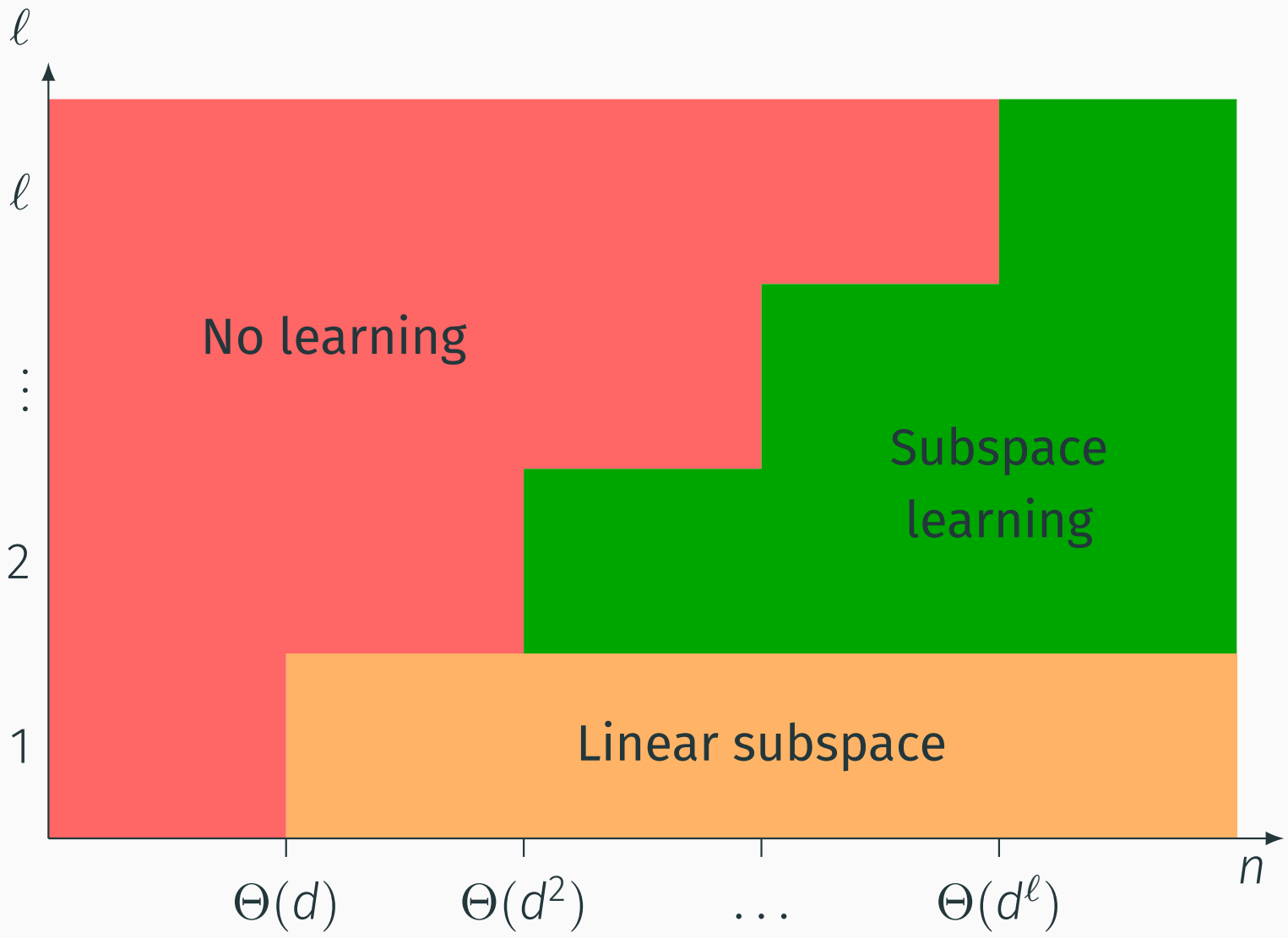
Theorem (DKLPS '23)

Assume that $\ell > 1$, and $n = \Omega(d^\ell)$. Then, choosing $\eta = pd^{\frac{\ell-1}{2}}$, the first layer learns, but still only a relevant subspace:

$$P_{V^\star} w_i \in U_\ell^\star, \quad U_\ell^\star = \mathrm{span}(\{w_k^\star : \mu_{k,\ell}^\star \neq 0\}).$$

Damian et al. '22

Results: one gradient step



Behind the scenes

We show that for any $i \in [p], k \in [r]$:

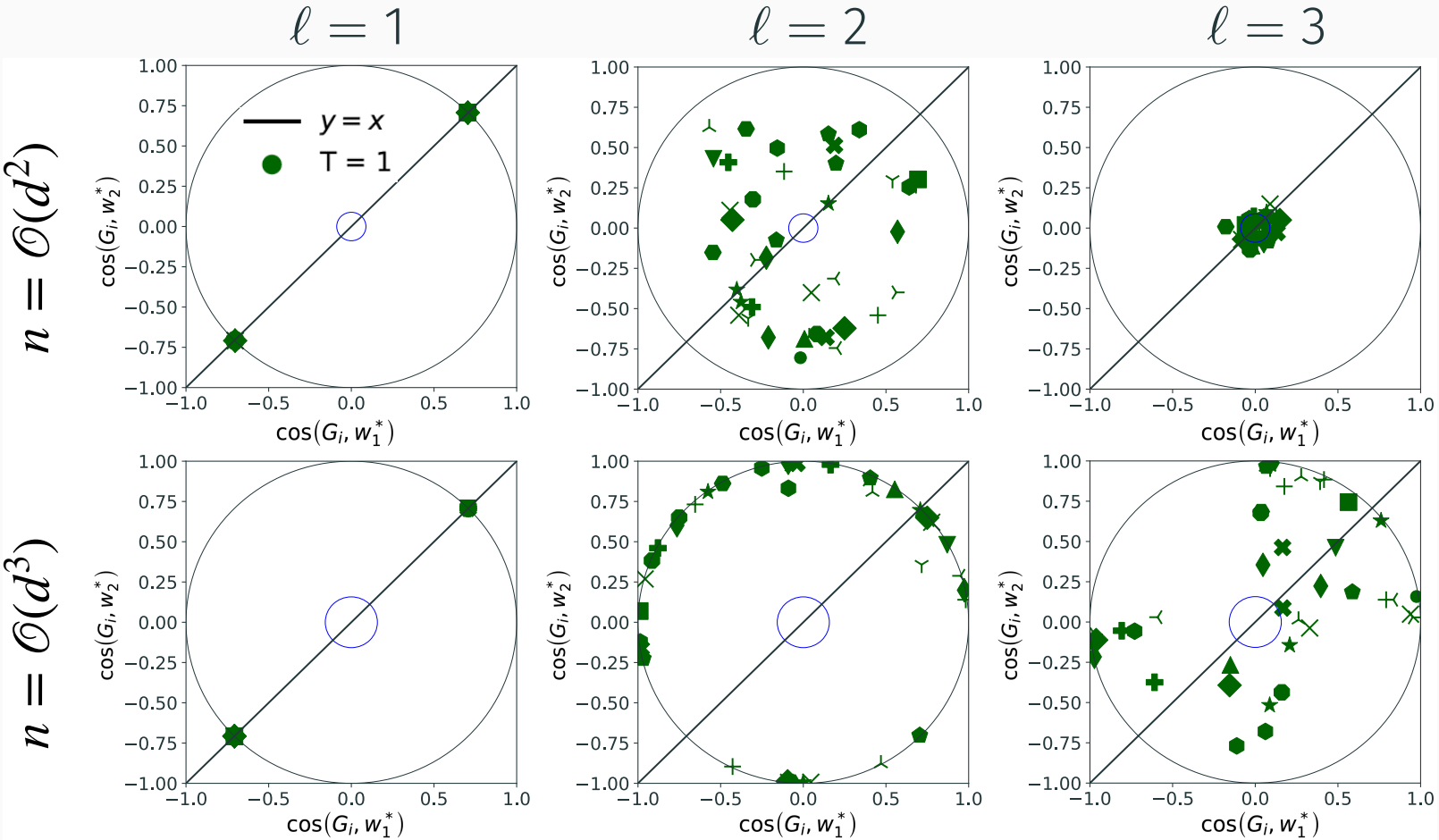
$$\langle w_i^1, w_k^\star \rangle \approx \frac{\eta a_i}{p} \cdot \mu_{k,\ell}^\star \cdot \underbrace{\langle w_i^0, w_k^\star \rangle}_{\sim d^{-1/2}}^{\ell-1}$$

$$\|w_i^1\| \asymp 1 + \frac{d\eta^2}{np^2}$$

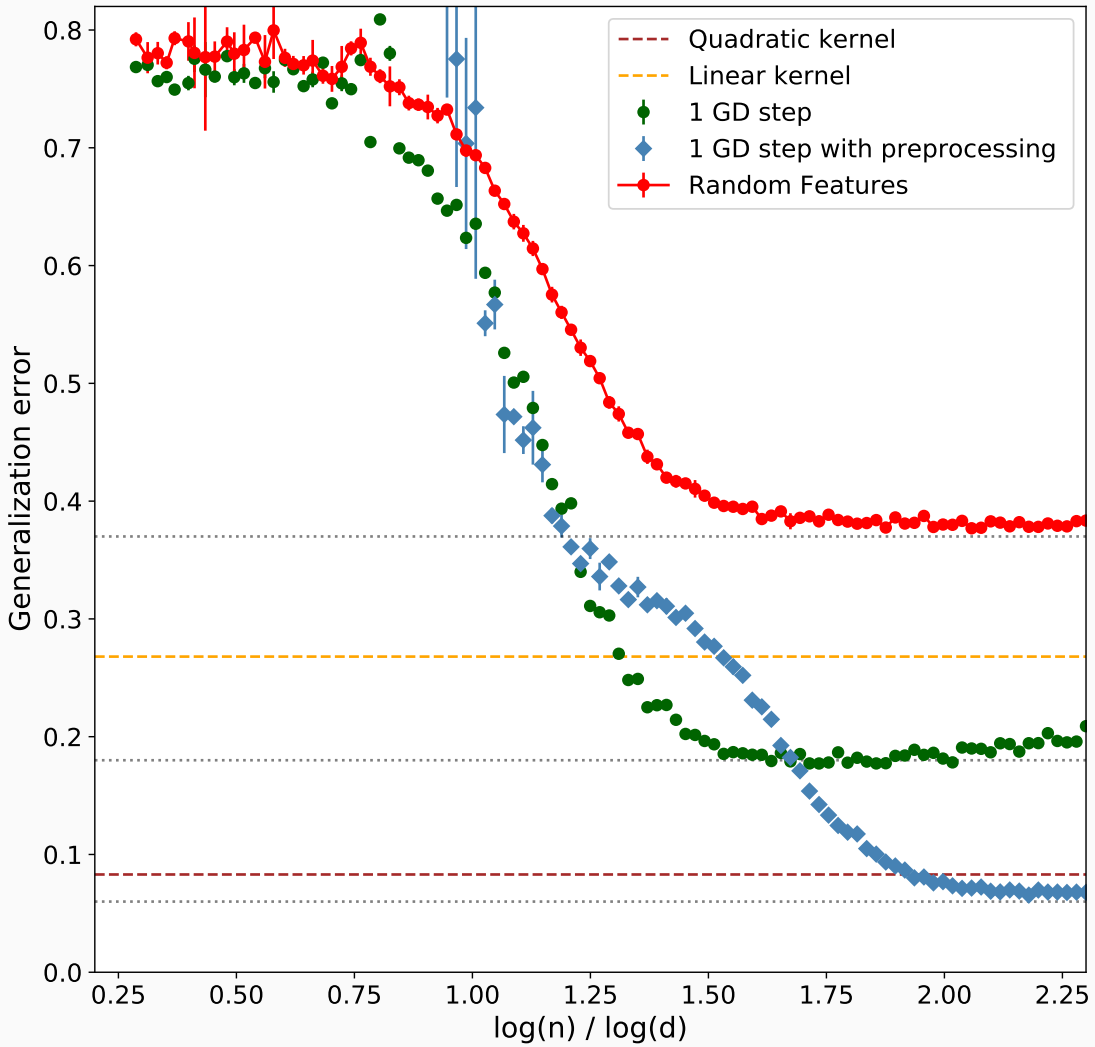
As a result:

- we need $\eta \asymp p d^{(\ell-1)/2}$ to have $\langle w_i^1, w_k^\star \rangle = \Omega(1)$
- this means that $\|w_i^1\| \asymp 1 + d^\ell/n \Rightarrow$ we need $n = \Omega(d^\ell)$
- Finally:
  - if $\ell = 1, \langle w_i^1, w_k^\star \rangle \approx a_i \mu_{k,\ell}^\star \to$ rank-one matrix !
  - otherwise, $\langle w_i^0, w_k^\star \rangle^{\ell-1}$ is a.s. full rank

Example (1)


$\ell = 1$
$\ell = 2$
$\ell = 3$
$n = \mathcal{O}(d^2)$
$n = \mathcal{O}(d^3)$
$y = x$
T = 1
$\cos(G_i, w_1^*)$
$\cos(G_i, w_2^*)$

Example (2)



$\ell = 1$

$\ell = 2$

Multiple gradient steps and the staircase property

Results: several gradient steps

Depends on the regime !

Small batch size:

Conjecture

If $n = O(d^{\ell-\varepsilon})$, whatever the choices of step-sizes $\eta_1, \dots \eta_T$, the first layer never learns even after T gradient steps.

"Hard" regime ($\ell > 1$): it's complicated

Results: several gradient steps

Depends on the regime !

Small batch size:

Conjecture

If $n = O(d^{\ell-\varepsilon})$, whatever the choices of step-sizes $\eta_1, \dots \eta_T$, the first layer never learns even after T gradient steps.

"Hard" regime ($\ell > 1$): it's complicated

What about the case $\ell = 1$?

A staircase ?

Main intuition: a learned direction can be used as a step to reach higher-order directions !

$\Rightarrow$ possible to learn new directions at each step

Abbe et al. '{21, 22, 23}

Main conjecture

Definition 2 (Subspace conditioning). *Let V be a vector space, and $U \subseteq V$ a subspace. For any function $f : V \to \mathbb{R}$, and $\mathbf{x} \in U$, we define the* conditional *function $f_{U,\mathbf{x}} : U^\perp \to \mathbb{R}$ as*

$$f_{U,\mathbf{x}}(\mathbf{x}^\perp) = f(\mathbf{x} + \mathbf{x}^\perp) \tag{13}$$

In short, the function $f_{U,\mathbf{x}}$ corresponds to f "conditioned" on the projection of its argument in U. Its first Hermite coefficient will be denoted as

$$\mu_{U,\mathbf{x}}(f) = \mathbb{E}_{\mathbf{x}^\perp \sim \mathcal{N}(0,I)} \left[\nabla_{\mathbf{x}^\perp} f_{U,\mathbf{x}}\right] \tag{14}$$

We are now equipped to state our conjecture:

Conjecture 1. *Assume that $n = \mathcal{O}(d)$, and $\eta = \sqrt{p}$. Define a sequence of nested subspaces $U_0^\star \subseteq U_1^\star \subseteq \cdots \subseteq U_t^\star \subseteq \ldots$ as*

- $U_0^\star = \{0\}$,
- *for any $t \geq 0$, $U_{t+1}^\star = U_t^\star \oplus \operatorname{span}\left(\{\mu_{U_t^\star,\mathbf{x}}(f^\star) : \mathbf{x} \in U_t^\star\}\right)$.*

Then, after t gradient steps of the form (6), the first layer learns exactly $U_t^\star$.

Main conjecture

Definition 2 (Subspace conditioning). *Let V be a vector space, and $U \subseteq V$ a subspace. For any function $f : V \to \mathbb{R}$, and $\mathbf{x} \in U$, we define the* conditional *function $f_{U,\mathbf{x}} : U^\perp \to \mathbb{R}$ as*

$$f_{U,\mathbf{x}}(\mathbf{x}^\perp) = f(\mathbf{x} + \mathbf{x}^\perp) \tag{13}$$

In short, the function $f_{U,\mathbf{x}}$ corresponds to f "conditioned" on the projection of its argument in U. Its first Hermite coefficient will be denoted as

$$\mu_{U,\mathbf{x}}(f) = \mathbb{E}_{\mathbf{x}^\perp \sim \mathcal{N}(0,I)} \left[\nabla_{\mathbf{x}^\perp} f_{U,\mathbf{x}}\right] \tag{14}$$

We are now equipped to state our conjecture:

Conjecture 1. *Assume that $n = \mathcal{O}(d)$, and $\eta = \sqrt{p}$. Define a sequence of nested subspaces $U_0^\star \subseteq U_1^\star \subseteq \cdots \subseteq U_t^\star \subseteq \ldots$ as*

- $U_0^\star = \{0\}$,
- *for any $t \geq 0$, $U_{t+1}^\star = U_t^\star \oplus \operatorname{span}\left(\{\mu_{U_t^\star,\mathbf{x}}(f^\star) : \mathbf{x} \in U_t^\star\}\right)$.*

Then, after t gradient steps of the form (6), the first layer learns exactly $U_t^\star$.

Informally: what is linear conditioned on the already learned directions can be learned!

