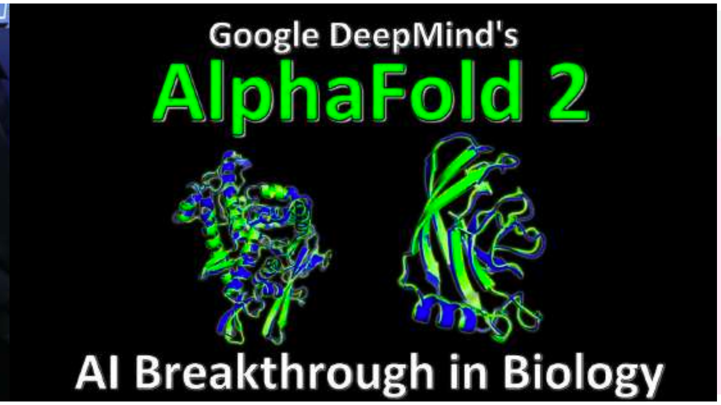
Overparamaterized Learning Beyond the Lazy Regime

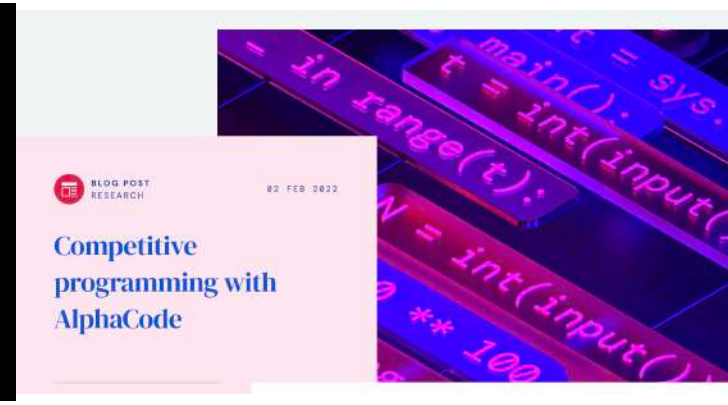
Mahdi Soltanolkotabi
ECE, CS, & ISE
USC

Erice, Italy
May 24, 2022

Many success stories

Modern learning algorithms have been extremely successful





Why do we need Theoretical Foundations?

Catastrophic Failures

Microsoft silences its new A.I. bot Tay, after Twitter users teach it racism [Updated]

Sarah Perez @sarahintampa / 7:16 AM PDT • March 24, 2016 Comment

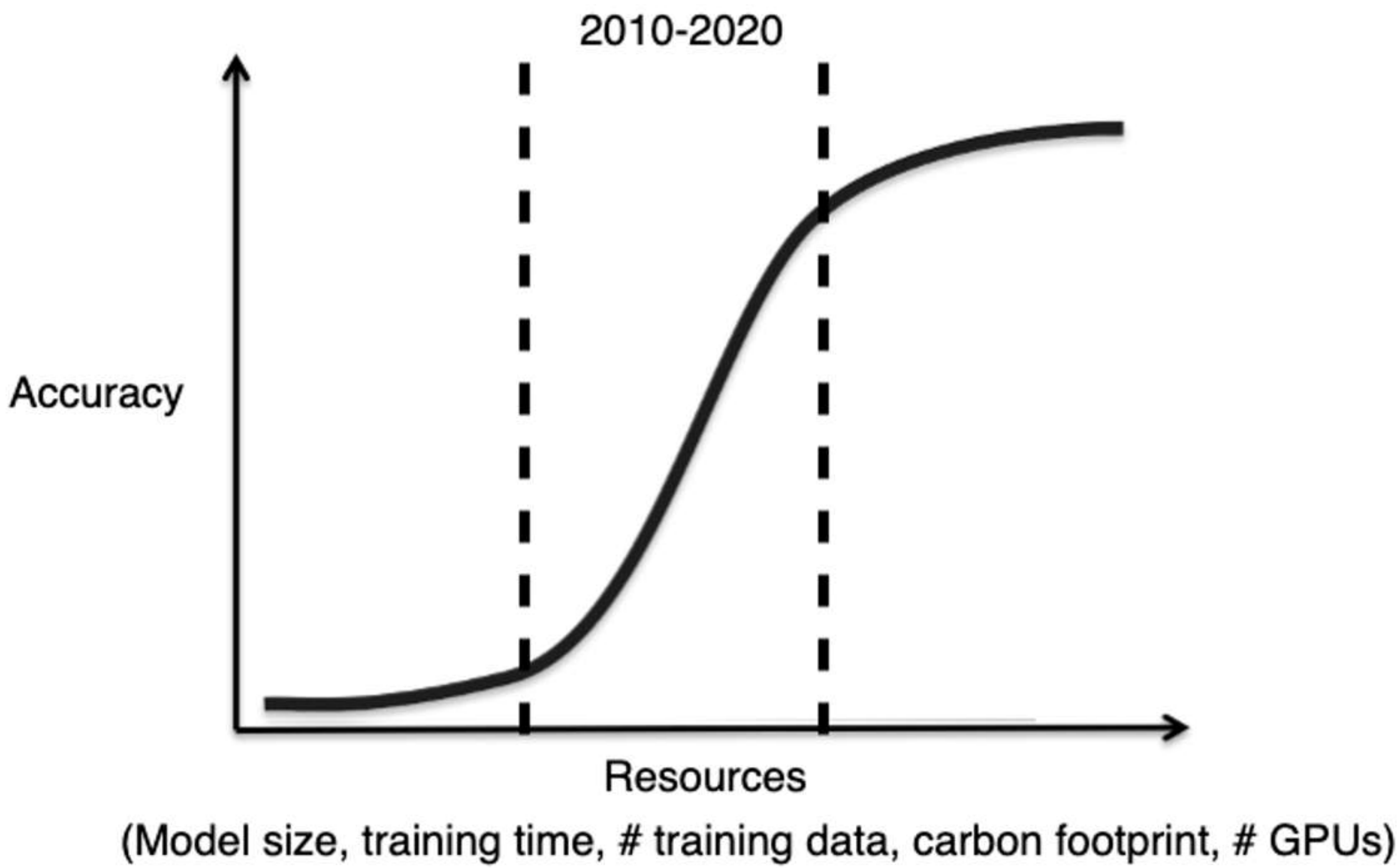
The Grim Conclusions of the Largest-Ever Study of Fake News

Tesla's "Full Self Driving" Beta Is Just Laughably Bad and Potentially Dangerous

If you think we're anywhere near fully autonomous cars, this video might convince you otherwise.

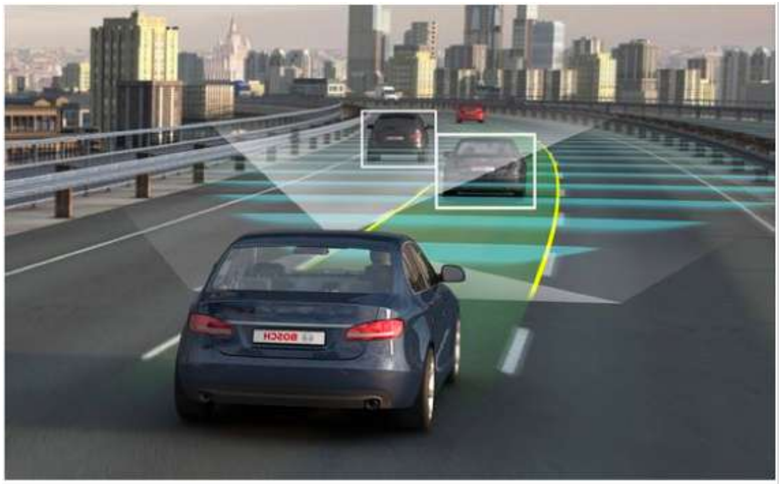
BY MACK HOGAN MAR 19, 2021

Hitting the S-curve



Need more principled understanding...

Modern learning algorithms increasingly used in human facing services



Existing Foundations?

A contemporary title for papers/talks:

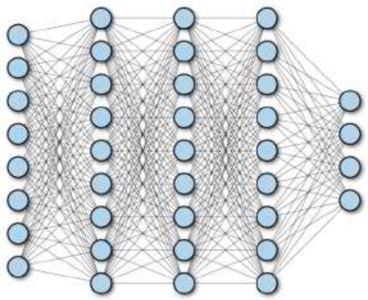
Theoretical Foundations for X

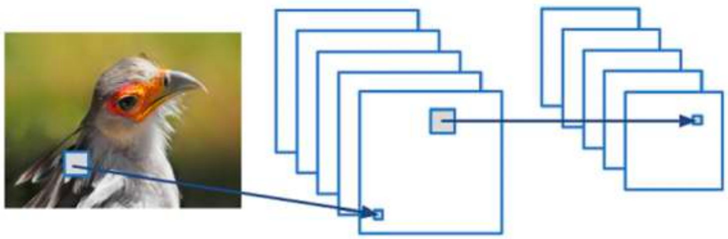
X= deep learning, Reinforcement learning, AI, …

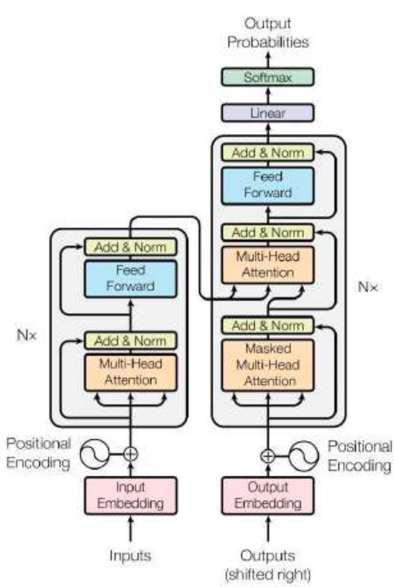
Why do we Need “Stronger” Foundations?

