Optimization and Generalization for Deep Linear Neural Networks via Trajectories of Gradient Descent

Nadav Cohen

Tel Aviv University

Princeton University, Computer Science Department

Theoretical Deep Learning Course (COS 597B)

6 December 2019
Outline

1. Optimization and Generalization in Deep Learning via Trajectories

2. Case Study: Linear Neural Networks
   - Trajectory Analysis
   - Optimization
   - Generalization

3. Conclusion
Optimization

Fitting training data by minimizing an objective (loss) function
Controlling gap between train and test errors, e.g. by adding regularization term/constraint to objective.
**Theme:** make sure objective is **convex**!
Optimization and Generalization in Deep Learning via Trajectories

Classical Machine Learning

**Theme:** make sure objective is **convex**!

**Optimization**
- Single global minimum, efficiently attainable
- Choice of *algorithm* affects only *speed* of convergence
Optimization and Generalization in Deep Learning via Trajectories

Classical Machine Learning

**Theme:** make sure objective is convex!

**Optimization**
- Single global minimum, efficiently attainable
- Choice of algorithm affects only speed of convergence

**Generalization**
Bias-variance trade-off:

<table>
<thead>
<tr>
<th>regularization</th>
<th>train/test gap</th>
<th>train err</th>
</tr>
</thead>
<tbody>
<tr>
<td>more</td>
<td>↘</td>
<td>↗</td>
</tr>
<tr>
<td>less</td>
<td>↗</td>
<td>↘</td>
</tr>
</tbody>
</table>
Classical Machine Learning

**Theme:** make sure objective is **convex**!

**Optimization**
- Single global minimum, efficiently attainable
- Choice of algorithm affects only speed of convergence

**Generalization**
Bias-variance trade-off:

<table>
<thead>
<tr>
<th>regularization</th>
<th>train/test gap</th>
<th>train err</th>
</tr>
</thead>
<tbody>
<tr>
<td>more</td>
<td>↘</td>
<td>↗</td>
</tr>
<tr>
<td>less</td>
<td>↗</td>
<td>↘</td>
</tr>
</tbody>
</table>
Theme: allow objective to be non-convex
Deep Learning (DL)

Theme: allow objective to be non-convex

Optimization

- Multiple minima, a-priori not efficiently attainable
- Variants of gradient descent (GD) somehow reach global min

Not well understood
Deep Learning (DL)

**Theme:** allow objective to be non-convex

**Optimization**
- Multiple minima, a-priori not efficiently attainable
- Variants of gradient descent (GD) somehow reach global min

**Generalization**
- Some global minima generalize well, others don’t
- With typical data, solution found by GD often generalizes well
- No bias-variance trade-off — regularization implicitly induced by GD
Deep Learning (DL)

**Theme:** allow objective to be non-convex

**Optimization**
- Multiple minima, a-priori not efficiently attainable
- Variants of gradient descent (GD) somehow reach global min

**Generalization**
- Some global minima generalize well, others don’t
- With typical data, solution found by GD often generalizes well
- No bias-variance trade-off — regularization implicitly induced by GD
Perspective

- Language of classical learning theory may be insufficient for DL
Perspective

- Language of classical learning theory may be insufficient for DL
- Need to carefully analyze course of learning, i.e. trajectories of GD!
Perspective

- Language of classical learning theory may be insufficient for DL
- Need to carefully analyze course of learning, i.e. trajectories of GD!

We will demonstrate this for deep linear neural networks
Outline

1. Optimization and Generalization in Deep Learning via Trajectories

2. Case Study: Linear Neural Networks
   - Trajectory Analysis
   - Optimization
   - Generalization

3. Conclusion
Sources

On the Optimization of Deep Networks: Implicit Acceleration by Overparameterization
Arora + C + Hazan
*International Conference on Machine Learning (ICML) 2018*

A Convergence Analysis of Gradient Descent for Deep Linear Neural Networks
Arora + C + Golowich + Hu
*International Conference on Learning Representations (ICLR) 2019*

Implicit Regularization in Deep Matrix Factorization
Arora + C + Hu + Luo
*Conference on Neural Information Processing Systems (NeurIPS) 2019*
Linear neural networks (LNN) are fully-connected neural networks with linear (no) activation
**Linear Neural Networks**

**Linear neural networks** (LNN) are fully-connected neural networks with linear (no) activation

\[ x \rightarrow W_1 \rightarrow W_2 \rightarrow \cdots \rightarrow W_N \rightarrow y = W_N \cdots W_2 W_1 x \]

LNN realize only linear mappings, but are highly non-trivial in terms of optimization and generalization.
**Linear Neural Networks**

**Linear neural networks** (LNN) are fully-connected neural networks with linear (no) activation

\[
x \rightarrow W_1 \rightarrow W_2 \rightarrow \cdots \rightarrow W_N \rightarrow y = W_N \cdots W_2 W_1 x
\]

LNN realize only linear mappings, but are highly non-trivial in terms of optimization and generalization

Studied extensively as surrogate for non-linear neural networks:

- Saxe et al. 2014
- Kawaguchi 2016
- Advani & Saxe 2017
- Hardt & Ma 2017
- Laurent & Brecht 2018
- Gunasekar et al. 2018
- Ji & Telgarsky 2019
- Lampinen & Ganguli 2019
Outline

1. Optimization and Generalization in Deep Learning via Trajectories

2. Case Study: Linear Neural Networks
   - Trajectory Analysis
   - Optimization
   - Generalization

3. Conclusion
Gradient flow (GF) is a continuous version of GD (step size → 0):

$$\frac{d}{dt} \alpha(t) = -\nabla f(\alpha(t)) , \ t \in \mathbb{R}_{>0}$$
Gradient flow (GF) is a continuous version of GD (step size $\rightarrow 0$):

$$\frac{d}{dt} \alpha(t) = -\nabla f(\alpha(t)) \quad , \quad t \in \mathbb{R}_{>0}$$

Admits use of theoretical tools from differential geometry/equations
Balanced Trajectories

\[ x \rightarrow W_1 \rightarrow W_2 \rightarrow \cdots \rightarrow W_N \rightarrow y = W_N \cdots W_2 W_1 x \]
Balanced Trajectories

Loss $\ell(\cdot)$ for linear model induces **overparameterized objective** for LNN:

$$\phi(W_1, \ldots, W_N) := \ell(W_N \cdots W_2 W_1)$$
Balanced Trajectories

\[ x \rightarrow W_1 \rightarrow W_2 \rightarrow \cdots \rightarrow W_N \rightarrow y = W_N \cdots W_2 W_1 x \]

Loss \( \ell(\cdot) \) for linear model induces **overparameterized objective** for LNN:

\[ \phi(W_1, \ldots, W_N) := \ell(W_N \cdots W_2 W_1) \]

**Definition**

Weights \( W_1 \ldots W_N \) are **balanced** if \( W_{j+1}^T W_{j+1} = W_j W_j^T \), \( \forall j \).
Balanced Trajectories

Loss \( \ell(\cdot) \) for linear model induces **overparameterized objective** for LNN:

\[
\phi(W_1, \ldots, W_N) := \ell(W_N \cdots W_2 W_1)
\]

**Definition**

Weights \( W_1 \ldots W_N \) are **balanced** if \( W_{j+1}^T W_{j+1} = W_j W_j^T, \forall j \).

\[
\uparrow
\]

Holds approximately under \( \approx 0 \) init, exactly under residual \((I_d)\) init
Balanced Trajectories

\[ x \rightarrow W_1 \rightarrow W_2 \rightarrow \cdots \rightarrow W_N \rightarrow y = W_N \cdots W_2 W_1 x \]

Loss \( \ell(\cdot) \) for linear model induces **overparameterized objective** for LNN:

\[ \phi(W_1, \ldots, W_N) := \ell(W_N \cdots W_2 W_1) \]

**Definition**

Weights \( W_1 \ldots W_N \) are **balanced** if \( W_{j+1}^\top W_{j+1} = W_j W_j^\top \), \( \forall j \).

\[ \uparrow \]

Holds approximately under \( \approx 0 \) init, exactly under residual (\( I_d \)) init

**Claim**

*Trajectories of GF over LNN preserve balancedness: if \( W_1 \ldots W_N \) are balanced at init, they remain that way throughout GF optimization*
Balanced Trajectories — Proof

Claim

Trajectories of GF over LNN preserve balancedness: if $W_1 \ldots W_N$ are balanced at init, they remain that way throughout GF optimization.

Proof

Take transpose of eq, add to itself, and integrate (w.r.t. $t$):
Balanced Trajectories — Proof

Claim

*Trajectories of GF over LNN preserve balancedness: if $W_1 \ldots W_N$ are balanced at init, they remain that way throughout GF optimization*

Proof

GF over LNN:

$$\frac{d}{dt} W_j(t) = - \frac{\partial}{\partial W_j} \phi(W_1(t), \ldots, W_N(t))$$
Balanced Trajectories — Proof

Claim

Trajectories of GF over LNN preserve balancedness: if $W_1 \ldots W_N$ are balanced at init, they remain that way throughout GF optimization.

Proof

GF over LNN:

$$\frac{d}{dt} W_j(t) = - \frac{\partial}{\partial W_j} \phi(W_1(t), \ldots, W_N(t))$$

$$= - \prod_{i=j+1}^N W_i(t) \top \cdot \nabla \ell(W_N(t) \cdots W_1(t)) \cdot \prod_{i=1}^{j-1} W_i(t) \top$$
Balanced Trajectories — Proof

Claim

*Trajectories of GF over LNN preserve balancedness: if $W_1 \ldots W_N$ are balanced at init, they remain that way throughout GF optimization*  

Proof

GF over LNN:

$$\frac{d}{dt} W_j(t) = -\frac{\partial}{\partial W_j} \phi(W_1(t), \ldots, W_N(t))$$

$$= - \prod_{i=j+1}^{N} W_i(t) \cdot \nabla \ell(W_N(t) \cdots W_1(t)) \cdot \prod_{i=1}^{j-1} W_i(t)$$

$$\Rightarrow \left(\frac{d}{dt} W_j(t)\right) W_j(t)^\top \equiv W_{j+1}(t)^\top \left(\frac{d}{dt} W_{j+1}(t)\right)$$
Balanced Trajectories — Proof

Claim

Trajectories of GF over LNN preserve balancedness: if $W_1 \ldots W_N$ are balanced at init, they remain that way throughout GF optimization.

Proof

GF over LNN:

$$\frac{d}{dt} W_j(t) = -\frac{\partial}{\partial W_j} \phi \left( W_1(t), \ldots, W_N(t) \right)$$

$$= - \prod_{i=j+1}^N W_i(t) \top \cdot \nabla \ell \left( W_N(t) \cdots W_1(t) \right) \cdot \prod_{i=1}^{j-1} W_i(t) \top$$

$$\implies \left( \frac{d}{dt} W_j(t) \right) W_j(t) \top \equiv W_{j+1}(t) \top \left( \frac{d}{dt} W_{j+1}(t) \right)$$

Take transpose of eq, add to itself, and integrate (w.r.t. $t$):

$$W_j(t) W_j(t) \top \equiv W_{j+1}(t) \top W_{j+1}(t) + \text{const}$$
** Balanced Trajectories — Proof **

** Claim **

*Trajectories of GF over LNN preserve balancedness: if $W_1 \ldots W_N$ are balanced at init, they remain that way throughout GF optimization*

** Proof **

GF over LNN:

$$
\frac{d}{dt} W_j(t) = -\frac{\partial}{\partial W_j} \phi \left( W_1(t), \ldots, W_N(t) \right)
= - \prod_{i=j+1}^{N} W_i(t)^\top \cdot \nabla \ell \left( W_N(t) \cdots W_1(t) \right) \cdot \prod_{i=1}^{j-1} W_i(t)^\top
$$

$$
\Rightarrow \left( \frac{d}{dt} W_j(t) \right) W_j(t)^\top \equiv W_{j+1}(t)^\top \left( \frac{d}{dt} W_{j+1}(t) \right)
$$

Take transpose of eq, add to itself, and integrate (w.r.t. $t$):

$$
W_j(t) W_j(t)^\top \equiv W_{j+1}(t)^\top W_{j+1}(t) + \text{const}
$$

Balance at init $\Rightarrow \text{const} = 0$
Implicit Preconditioning

Question

How does \textbf{end-to-end matrix} $W_{1:N} := W_N \cdots W_1$ move on GF trajectories?

\begin{align*}
\textbf{Linear Neural Network} & \quad \textbf{Equivalent Linear Model} \\
\rightarrow W_1 \rightarrow W_2 \rightarrow \cdots \rightarrow W_N & \quad \Rightarrow W_{1:N} \\
\text{Gradient flow over } \phi(W_1,\ldots,W_N) & \quad ?
\end{align*}
Question

How does **end-to-end matrix** \( W_{1:N} := W_N \cdots W_1 \) move on GF trajectories?

**Linear Neural Network**

\[ W_1 \rightarrow W_2 \rightarrow \cdots \rightarrow W_N \]

Gradient flow over \( \phi(W_1, \ldots, W_N) \)

**Equivalent Linear Model**

\[ \rightarrow W_{1:N} \]

Preconditioned gradient flow over \( \ell(W_{1:N}) \)

**Theorem**

If \( W_1 \ldots W_N \) are balanced at init, \( W_{1:N} \) follows **end-to-end dynamics**:

\[
\frac{d}{dt} \text{vec} [W_{1:N}(t)] = -P_{W_{1:N}(t)} \cdot \text{vec} [\nabla \ell(W_{1:N}(t))]
\]

where \( P_{W_{1:N}(t)} \) is a preconditioner (PSD matrix) that “reinforces” \( W_{1:N}(t) \)
Implicit Preconditioning

Question

How does end-to-end matrix $W_{1:N} := W_N \cdots W_1$ move on GF trajectories?

**Linear Neural Network**

$\rightarrow W_1 \rightarrow W_2 \rightarrow \cdots \rightarrow W_N \rightarrow$

Gradient flow over $\phi(W_1, \ldots, W_N)$

**Equivalent Linear Model**

$\rightarrow W_{1:N} \rightarrow$

Preconditioned gradient flow over $\ell(W_{1:N})$

**Theorem**

If $W_1 \ldots W_N$ are balanced at init, $W_{1:N}$ follows end-to-end dynamics:

$$\frac{d}{dt} \text{vec} [W_{1:N}(t)] = -P_{W_{1:N}(t)} \cdot \text{vec} [\nabla \ell(W_{1:N}(t))]$$

where $P_{W_{1:N}(t)}$ is a preconditioner (PSD matrix) that “reinforces” $W_{1:N}(t)$

$$P_{W_{1:N}(t)} \cdot \text{vec} [\nabla \ell(W_{1:N}(t))] = \text{vec} \left[ \sum_{j=1}^{N} [W_{1:N}(t)W_{1:N}(t)^\top]_{N-j}^{N-j} \cdot \nabla \ell(W_{1:N}(t)) \cdot [W_{1:N}(t)^\top W_{1:N}(t)]_{j-1}^{j-1} \right]$$

Nadav Cohen (TAU)
**Case Study: Linear Neural Networks**

**Implicit Preconditioning**

**Question**

How does **end-to-end matrix** $W_{1:N} := W_N \cdots W_1$ move on GF trajectories?

