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

Nadav Cohen

Tel Aviv University & IMUBIT

AI Week at Tel Aviv University

18 November 2019

Outline

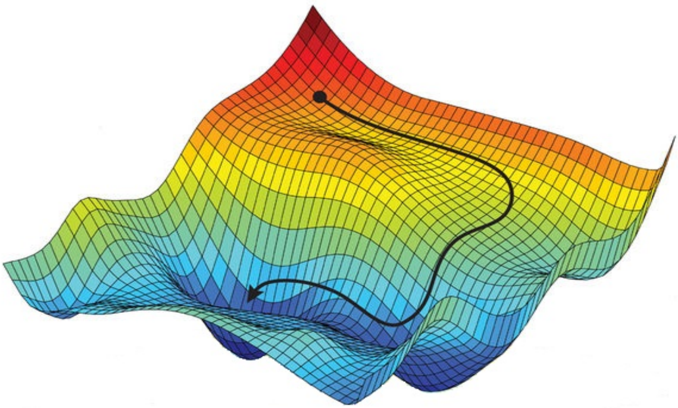
1. Optimization and Generalization in Deep Learning via Trajectories

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

3. Conclusion

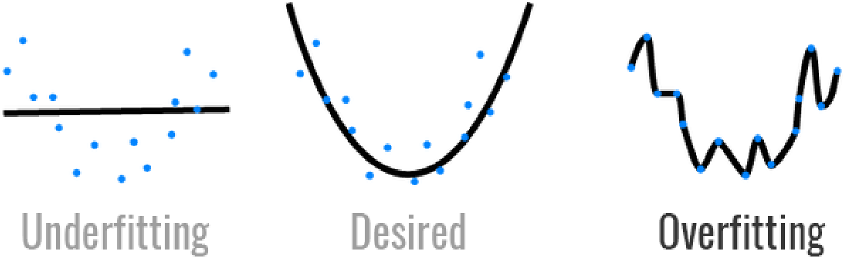
Optimization

Fitting training data by minimizing an objective (loss) function

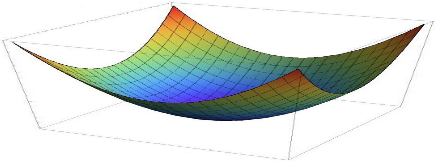


Generalization

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

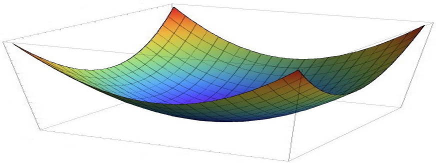


Classical Machine Learning



Theme: make sure objective is convex!

Classical Machine Learning

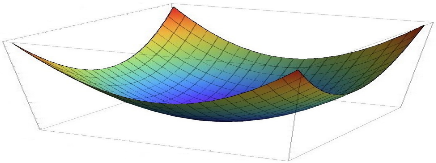


Theme: make sure objective is convex!

Optimization

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

Classical Machine Learning



Theme: make sure objective is convex!

Optimization

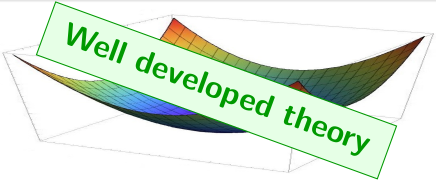
- Single global minimum, efficiently attainable
- Choice of algorithm affects only speed of convergence

Generalization

Bias-variance trade-off:

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

Classical Machine Learning



Theme: make sure objective is convex!