Answer I: Inability to explain contemporary practices

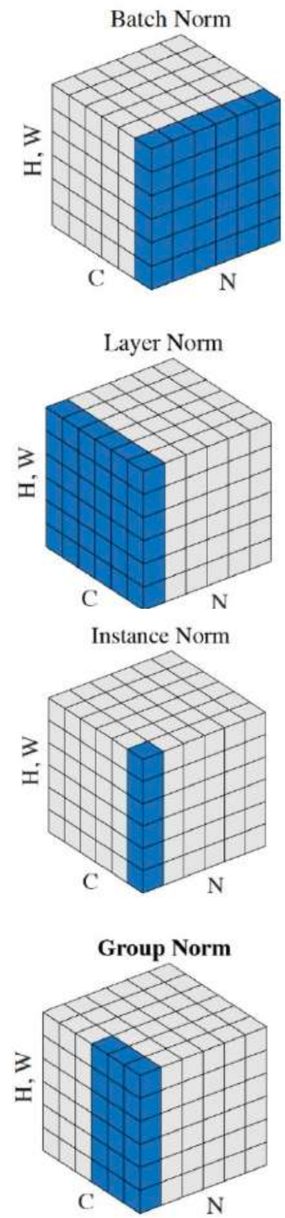
Choice of architecture



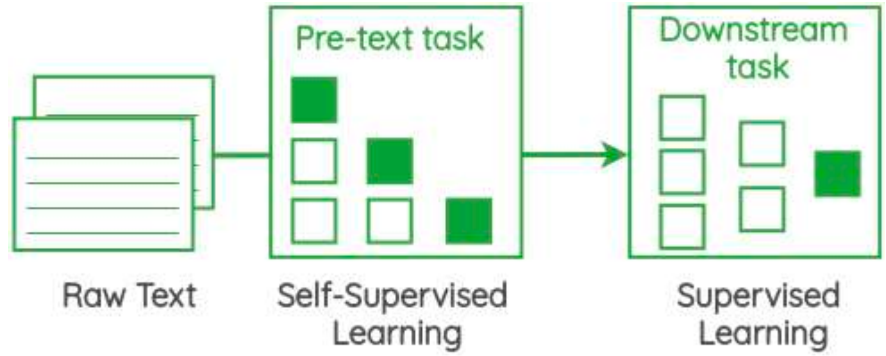




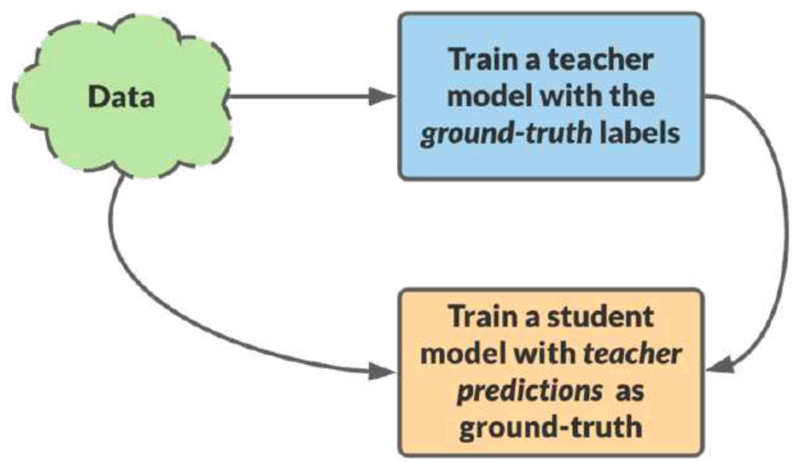
Normalization



Representation Learning/ pre-training+fine tuning

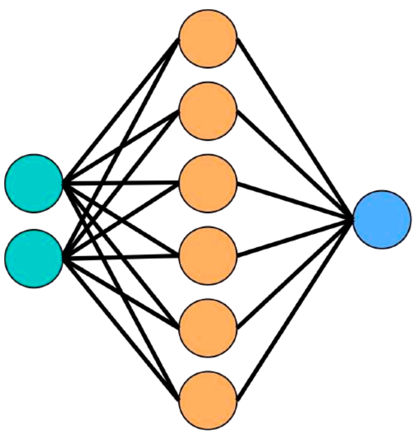


Distillation

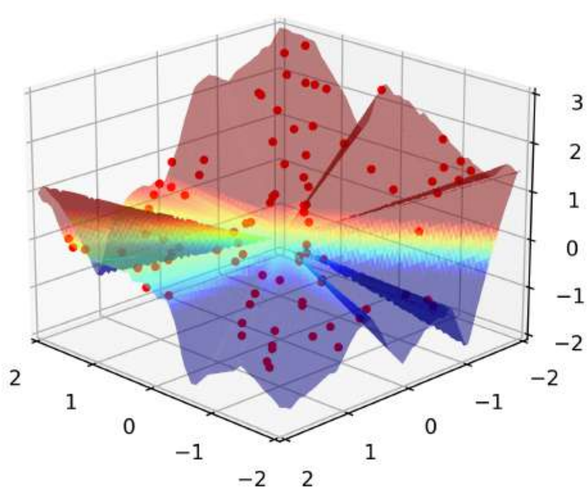


Answer II: current theory fails even in toy settings…

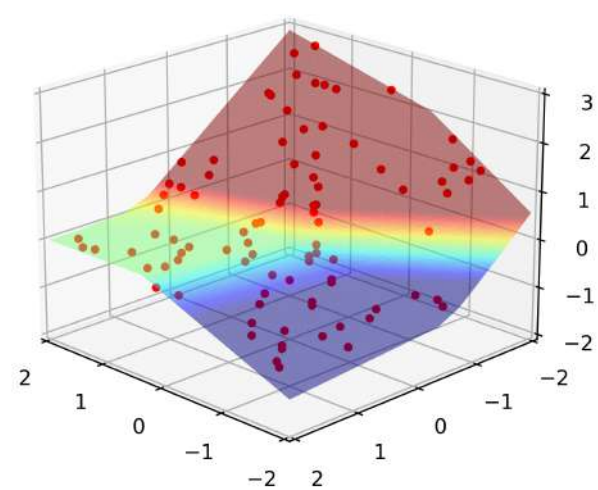
Existing theory operates with unrealistic hyper-parameter choices
(very small step size, very wide networks, very large init. scale, etc.)



theory hyperparameters



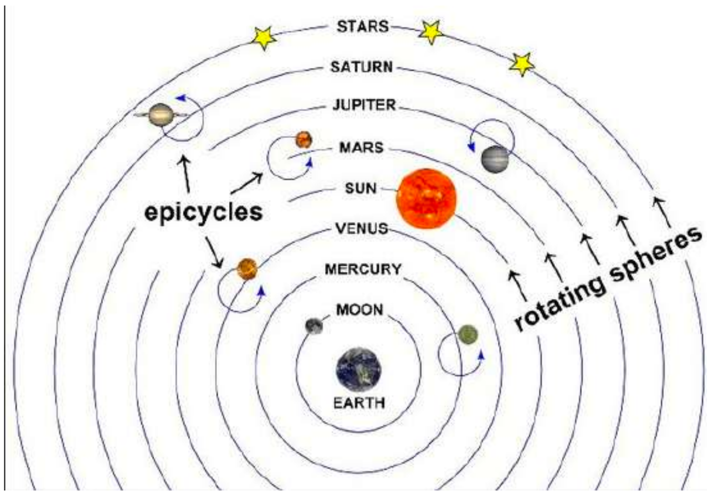
practical hyperparameters



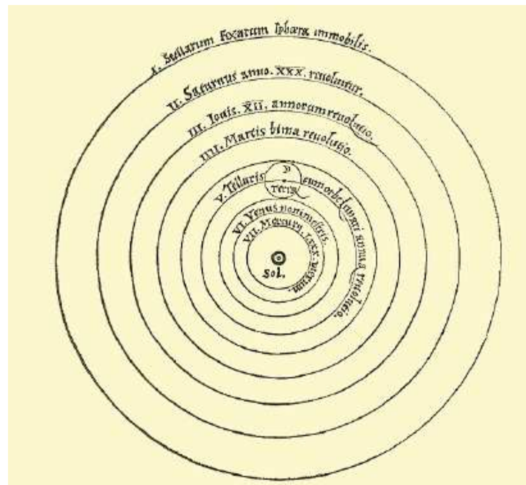
existing theory does not apply in practical regimes…