**Linear Neural Network**

\[ \begin{align*}
&\rightarrow W_1 \rightarrow W_2 \rightarrow \cdots \rightarrow W_N \\
\text{Gradient flow over } &\phi(W_1, \ldots, W_N)
\end{align*} \]

**Equivalent Linear Model**

\[ \begin{align*}
&\rightarrow W_{1:N} \\
\text{Preconditioned } &\text{gradient flow over } \ell(W_{1:N})
\end{align*} \]

**Theorem**

If $W_1 \ldots W_N$ are balanced at init, $W_{1:N}$ follows **end-to-end dynamics**:

\[
\frac{d}{dt} \text{vec } [W_{1:N}(t)] = -P_{W_{1:N}(t)} \cdot \text{vec } [\nabla \ell(W_{1:N}(t))] 
\]

where $P_{W_{1:N}(t)}$ is a preconditioner (PSD matrix) that “reinforces” $W_{1:N}(t)$

**Adding (redundant) linear layers to classic linear model** induces preconditioner promoting movement in directions already taken!
Implicit Preconditioning — Proof Sketch

Theorem

If $W_1 \ldots W_N$ are balanced at init, $W_{1:N}$ follows end-to-end dynamics:

$$\frac{d}{dt} \text{vec} [W_{1:N}(t)] = -P_{W_{1:N}(t)} \cdot \text{vec} [\nabla \ell(W_{1:N}(t))]$$

where $P_{W_{1:N}(t)}$ is a preconditioner (PSD matrix) that “reinforces” $W_{1:N}(t)$

Proof Sketch
Theorem

If $W_1 \ldots W_N$ are balanced at init, $W_{1:N}$ follows end-to-end dynamics:

$$\frac{d}{dt} \text{vec}[W_{1:N}(t)] = -P_{W_{1:N}(t)} \cdot \text{vec}[\nabla \ell(W_{1:N}(t))]$$

where $P_{W_{1:N}(t)}$ is a preconditioner (PSD matrix) that “reinforces” $W_{1:N}(t)$

Proof Sketch

SVD: $W_j(t) = U_j(t)S_j(t)V_j(t)^\top$
Implicit Preconditioning — Proof Sketch

**Theorem**

If $W_1 \ldots W_N$ are balanced at init, $W_{1:N}$ follows **end-to-end dynamics**:

$$\frac{d}{dt} \text{vec} [W_{1:N}(t)] = -P_{W_{1:N}(t)} \cdot \text{vec} [\nabla \ell (W_{1:N}(t))]$$

where $P_{W_{1:N}(t)}$ is a preconditioner (PSD matrix) that “reinforces” $W_{1:N}(t)$

**Proof Sketch**

**SVD:** $W_j(t) = U_j(t)S_j(t)V_j(t)^\top$

Balance $(W_j(t)W_j(t)^\top \equiv W_{j+1}(t)^\top W_{j+1}(t)) \implies S_j(t) \equiv S_{j+1}(t) \land U_j(t) \equiv V_{j+1}(t)$
Implicit Preconditioning — Proof Sketch

**Theorem**

If \( W_1 \ldots W_N \) are balanced at init, \( W_{1:N} \) follows end-to-end dynamics:

\[
\frac{d}{dt} \text{vec}[W_{1:N}(t)] = -P_{W_{1:N}(t)} \cdot \text{vec}[\nabla \ell(W_{1:N}(t))]
\]

where \( P_{W_{1:N}(t)} \) is a preconditioner (PSD matrix) that “reinforces” \( W_{1:N}(t) \)

**Proof Sketch**

SVD: \( W_j(t) = U_j(t)S_j(t)V_j(t)^T \)

Balance \( (W_j(t)W_j(t)^T = W_{j+1}(t)^T W_{j+1}(t)) \) \( \implies S_j(t) = S_{j+1}(t) \land U_j(t) = V_{j+1}(t) \)

Products of weights thus simplify, yielding:

\[
\frac{d}{dt} W_{1:N}(t) = \\
\sum_{j=1}^{N} \prod_{i=j+1}^{N} W_i(t) \cdot \frac{d}{dt} W_j(t) \cdot \prod_{i=1}^{j-1} W_i(t)
\]
Implicit Preconditioning — Proof Sketch

**Theorem**

If \( W_1 \ldots W_N \) are balanced at init, \( W_{1:N} \) follows **end-to-end dynamics**:

\[
\frac{d}{dt} \text{vec} [W_{1:N}(t)] = -P_{W_{1:N}(t)} \cdot \text{vec} [\nabla \ell(W_{1:N}(t))]
\]

where \( P_{W_{1:N}(t)} \) is a preconditioner (PSD matrix) that “reinforces” \( W_{1:N}(t) \)

**Proof Sketch**

**SVD:** \( W_j(t) = U_j(t)S_j(t)V_j(t)^\top \)

Balance \((W_j(t)W_j(t)^\top)\equiv(W_{j+1}(t)^\top W_{j+1}(t))\) \(\implies\) \(S_j(t)\equiv S_{j+1}(t) \land U_j(t)\equiv V_{j+1}(t)\)

Products of weights thus simplify, yielding:

\[
\frac{d}{dt} W_{1:N}(t) = 
\sum_{j=1}^{N} \prod_{j+1} W_i(t) \cdot \left( -\frac{\partial}{\partial W_j} \phi(W_1(t), \ldots, W_N(t)) \right) \cdot \prod_{1}^{i=j-1} W_i(t)
\]
Implicit Preconditioning — Proof Sketch

**Theorem**

If \( W_1 \ldots W_N \) are balanced at init, \( W_{1:N} \) follows **end-to-end dynamics:**

\[
\frac{d}{dt}\vec{[W_{1:N}(t)]} = -P_{W_{1:N}(t)} \cdot \vec{[\nabla \ell(W_{1:N}(t))]} 
\]

where \( P_{W_{1:N}(t)} \) is a preconditioner (PSD matrix) that “reinforces” \( W_{1:N}(t) \)

**Proof Sketch**

**SVD:** \( W_j(t) = U_j(t)S_j(t)V_j(t)\)^T

**Balance** \( (W_j(t)W_j(t)^\top \equiv W_{j+1}(t)^\top W_{j+1}(t)) \implies S_j(t) \equiv S_{j+1}(t) \land U_j(t) \equiv V_{j+1}(t) \)

Products of weights thus simplify, yielding:

\[
\frac{d}{dt} W_{1:N}(t) = \\
\sum_{j=1}^{N} \prod_{i=j+1}^{N} W_i(t) \cdot \left( - \prod_{i=j+1}^{N} W_i(t)^\top \nabla \ell(W_{1:N}(t)) \prod_{i=1}^{j-1} W_i(t)^\top \right) \cdot \prod_{i=1}^{i=j-1} W_i(t)
\]
Implicit Preconditioning — Proof Sketch

**Theorem**

If $W_1 \ldots W_N$ are balanced at init, $W_{1:N}$ follows **end-to-end dynamics**:

\[
\frac{d}{dt} \text{vec}[W_{1:N}(t)] = -P_{W_{1:N}(t)} \cdot \text{vec}[\nabla \ell(W_{1:N}(t))]
\]

where $P_{W_{1:N}(t)}$ is a preconditioner (PSD matrix) that “reinforces” $W_{1:N}(t)$

**Proof Sketch**

SVD: $W_j(t) = U_j(t)S_j(t)V_j(t)^\top$

Balance: $(W_j(t)W_j(t)^\top = W_{j+1}(t)^\top W_{j+1}(t)) \implies S_j(t) = S_{j+1}(t) \land U_j(t) = V_{j+1}(t)$

Products of weights thus simplify, yielding:

\[
\frac{d}{dt} W_{1:N}(t) = \\
- \sum_{j=1}^{N} \prod_{i=j+1}^{N} W_i(t) \prod_{i=j+1}^{N} W_i(t)^\top \cdot \nabla \ell(W_{1:N}(t)) \cdot \prod_{i=1}^{j-1} W_i(t)^\top \prod_{i=1}^{i=j-1} W_i(t)
\]
Implicit Preconditioning — Proof Sketch

**Theorem**

If $W_1 \ldots W_N$ are balanced at init, $W_{1:N}$ follows **end-to-end dynamics**:

$$
\frac{d}{dt} \text{vec}[W_{1:N}(t)] = -P_{W_{1:N}(t)} \cdot \text{vec}[\nabla \ell(W_{1:N}(t))]
$$

where $P_{W_{1:N}(t)}$ is a preconditioner (PSD matrix) that “reinforces” $W_{1:N}(t)$

**Proof Sketch**

SVD: $W_j(t) = U_j(t)S_j(t)V_j(t)^\top$

Balance $(W_j(t)W_j(t)^\top \equiv W_{j+1}(t)^\top W_{j+1}(t)) \implies S_j(t) \equiv S_{j+1}(t) \land U_j(t) \equiv V_{j+1}(t)$

Products of weights thus simplify, yielding:

$$
\frac{d}{dt} W_{1:N}(t) =
$$

$$
- \sum_{j=1}^{N} \prod_{i=j+1}^{N} W_i(t) \prod_{i=j+1}^{N} W_i(t)^\top \cdot \nabla \ell(W_{1:N}(t)) \cdot \prod_{i=1}^{j-1} W_i(t)^\top \prod_{i=1}^{i=j-1} W_i(t)
$$
Implicit Preconditioning — Proof Sketch

**Theorem**

If $W_1 \ldots W_N$ are balanced at init, $W_{1:N}$ follows end-to-end dynamics:

$$\frac{d}{dt} \text{vec} [W_{1:N}(t)] = -P_{W_{1:N}(t)} \cdot \text{vec} \left[ \nabla \ell(W_{1:N}(t)) \right]$$

where $P_{W_{1:N}(t)}$ is a preconditioner (PSD matrix) that “reinforces” $W_{1:N}(t)$

**Proof Sketch**

**SVD:** $W_j(t) = U_j(t)S_j(t)V_j(t)\top$

Balance $(W_j(t)W_j(t)\top \equiv W_{j+1}(t)\top W_{j+1}(t)) \implies S_j(t) \equiv S_{j+1}(t) \land U_j(t) \equiv V_{j+1}(t)$

Products of weights thus simplify, yielding:

$$\frac{d}{dt} W_{1:N}(t) =$$

$$- \sum_{j=1}^{N} \left[ W_{1:N}(t)W_{1:N}(t)\top \right]^{\frac{N-j}{N}} \cdot \nabla \ell(W_{1:N}(t)) \cdot \left[ W_{1:N}(t)\top W_{1:N}(t) \right]^{\frac{j-1}{N}}$$
Case Study: Linear Neural Networks

Trajectory Analysis

Implicit Preconditioning — Proof Sketch

Theorem

If $W_1 \ldots W_N$ are balanced at init, $W_{1:N}$ follows end-to-end dynamics:

$$\frac{d}{dt} \text{vec } [W_{1:N}(t)] = -P_{W_{1:N}(t)} \cdot \text{vec } [\nabla \ell(W_{1:N}(t))]$$

where $P_{W_{1:N}(t)}$ is a preconditioner (PSD matrix) that “reinforces” $W_{1:N}(t)$

Proof Sketch

SVD: $W_j(t) = U_j(t)S_j(t)V_j(t)^\top$

Balance $(W_j(t)W_j(t)^\top \equiv W_{j+1}(t)^\top W_{j+1}(t)) \implies S_j(t) \equiv S_{j+1}(t) \wedge U_j(t) \equiv V_{j+1}(t)$

Products of weights thus simplify, yielding:

$$\frac{d}{dt} W_{1:N}(t) =$$

$$- \sum_{j=1}^{N} \left[ W_{1:N}(t)W_{1:N}(t)^\top \right]^{\frac{N-j}{N}} \cdot \nabla \ell(W_{1:N}(t)) \cdot \left[ W_{1:N}(t)^\top W_{1:N}(t) \right]^{\frac{j-1}{N}}$$

Vectorizing gives end-to-end dynamics (with closed-form expression for $P_{W_{1:N}(t)}$)
Trajectories Cannot Be Emulated via Regularization

End-to-end dynamics (implicit preconditioning):

\[
\frac{d}{dt} \text{vec} [W_{1:N}(t)] = -P_{W_{1:N}(t)} \cdot \text{vec} [\nabla \ell (W_{1:N}(t))]
\]
Trajectories Cannot Be Emulated via Regularization

End-to-end dynamics (implicit preconditioning):

$$\frac{d}{dt} \text{vec} [W_{1:N}(t)] = -P_{W_{1:N}(t)} \cdot \text{vec} [\nabla \ell (W_{1:N}(t))]$$

Theorem

If $\nabla \ell (0) \neq 0$ then there exists no function $F(W)$ such that

$$\text{vec} [\nabla F(W)] = P_W \cdot \text{vec} [\nabla \ell (W)]$$
End-to-end dynamics (implicit preconditioning):

\[
\frac{d}{dt} \text{vec}[W_{1:N}(t)] = -P_{W_{1:N}(t)} \cdot \text{vec}[\nabla \ell(W_{1:N}(t))] \neq -\text{vec}[\nabla F(W_{1:N}(t))]
\]

Theorem

If \( \nabla \ell(0) \neq 0 \) then \( \nexists \) function \( F(W) \) s.t. \( \text{vec}[\nabla F(W)] = P_W \cdot \text{vec}[\nabla \ell(W)] \)
Trajectories Cannot Be Emulated via Regularization

End-to-end dynamics (implicit preconditioning):

$$\frac{d}{dt} \text{vec} [W_{1:N}(t)] = -P_{W_{1:N}(t)} \cdot \text{vec} [\nabla \ell (W_{1:N}(t))] \neq -\text{vec} [\nabla F(W_{1:N}(t))]$$

Theorem

*If* $\nabla \ell (0) \neq 0$ *then* $\exists$ *function* $F(W)$ *s.t.* $\text{vec} [\nabla F(W)] = P_W \cdot \text{vec} [\nabla \ell (W)]$

Trajectories with LNN cannot be emulated by regularizing objective!
Trajectories Cannot Be Emulated via Regularization

End-to-end dynamics (implicit preconditioning):
\[
\frac{d}{dt} \operatorname{vec}[W_{1:N}(t)] = -P_{W_{1:N}(t)} \cdot \operatorname{vec}[\nabla \ell(W_{1:N}(t))] \neq -\operatorname{vec}[\nabla F(W_{1:N}(t))] 
\]

Theorem
If \( \nabla \ell(0) \neq 0 \) then \( \nexists \) function \( F(W) \) s.t. \( \operatorname{vec}[\nabla F(W)] = P_W \cdot \operatorname{vec}[\nabla \ell(W)] \)

Trajectories with LNN cannot be emulated by regularizing objective!

\[
\int_\Gamma P_W \cdot \operatorname{vec}[\nabla \ell(W)] \neq 0 
\]

contradicts gradient theorem!
Outline

1. Optimization and Generalization in Deep Learning via Trajectories

2. Case Study: Linear Neural Networks
   - Trajectory Analysis
   - Optimization
   - Generalization

3. Conclusion
Prominent approach for analyzing optimization in DL (in spirit of classical learning theory) is via critical points in the objective

- **Good local minimum** (≈ global minimum)
- **Poor local minimum**
- **Strict saddle**
- **Non-strict saddle**

Result (cf. Ge et al. 2015; Lee et al. 2016)

If:
1. there are no poor local minima; and
2. all saddle points are strict,
then GD converges to global min
Classic Approach: Characterization of Critical Points

Prominent approach for analyzing optimization in DL (in spirit of classical learning theory) is via **critical points** in the objective function.

![Diagram showing different types of critical points](image)

**Good local minimum (≈ global minimum)**

**Poor local minimum**

**Strict saddle**

**Non-strict saddle**

**Result** *(cf. Ge et al. 2015; Lee et al. 2016)*

If: (1) there are no poor local minima; and (2) all saddle points are strict, then GD converges to global min.
Classic Approach: Characterization of Critical Points

Prominent approach for analyzing optimization in DL (in spirit of classical learning theory) is via **critical points** in the objective.

### Result (cf. Ge et al. 2015; Lee et al. 2016)

If: **(1)** there are no poor local minima; and **(2)** all saddle points are strict, then GD converges to global min.