Main conjecture

Definition 2 (Subspace conditioning). *Let V be a vector space, and $U \subseteq V$ a subspace. For any function $f : V \to \mathbb{R}$, and $\mathbf{x} \in U$, we define the* conditional *function $f_{U,\mathbf{x}} : U^\perp \to \mathbb{R}$ as*

$$f_{U,\mathbf{x}}(\mathbf{x}^\perp) = f(\mathbf{x} + \mathbf{x}^\perp) \tag{13}$$

In short, the function $f_{U,\mathbf{x}}$ corresponds to f "conditioned" on the projection of its argument in U. Its first Hermite coefficient will be denoted as

$$\mu_{U,\mathbf{x}}(f) = \mathbb{E}_{\mathbf{x}^\perp \sim \mathcal{N}(0,I)} \left[\nabla_{\mathbf{x}^\perp} f_{U,\mathbf{x}}\right] \tag{14}$$

We are now equipped to state our conjecture:

Conjecture 1. *Assume that $n = \mathcal{O}(d)$, and $\eta = \sqrt{p}$. Define a sequence of nested subspaces $U_0^\star \subseteq U_1^\star \subseteq \cdots \subseteq U_t^\star \subseteq \ldots$ as*

- $U_0^\star = \{0\}$,
- *for any $t \geq 0$, $U_{t+1}^\star = U_t^\star \oplus \operatorname{span}\left(\{\mu_{U_t^\star,\mathbf{x}}(f^\star) : \mathbf{x} \in U_t^\star\}\right)$.*

Then, after t gradient steps of the form (6), the first layer learns exactly $U_t^\star$.

Theorem (DKLPS '23)

Conjecture 1 is true, in expectation, for $T = 2$

Example (1)

$$r = 2, \quad \sigma_1^\star(z) = z + z^2 - 1, \quad \sigma_2^\star(z) = z - (z^2 - 1)$$

$$f^\star(x) = \langle x, w_1^\star \rangle + \langle x, w_2^\star \rangle + \langle x, w_1^\star \rangle^2 - \langle x, w_2^\star \rangle^2$$

Example (1)

$$r = 2, \quad \sigma_1^\star(z) = z + z^2 - 1, \quad \sigma_2^\star(z) = z - (z^2 - 1)$$

$$f^\star(x) = \langle x, \underbrace{w_1^\star + w_2^\star}_{v^\star} \rangle + \langle x, w_1^\star \rangle^2 - \langle x, w_2^\star \rangle^2$$

- First step: we learn $v^\star \propto w_1^\star + w_2^\star$

Example (1)

$$r = 2, \quad \sigma_1^\star(z) = z + z^2 - 1, \quad \sigma_2^\star(z) = z - (z^2 - 1)$$

$$f^\star(x) = \langle x, w_1^\star + w_2^\star \rangle + \langle x, w_1^\star + w_2^\star \rangle \langle x, w_1^\star - w_2^\star \rangle$$

- First step: we learn $v^\star \propto w_1^\star + w_2^\star$

Example (1)

$$r = 2, \quad \sigma_1^\star(z) = z + z^2 - 1, \quad \sigma_2^\star(z) = z - (z^2 - 1)$$

$$f^\star(x) = \langle x, w_1^\star + w_2^\star \rangle + \langle x, w_1^\star + w_2^\star \rangle \langle x, w_1^\star - w_2^\star \rangle$$

- First step: we learn $v^\star \propto w_1^\star + w_2^\star$
- Second step: $w_1^\star - w_2^\star$ is now linear !

Example (2)

$$r = 2, \quad \sigma_1^\star = \sigma_2^\star = x + x^2 - 1$$

$$f^\star(x) = \langle x, w_1^\star \rangle + \langle x, w_2^\star \rangle + \langle x, w_1^\star \rangle^2 + \langle x, w_2^\star \rangle^2 - 2$$

Example (2)

$$r = 2, \quad \sigma_1^\star = \sigma_2^\star = x + x^2 - 1$$

$$f^\star(x) = \langle x, \underbrace{w_1^\star + w_2^\star}_{v^\star} \rangle + \langle x, w_1^\star \rangle^2 + \langle x, w_2^\star \rangle^2 - 2$$

- First step: we learn $v^\star \propto w_1^\star + w_2^\star$

Example (2)

$$r = 2, \quad \sigma_1^\star = \sigma_2^\star = x + x^2 - 1$$

$$f^\star(x) = \langle x, w_1^\star + w_2^\star \rangle + \left\langle x, \frac{w_1^\star + w_2^\star}{\sqrt{2}} \right\rangle^2 - 1 + \left\langle x, \frac{w_1^\star - w_2^\star}{\sqrt{2}} \right\rangle^2 - 1$$

- First step: we learn $v^\star \propto w_1^\star + w_2^\star$

Example (2)

$$r = 2, \quad \sigma_1^\star = \sigma_2^\star = x + x^2 - 1$$

$$f^\star(x) = \langle x, w_1^\star + w_2^\star \rangle + \left\langle x, \frac{w_1^\star + w_2^\star}{\sqrt{2}} \right\rangle^2 - 1 + \left\langle x, \frac{w_1^\star - w_2^\star}{\sqrt{2}} \right\rangle^2 - 1$$

- First step: we learn $v^\star \propto w_1^\star + w_2^\star$
- Second step: no new learning 😦

Example (2)