Historical analogy to theory of physics

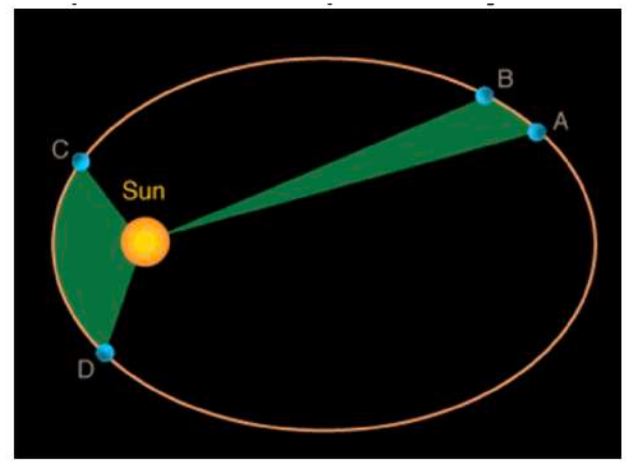
Ptolemy's model



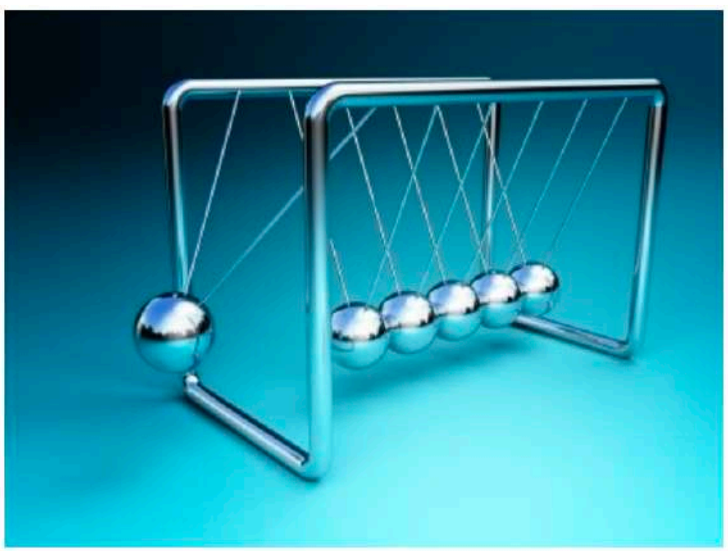
Copernican heliocentrism



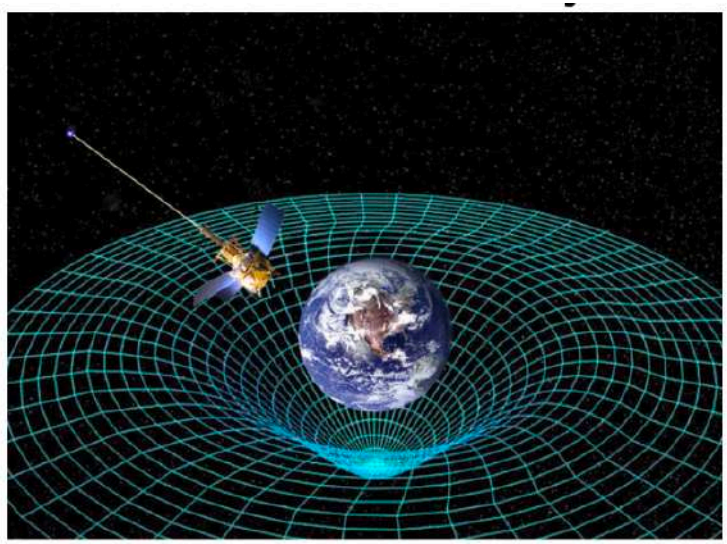
Kepler's law of planetary motion



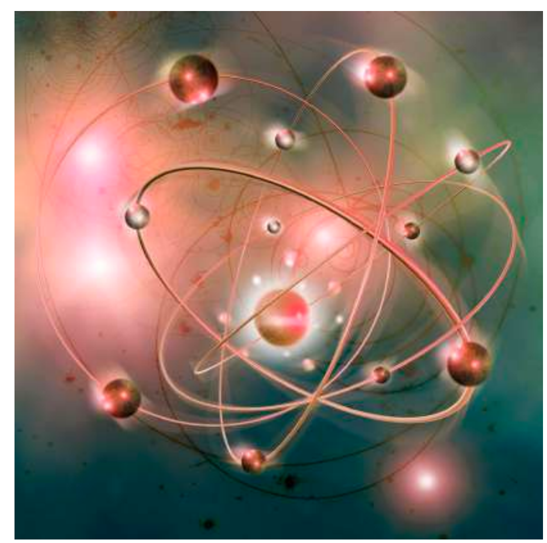
Newtonian Mechanics



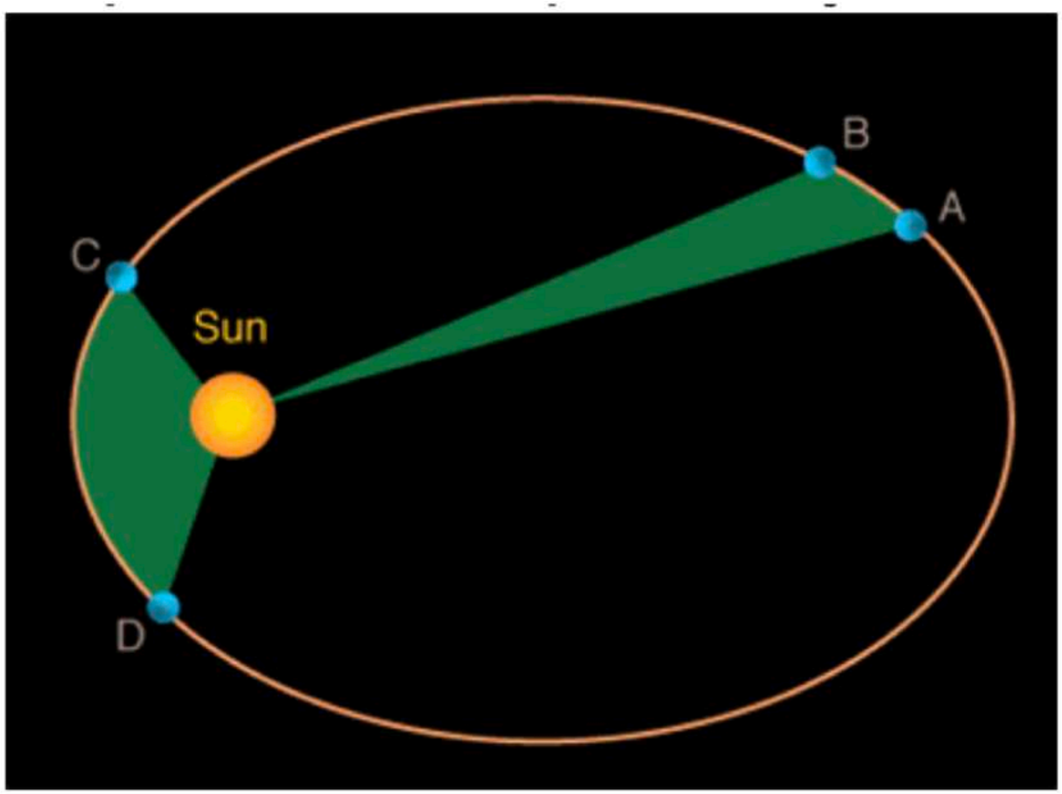
Relativity



Quantum Mechanics



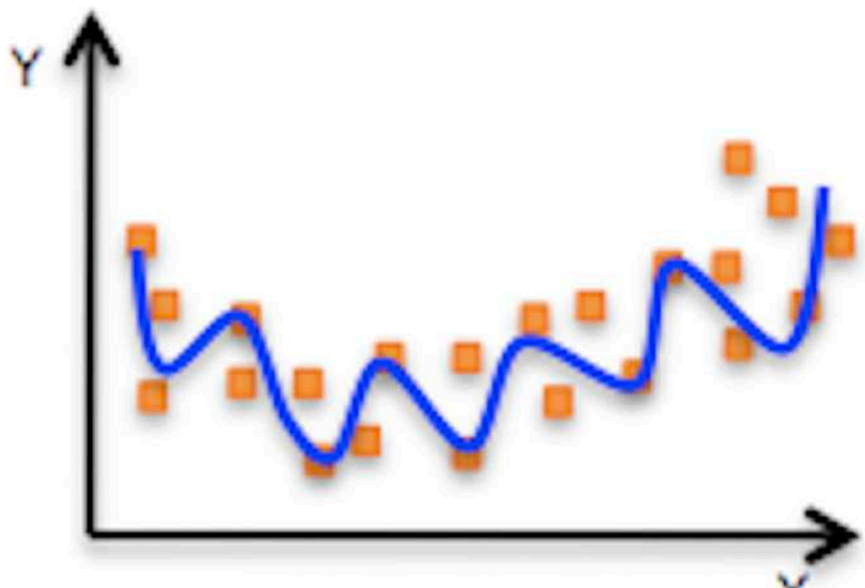
Stronger Foundations



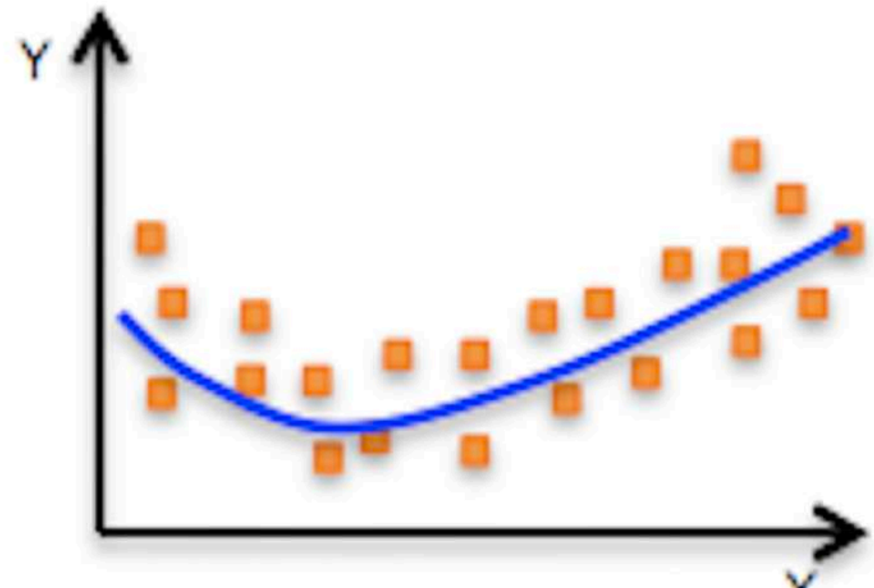
Motivation: overparameterization without overfitting

Mystery:

# of parameters >> # of training data

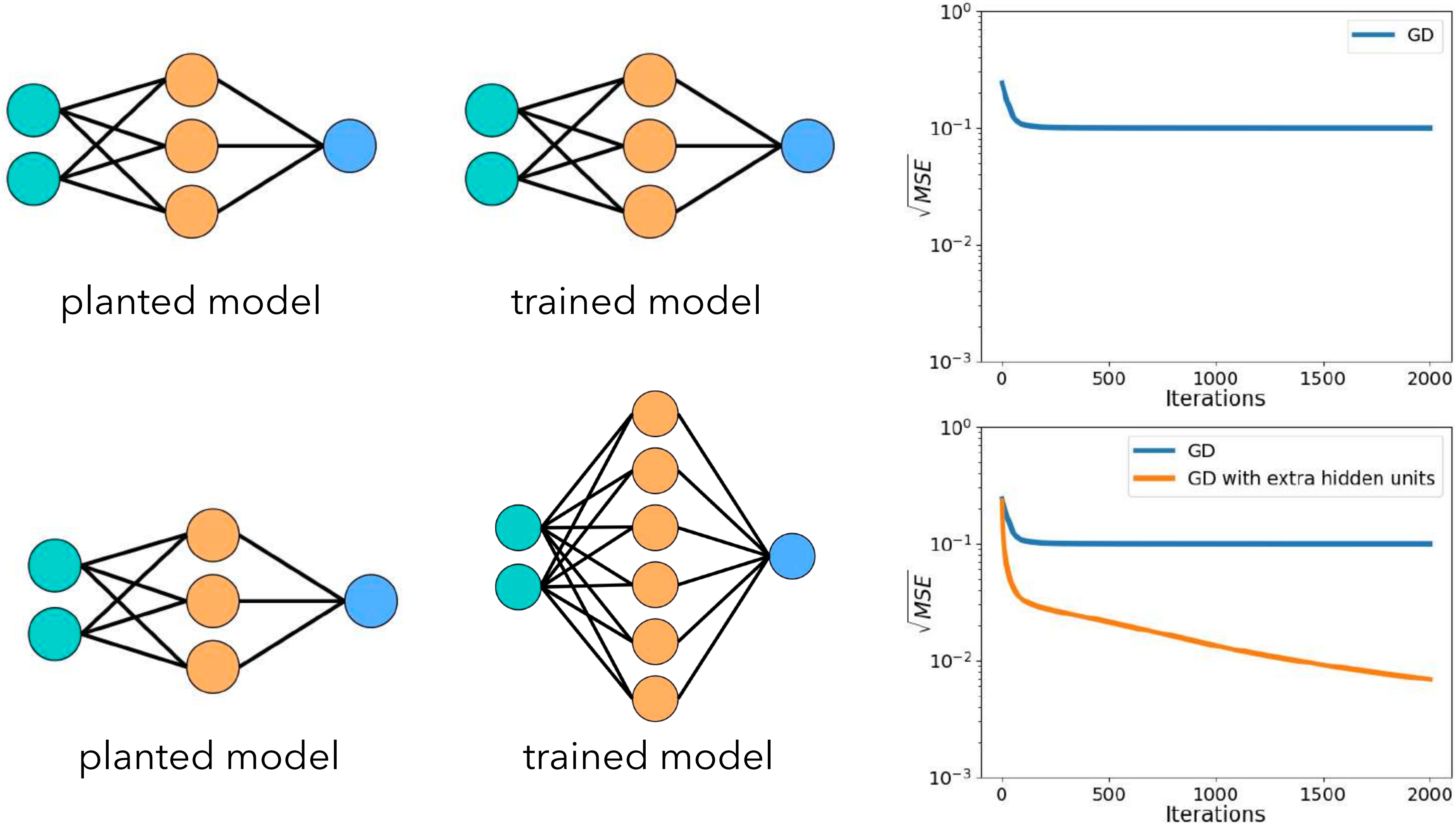


overfitting



just right!

Mystery I: Optimization

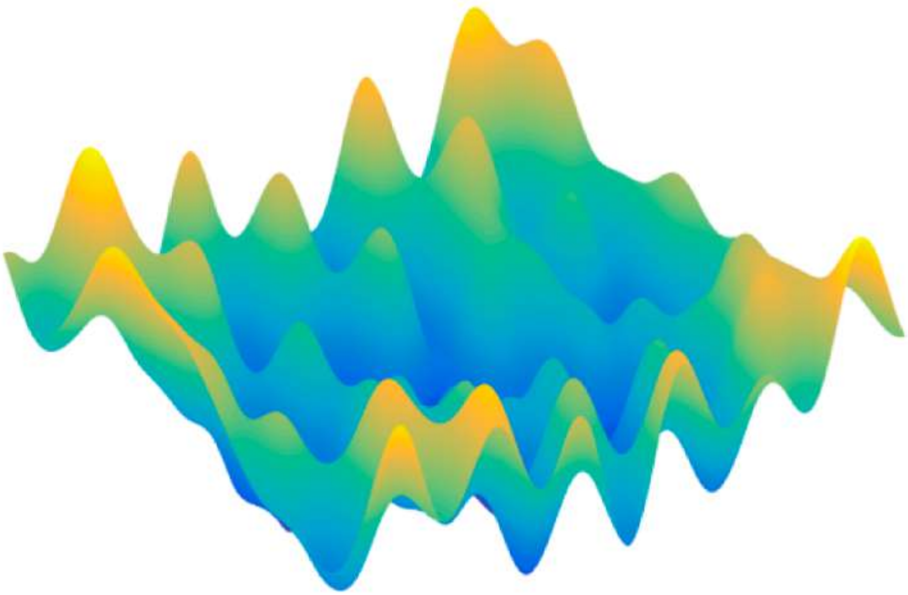


Challenge:

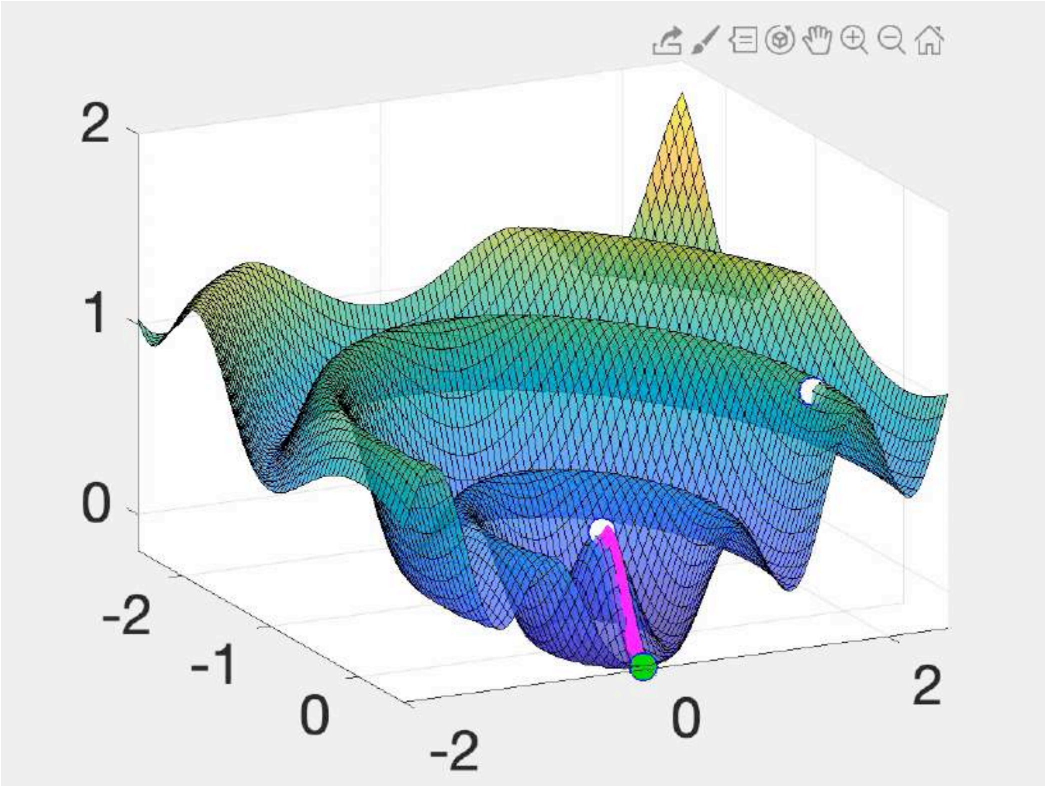
How to establish global convergence of gradient descent from random init.?

Mystery II: Generalization

Many global optima in the training loss



training loss

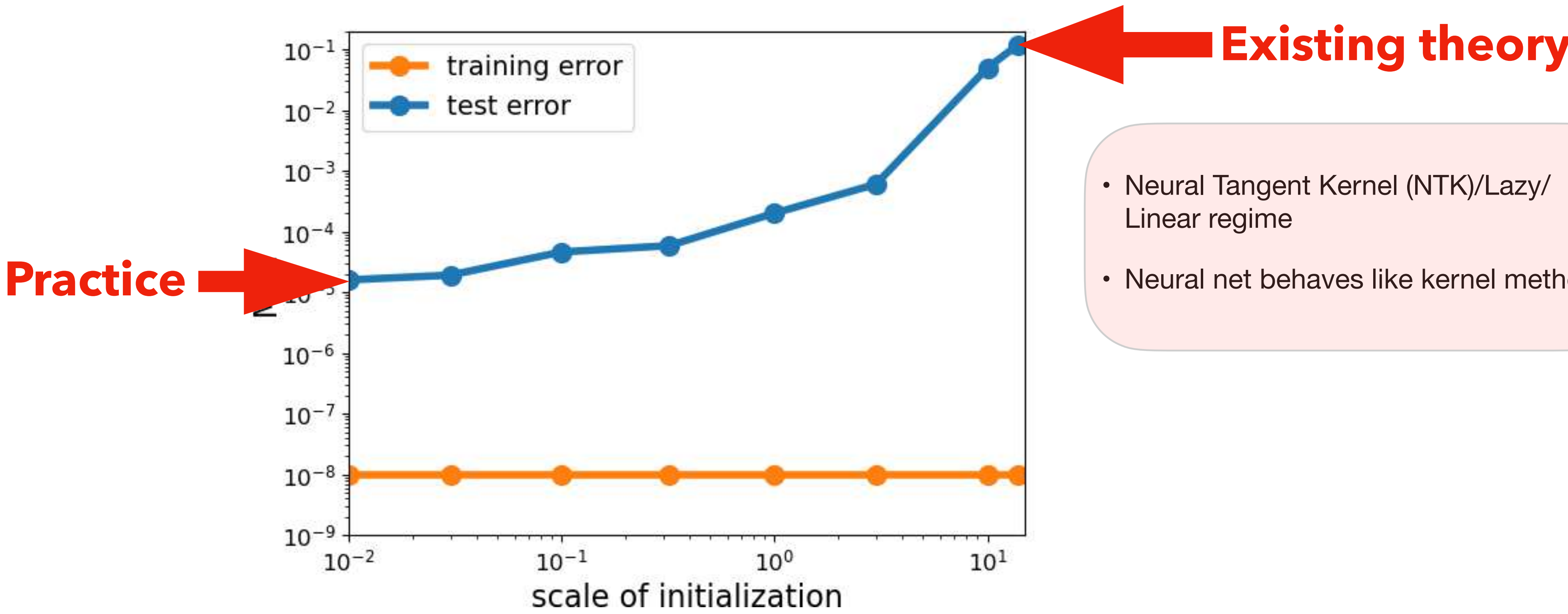


test loss

Can reach different global optima with different init. scale

Mystery II: Generalization (cont.)