Optimization

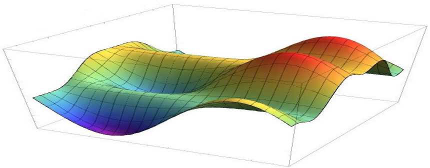
- Single global minimum, efficiently attainable
- Choice of algorithm affects only speed of convergence

Generalization

Bias-variance trade-off:

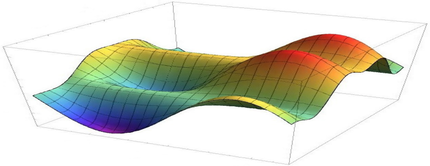
regularization	*train/test gap*	*train err*
more	$\searrow$	$\nearrow$
less	$\nearrow$	$\searrow$

Deep Learning (DL)



Theme: allow objective to be non-convex

Deep Learning (DL)

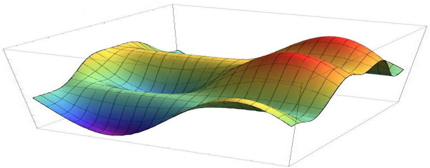


Theme: allow objective to be non-convex

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

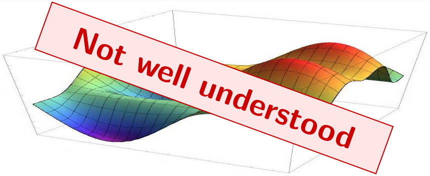
Optimization

- 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

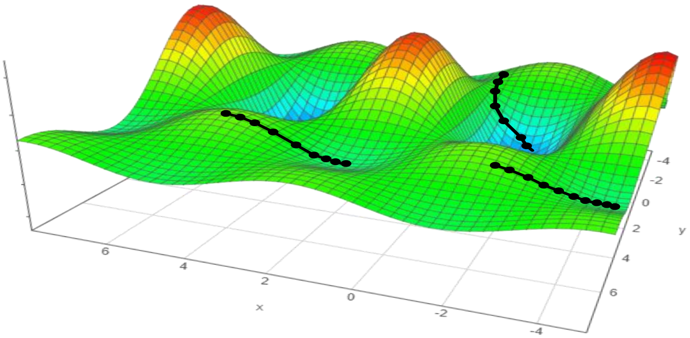
Perspective

- Language of classical learning theory may be insufficient for DL

Analysis via Trajectories of Gradient Descent

Perspective

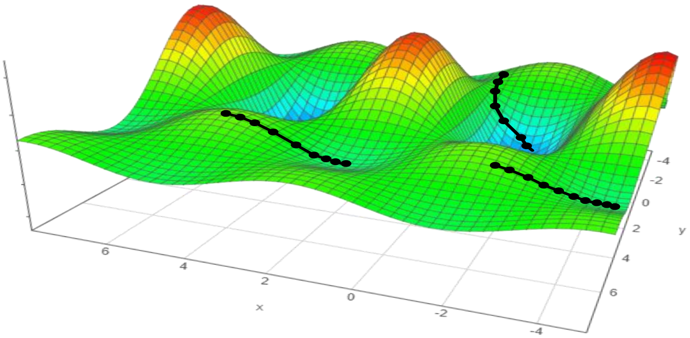
- Language of classical learning theory may be insufficient for DL
- Need to carefully analyze course of learning, i.e. trajectories of GD!



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

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 (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)

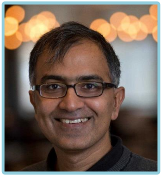
International Conference on Learning Representations (ICLR) 2019

Implicit Regularization in Deep Matrix Factorization

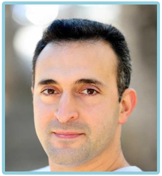
Arora + ***C*** + Hu + Luo (alphabetical order)

Conference on Neural Information Processing Systems (NeurIPS) 2019

Collaborators



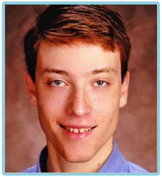
Sanjeev Arora



Elad Hazan

Yuping Luo

Wei Hu



Noah Golowich

Linear Neural Networks

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

$$\mathbf{x} \longrightarrow \boxed{W_1} \longrightarrow \boxed{W_2} \longrightarrow \cdots \longrightarrow \boxed{W_N} \longrightarrow \mathbf{y} = W_N \cdots W_2 W_1 \mathbf{x}$$