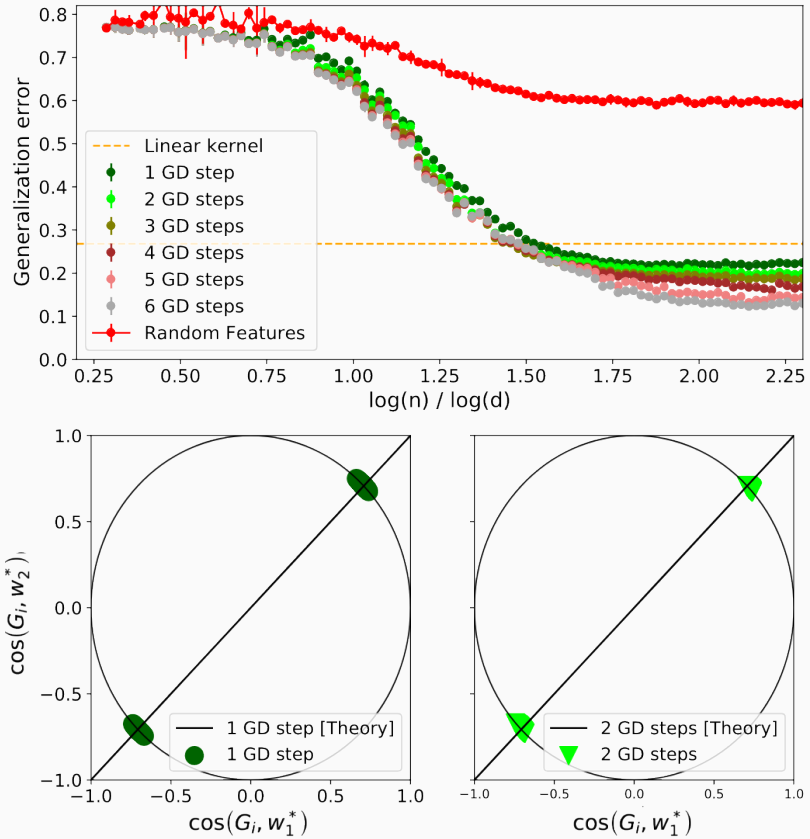
$$r = 2, \quad \sigma_1^\star = \sigma_2^\star = x + x^2 - 1$$

$$f^\star(x) = \langle x, w_1^\star + w_2^\star \rangle + \left\langle x, \frac{w_1^\star + w_2^\star}{\sqrt{2}} \right\rangle^2 - 1 + \left\langle x, \frac{w_1^\star - w_2^\star}{\sqrt{2}} \right\rangle^2 - 1$$

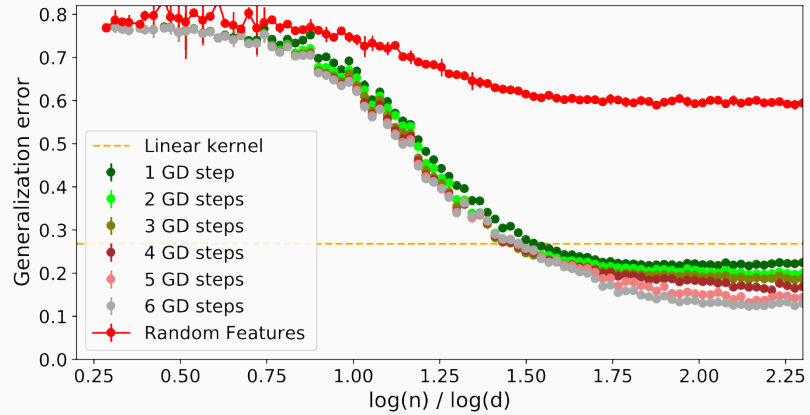
- First step: we learn $v^\star \propto w_1^\star + w_2^\star$
- Second step: no new learning 😦

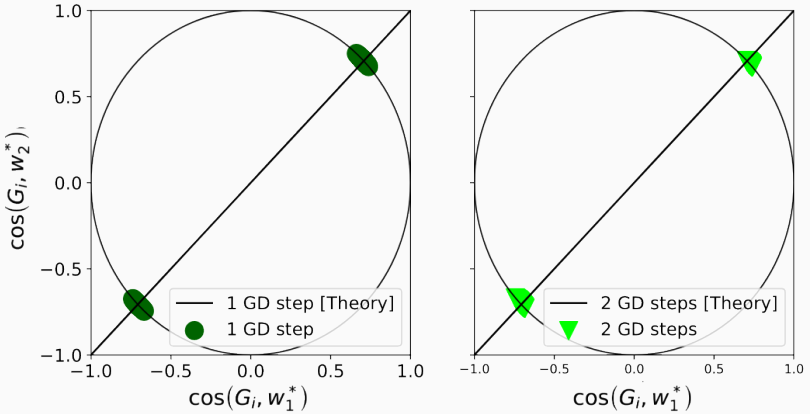
Always happens if $\sigma_1^\star = \cdots = \sigma_r^\star$

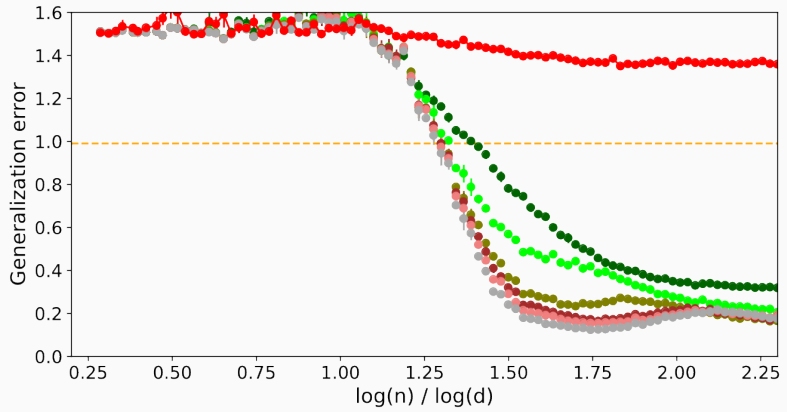
Example (1+2)