### Motivation

Motivated by this, many\(^1\) studied the validity of **(1)** and/or **(2)**

---

\(^1\) e.g. Haeffele & Vidal 2015; Kawaguchi 2016; Soudry & Carmon 2016; Safran & Shamir 2018
Prominent approach for analyzing optimization in DL (in spirit of classical learning theory) is via critical points in the objective

Result (cf. Ge et al. 2015; Lee et al. 2016)
If: (1) there are no poor local minima; and (2) all saddle points are strict, then GD converges to global min

Motivated by this, many\(^1\) studied the validity of (1) and/or (2)

Limitation: deep (≥ 3 layer) models violate (2) (consider all weights = 0)!

\(^1\) e.g. Haeffele & Vidal 2015; Kawaguchi 2016; Soudry & Carmon 2016; Safran & Shamir 2018
Applying Our Trajectory Analysis

Trajectory analysis revealed implicit preconditioning on end-to-end matrix:

\[ \frac{d}{dt} \text{vec} \left[ W_1: N(t) \right] = -P W_1: N(t) \cdot \text{vec} \left[ \nabla \ell (W_1: N(t)) \right] \]

\( P W_1: N(t) \succ 0 \) when \( W_1: N(t) \) has full rank

⇒ loss decreases until:

1. \( \nabla \ell (W_1: N(t)) = 0 \)
2. \( W_1: N(t) \) is singular

\( \ell (\cdot) \) is typically convex ⇒ (1) means global min was reached

Corollary

Assume \( \ell (\cdot) \) is convex and LNN is init such that:

1. \( \ell (W_1: N) < \ell (W) \) for any singular \( W_2 \ldots W_N \) are balanced

Then, GF converges to global min
Applying Our Trajectory Analysis

Trajectory analysis revealed **implicit preconditioning** on end-to-end matrix:

\[
\frac{d}{dt} \text{vec} [W_{1:N}(t)] = -P_{W_{1:N}(t)} \cdot \text{vec} [\nabla \ell (W_{1:N}(t))] 
\]

\( P_{W_{1:N}(t)} \succ 0 \) when \( W_{1:N}(t) \) has full rank

\( \nabla \ell (\cdot) \) is typically convex

\( (1) \) means global min was reached

Corollary

Assume \( \ell (\cdot) \) is convex and LNN is init such that:

1. \( \ell (W_{1:N}) < \ell (W) \) for any singular \( W_{2:N} \)
2. \( W_{1:N} \) are balanced

Then, GF converges to global min
Applying Our Trajectory Analysis

Trajectory analysis revealed **implicit preconditioning** on end-to-end matrix:

\[
\frac{d}{dt} \text{vec } [W_{1:N}(t)] = -P_{W_{1:N}(t)} \cdot \text{vec } [\nabla \ell (W_{1:N}(t))]
\]

\[P_{W_{1:N}(t)} \succ 0 \text{ when } W_{1:N}(t) \text{ has full rank}\]
Case Study: Linear Neural Networks

Optimization

Applying Our Trajectory Analysis

Trajectory analysis revealed **implicit preconditioning** on end-to-end matrix:

$$\frac{d}{dt} \text{vec} [W_{1:N}(t)] = -P_{W_{1:N}(t)} \cdot \text{vec} [\nabla \ell (W_{1:N}(t))]$$

$$P_{W_{1:N}(t)} \succ 0 \text{ when } W_{1:N}(t) \text{ has full rank } \implies \text{loss decreases until:}$$

1. $$\nabla \ell (W_{1:N}(t)) = 0$$  
   or  
2. $$W_{1:N}(t)$$ is singular

**Corollary**

Assume $$\ell (\cdot)$$ is convex and LNN is init such that:

1. $$\ell (W_1) < \ell (W)$$ for any singular $$W_2 \cdots W_N$$ are balanced

Then, GF converges to global min
Applying Our Trajectory Analysis

Trajectory analysis revealed \textit{implicit preconditioning} on end-to-end matrix:

\[
\frac{d}{dt} \text{vec } [W_{1:N}(t)] = -P_{W_{1:N}(t)} \cdot \text{vec } [\nabla \ell(W_{1:N}(t))]
\]

\(P_{W_{1:N}(t)} \succ 0\) when \(W_{1:N}(t)\) has full rank \(\implies\) loss decreases until:

\(1\) \(\nabla \ell(W_{1:N}(t)) = 0\) \quad \text{or} \quad \(2\) \(W_{1:N}(t)\) is singular

\(\ell(\cdot)\) is typically convex \(\implies\) \(1\) means global min was reached
Applying Our Trajectory Analysis

Trajectory analysis revealed implicit preconditioning on end-to-end matrix:

$$\frac{d}{dt} \text{vec} [W_{1:N}(t)] = -P_{W_{1:N}(t)} \cdot \text{vec} [\nabla \ell(W_{1:N}(t))]$$

$P_{W_{1:N}(t)} \succ 0$ when $W_{1:N}(t)$ has full rank $\implies$ loss decreases until:

1. $\nabla \ell(W_{1:N}(t)) = 0$ or 
2. $W_{1:N}(t)$ is singular

$\ell(\cdot)$ is typically convex $\implies$ (1) means global min was reached

Corollary

Assume $\ell(\cdot)$ is convex and LNN is init such that:

1. $\ell(W_{1:N}) < \ell(W)$ for any singular $W$
2. $W_1 \ldots W_N$ are balanced

Then, GF converges to global min
Corollary

Assume $\ell(\cdot)$ is convex and LNN is init such that:

1. $\ell(W_1: \mathbb{N}) < \ell(W)$ for any singular $W_1 \ldots W_N$ are balanced

Then, GF converges to global min

Claim

Our assumptions on init:

- Are necessary (violating any of them can lead to divergence)
- For out dim 1, hold with const prob under random "balanced" init

Guarantee of efficient (linear rate) convergence to global min!

Most general guarantee to date for GD efficiently training deep net.
Corollary

Assume $\ell(\cdot)$ is convex and LNN is init such that:

1. $\ell(W_{1:N}) < \ell(W)$ for any singular $W$
2. $W_1 \ldots W_N$ are balanced

Then, GF converges to global min
Corollary

Assume $\ell(\cdot)$ is convex and LNN is init such that:

1. $\ell(W_{1:N}) < \ell(W)$, $\forall W$ s.t. $\sigma_{\text{min}}(W) = 0$
2. $W_1 \ldots W_N$ are balanced

Then, GF converges to global min
Corollary

Assume $\ell(\cdot)$ is convex and LNN is init such that:

1. $\ell(W_{1:N}) < \ell(W)$, $\forall W$ s.t. $\sigma_{\text{min}}(W) = 0$
2. $W_{j+1}^T W_{j+1} = W_j W_j^T$, $\forall j$

Then, GF converges to global min
Corollary

Assume $\ell(\cdot)$ is convex and LNN is init such that:

1. $\ell(W_{1:N}) < \ell(W)$, $\forall W$ s.t. $\sigma_{\min}(W) = 0$

2. $\|W_{j+1}^T W_{j+1} - W_j W_j^T\|_F = 0$, $\forall j$

Then, GF converges to global min
**Theorem**

Assume $\ell(\cdot)$ is convex and LNN is init such that:

1. $\ell(W_{1:N}) < \ell(W)$, $\forall W$ s.t. $\sigma_{\min}(W) = 0$
2. $\|W_{j+1}^T W_{j+1} - W_j W_j^T\|_F = 0$, $\forall j$

Then, GF converges to global min
Theorem

Assume $\ell(\cdot) = \ell_2$ loss and LNN is init such that:

1. $\ell(W_{1:N}) < \ell(W)$, $\forall W$ s.t. $\sigma_{\min}(W) = 0$

2. $\|W_{j+1}^T W_{j+1} - W_j W_j^T\|_F = 0$, $\forall j$

Then, GF converges to global min
From Gradient Flow to Gradient Descent

**Theorem**

Assume $\ell(\cdot) = \ell_2$ loss and LNN is init such that:

1. $\ell(W_{1:N}) < \ell(W)$, $\forall W$ s.t. $\sigma_{\min}(W) \leq c$
2. $\|W_{j+1}^T W_{j+1} - W_j W_j^T\|_F = 0$, $\forall j$

Then, GF converges to global min
Case Study: Linear Neural Networks

From Gradient Flow to Gradient Descent

Theorem

Assume $\ell(\cdot) = \ell_2$ loss and LNN is init such that:

1. $\ell(W_{1:N}) < \ell(W), \forall W \text{ s.t. } \sigma_{\min}(W) \leq c$
2. $\|W_{j+1}^T W_{j+1} - W_j W_j^T\|_F \leq O(c^2) \quad \forall j$

Then, GF converges to global min
Theorem

Assume \( \ell(\cdot) = \ell_2 \) loss and LNN is init such that:

1. \( \ell(W_{1:N}) < \ell(W), \forall W \text{ s.t. } \sigma_{\text{min}}(W) \leq c \)
2. \( \|W_{j+1}^T W_{j+1} - W_j W_j^T\|_F \leq O(c^2), \forall j \)

Then, GD with step size \( \eta \leq O(c^4) \) gives: \( \text{loss(iteration } t) \leq e^{-\Omega(c^2 \eta t)} \)
Case Study: Linear Neural Networks

From Gradient Flow to Gradient Descent

**Theorem**

Assume $\ell(\cdot) = \ell_2$ loss and LNN is init such that:

1. $\ell(W_{1:N}) < \ell(W)$, $\forall W$ s.t. $\sigma_{\text{min}}(W) \leq c$

2. $\|W_{j+1}^T W_{j+1} - W_j W_j^T\|_F \leq O(c^2)$, $\forall j$

Then, GD with step size $\eta \leq O(c^4)$ gives: $\text{loss(iteration } t) \leq e^{-\Omega(c^2 \eta t)}$

**Claim**

*Our assumptions on init:*

- Necessary (violating any of them can lead to divergence)
- For out dim 1, hold with const prob under random "balanced" init
- Guarantee of efficient (linear rate) convergence to global min!
- Most general guarantee to date for GD efficiently training deep net.

Nadav Cohen (TAU)
Deep Linear Nets via Trajectories of GD
Princeton COS 597B, Dec'19
Case Study: Linear Neural Networks

From Gradient Flow to Gradient Descent

**Theorem**

Assume $\ell(\cdot) = \ell_2$ loss and LNN is init such that:

1. $\ell(W_{1:N}) < \ell(W)$, $\forall W$ s.t. $\sigma_{\text{min}}(W) \leq c$
2. $\|W_{j+1}^T W_{j+1} - W_j W_j^T\|_F \leq \mathcal{O}(c^2)$, $\forall j$

Then, GD with step size $\eta \leq \mathcal{O}(c^4)$ gives: $\text{loss(iteration } t) \leq e^{-\Omega(c^2 \eta t)}$

**Claim**

Our assumptions on init:

- Are necessary (violating any of them can lead to divergence)
Theorem

Assume $\ell(\cdot) = \ell_2$ loss and LNN is init such that:

1. $\ell(W_{1:N}) < \ell(W), \forall W \text{ s.t. } \sigma_{\min}(W) \leq c$
2. $\|W_{j+1}^T W_{j+1} - W_j W_j^T\|_F \leq O(c^2), \forall j$

Then, GD with step size $\eta \leq O(c^4)$ gives: $\text{loss(iteration } t) \leq e^{-\Omega(c^2 \eta t)}$

Claim

Our assumptions on init:

- Are necessary (violating any of them can lead to divergence)
- For out dim 1, hold with const prob under random “balanced” init
Theorem

Assume $\ell(\cdot) = \ell_2$ loss and LNN is init such that:

1. $\ell(W_{1:N}) < \ell(W)$, $\forall W$ s.t. $\sigma_{\text{min}}(W) \leq c$
2. $\|W_{j+1}^T W_{j+1} - W_j W_j^T\|_F \leq O(c^2)$, $\forall j$

Then, GD with step size $\eta \leq O(c^4)$ gives: $\text{loss(\text{iteration } t)} \leq e^{-\Omega(c^2 \eta t)}$

Claim

Our assumptions on init:

- Are necessary (violating any of them can lead to divergence)
- For out dim 1, hold with const prob under random “balanced” init

Guarantee of efficient (linear rate) convergence to global min! Most general guarantee to date for GD efficiently training deep net.
Effect of Depth on Optimization

Viewpoint of classical learning theory:
Convex optimization is easier than non-convex
Hence depth complicates optimization

Our trajectory analysis reveals:
not always true...
Effect of Depth on Optimization

Viewpoint of classical learning theory:

- Convex optimization is easier than non-convex
Effect of Depth on Optimization

Viewpoint of classical learning theory:

- Convex optimization is easier than non-convex

Hence depth complicates optimization
Effect of Depth on Optimization

Viewpoint of classical learning theory:
- Convex optimization is easier than non-convex

Hence depth complicates optimization

Our trajectory analysis reveals: not always true...
Discrete version of end-to-end dynamics for LNN:

$$\text{vec}\left[ W_1 : N \right](t+1) \leftarrow \text{vec}\left[ W_1 : N \right](t) - \eta \cdot P_{W_1 : N}(t) \cdot \text{vec}\left[ \nabla \ell(W_1 : N)(t) \right]$$

Claim

$$\forall p > 2, \exists \text{settings where }\ell(\cdot) = \ell_p\text{ loss (i.e. }\ell(W) = \frac{1}{m} \sum_{i=1}^{m} \| W_{x_i} - y_i \|^p)$$

and disc end-to-end dynamics reach global min arbitrarily faster than GD

Experiment

Regression problem from UCI ML Repository; $$\ell_4$$ loss

Depth can speed-up GD, even without any gain in expressiveness, and despite introducing non-convexity!
Acceleration by Depth

Discrete version of end-to-end dynamics for LNN:

\[
\text{vec} \left[ W_{1:N}(t + 1) \right] \leftarrow \text{vec} \left[ W_{1:N}(t) \right] - \eta \cdot P_{W_{1:N}(t)} \cdot \text{vec} \left[ \nabla \ell(W_{1:N}(t)) \right]
\]
Acceleration by Depth

Discrete version of end-to-end dynamics for LNN:

\[
\text{vec}\left[ W_{1:N}(t + 1) \right] \leftarrow \text{vec}\left[ W_{1:N}(t) \right] - \eta \cdot P_{W_{1:N}(t)} \cdot \text{vec}\left[ \nabla \ell(W_{1:N}(t)) \right]
\]

Claim

∀p > 2, ∃ settings where \( \ell(\cdot) = \ell_p \) loss (i.e. \( \ell(W) = \frac{1}{m} \sum_{i=1}^{m} \| Wx_i - y_i \|_p^p \)) and disc end-to-end dynamics reach global min arbitrarily faster than GD.
Acceleration by Depth

Discrete version of end-to-end dynamics for LNN:
\[
\text{vec}[W_{1:N}(t + 1)] \leftarrow \text{vec}[W_{1:N}(t)] - \eta \cdot P_{W_{1:N}(t)} \cdot \text{vec}[\nabla \ell(W_{1:N}(t))]
\]

Claim

\(\forall p > 2, \exists\) settings where \(\ell(\cdot) = \ell_p\) loss (i.e. \(\ell(W) = \frac{1}{m} \sum_{i=1}^{m} \|W x_i - y_i\|_p^p\)) and disc end-to-end dynamics reach global min arbitrarily faster than GD

Experiment
Acceleration by Depth

Discrete version of end-to-end dynamics for LNN:

$$\text{vec}[W_{1:N}(t + 1)] \leftarrow \text{vec}[W_{1:N}(t)] - \eta \cdot P_{W_{1:N}(t)} \cdot \text{vec}[\nabla \ell(W_{1:N}(t))]$$

**Claim**

$$\forall p > 2, \exists \text{ settings where } \ell(\cdot) = \ell_p \text{ loss (i.e. } \ell(W) = \frac{1}{m} \sum_{i=1}^{m} \|Wx_i - y_i\|_p^p) \text{ and disc end-to-end dynamics reach global min arbitrarily faster than GD}$$

**Experiment**

Regression problem from UCI ML Repository ; \(\ell_4\) loss
Acceleration by Depth

Discrete version of end-to-end dynamics for LNN:

\[ \text{vec}[W_{1:N}(t + 1)] \leftarrow \text{vec}[W_{1:N}(t)] - \eta \cdot P_{W_{1:N}(t)} \cdot \text{vec}[\nabla \ell(W_{1:N}(t))] \]

Claim

\( \forall p > 2, \exists \text{ settings where } \ell(\cdot) = \ell_p \text{ loss (i.e. } \ell(W) = \frac{1}{m} \sum_{i=1}^{m} \|Wx_i - y_i\|_p \text{)} \) and disc end-to-end dynamics reach global min arbitrarily faster than GD

Experiment

Regression problem from UCI ML Repository; \( \ell_4 \) loss
Acceleration by Depth

Discrete version of end-to-end dynamics for LNN:

\[
\text{vec}[W_{1:N}(t+1)] \leftarrow \text{vec}[W_{1:N}(t)] - \eta \cdot P_{W_{1:N}(t)} \cdot \text{vec}[\nabla \ell(W_{1:N}(t))]
\]

Claim

\(\forall p > 2, \exists \text{ settings where } \ell(\cdot) = \ell_p \text{ loss (i.e. } \ell(W) = \frac{1}{m} \sum_{i=1}^{m} \|Wx_i - y_i\|_p^p}\) and disc end-to-end dynamics reach global min arbitrarily faster than GD.