Linear Neural Networks

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

$$\boldsymbol{x} \longrightarrow W_1 \longrightarrow W_2 \longrightarrow \cdots \longrightarrow W_N \longrightarrow \boldsymbol{y} = W_N \cdots W_2 W_1 \boldsymbol{x}$$

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

$$\mathbf{x} \longrightarrow \boxed{W_1} \longrightarrow \boxed{W_2} \longrightarrow \cdots \longrightarrow \boxed{W_N} \longrightarrow \mathbf{y} = W_N \cdots W_2 W_1 \mathbf{x}$$

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:

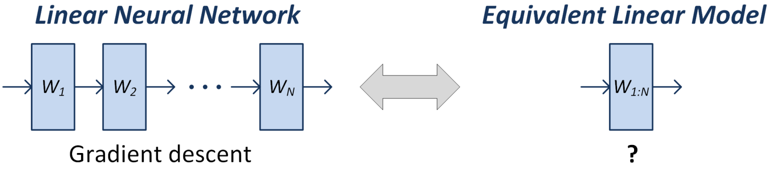
- Saxe et al. 2014
- Kawaguchi 2016
- Advani & Saxe 2017
- Hardt & Ma 2017
- Laurent & Brecht 2018
- Gunasekar et al. 2018
- Ji & Telgarsky 2019
- Lampinen & Ganguli 2019

Outline

Implicit Preconditioning

Question

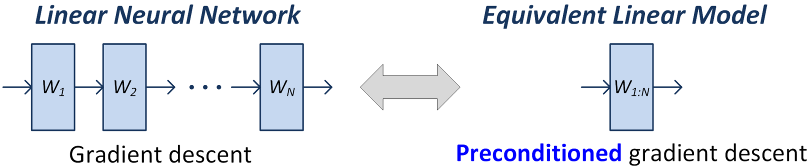
How does **end-to-end matrix** $W_{1:N} := W_N \cdots W_1$ move on GD trajectories?



Implicit Preconditioning

Question

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



Theorem

*$W_{1:N}$ follows **end-to-end dynamics**:*

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

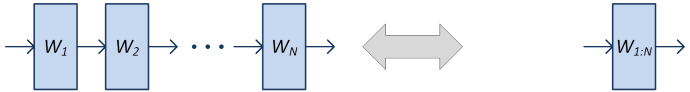
where $P_{W_{1:N}(t)}$ is a preconditioner (PSD matrix) that "reinforces" $W_{1:N}(t)$

Implicit Preconditioning

Question

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

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



Gradient descent | **Preconditioned** gradient descent

Theorem

*$W_{1:N}$ follows **end-to-end dynamics**:*

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

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

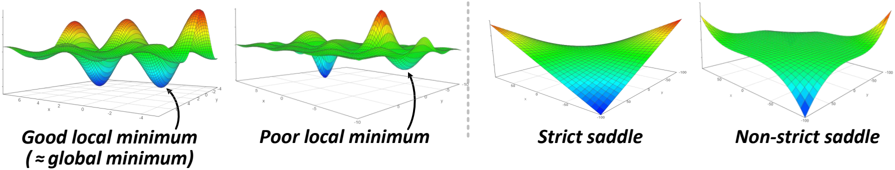
1. Optimization and Generalization in Deep Learning via Trajectories

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

3. Conclusion

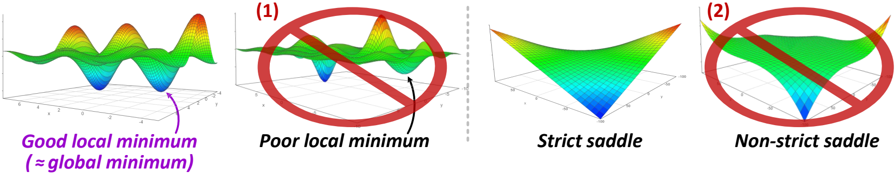
Classic Approach: Characterization of Critical Points