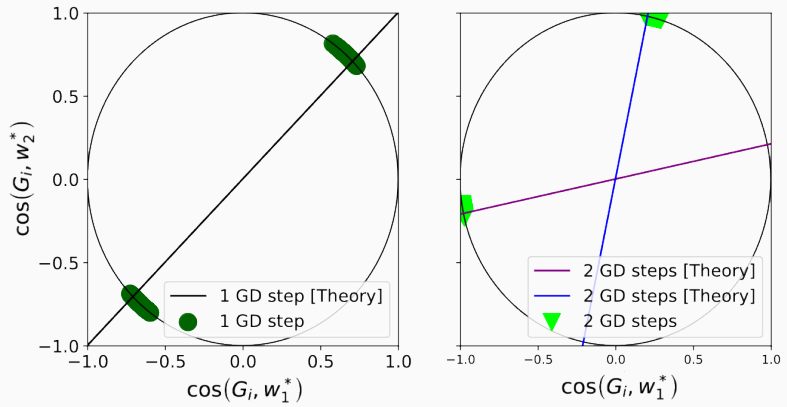

Generalization error
log(n) / log(d)
Linear kernel
1 GD step
2 GD steps
3 GD steps
4 GD steps
5 GD steps
6 GD steps
Random Features
1 GD step [Theory]
1 GD step
2 GD steps [Theory]
2 GD steps
$\cos(G_i, w_1^*)$
$\cos(G_i, w_2^*)$

Example (1+2)


Generalization error
log(n) / log(d)
Linear kernel
1 GD step
2 GD steps
3 GD steps
4 GD steps
5 GD steps
6 GD steps
Random Features


$\cos(G_i, w_2^*)$
$\cos(G_i, w_1^*)$
1 GD step [Theory]
1 GD step
2 GD steps [Theory]
2 GD steps


Generalization error
log(n) / log(d)


$\cos(G_i, w_2^*)$
$\cos(G_i, w_1^*)$
1 GD step [Theory]
1 GD step
2 GD steps [Theory]
2 GD steps [Theory]
2 GD steps

Example (3)

$$f_1^\star(x) = \langle x, w_1^\star \rangle + \langle x, w_1^\star \rangle \langle x, w_2^\star \rangle + \langle x, w_1^\star \rangle \langle x, w_3^\star \rangle$$

$$f_2^\star(x) = \langle x, w_1^\star \rangle + \langle x, w_1^\star \rangle \langle x, w_2^\star \rangle + H_3(\langle x, w_1^\star \rangle) \langle x, w_3^\star \rangle$$

Example (3)

$$f_1^\star(x) = \langle x, w_1^\star \rangle + \langle x, w_1^\star \rangle \langle x, w_2^\star \rangle + \langle x, w_1^\star \rangle \langle x, w_3^\star \rangle$$
$$f_2^\star(x) = \langle x, w_1^\star \rangle + \langle x, w_1^\star \rangle \langle x, w_2^\star \rangle + H_3(\langle x, w_1^\star \rangle) \langle x, w_3^\star \rangle$$

- First step: we learn $v^\star = w_1^\star$
- Second step:

Example (3)

$$f_1^\star(x) = \langle x, w_1^\star \rangle + \langle x, w_1^\star \rangle \langle x, w_2^\star + w_3^\star \rangle$$
$$f_2^\star(x) = \langle x, w_1^\star \rangle + \langle x, w_1^\star \rangle \langle x, w_2^\star \rangle + H_3(\langle x, w_1^\star \rangle) \langle x, w_3^\star \rangle$$

- First step: we learn $v^\star = w_1^\star$
- Second step:
  - First case: we learn $w_2^\star + w_3^\star$

Example (3)

$$f_1^\star(x) = \langle x, w_1^\star \rangle + \langle x, w_1^\star \rangle \langle x, w_2^\star + w_3^\star \rangle$$
$$f_2^\star(x) = \langle x, w_1^\star \rangle + \langle x, w_1^\star \rangle \langle x, w_2^\star \rangle + H_3(\langle x, w_1^\star \rangle) \langle x, w_3^\star \rangle$$

- First step: we learn $v^\star = w_1^\star$
- Second step:
  - First case: we learn $w_2^\star + w_3^\star$
  - Second case: (H_1, H_3) are linearly independent
    $\Rightarrow$ we learn $\mathrm{span}(w_2^\star, w_3^\star)$!

Training the second layer

Training procedure

We decouple the training of the first and the second layer:

- first, we do T training steps for the first layer
- then, we train the second layer

Training procedure

We decouple the training of the first and the second layer:

- first, we do T training steps for the first layer
- then, we train the second layer

$$a^\star = \min_{a \in \mathbb{R}^p} \frac{1}{n} \sum_{\mu=1}^{n} \ell(f^\star(x^\mu), \hat{f}(x^\mu; W^T, a)) + \frac{\lambda}{p} \|a\|^2$$

Training procedure

We decouple the training of the first and the second layer:

- first, we do T training steps for the first layer
- then, we train the second layer

$$a^\star = \min_{a \in \mathbb{R}^p} \frac{1}{n} \sum_{\mu=1}^{n} \ell(f^\star(x^\mu), \hat{f}(x^\mu; W^T, a)) + \frac{\lambda}{p} \|a\|^2$$

Same batch size n, but new data!

Finite p regime

Theorem (DKLPS '23)

When $p = O(1)$, if W^T only learned a subspace $U^\star \subseteq V^\star$, then for any bounded second layer a,

$$\mathbb{E}\left[\left(f^\star(x) - \hat{f}(x; W^T, a)\right)^2\right] \geq \mathbb{E}[\mathrm{Var}(f^\star(x)|P_{U^\star}x)] + o(1)$$

Finite p regime

Theorem (DKLPS '23)

When $p = O(1)$, if W^T only learned a subspace $U^\star \subseteq V^\star$, then for any bounded second layer a,

$$\mathbb{E}\left[\left(f^\star(x) - \hat{f}(x; W^T, a)\right)^2\right] \geq \mathbb{E}[\mathrm{Var}(f^\star(x)|P_{U^\star}x)] + o(1)$$

Informally: we cannot learn anything outside of $U^\star$!

Finite p regime

Theorem (DKLPS '23)

When $p = O(1)$, if W^T only learned a subspace $U^\star \subseteq V^\star$, then for any bounded second layer a,

$$\mathbb{E}\left[\left(f^\star(x) - \hat{f}(x; W^T, a)\right)^2\right] \geq \mathbb{E}[\mathrm{Var}(f^\star(x)|P_{U^\star}x)] + o(1)$$

Informally: we cannot learn anything outside of $U^\star$!