Experiment

Regression problem from UCI ML Repository; \(\ell_4\) loss

Depth can speed-up GD, even without any gain in expressiveness, and despite introducing non-convexity!
Outline

1. Optimization and Generalization in Deep Learning via Trajectories

2. Case Study: Linear Neural Networks
   - Trajectory Analysis
   - Optimization
   - Generalization

3. Conclusion
Setting: Matrix Completion

Matrix completion: recover matrix given subset of entries

<table>
<thead>
<tr>
<th></th>
<th>4</th>
<th>?</th>
<th>?</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob</td>
<td>4</td>
<td>?</td>
<td>?</td>
<td>4</td>
</tr>
<tr>
<td>Alice</td>
<td>?</td>
<td>5</td>
<td>4</td>
<td>?</td>
</tr>
<tr>
<td>Joe</td>
<td>?</td>
<td>5</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>
Setting: Matrix Completion

Matrix completion: recover matrix given subset of entries

<table>
<thead>
<tr>
<th></th>
<th>Bob</th>
<th>?</th>
<th>?</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>?</td>
<td>5</td>
<td>4</td>
<td>?</td>
</tr>
<tr>
<td>Joe</td>
<td>?</td>
<td>5</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

Can be viewed as classification (regression) problem:

observed entries ↔ training data
unobserved entries ↔ test data
### Setting: Matrix Completion

**Matrix completion**: recover matrix given subset of entries

<table>
<thead>
<tr>
<th></th>
<th>Bob</th>
<th>Alice</th>
<th>Joe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Movie</td>
<td>4</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Rating</td>
<td>?</td>
<td>4</td>
<td>?</td>
</tr>
</tbody>
</table>

Can be viewed as classification (regression) problem:

- *observed entries* $\leftrightarrow$ *training data*
- *unobserved entries* $\leftrightarrow$ *test data*

**Standard Assumption**

Matrix to recover (**ground truth**) has low rank
Setting: Matrix Completion

**Matrix completion:** recover matrix given subset of entries

<table>
<thead>
<tr>
<th></th>
<th>Bob</th>
<th>?</th>
<th>?</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>?</td>
<td>5</td>
<td>4</td>
<td>?</td>
</tr>
<tr>
<td>Joe</td>
<td>?</td>
<td>5</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

Can be viewed as classification (regression) problem:

- **observed entries**  ↔  **training data**
- **unobserved entries**  ↔  **test data**

**Standard Assumption**
Matrix to recover (ground truth) has low rank

**Classical Result** *(cf. Candes & Recht 2008)*
Nuclear norm minimization (convex program) perfectly recovers (“almost any”) low rank matrix if observations are sufficiently many
Matrix completion via two-layer LNN:

- Parameterize ground truth as $W_2 W_1$

\[
\begin{bmatrix}
4 & ? & ? & 4 \\
? & 5 & 4 & ? \\
? & 5 & ? & ? \\
\end{bmatrix}
= W_2 \ast W_1
\]
Two-Layer Network $\leftrightarrow$ Matrix Factorization

Matrix completion via two-layer LNN:

- Parameterize ground truth as $W_2 W_1$

\[
\begin{array}{ccc}
4 & ? & ? & 4 \\
? & 5 & 4 & ? \\
? & 5 & ? & ?
\end{array} = W_2 \ast W_1
\]

- Known as matrix factorization (MF)
Matrix completion via two-layer LNN:

- Parameterize ground truth as $W_2 W_1$

\[
\begin{array}{ccc}
4 & ? & ? & 4 \\
? & 5 & 4 & ? \\
? & 5 & ? & ? \\
\end{array}
= W_2 \ast W_1
\]

- Known as matrix factorization (MF)

**Empirical Phenomenon**

GD (with step size $\ll 1$ and init $\approx 0$) over MF recovers low rank matrices, even when shared dim of $W_1, W_2$ doesn’t constrain rank!
Two-Layer Network $\leftrightarrow$ Matrix Factorization

Matrix completion via two-layer LNN:

- Parameterize ground truth as $W_2 W_1$

\[
\begin{array}{ccc}
4 & ? & ? \\
? & 5 & 4 \\
? & 5 & ? \\
\end{array} = W_2 \ast W_1
\]

- Known as matrix factorization (MF)

Empirical Phenomenon

GD (with step size $\ll 1$ and init $\approx 0$) over MF recovers low rank matrices, even when shared dim of $W_1, W_2$ doesn’t constrain rank!

Conjecture (Gunasekar et al. 2017)

GD (with step size $\ll 1$ and init $\approx 0$) over MF converges to solution with min nuclear norm (among those fitting observations)
Two-Layer Network \( \iff \) Matrix Factorization

Matrix completion via two-layer LNN:

- Parameterize ground truth as \( W_2 W_1 \)

\[
\begin{bmatrix}
4 & ? & ? & 4 \\
? & 5 & 4 & ? \\
? & 5 & ? & ? \\
\end{bmatrix}
= W_2 \ast W_1
\]

- Known as matrix factorization (MF)

**Empirical Phenomenon**

GD (with step size \( \ll 1 \) and init \( \approx 0 \)) over MF recovers low rank matrices, even when shared dim of \( W_1, W_2 \) doesn’t constrain rank!

**Conjecture (Gunasekar et al. 2017)**

\( GD (\text{with step size} \ll 1 \text{ and init} \approx 0) \text{ over MF converges to solution with min nuclear norm (among those fitting observations)} \)

Gunasekar et al. proved conjecture for certain restricted setting
Matrix completion via $N$-layer LNN:

- Parameterize ground truth as $W_N \cdots W_2 W_1$

\[
\begin{array}{ccc}
4 & ? & ? \\
? & 5 & 4 \\
? & 5 & ? \\
\end{array}
\Rightarrow
\begin{array}{c}
W_N \\
\cdot \\
\cdot \\
W_1
\end{array}
\]
Matrix completion via $N$-layer LNN:

- Parameterize ground truth as $W_N \cdots W_2 W_1$

\[
\begin{bmatrix}
4 & ? & ? & 4 \\
? & 5 & 4 & ? \\
? & 5 & ? & ?
\end{bmatrix}
= \begin{bmatrix}
W_N \\
\end{bmatrix} \begin{bmatrix}
* \\
\end{bmatrix} \begin{bmatrix}
W_2 \\
\end{bmatrix} \begin{bmatrix}
* \\
\end{bmatrix} \begin{bmatrix}
W_1 \\
\end{bmatrix}
\]

- We refer to this as **deep matrix factorization (DMF)**
Matrix completion via $N$-layer LNN:

- Parameterize ground truth as $W_N \cdots W_2 W_1$

$$
\begin{array}{ccc}
4 & ? & ? & 4 \\
? & 5 & 4 & ? \\
? & 5 & ? & ?
\end{array}
\begin{array}{ccc}
W_N & * & \cdots & * \\
W_2 & * \\
W_1
\end{array}
$$

- We refer to this as deep matrix factorization (DMF)

**Experiment**

Completion of low rank matrix via GD over DMF

![Graph showing reconstruction error vs. number of observations with different depths]
Matrix completion via $N$-layer LNN:

- Parameterize ground truth as $W_N \cdots W_2 W_1$

\[
\begin{bmatrix}
4 & ? & ? & 4 \\
? & 5 & 4 & ? \\
? & 5 & ? & ?
\end{bmatrix} = 
\begin{bmatrix}
\* & \cdots & \* \\
W_N & \* & \* \\
W_2 & \* & \* \\
W_1 & \* & \*
\end{bmatrix}
\]

- We refer to this as \textit{deep matrix factorization (DMF)}

\section*{Experiment}
Completion of low rank matrix via GD over DMF

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{chart.png}
\caption{Depth enhanced implicit regularization towards low rank!}
\end{figure}
Can the Implicit Regularization Be Captured by Norms?

Conjecture of Gunasekar et al. 2017 (in spirit of classical learning theory):

\[ \text{implicit regularization with depth } 2^{\text{LNN (MF)}} \leftarrow \rightarrow \minimizing \text{nuclear norm (surrogate for rank)} \]

In light of our experiment, natural to hypothesize:

\[ \text{implicit regularization with deeper LNN (DMF)} \leftarrow \rightarrow \minimizing \text{other norm closer to rank} \]

Example:

Schatten-\(p\) quasi-norm to the power of \(p\):
\[
\| W \|_p^{\text{S}} := \sum_r \sigma_r^p (W)
\]

\(\sigma_r\) are singular values of \(W\):
\(p = 1\): nuclear norm, corresponds to depth 2 by Gunasekar et al. 2017
\(0 < p < 1\): closer to rank, may correspond to higher depths
Can the Implicit Regularization Be Captured by Norms?

Conjecture of Gunasekar et al. 2017 (in spirit of classical learning theory):

\[ \text{implicit regularization with depth 2 LNN (MF)} \Leftrightarrow \text{minimizing nuclear norm (surrogate for rank)} \]
Can the Implicit Regularization Be Captured by Norms?

Conjecture of Gunasekar et al. 2017 (in spirit of classical learning theory):

\[
\text{implicit regularization with depth 2 LNN (MF)} \quad \iff \quad \text{minimizing nuclear norm (surrogate for rank)}
\]

In light of our experiment, natural to hypothesize:

\[
\text{implicit regularization with deeper LNN (DMF)} \quad \iff \quad \text{minimizing other norm closer to rank}
\]
Case Study: Linear Neural Networks
Generalization

Can the Implicit Regularization Be Captured by Norms?

Conjecture of Gunasekar et al. 2017 (in spirit of classical learning theory):

\[
\text{implicit regularization with depth 2 LNN (MF)} \iff \text{minimizing nuclear norm (surrogate for rank)}
\]

In light of our experiment, natural to hypothesize:

\[
\text{implicit regularization with deeper LNN (DMF)} \iff \text{minimizing other norm closer to rank}
\]

Example
Can the Implicit Regularization Be Captured by Norms?

Conjecture of Gunasekar et al. 2017 (in spirit of classical learning theory):

\[
\text{implicit regularization with depth 2 LNN (MF)} \iff \text{minimizing nuclear norm (surrogate for rank)}
\]

In light of our experiment, natural to hypothesize:

\[
\text{implicit regularization with deeper LNN (DMF)} \iff \text{minimizing other norm closer to rank}
\]

**Example**

Schatten-\(p\) quasi-norm to the power of \(p\):

\[
\|W\|_{S_p}^p := \sum_r \sigma_r^p(W) \text{ where } \sigma_r(W) \text{ are singular vals of } W
\]
Can the Implicit Regularization Be Captured by Norms?

Conjecture of Gunasekar et al. 2017 (in spirit of classical learning theory):

\[
\text{implicit regularization with depth 2 LNN (MF)} \quad \iff \quad \text{minimizing nuclear norm (surrogate for rank)}
\]

In light of our experiment, natural to hypothesize:

\[
\text{implicit regularization with deeper LNN (DMF)} \quad \iff \quad \text{minimizing other norm closer to rank}
\]

Example

Schatten-\(p\) quasi-norm to the power of \(p\):

\[
\| W \|_{S_p}^p := \sum_r \sigma_r^p(W) \text{ where } \sigma_r(W) \text{ are singular vals of } W
\]

\(p = 1\): nuclear norm, corresponds to depth 2 by Gunasekar et al. 2017
Can the Implicit Regularization Be Captured by Norms?

Conjecture of Gunasekar et al. 2017 (in spirit of classical learning theory):

*implicit regularization* with depth 2 LNN (MF) $\iff$ *minimizing nuclear norm* (surrogate for rank)

In light of our experiment, natural to hypothesize:

*implicit regularization* with deeper LNN (DMF) $\iff$ *minimizing other norm* closer to rank

**Example**

Schatten-$p$ quasi-norm to the power of $p$:

- $\| W \|_{Sp}^p := \sum_r \sigma_r^p(W)$ where $\sigma_r(W)$ are singular vals of $W$
- $p = 1$: nuclear norm, corresponds to depth 2 by Gunasekar et al. 2017
- $0 < p < 1$: closer to rank, may correspond to higher depths
Current Theory is Oblivious to Depth

Theorem
In restricted setting where Gunasekar et al. proved depth $2$ minimizes nuclear norm, any depth $> 2$ does so as well.

Proposition
$\exists$ instances of this setting where nuclear norm minimization contradicts Schatten-p quasi-norm minimization (even locally)

$\forall p \in (0, 1)$

This implies:
implicit regularization for any depth $\not\equiv$ Schatten quasi-norm minimization

Instead, adopting lens of Gunasekar et al. leads to conjecturing:
implicit regularization for all depths $\equiv$ nuclear norm minimization

But our experiment shows depth changes implicit regularization!
Current Theory is Oblivious to Depth

Theorem

In restricted setting where Gunasekar et al. proved depth 2 minimizes nuclear norm, any depth > 2 does so as well
Current Theory is Oblivious to Depth

Theorem

In restricted setting where Gunasekar et al. proved depth 2 minimizes nuclear norm, any depth \( > 2 \) does so as well.

