Analyzing Optimization and Generalization in Deep Learning via Trajectories of Gradient Descent

Nadav Cohen

Institute for Advanced Study $\longrightarrow$ Tel Aviv University

Frontiers of Deep Learning Workshop

Simons Institute for the Theory of Computing

15 July 2019

Outline

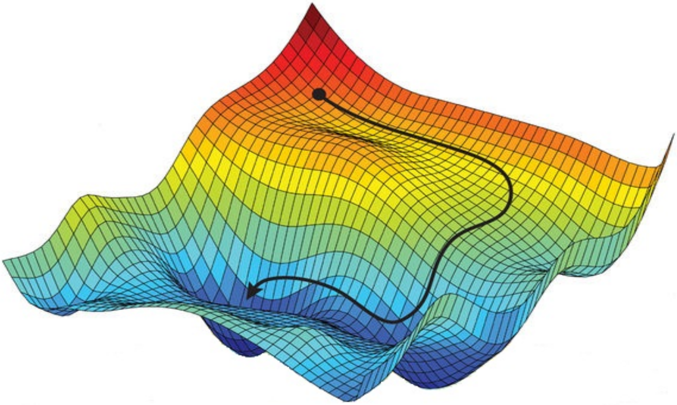
1. Optimization and Generalization in Deep Learning via Trajectories

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

3. Conclusion

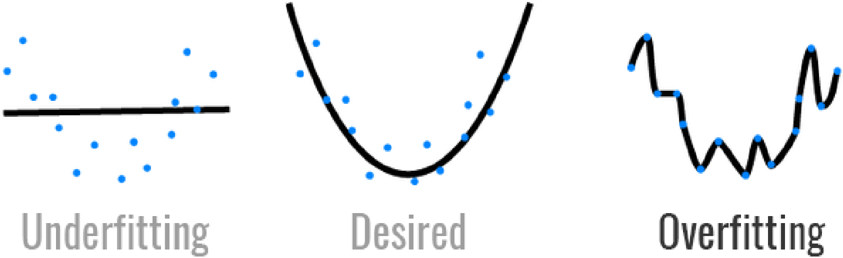
Optimization

Fitting training data by minimizing an objective (loss) function

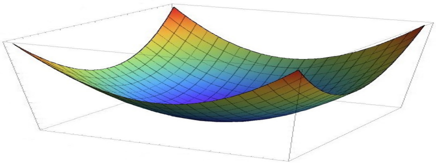


Generalization

Controlling gap between train and test errors, e.g. by adding regularization term/constraint to objective

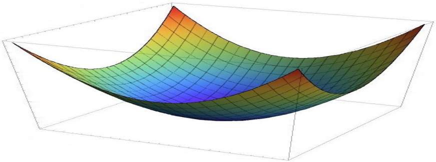


Classical Machine Learning



Theme: make sure objective is convex!

Classical Machine Learning

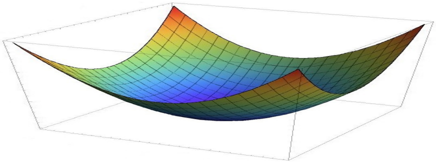


Theme: make sure objective is convex!

Optimization

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

Classical Machine Learning



Theme: make sure objective is convex!

Optimization

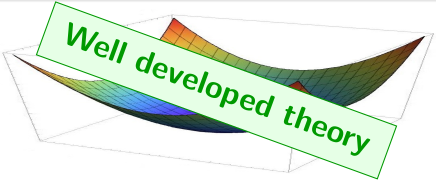
- Single global minimum, efficiently attainable
- Choice of algorithm affects only speed of convergence

Generalization

Bias-variance trade-off:

regularization	*train/test gap*	*train err*
more	↘	↗
less	↗	↘

Classical Machine Learning



Theme: make sure objective is convex!

Optimization

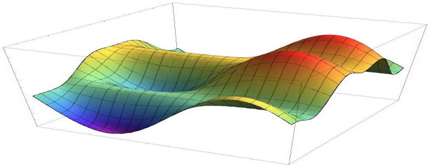
- Single global minimum, efficiently attainable
- Choice of algorithm affects only speed of convergence

Generalization

Bias-variance trade-off:

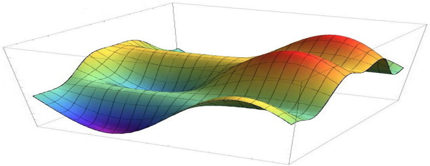
regularization	*train/test gap*	*train err*
more	$\searrow$	$\nearrow$
less	$\nearrow$	$\searrow$

Deep Learning (DL)



Theme: allow objective to be non-convex

Deep Learning (DL)

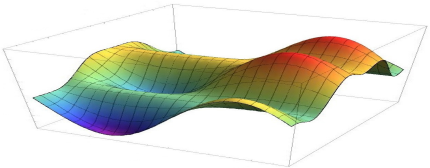


Theme: allow objective to be non-convex

Optimization

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

Deep Learning (DL)



Theme: allow objective to be non-convex

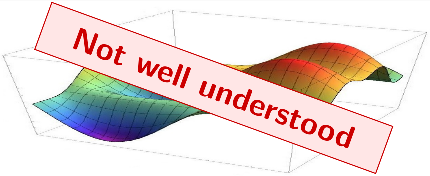
Optimization

- 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

Analysis via Trajectories of Gradient Descent

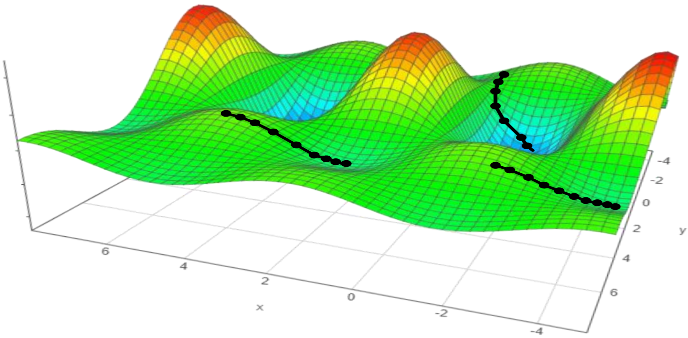
Perspective

- Language of classical learning theory may be insufficient for DL

Analysis via Trajectories of Gradient Descent

Perspective

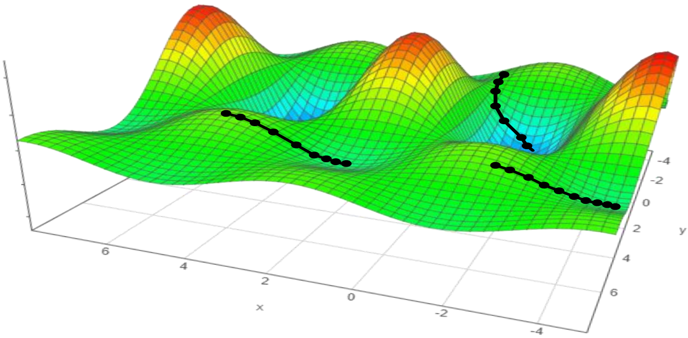
- Language of classical learning theory may be insufficient for DL
- Need to carefully analyze course of learning, i.e. trajectories of GD!