Can reach different global optima with different init. scale



- Neural Tangent Kernel (NTK)/Lazy/ Linear regime
- Neural net behaves like kernel methods

Challenge:

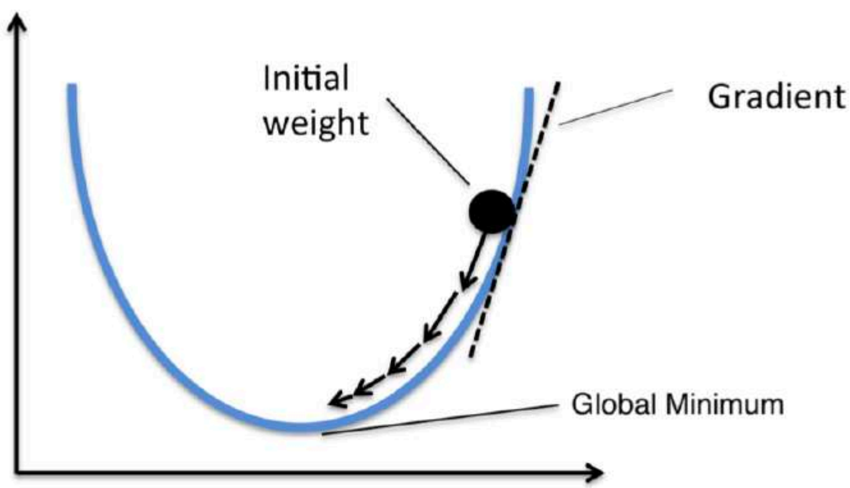
How to establish generalization of vanilla gradient descent from small random initialization?

Prelude: Overparameterized Least Squares

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^p} \mathcal{L}(\boldsymbol{\theta}) := \frac{1}{2}\|\boldsymbol{X}\boldsymbol{\theta}-\boldsymbol{y}\|_{\ell_2}^2 \quad \text{with} \quad \boldsymbol{X}\in\mathbb{R}^{n\times p} \quad \text{and} \quad n\le p.$$

Gradient descent starting from $\boldsymbol{\theta}_0$ has three properties:

- Global convergence
- Converges to a global optimum which is closest to $\boldsymbol{\theta}_0$
- Total gradient path length is relatively short



Overparameterized nonlinear Least Squares

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^p} \mathcal{L}(\boldsymbol{\theta}) := \frac{1}{2} \|f(\boldsymbol{\theta}) - \boldsymbol{y}\|_{\ell_2}^2 ,$$

where

$$\boldsymbol{y} := \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \vdots \\ \boldsymbol{y}_n \end{bmatrix} \in \mathbb{R}^n, \quad f(\boldsymbol{\theta}) := \begin{bmatrix} f(\boldsymbol{x}_1; \boldsymbol{\theta}) \\ f(\boldsymbol{x}_2; \boldsymbol{\theta}) \\ \vdots \\ f(\boldsymbol{x}_n; \boldsymbol{\theta}) \end{bmatrix} \in \mathbb{R}^n, \quad \text{and} \quad n \leq p.$$

Gradient descent: start from some initial parameter $\boldsymbol{\theta}_0$ and run

$$\boldsymbol{\theta}_{\tau+1} = \boldsymbol{\theta}_\tau - \eta_\tau \nabla \mathcal{L}(\boldsymbol{\theta}_\tau),$$

$$\nabla \mathcal{L}(\boldsymbol{\theta}) = \mathcal{J}(\boldsymbol{\theta})^T (f(\boldsymbol{\theta}) - \boldsymbol{y}).$$

Here, $\mathcal{J}(\boldsymbol{\theta}) \in \mathbb{R}^{n \times p}$ is the Jacobian matrix with entries $\mathcal{J}_{ij} = \frac{\partial f(\boldsymbol{x}_i, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}_j}$.

Overparameterized nonlinear Least Squares

Lemma

Under some technical assumptions which hold when

- *network is sufficiently wide*
- *initialization is sufficiently large*

Then along the trajectory of gradient descent

$$f(\boldsymbol{\theta}_\tau) \approx f(\boldsymbol{\theta}_0) + \mathcal{J}(\boldsymbol{\theta}_0)(\boldsymbol{\theta} - \boldsymbol{\theta}_0)$$

Then along the trajectory of gradient descent

$$f(\boldsymbol{\theta}_\tau) \approx f(\boldsymbol{\theta}_0) + \mathcal{J}(\boldsymbol{\theta}_0)(\boldsymbol{\theta} - \boldsymbol{\theta}_0)$$

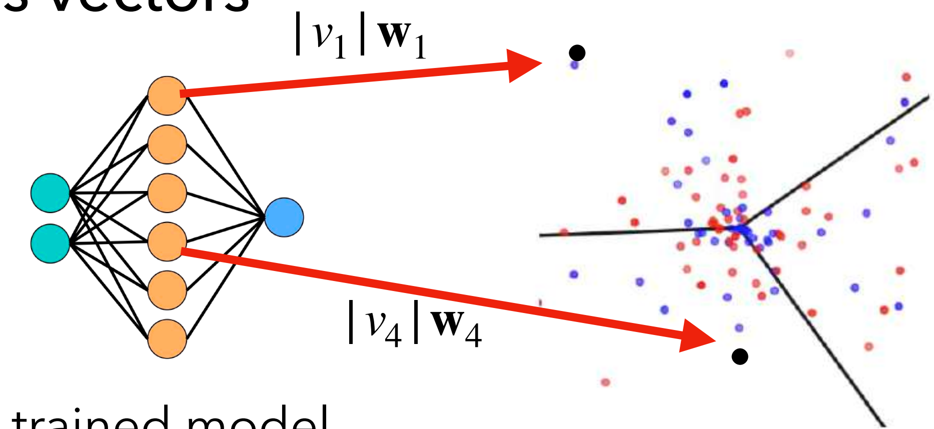
Historical notes

- First usage of linearization principle (?) [Soltanolkotabi, Javanmard, Lee 2017]
- popularized by [Jacot et. al. 2018], [Du et. al. 2019], [Oymak and Soltanolkotabi 2019], [Arora et. al. 2019] and many others

Lazy vs. non-lazy training

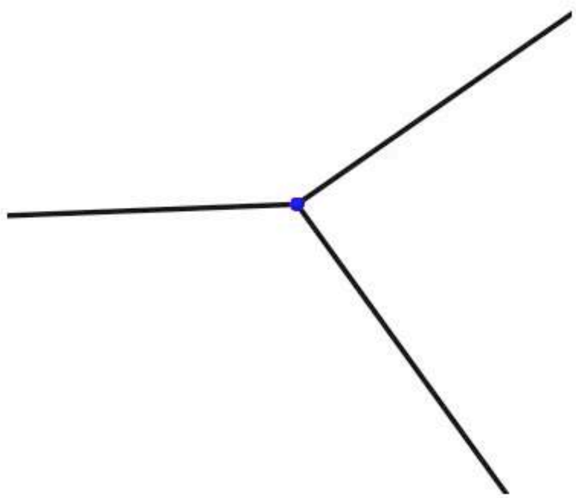
Embed hidden nodes as vectors

$$\mathbf{x} \mapsto \sum_{\ell=1}^{m} v_\ell ReLU\left(\mathbf{w}_\ell^T \mathbf{x}\right)$$

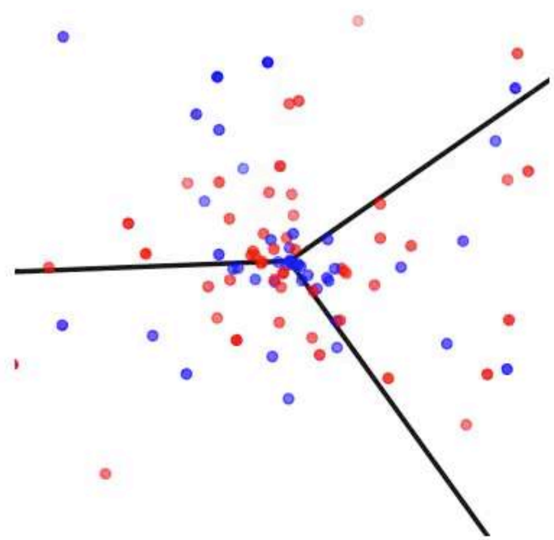


trained model

non-Lazy



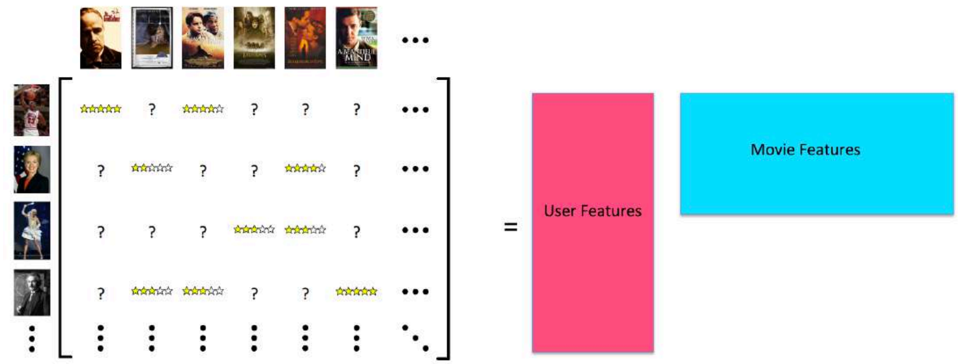
Lazy



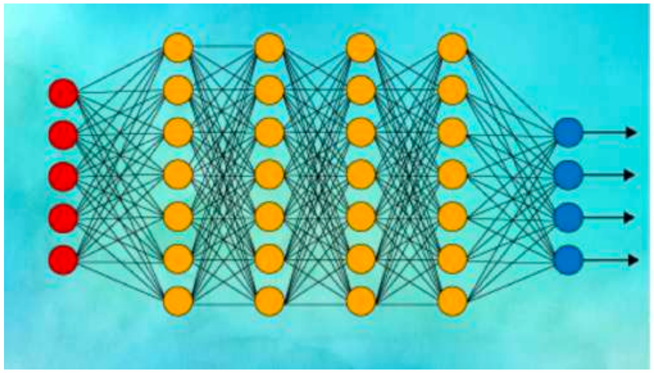
existing theory does not apply in practical regimes…

