# Is Gradient Descent Needed for Generalization in Deep Learning? Matrix Factorization as a Theoretical Testbed

**Nadav Cohen**

Tel Aviv University

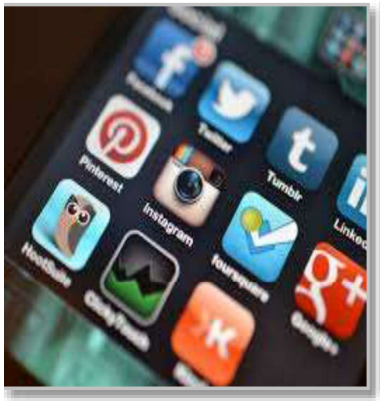
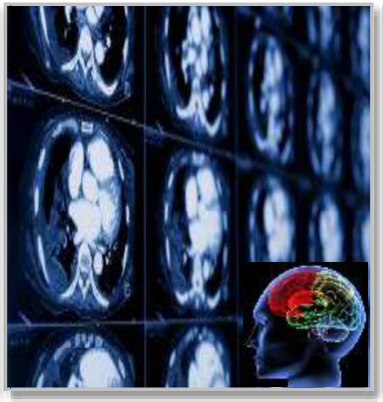
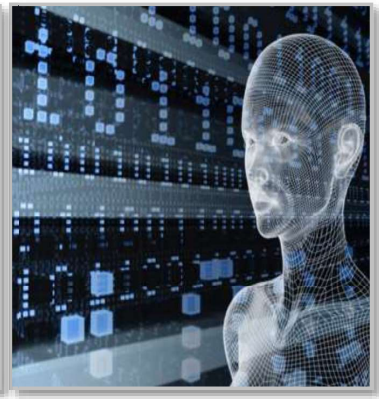
**A Day on Random Matrix Theory and Deep Learning**

*Rome Center on Mathematics for Modeling and Data Sciences (RoMaDS)*

25 September 2025

# Deep Learning

## EVERY INDUSTRY WANTS DEEP LEARNING

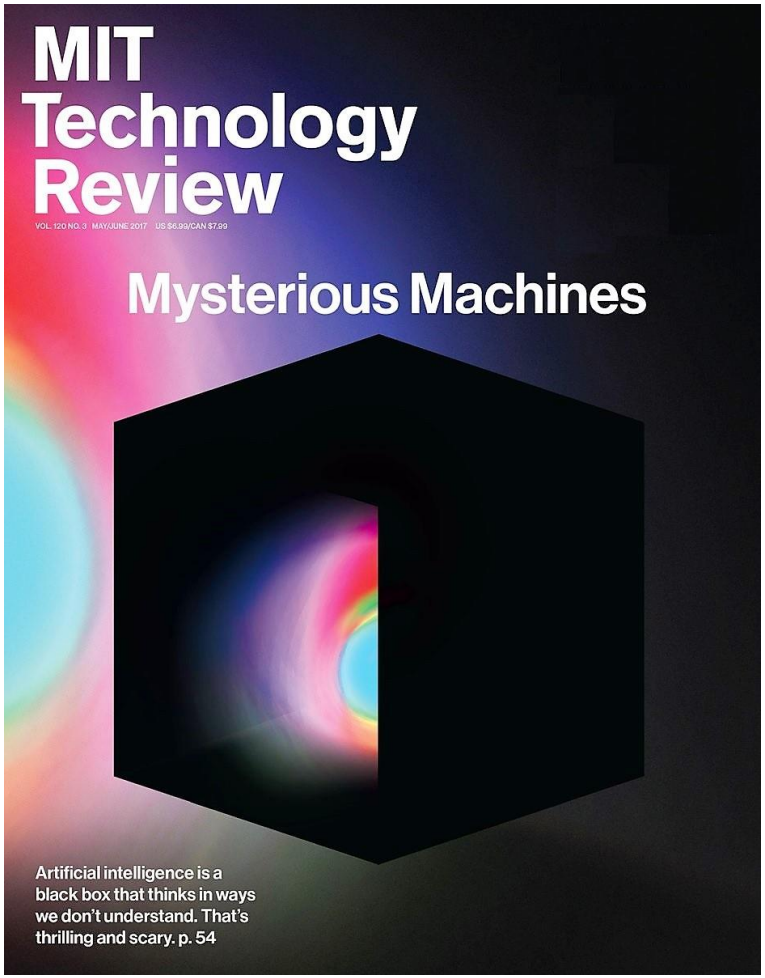
| Cloud Service Provider | Medicine | Media & Entertainment | Security & Defense | Autonomous Machines |
|---|---|---|---|---|
|  |  |  |  |  |
| ➢ Image/Video classification | ➢ Cancer cell detection | ➢ Video captioning | ➢ Face recognition | ➢ Pedestrian detection |
| ➢ Speech recognition | ➢ Diabetic grading | ➢ Content based search | ➢ Video surveillance | ➢ Lane tracking |
| ➢ Natural language processing | ➢ Drug discovery | ➢ Real time translation | ➢ Cyber security | ➢ Recognize traffic sign |

NVIDIA

Source

NVIDIA (`www.slideshare.net/openomics/the-revolution-of-deep-learning`)

# Formal Understanding of Deep Learning is Limited




Intelligent Machines

## The Dark Secret at the Heart of AI

No one really knows how the most advanced algorithms do what they do. That could be a problem.

by Will Knight April 11, 2017

**Last year, a strange self-driving car was released onto the quiet** roads of Monmouth County, New Jersey. The experimental vehicle, developed by researchers at the chip maker Nvidia, didn't look different from other autonomous cars, but it was unlike anything demonstrated by Google, Tesla, or General Motors, and it showed the rising power of artificial intelligence. The car didn't follow a single instruction provided by an engineer or programmer. Instead, it relied entirely on an algorithm that had taught itself to drive by watching a human do it.

# Outline

# Statistical Learning Setup

# Statistical Learning Setup

$\mathcal{X}$ — instance space (e.g. $\{0,1\}^{100\times 100}$ for 100-by-100 binary images)

# Statistical Learning Setup

$\mathcal{X}$ — instance space (e.g. $\{0,1\}^{100\times 100}$ for 100-by-100 binary images)

$\mathcal{Y}$ — label space (e.g. $\mathbb{R}$ for regression or $\{1,\ldots,k\}$ for classification)

# Statistical Learning Setup

$\mathcal{X}$ — instance space (e.g. $\{0,1\}^{100\times 100}$ for 100-by-100 binary images)

$\mathcal{Y}$ — label space (e.g. $\mathbb{R}$ for regression or $\{1,\ldots,k\}$ for classification)

$\mathcal{D}$ — distribution over $\mathcal{X}\times\mathcal{Y}$ (unknown)

# Statistical Learning Setup

$\mathcal{X}$ — instance space (e.g. $\{0,1\}^{100\times 100}$ for 100-by-100 binary images)

$\mathcal{Y}$ — label space (e.g. $\mathbb{R}$ for regression or $\{1,\ldots,k\}$ for classification)

$\mathcal{D}$ — distribution over $\mathcal{X}\times\mathcal{Y}$ (unknown)

$\ell : \mathcal{Y}\times\mathcal{Y} \to \mathbb{R}_{\geq 0}$ — loss func (e.g. $\ell(y,\hat{y}) = (y-\hat{y})^2$ for $\mathcal{Y}=\mathbb{R}$)

# Statistical Learning Setup

$\mathcal{X}$ — instance space (e.g. $\{0,1\}^{100\times 100}$ for 100-by-100 binary images)

$\mathcal{Y}$ — label space (e.g. $\mathbb{R}$ for regression or $\{1,\ldots,k\}$ for classification)

$\mathcal{D}$ — distribution over $\mathcal{X}\times\mathcal{Y}$ (unknown)

$\ell : \mathcal{Y}\times\mathcal{Y} \to \mathbb{R}_{\geq 0}$ — loss func (e.g. $\ell(y,\hat{y}) = (y-\hat{y})^2$ for $\mathcal{Y}=\mathbb{R}$)

**Task**

Given training set $S = \{(X_i, y_i)\}_{i=1}^m$ drawn i.i.d. from $\mathcal{D}$, return predictor $h : \mathcal{X} \to \mathcal{Y}$ that minimizes population loss:

$$L_{\mathcal{D}}(h) := \mathbb{E}_{(X,y)\sim\mathcal{D}}[\ell(y, h(X))]$$

# Statistical Learning Setup

$\mathcal{X}$ — instance space (e.g. $\{0,1\}^{100\times 100}$ for 100-by-100 binary images)

$\mathcal{Y}$ — label space (e.g. $\mathbb{R}$ for regression or $\{1,\ldots,k\}$ for classification)

$\mathcal{D}$ — distribution over $\mathcal{X}\times\mathcal{Y}$ (unknown)

$\ell : \mathcal{Y}\times\mathcal{Y} \to \mathbb{R}_{\geq 0}$ — loss func (e.g. $\ell(y,\hat{y}) = (y-\hat{y})^2$ for $\mathcal{Y}=\mathbb{R}$)

**Task**

Given training set $S = \{(X_i, y_i)\}_{i=1}^m$ drawn i.i.d. from $\mathcal{D}$, return predictor $h : \mathcal{X} \to \mathcal{Y}$ that minimizes population loss:

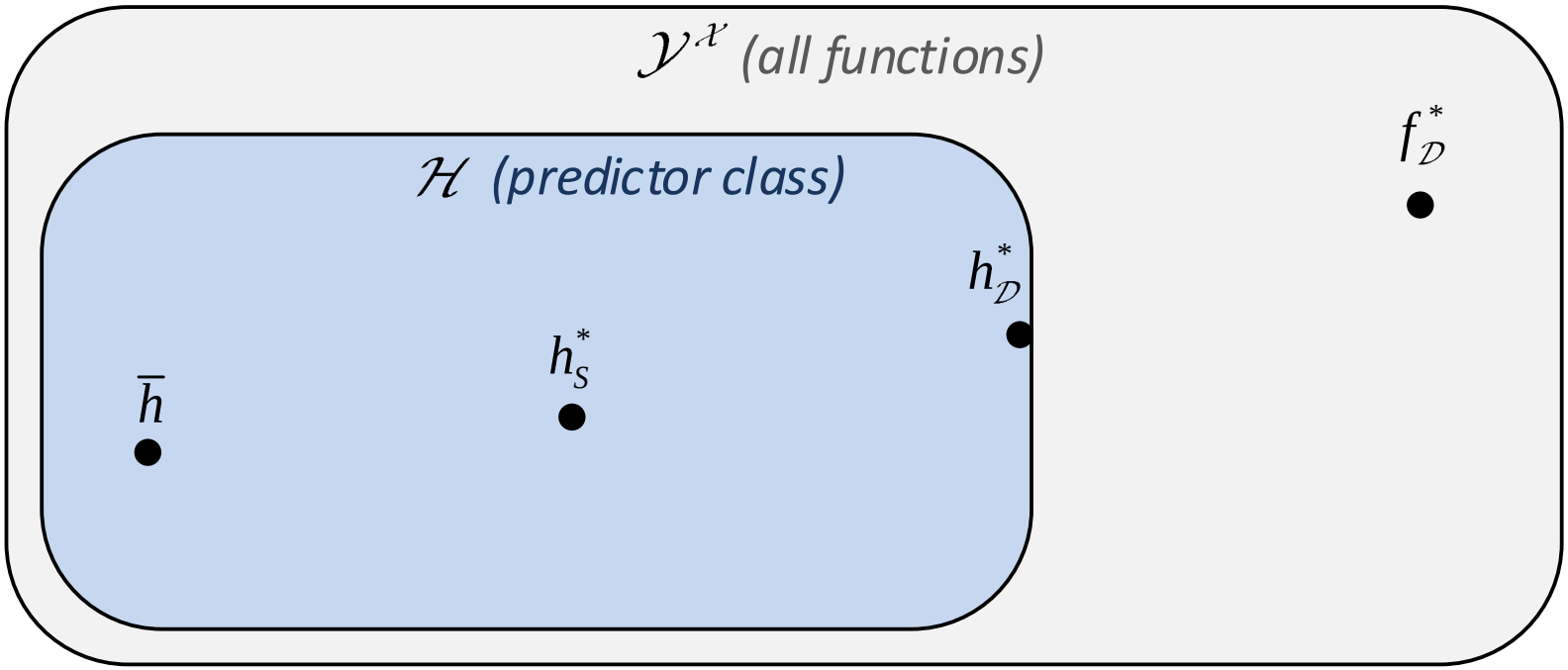
$$L_{\mathcal{D}}(h) := \mathbb{E}_{(X,y)\sim\mathcal{D}}[\ell(y, h(X))]$$