Proposition

\( \exists \) instances of this setting where nuclear norm minimization contradicts Schatten-\( p \) quasi-norm minimization (even locally) \( \forall p \in (0, 1) \)
Current Theory is Oblivious to Depth

**Theorem**

*In restricted setting where Gunasekar et al. proved depth 2 minimizes nuclear norm, any depth > 2 does so as well*

**Proposition**

∃ instances of this setting where nuclear norm minimization contradicts Schatten-p quasi-norm minimization (even locally) ∀p ∈ (0, 1)

This implies:

implicit regularization for any depth ≠ Schatten quasi-norm minimization
Current Theory is Oblivious to Depth

Theorem

In restricted setting where Gunasekar et al. proved depth 2 minimizes nuclear norm, any depth > 2 does so as well

Proposition

∃ instances of this setting where nuclear norm minimization contradicts Schatten-p quasi-norm minimization (even locally) ∀p ∈ (0, 1)

This implies:

implicit regularization for any depth ≠ Schatten quasi-norm minimization

Instead, adopting lens of Gunasekar et al. leads to conjecturing:

implicit regularization for all depths ≡ nuclear norm minimization
Current Theory is Oblivious to Depth

Theorem

In restricted setting where Gunasekar et al. proved depth 2 minimizes nuclear norm, any depth > 2 does so as well

Proposition

∃ instances of this setting where nuclear norm minimization contradicts Schatten-p quasi-norm minimization (even locally) ∀p ∈ (0, 1)

This implies:

implicit regularization for any depth \(\neq\) Schatten quasi-norm minimization

Instead, adopting lens of Gunasekar et al. leads to conjecturing:

implicit regularization for all depths \(\equiv\) nuclear norm minimization

But our experiment shows depth changes implicit regularization!
Experiments Testing Nuclear Norm Conjecture
Experiments Testing Nuclear Norm Conjecture

Setup:

- Completion of $100 \times 100$ rank 5 matrix
- Observed entries chosen uniformly at random
Experiments Testing Nuclear Norm Conjecture

Setup:

- Completion of $100 \times 100$ rank 5 matrix
- Observed entries chosen uniformly at random

Many (5K) Observations:

<table>
<thead>
<tr>
<th></th>
<th>reconst err</th>
<th>nuclear norm</th>
<th>effective rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>nuclear norm min</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>depth 2 LNN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>depth 3 LNN</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Experiments Testing Nuclear Norm Conjecture

Setup:
- Completion of $100 \times 100$ rank 5 matrix
- Observed entries chosen uniformly at random

Many (5K) Observations:

<table>
<thead>
<tr>
<th></th>
<th>reconst err</th>
<th>nuclear norm</th>
<th>effective rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>nuclear norm min</td>
<td>8 e-07</td>
<td>221</td>
<td>5</td>
</tr>
<tr>
<td>depth 2 LNN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>depth 3 LNN</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Nuclear norm min recovers ground truth
Experiments Testing Nuclear Norm Conjecture

Setup:
- Completion of $100 \times 100$ rank 5 matrix
- Observed entries chosen uniformly at random

Many (5K) Observations:

<table>
<thead>
<tr>
<th></th>
<th>reconst err</th>
<th>nuclear norm</th>
<th>effective rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>nuclear norm min</td>
<td>8 e-07</td>
<td>221</td>
<td>5</td>
</tr>
<tr>
<td>depth 2 LNN</td>
<td>5 e-06</td>
<td>221</td>
<td>5</td>
</tr>
<tr>
<td>depth 3 LNN</td>
<td>4 e-06</td>
<td>221</td>
<td>5</td>
</tr>
</tbody>
</table>

- Nuclear norm min recovers ground truth
- LNN do so too
Experiments Testing Nuclear Norm Conjecture

Setup:
- Completion of $100 \times 100$ rank 5 matrix
- Observed entries chosen uniformly at random

Many (5K) Observations:

<table>
<thead>
<tr>
<th></th>
<th>reconst err</th>
<th>nuclear norm</th>
<th>effective rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>nuclear norm min</td>
<td>$8 \times 10^{-7}$</td>
<td>221</td>
<td>5</td>
</tr>
<tr>
<td>depth 2 LNN</td>
<td>$5 \times 10^{-6}$</td>
<td>221</td>
<td>5</td>
</tr>
<tr>
<td>depth 3 LNN</td>
<td>$4 \times 10^{-6}$</td>
<td>221</td>
<td>5</td>
</tr>
</tbody>
</table>

- Nuclear norm min recovers ground truth
- LNN do so too
- Correspondence, but can’t distinguish between nuclear norm min and any bias leading to low rank
Experiments Testing Nuclear Norm Conjecture (cont’)

Few (2K) Observations:

<table>
<thead>
<tr>
<th></th>
<th>reconst err</th>
<th>nuclear norm</th>
<th>effective rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>nuclear norm min</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>depth 2 LNN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>depth 3 LNN</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Case Study: Linear Neural Networks

Generalization

Experiments Testing Nuclear Norm Conjecture (cont’)

Few (2K) Observations:

<table>
<thead>
<tr>
<th></th>
<th>reconst err</th>
<th>nuclear norm</th>
<th>effective rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>nuclear norm min</td>
<td>2 e -01</td>
<td>217</td>
<td>8</td>
</tr>
<tr>
<td>depth 2 LNN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>depth 3 LNN</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Nuclear norm min doesn’t recover ground truth
Experiments Testing Nuclear Norm Conjecture (cont’)

Few (2K) Observations:

<table>
<thead>
<tr>
<th></th>
<th>reconst err</th>
<th>nuclear norm</th>
<th>effective rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>nuclear norm min</td>
<td>2 e −01</td>
<td>217</td>
<td>8</td>
</tr>
<tr>
<td>depth 2 LNN</td>
<td>6 e −02</td>
<td>220</td>
<td>6</td>
</tr>
<tr>
<td>depth 3 LNN</td>
<td>3 e −05</td>
<td>221</td>
<td>5</td>
</tr>
</tbody>
</table>

- Nuclear norm min doesn’t recover ground truth
- LNN focus on lowering effective rank at expense of nuclear norm
Experiments Testing Nuclear Norm Conjecture (cont’)

Few (2K) Observations:

<table>
<thead>
<tr>
<th></th>
<th>reconst err</th>
<th>nuclear norm</th>
<th>effective rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>nuclear norm min</td>
<td>2 e-01</td>
<td>217</td>
<td>8</td>
</tr>
<tr>
<td>depth 2 LNN</td>
<td>6 e-02</td>
<td>220</td>
<td>6</td>
</tr>
<tr>
<td>depth 3 LNN</td>
<td>3 e-05</td>
<td>221</td>
<td>5</td>
</tr>
</tbody>
</table>

- Nuclear norm min doesn’t recover ground truth
- LNN focus on lowering effective rank at expense of nuclear norm
- Discrepancy!
Experiments Testing Nuclear Norm Conjecture (cont’)

Few (2K) Observations:

<table>
<thead>
<tr>
<th></th>
<th>reconst err</th>
<th>nuclear norm</th>
<th>effective rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>nuclear norm min</td>
<td>2e-01</td>
<td>217</td>
<td>8</td>
</tr>
<tr>
<td>depth 2 LNN</td>
<td>6e-02</td>
<td>220</td>
<td>6</td>
</tr>
<tr>
<td>depth 3 LNN</td>
<td>3e-05</td>
<td>221</td>
<td>5</td>
</tr>
</tbody>
</table>

- Nuclear norm min doesn’t recover ground truth
- LNN focus on lowering effective rank at expense of nuclear norm
- Discrepancy!

LNN implicitly minimize nuclear norm sometimes but not always!
Experiments Testing Nuclear Norm Conjecture (cont‘)

Few (2K) Observations:

<table>
<thead>
<tr>
<th></th>
<th>reconst err</th>
<th>nuclear norm</th>
<th>effective rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>nuclear norm min</td>
<td>$2 \times 10^{-1}$</td>
<td>217</td>
<td>8</td>
</tr>
<tr>
<td>depth 2 LNN</td>
<td>$6 \times 10^{-2}$</td>
<td>220</td>
<td>6</td>
</tr>
<tr>
<td>depth 3 LNN</td>
<td>$3 \times 10^{-5}$</td>
<td>221</td>
<td>5</td>
</tr>
</tbody>
</table>

- Nuclear norm min doesn’t recover ground truth
- LNN focus on lowering effective rank at expense of nuclear norm
- Discrepancy!

LNN implicitly minimize nuclear norm sometimes but not always!

Hypothesis

Single norm (or quasi-norm) not enough to capture implicit regularization, detailed account for trajectories is needed
Trajectory Analysis $\rightarrow$ Dynamics of Singular Values

$\text{Trajectory analysis gave dynamics for end-to-end matrix of } N\text{-layer LNN:}$

$$
\frac{d}{dt} \text{vec} \left[ W_1: \ldots : N \right](t) = - P W_1: \ldots : N(t) \cdot \text{vec} \left[ \nabla \ell (W_1: \ldots : N(t)) \right]
$$

Denote:

- $\{\sigma_r(t)\}$ — singular vals of $W_1: \ldots : N(t)$
- $\{u_r(t)\} / \{v_r(t)\}$ — corresponding left/right singular vecs

**Theorem**

$$
\frac{d}{dt} \sigma_r(t) = - N \cdot \sigma_r^2 - 2 N r(t) \cdot \left\langle \nabla \ell (W_1: \ldots : N(t)), u_r(t) v_r^\top(t) \right\rangle
$$

**Interpretation**

Given $W_1: \ldots : N(t)$, depth affects evolution only via factors $N \cdot \sigma_r^2 - 2 N r(t)$. For $N = 1$ (classic linear model): factors reduce to 1. For $N \geq 2$: factors speed up (slow down) large (small) singular vals, more so for larger $N$ (higher depth).
Trajectory Analysis  ➞ Dynamics of Singular Values

Trajectory analysis gave dynamics for end-to-end matrix of $N$-layer LNN:

$$\frac{d}{dt} \text{vec} [W_{1:N}(t)] = -P_{W_{1:N}(t)} \cdot \text{vec} [\nabla \ell (W_{1:N}(t))]$$
Trajectory Analysis ——> Dynamics of Singular Values

Trajectory analysis gave dynamics for end-to-end matrix of $N$-layer LNN:

$$\frac{d}{dt} \text{vec} [W_{1:N}(t)] = -P_{W_{1:N}(t)} \cdot \text{vec} [\nabla \ell(W_{1:N}(t))]$$

Denote:

- $\{\sigma_r(t)\}_r$ — singular vals of $W_{1:N}(t)$
- $\{u_r(t)\}_r / \{v_r(t)\}_r$ — corresponding left/right singular vecs
Trajectory Analysis ➞ Dynamics of Singular Values

Trajectory analysis gave dynamics for end-to-end matrix of \( N \)-layer LNN:

\[
\frac{d}{dt} \text{vec} \left[ W_{1:N}(t) \right] = -P_{W_{1:N}(t)} \cdot \text{vec} \left[ \nabla \ell( W_{1:N}(t) ) \right]
\]

Denote:

- \( \{ \sigma_r(t) \}_r \) — singular vals of \( W_{1:N}(t) \)
- \( \{ u_r(t) \}_r / \{ v_r(t) \}_r \) — corresponding left/right singular vecs

**Theorem**

\[
\frac{d}{dt} \sigma_r(t) = -N \cdot \sigma_r^{2 - \frac{2}{N}}(t) \cdot \left\langle \nabla \ell(W_{1:N}(t)), u_r(t) v_r^\top(t) \right\rangle
\]
Trajectory Analysis $\rightarrow$ Dynamics of Singular Values

Trajectory analysis gave dynamics for end-to-end matrix of $N$-layer LNN:

$$\frac{d}{dt} \text{vec} [W_{1:N}(t)] = -P_{W_{1:N}(t)} \cdot \text{vec} [\nabla \ell (W_{1:N}(t))]$$

Denote:
- $\{\sigma_r(t)\}_r$ — singular vals of $W_{1:N}(t)$
- $\{u_r(t)\}_r/\{v_r(t)\}_r$ — corresponding left/right singular vecs

**Theorem**

$$\frac{d}{dt} \sigma_r(t) = -N \cdot \sigma_r^{2 - \frac{2}{N}}(t) \cdot \left\langle \nabla \ell (W_{1:N}(t)), u_r(t)v_r^\top(t) \right\rangle$$

**Interpretation**
- Given $W_{1:N}(t)$, depth affects evolution only via factors $N \cdot \sigma_r^{2 - \frac{2}{N}}(t)$

Nadav Cohen (TAU)
Trajectory Analysis ➔ Dynamics of Singular Values

Trajectory analysis gave dynamics for end-to-end matrix of $N$-layer LNN:

$$\frac{d}{dt} \text{vec} \left[ W_{1:N}(t) \right] = -P_{W_{1:N}(t)} \cdot \text{vec} \left[ \nabla \ell (W_{1:N}(t)) \right]$$

Denote:
- $\{\sigma_r(t)\}_r$ — singular vals of $W_{1:N}(t)$
- $\{u_r(t)\}_r/\{v_r(t)\}_r$ — corresponding left/right singular vecs

**Theorem**

$$\frac{d}{dt} \sigma_r(t) = -N \cdot \sigma_r^{2-\frac{2}{N}}(t) \cdot \left< \nabla \ell (W_{1:N}(t)), u_r(t)v_r^T(t) \right>$$

**Interpretation**
- Given $W_{1:N}(t)$, depth affects evolution only via factors $N \cdot \sigma_r^{2-\frac{2}{N}}(t)$
- $N = 1$ (classic linear model): factors reduce to 1
Trajectory Analysis → Dynamics of Singular Values

Trajectory analysis gave dynamics for end-to-end matrix of $N$-layer LNN:

$$\frac{d}{dt} \text{vec } [W_{1:N}(t)] = -P_{W_{1:N}(t)} \cdot \text{vec } [\nabla \ell (W_{1:N}(t))]$$

Denote:

- $\{\sigma_r(t)\}_r$ — singular vals of $W_{1:N}(t)$
- $\{u_r(t)\}_r / \{v_r(t)\}_r$ — corresponding left/right singular vecs

**Theorem**

$$\frac{d}{dt} \sigma_r(t) = -N \cdot \sigma_r^{\frac{2}{N}}(t) \cdot \left\langle \nabla \ell (W_{1:N}(t)), u_r(t)v_r^T(t) \right\rangle$$

**Interpretation**

- Given $W_{1:N}(t)$, depth affects evolution only via factors $N \cdot \sigma_r^{\frac{2}{N}}(t)$
- $N = 1$ (classic linear model): factors reduce to 1
- $N \geq 2$: factors speed up (slow down) large (small) singular vals, more so for larger $N$ (higher depth)
Dynamics of Singular Values — Proof Sketch

**Theorem**

\[
\frac{d}{dt} \sigma_r(t) = -N \cdot \sigma_r^{2-\frac{2}{N}}(t) \cdot \langle \nabla \ell(W_{1:N}(t)), u_r(t)v_r^T(t) \rangle
\]

**Proof Sketch**
Dynamics of Singular Values — Proof Sketch

Theorem
\[
\frac{d}{dt} \sigma_r(t) = -N \cdot \sigma_r^{2 - \frac{2}{N}}(t) \cdot \langle \nabla \ell(W_{1:N}(t)), u_r(t)v_r^T(t) \rangle
\]

Proof Sketch
SVD: \( W_{1:N}(t) = U(t)S(t)V(t)^T \) \( (S = \text{diag}(\sigma_1, \sigma_2, \ldots) \quad U = [u_1, u_2, \ldots] \quad V = [v_1, v_2, \ldots]) \)
Dynamics of Singular Values — Proof Sketch

**Theorem**

\[
\frac{d}{dt} \sigma_r(t) = -N \cdot \sigma_r^{2 - \frac{2}{N}}(t) \cdot \langle \nabla \ell(W_{1:N}(t)), u_r(t)v_r^T(t) \rangle
\]

**Proof Sketch**

SVD: \( W_{1:N}(t) = U(t)S(t)V(t)^T \) \( (S = \text{diag}(\sigma_1, \sigma_2, \ldots) \quad U = [u_1, u_2, \ldots] \quad V = [v_1, v_2, \ldots]) \)

\[\Rightarrow \frac{d}{dt} W_{1:N}(t) = \frac{d}{dt} U(t) \cdot S(t) \cdot V(t)^T + U(t) \cdot \frac{d}{dt} S(t) \cdot V(t)^T + U(t) \cdot S(t) \cdot \frac{d}{dt} V(t)^T \]
Dynamics of Singular Values — Proof Sketch