Learning beyond the lazy regime

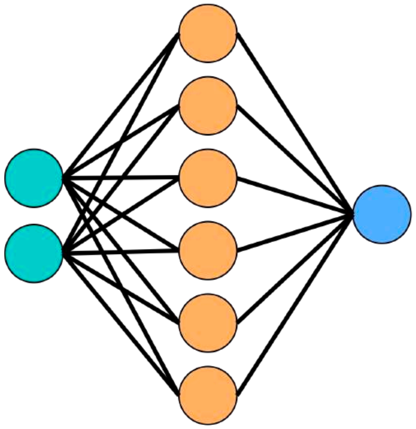
- Low-rank reconstruction



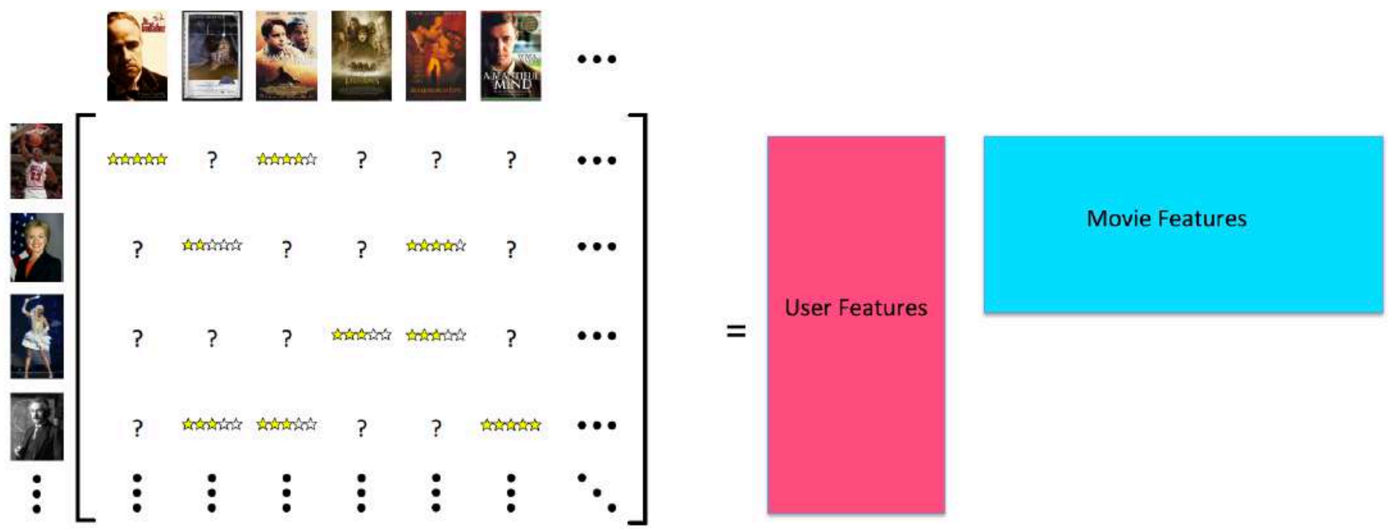
- Deep linear networks



- One-hidden layer networks



Part I: Low-rank reconstruction



Collaborator:

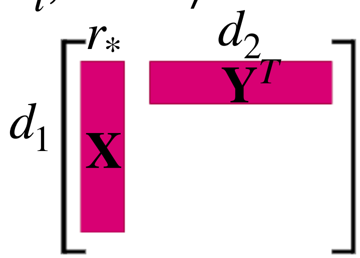
Dominik Stoeger

Changzhi Xie

Low-rank reconstruction

- Measurement model:

$$y_i = \langle \mathbf{A}_i, \mathbf{X}\mathbf{Y}^T \rangle \quad i = 1,2,\ldots,n \quad \Leftrightarrow \quad \mathbf{y} = \mathscr{A}\left(\mathbf{X}\mathbf{Y}^T\right)$$

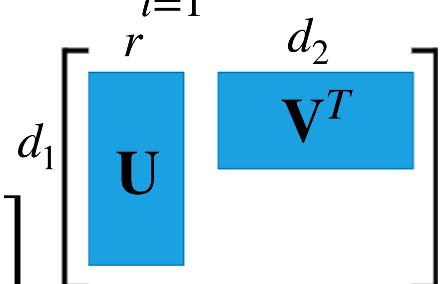


with signal $\mathbf{X} \in \mathbb{R}^{d_1 \times r_*}$ & $\mathbf{Y} \in \mathbb{R}^{d_2 \times r_*}$ and measurement matrices $\mathbf{A}_i \in \mathbb{R}^{d_1 \times d_2}$

- Optimization formulation:

$$\min_{\mathbf{U} \in \mathbb{R}^{d_1 \times r} \& \mathbf{V} \in \mathbb{R}^{d_2 \times r}} \mathscr{L}(\mathbf{U}, \mathbf{V}) := \min_{\mathbf{U} \in \mathbb{R}^{d_1 \times r} \& \mathbf{V} \in \mathbb{R}^{d_2 \times r}} \frac{1}{4} \sum_{i=1}^{n} \left(y_i - \langle \mathbf{A}_i, \mathbf{U}\mathbf{V}^T \rangle\right)^2$$

$r \geq r_*$



- Algorithm: $\begin{bmatrix} \mathbf{U}_{t+1} \\ \mathbf{V}_{t+1} \end{bmatrix} = \begin{bmatrix} \mathbf{U}_t - \mu \nabla_{\mathbf{U}} \mathscr{L}(\mathbf{U}_t, \mathbf{V}_t) \\ \mathbf{V}_t - \mu \nabla_{\mathbf{V}} \mathscr{L}(\mathbf{U}_t, \mathbf{V}_t) \end{bmatrix}$

$\begin{bmatrix} \mathbf{U}_0 \\ \mathbf{V}_0 \end{bmatrix} = \alpha \begin{bmatrix} \mathbf{U} \\ \mathbf{V} \end{bmatrix}$ random init. matrix

Challenge I: Nonconvexity

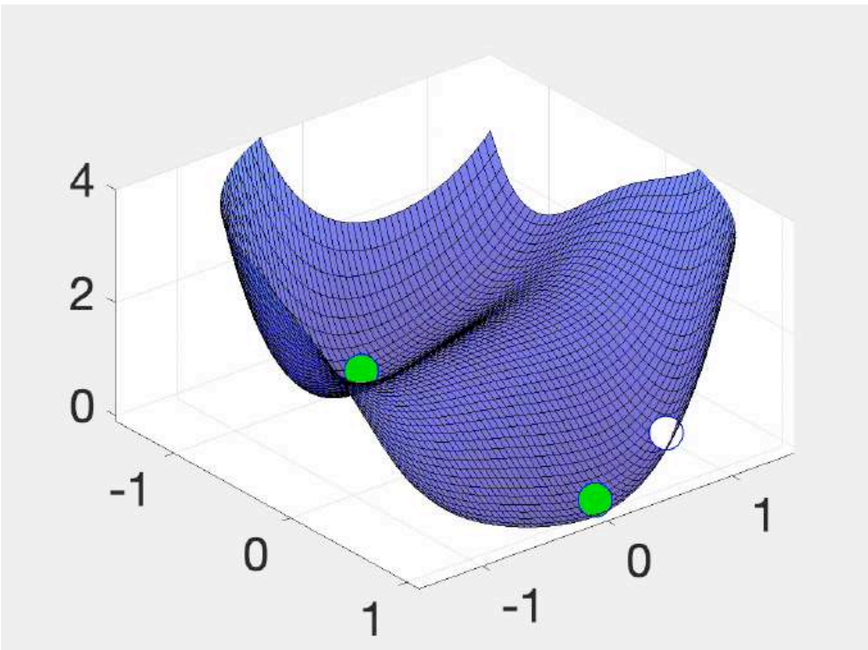
- Spectral init.+local convergence

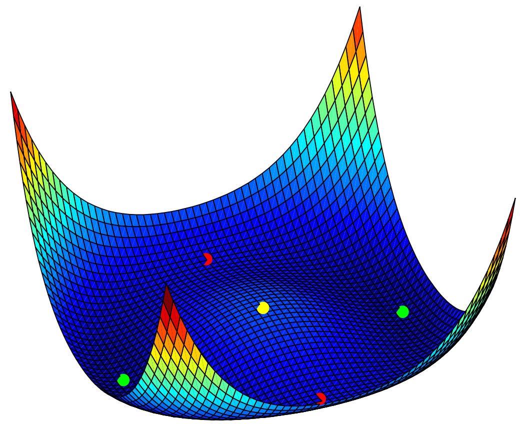
  Wirtinger Flow, Procrustes Flow, etc. by us

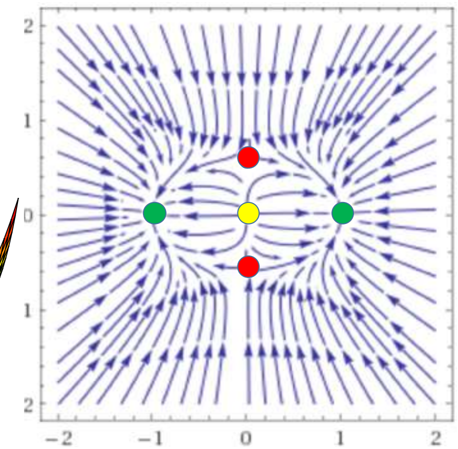
  JUH: Rene, ,…

- Landscape analysis

  [Sun et. al.]), [Ge et. al.], [Bhojanapalli et. al.], …







Challenges:

How to establish *global convergence* *of vanilla gradient descent from small random initialization?*

Challenge II: Generalization

Interested in the overparameterized regime

$$r(d_1 + d_2) \geq n \gtrsim r_*(d_1 + d_2)$$

#params in model

training data

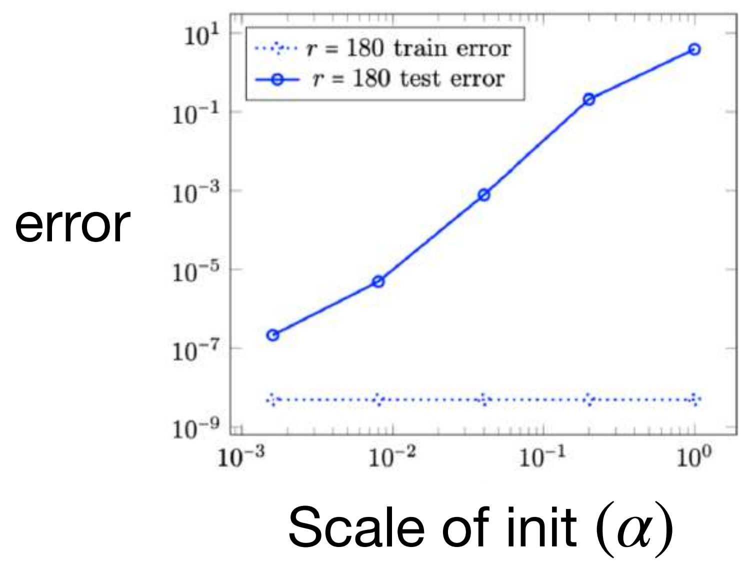
true

With large initialization global convergence occurs as soon as $rd \gtrsim n$ [Oymak & S. '19]

Many global optima

- Small training loss $\mathscr{L}(\mathbf{U}, \mathbf{V}) \approx 0$
- Test error $\|\mathbf{U}\mathbf{V}^T - \mathbf{X}\mathbf{Y}^T\|_F$ potentially large

Example $r_* = 5\ \&\ n = 5r_*d$

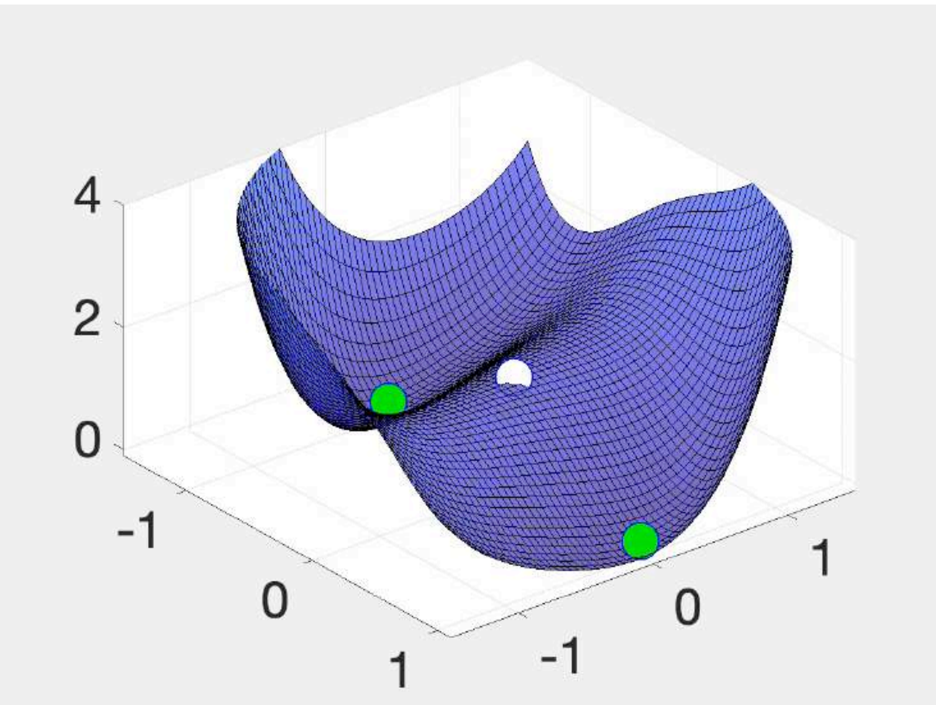


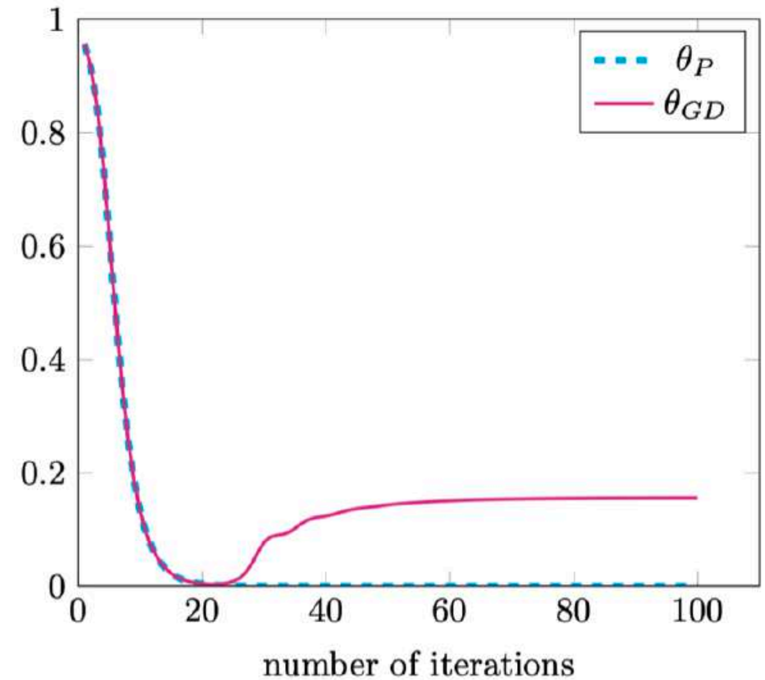
Challenge:

How to establish generalization of vanilla gradient descent from small random initialization?