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



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

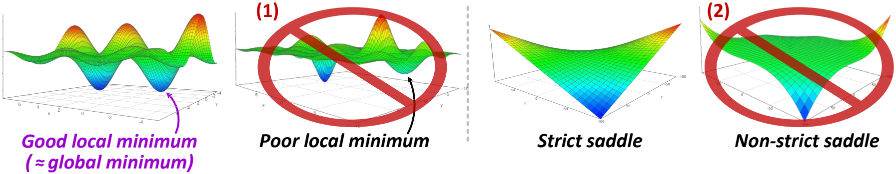


Result *(cf. Ge et al. 2015; Lee et al. 2016)*

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)*

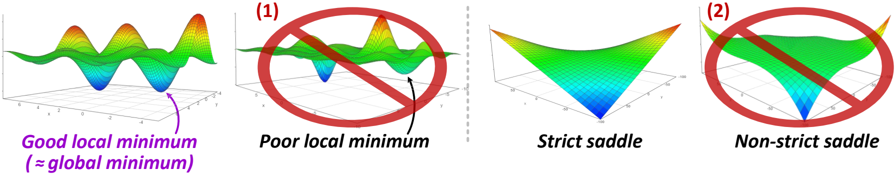
If: **(1)** there are no poor local minima; and **(2)** all saddle points are strict, then GD converges to global min

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

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

Applying Our Trajectory Analysis

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) $\leq e^{-\Omega(c^2 \eta t)}$

Applying Our Trajectory Analysis

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:

Applying Our Trajectory Analysis

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)*

Applying Our Trajectory Analysis

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:

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

Applying Our Trajectory Analysis

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*

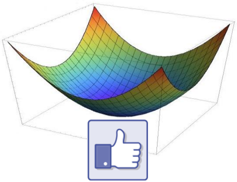
Guarantee of efficient (linear rate) convergence to global min!
Most general guarantee to date for GD efficiently training deep net.

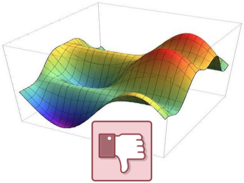
Effect of Depth on Optimization

Effect of Depth on Optimization

Viewpoint of classical learning theory:

- Convex optimization is easier than non-convex

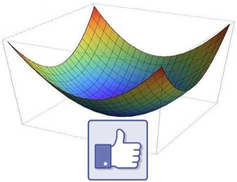


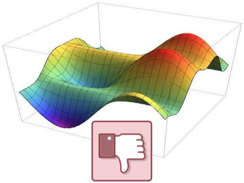


Effect of Depth on Optimization

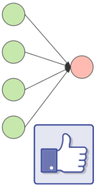
Viewpoint of classical learning theory:

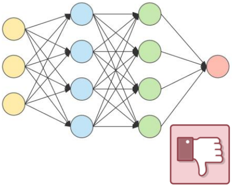
- Convex optimization is easier than non-convex





- Hence depth complicates optimization

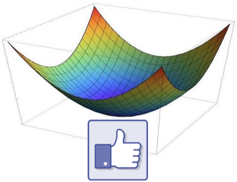


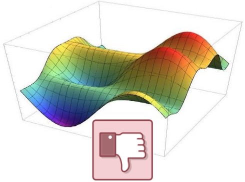


Effect of Depth on Optimization

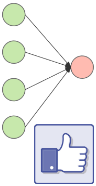
Viewpoint of classical learning theory:

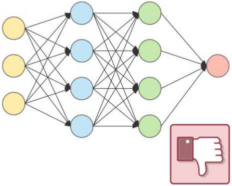
- Convex optimization is easier than non-convex





- Hence depth complicates optimization





Our trajectory analysis reveals: not always true...