Analysis via Trajectories of Gradient Descent

Perspective

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



Case will be made via deep linear neural networks

Outline

Sources

On the Optimization of Deep Networks:
Implicit Acceleration by Overparameterization

Arora + ***C*** + Hazan (alphabetical order)

International Conference on Machine Learning (ICML) 2018

A Convergence Analysis of Gradient Descent for Deep Linear Neural Networks

Arora + ***C*** + Golowich + Hu (alphabetical order)

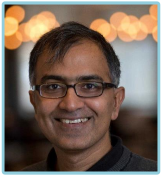
International Conference on Learning Representations (ICLR) 2019

Implicit Regularization in Deep Matrix Factorization

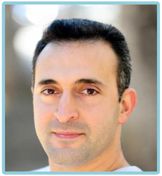
Arora + ***C*** + Hu + Luo (alphabetical order)

Preprint 2019

Collaborators



Sanjeev Arora



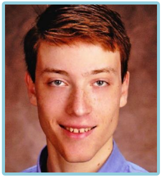
Elad Hazan

Google

PRINCETON UNIVERSITY

Yuping Luo

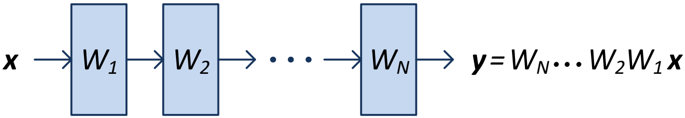
Wei Hu



Noah Golowich

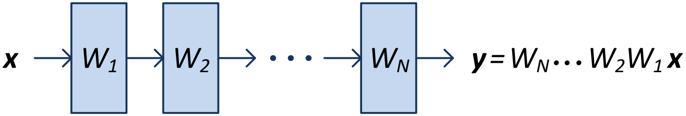
Linear Neural Networks

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



Linear Neural Networks

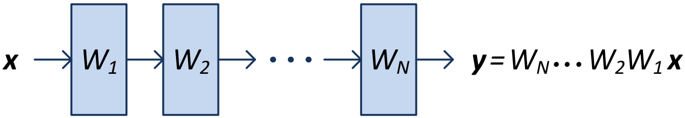
Linear neural networks (LNN) are fully-connected neural networks with linear (no) activation



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



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

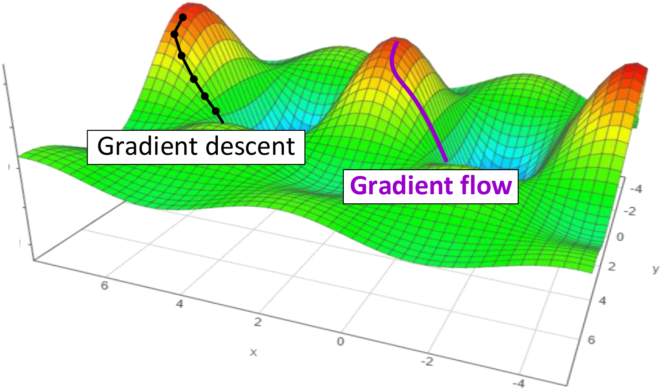
2. Case Study: Linear Neural Networks
   - Trajectory Analysis
   - Optimization
   - Generalization

3. Conclusion

Gradient Flow

Gradient flow (GF) is a continuous version of GD (step size $\to 0$):

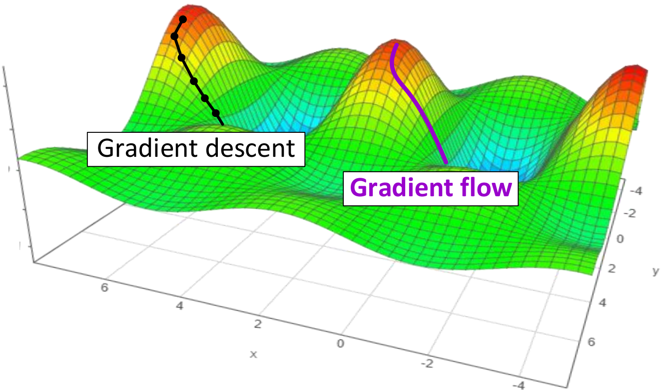
$$\frac{d}{dt}\alpha(t) = -\nabla f(\alpha(t)) \quad , \; t \in \mathbb{R}_{>0}$$



Gradient Flow

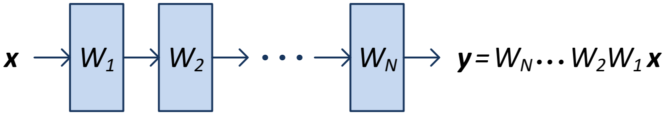
Gradient flow (GF) is a continuous version of GD (step size $\to 0$):

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

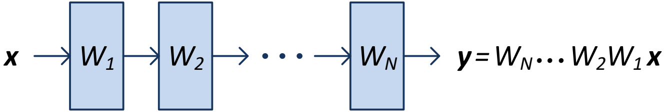


Admits use of theoretical tools from differential geometry/equations

Balanced Trajectories



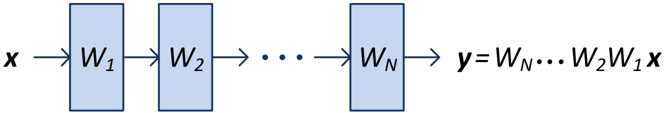
Balanced Trajectories



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



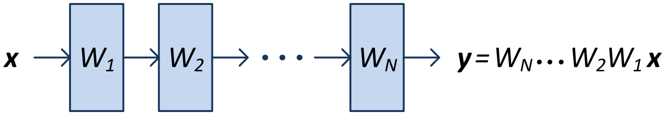
Loss $\ell(\cdot)$ for linear model induces **overparameterized objective** for LNN:

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

Definition

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

Balanced Trajectories



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

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

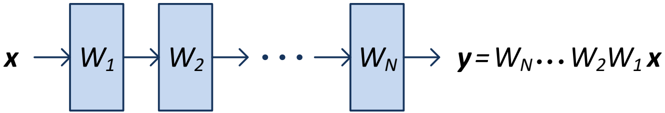
Definition

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

↑

Holds approximately under ≈ 0 init, exactly under residual (I_d) init

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 $W_{j+1}^\top W_{j+1} = W_j W_j^\top \ , \forall j$.