**Approach**

Predetermine predictor class $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$, and return predictor $h \in \mathcal{H}$ that minimizes empirical loss:

$$L_S(h) := \frac{1}{m}\sum_{i=1}^{m} \ell(y_i, h(X_i))$$

# Three Pillars of Statistical Learning Theory: Expressiveness, Generalization and Optimization



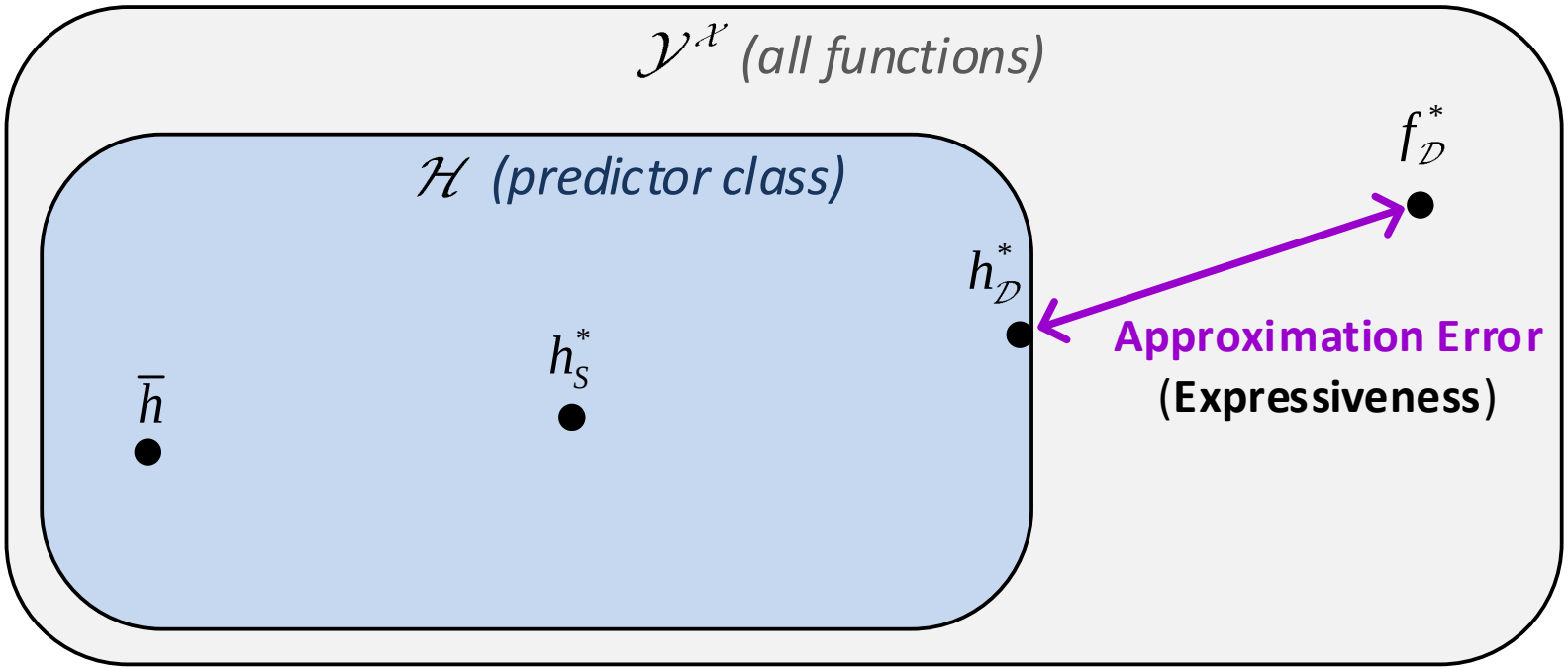

$f^*_{\mathcal{D}}$ — ground truth (minimizer of population loss over $\mathcal{Y}^{\mathcal{X}}$)

$h^*_{\mathcal{D}}$ — optimal predictor (minimizer of population loss over $\mathcal{H}$)

$h^*_S$ — empirically optimal predictor (minimizer of empirical loss over $\mathcal{H}$)

$\bar{h}$ — returned predictor

# Three Pillars of Statistical Learning Theory: Expressiveness, Generalization and Optimization



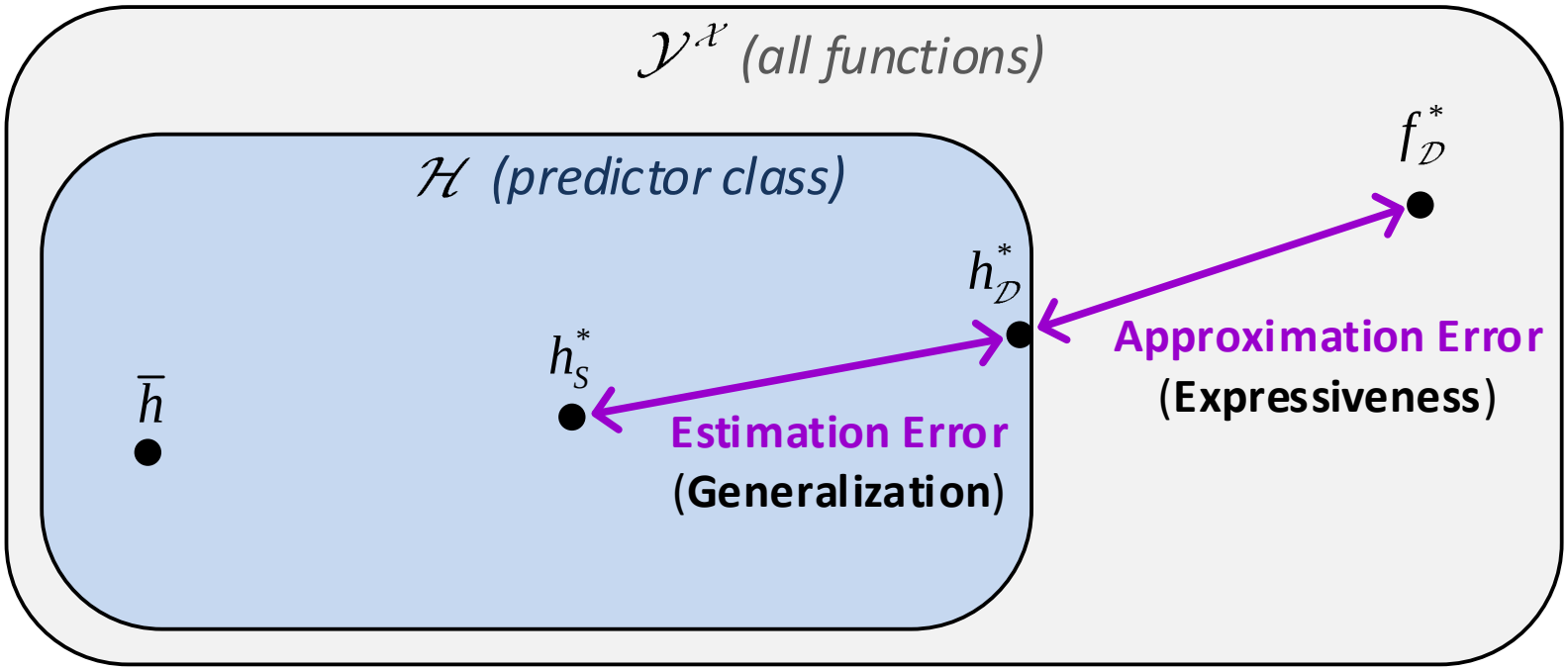

$f^*_{\mathcal{D}}$ — ground truth (minimizer of population loss over $\mathcal{Y}^{\mathcal{X}}$)

$h^*_{\mathcal{D}}$ — optimal predictor (minimizer of population loss over $\mathcal{H}$)

$h^*_S$ — empirically optimal predictor (minimizer of empirical loss over $\mathcal{H}$)

$\bar{h}$ — returned predictor

# Three Pillars of Statistical Learning Theory: Expressiveness, Generalization and Optimization



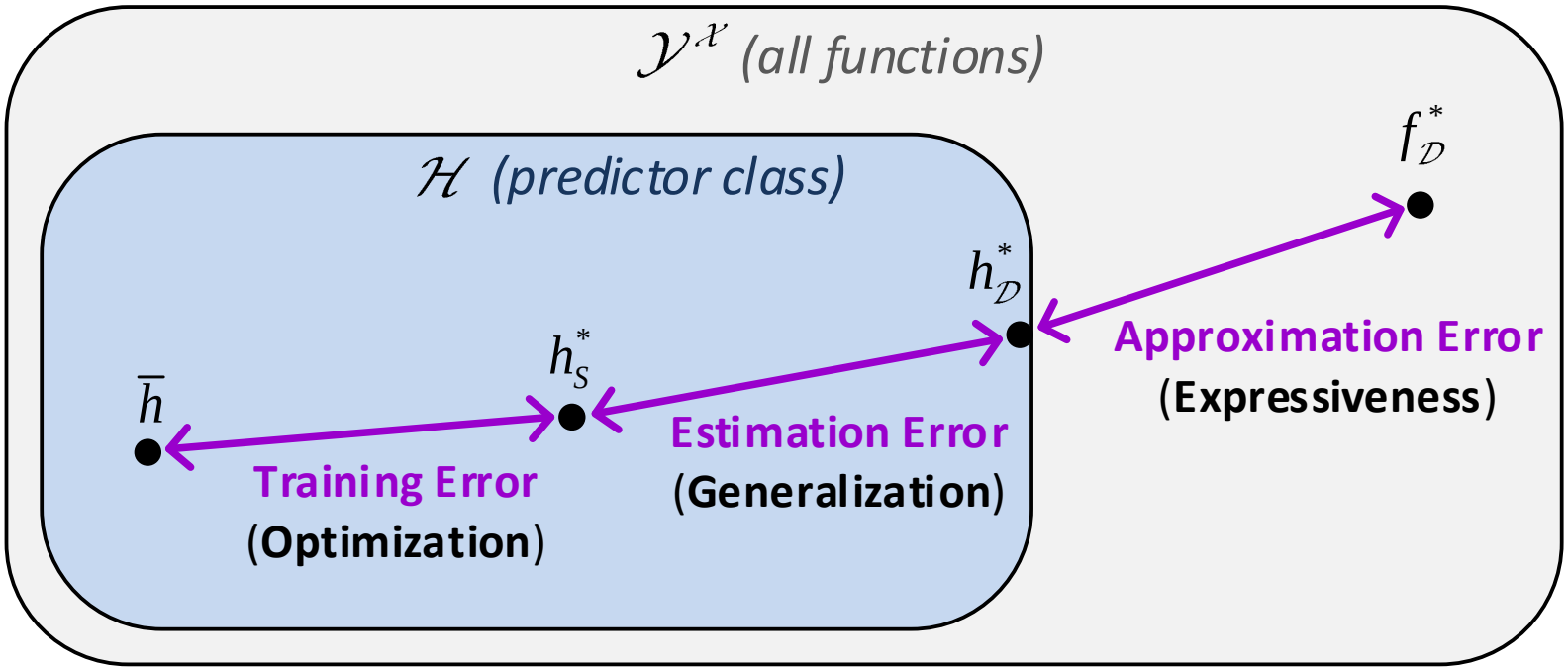

$f^*_{\mathcal{D}}$ — ground truth (minimizer of population loss over $\mathcal{Y}^{\mathcal{X}}$)

$h^*_{\mathcal{D}}$ — optimal predictor (minimizer of population loss over $\mathcal{H}$)

$h^*_S$ — empirically optimal predictor (minimizer of empirical loss over $\mathcal{H}$)

$\bar{h}$ — returned predictor

# Three Pillars of Statistical Learning Theory: Expressiveness, Generalization and Optimization




$f^*_{\mathcal{D}}$ — ground truth (minimizer of population loss over $\mathcal{Y}^{\mathcal{X}}$)

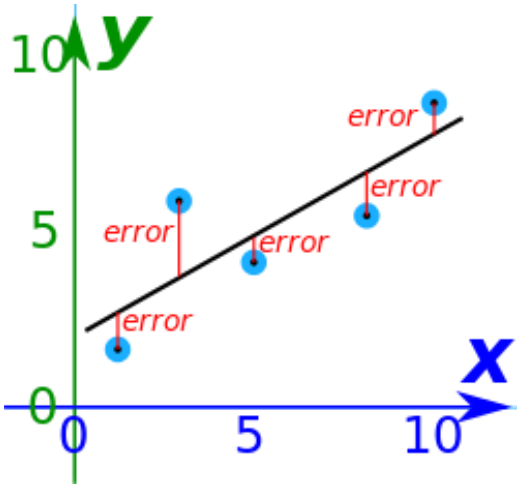
$h^*_{\mathcal{D}}$ — optimal predictor (minimizer of population loss over $\mathcal{H}$)

$h^*_S$ — empirically optimal predictor (minimizer of empirical loss over $\mathcal{H}$)