Key idea: implicit spectral bias of GD

GD + overparameterization = power method on spectral initialization





gradient descent

power method on spectral matrix

θ_{GD} & θ_P angle with top eigen directions of spectral init.

Our result

For simplicity, assume $\kappa := \dfrac{\|\mathbf{X}\mathbf{Y}^T\|}{\sigma_{r_*}(\mathbf{X}\mathbf{Y}^T)} \asymp 1$ and Gaussian mapping $\mathbf{A}_i$

Theorem (Xie, Stoeger & Soltanolkotabi '22)

Assume

- $r \geq r_*$
- $n \gtrsim r_\star^2(d_1 + d_2)$.
- *small random init*
  - $\begin{bmatrix} \boldsymbol{U}_0 \\ \boldsymbol{V}_0 \end{bmatrix} := \alpha \begin{bmatrix} \boldsymbol{U} \\ \boldsymbol{V} \end{bmatrix}$ *with* $\boldsymbol{U} \in \mathbb{R}^{d_1 \times r} \& \boldsymbol{V} \in \mathbb{R}^{d_2 \times r}$ *i.i.d.* $\mathcal{N}(0,1)$ *entries*
  - $\alpha \leq \ldots$

Then, w.h.p., after $T \asymp \ldots$ *iterations*

$$\frac{\left\|\boldsymbol{U}_T\boldsymbol{V}_T^T - \boldsymbol{X}\boldsymbol{Y}^T\right\|_F}{\|\boldsymbol{X}\boldsymbol{Y}^T\|_F} \lesssim \textit{poly}\,(d_1 + d_2, r_\star, r)\, \alpha^{21/16}$$

Some comments

- Gaussian assumption $\mapsto$ Restricted Isometry Property of order $2r_* + 1$
- Case $r = r_*$ first deterministic result for GD with random init.
  - Random results based on leave-one-out [Chen-Chi-Ma 2019]
- Special case $r = d$ by [Li et. al. 18] proving conjecture of [Gunasekar et. al.]
  - Sample size goes to infinity as $\alpha \to 0$
  - many other technical benefits

Proof sketch

Reduction to symmetric

Symmetrization I

- Symmetrization operation $\quad \mathrm{Sym}(\boldsymbol{A}) := \begin{bmatrix} \mathbf{0}_{n_1 \times n_1} & \boldsymbol{A} \\ \boldsymbol{A}^\top & \mathbf{0}_{n_2 \times n_2} \end{bmatrix}.$
- Symmetrize measurements $\quad \mathcal{B}(\boldsymbol{X})_k := \langle \boldsymbol{B}_k, \boldsymbol{X} \rangle, \qquad \boldsymbol{B}_k := \mathrm{Sym}(\boldsymbol{A}_k)$
- Lift variables $\quad \boldsymbol{W} := \begin{bmatrix} \boldsymbol{U} \\ \boldsymbol{V} \end{bmatrix}, \quad \boldsymbol{W}_\tau := \begin{bmatrix} \boldsymbol{U}_\tau \\ \boldsymbol{V}_\tau \end{bmatrix}, \quad \boldsymbol{Z} := \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \end{bmatrix}, \quad \text{and} \quad \widetilde{\boldsymbol{Z}} := \begin{bmatrix} \boldsymbol{X} \\ -\boldsymbol{Y} \end{bmatrix}$
- Loss reformulated as

$$
\begin{aligned}
\mathscr{L}(\mathbf{W}) &= \frac{1}{2}\|\mathscr{A}(UV^T) - \mathscr{A}(\mathbf{X}\mathbf{Y}^T)\|_{\ell_2}^2 \\
&= \frac{1}{4}\|\mathscr{B}(sym(UV^T)) - \mathscr{B}(sym(\mathbf{X}\mathbf{Y}^T))\|_{\ell_2}^2 \\
&= \frac{1}{4}\|\mathscr{B}(\mathbf{W}\mathbf{W}^T) - \mathscr{B}(\mathbf{Z}\mathbf{Z}^T) - \left(\mathscr{B}(\tilde{\mathbf{W}}\tilde{\mathbf{W}}^T) - \mathscr{B}(\tilde{\mathbf{Z}}\tilde{\mathbf{Z}}^T\right) \|_{\ell_2}^2
\end{aligned}
$$

Symmetrization II

- When $\mathbf{U}^T\mathbf{U} \approx \mathbf{V}^T\mathbf{V} \;\Rightarrow\; \mathbf{W}^T\tilde{\mathbf{W}} \approx \mathbf{0}$
- As if we have

$$\mathscr{L}(\mathbf{W}) = \frac{1}{4}\|\mathscr{B}(\mathbf{W}\mathbf{W}^T) - \mathscr{B}(\mathbf{Z}\mathbf{Z}^T)\|_{\ell_2}^2 \quad \& \quad \mathscr{L}(\tilde{\mathbf{W}}) = \frac{1}{4}\|\mathscr{B}(\tilde{\mathbf{W}}\tilde{\mathbf{W}}^T) - \mathscr{B}(\tilde{\mathbf{Z}}\tilde{\mathbf{Z}}^T)\|_{\ell_2}^2$$

- How to show $\mathbf{U}_\tau^T\mathbf{U}_\tau \approx \mathbf{V}_\tau^T\mathbf{V}_\tau$????
- We show

$$\|\mathbf{U}_\tau^T\mathbf{U}_\tau - \mathbf{V}_\tau^T\mathbf{V}_\tau\|_F \leq c\|\mathbf{U}_0^T\mathbf{U}_0 - \mathbf{V}_0^T\mathbf{V}_0\|_F$$

Small at initialization

Proof of $\|\mathbf{U}_\tau^T\mathbf{U}_\tau - \mathbf{V}_\tau^T\mathbf{V}_\tau\|_F \le c\|\mathbf{U}_0^T\mathbf{U}_0 - \mathbf{V}_0^T\mathbf{V}_0\|_F$

$B_t = V_t^T V_t - W_t^T W_t.$

- Lemma: $\|B_t\|_F \le \|B_0\|_F + 2\mu(\mathcal{L}(V_0, W_0) - \mathcal{L}(V_t, W_t))$
- Key idea: $V_t^T \nabla_V \mathcal{L}(V_t, W_t) = \nabla_W \mathcal{L}(V_t, W_t)^T W_t$
- Proof of Lemma:

$$\begin{aligned}
B_{t+1} =& (V_t - \mu\nabla_V \mathcal{L}(V_t, W_t))^T (V_t - \mu\nabla_V \mathcal{L}(V_t, W_t)) \\
& - (W_t - \mu\nabla_W \mathcal{L}(V_t, W_t))^T (W_t - \mu\nabla_W \mathcal{L}(V_t, W_t)) \\
=& V_t^T V_t + \mu^2 \nabla_V \mathcal{L}(V_t, W_t)^T \nabla_V \mathcal{L}(V_t, W_t) \\
& - W_t^T W_t - \mu^2 \nabla_W \mathcal{L}(V_t, W_t)^T \nabla_W \mathcal{L}(V_t, W_t) \\
=& B_t + \mu^2 (\nabla_V \mathcal{L}(V_t, W_t)^T \nabla_V \mathcal{L}(V_t, W_t) - \nabla_W \mathcal{L}(V_t, W_t)^T \nabla_W \mathcal{L}(V_t, W_t)).
\end{aligned}$$

- Final step

$$\begin{aligned}
\|B_{t+1} - B_t\|_F \le& \mu^2 (\|\nabla_V \mathcal{L}(V_t, W_t)\|_F^2 + \|\nabla_W \mathcal{L}(V_t, W_t)\|_F^2) \\
\le& 2\mu(\mathcal{L}(V_t, W_t) - \mathcal{L}(V_{t+1}, W_{t+1})).
\end{aligned}$$

Symmetric case

$$\mathbf{U} = \mathbf{V}$$

How does small initialization help?

- Look at the first gradient:

$$
\begin{aligned}
-\nabla \mathscr{L}(\mathbf{U}_0) &= \mathscr{A}^*\mathscr{A}\left(\mathbf{X}\mathbf{X}^T - \mathbf{U}_0\mathbf{U}_0^T\right)\mathbf{U}_0 \\
&\approx \mathscr{A}^*\mathscr{A}\left(\mathbf{X}\mathbf{X}^T\right)\mathbf{U}_0 := \mathbf{Z}\mathbf{U}_0
\end{aligned}
$$

- Hence

$$
\mathbf{U}_1 = \mathbf{U}_0 - \mu \nabla \mathscr{L}\left(\mathbf{U}_0\right) \approx \left(\mathbf{I} + \mu \mathbf{Z}\right)\mathbf{U}_0
$$

Role of randomness+overparameterization

- Hence, for small t

$$\mathbf{U}_t \approx \left(\mathbf{I} + \mu\mathbf{Z}\right)^t \mathbf{U}_0 =: \tilde{\mathbf{U}}_t$$

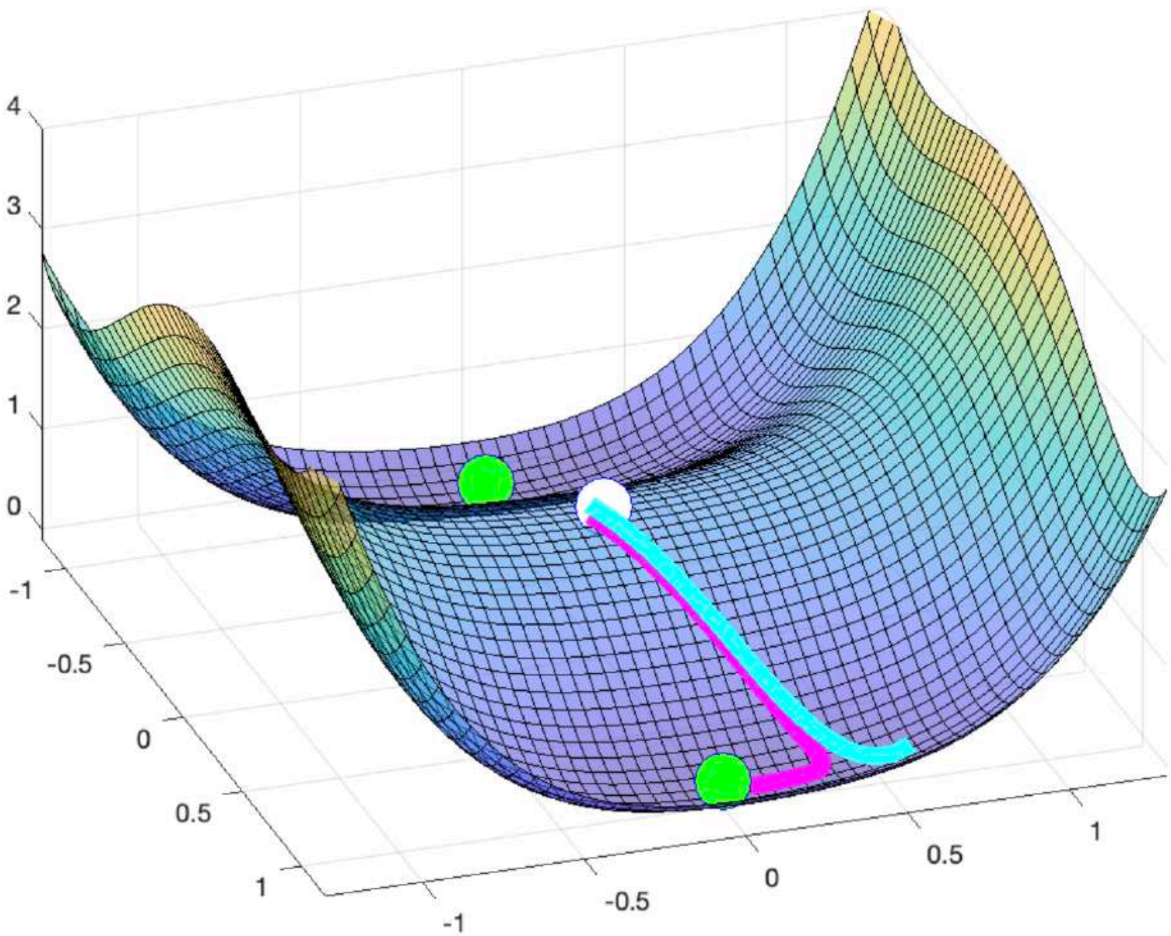
- Up to normalization, this is the <u>power method!</u>