**Theorem**

\[ \frac{d}{dt} \sigma_r(t) = -N \cdot \sigma_r \frac{2}{N} (t) \cdot \langle \nabla \ell(W_{1:N}(t)), u_r(t)v_r^T (t) \rangle \]

**Proof Sketch**

SVD: \( W_{1:N}(t) = U(t)S(t)V(t)^T \)  \( (S = diag(\sigma_1, \sigma_2, ...) \quad U = [u_1, u_2, ...] \quad V = [v_1, v_2, ...] ) \)

\[ \Rightarrow \frac{d}{dt} W_{1:N}(t) = \frac{d}{dt} U(t) \cdot S(t) \cdot V(t)^T + U(t) \cdot \frac{d}{dt} S(t) \cdot V(t)^T + U(t) \cdot S(t) \cdot \frac{d}{dt} V(t)^T \]

\[ \Rightarrow U(t)^T \cdot \frac{d}{dt} W_{1:N}(t) \cdot V(t) = U(t)^T \cdot \frac{d}{dt} U(t) \cdot S(t) + \frac{d}{dt} S(t) + S(t) \cdot \frac{d}{dt} V(t)^T \cdot V(t) \]
Dynamics of Singular Values — Proof Sketch

**Theorem**

\[
\frac{d}{dt} \sigma_r(t) = -N \cdot \sigma_r^2(t) - \frac{2}{N} \langle \nabla \ell(W_{1:N}(t)), u_r(t)v_r^T(t) \rangle
\]

**Proof Sketch**

SVD: \( W_{1:N}(t) = U(t)S(t)V(t)^\top \) \( (S = \text{diag}(\sigma_1, \sigma_2, \ldots) \quad U = [u_1, u_2, \ldots] \quad V = [v_1, v_2, \ldots]) \)

\[
\Rightarrow \frac{d}{dt} W_{1:N}(t) = \frac{d}{dt} U(t) \cdot S(t) \cdot V(t)^\top + U(t) \cdot \frac{d}{dt} S(t) \cdot V(t)^\top + U(t) \cdot S(t) \cdot \frac{d}{dt} V(t)^\top
\]

\[
\Rightarrow U(t)^\top \cdot \frac{d}{dt} W_{1:N}(t) \cdot V(t) = U(t)^\top \cdot \frac{d}{dt} U(t) \cdot S(t) + \frac{d}{dt} S(t) + S(t) \cdot \frac{d}{dt} V(t)^\top \cdot V(t)
\]

End-to-end dynamics:

\[
\frac{d}{dt} W_{1:N}(t) = -\sum_{j=1}^{N} \left[ W_{1:N}(t) W_{1:N}(t)^\top \right]_{N-j}^{N} \cdot \nabla \ell(W_{1:N}(t)) \cdot \left[ W_{1:N}(t)^\top W_{1:N}(t) \right]_{j-1}^{N}^\frac{1}{N}
\]
Dynamics of Singular Values — Proof Sketch

**Theorem**

\[
\frac{d}{dt} \sigma_r(t) = -N \cdot \sigma_r^{\frac{2}{N}}(t) \cdot \langle \nabla \ell(W_{1:N}(t)), u_r(t)v_r^T(t) \rangle
\]

**Proof Sketch**

**SVD:** \(W_{1:N}(t) = U(t)S(t)V(t)^T\)  \(S = \text{diag}(\sigma_1, \sigma_2, \ldots)\)  \(U = [u_1, u_2, \ldots]\)  \(V = [v_1, v_2, \ldots]\)

\[
\Rightarrow \quad \frac{d}{dt} W_{1:N}(t) = \frac{d}{dt} U(t) \cdot S(t) \cdot V(t)^T + U(t) \cdot \frac{d}{dt} S(t) \cdot V(t)^T + U(t) \cdot S(t) \cdot \frac{d}{dt} V(t)^T
\]

\[
\Rightarrow \quad U(t)^T \cdot \frac{d}{dt} W_{1:N}(t) \cdot V(t) = U(t)^T \cdot \frac{d}{dt} U(t) \cdot S(t) + \frac{d}{dt} S(t) + S(t) \cdot \frac{d}{dt} V(t)^T \cdot V(t)
\]

**End-to-end dynamics:**

\[
\frac{d}{dt} W_{1:N}(t) = -\sum_{j=1}^{N} \left[ W_{1:N}(t)W_{1:N}(t)^T \right]^{\frac{N-j}{N}} \cdot \nabla \ell(W_{1:N}(t)) \cdot \left[ W_{1:N}(t)^T W_{1:N}(t) \right]^{\frac{j-1}{N}}
\]
Dynamics of Singular Values — Proof Sketch

**Theorem**

\[ \frac{d}{dt} \sigma_r(t) = -N \cdot \sigma_r^{2-\frac{2}{N}}(t) \cdot \langle \nabla \ell(W_{1:N}(t)), u_r(t)v_r^T(t) \rangle \]

**Proof Sketch**

SVD: \( W_{1:N}(t) = U(t)S(t)V(t)^T \) \( (S = \text{diag}(\sigma_1, \sigma_2, \ldots) \quad U = [u_1, u_2, \ldots] \quad V = [v_1, v_2, \ldots]) \)

\[ \Rightarrow \frac{d}{dt} W_{1:N}(t) = \frac{d}{dt} U(t) \cdot S(t) \cdot V(t)^T + U(t) \cdot \frac{d}{dt} S(t) \cdot V(t)^T + U(t) \cdot S(t) \cdot \frac{d}{dt} V(t)^T \]

\[ \Rightarrow U(t)^T \cdot \frac{d}{dt} W_{1:N}(t) \cdot V(t) = U(t)^T \cdot \frac{d}{dt} U(t) \cdot S(t) + \frac{d}{dt} S(t) + S(t) \cdot \frac{d}{dt} V(t)^T \cdot V(t) \]

End-to-end dynamics:

\[ \frac{d}{dt} W_{1:N}(t) = - \sum_{j=1}^{N} U(t) \left[ S(t)S(t)^T \right]^{N-j} U(t)^T \cdot \nabla \ell(W_{1:N}(t)) \cdot V(t) \left[ S(t)^T S(t) \right]^{j-1} V(t)^T \]
Dynamics of Singular Values — Proof Sketch

**Theorem**

\[
\frac{d}{dt} \sigma_r(t) = -N \cdot \sigma_r^2 \frac{2}{N}(t) \cdot \langle \nabla \ell(W_{1:N}(t)), u_r(t)v_r^T(t) \rangle
\]

**Proof Sketch**

**SVD:**

\[
W_{1:N}(t) = U(t)S(t)V(t)^T \quad (S = \text{diag}(\sigma_1, \sigma_2, \ldots) \quad U = [u_1, u_2, \ldots] \quad V = [v_1, v_2, \ldots])
\]

\[
\Rightarrow \frac{d}{dt} W_{1:N}(t) = \frac{d}{dt} U(t) \cdot S(t) \cdot V(t)^T + U(t) \cdot \frac{d}{dt} S(t) \cdot V(t)^T + U(t) \cdot S(t) \cdot \frac{d}{dt} V(t)^T
\]

\[
\Rightarrow U(t)^T \cdot \frac{d}{dt} W_{1:N}(t) \cdot V(t) = U(t)^T \cdot \frac{d}{dt} U(t) \cdot S(t) + \frac{d}{dt} S(t) + S(t) \cdot \frac{d}{dt} V(t)^T \cdot V(t)
\]

**End-to-end dynamics:**

\[
\frac{d}{dt} W_{1:N}(t) = -\sum_{j=1}^{N} U(t) \left[ S(t)S(t)^T \right]^{\frac{N-j}{N}} U(t)^T \cdot \nabla \ell(W_{1:N}(t)) \cdot V(t) \left[ S(t)^T S(t) \right]^{\frac{j-1}{N}} V(t)^T
\]

\[
\Rightarrow U(t)^T \cdot \frac{d}{dt} W_{1:N}(t) \cdot V(t)
\]

\[
= -\sum_{j=1}^{N} \left[ S(t)S(t)^T \right]^{\frac{N-j}{N}} U(t)^T \cdot \nabla \ell(W_{1:N}(t)) \cdot V(t) \left[ S(t)^T S(t) \right]^{\frac{j-1}{N}}
\]
Dynamics of Singular Values — Proof Sketch

**Theorem**

$$\frac{d}{dt} \sigma_r(t) = -N \cdot \sigma_r^2 \frac{2}{N}(t) \cdot \langle \nabla \ell(W_{1:N}(t)), u_r(t)v_r^T(t) \rangle$$

**Proof Sketch**

**SVD:**

$$W_{1:N}(t) = U(t)S(t)V(t)^\top \quad (S = \text{diag}(\sigma_1, \sigma_2, \ldots) \quad U = [u_1, u_2, \ldots] \quad V = [v_1, v_2, \ldots])$$

$$\implies \frac{d}{dt} W_{1:N}(t) = \frac{d}{dt} U(t) \cdot S(t) \cdot V(t)^\top + U(t) \cdot \frac{d}{dt} S(t) \cdot V(t)^\top + U(t) \cdot S(t) \cdot \frac{d}{dt} V(t)^\top$$

$$\implies U(t)^\top \cdot \frac{d}{dt} W_{1:N}(t) \cdot V(t) = U(t)^\top \cdot \frac{d}{dt} U(t) \cdot S(t) + \frac{d}{dt} S(t) + S(t) \cdot \frac{d}{dt} V(t)^\top \cdot V(t)$$

**End-to-end dynamics:**

$$\frac{d}{dt} W_{1:N}(t) = -\sum_{j=1}^N U(t) \left[ S(t)S(t)^\top \right]^{N-j} \frac{N}{N} U(t)^\top \cdot \nabla \ell(W_{1:N}(t)) \cdot V(t) \left[ S(t)^\top S(t) \right]^{j-1} \frac{N}{N} V(t)^\top$$

$$\implies U(t)^\top \cdot \frac{d}{dt} W_{1:N}(t) \cdot V(t)$$

$$= -\sum_{j=1}^N \left[ S(t)S(t)^\top \right]^{N-j} \frac{N}{N} U(t)^\top \cdot \nabla \ell(W_{1:N}(t)) \cdot V(t) \left[ S(t)^\top S(t) \right]^{j-1} \frac{N}{N}$$
Dynamics of Singular Values — Proof Sketch

Theorem

\[
\frac{d}{dt}\sigma_r(t) = -N \cdot \sigma_r^2 \cdot \frac{2}{N}(t) \cdot \langle \nabla \ell(W_{1:N}(t)), u_r(t)v_r^T(t) \rangle
\]

Proof Sketch

SVD: \( W_{1:N}(t) = U(t)S(t)V(t)^T \) \( S = diag(\sigma_1, \sigma_2, \ldots) \) \( U = [u_1, u_2, \ldots] \) \( V = [v_1, v_2, \ldots] \)

\[\Rightarrow \frac{d}{dt} W_{1:N}(t) = \frac{d}{dt} U(t) \cdot S(t) \cdot V(t)^T + U(t) \cdot \frac{d}{dt} S(t) \cdot V(t)^T + U(t) \cdot S(t) \cdot \frac{d}{dt} V(t)^T \]

\[\Rightarrow U(t)^T \cdot \frac{d}{dt} W_{1:N}(t) \cdot V(t) = U(t)^T \cdot \frac{d}{dt} U(t) \cdot S(t) + \frac{d}{dt} S(t) + S(t) \cdot \frac{d}{dt} V(t)^T \cdot V(t) \]

End-to-end dynamics:

\[\frac{d}{dt} W_{1:N}(t) = -\sum_{j=1}^{N} U(t) \left [ S(t)S(t)^T \right ]^{\frac{N-j}{N}} U(t)^T \cdot \nabla \ell(W_{1:N}(t)) \cdot V(t) \left [ S(t)^T S(t) \right ]^{\frac{i-1}{N}} V(t)^T \]

\[\Rightarrow U(t)^T \cdot \frac{d}{dt} U(t) \cdot S(t) + \frac{d}{dt} S(t) + S(t) \cdot \frac{d}{dt} V(t)^T \cdot V(t) \]

\[= -\sum_{j=1}^{N} \left [ S(t)S(t)^T \right ]^{\frac{N-j}{N}} U(t)^T \cdot \nabla \ell(W_{1:N}(t)) \cdot V(t) \left [ S(t)^T S(t) \right ]^{\frac{i-1}{N}} V(t)^T \]
Dynamics of Singular Values — Proof Sketch

**Theorem**

\[
\frac{d}{dt} \sigma_r(t) = -N \cdot \sigma_r^2 \frac{2}{N}(t) \cdot \langle \nabla \ell(W_{1:N}(t)), u_r(t)v_r^\top(t) \rangle
\]

**Proof Sketch**

**SVD:**

\[
W_{1:N}(t) = U(t)S(t)V(t)^\top \quad (S = \text{diag}(\sigma_1, \sigma_2, \ldots) \quad U = [u_1, u_2, \ldots] \quad V = [v_1, v_2, \ldots])
\]

\[
\Rightarrow \quad \frac{d}{dt} W_{1:N}(t) = \frac{d}{dt} U(t) \cdot S(t) \cdot V(t)^\top + U(t) \cdot \frac{d}{dt} S(t) \cdot V(t)^\top + U(t) \cdot S(t) \cdot \frac{d}{dt} V(t)^\top
\]

\[
\Rightarrow \quad U(t)^\top \cdot \frac{d}{dt} W_{1:N}(t) \cdot V(t) = U(t)^\top \cdot \frac{d}{dt} U(t) \cdot S(t) + \frac{d}{dt} S(t) + S(t) \cdot \frac{d}{dt} V(t)^\top \cdot V(t)
\]

**End-to-end dynamics:**

\[
\frac{d}{dt} W_{1:N}(t) = -\sum_{j=1}^N U(t) \left[ S(t)S(t)^\top \right]^{\frac{N-j}{N}} U(t)^\top \cdot \nabla \ell(W_{1:N}(t)) \cdot V(t) \left[ S(t)^\top S(t) \right]^{\frac{j-1}{N}} V(t)^\top
\]

\[
\Rightarrow \quad U(t)^\top \cdot \frac{d}{dt} U(t) \cdot S(t) + \frac{d}{dt} S(t) + S(t) \cdot \frac{d}{dt} V(t)^\top \cdot V(t)
\]

\[
= -\sum_{j=1}^N \left[ S(t)S(t)^\top \right]^{\frac{N-j}{N}} U(t)^\top \cdot \nabla \ell(W_{1:N}(t)) \cdot V(t) \left[ S(t)^\top S(t) \right]^{\frac{j-1}{N}}
\]

Restrict attention to \(r\)'th diagonal element:
Dynamics of Singular Values — Proof Sketch

**Theorem**

\[
\frac{d}{dt} \sigma_r(t) = -N \cdot \sigma_r^{-\frac{2}{N}}(t) \cdot \langle \nabla \ell(W_{1:N}(t)), u_r(t)v_r^\top(t) \rangle
\]

**Proof Sketch**

**SVD:**

\[ W_{1:N}(t) = U(t)S(t)V(t)^\top \quad (S = \text{diag}(\sigma_1, \sigma_2, \ldots) \quad U = [u_1, u_2, \ldots] \quad V = [v_1, v_2, \ldots]) \]

\[ \Rightarrow \frac{d}{dt} W_{1:N}(t) = \frac{d}{dt} U(t) \cdot S(t) \cdot V(t)^\top + U(t) \cdot \frac{d}{dt} S(t) \cdot V(t)^\top + U(t) \cdot S(t) \cdot \frac{d}{dt} V(t)^\top \]

\[ \Rightarrow U(t)^\top \cdot \frac{d}{dt} W_{1:N}(t) \cdot V(t) = U(t)^\top \cdot \frac{d}{dt} U(t) \cdot S(t) + \frac{d}{dt} S(t) + S(t) \cdot \frac{d}{dt} V(t)^\top \cdot V(t) \]