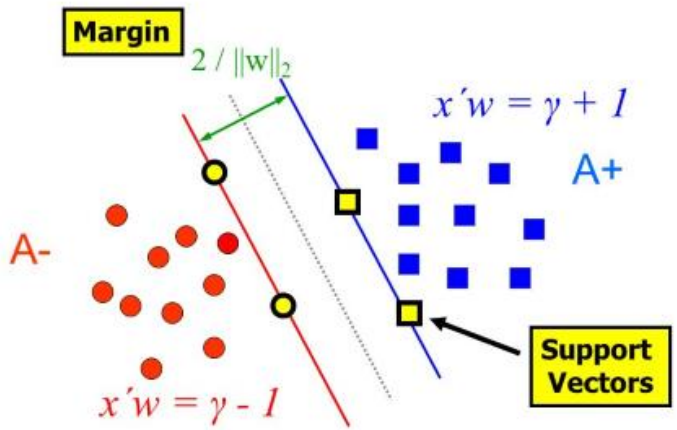
$\bar{h}$ — returned predictor

# Classical Machine Learning

Linear predictor classes

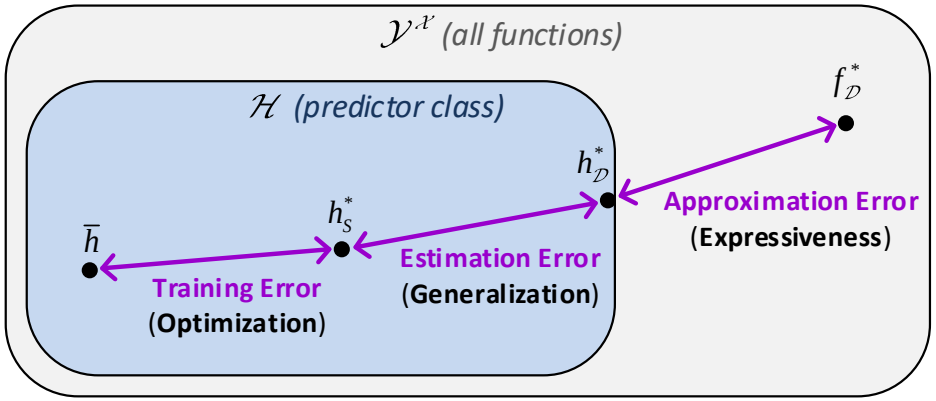



*Least Squares Regression*

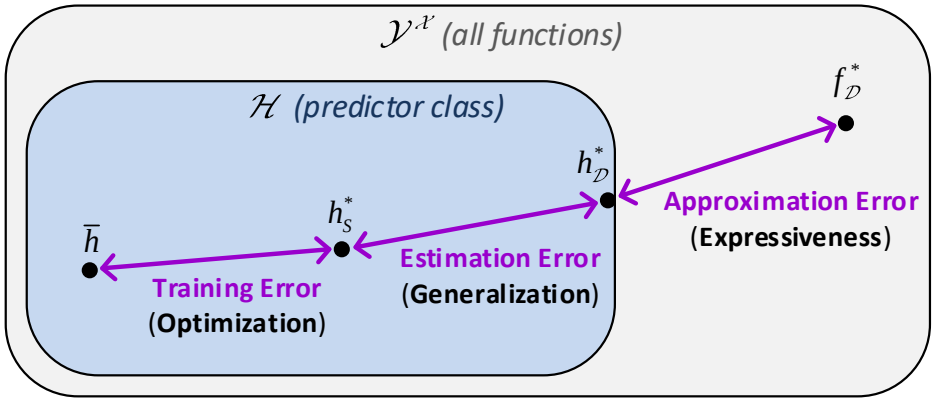



*Support Vector Machine*

# Classical Machine Learning — Three Pillars



# Classical Machine Learning — Three Pillars

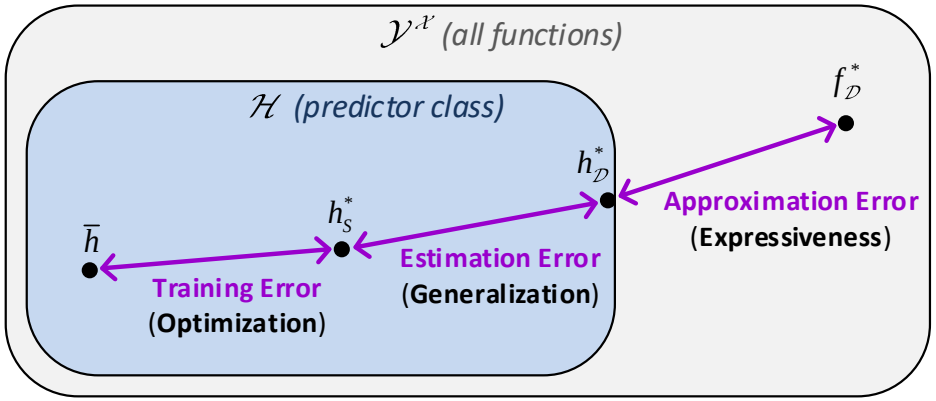



## Optimization

Empirical loss minimization is a convex program:

$$\bar{h} \approx h^*_S \quad (\text{ training err} \approx 0 \text{ })$$

# Classical Machine Learning — Three Pillars




## Optimization

Empirical loss minimization is a convex program:

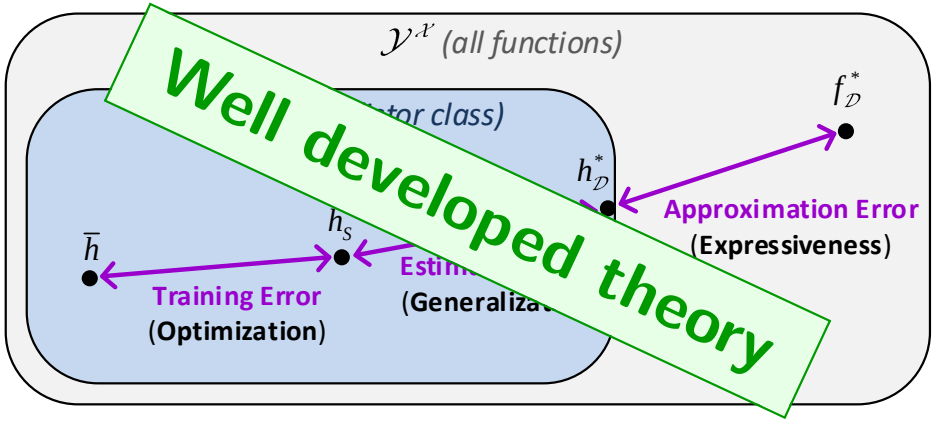
$$\bar{h} \approx h^*_S \quad (\text{ training err} \approx 0 \text{ })$$

## Expressiveness & Generalization

Bias-variance trade-off:

| $\mathcal{H}$ | approximation err | estimation err |
|---|---|---|
| *expands* | ↘ | ↗ |
| *shrinks* | ↗ | ↘ |

# Classical Machine Learning — Three Pillars




## Optimization

Empirical loss minimization is a convex program:

$$\bar{h} \approx h^*_S \quad (\text{ training err} \approx 0 \text{ })$$

## Expressiveness & Generalization

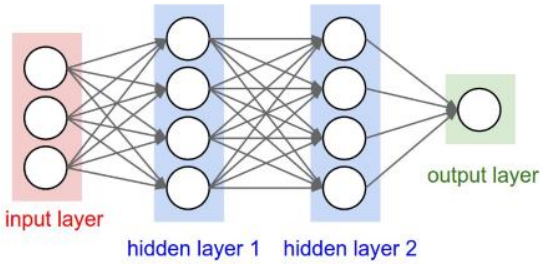
Bias-variance trade-off:

| $\mathcal{H}$ | approximation err | estimation err |
|---|---|---|
| *expands* | ↘ | ↗ |
| *shrinks* | ↗ | ↘ |

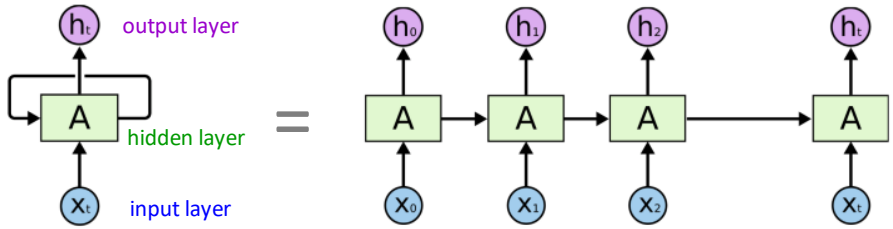
# Deep Learning

Neural networks (NNs) — composite predictor classes

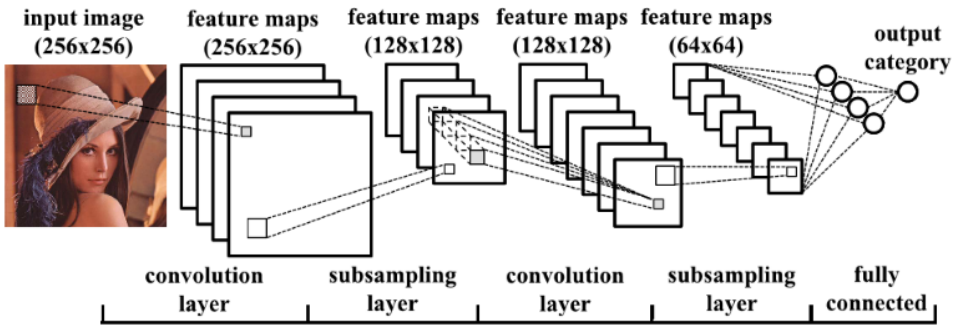
***Fully-Connected Networks***



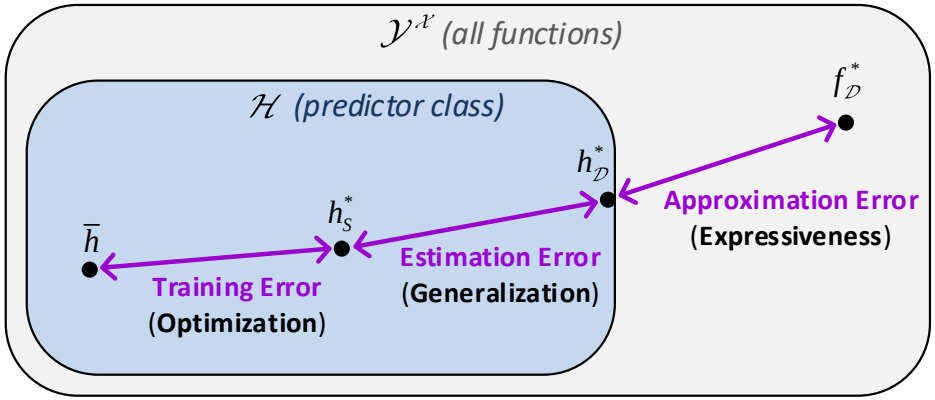

***Recurrent Networks***



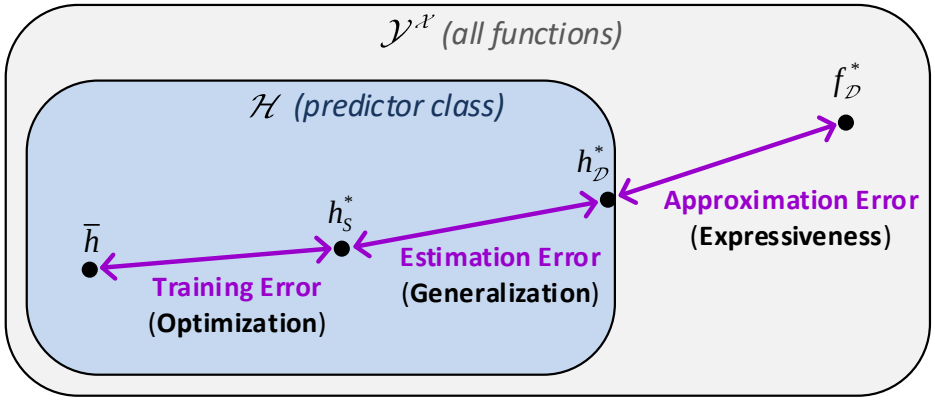

***Convolutional Networks***



# Deep Learning — Three Pillars



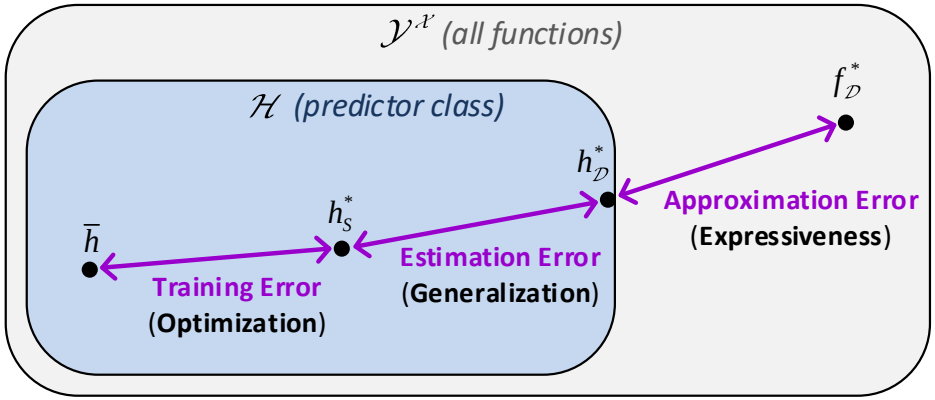
# Deep Learning — Three Pillars




## Optimization

Empirical loss minimization is a non-convex program:

# Deep Learning — Three Pillars

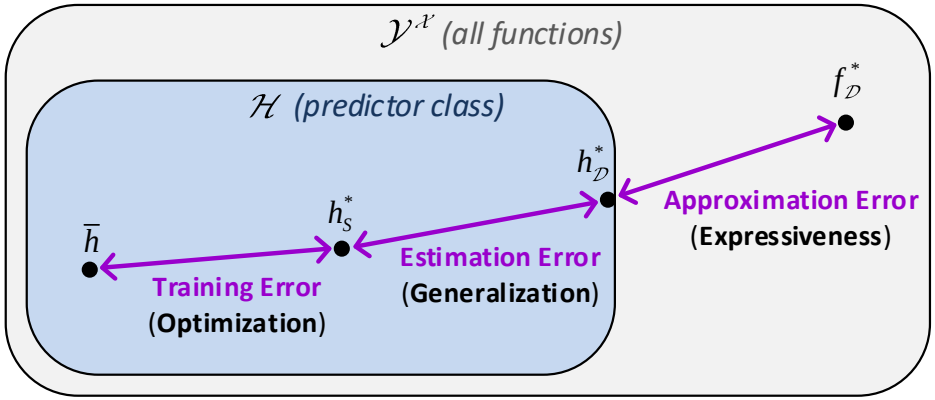



## Optimization

Empirical loss minimization is a non-convex program:

- $h^*_S$ is not unique — many predictors have low empirical loss

# Deep Learning — Three Pillars

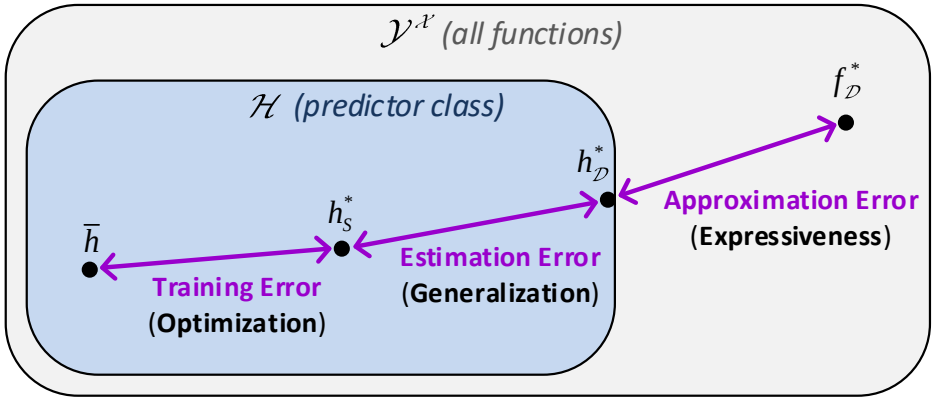



## Optimization

Empirical loss minimization is a non-convex program:

- $h^*_S$ is not unique — many predictors have low empirical loss
- Gradient descent (GD) somehow reaches one of these

# Deep Learning — Three Pillars




## Optimization

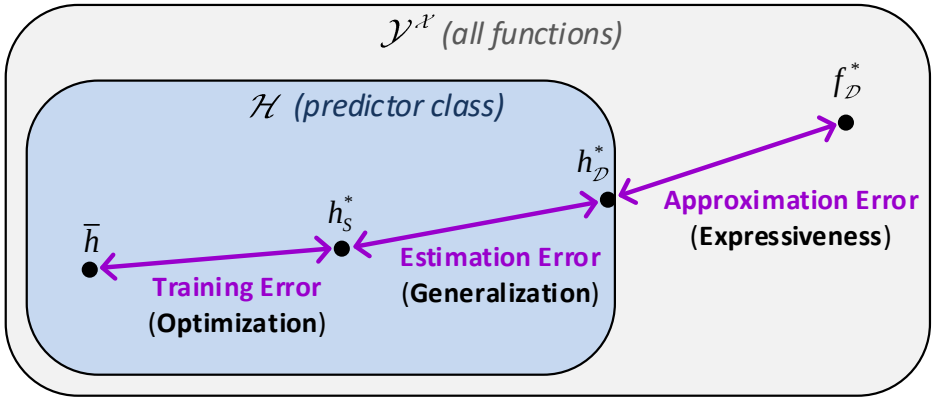
Empirical loss minimization is a non-convex program:

- $h^*_S$ is not unique — many predictors have low empirical loss
- Gradient descent (GD) somehow reaches one of these

## Expressiveness & Generalization

Vast difference from classical ML:

# Deep Learning — Three Pillars




## Optimization

Empirical loss minimization is a non-convex program:

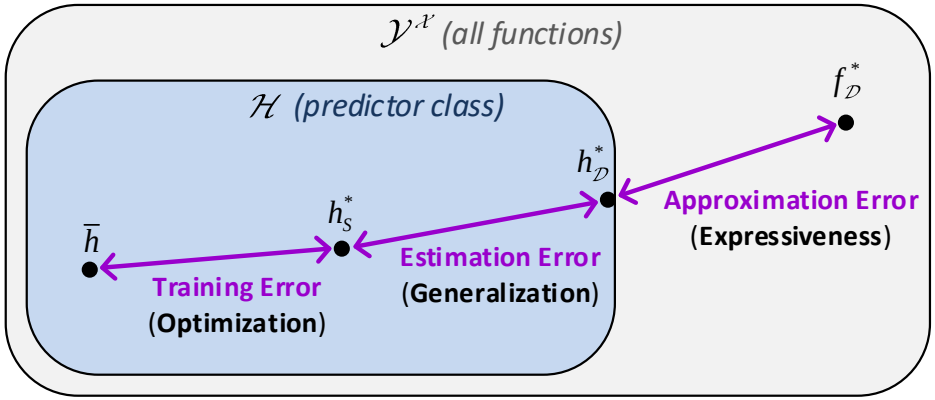
- $h^*_S$ is not unique — many predictors have low empirical loss
- Gradient descent (GD) somehow reaches one of these

## Expressiveness & Generalization

Vast difference from classical ML:

- Some low empirical loss predictors generalize well, others don't

# Deep Learning — Three Pillars




## Optimization

Empirical loss minimization is a non-convex program:

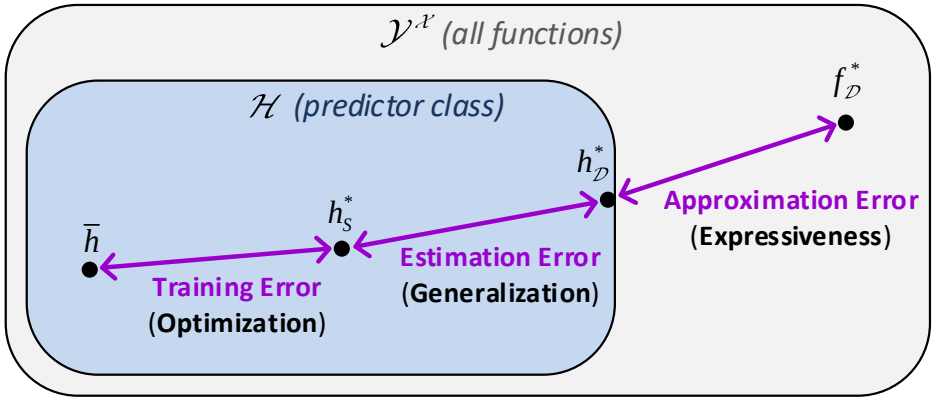
- $h^*_S$ is not unique — many predictors have low empirical loss
- Gradient descent (GD) somehow reaches one of these

## Expressiveness & Generalization

Vast difference from classical ML:

- Some low empirical loss predictors generalize well, others don't
- With "natural" data, predictor returned by GD often generalizes well

# Deep Learning — Three Pillars




## Optimization

Empirical loss minimization is a non-convex program:

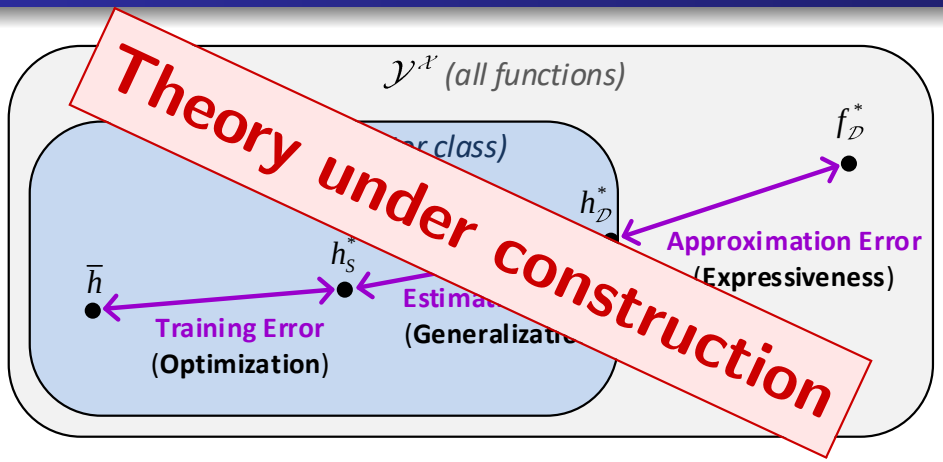
- $h^*_S$ is not unique — many predictors have low empirical loss
- Gradient descent (GD) somehow reaches one of these

## Expressiveness & Generalization

Vast difference from classical ML:

- Some low empirical loss predictors generalize well, others don't
- With "natural" data, predictor returned by GD often generalizes well
- Expanding $\mathcal{H}$ reduces approximation err, but also estimation err!

# Deep Learning — Three Pillars




## Optimization

Empirical loss minimization is a non-convex program:

- $h^*_S$ is not unique — many predictors have low empirical loss
- Gradient descent (GD) somehow reaches one of these

## Expressiveness & Generalization

Vast difference from classical ML:

- Some low empirical loss predictors generalize well, others don't
- With "natural" data, predictor returned by GD often generalizes well
- Expanding $\mathcal{H}$ reduces approximation err, but also estimation err!

# Outline

# Sources

**Do Neural Networks Need Gradient Descent to Generalize? A Theoretical Study**

Alexander + Slutzky + Ran-Milo + **C**

*NeurIPS 2025 (to appear)*

**Implicit Regularization in Deep Matrix Factorization**

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

*NeurIPS 2019*

# Outline

# Generalization via Bias-Variance Tradeoff

# Generalization via Bias-Variance Tradeoff

Classically, generalization is understood via the bias-variance tradeoff:

# Generalization via Bias-Variance Tradeoff

Classically, generalization is understood via the bias-variance tradeoff:

# Generalization via Bias-Variance Tradeoff

Classically, generalization is understood via the bias-variance tradeoff:


Tradeoff can be controlled through:

# Generalization via Bias-Variance Tradeoff

Classically, generalization is understood via the bias-variance tradeoff:


Tradeoff can be controlled through:

- Limiting model size

# Generalization via Bias-Variance Tradeoff

Classically, generalization is understood via the bias-variance tradeoff:


Tradeoff can be controlled through:

- Limiting model size
- Adding regularization (e.g. $\ell_2$ penalty)

# Generalization in Deep Learning

# Generalization in Deep Learning

NNs trained with GD generalize well despite:

# Generalization in Deep Learning

NNs trained with GD generalize well despite:

- Model size $\gg$ training set size

# Generalization in Deep Learning

NNs trained with GD generalize well despite:

- Model size $\gg$ training set size
- No explicit regularization

# Generalization in Deep Learning

NNs trained with GD generalize well despite:

- Model size $\gg$ training set size
- No explicit regularization

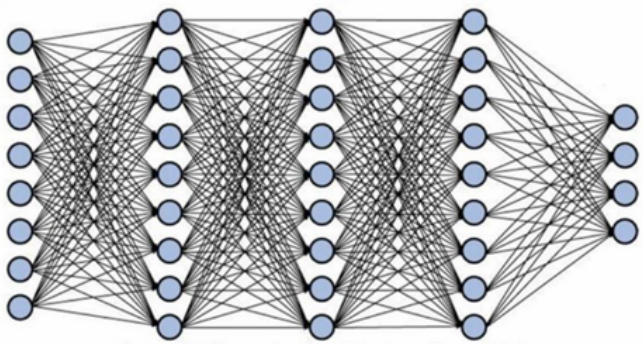
} Overfitting solutions exist

# Generalization in Deep Learning

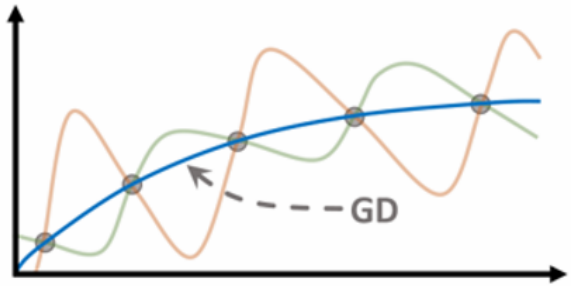
NNs trained with GD generalize well despite:

- Model size $\gg$ training set size
- No explicit regularization

} Overfitting solutions exist



$\gg$

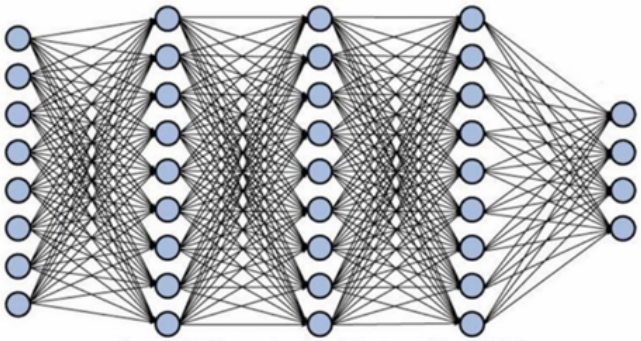



# Generalization in Deep Learning

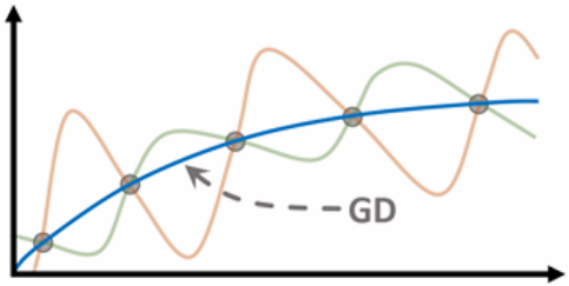
NNs trained with GD generalize well despite:

- Model size $\gg$ training set size
- No explicit regularization

} Overfitting solutions exist



$\gg$


Why does this happen?

# Generalization in Deep Learning: Hypotheses

# Generalization in Deep Learning: Hypotheses

## Conventional Wisdom

# Generalization in Deep Learning: Hypotheses

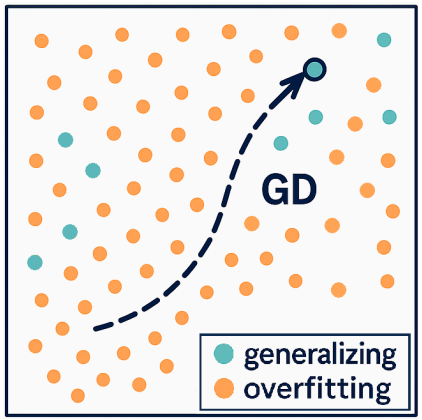
## Conventional Wisdom

**Implicit bias**:

# Generalization in Deep Learning: Hypotheses

## Conventional Wisdom

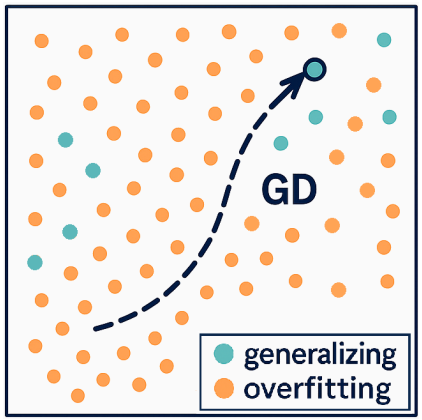
**Implicit bias**: among predictors fitting training set, those that generalize well are special, yet GD finds one



# Generalization in Deep Learning: Hypotheses

## Conventional Wisdom

**Implicit bias**: among predictors fitting training set, those that generalize well are special, yet GD finds one

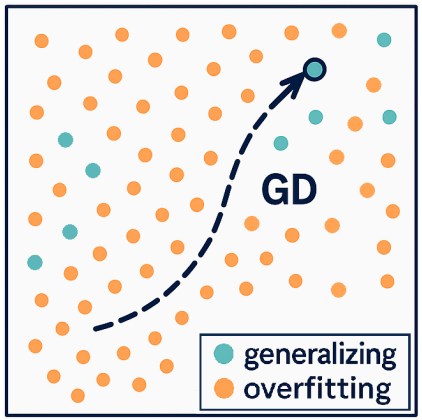



## Recent View

# Generalization in Deep Learning: Hypotheses

## Conventional Wisdom

**Implicit bias**: among predictors fitting training set, those that generalize well are special, yet GD finds one



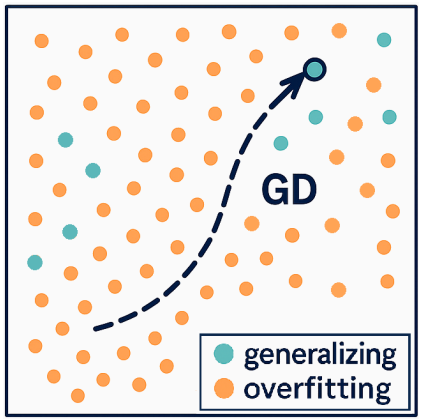

## Recent View

**Volume hypothesis**:

# Generalization in Deep Learning: Hypotheses

## Conventional Wisdom

**Implicit bias**: among predictors fitting training set, those that generalize well are special, yet GD finds one




## Recent View

**Volume hypothesis**: most predictors fitting training set generalize well, and GD merely finds typical one

# Conflicting Evidence

# Conflicting Evidence

Implicit bias:

# Conflicting Evidence

Implicit bias:

- Many works showed properties of GD ensure good generalization (Soudry et al., 2018; Gunasekar et al., 2018b; Kou et al., 2023; Li et al., 2024)

# Conflicting Evidence

Implicit bias:

- Many works showed properties of GD ensure good generalization (Soudry et al., 2018; Gunasekar et al., 2018b; Kou et al., 2023; Li et al., 2024)

Volume hypothesis:

# Conflicting Evidence

Implicit bias:

- Many works showed properties of GD ensure good generalization (Soudry et al., 2018; Gunasekar et al., 2018b; Kou et al., 2023; Li et al., 2024)

Volume hypothesis:

- Observed in wide and deep NNs (Chiang et al., 2022; Hanin and Zlokapa, 2023; Buzaglo et al., 2024; Harel et al., 2024)

# Conflicting Evidence

Implicit bias:

- Many works showed properties of GD ensure good generalization (Soudry et al., 2018; Gunasekar et al., 2018b; Kou et al., 2023; Li et al., 2024)

Volume hypothesis:

- Observed in wide and deep NNs (Chiang et al., 2022; Hanin and Zlokapa, 2023; Buzaglo et al., 2024; Harel et al., 2024)
- Recent work found GD to generalize better than typical solutions in wide NNs (Peleg and Hein, 2024)

# Conflicting Evidence

Implicit bias:

- Many works showed properties of GD ensure good generalization (Soudry et al., 2018; Gunasekar et al., 2018b; Kou et al., 2023; Li et al., 2024)

Volume hypothesis:

- Observed in wide and deep NNs (Chiang et al., 2022; Hanin and Zlokapa, 2023; Buzaglo et al., 2024; Harel et al., 2024)
- Recent work found GD to generalize better than typical solutions in wide NNs (Peleg and Hein, 2024)

Conflicting evidence

# Conflicting Evidence

Implicit bias:

- Many works showed properties of GD ensure good generalization (Soudry et al., 2018; Gunasekar et al., 2018b; Kou et al., 2023; Li et al., 2024)

Volume hypothesis:

- Observed in wide and deep NNs (Chiang et al., 2022; Hanin and Zlokapa, 2023; Buzaglo et al., 2024; Harel et al., 2024)
- Recent work found GD to generalize better than typical solutions in wide NNs (Peleg and Hein, 2024)

Conflicting evidence $\implies$ situation remains unclear

# Outline

# Matrix Sensing

**Matrix sensing**:

# Matrix Sensing

**Matrix sensing**:

- Fundamental problem in science and engineering (Fazel et al., 2008; Recht et al., 2010; Fornasier et al., 2011; Cai and Wei, 2018)

# Matrix Sensing

**Matrix sensing**:

- Fundamental problem in science and engineering (Fazel et al., 2008; Recht et al., 2010; Fornasier et al., 2011; Cai and Wei, 2018)
- **Task**: recover low rank ground truth matrix $W^* \in \mathbb{R}^{m,m'}$ given linear measurements $y_i$ corresponding to measurement matrices $A_i$

$$y_i = \langle A_i, W^* \rangle$$

# Matrix Sensing

**Matrix sensing**:

- Fundamental problem in science and engineering (Fazel et al., 2008; Recht et al., 2010; Fornasier et al., 2011; Cai and Wei, 2018)
- **Task**: recover low rank ground truth matrix $W^* \in \mathbb{R}^{m,m'}$ given linear measurements $y_i$ corresponding to measurement matrices $A_i$

$$y_i = \langle A_i, W^* \rangle$$

**Matrix Completion**:

# Matrix Sensing

**Matrix sensing**:

- Fundamental problem in science and engineering (Fazel et al., 2008; Recht et al., 2010; Fornasier et al., 2011; Cai and Wei, 2018)
- **Task**: recover low rank ground truth matrix $W^* \in \mathbb{R}^{m,m'}$ given linear measurements $y_i$ corresponding to measurement matrices $A_i$

$$y_i = \langle A_i, W^* \rangle$$

**Matrix Completion**:

- Important special case (Keshavan et al., 2009; Candes and Tao, 2010; Mazumder et al., 2010; Candes and Recht, 2012)

# Matrix Sensing

**Matrix sensing**:

- Fundamental problem in science and engineering (Fazel et al., 2008; Recht et al., 2010; Fornasier et al., 2011; Cai and Wei, 2018)
- **Task**: recover low rank ground truth matrix $W^* \in \mathbb{R}^{m,m'}$ given linear measurements $y_i$ corresponding to measurement matrices $A_i$

$$y_i = \langle A_i, W^* \rangle$$

**Matrix Completion**:

- Important special case (Keshavan et al., 2009; Candes and Tao, 2010; Mazumder et al., 2010; Candes and Recht, 2012)
- **Task**: recover $W^*$ given subset of its entries

# Matrix Sensing

**Matrix sensing**:

- Fundamental problem in science and engineering (Fazel et al., 2008; Recht et al., 2010; Fornasier et al., 2011; Cai and Wei, 2018)
- **Task**: recover low rank ground truth matrix $W^* \in \mathbb{R}^{m,m'}$ given linear measurements $y_i$ corresponding to measurement matrices $A_i$

$$y_i = \langle A_i, W^* \rangle$$

**Matrix Completion**:

- Important special case (Keshavan et al., 2009; Candes and Tao, 2010; Mazumder et al., 2010; Candes and Recht, 2012)
- **Task**: recover $W^*$ given subset of its entries

# Generalization in Matrix Sensing

# Generalization in Matrix Sensing

**Training loss**:

# Generalization in Matrix Sensing

**Training loss**: MSE over measurement matrices $A_i$

$$\mathcal{L}_{\text{train}}(W) := \frac{1}{n} \sum_{i=1}^{n} \big( \langle A_i, W \rangle - \langle A_i, W^* \rangle \big)^2$$

# Generalization in Matrix Sensing

**Training loss**: MSE over measurement matrices $A_i$

$$\mathcal{L}_{\text{train}}(W) := \frac{1}{n} \sum_{i=1}^{n} \big( \langle A_i, W \rangle - \langle A_i, W^* \rangle \big)^2$$

**Generalization loss**:

# Generalization in Matrix Sensing

**Training loss**: MSE over measurement matrices $A_i$

$$\mathcal{L}_{\text{train}}(W) := \frac{1}{n} \sum_{i=1}^{n} \big( \langle A_i,\, W \rangle - \langle A_i,\, W^* \rangle \big)^2$$

**Generalization loss**: MSE over matrices orthogonal to $A_i$

$$\mathcal{L}_{\text{gen}}(W) := \frac{1}{|\mathcal{B}|} \sum_{A \in \mathcal{B}} \big( \langle A,\, W \rangle - \langle A,\, W^* \rangle \big)^2$$