Acceleration by Depth

Acceleration by Depth

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

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

Acceleration by Depth

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

Experiment

Acceleration by Depth

End-to-end dynamics for LNN:

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

Claim

$\forall p > 2$, $\exists$ settings where $\ell(\cdot) = \ell_p$ loss and end-to-end dynamics reach global min arbitrarily faster than GD

Experiment

Regression problem from UCI ML Repository ; ℓ_4 loss

Acceleration by Depth

End-to-end dynamics for LNN:

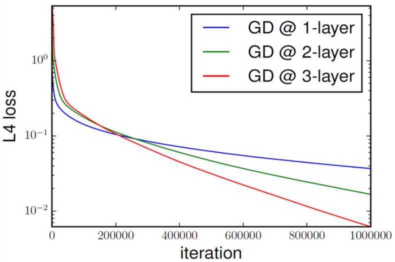
$$vec\left[W_{1:N}(t+1)\right] \leftarrow vec\left[W_{1:N}(t)\right] - \eta \cdot P_{W_{1:N}(t)} \cdot vec\left[\nabla \ell(W_{1:N}(t))\right]$$

Claim

$\forall p > 2$, $\exists$ settings where $\ell(\cdot) = \ell_p$ loss and end-to-end dynamics reach global min arbitrarily faster than GD

Experiment

Regression problem from UCI ML Repository ; ℓ_4 loss



Acceleration by Depth

End-to-end dynamics for LNN:

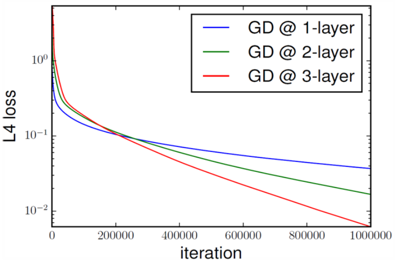
$$vec\left[W_{1:N}(t+1)\right] \leftarrow vec\left[W_{1:N}(t)\right] - \eta \cdot P_{W_{1:N}(t)} \cdot vec\left[\nabla \ell(W_{1:N}(t))\right]$$

Claim

$\forall p > 2$, $\exists$ settings where $\ell(\cdot) = \ell_p$ loss and 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 low rank matrix given subset of entries

Setting: Matrix Completion

Matrix completion: recover low rank matrix given subset of entries

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

Netflix Prize

Setting: Matrix Completion

Matrix completion: recover low rank matrix given subset of entries

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

Netflix Prize

$$\boxed{\min \ rank(W) \quad s.t. \quad W \ agrees \ with \ observations}$$

Setting: Matrix Completion

Matrix completion: recover low rank matrix given subset of entries

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

Netflix Prize

$$\boxed{\min \ rank(W) \quad s.t. \quad W \ agrees \ with \ observations}$$

Convex Programming Approach

Replace rank by nuclear norm:

$$\min \ \|W\|_{nuclear} \quad s.t. \quad W \ agrees \ with \ observations$$

Setting: Matrix Completion

Matrix completion: recover low rank matrix given subset of entries

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

Netflix Prize

$$\boxed{\min \ rank(W) \quad s.t. \quad W \ agrees \ with \ observations}$$

Convex Programming Approach

Replace rank by nuclear norm:

$$\min \ \|W\|_{nuclear} \quad s.t. \quad W \ agrees \ with \ observations$$

Perfectly recovers if observations are sufficiently many [1]

[1] Cf. Candes & Recht 2008

Linear Neural Network $\longleftrightarrow$ "Deep Matrix Factorization"

Deep Learning Approach ("**deep matrix factorization**")

Linear Neural Network $\longleftrightarrow$ "Deep Matrix Factorization"

Deep Learning Approach ("**deep matrix factorization**")

Parameterize solution as LNN and fit observations using GD

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

hidden dims do not necessarily constrain rank

Linear Neural Network $\longleftrightarrow$ "Deep Matrix Factorization"

Deep Learning Approach ("**deep matrix factorization**")

Parameterize solution as LNN and fit observations using GD

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

hidden dims do not necessarily constrain rank

Past Work (Gunasekar et al. 2017)

For depth 2 only:

Linear Neural Network $\longleftrightarrow$ "Deep Matrix Factorization"

Deep Learning Approach ("**deep matrix factorization**")