**End-to-end dynamics:**

\[ \frac{d}{dt} W_{1:N}(t) = -\sum_{j=1}^{N} U(t) \left[ S(t)S(t)^\top \right]^{\frac{N-j}{N}} U(t)^\top \cdot \nabla \ell(W_{1:N}(t)) \cdot V(t) \left[ S(t)^\top S(t) \right]^{\frac{j-1}{N}} V(t)^\top \]

\[ \Rightarrow U(t)^\top \cdot \frac{d}{dt} U(t) \cdot S(t) + \frac{d}{dt} S(t) + S(t) \cdot \frac{d}{dt} V(t)^\top \cdot V(t) \]

\[ = -\sum_{j=1}^{N} \left[ S(t)S(t)^\top \right]^{\frac{N-j}{N}} U(t)^\top \cdot \nabla \ell(W_{1:N}(t)) \cdot V(t) \left[ S(t)^\top S(t) \right]^{\frac{j-1}{N}} \]

**Restrict attention to r’th diagonal element:**

\[ u_r(t)^\top \cdot \frac{d}{dt} u_r(t) \cdot \sigma_r(t) + \frac{d}{dt} \sigma_r(t) + \sigma_r(t) \cdot \frac{d}{dt} v_r(t)^\top \cdot v_r(t) = \]

\[ -\sum_{j=1}^{N} \sigma_r^{\frac{N-j}{N}}(t) \cdot u_r(t)^\top \cdot \nabla \ell(W_{1:N}(t)) \cdot v_r(t) \cdot \sigma_r^{\frac{j-1}{N}}(t) \]
Dynamics of Singular Values — Proof Sketch

**Theorem**

\[
\frac{d}{dt} \sigma_r(t) = -N \cdot \sigma_r \left( \frac{2}{N} \right) \cdot \langle \nabla \ell(W_{1:N}(t)) , u_r(t)v_r^T(t) \rangle
\]

**Proof Sketch**

**SVD:**

\[
W_{1:N}(t) = U(t)S(t)V(t)^T \quad (S = \text{diag}(\sigma_1, \sigma_2, ...) \quad U = [u_1, u_2, ...] \quad V = [v_1, v_2, ...])
\]

\[
\Rightarrow \frac{d}{dt} W_{1:N}(t) = \frac{d}{dt} U(t) \cdot S(t) \cdot V(t)^T + U(t) \cdot \frac{d}{dt} S(t) \cdot V(t)^T + U(t) \cdot S(t) \cdot \frac{d}{dt} V(t)^T
\]

\[
\Rightarrow U(t)^T \cdot \frac{d}{dt} W_{1:N}(t) \cdot V(t) = U(t)^T \cdot \frac{d}{dt} U(t) \cdot S(t) + \frac{d}{dt} S(t) + S(t) \cdot \frac{d}{dt} V(t)^T \cdot V(t)
\]

**End-to-end dynamics:**

\[
\frac{d}{dt} W_{1:N}(t) = -\sum_{j=1}^{N} U(t) \left[ S(t)S(t)^T \right]^{\frac{N-j}{N}} U(t)^T \cdot \nabla \ell(W_{1:N}(t)) \cdot V(t) \left[ S(t)^T S(t) \right]^{\frac{j-1}{N}} V(t)^T
\]

\[
\Rightarrow U(t)^T \cdot \frac{d}{dt} U(t) \cdot S(t) + \frac{d}{dt} S(t) + S(t) \cdot \frac{d}{dt} V(t)^T \cdot V(t)
\]

\[
= -\sum_{j=1}^{N} \left[ S(t)S(t)^T \right]^{\frac{N-j}{N}} U(t)^T \cdot \nabla \ell(W_{1:N}(t)) \cdot V(t) \left[ S(t)^T S(t) \right]^{\frac{j-1}{N}}
\]

**Restrict attention to r’th diagonal element:**

\[
u_r(t)^T \cdot \frac{d}{dt} u_r(t) \cdot \sigma_r(t) + \frac{d}{dt} \sigma_r(t) + \sigma_r(t) \cdot \frac{d}{dt} v_r(t)^T \cdot v_r(t) =
\]

\[
-\sum_{j=1}^{N} \sigma_r \left( \frac{2}{N} \right) \cdot u_r(t)^T \cdot \nabla \ell(W_{1:N}(t)) \cdot v_r(t)
\]
Dynamics of Singular Values — Proof Sketch

**Theorem**

\[
\frac{d}{dt} \sigma_r(t) = -N \cdot \sigma_r \frac{2}{N} (t) \cdot \langle \nabla \ell(W_{1:N}(t)) , u_r(t)v_r^T(t) \rangle
\]

**Proof Sketch**

SVD: \( W_{1:N}(t) = U(t)S(t)V(t)^T \) \( S = \text{diag}(\sigma_1, \sigma_2, ...) \) \( U = [u_1, u_2, ...] \) \( V = [v_1, v_2, ...] \)

\[
\frac{d}{dt} W_{1:N}(t) = \frac{d}{dt} U(t) \cdot S(t) \cdot V(t)^T + U(t) \cdot \frac{d}{dt} S(t) \cdot V(t)^T + U(t) \cdot S(t) \cdot \frac{d}{dt} V(t)^T
\]

\[
\Rightarrow U(t)^T \cdot \frac{d}{dt} W_{1:N}(t) \cdot V(t) = U(t)^T \cdot \frac{d}{dt} U(t) \cdot S(t) + \frac{d}{dt} S(t) + S(t) \cdot \frac{d}{dt} V(t)^T \cdot V(t)
\]

End-to-end dynamics:

\[
\frac{d}{dt} W_{1:N}(t) = -N \sum_{j=1}^{N} U(t) \left[ S(t)S(t)^T \right]^{\frac{N-j}{N}} U(t)^T \cdot \nabla \ell(W_{1:N}(t)) \cdot V(t) \left[ S(t)^T S(t) \right]^{\frac{i-1}{N}} V(t)^T
\]

\[
\Rightarrow U(t)^T \cdot \frac{d}{dt} U(t) \cdot S(t) + \frac{d}{dt} S(t) + S(t) \cdot \frac{d}{dt} V(t)^T \cdot V(t)
\]

\[
= -N \sum_{j=1}^{N} U(t) \left[ S(t)S(t)^T \right]^{\frac{N-j}{N}} U(t)^T \cdot \nabla \ell(W_{1:N}(t)) \cdot V(t) \left[ S(t)^T S(t) \right]^{\frac{i-1}{N}} V(t)^T
\]

Restrict attention to \( r \)'th diagonal element:

\[
u_r(t)^T \cdot \frac{d}{dt} u_r(t) \cdot \sigma_r(t) + \frac{d}{dt} \sigma_r(t) + \sigma_r(t) \cdot \frac{d}{dt} v_r(t)^T \cdot v_r(t) =
\]

\[
- N \cdot \sigma_r \frac{2}{N} (t) \cdot u_r(t)^T \cdot \nabla \ell(W_{1:N}(t)) \cdot v_r(t)
\]
Case Study: Linear Neural Networks

Generalization

Dynamics of Singular Values — Proof Sketch

**Theorem**

\[
\frac{d}{dt} \sigma_r(t) = -N \cdot \sigma_r^{\frac{2}{N}}(t) \cdot \langle \nabla \ell(W_{1:N}(t)), u_r(t)v_r^T(t) \rangle
\]

**Proof Sketch**

**SVD:**

\[
W_{1:N}(t) = U(t)S(t)V(t)^T \quad (S = \text{diag}(\sigma_1, \sigma_2, \ldots) \quad U = [u_1, u_2, \ldots] \quad V = [v_1, v_2, \ldots])
\]

\[
\implies \frac{d}{dt} W_{1:N}(t) = \frac{d}{dt} U(t) \cdot S(t) \cdot V(t)^T + U(t) \cdot \frac{d}{dt} S(t) \cdot V(t)^T + U(t) \cdot S(t) \cdot \frac{d}{dt} V(t)^T
\]

\[
\implies U(t)^T \cdot \frac{d}{dt} W_{1:N}(t) \cdot V(t) = U(t)^T \cdot \frac{d}{dt} U(t) \cdot S(t) + \frac{d}{dt} S(t) + S(t) \cdot \frac{d}{dt} V(t)^T \cdot V(t)
\]

**End-to-end dynamics:**

\[
\frac{d}{dt} W_{1:N}(t) = -\sum_{j=1}^N U(t)^T \left[ S(t)S(t)^T \right]^{\frac{N-j}{N}} \cdot \frac{d}{dt} U(t) \cdot \nabla \ell(W_{1:N}(t)) \cdot V(t) \cdot S(t)^T S(t) \left[ S(t)^T S(t) \right]^{\frac{j-1}{N}} \cdot V(t)^T
\]

\[
\implies U(t)^T \cdot \frac{d}{dt} U(t) \cdot S(t) + \frac{d}{dt} S(t) + S(t) \cdot \frac{d}{dt} V(t)^T \cdot V(t)
\]

\[
= -\sum_{j=1}^N U(t)^T \left[ S(t)S(t)^T \right]^{\frac{N-j}{N}} \cdot \frac{d}{dt} U(t) \cdot \nabla \ell(W_{1:N}(t)) \cdot V(t) \cdot S(t)^T S(t) \left[ S(t)^T S(t) \right]^{\frac{j-1}{N}}
\]

**Restrict attention to r'th diagonal element:**

\[
u_r(t)^T \cdot \frac{d}{dt} u_r(t) \cdot \sigma_r(t) + \frac{d}{dt} \sigma_r(t) + \sigma_r(t) \cdot \frac{d}{dt} v_r(t)^T \cdot v_r(t) =
\]

\[
-\sum_{j=1}^{N-1} \sigma_r^{\frac{2(N-j)}{N}}(t) \cdot \langle \nabla \ell(W_{1:N}(t)), u_r(t)v_r^T(t) \rangle
\]
Dynamics of Singular Values — Proof Sketch

Theorem

\[
\frac{d}{dt} \sigma_r(t) = -N \cdot \sigma_r^2 \frac{2}{N} (t) \cdot \langle \nabla \ell(W_{1:N}(t)), u_r(t)v_r^T(t) \rangle
\]

Proof Sketch

SVD: \( W_{1:N}(t) = U(t)S(t)V(t)^T \) \( S = \text{diag}(\sigma_1, \sigma_2, \ldots) \) \( U = [u_1, u_2, \ldots] \) \( V = [v_1, v_2, \ldots] \)

\[
\frac{d}{dt} W_{1:N}(t) = \frac{d}{dt} U(t) \cdot S(t) \cdot V(t)^T + U(t) \cdot \frac{d}{dt} S(t) \cdot V(t)^T + U(t) \cdot S(t) \cdot \frac{d}{dt} V(t)^T
\]

\[
\Rightarrow U(t)^T \cdot \frac{d}{dt} W_{1:N}(t) \cdot V(t) = U(t)^T \cdot \frac{d}{dt} U(t) \cdot S(t) + \frac{d}{dt} S(t) + S(t) \cdot \frac{d}{dt} V(t)^T \cdot V(t)
\]

End-to-end dynamics:

\[
\frac{d}{dt} W_{1:N}(t) = -\sum_{j=1}^{N} U(t) \left[ S(t)S(t)^T \right] \frac{N-j}{N} U(t)^T \cdot \nabla \ell(W_{1:N}(t)) \cdot V(t) \left[ S(t)^T S(t) \right] \frac{i-1}{N} V(t)^T
\]

\[
\Rightarrow U(t)^T \cdot \frac{d}{dt} U(t) \cdot S(t) + \frac{d}{dt} S(t) + S(t) \cdot \frac{d}{dt} V(t)^T \cdot V(t)
\]

\[
= -\sum_{j=1}^{N} \left[ S(t)S(t)^T \right] \frac{N-j}{N} U(t)^T \cdot \nabla \ell(W_{1:N}(t)) \cdot V(t) \left[ S(t)^T S(t) \right] \frac{i-1}{N}
\]

Restrict attention to \( r \)'th diagonal element:

\[
\begin{align*}
\langle u_r(t)^T \cdot \frac{d}{dt} u_r(t) \cdot \sigma_r(t) + \frac{d}{dt} \sigma_r(t) + \sigma_r(t) \cdot \frac{d}{dt} v_r(t)^T \cdot v_r(t) = & \\
& -N \cdot \sigma_r^2 \frac{2}{N} (t) \cdot \langle \nabla \ell(W_{1:N}(t)), u_r(t)v_r^T(t) \rangle
\end{align*}
\]
Theorem

\[
\frac{d}{dt} \sigma_r(t) = -\mathbf{N} \cdot \sigma_r^{\frac{2}{N}}(t) \cdot \langle \nabla \ell(W_{1:N}(t)), \mathbf{u}_r(t)\mathbf{v}_r^\top(t) \rangle
\]

Proof Sketch

SVD: \( W_{1:N}(t) = U(t)S(t)V(t)^\top \) \( (S = \text{diag}(\sigma_1, \sigma_2, \ldots) \quad U = [\mathbf{u}_1, \mathbf{u}_2, \ldots] \quad V = [\mathbf{v}_1, \mathbf{v}_2, \ldots]) \)

\[ \frac{d}{dt} W_{1:N}(t) = \frac{d}{dt} U(t) \cdot S(t) \cdot V(t)^\top + U(t) \cdot \frac{d}{dt} S(t) \cdot V(t)^\top + U(t) \cdot S(t) \cdot \frac{d}{dt} V(t)^\top \]

\[ \frac{d}{dt} W_{1:N}(t) \cdot V(t) = U(t)^\top \cdot \frac{d}{dt} U(t) \cdot S(t) + \frac{d}{dt} S(t) + S(t) \cdot \frac{d}{dt} V(t)^\top \cdot V(t) \]

End-to-end dynamics:

\[ \frac{d}{dt} W_{1:N}(t) = -\sum_{j=1}^{N} U(t) \left[ S(t)S(t)^\top \right]^{\frac{N-j}{N}} U(t)^\top \cdot \nabla \ell(W_{1:N}(t)) \cdot V(t) \left[ S(t)^\top S(t) \right]^{\frac{j-1}{N}} V(t)^\top \]

\[ \frac{d}{dt} W_{1:N}(t) \cdot V(t) = U(t)^\top \cdot \frac{d}{dt} U(t) \cdot S(t) + \frac{d}{dt} S(t) + S(t) \cdot \frac{d}{dt} V(t)^\top \cdot V(t) \]

\[ = -\sum_{j=1}^{N} \left[ S(t)S(t)^\top \right]^{\frac{N-j}{N}} U(t)^\top \cdot \nabla \ell(W_{1:N}(t)) \cdot V(t) \left[ S(t)^\top S(t) \right]^{\frac{j-1}{N}} \]

Restrict attention to \( r \)'th diagonal element:

\[ \frac{1}{2} \frac{d}{dt} \| \mathbf{u}_r(t) \|_2^2 \cdot \sigma_r(t) + \frac{d}{dt} \sigma_r(t) + \sigma_r(t) \cdot \frac{1}{2} \frac{d}{dt} \| \mathbf{v}_r(t) \|_2^2 = -\mathbf{N} \cdot \sigma_r^{\frac{2(N-1)}{N}}(t) \cdot \langle \nabla \ell(W_{1:N}(t)), \mathbf{u}_r(t)\mathbf{v}_r^\top(t) \rangle \]
Dynamics of Singular Values — Proof Sketch