↑

Holds approximately under ≈ 0 init, exactly under residual (I_d) init

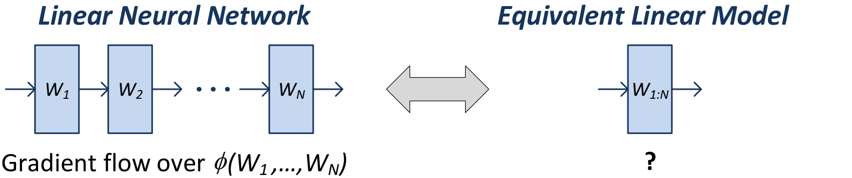
Claim

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

Implicit Preconditioning

Question

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

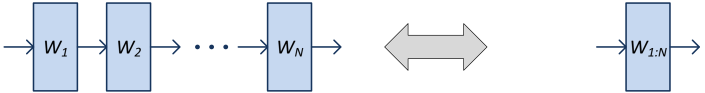


Implicit Preconditioning

Question

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

Linear Neural Network | ***Equivalent Linear Model***



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

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****:*

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

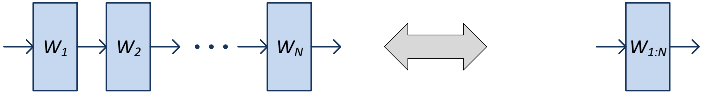
where $P_{W_{1:N}(t)}$ is a preconditioner (PSD matrix) that "reinforces" $W_{1:N}(t)$

Implicit Preconditioning

Question

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

Linear Neural Network ***Equivalent Linear Model***



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

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**:*

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

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

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

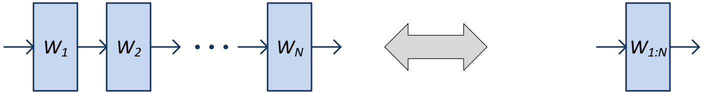
$$vec\left[\sum_{j=1}^{N}\left[W_{1:N}(t) W_{1:N}(t)^\top\right]^{\frac{N-j}{N}} \cdot \nabla \ell\left(W_{1:N}(t)\right) \cdot \left[W_{1:N}(t)^\top W_{1:N}(t)\right]^{\frac{j-1}{N}}\right]$$

Implicit Preconditioning

Question

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

Linear Neural Network ***Equivalent Linear Model***



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

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****:*

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

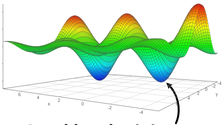
where $P_{W_{1:N}(t)}$ is a preconditioner (PSD matrix) that "reinforces" $W_{1:N}(t)$

Adding (redundant) linear layers to classic linear model induces preconditioner promoting movement in directions already taken!

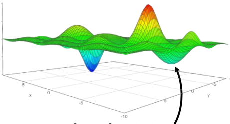
Outline

Classic Approach: Characterization of Critical Points

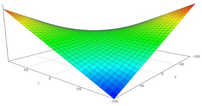
Prominent approach for analyzing optimization in DL (in spirit of classical learning theory) is via critical points in the objective



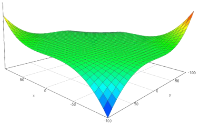
Good local minimum (≈ global minimum)



Poor local minimum



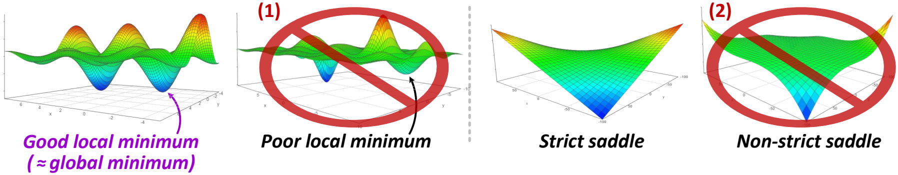
Strict saddle



Non-strict saddle

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

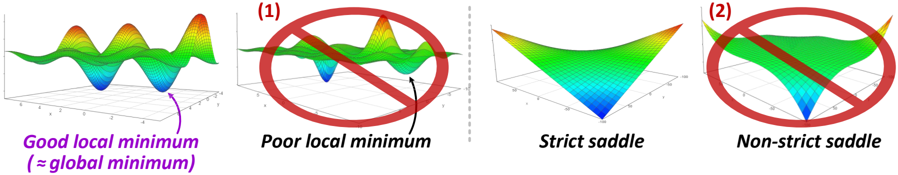


Result *(cf. Ge et al. 2015; Lee et al. 2016)*

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

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)*

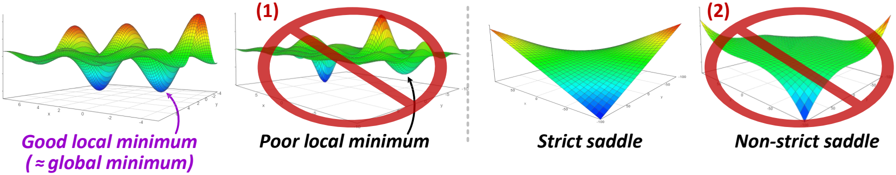
If: **(1)** there are no poor local minima; and **(2)** all saddle points are strict, then GD converges to global min

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

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

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

Applying Our Trajectory Analysis

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

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

Applying Our Trajectory Analysis

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

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

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

Applying Our Trajectory Analysis

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

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

$P_{W_{1:N}(t)} \succ 0$ 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

Applying Our Trajectory Analysis

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

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

$P_{W_{1:N}(t)} \succ 0$ 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

Applying Our Trajectory Analysis

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

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

$P_{W_{1:N}(t)} \succ 0$ 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

From Gradient Flow to Gradient Descent

From Gradient Flow to Gradient Descent

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

From Gradient Flow to Gradient Descent

Corollary

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

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

Then, GF converges to global min

From Gradient Flow to Gradient Descent

Corollary

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

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

Then, GF converges to global min

From Gradient Flow to Gradient Descent

Corollary

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

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

Then, GF converges to global min

From Gradient Flow to Gradient Descent

Theorem

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

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

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

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

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

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

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

From Gradient Flow to Gradient Descent

Theorem

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

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

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

Claim

Our assumptions on init:

From Gradient Flow to Gradient Descent

Theorem

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

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

Then, GD with step size $\eta \leq \mathcal{O}(c^4)$ gives: 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)*

From Gradient Flow to Gradient Descent

Theorem

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

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

Then, GD with step size $\eta \leq \mathcal{O}(c^4)$ gives: 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*

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

Then, GD with step size $\eta \leq \mathcal{O}(c^4)$ gives: 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*

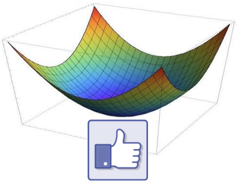
Guarantee of efficient (linear rate) convergence to global min!
Most general guarantee to date for GD efficiently training deep net.