Parameterize solution as LNN and fit observations using GD

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

hidden dims do not necessarily constrain rank

Past Work (Gunasekar et al. 2017)

For depth 2 only:

- Experiments: recovery often accurate *(w/o explicit regularization)*

Linear Neural Network $\longleftrightarrow$ "Deep Matrix Factorization"

Deep Learning Approach ("**deep matrix factorization**")

Parameterize solution as LNN and fit observations using GD

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

hidden dims do not necessarily constrain rank

Past Work (Gunasekar et al. 2017)

For depth 2 only:

- Experiments: recovery often accurate *(w/o explicit regularization)*
- Conjecture: GD converges to min nuclear norm solution

Linear Neural Network $\longleftrightarrow$ "Deep Matrix Factorization"

Deep Learning Approach ("**deep matrix factorization**")

Parameterize solution as LNN and fit observations using GD

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

hidden dims do not necessarily constrain rank

Past Work (Gunasekar et al. 2017)

For depth 2 only:

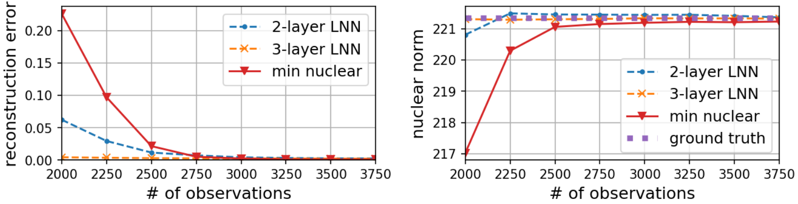
- Experiments: recovery often accurate *(w/o explicit regularization)*
- Conjecture: GD converges to min nuclear norm solution
- Theorem: conjecture holds for certain restricted case

Can the Implicit Regularization Be Captured by Norms?

Can the Implicit Regularization Be Captured by Norms?

Experiment

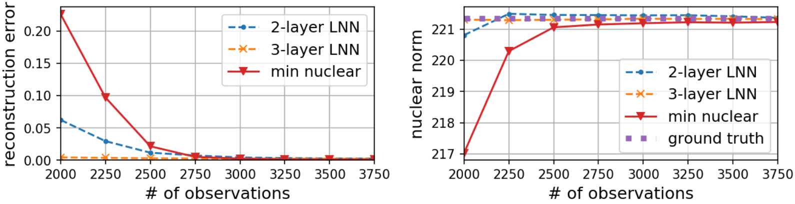
LNN vs. min nuclear norm



Can the Implicit Regularization Be Captured by Norms?

Experiment

LNN vs. min nuclear norm

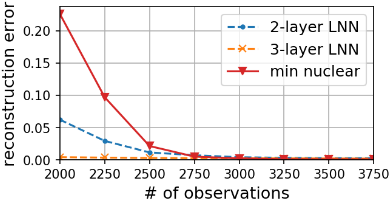


Depth enhanced implicit regularization towards low rank

Can the Implicit Regularization Be Captured by Norms?

Experiment

LNN vs. min nuclear norm



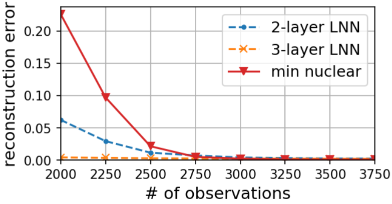
Depth enhanced implicit regularization towards low rank

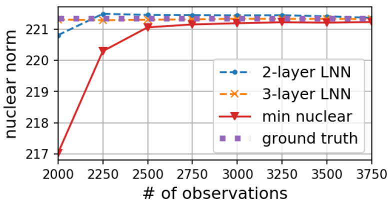
Implicit regularization $\neq$ nuclear norm minimization

Can the Implicit Regularization Be Captured by Norms?