**Theorem**

\[
\frac{d}{dt} \sigma_r(t) = -N \cdot \sigma_r^{2 - \frac{2}{N}}(t) \cdot \langle \nabla \ell(W_{1:N}(t)), u_r(t)v_r^T(t) \rangle
\]

**Proof Sketch**

**SVD:**

\[W_{1:N}(t) = U(t)S(t)V(t)^T \quad (S = \text{diag}(\sigma_1, \sigma_2, \ldots) \quad U = [u_1, u_2, \ldots] \quad V = [v_1, v_2, \ldots])\]

\[\implies \frac{d}{dt} W_{1:N}(t) = \frac{d}{dt} U(t) \cdot S(t) \cdot V(t)^T + U(t) \cdot \frac{d}{dt} S(t) \cdot V(t)^T + U(t) \cdot S(t) \cdot \frac{d}{dt} V(t)^T\]

\[\implies U(t)^T \cdot \frac{d}{dt} W_{1:N}(t) \cdot V(t) = U(t)^T \cdot \frac{d}{dt} U(t) \cdot S(t) + \frac{d}{dt} S(t) + S(t) \cdot \frac{d}{dt} V(t)^T \cdot V(t)\]

**End-to-end dynamics:**

\[
\frac{d}{dt} W_{1:N}(t) = -\sum_{j=1}^{N} U(t) \left[ S(t)S(t)^T \right]^{N-j \atop N} U(t)^T \cdot \nabla \ell(W_{1:N}(t)) \cdot V(t) \left[ S(t)^T S(t) \right]^{j-1 \atop N} V(t)^T
\]

\[\implies U(t)^T \cdot \frac{d}{dt} U(t) \cdot S(t) + \frac{d}{dt} S(t) + S(t) \cdot \frac{d}{dt} V(t)^T \cdot V(t)
\]

\[= -\sum_{j=1}^{N} \left[ S(t)S(t)^T \right]^{N-j \atop N} U(t)^T \cdot \nabla \ell(W_{1:N}(t)) \cdot V(t) \left[ S(t)^T S(t) \right]^{j-1 \atop N}\]

**Restrict attention to** \(r\)'**th diagonal element:**

\[
\frac{1}{2} \frac{d}{dt} \|u_r(t)\|_2^2 \cdot \sigma_r(t) + \frac{d}{dt} \sigma_r(t) + \sigma_r(t) \cdot \frac{1}{2} \frac{d}{dt} \|v_r(t)\|_2^2 = -N \cdot \sigma_r^{2 \cdot \frac{N-1}{N}}(t) \cdot \langle \nabla \ell(W_{1:N}(t)), u_r(t)v_r^T(t) \rangle
\]

\[\equiv 1\]

\[\equiv 1\]
Dynamics of Singular Values — Proof Sketch

**Theorem**

\[
\frac{d}{dt} \sigma_r(t) = -N \cdot \sigma_r^{\frac{2}{N}}(t) \cdot \langle \nabla \ell(W_{1:N}(t)), u_r(t)v_r^T(t) \rangle
\]

**Proof Sketch**

**SVD:**

\[
W_{1:N}(t) = U(t)S(t)V(t)^T \quad (S = \text{diag}(\sigma_1, \sigma_2, \ldots) \quad U = [u_1, u_2, \ldots] \quad V = [v_1, v_2, \ldots])
\]

\[
\Rightarrow \frac{d}{dt} W_{1:N}(t) = \frac{d}{dt} U(t) \cdot S(t) \cdot V(t)^T + U(t) \cdot \frac{d}{dt} S(t) \cdot V(t)^T + U(t) \cdot S(t) \cdot \frac{d}{dt} V(t)^T
\]

\[
\Rightarrow U(t)^T \cdot \frac{d}{dt} W_{1:N}(t) \cdot V(t) = U(t)^T \cdot \frac{d}{dt} U(t) \cdot S(t) + \frac{d}{dt} S(t) + S(t) \cdot \frac{d}{dt} V(t)^T \cdot V(t)
\]

**End-to-end dynamics:**

\[
\frac{d}{dt} W_{1:N}(t) = -\sum_{j=1}^{N} U(t) \left[ S(t)S(t)^T \right]^{\frac{N-j}{N}} U(t)^T \cdot \nabla \ell(W_{1:N}(t)) \cdot V(t) \left[ S(t)^T S(t) \right]^{\frac{i-1}{N}} V(t)^T
\]

\[
\Rightarrow U(t)^T \cdot \frac{d}{dt} U(t) \cdot S(t) + \frac{d}{dt} S(t) + S(t) \cdot \frac{d}{dt} V(t)^T \cdot V(t)
\]

\[
= -\sum_{j=1}^{N} \left[ S(t)S(t)^T \right]^{\frac{N-j}{N}} U(t)^T \cdot \nabla \ell(W_{1:N}(t)) \cdot V(t) \left[ S(t)^T S(t) \right]^{\frac{i-1}{N}}
\]

Restrict attention to \(r\)'th diagonal element:

\[
0 \cdot \sigma_r(t) + \frac{d}{dt} \sigma_r(t) + \sigma_r(t) \cdot 0 = -N \cdot \sigma_r^{\frac{2}{N}}(t) \cdot \langle \nabla \ell(W_{1:N}(t)), u_r(t)v_r^T(t) \rangle
\]
Dynamics of Singular Values — Proof Sketch

**Theorem**

\[
\frac{d}{dt} \sigma_r(t) = -N \cdot \sigma_r \left( 2 - \frac{2}{N} \right) (t) \cdot \langle \nabla \ell(W_{1:N}(t)), u_r(t)v_r^T(t) \rangle
\]

**Proof Sketch**

SVD: \( W_{1:N}(t) = U(t)S(t)V(t)^T \quad (S = \text{diag}(\sigma_1, \sigma_2, \ldots) \quad U = [u_1, u_2, \ldots] \quad V = [v_1, v_2, \ldots]) \)

\[
\implies \frac{d}{dt} W_{1:N}(t) = \frac{d}{dt} U(t) \cdot S(t) \cdot V(t)^T + U(t) \cdot \frac{d}{dt} S(t) \cdot V(t)^T + U(t) \cdot S(t) \cdot \frac{d}{dt} V(t)^T
\]

\[
\implies U(t)^T \cdot \frac{d}{dt} W_{1:N}(t) \cdot V(t) = U(t)^T \cdot \frac{d}{dt} U(t) \cdot S(t) + \frac{d}{dt} S(t) + S(t) \cdot \frac{d}{dt} V(t)^T \cdot V(t)
\]

End-to-end dynamics:

\[
\frac{d}{dt} W_{1:N}(t) = - \sum_{j=1}^{N} U(t) \left[ S(t)S(t)^T \right]^{N-j} \frac{N-j}{N} U(t)^T \cdot \nabla \ell(W_{1:N}(t)) \cdot V(t) \left[ S(t)^T S(t) \right]^{j-1} \frac{1}{N} V(t)^T
\]

\[
\implies U(t)^T \cdot \frac{d}{dt} U(t) \cdot S(t) + \frac{d}{dt} S(t) + S(t) \cdot \frac{d}{dt} V(t)^T \cdot V(t)
\]

\[
= - \sum_{j=1}^{N} \left[ S(t)S(t)^T \right]^{N-j} \frac{N-j}{N} U(t)^T \cdot \nabla \ell(W_{1:N}(t)) \cdot V(t) \left[ S(t)^T S(t) \right]^{j-1} \frac{1}{N}
\]

Restrict attention to \( r \)'th diagonal element:

\[
\frac{d}{dt} \sigma_r(t) = -N \cdot \sigma_r \left( 2 - \frac{2}{N} \right) (t) \cdot \langle \nabla \ell(W_{1:N}(t)), u_r(t)v_r^T(t) \rangle
\]
Dynamics of Singular Values — Proof Sketch

\[ \frac{d}{dt} \sigma_r(t) = -N \cdot \sigma_r^{2 - \frac{2}{N}}(t) \cdot \langle \nabla \ell(W_1:N(t)), u_r(t)v_r^T(t) \rangle \]

Proof Sketch

SVD: \( W_1:N(t) = U(t)S(t)V(t)^\top \) (\( S = \text{diag}(\sigma_1, \sigma_2, \ldots) \quad U = [u_1, u_2, \ldots] \quad V = [v_1, v_2, \ldots] \))

\[ \implies \frac{d}{dt} W_1:N(t) = \frac{d}{dt} U(t) \cdot S(t) \cdot V(t)^\top + U(t) \cdot \frac{d}{dt} S(t) \cdot V(t)^\top + U(t) \cdot S(t) \cdot \frac{d}{dt} V(t)^\top \]

\[ \implies U(t)^\top \cdot \frac{d}{dt} W_1:N(t) \cdot V(t) = U(t)^\top \cdot \frac{d}{dt} U(t) \cdot S(t) + \frac{d}{dt} S(t) + S(t) \cdot \frac{d}{dt} V(t)^\top \cdot V(t) \]

End-to-end dynamics:

\[ \frac{d}{dt} W_1:N(t) = -\sum_{j=1}^{N} U(t) \left[ S(t)S(t)^\top \right]^{\frac{j-1}{N}} U(t)^\top \cdot \nabla \ell(W_1:N(t)) \cdot V(t) \left[ S(t)^\top S(t) \right]^{\frac{j-1}{N}} V(t)^\top \]

\[ \implies U(t)^\top \cdot \frac{d}{dt} U(t) \cdot S(t) + \frac{d}{dt} S(t) + S(t) \cdot \frac{d}{dt} V(t)^\top \cdot V(t) \]

\[ = -\sum_{j=1}^{N} \left[ S(t)S(t)^\top \right]^{\frac{j-1}{N}} U(t)^\top \cdot \nabla \ell(W_1:N(t)) \cdot V(t) \left[ S(t)^\top S(t) \right]^{\frac{j-1}{N}} \]

Restrict attention to \( r \)'th diagonal element:

\[ \frac{d}{dt} \sigma_r(t) = -N \cdot \sigma_r^{2 - \frac{2}{N}}(t) \cdot \langle \nabla \ell(W_1:N(t)), u_r(t)v_r^T(t) \rangle \]
Implicit Bias Towards Low Rank

Experiment:
Completion of low rank matrix via GD over LNN

Theoretical Example:
For one observed entry and $\ell_2$ loss, relationship between singular vals is:
- depth 1: linear
- depth $\geq 3$: asymptotic

Depth leads to larger gaps between singular vals (lower rank)!
Implicit Bias Towards Low Rank

Experiment

Completion of low rank matrix via GD over LNN

- **depth 1** (reconst error: 8e-01)
- **depth 2** (reconst error: 6e-02)
- **depth 3** (reconst error: 3e-05)
Implicit Bias Towards Low Rank

**Experiment**
Completion of low rank matrix via GD over LNN

- **depth 1** (reconst error: 8e-01)
- **depth 2** (reconst error: 6e-02)
- **depth 3** (reconst error: 3e-05)

**Theoretical Example**
For one observed entry and $\ell_2$ loss, relationship between singular vals is:

- **depth 1**: linear
- **depth 2**: polynomial
- **depth $\geq 3$**: asymptotic
Implicit Bias Towards Low Rank

**Experiment**

Completion of low rank matrix via GD over LNN

- **Depth 1** (reconst error: 8e-01)
  - singular vals
  - iteration

- **Depth 2** (reconst error: 6e-02)
  - singular vals
  - iteration

- **Depth 3** (reconst error: 3e-05)
  - singular vals
  - iteration

**Theoretical Example**

For one observed entry and $\ell_2$ loss, relationship between singular vals is:

- **Depth 1**: linear
- **Depth 2**: polynomial
- **Depth $\geq 3$**: asymptotic

Depth leads to larger gaps between singular vals (lower rank)!
Conclusion

Outline

1. Optimization and Generalization in Deep Learning via Trajectories

2. Case Study: Linear Neural Networks
   - Trajectory Analysis
   - Optimization
   - Generalization

3. Conclusion
Recap

To understand optimization and generalization in deep learning:

- Language of classical learning theory may be insufficient
- Might need to analyze trajectories of gradient descent

Case Study — Deep Linear Neural Networks

- Trajectory analysis: Depth induces preconditioner promoting movement in directions taken
- Optimization: Guarantee of efficient convergence to global min (most general yet)
  - Depth can accelerate convergence (w/o any gain in expressiveness)
- Generalization: Depth enhances implicit regularization towards low rank, yielding generalization for problems such as matrix completion
Perspective
To understand optimization and generalization in deep learning:
Recap

Perspective

To understand optimization and generalization in deep learning:

- Language of classical learning theory may be insufficient
Recap

**Perspective**

To understand optimization and generalization in deep learning:

- Language of classical learning theory may be insufficient
- Might need to analyze trajectories of gradient descent
Recap

Perspective
To understand optimization and generalization in deep learning:
- Language of classical learning theory may be insufficient
- Might need to analyze trajectories of gradient descent

Case Study — Deep Linear Neural Networks
Recap

Perspective
To understand optimization and generalization in deep learning:

- Language of classical learning theory may be insufficient
- Might need to analyze trajectories of gradient descent

Case Study — Deep Linear Neural Networks
Trajectory analysis:
Recap

Perspective

To understand optimization and generalization in deep learning:

- Language of classical learning theory may be insufficient
- Might need to analyze trajectories of gradient descent

Case Study — Deep Linear Neural Networks

Trajectory analysis:

- **Depth induces preconditioner** promoting movement in directions taken
Recap

**Perspective**

To understand optimization and generalization in deep learning:

- Language of classical learning theory may be insufficient
- Might need to analyze trajectories of gradient descent

**Case Study — Deep Linear Neural Networks**

Trajectory analysis:

- **Depth induces preconditioner** promoting movement in directions taken

Optimization:
Recap

Perspective
To understand optimization and generalization in deep learning:
- Language of classical learning theory may be insufficient
- Might need to analyze trajectories of gradient descent

Case Study — Deep Linear Neural Networks
Trajectory analysis:
- Depth induces preconditioner promoting movement in directions taken

Optimization:
- Guarantee of efficient convergence to global min (most general yet)
Recap

Perspective
To understand optimization and generalization in deep learning:
- Language of classical learning theory may be insufficient
- Might need to analyze trajectories of gradient descent

Case Study — Deep Linear Neural Networks
Trajectory analysis:
- Depth induces preconditioner promoting movement in directions taken

Optimization:
- Guarantee of efficient convergence to global min (most general yet)
- Depth can accelerate convergence (w/o any gain in expressiveness)!
Recap

Perspective
To understand optimization and generalization in deep learning:
- Language of classical learning theory may be insufficient
- Might need to analyze trajectories of gradient descent

Case Study — Deep Linear Neural Networks

Trajectory analysis:
- **Depth induces preconditioner** promoting movement in directions taken

Optimization:
- **Guarantee of efficient convergence to global min** (most general yet)
- **Depth can accelerate convergence** (w/o any gain in expressiveness)

Generalization:
Recap

Perspective
To understand optimization and generalization in deep learning:
- Language of classical learning theory may be insufficient
- Might need to analyze trajectories of gradient descent

Case Study — Deep Linear Neural Networks

Trajectory analysis:
- **Depth induces preconditioner** promoting movement in directions taken

Optimization:
- **Guarantee of efficient convergence to global min** (most general yet)
- **Depth can accelerate convergence** (w/o any gain in expressiveness)!

Generalization:
- **Depth enhances implicit regularization towards low rank**, yielding generalization for problems such as matrix completion
1. Optimization and Generalization in Deep Learning via Trajectories

2. Case Study: Linear Neural Networks
   - Trajectory Analysis
   - Optimization
   - Generalization

3. Conclusion
Thank You