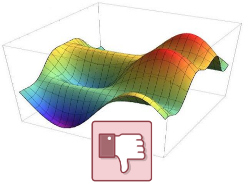
Effect of Depth on Optimization

Effect of Depth on Optimization

Viewpoint of classical learning theory:

- Convex optimization is easier than non-convex

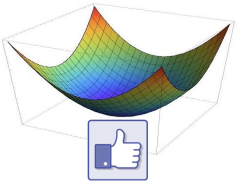


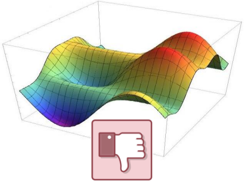


Effect of Depth on Optimization

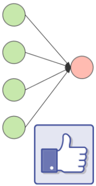
Viewpoint of classical learning theory:

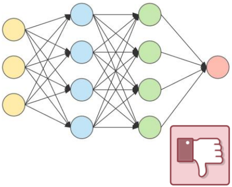
- Convex optimization is easier than non-convex





- Hence depth complicates optimization

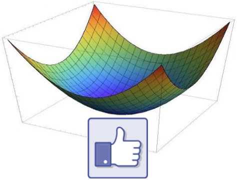


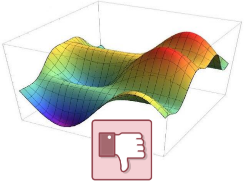


Effect of Depth on Optimization

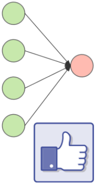
Viewpoint of classical learning theory:

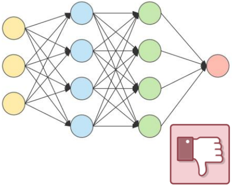
- Convex optimization is easier than non-convex





- Hence depth complicates optimization





Our trajectory analysis reveals: not always true...

Acceleration by Depth

Acceleration by Depth

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

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

Acceleration by Depth

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

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

Claim

$\forall p > 2$, $\exists$ settings where $\ell(\cdot) = \ell_p$ loss (i.e. $\ell(W) = \frac{1}{m}\sum_{i=1}^{m} \| W\mathbf{x}_i - \mathbf{y}_i \|_p^p$) 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:

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

Claim

$\forall p > 2$, $\exists$ settings where $\ell(\cdot) = \ell_p$ loss (i.e. $\ell(W) = \frac{1}{m}\sum_{i=1}^{m} \|W\mathbf{x}_i - \mathbf{y}_i\|_p^p$) 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:

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

Claim

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

Experiment

Regression problem from UCI ML Repository ; ℓ_4 loss

Acceleration by Depth

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

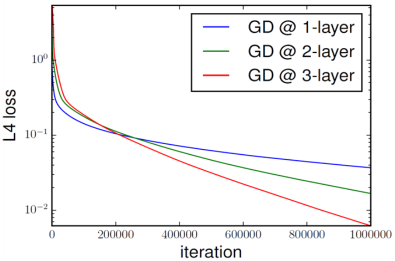
$$vec\left[W_{1:N}(t+1)\right] \hookleftarrow vec\left[W_{1:N}(t)\right] - \eta \cdot P_{W_{1:N}(t)} \cdot vec\left[\nabla\ell(W_{1:N}(t))\right]$$

Claim

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

Experiment

Regression problem from UCI ML Repository ; ℓ_4 loss



Acceleration by Depth

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

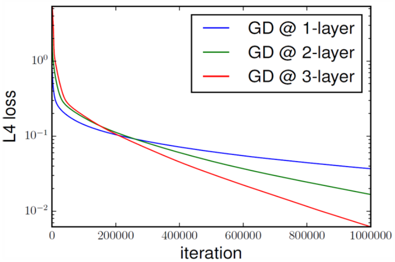
$$vec\left[W_{1:N}(t+1)\right] \leftarrow vec\left[W_{1:N}(t)\right] - \eta \cdot P_{W_{1:N}(t)} \cdot vec\left[\nabla\ell(W_{1:N}(t))\right]$$

Claim

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

Experiment

Regression problem from UCI ML Repository ; ℓ_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

Bob	4	?	?	4
Alice	?	5	4	?
Joe	?	5	?	?

Setting: Matrix Completion

Matrix completion: recover matrix given subset of entries

Bob	4	?	?	4
Alice	?	5	4	?
Joe	?	5	?	?

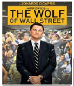
Can be viewed as classification (regression) problem:

observed entries $\longleftrightarrow$ *training data*

unobserved entries $\longleftrightarrow$ *test data*

Setting: Matrix Completion

Matrix completion: recover matrix given subset of entries

				
Bob	4	?	?	4
Alice	?	5	4	?
Joe	?	5	?	?

Can be viewed as classification (regression) problem:

observed entries $\longleftrightarrow$ *training data*

unobserved entries $\longleftrightarrow$ *test data*

Standard Assumption

Matrix to recover (ground truth) is low-rank

Setting: Matrix Completion

Matrix completion: recover matrix given subset of entries

Bob	4	?	?	4
Alice	?	5	4	?
Joe	?	5	?	?

Can be viewed as classification (regression) problem:

observed entries $\longleftrightarrow$ *training data*

unobserved entries $\longleftrightarrow$ *test data*

Standard Assumption

Matrix to recover (ground truth) is 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

Two-Layer Network ⟷ Matrix Factorization

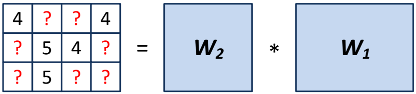
Matrix completion via two-layer LNN:

- Parameterize ground truth as $W_2 W_1$

Two-Layer Network $\longleftrightarrow$ Matrix Factorization

Matrix completion via two-layer LNN:

- Parameterize ground truth as $W_2 W_1$



- Known as **matrix factorization** (MF)

Two-Layer Network $\longleftrightarrow$ Matrix Factorization

Matrix completion via two-layer LNN:

- Parameterize ground truth as $W_2 W_1$

4	?	?	4
?	5	4	?
?	5	?	?