Conjecture

The converse bound is true if p is large enough

Damian et al. '22

Large p regime: random feature regression

Even for $T = 0$, training the second layer picks up some signal

$\Rightarrow$ random feature regression

Large p regime: random feature regression

Even for $T = 0$, training the second layer picks up some signal $\Rightarrow$ random feature regression

Very well studied, most recent relevant result:

Theorem (MMM '22, Informal)

Assume that $n \asymp d^{\kappa_1}, p \asymp d^{\kappa_2}$. Then

$$\mathbb{E}\left[\left(f^\star(x) - \hat{f}(x; W^T, a^\star)\right)^2\right] = \|P_{>\kappa} f^\star\|_{\mathcal{L}^2}^2 + o(1)$$

where $\kappa = \min(\kappa_1, \kappa_2)$.

Large p regime: random feature regression

Even for $T = 0$, training the second layer picks up some signal $\Rightarrow$ random feature regression

Very well studied, most recent relevant result:

Theorem (MMM '22, Informal)

Assume that $n \asymp d^{\kappa_1}, p \asymp d^{\kappa_2}$. Then

$$\mathbb{E}\left[\left(f^\star(x) - \hat{f}(x; W^T, a^\star)\right)^2\right] = \|P_{>\kappa} f^\star\|_{\mathcal{L}^2}^2 + o(1)$$

where $\kappa = \min(\kappa_1, \kappa_2)$.

Possible to learn a polynomial approximation of degree κ if and only if $n, p = \Omega(d^\kappa)$.

What about with feature learning ?

Definition 3. *Let V be a vector space, and $U \subseteq V$ a subspace. For any $\ell \geq 0$, we define the space $\mathcal{P}_{U,\ell}$ of functions $f : V \to \mathbb{R}$ such that for any $\boldsymbol{x} \in U$, the function $f_{U,\boldsymbol{x}}$ introduced in Definition 2 is a polynomial of degree at most ℓ.*

Simply put, the space $\mathcal{P}_{U,\ell}$ consists of polynomials in $\boldsymbol{x}^\perp$ of degree at most ℓ, whose coefficients can be functions of $\boldsymbol{x}$. We will denote by $P_{U,\leq\ell}$ and $P_{U,>\ell}$ the projections on $\mathcal{P}_{U,\ell}$ and $\mathcal{P}_{U,\ell}^\perp$ in $\ell^2(\mathbb{R}, \gamma)$, respectively, where γ is the Gaussian measure. Then, we conjecture the following:

Conjecture 3. *Assume that $p = \mathcal{O}(d^{\kappa_1})$ and $n = \mathcal{O}(d^{\kappa_2})$, and that the first layer W only learns a subspace $U^\star \subseteq V^\star$. Then, if $\hat{\boldsymbol{a}}$ is obtained as in Eq. (7) for any value of λ,*

$$\mathbb{E}\left[\left(f^\star(\boldsymbol{z}) - \hat{f}(\boldsymbol{z}; W, \hat{\boldsymbol{a}})\right)^2\right] \geq \|P_{U^\star,>\kappa} f^\star\|_\gamma^2 + o(1), \tag{19}$$

where $\kappa = \min(\kappa_1, \kappa_2)$ and $\|\cdot\|_\gamma$ is the norm in $\ell^2(\mathbb{R}, \gamma)$.

What about with feature learning ?

Definition 3. *Let V be a vector space, and $U \subseteq V$ a subspace. For any $\ell \geq 0$, we define the space $\mathcal{P}_{U,\ell}$ of functions $f : V \to \mathbb{R}$ such that for any $\boldsymbol{x} \in U$, the function $f_{U,\boldsymbol{x}}$ introduced in Definition 2 is a polynomial of degree at most ℓ.*

Simply put, the space $\mathcal{P}_{U,\ell}$ consists of polynomials in $\boldsymbol{x}^\perp$ of degree at most ℓ, whose coefficients can be functions of $\boldsymbol{x}$. We will denote by $P_{U,\leq\ell}$ and $P_{U,>\ell}$ the projections on $\mathcal{P}_{U,\ell}$ and $\mathcal{P}^\perp_{U,\ell}$ in $\ell^2(\mathbb{R},\gamma)$, respectively, where γ is the Gaussian measure. Then, we conjecture the following:

Conjecture 3. *Assume that $p = \mathcal{O}(d^{\kappa_1})$ and $n = \mathcal{O}(d^{\kappa_2})$, and that the first layer W only learns a subspace $U^\star \subseteq V^\star$. Then, if $\hat{\boldsymbol{a}}$ is obtained as in Eq. (7) for any value of λ,*

$$\mathbb{E}\left[\left(f^\star(\boldsymbol{z}) - \hat{f}(\boldsymbol{z}; W, \hat{\boldsymbol{a}})\right)^2\right] \geq \|P_{U^\star,>\kappa} f^\star\|_\gamma^2 + o(1), \tag{19}$$

where $\kappa = \min(\kappa_1, \kappa_2)$ and $\|\cdot\|_\gamma$ is the norm in $\ell^2(\mathbb{R},\gamma)$.