# Generalization in Matrix Sensing

**Training loss**: MSE over measurement matrices $A_i$

$$\mathcal{L}_{\text{train}}(W) := \frac{1}{n} \sum_{i=1}^{n} \big( \langle A_i,\, W \rangle - \langle A_i,\, W^* \rangle \big)^2$$

**Generalization loss**: MSE over matrices orthogonal to $A_i$

$$\mathcal{L}_{\text{gen}}(W) := \frac{1}{|\mathcal{B}|} \sum_{A \in \mathcal{B}} \big( \langle A,\, W \rangle - \langle A,\, W^* \rangle \big)^2$$

Orthonormal basis for $\text{span}\{A_1, \ldots, A_n\}^\perp$

# Matrix Factorization

**Matrix factorization (MF)**:

# Matrix Factorization

**Matrix factorization** (**MF**):

- Common testbed in deep learning theory (Gunasekar et al., 2017a, 2018a; Li et al., 2020; Chou et al., 2024)

# Matrix Factorization

**Matrix factorization** (**MF**):

- Common testbed in deep learning theory (Gunasekar et al., 2017a, 2018a; Li et al., 2020; Chou et al., 2024)
- **Model**: parameterize matrix sensing solution as product of matrices, possibly with pointwise nonlinearity

$$W = W_L \, \sigma(W_{L-1} \, \sigma(W_{L-2} \cdots \sigma(W_1)) \cdots)$$

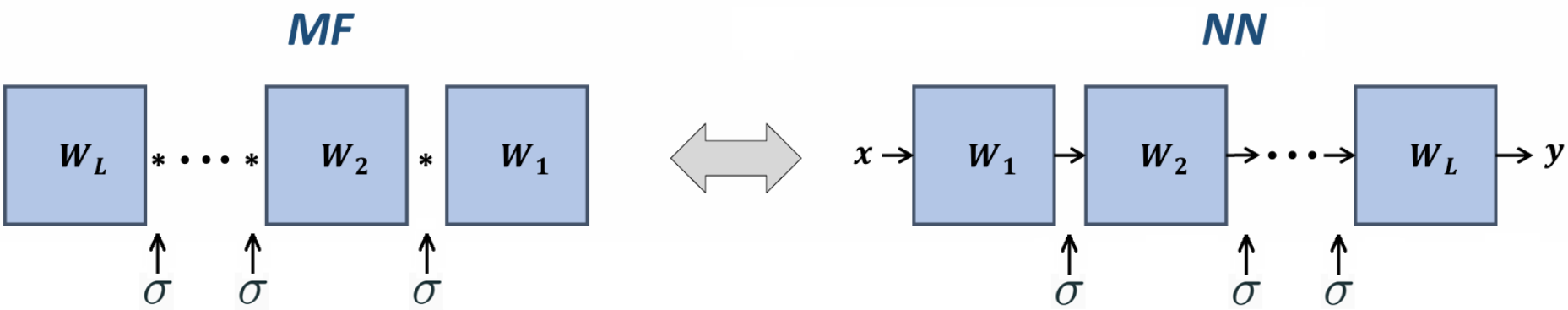
# Matrix Factorization

**Matrix factorization** (**MF**):

- Common testbed in deep learning theory (Gunasekar et al., 2017a, 2018a; Li et al., 2020; Chou et al., 2024)
- **Model**: parameterize matrix sensing solution as product of matrices, possibly with pointwise nonlinearity

$$W = W_L\,\sigma(W_{L-1}\,\sigma(W_{L-2}\cdots\sigma(W_1))\cdots)$$

MFs are analogous to NNs:



# Training Algorithms for Matrix Factorization

# Training Algorithms for Matrix Factorization

**GD**:

# Training Algorithms for Matrix Factorization

**GD**:

Init $W_1^{(0)}, \ldots, W_L^{(0)}$ near zero,

# Training Algorithms for Matrix Factorization

**GD**:

Init $W_1^{(0)}, \ldots, W_L^{(0)}$ near zero, and then update:

$$W_j^{(t+1)} \leftarrow W_j^{(t)} - \eta \frac{\partial}{\partial W_j} \mathcal{L}_{\text{train}} \left( W_1^{(t)}, \ldots, W_L^{(t)} \right)$$

# Training Algorithms for Matrix Factorization

**GD**:

Init $W_1^{(0)}, \ldots, W_L^{(0)}$ near zero, and then update:

$$W_j^{(t+1)} \leftarrow W_j^{(t)} - \eta \frac{\partial}{\partial W_j} \mathcal{L}_{\text{train}} \left( W_1^{(t)}, \ldots, W_L^{(t)} \right)$$

Learning rate

# Training Algorithms for Matrix Factorization

**GD**:

Init $W_1^{(0)}, \ldots, W_L^{(0)}$ near zero, and then update:

$$W_j^{(t+1)} \leftarrow W_j^{(t)} - \eta \frac{\partial}{\partial W_j} \mathcal{L}_{\text{train}} \left( W_1^{(t)}, \ldots, W_L^{(t)} \right)$$

Learning rate

**Guess and Check** (**G&C**):

# Training Algorithms for Matrix Factorization

**GD**:

Init $W_1^{(0)}, \ldots, W_L^{(0)}$ near zero, and then update:

$$W_j^{(t+1)} \leftarrow W_j^{(t)} - \eta \frac{\partial}{\partial W_j} \mathcal{L}_{\text{train}} \left( W_1^{(t)}, \ldots, W_L^{(t)} \right)$$

Learning rate

**Guess and Check** (**G&C**):

Given a distribution $\mathcal{P}(\cdot)$ over $W_1, \ldots, W_L$ and a threshold $\epsilon_{\text{train}} > 0$:

# Training Algorithms for Matrix Factorization

**GD**:

Init $W_1^{(0)}, \ldots, W_L^{(0)}$ near zero, and then update:

$$W_j^{(t+1)} \leftarrow W_j^{(t)} - \eta \frac{\partial}{\partial W_j} \mathcal{L}_{\text{train}} \left( W_1^{(t)}, \ldots, W_L^{(t)} \right)$$

Learning rate

**Guess and Check** (**G&C**):

Given a distribution $\mathcal{P}(\cdot)$ over $W_1, \ldots, W_L$ and a threshold $\epsilon_{\text{train}} > 0$:

- Draw from $\mathcal{P}(\cdot)$ until $\mathcal{L}_{\text{train}}(W_1, \ldots, W_L) < \epsilon_{\text{train}}$

# Training Algorithms for Matrix Factorization

**GD**:

Init $W_1^{(0)}, \ldots, W_L^{(0)}$ near zero, and then update:

$$W_j^{(t+1)} \leftarrow W_j^{(t)} - \eta \frac{\partial}{\partial W_j} \mathcal{L}_{\text{train}} \left( W_1^{(t)}, \ldots, W_L^{(t)} \right)$$

Learning rate

**Guess and Check** (**G&C**):

Given a distribution $\mathcal{P}(\cdot)$ over $W_1, \ldots, W_L$ and a threshold $\epsilon_{\text{train}} > 0$:

- Draw from $\mathcal{P}(\cdot)$ until $\mathcal{L}_{\text{train}}(W_1, \ldots, W_L) < \epsilon_{\text{train}}$
- Equivalent to drawing from $\mathcal{P}\left(\cdot | \mathcal{L}_{\text{train}}(W_1, \ldots, W_L) < \epsilon_{\text{train}}\right)$

# Training Algorithms for Matrix Factorization

**GD**:

Init $W_1^{(0)}, \ldots, W_L^{(0)}$ near zero, and then update:

$$W_j^{(t+1)} \leftarrow W_j^{(t)} - \eta \frac{\partial}{\partial W_j} \mathcal{L}_{\text{train}} \left( W_1^{(t)}, \ldots, W_L^{(t)} \right)$$

Learning rate

**Guess and Check** (**G&C**):

Given a distribution $\mathcal{P}(\cdot)$ over $W_1, \ldots, W_L$ and a threshold $\epsilon_{\text{train}} > 0$:

- Draw from $\mathcal{P}(\cdot)$ until $\mathcal{L}_{\text{train}}(W_1, \ldots, W_L) < \epsilon_{\text{train}}$
- Equivalent to drawing from $\mathcal{P}\left(\cdot | \mathcal{L}_{\text{train}}(W_1, \ldots, W_L) < \epsilon_{\text{train}}\right)$

Volume hypothesis $\iff$ G&C generalizes well

# Gradient Descent for Matrix Factorization

Numerous studies proved GD for MF generalizes well in various settings (Gunasekar et al., 2017b; Arora et al., 2019; Chen et al., 2023; Chou et al., 2024)

# Gradient Descent for Matrix Factorization

Numerous studies proved GD for MF generalizes well in various settings (Gunasekar et al., 2017b; Arora et al., 2019; Chen et al., 2023; Chou et al., 2024)

**Intuition** *(Arora + **C** + Hu + Luo, NeurIPS 2019)*

# Gradient Descent for Matrix Factorization

Numerous studies proved GD for MF generalizes well in various settings (Gunasekar et al., 2017b; Arora et al., 2019; Chen et al., 2023; Chou et al., 2024)

**Intuition** *(Arora + **C** + Hu + Luo, NeurIPS 2019)*

Let $\sigma^{(r)}$ be the singular values of $W = W_L W_{L-1} \cdots W_1$

# Gradient Descent for Matrix Factorization

Numerous studies proved GD for MF generalizes well in various settings (Gunasekar et al., 2017b; Arora et al., 2019; Chen et al., 2023; Chou et al., 2024)

**Intuition** *(Arora + **C** + Hu + Luo, NeurIPS 2019)*

Let $\sigma^{(r)}$ be the singular values of $W = W_L W_{L-1} \cdots W_1$

**Theorem**

*GD with small init and learning rate leads to:* $\frac{d}{dt}\sigma^{(r)}(t) \propto \sigma^{(r)}(t)^{2-\frac{2}{L}}$

# Gradient Descent for Matrix Factorization

Numerous studies proved GD for MF generalizes well in various settings (Gunasekar et al., 2017b; Arora et al., 2019; Chen et al., 2023; Chou et al., 2024)

**Intuition** *(Arora + **C** + Hu + Luo, NeurIPS 2019)*

Let $\sigma^{(r)}$ be the singular values of $W = W_L W_{L-1} \cdots W_1$

**Theorem**

*GD with small init and learning rate leads to:* $\frac{d}{dt}\sigma^{(r)}(t) \propto \sigma^{(r)}(t)^{2-\frac{2}{L}}$

Singular values move slower when small and faster when large!

# Gradient Descent for Matrix Factorization

Numerous studies proved GD for MF generalizes well in various settings (Gunasekar et al., 2017b; Arora et al., 2019; Chen et al., 2023; Chou et al., 2024)

**Intuition** *(Arora + **C** + Hu + Luo, NeurIPS 2019)*

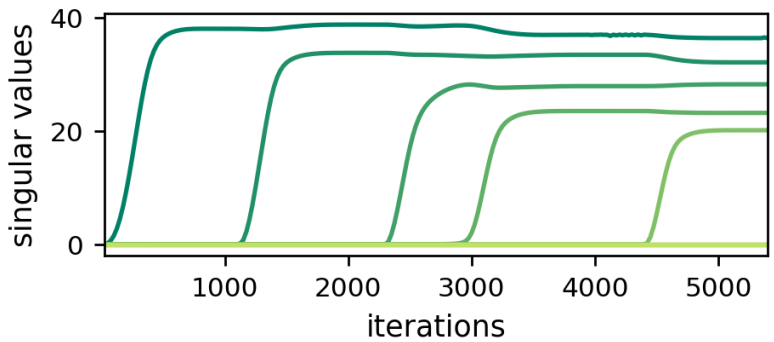
Let $\sigma^{(r)}$ be the singular values of $W = W_L W_{L-1} \cdots W_1$

**Theorem**

*GD with small init and learning rate leads to:* $\frac{d}{dt}\sigma^{(r)}(t) \propto \sigma^{(r)}(t)^{2-\frac{2}{L}}$

Singular values move slower when small and faster when large!

Experiment: completion of low rank matrix via GD over MF



# Gradient Descent for Matrix Factorization

Numerous studies proved GD for MF generalizes well in various settings (Gunasekar et al., 2017b; Arora et al., 2019; Chen et al., 2023; Chou et al., 2024)

**Intuition** *(Arora + **C** + Hu + Luo, NeurIPS 2019)*

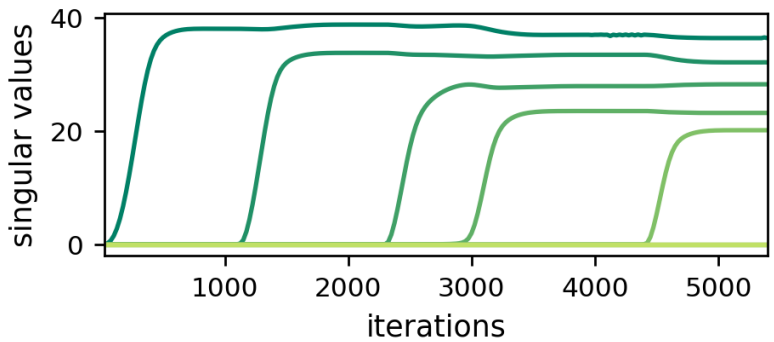
Let $\sigma^{(r)}$ be the singular values of $W = W_L W_{L-1} \cdots W_1$

**Theorem**

*GD with small init and learning rate leads to:* $\frac{d}{dt}\sigma^{(r)}(t) \propto \sigma^{(r)}(t)^{2-\frac{2}{L}}$

Singular values move slower when small and faster when large!