= W_2 * W_1

- Known as **matrix factorization** (MF)

Empirical Phenomenon

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

Two-Layer Network $\longleftrightarrow$ Matrix Factorization

Matrix completion via two-layer LNN:

- Parameterize ground truth as $W_2 W_1$

$$\begin{bmatrix} 4 & ? & ? & 4 \\ ? & 5 & 4 & ? \\ ? & 5 & ? & ? \end{bmatrix} = \boldsymbol{W_2} * \boldsymbol{W_1}$$

- Known as **matrix factorization** (MF)

Empirical Phenomenon

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

Two-Layer Network $\longleftrightarrow$ Matrix Factorization

Matrix completion via two-layer LNN:

- Parameterize ground truth as $W_2 W_1$

- Known as **matrix factorization** (MF)

Empirical Phenomenon

GD (with step size $\ll 1$ and init ≈ 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)

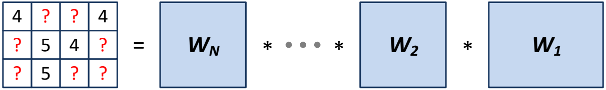
GD (with step size $\ll 1$ and init ≈ 0) over MF converges to solution with min nuclear norm (among those fitting observations)

Gunasekar et al. 2017 proved conjecture for a certain restricted setting

N-Layer Network $\longleftrightarrow$ "Deep Matrix Factorization"

Matrix completion via N-layer LNN:

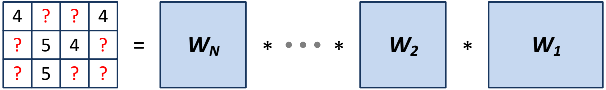
- Parameterize ground truth as $W_N \cdots W_2 W_1$



N-Layer Network $\longleftrightarrow$ "Deep Matrix Factorization"

Matrix completion via N-layer LNN:

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

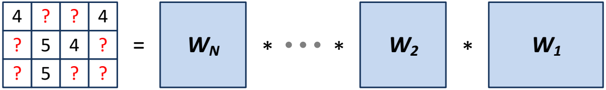


- We refer to this as **deep matrix factorization** (DMF)

N-Layer Network $\longleftrightarrow$ "Deep Matrix Factorization"

Matrix completion via N-layer LNN:

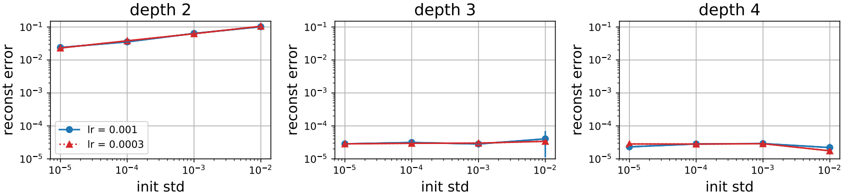
- Parameterize ground truth as $W_N \cdots W_2 W_1$



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

Experiment

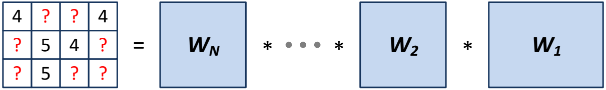
Completion of low-rank matrix via GD over DMF



N-Layer Network $\longleftrightarrow$ "Deep Matrix Factorization"

Matrix completion via N-layer LNN:

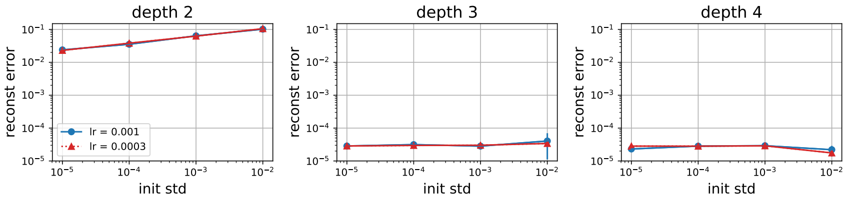
- Parameterize ground truth as $W_N \cdots W_2 W_1$



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

Experiment

Completion of low-rank matrix via GD over DMF



Depth enhanced implicit regularization towards low rank!

Can the Implicit Regularization Be Captured by Norms?

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)	$\longleftrightarrow$	*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):

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

In light of our experiments, natural to hypothesize:

implicit regularization with deeper LNN (DMF) $\longleftrightarrow$ *minimizing other norm or quasi-norm closer to rank*

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)	$\longleftrightarrow$	*minimizing nuclear norm (surrogate for rank)*

In light of our experiments, natural to hypothesize:

implicit regularization with deeper LNN (DMF)	$\longleftrightarrow$	*minimizing other norm or quasi-norm closer to rank*

Example

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

- $\|W\|_{S_p}^p := \sum_r \sigma_r^p(W)$ where $\sigma_r(W)$ 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):

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

In light of our experiments, natural to hypothesize:

implicit regularization with deeper LNN (DMF) $\longleftrightarrow$ *minimizing other norm or quasi-norm closer to rank*

Example

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

- $\|W\|_{S_p}^p := \sum_r \sigma_r^p(W)$ where $\sigma_r(W)$ are singular vals of W
- $p = 1$: nuclear norm, corresponds to depth 2 by Gunasekar et al. 2017

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) $\longleftrightarrow$ *minimizing nuclear norm (surrogate for rank)*

In light of our experiments, natural to hypothesize:

implicit regularization with deeper LNN (DMF) $\longleftrightarrow$ *minimizing other norm or quasi-norm closer to rank*

Example

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

- $\|W\|_{S_p}^p := \sum_r \sigma_r^p(W)$ 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

Current Theory is Oblivious to Depth

Theorem

In the restricted setting where Gunasekar et al. 2017 proved conjecture, nuclear norm is minimized not just with depth 2, but with any depth ≥ 2

Current Theory is Oblivious to Depth

Theorem

In the restricted setting where Gunasekar et al. 2017 proved conjecture, nuclear norm is minimized not just with depth 2, but with any depth ≥ 2

Proposition

There exist 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 the restricted setting where Gunasekar et al. 2017 proved conjecture, nuclear norm is minimized not just with depth 2, but with any depth ≥ 2

Proposition

There exist 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 with any depth $\neq$ Schatten quasi-norm minimization

Current Theory is Oblivious to Depth

Theorem

In the restricted setting where Gunasekar et al. 2017 proved conjecture, nuclear norm is minimized not just with depth 2, but with any depth ≥ 2

Proposition

There exist 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 with any depth $\neq$ Schatten quasi-norm minimization

Instead, adopting lens of Gunasekar et al. 2017 leads to conjecturing:

implicit regularization with any depth $=$ nuclear norm minimization

Current Theory is Oblivious to Depth

Theorem

In the restricted setting where Gunasekar et al. 2017 proved conjecture, nuclear norm is minimized not just with depth 2, but with any depth ≥ 2

Proposition

There exist 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 with any depth $\neq$ Schatten quasi-norm minimization

Instead, adopting lens of Gunasekar et al. 2017 leads to conjecturing:

implicit regularization with any depth $=$ nuclear norm minimization

But our experiments show depth changes implicit regularization!