- Since $\mathbf{A}_i$ are Gaussian, w.h.p.

$$\mathbf{Z} = \mathscr{A}^*\mathscr{A}\left(\mathbf{X}\mathbf{X}^T\right) = \frac{1}{n}\sum_{i=1}^{n}\langle \mathbf{A}_i, \mathbf{X}\mathbf{X}^T\rangle \mathbf{A}_i \approx \mathbf{X}\mathbf{X}^T$$

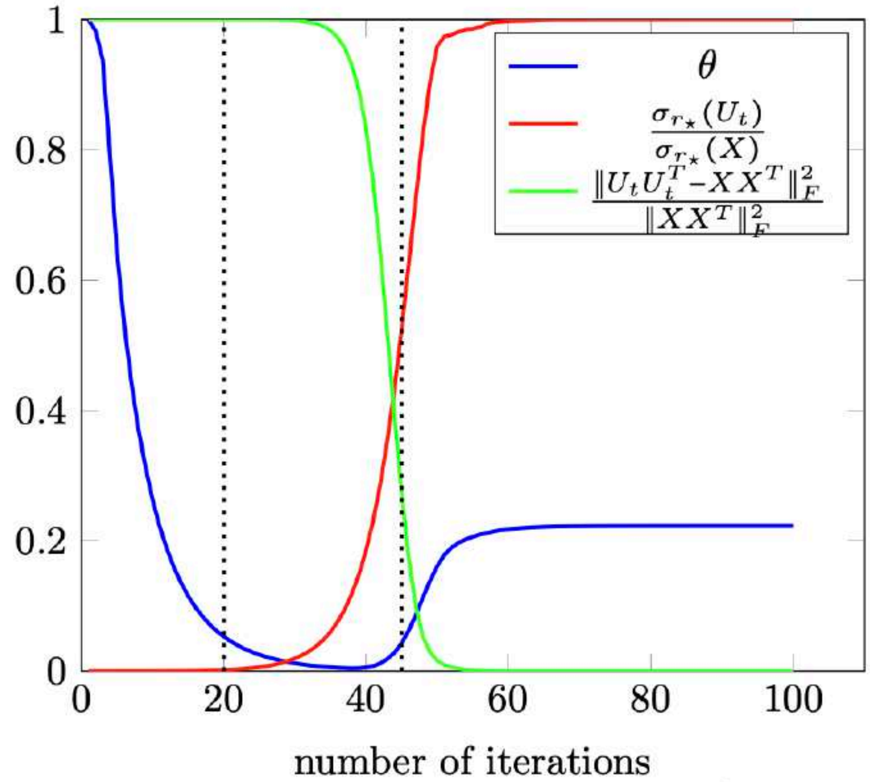
Is this really true?

Set $r = r_* = 1, d = 2, n = 6$



- $\boldsymbol{U}_t$
- $\widetilde{\boldsymbol{U}}_t = (\boldsymbol{I} + \mu \boldsymbol{Z})^t \boldsymbol{U}_0$

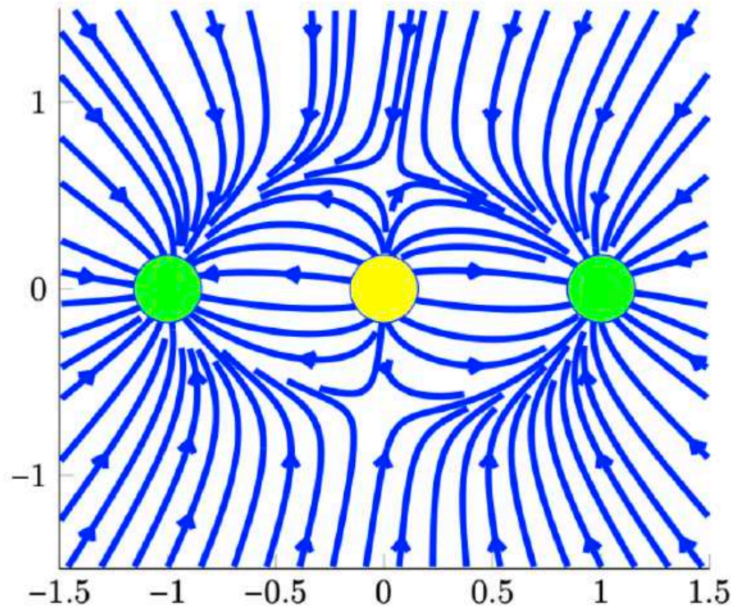
Convergence phases



$$\frac{1}{\mu\sigma_{\min}(X)^2}\left[\underbrace{\ln\left(2\kappa^2\sqrt{\frac{n}{\min\{r;n\}}}\right)}_{\text{Phase I: spectral/alignment phase}}+\underbrace{\ln\left(\frac{\sigma_{\min}(X)}{\alpha}\right)}_{\text{Phase II: saddle avoidance phase}}\underbrace{+\ln\left(\max\left\{1;\frac{\kappa r_\star}{\min\{r;n\}-r_\star}\right\}\frac{\|X\|}{\alpha}\right)}_{\text{Phase III: local refinement phase}}\right].$$

- Phase I: spectral phase
- Phase II: saddle avoidance phase
- Phase III: refinement phase

Saddle avoidance and local convergence phase



Decompose

$$\boldsymbol{U}_t = \underbrace{\boldsymbol{U}_t \boldsymbol{W}_t {\boldsymbol{W}_t}^T}_{\text{signal term}} + \underbrace{\boldsymbol{U}_t \left(\boldsymbol{I} - \boldsymbol{W}_t {\boldsymbol{W}_t}^T \right)}_{\text{noise term}}$$

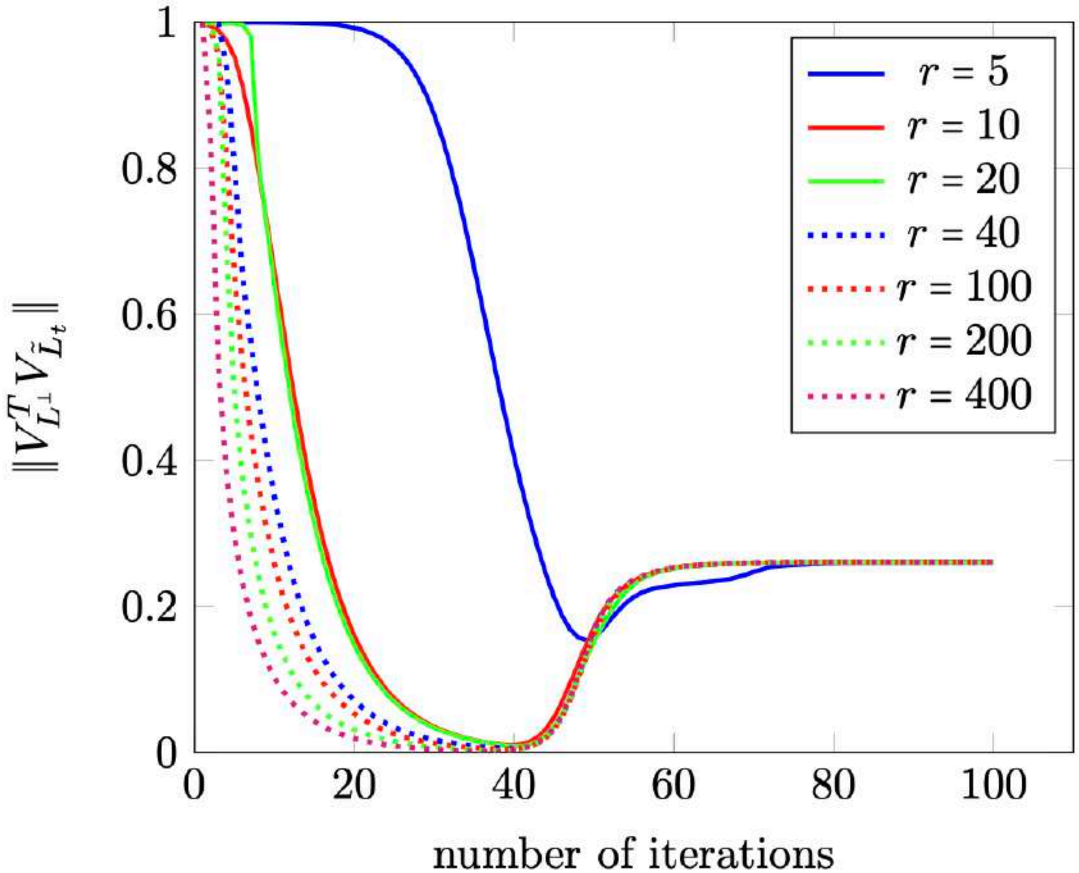
$\boldsymbol{W}_t \in \mathbb{R}^{n \times r_\star}$ properly chosen isometric embedding

- saddle avoidance: minimum eigenvalue of $\boldsymbol{U}_t \boldsymbol{W}_t$ grows
- local convergence: signal term converges to $\boldsymbol{X}$, while the noise term stays small (scaling with α)

Insights and predictions

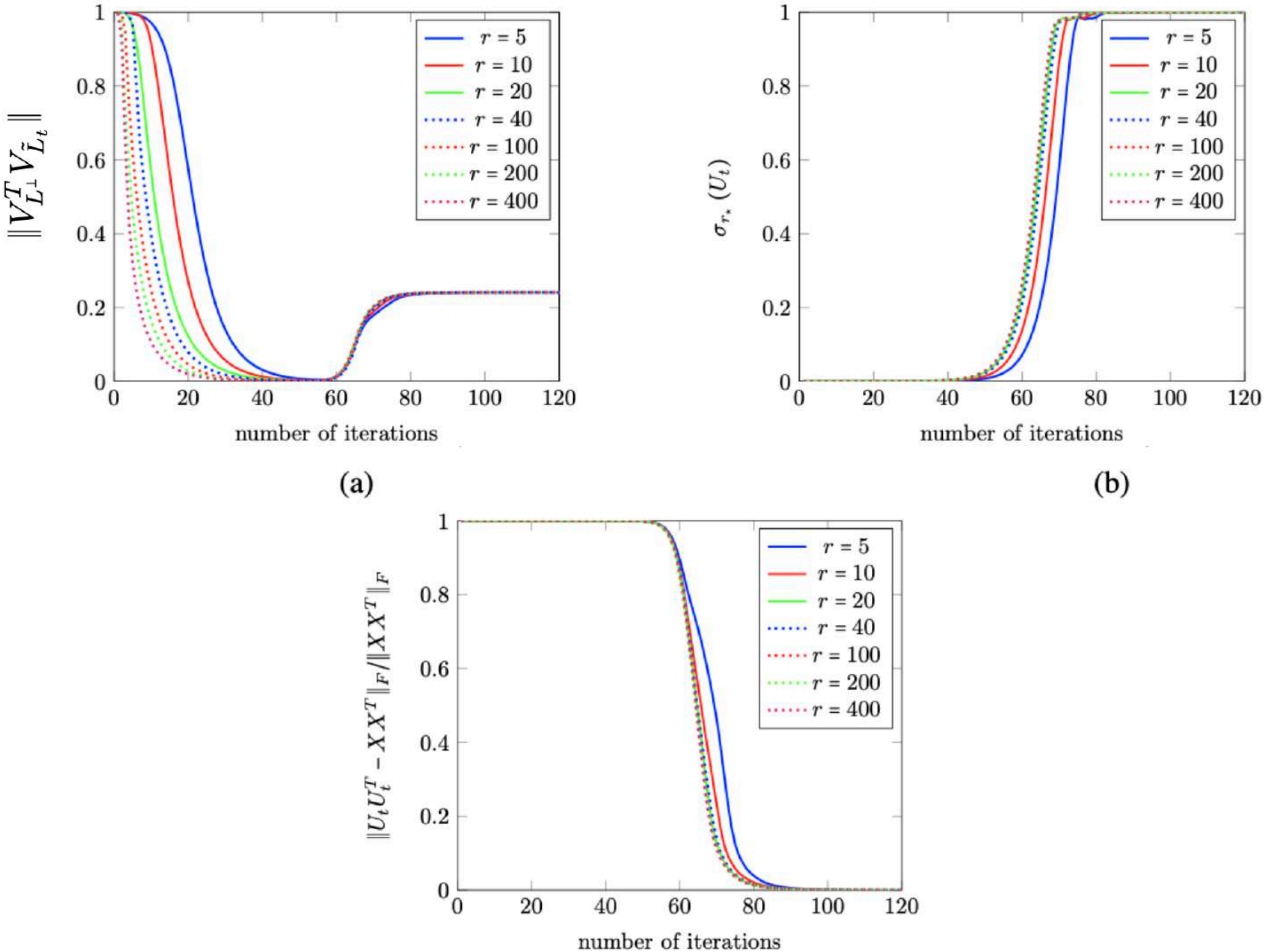
How does more overparameterization help?