Experiment

LNN vs. min nuclear norm





Depth enhanced implicit regularization towards low rank

Implicit regularization $\neq$ nuclear norm minimization

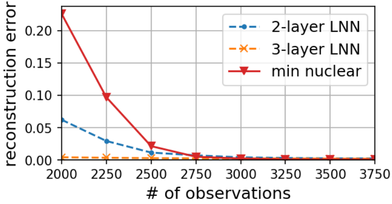
Theorem

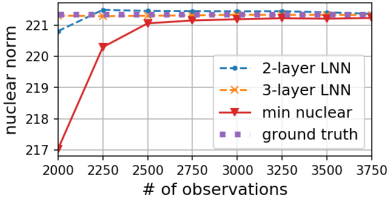
In restricted case where Gunasekar et al. proved depth 2 minimizes nuclear norm, any depth > 2 does so as well

Can the Implicit Regularization Be Captured by Norms?

Experiment

LNN vs. min nuclear norm





Depth enhanced implicit regularization towards low rank

Implicit regularization $\neq$ nuclear norm minimization

Theorem

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

Capturing implicit regularization via single norm may not be possible

Applying Our Trajectory Analysis

Applying Our Trajectory Analysis

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

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

Applying Our Trajectory Analysis

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

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

Theorem

Let $\{\sigma_r(t)\}_r$ be $W_{1:N}(t)$'s singular vals. Then $\sigma_r(t)$ evolves $\propto {\sigma_r}^{2-2/N}(t)$.

Applying Our Trajectory Analysis

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

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

Theorem

Let $\{\sigma_r(t)\}_r$ be $W_{1:N}(t)$'s singular vals. Then $\sigma_r(t)$ evolves $\propto {\sigma_r}^{2-2/N}(t)$.

$\implies$ $\sigma_r(t)$ moves slower when small, faster when large

Applying Our Trajectory Analysis

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

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

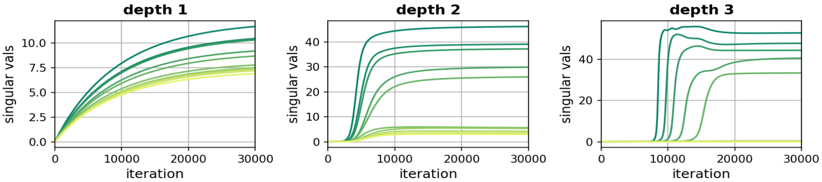
Theorem

Let $\{\sigma_r(t)\}_r$ be $W_{1:N}(t)$'s singular vals. Then $\sigma_r(t)$ evolves $\propto {\sigma_r}^{2-2/N}(t)$.

$\implies$ $\sigma_r(t)$ moves slower when small, faster when large

Experiment

Evolution of singular vals during GD on LNN



Applying Our Trajectory Analysis

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

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

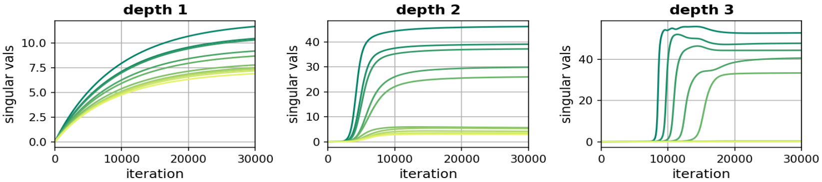
Theorem

Let $\{\sigma_r(t)\}_r$ be $W_{1:N}(t)$'s singular vals. Then $\sigma_r(t)$ evolves $\propto \sigma_r{}^{2-2/N}(t)$.

$\implies$ $\sigma_r(t)$ moves slower when small, faster when large

Experiment

Evolution of singular vals during GD on LNN



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

Outline

Recap

Recap

Perspective

Understanding optimization and generalization in deep learning:

Recap

Perspective

Understanding optimization and generalization in deep learning:

- Language of classical learning theory insufficient

Recap

Perspective

Understanding optimization and generalization in deep learning:

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

Recap

Perspective

Understanding optimization and generalization in deep learning:

- Language of classical learning theory 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 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 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 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 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 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 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 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

Outline

Thank You