Experiments Testing Nuclear Norm Conjecture

Experiments Testing Nuclear Norm Conjecture

Setup:

- Completion of 100×100 rank 5 matrix
- Observed entries chosen uniformly at random

Experiments Testing Nuclear Norm Conjecture

Setup:

- Completion of 100×100 rank 5 matrix
- Observed entries chosen uniformly at random

Many (5K) Observations:

	reconst err	*nuclear norm*	*effective rank*
nuclear norm min			
depth-2 LNN			
depth-3 LNN			
depth-4 LNN			

Experiments Testing Nuclear Norm Conjecture

Setup:

- Completion of 100×100 rank 5 matrix
- Observed entries chosen uniformly at random

Many (5K) Observations:

	reconst err	***nuclear norm***	***effective rank***
nuclear norm min	8 e −07	221	5
depth-2 LNN			
depth-3 LNN			
depth-4 LNN			

- Nuclear norm minimization recovers ground truth

Experiments Testing Nuclear Norm Conjecture

Setup:

- Completion of 100×100 rank 5 matrix
- Observed entries chosen uniformly at random

Many (5K) Observations:

	reconst err	*nuclear norm*	*effective rank*
nuclear norm min	8 e −07	221	5
depth-2 LNN	5 e −06	221	5
depth-3 LNN	4 e −06	221	5
depth-4 LNN	4 e −06	221	5

- Nuclear norm minimization recovers ground truth
- LNN do so too

Experiments Testing Nuclear Norm Conjecture

Setup:

- Completion of 100×100 rank 5 matrix
- Observed entries chosen uniformly at random

Many (5K) Observations:

	reconst err	*nuclear norm*	*effective rank*
nuclear norm min	8 e −07	221	5
depth-2 LNN	5 e −06	221	5
depth-3 LNN	4 e −06	221	5
depth-4 LNN	4 e −06	221	5

- Nuclear norm minimization recovers ground truth
- LNN do so too
- Correspondence, but can't distinguish nuclear norm minimization from any other bias leading to low rank

Experiments Testing Nuclear Norm Conjecture (cont')

Few (2K) Observations:

	reconst err	*nuclear norm*	*effective rank*
nuclear norm min			
depth-2 LNN			
depth-3 LNN			
depth-4 LNN			

Experiments Testing Nuclear Norm Conjecture (cont')

Few (2K) Observations:

	reconst err	***nuclear norm***	***effective rank***
nuclear norm min	1 e −01	210	9
depth-2 LNN			
depth-3 LNN			
depth-4 LNN			

- Nuclear norm minimization does not recover ground truth

Experiments Testing Nuclear Norm Conjecture (cont')

Few (2K) Observations:

	reconst err	***nuclear norm***	***effective rank***
nuclear norm min	1 e −01	210	9
depth-2 LNN	2 e −02	217	7
depth-3 LNN	3 e −05	221	5
depth-4 LNN	2 e −05	221	5

- Nuclear norm minimization does not recover ground truth
- LNN focus on lowering effective rank at expense of nuclear norm

Experiments Testing Nuclear Norm Conjecture (cont')

Few (2K) Observations:

	reconst err	***nuclear norm***	***effective rank***
nuclear norm min	1 e −01	210	9
depth-2 LNN	2 e −02	217	7
depth-3 LNN	3 e −05	221	5
depth-4 LNN	2 e −05	221	5

- Nuclear norm minimization does not recover ground truth
- LNN focus on lowering effective rank at expense of nuclear norm
- Discrepancy!

Experiments Testing Nuclear Norm Conjecture (cont')

Few (2K) Observations:

	reconst err	***nuclear norm***	***effective rank***
nuclear norm min	1 e −01	210	9
depth-2 LNN	2 e −02	217	7
depth-3 LNN	3 e −05	221	5
depth-4 LNN	2 e −05	221	5

- Nuclear norm minimization does not 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:

	reconst err	*nuclear norm*	*effective rank*
nuclear norm min	1 e −01	210	9
depth-2 LNN	2 e −02	217	7
depth-3 LNN	3 e −05	221	5
depth-4 LNN	2 e −05	221	5

- Nuclear norm minimization does not 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 $\longrightarrow$ Dynamics of Singular Values

Trajectory Analysis ⟶ Dynamics of Singular Values

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

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

Trajectory Analysis $\longrightarrow$ Dynamics of Singular Values

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

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

Denote:

- $\{\sigma_r(t)\}_r$ — singular vals of $W_{1:N}(t)$
- $\{\mathbf{u}_r(t)\}_r / \{\mathbf{v}_r(t)\}_r$ — corresponding left/right (resp) singular vecs

Trajectory Analysis $\longrightarrow$ Dynamics of Singular Values

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

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

Denote:

- $\{\sigma_r(t)\}_r$ — singular vals of $W_{1:N}(t)$
- $\{\mathbf{u}_r(t)\}_r / \{\mathbf{v}_r(t)\}_r$ — corresponding left/right (resp) singular vecs

Theorem

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

Trajectory Analysis ⟶ Dynamics of Singular Values

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

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

Denote:

- $\{\sigma_r(t)\}_r$ — singular vals of $W_{1:N}(t)$
- $\{\mathbf{u}_r(t)\}_r / \{\mathbf{v}_r(t)\}_r$ — corresponding left/right (resp) singular vecs

Theorem

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

Interpretation

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

Trajectory Analysis $\longrightarrow$ Dynamics of Singular Values

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

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

Denote:

- $\{\sigma_r(t)\}_r$ — singular vals of $W_{1:N}(t)$
- $\{\mathbf{u}_r(t)\}_r / \{\mathbf{v}_r(t)\}_r$ — corresponding left/right (resp) singular vecs