Experiment: completion of low rank matrix via GD over MF




**Incremental learning of singular values leads to low rank!**

# Gradient Descent for Matrix Factorization (cont')

## Practical Application

# Gradient Descent for Matrix Factorization (cont')

## Practical Application

Parameterize layers of NN as MF

# Gradient Descent for Matrix Factorization (cont')

**Practical Application**

Parameterize layers of NN as MF

$\Longrightarrow$ implicit rank minimization induces compressibility and generalization

# Gradient Descent for Matrix Factorization (cont')

## Practical Application

Parameterize layers of NN as MF

$\Longrightarrow$ implicit rank minimization induces compressibility and generalization

**Implicit Rank-Minimizing Autoencoder**

**Li Jing** Facebook AI Research New York
**Jure Zbontar** Facebook AI Research New York
**Yann LeCun** Facebook AI Research New York

**ExpandNets: Linear Over-parameterization to Train Compact Convolutional Networks**

**Shuxuan Guo** CVLab, EPFL
**Jose M. Alvarez** NVIDIA
**Mathieu Salzmann** CVLab, EPFL

THE LOW-RANK SIMPLICITY BIAS IN DEEP NETWORKS

**Minyoung Huh** MIT CSAIL
**Hossein Mobahi** Google Research
**Richard Zhang** Adobe Research
**Brian Cheung** MIT CSAIL & BCS
**Pulkit Agrawal** MIT CSAIL
**Phillip Isola** MIT CSAIL

**Understanding Generalization in Deep Learning via Tensor Methods**

**Jingling Li**[1,3] **Yanchao Sun**[1] **Jiahao Su**[4] **Taiji Suzuki**[2,3] **Furong Huang**[1]
[1]Department of Computer Science, University of Maryland, College Park
[2]Graduate School of Information Science and Technology, The University of Tokyo
[3]Center for Advanced Intelligence Project, RIKEN
[4]Department of Electrical and Computer Engineering, University of Maryland, College Park

# Outline

# Wide Matrix Factorization: Guess and Check

# Wide Matrix Factorization: Guess and Check

## Theorem

Let $\epsilon_{\text{train}}, \epsilon_{\text{gen}} > 0$.

# Wide Matrix Factorization: Guess and Check

**Theorem**

Let $\epsilon_{\text{train}}, \epsilon_{\text{gen}} > 0$. Assume activation $\sigma$ is antisymmetric (e.g. $\tanh(\cdot)$).

# Wide Matrix Factorization: Guess and Check

**Theorem**

Let $\epsilon_{\text{train}}, \epsilon_{\text{gen}} > 0$. Assume activation $\sigma$ is antisymmetric (e.g. $\tanh(\cdot)$).

Then, under mild conditions, if hidden width $k$ is sufficiently large:

# Wide Matrix Factorization: Guess and Check

## Theorem

Let $\epsilon_{\text{train}}, \epsilon_{\text{gen}} > 0$. Assume activation $\sigma$ is antisymmetric (e.g. $\tanh(\cdot)$).

Then, under mild conditions, if hidden width $k$ is sufficiently large:

$$\underbrace{\mathcal{P}\big(\mathcal{L}_{\text{gen}}(W) < \epsilon_{\text{gen}} \,\big|\, \mathcal{L}_{\text{train}}(W) < \epsilon_{\text{train}}\big)}_{\text{Generalize well conditioned on low training loss}} - \underbrace{\mathcal{P}\big(\mathcal{L}_{\text{gen}}(W) < \epsilon_{\text{gen}}\big)}_{\text{Generalize well unconditionally}} \lesssim k^{-1/2}$$

# Wide Matrix Factorization: Guess and Check

**Theorem**

Let $\epsilon_{\text{train}}, \epsilon_{\text{gen}} > 0$. Assume activation $\sigma$ is antisymmetric (e.g. $\tanh(\cdot)$).

Then, under mild conditions, if hidden width $k$ is sufficiently large:

$$\underbrace{\mathcal{P}\big(\mathcal{L}_{\text{gen}}(W) < \epsilon_{\text{gen}} \,\big|\, \mathcal{L}_{\text{train}}(W) < \epsilon_{\text{train}}\big)}_{\text{Generalize well conditioned on low training loss}} - \underbrace{\mathcal{P}\big(\mathcal{L}_{\text{gen}}(W) < \epsilon_{\text{gen}}\big)}_{\text{Generalize well unconditionally}} \lesssim k^{-1/2}$$

G&C is no better than ignoring measurements and drawing from prior!

# Wide Matrix Factorization: Guess and Check (cont')

## Theorem

Let $\epsilon_{\text{train}}, \epsilon_{\text{gen}} > 0$. Assume activation $\sigma$ is antisymmetric (e.g. $\tanh(\cdot)$).

Then, under mild conditions, if hidden width $k$ is sufficiently large:

$$\mathcal{P}\big(\mathcal{L}_{\text{gen}}(W) < \epsilon_{\text{gen}} \,\big|\, \mathcal{L}_{\text{train}}(W) < \epsilon_{\text{train}}\big) \;-\; \mathcal{P}\big(\mathcal{L}_{\text{gen}}(W) < \epsilon_{\text{gen}}\big) \;\lesssim\; k^{-1/2}$$

# Wide Matrix Factorization: Guess and Check (cont')

## Theorem

Let $\epsilon_{\text{train}}, \epsilon_{\text{gen}} > 0$. Assume activation $\sigma$ is antisymmetric (e.g. $\tanh(\cdot)$). Then, under mild conditions, if hidden width $k$ is sufficiently large:

$$\mathcal{P}\big(\mathcal{L}_{\text{gen}}(W) < \epsilon_{\text{gen}} \,\big|\, \mathcal{L}_{\text{train}}(W) < \epsilon_{\text{train}}\big) \;-\; \mathcal{P}\big(\mathcal{L}_{\text{gen}}(W) < \epsilon_{\text{gen}}\big) \;\lesssim\; k^{-1/2}$$

**Proof Sketch**

# Wide Matrix Factorization: Guess and Check (cont')

**Theorem**

Let $\epsilon_{\text{train}}, \epsilon_{\text{gen}} > 0$. Assume activation $\sigma$ is antisymmetric (e.g. $\tanh(\cdot)$). Then, under mild conditions, if hidden width $k$ is sufficiently large:

$$\mathcal{P}\big(\mathcal{L}_{\text{gen}}(W) < \epsilon_{\text{gen}} \,\big|\, \mathcal{L}_{\text{train}}(W) < \epsilon_{\text{train}}\big) \;-\; \mathcal{P}\big(\mathcal{L}_{\text{gen}}(W) < \epsilon_{\text{gen}}\big) \;\lesssim\; k^{-1/2}$$

**Proof Sketch**

- Cols of $W$ = outputs of NN for standard basis vector inputs

# Wide Matrix Factorization: Guess and Check (cont')

**Theorem**

Let $\epsilon_{\text{train}}, \epsilon_{\text{gen}} > 0$. Assume activation $\sigma$ is antisymmetric (e.g. $\tanh(\cdot)$).

Then, under mild conditions, if hidden width $k$ is sufficiently large:

$$\mathcal{P}\big(\mathcal{L}_{\text{gen}}(W) < \epsilon_{\text{gen}} \,\big|\, \mathcal{L}_{\text{train}}(W) < \epsilon_{\text{train}}\big) \;-\; \mathcal{P}\big(\mathcal{L}_{\text{gen}}(W) < \epsilon_{\text{gen}}\big) \;\lesssim\; k^{-1/2}$$

**Proof Sketch**

- Cols of $W$ = outputs of NN for standard basis vector inputs
- By Gaussian process results for NNs: as $k \to \infty$, $W$ converges to $W_{\text{iid}}$, whose entries are i.i.d. $\mathcal{N}(0, \nu)$

# Wide Matrix Factorization: Guess and Check (cont')

**Theorem**

Let $\epsilon_{\text{train}}, \epsilon_{\text{gen}} > 0$. Assume activation $\sigma$ is antisymmetric (e.g. $\tanh(\cdot)$).

Then, under mild conditions, if hidden width $k$ is sufficiently large:

$$\mathcal{P}\big(\mathcal{L}_{\text{gen}}(W) < \epsilon_{\text{gen}} \,\big|\, \mathcal{L}_{\text{train}}(W) < \epsilon_{\text{train}}\big) \;-\; \mathcal{P}\big(\mathcal{L}_{\text{gen}}(W) < \epsilon_{\text{gen}}\big) \;\lesssim\; k^{-1/2}$$

**Proof Sketch**

- Cols of $W$ = outputs of NN for standard basis vector inputs
- By Gaussian process results for NNs: as $k \to \infty$, $W$ converges to $W_{\text{iid}}$, whose entries are i.i.d. $\mathcal{N}(0, \nu)$
- Measurement matrices $A_i$ orthogonal to basis defining $\mathcal{L}_{\text{gen}}(\cdot)$

# Wide Matrix Factorization: Guess and Check (cont')

**Theorem**

Let $\epsilon_{\text{train}}, \epsilon_{\text{gen}} > 0$. Assume activation $\sigma$ is antisymmetric (e.g. $\tanh(\cdot)$).

Then, under mild conditions, if hidden width $k$ is sufficiently large:

$$\mathcal{P}\big(\mathcal{L}_{\text{gen}}(W) < \epsilon_{\text{gen}} \,\big|\, \mathcal{L}_{\text{train}}(W) < \epsilon_{\text{train}}\big) \;-\; \mathcal{P}\big(\mathcal{L}_{\text{gen}}(W) < \epsilon_{\text{gen}}\big) \;\lesssim\; k^{-1/2}$$

**Proof Sketch**

- Cols of $W$ = outputs of NN for standard basis vector inputs
- By Gaussian process results for NNs: as $k \to \infty$, $W$ converges to $W_{\text{iid}}$, whose entries are i.i.d. $\mathcal{N}(0, \nu)$
- Measurement matrices $A_i$ orthogonal to basis defining $\mathcal{L}_{\text{gen}}(\cdot) \Rightarrow$ events $\{\mathcal{L}_{\text{train}}(W_{\text{iid}}) < \epsilon_{\text{train}}\}$ and $\{\mathcal{L}_{\text{gen}}(W_{\text{iid}}) < \epsilon_{\text{gen}}\}$ independent

# Wide Matrix Factorization: Guess and Check (cont')

**Theorem**

Let $\epsilon_{\text{train}}, \epsilon_{\text{gen}} > 0$. Assume activation $\sigma$ is antisymmetric (e.g. $\tanh(\cdot)$).

Then, under mild conditions, if hidden width $k$ is sufficiently large:

$$\mathcal{P}\big(\mathcal{L}_{\text{gen}}(W) < \epsilon_{\text{gen}} \,\big|\, \mathcal{L}_{\text{train}}(W) < \epsilon_{\text{train}}\big) \;-\; \mathcal{P}\big(\mathcal{L}_{\text{gen}}(W) < \epsilon_{\text{gen}}\big) \;\lesssim\; k^{-1/2}$$

**Proof Sketch**

- Cols of $W$ = outputs of NN for standard basis vector inputs
- By Gaussian process results for NNs: as $k \to \infty$, $W$ converges to $W_{\text{iid}}$, whose entries are i.i.d. $\mathcal{N}(0, \nu)$
- Measurement matrices $A_i$ orthogonal to basis defining $\mathcal{L}_{\text{gen}}(\cdot) \Rightarrow$ events $\{\mathcal{L}_{\text{train}}(W_{\text{iid}}) < \epsilon_{\text{train}}\}$ and $\{\mathcal{L}_{\text{gen}}(W_{\text{iid}}) < \epsilon_{\text{gen}}\}$ independent $\Rightarrow$ conditioning on former event does not affect probability of latter

# Wide Matrix Factorization: Guess and Check (cont')

## Theorem

Let $\epsilon_{\text{train}}, \epsilon_{\text{gen}} > 0$. Assume activation $\sigma$ is antisymmetric (e.g. $\tanh(\cdot)$).