Informally:

- on the directions learned by W, we learn everything
- on the directions not learned, we cannot do better than a random feature model

Example

$$f^\star(x) = x_1 + H_2(x_2) + H_2(x_1)x_2 + H_4(x_1)H_2(x_2)x_3 + H_2(x_2)H_2(x_3)$$

$n \asymp d$	$n \asymp d$	$n \asymp d^2$	$n \asymp d^2$	$n \asymp d^2$
$p \asymp 1$	$p \asymp d$	$p \asymp 1$	$p \asymp d$	$p \asymp d^2$

- Learned direction(s):
- Final fitted function: see equation

$x_i = \langle x, w_i^\star \rangle$

Example

$$f^\star(x) = \underline{x_1} + H_2(x_2) + H_2(x_1)x_2 + H_4(x_1)H_2(x_2)x_3 + H_2(x_2)H_2(x_3)$$

$n \asymp d$	$n \asymp d$	$n \asymp d^2$	$n \asymp d^2$	$n \asymp d^2$
$p \asymp 1$	$p \asymp d$	$p \asymp 1$	$p \asymp d$	$p \asymp d^2$

- Learned direction(s): $w_1^\star$
- Final fitted function: see equation

$x_i = \langle x, w_i^\star \rangle$

Example

$$f^\star(x) = \underline{x_1} + H_2(x_2) + \underline{H_2(x_1)x_2} + H_4(x_1)H_2(x_2)x_3 + H_2(x_2)H_2(x_3)$$

$n \asymp d$	$n \asymp d$	$n \asymp d^2$	$n \asymp d^2$	$n \asymp d^2$
$p \asymp 1$	$p \asymp d$	$p \asymp 1$	$p \asymp d$	$p \asymp d^2$

- Learned direction(s): $w_1^\star$
- Final fitted function: see equation

$x_i = \langle x, w_i^\star \rangle$

Example

$$f^\star(x) = \underline{x_1} + \underline{H_2(x_2)} + \underline{H_2(x_1)x_2} + H_4(x_1)H_2(x_2)x_3 + H_2(x_2)H_2(x_3)$$

$n \asymp d$ $p \asymp 1$ | $n \asymp d$ $p \asymp d$ | $\boxed{n \asymp d^2 \quad p \asymp 1}$ | $n \asymp d^2$ $p \asymp d$ | $n \asymp d^2$ $p \asymp d^2$

- Learned direction(s): $\mathrm{span}(w_1^\star, w_2^\star)$
- Final fitted function: see equation

$x_i = \langle x, w_i^\star \rangle$

Example

$$f^\star(x) = \underline{x_1} + \underline{H_2(x_2)} + \underline{H_2(x_1)x_2} + \underline{H_4(x_1)H_2(x_2)x_3} + H_2(x_2)H_2(x_3)$$

$n \asymp d$	$n \asymp d$	$n \asymp d^2$	$n \asymp d^2$	$n \asymp d^2$
$p \asymp 1$	$p \asymp d$	$p \asymp 1$	$p \asymp d$	$p \asymp d^2$

- Learned direction(s): $\mathrm{span}(w_1^\star, w_2^\star)$
- Final fitted function: see equation

$x_i = \langle x, w_i^\star \rangle$

Example

$$f^\star(x) = \underline{x_1} + \underline{H_2(x_2)} + \underline{H_2(x_1)x_2} + \underline{H_4(x_1)H_2(x_2)x_3} + \underline{H_2(x_2)H_2(x_3)}$$

$n \asymp d$ $p \asymp 1$ | $n \asymp d$ $p \asymp d$ | $n \asymp d^2$ $p \asymp 1$ | $n \asymp d^2$ $p \asymp d$

$$\boxed{\begin{array}{c} n \asymp d^2 \\ p \asymp d^2 \end{array}}$$

- Learned direction(s): $\mathrm{span}(w_1^\star, w_2^\star)$
- Final fitted function: see equation

$x_i = \langle x, w_i^\star \rangle$

The linear regime and Gaussian equivalence

Second layer training equivalent to:

$$a^\star = \min_{a\in\mathbb{R}^p} \frac{1}{n}\sum_{\mu=1}^{n}\left(\frac{1}{p}\langle a, \phi_{CK}(x^\mu)\rangle - f^\star(x^\mu)\right)^2 + \frac{\lambda}{p}\|a\|^2$$

with the conjugate kernel features $\phi_{CK}(x) = \sigma(Wx)$.

The linear regime and Gaussian equivalence

Second layer training equivalent to:

$$a^\star = \min_{a\in\mathbb{R}^p} \frac{1}{n}\sum_{\mu=1}^{n}\left(\frac{1}{p}\langle a, \phi_{CK}(x^\mu)\rangle - f^\star(x^\mu)\right)^2 + \frac{\lambda}{p}\|a\|^2$$

with the conjugate kernel features $\phi_{CK}(x) = \sigma(Wx)$. In the linear regime $n \asymp p \asymp d$, with random features:

$$\phi_{CK}(x) \simeq \mu\mathbf{1} + \Phi x + \Psi^{1/2}\xi$$

Mei and Montanari '19, Hu and Lu '20

The linear regime and Gaussian equivalence

Second layer training equivalent to:

$$a^\star = \min_{a\in\mathbb{R}^p} \frac{1}{n}\sum_{\mu=1}^{n}\left(\frac{1}{p}\langle a, \phi_{CK}(x^\mu)\rangle - f^\star(x^\mu)\right)^2 + \frac{\lambda}{p}\|a\|^2$$

with the conjugate kernel features $\phi_{CK}(x) = \sigma(Wx)$. In the linear regime $n \asymp p \asymp d$, with random features:

$$\phi_{CK}(x) \simeq \mu\mathbf{1} + \Phi x + \Psi^{1/2}\xi$$

This is a (Gaussian) noisy linear model!

Mei and Montanari '19, Hu and Lu '20

In the feature learning regime

We can show that the whole gradient update is rank-one:

$$W^1 \approx W^0 + \lambda v v^\top$$

Accordingly, we decompose $x = x_v v + x^\perp$

Ba et al. '22

In the feature learning regime

We can show that the whole gradient update is rank-one:

$$W^1 \approx W^0 + \lambda v v^\top$$

Accordingly, we decompose $x = x_v v + x^\perp$

Theorem (Conditional Gaussian equivalence, [DKLPS '23])

In the linear regime $n \asymp p \asymp d$,

$$\phi_{CK}(x) \simeq \mu(x_v) + \Phi(x_v)x^\perp + \Psi(x_v)^{1/2}\xi$$

Ba et al. '22

Thank you !

arXiv:2305.18270