Theorem

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

Interpretation

- Given $W_{1:N}(t)$, depth affects evolution only via factors $N \cdot \sigma_r^{2-\frac{2}{N}}(t)$
- $N = 1$ (classic linear model): factors reduce to 1

Trajectory Analysis $\longrightarrow$ Dynamics of Singular Values

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

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

Denote:

- $\{\sigma_r(t)\}_r$ — singular vals of $W_{1:N}(t)$
- $\{\mathbf{u}_r(t)\}_r / \{\mathbf{v}_r(t)\}_r$ — corresponding left/right (resp) singular vecs

Theorem

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

Interpretation

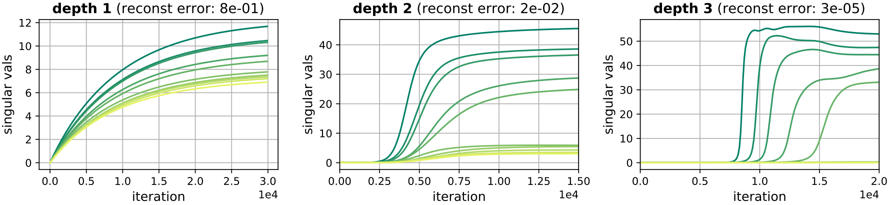
- Given $W_{1:N}(t)$, depth affects evolution only via factors $N \cdot \sigma_r^{2-\frac{2}{N}}(t)$
- $N = 1$ (classic linear model): factors reduce to 1
- $N \geq 2$: factors speed-up/slow-down large/small (resp) singular vals, in manner which intensifies with depth

Implicit Bias Towards Low Rank

Implicit Bias Towards Low Rank

Experiment

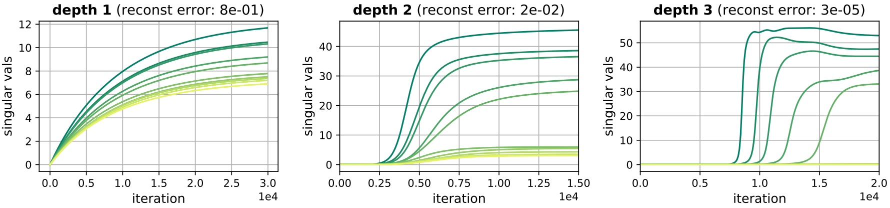
Completion of low-rank matrix via GD over LNN



Implicit Bias Towards Low Rank

Experiment

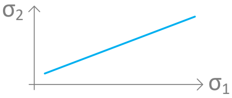
Completion of low-rank matrix via GD over LNN



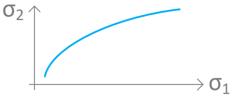
Theoretical Example

For one observed entry and ℓ_2 loss, relationship between singular vals is:

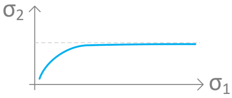
depth 1: linear



depth 2: polynomial



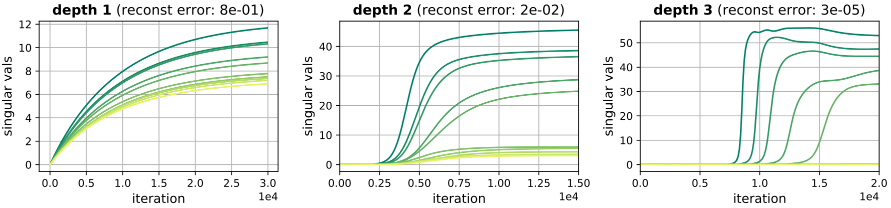
depth ≥ 3: asymptotic



Implicit Bias Towards Low Rank

Experiment

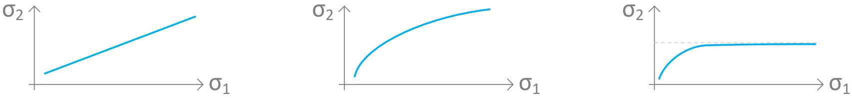
Completion of low-rank matrix via GD over LNN



Theoretical Example

For one observed entry and ℓ_2 loss, relationship between singular vals is:

depth 1: linear   *depth 2: polynomial*   *depth ≥ 3: asymptotic*



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

Outline

1. Optimization and Generalization in Deep Learning via Trajectories

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

3. Conclusion

Recap

Recap

Perspective

Understanding optimization and generalization in deep learning:

Recap

Perspective

Understanding optimization and generalization in deep learning:

- Language of classical learning theory is insufficient

Recap

Perspective

Understanding optimization and generalization in deep learning:

- Language of classical learning theory is insufficient
- Need to analyze trajectories of gradient descent

Recap

Perspective

Understanding optimization and generalization in deep learning:

- Language of classical learning theory is insufficient
- Need to analyze trajectories of gradient descent

Case Study — Deep Linear Neural Networks

Recap

Perspective

Understanding optimization and generalization in deep learning:

- Language of classical learning theory is insufficient
- Need to analyze trajectories of gradient descent

Case Study — Deep Linear Neural Networks

Trajectory analysis:

Recap

Perspective

Understanding optimization and generalization in deep learning:

- Language of classical learning theory is insufficient
- 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

Understanding optimization and generalization in deep learning:

- Language of classical learning theory is insufficient
- 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

Understanding optimization and generalization in deep learning:

- Language of classical learning theory is insufficient
- 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

Understanding optimization and generalization in deep learning:

- Language of classical learning theory is insufficient
- 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

Understanding optimization and generalization in deep learning:

- Language of classical learning theory is insufficient
- 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

Understanding optimization and generalization in deep learning:

- Language of classical learning theory is insufficient
- Need to analyze trajectories of gradient descent

Case Study — Deep Linear Neural Networks

Trajectory analysis:

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

Optimization:

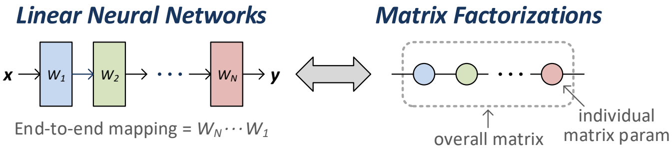
- **Guarantee of efficient convergence to global min** (most general yet)
- **Depth can accelerate convergence** (w/o any gain in expressiveness)!