$n = 200$, $r_\star = 5$, $m = 10nr_\star$

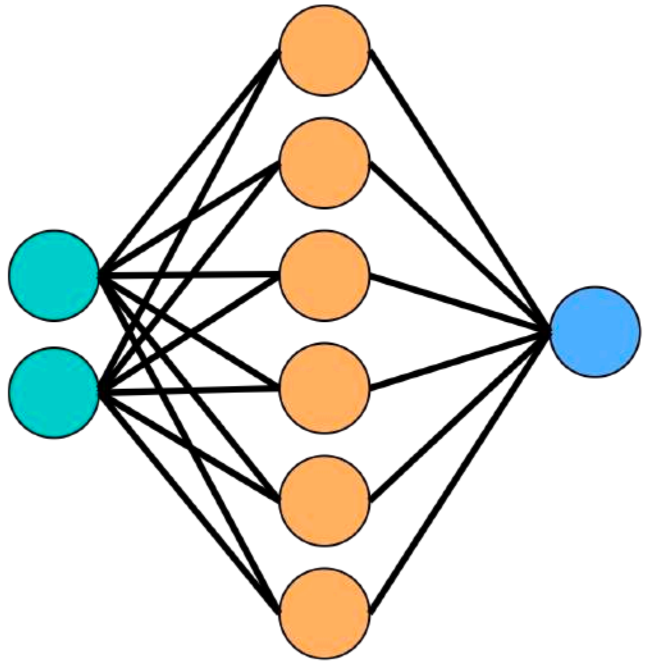


Prediction by our theory: Spectral phase needs $t_\star \asymp \frac{1}{\mu} \ln\left(\frac{2n}{r}\right)$ iterations $(\boldsymbol{U}_t \approx (\boldsymbol{I} + \mu \boldsymbol{Z})^t \boldsymbol{U}_0)$

Overparameterization does not affect other phases



Part II: one-hidden layer neural nets



Collaborators:

Alex Damian

Jason Lee

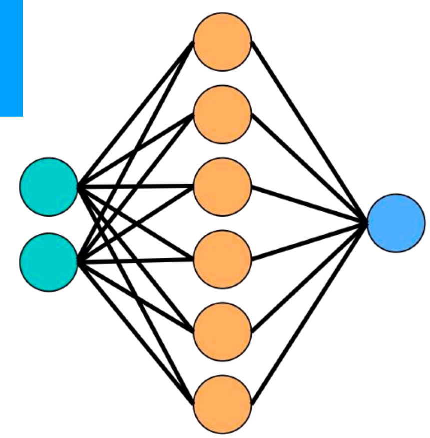
Learning polynomials with neural nets

- Inputs: $\mathbf{x}_i \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}\right)$

- Labels: $y_i = g\left(\mathbf{U}\mathbf{x}_i\right) \qquad i = 1,2,\ldots,n$

$g : \mathbb{R}^r \mapsto \mathbb{R}$
poly of degree p

$$\mathbf{U} = r \underbrace{\overset{d}{}}_{} \in \mathbf{R}^{r\times d},\ r << d$$



- Model: $\mathbf{x} \mapsto f_{\mathbf{v},\mathbf{W}}(\mathbf{x}) = \mathbf{v}^T ReLU\left(\mathbf{W}\mathbf{x}\right)$

- Loss: $\mathscr{L}\left(\mathbf{v}, \mathbf{W}\right) := \frac{1}{n}\sum_{i=1}^{n}\left(y_i - f_{\mathbf{v},\mathbf{W}}(\mathbf{x}_i)\right)^2$

- Algorithm: GD from small init

Our Result

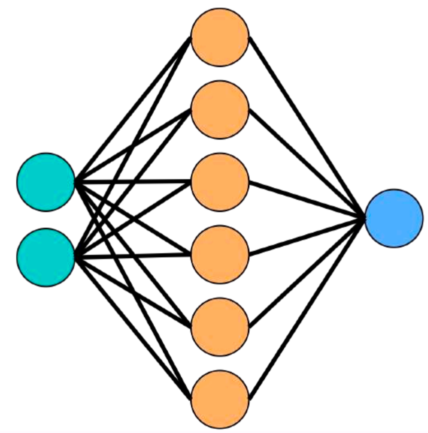
Data

$$y_i = g\left(\mathbf{U}\mathbf{x}_i\right)$$

$g : \mathbb{R}^r \mapsto \mathbb{R}$
poly of degree p

$\mathbf{U} \in \mathbb{R}^{r \times d} \quad r << d$

Model $\mathbf{x} \mapsto f_{\mathbf{v},\mathbf{b},\mathbf{W}}(\mathbf{x}) := \mathbf{v}^T ReLU\left(\mathbf{W}\mathbf{x} + \mathbf{b}\right)$



Theorem (Ghorbani et. al. '20)

In the lazy/NTK regime need at least $\gtrsim d^p$ samples

Theorem (Damian, Lee & Soltanolkotabi '22)

- *Hidden unites $\gtrsim r^p$*
- *Run GD from small random init*

Then, w.h.p., after $T \asymp \ldots$ iterations

$$\mathbb{E}_{\boldsymbol{x},y}\left|f_{\boldsymbol{v}_T,\boldsymbol{b}_T,\boldsymbol{W}_T}(\boldsymbol{x}) - y\right| \lesssim \sqrt{\frac{d^2 + r^{4p+1}}{n}}$$

need $n \gtrsim d^2 + r^{4p+1}$ vs. $n \gtrsim d^p$ for NTK/lazy regime

Transfer Learning Setup

Source Data (n samples)

Target Data (N samples)

$\mathbf{U} \in \mathbb{R}^{r \times d} \quad r << d$

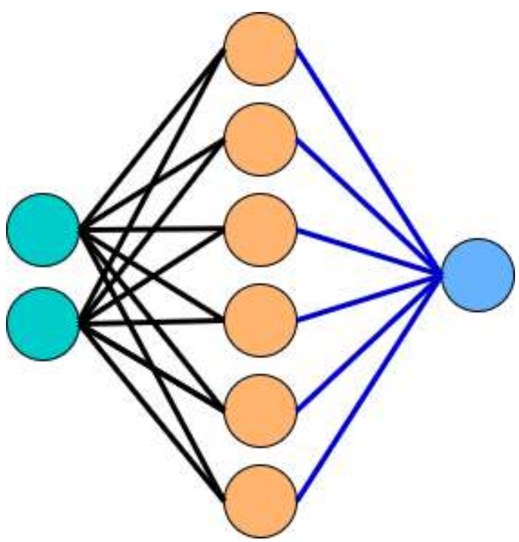
$$y_i = g_{\mathcal{S}}\left(\mathbf{U}\mathbf{x}_i\right)$$

$$y_i = g_{\mathcal{T}}\left(\mathbf{U}\mathbf{x}_i\right)$$

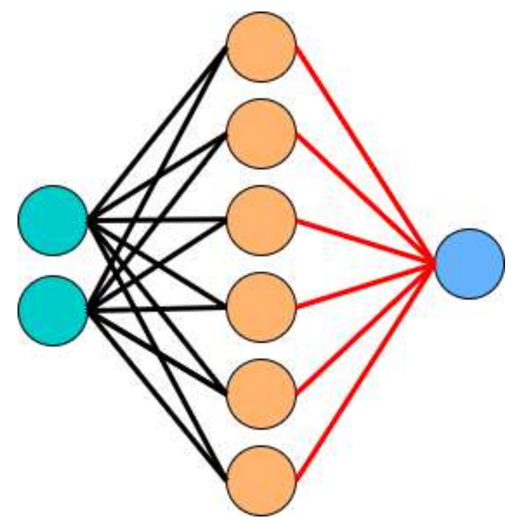
$g_{\mathcal{S}}, g_{\mathcal{T}} : \mathbb{R}^r \mapsto \mathbb{R}$

poly of degree p

train both layers
on source data



Retrain last layer
on target data



Transfer Learning Result

Source Data (n samples)

$$y_i = g_{\mathcal{S}}\left(\mathbf{U}\mathbf{x}_i\right)$$

train both layers
on source data

Target Data (N samples)

$$y_i = g_{\mathcal{T}}\left(\mathbf{U}\mathbf{x}_i\right)$$

Retrain last layer
on target data

Theorem (Damian, Lee & Soltanolkotabi '22)

- *Hidden unites* $\gtrsim r^p$

Then, w.h.p., after $T \asymp \ldots$ *iterations*

$$\mathbb{E}_{\boldsymbol{x},y\sim\mathcal{T}}\left|f_{\boldsymbol{v}_T,\boldsymbol{b}_T,\boldsymbol{W}_T}(\boldsymbol{x}) - y\right| \lesssim \underbrace{\sqrt{\frac{d^2 + r^{4p+1}}{n}}}_{\#\text{ data for learning representation}} + \underbrace{\sqrt{\frac{r^{4p+1}}{N}}}_{\#\text{ data for learning head}}$$

Very brief proof sketch

Consider Hermite polynomials in higher dimensions

$$S_1(x) = \mathbf{x}, \quad S_2(x) = \mathbf{x}\mathbf{x}^T - \mathbf{I}, \quad \ldots$$

We have the series

$$f(\boldsymbol{x}) = \sum_{t=1}^{+\infty} \langle \mathbb{E}\left[f(\boldsymbol{x}) S_t(\boldsymbol{x}) \right], S_t(\boldsymbol{x}) \rangle$$

By Stein

$$= \sum_{t=1}^{+\infty} \langle \mathbb{E}\left[f^{(t)}(\boldsymbol{x}) \right], S_t(\boldsymbol{x}) \rangle$$

Many intricate components

Lemma 1 *Consider a polynomial of degree p given by*

$$g(\boldsymbol{z}) := \sum_{s_j \in \mathbb{N} \cup \{0\}:\ \sum_{j=1}^{r} s_j \leq p} \nu_{s_1,\ldots,s_r} \prod_{j=1}^{r} z_j^{s_j}.$$

and denote $\boldsymbol{\nu}$ as the vector of all of the coefficients $\nu_{s_1,\ldots,s_r}$. Also let $\boldsymbol{U} \in \mathbb{R}^{r \times d}$. Then, as long as

$$n \geq \max\left(cd \frac{2\pi p \left(Cp^3 \beta \log n\right)^p}{\delta^2}, \left(\frac{6\sqrt{d}\left(\sqrt{2C}p^2\right)^p}{\delta} \right)^{\frac{4}{\beta}} \right),$$

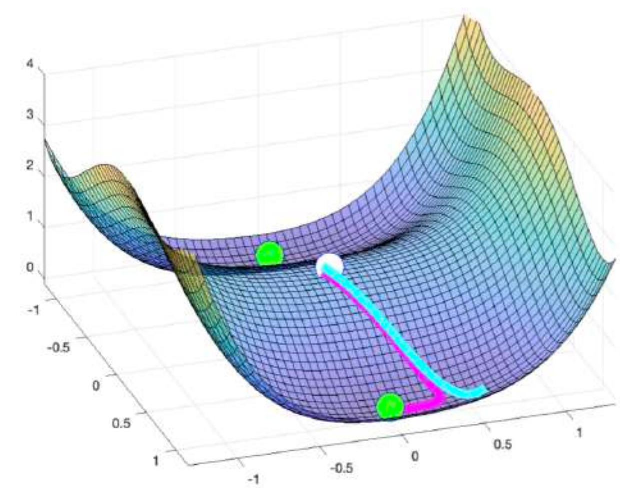
holds for some $\beta \geq 1$ and $\delta > 0$. Then,

$$\left\| \frac{1}{n} \sum_{i=1}^{n} g\left(\boldsymbol{U}\boldsymbol{x}_i\right) \boldsymbol{x}_i \mathbb{1}_{\{\boldsymbol{w}^T \boldsymbol{x}_i + b \geq 0\}} - \mathbb{E}\left[g\left(\boldsymbol{U}\boldsymbol{x}\right) \boldsymbol{x} \mathbb{1}_{\{\boldsymbol{w}^T \boldsymbol{x} + b \geq 0\}} \right] \right\| \leq \delta \sqrt{\mathbb{E}\left[g^2\left(\boldsymbol{U}\boldsymbol{x}\right) \right]}$$

holds with probability at least $1 - 2e^{-cd} - 2n^{-(\beta-1)}$.

Conclusion

- Stronger Theoretical Foundations
  - Go beyond lazy regime
  - Many settings Low rank reconstruction, deep linear networks, one-hidden layers
  - Key idea: implicit spectral bias of GD



Thanks!

Funding acknowledgement

FastNICs

LwLL