Generalization:

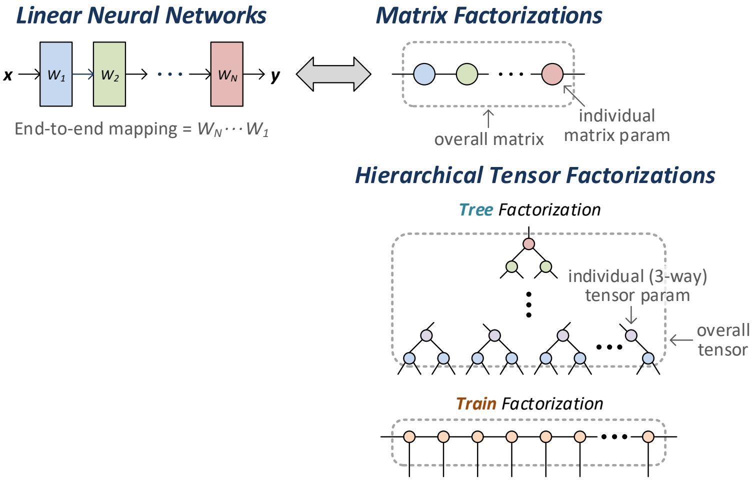
- **Depth enhances implicit regularization towards low rank**, yielding generalization for problems such as matrix completion

Beyond Linear Neural Networks

Beyond Linear Neural Networks

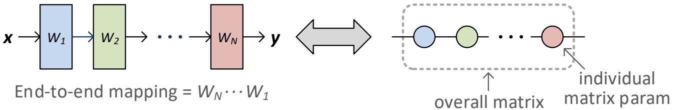


Beyond Linear Neural Networks



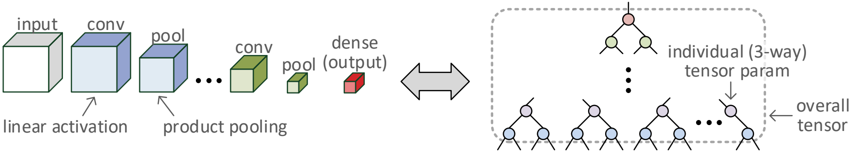
Beyond Linear Neural Networks

Linear Neural Networks ⟺ *Matrix Factorizations*

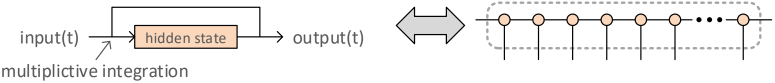


Arithmetic Neural Networks ⟺ *Hierarchical Tensor Factorizations*

Convolutional Network ⟺ *Tree Factorization*

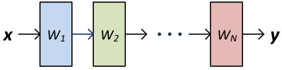


Recurrent Network ⟺ *Train Factorization*



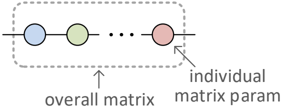
Beyond Linear Neural Networks

Linear Neural Networks



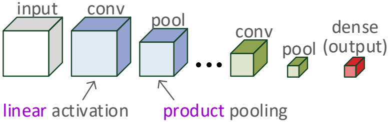
End-to-end mapping = $W_N \cdots W_1$

Matrix Factorizations



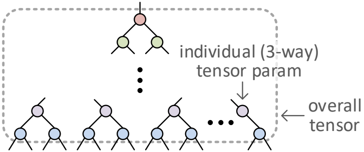
Arithmetic Neural Networks

Convolutional Network

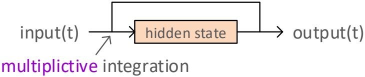


Hierarchical Tensor Factorizations

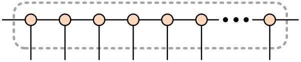
Tree Factorization



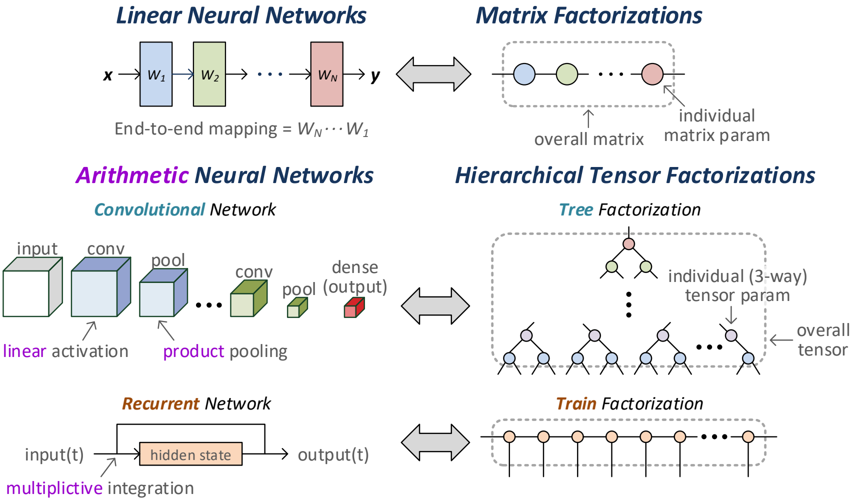
Recurrent Network



Train Factorization

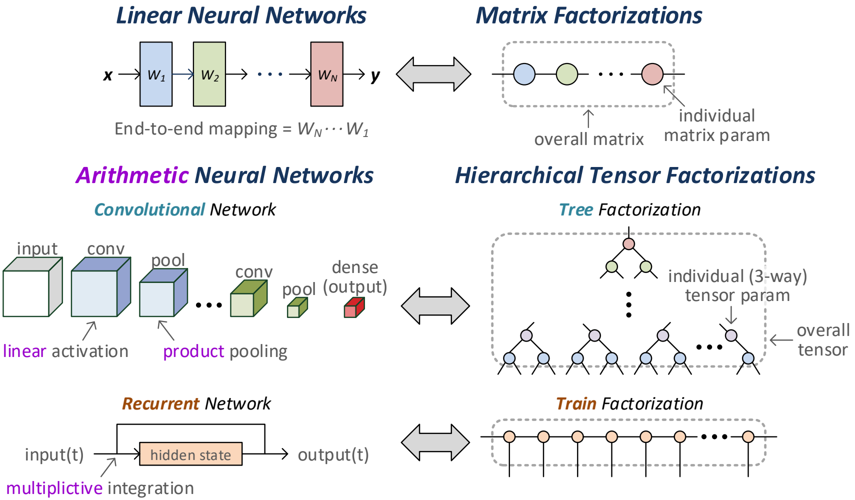


Beyond Linear Neural Networks



Arithmetic NN are competitive in practice, and admit algebraic structure

Beyond Linear Neural Networks



Arithmetic NN are competitive in practice, and admit algebraic structure

Preliminary analysis: their trajectories share properties with those of LNN...

Outline

1. Optimization and Generalization in Deep Learning via Trajectories

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

3. Conclusion

Thank You