Then, under mild conditions, if hidden width $k$ is sufficiently large:

$$\mathcal{P}\big(\mathcal{L}_{\text{gen}}(W) < \epsilon_{\text{gen}} \,\big|\, \mathcal{L}_{\text{train}}(W) < \epsilon_{\text{train}}\big) \;-\; \mathcal{P}\big(\mathcal{L}_{\text{gen}}(W) < \epsilon_{\text{gen}}\big) \;\lesssim\; k^{-1/2}$$

**Proof Sketch**

- Cols of $W$ = outputs of NN for standard basis vector inputs
- By Gaussian process results for NNs: as $k \to \infty$, $W$ converges to $W_{\text{iid}}$, whose entries are i.i.d. $\mathcal{N}(0, \nu)$
- Measurement matrices $A_i$ orthogonal to basis defining $\mathcal{L}_{\text{gen}}(\cdot) \Rightarrow$ events $\{\mathcal{L}_{\text{train}}(W_{\text{iid}}) < \epsilon_{\text{train}}\}$ and $\{\mathcal{L}_{\text{gen}}(W_{\text{iid}}) < \epsilon_{\text{gen}}\}$ independent $\Rightarrow$ conditioning on former event does not affect probability of latter
- Since $W$ is $k^{-1/2}$-close to $W_{\text{iid}}$:

$$\Big|\mathcal{P}(\mathcal{L}_{\text{gen}} < \epsilon_{\text{gen}} \mid \mathcal{L}_{\text{train}} < \epsilon_{\text{train}}) - \mathcal{P}(\mathcal{L}_{\text{gen}} < \epsilon_{\text{gen}})\Big| \;\lesssim\; k^{-1/2}$$

# Wide Matrix Factorization: Gradient Descent

# Wide Matrix Factorization: Gradient Descent

**Theorem** (Soltanolkotabi et al., 2023)

Under mild conditions, if width $k >$ dims of $W$ then GD generalizes well

# Wide Matrix Factorization: Gradient Descent

**Theorem** (Soltanolkotabi et al., 2023)

Under mild conditions, if width $k >$ dims of $W$ then GD generalizes well

**Corollary**

First proof of standard case where GD generalizes well while G&C does not

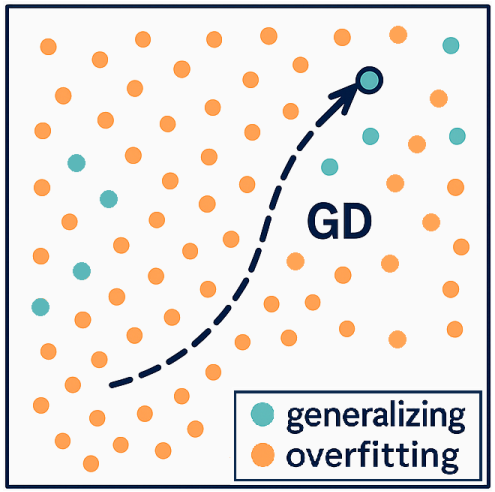
# Wide Matrix Factorization: Gradient Descent

**Theorem** (Soltanolkotabi et al., 2023)

Under mild conditions, if width $k >$ dims of $W$ then GD generalizes well

**Corollary**

First proof of standard case where GD generalizes well while G&C does not




✓ Implicit bias


✗ Volume hypothesis

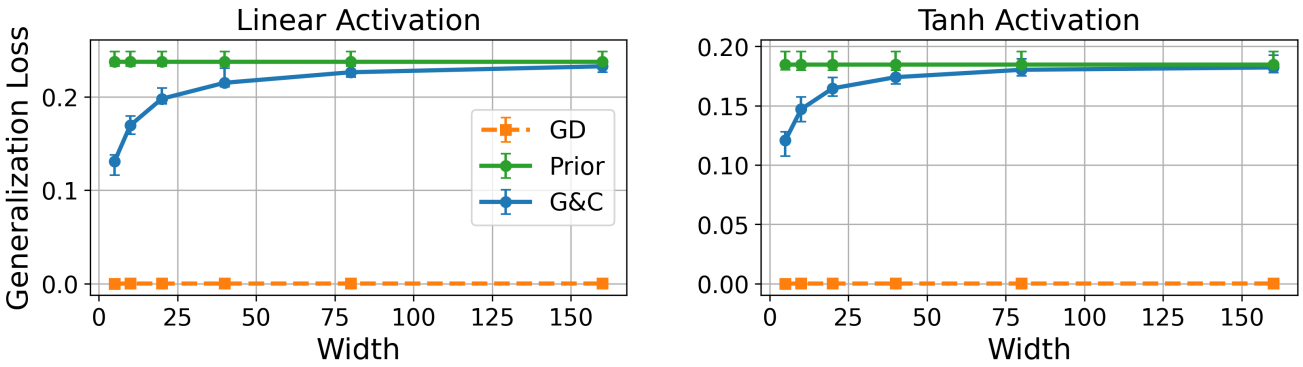
# Wide Matrix Factorization: Empirical Demonstrations

# Wide Matrix Factorization: Empirical Demonstrations

Experiments validate theory:

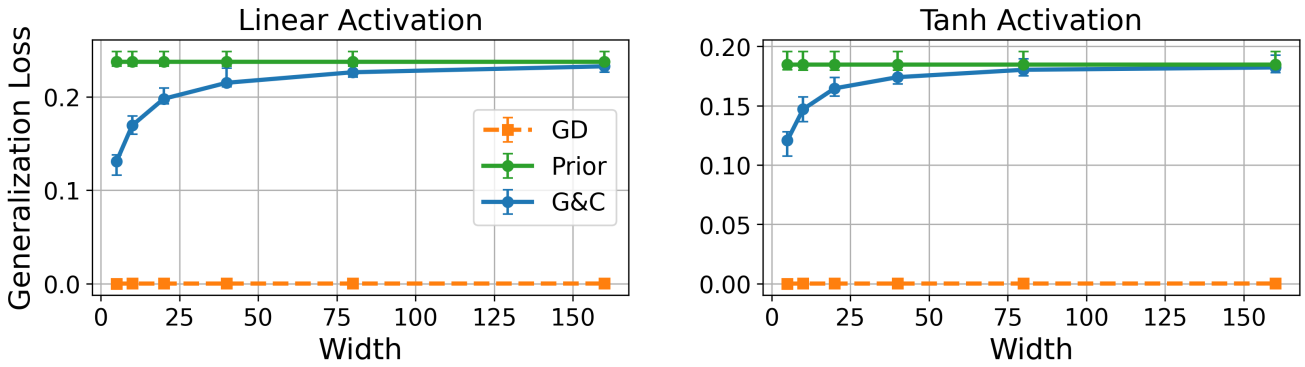
# Wide Matrix Factorization: Empirical Demonstrations

Experiments validate theory:



# Wide Matrix Factorization: Empirical Demonstrations

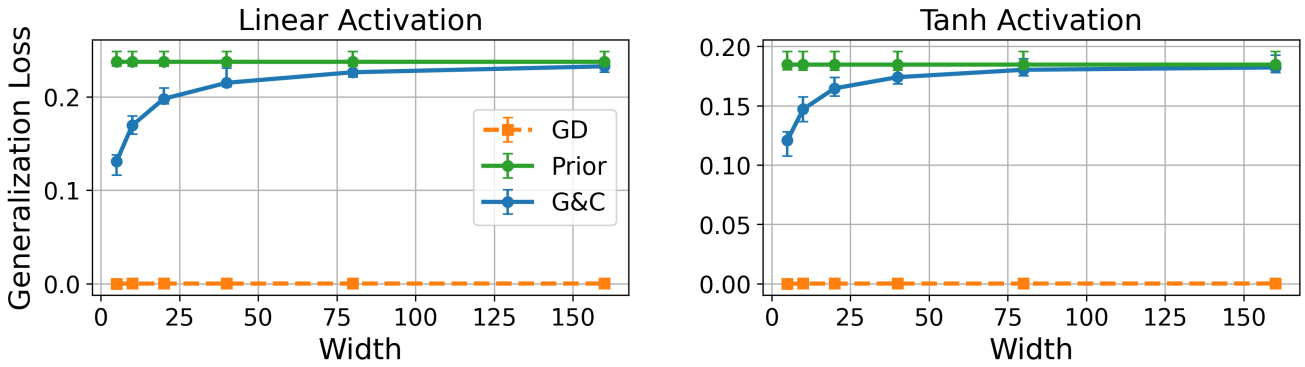
Experiments validate theory:




GD generalizes well throughout; G&C deteriorates as width grows

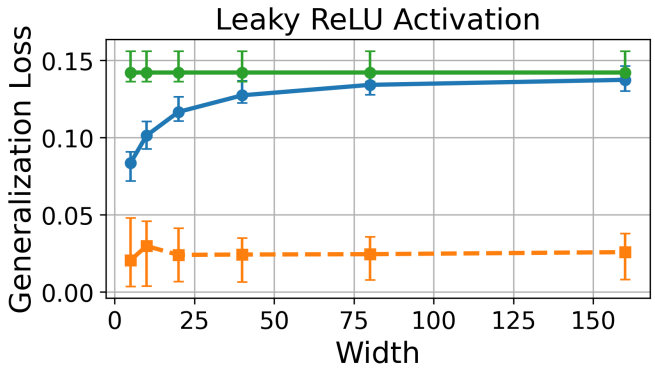
# Wide Matrix Factorization: Empirical Demonstrations

Experiments validate theory:




GD generalizes well throughout; G&C deteriorates as width grows

Phenomenon extends to non-symmetric activations:



# Outline

# Deep Matrix Factorization: Guess and Check

# Deep Matrix Factorization: Guess and Check

## Theorem

Let $\epsilon_{\text{train}}, \epsilon_{\text{gen}} > 0$.

# Deep Matrix Factorization: Guess and Check

## Theorem

Let $\epsilon_{\text{train}}, \epsilon_{\text{gen}} > 0$. Assume ground truth $W^*$ has rank 1 and activation $\sigma$ is linear ($\sigma(\alpha) = \alpha$).

# Deep Matrix Factorization: Guess and Check

## Theorem

Let $\epsilon_{\text{train}}, \epsilon_{\text{gen}} > 0$. Assume ground truth $W^*$ has rank 1 and activation $\sigma$ is linear ($\sigma(\alpha) = \alpha$). Then, under mild conditions, $\exists c > 0$ such that for any depth $L$:

$$\underbrace{\mathcal{P}\big(\mathcal{L}_{\text{gen}}(W) < c\epsilon_{\text{train}} \,\big|\, \mathcal{L}_{\text{train}}(W) < \epsilon_{\text{train}}\big)}_{\text{Generalize well conditioned on low training loss}} \geq 1 - O\big(\tfrac{1}{L}\big).$$

# Deep Matrix Factorization: Guess and Check

## Theorem

Let $\epsilon_{\text{train}}, \epsilon_{\text{gen}} > 0$. Assume ground truth $W^*$ has rank 1 and activation $\sigma$ is linear ($\sigma(\alpha) = \alpha$). Then, under mild conditions, $\exists c > 0$ such that for any depth $L$:

$$\underbrace{\mathcal{P}\big(\mathcal{L}_{\text{gen}}(W) < c\epsilon_{\text{train}} \,\big|\, \mathcal{L}_{\text{train}}(W) < \epsilon_{\text{train}}\big)}_{\text{Generalize well conditioned on low training loss}} \geq 1 - O\big(\tfrac{1}{L}\big).$$

G&C generalizes well; GD not needed for good generalization!

# Deep Matrix Factorization: Guess and Check

## Theorem

Let $\epsilon_{\text{train}}, \epsilon_{\text{gen}} > 0$. Assume ground truth $W^*$ has rank 1 and activation $\sigma$ is linear ($\sigma(\alpha) = \alpha$). Then, under mild conditions, $\exists c > 0$ such that for any depth $L$:

$$\underbrace{\mathcal{P}\big(\mathcal{L}_{\text{gen}}(W) < c\epsilon_{\text{train}} \,\big|\, \mathcal{L}_{\text{train}}(W) < \epsilon_{\text{train}}\big)}_{\text{Generalize well conditioned on low training loss}} \geq 1 - O\big(\tfrac{1}{L}\big).$$

G&C generalizes well; GD not needed for good generalization!


✓ Volume hypothesis

# Deep Matrix Factorization: Guess and Check (cont')

## Theorem

Let $\epsilon_{\text{train}}, \epsilon_{\text{gen}} > 0$. Assume ground truth $W^*$ has rank 1 and activation $\sigma$ is linear ($\sigma(\alpha) = \alpha$). Then, under mild conditions, $\exists c > 0$ such that for any depth $L$:

$$\mathcal{P}\big(\mathcal{L}_{\text{gen}}(W) < c\epsilon_{\text{train}} \,\big|\, \mathcal{L}_{\text{train}}(W) < \epsilon_{\text{train}}\big) \;\geq\; 1 - O\big(\tfrac{1}{L}\big).$$

# Deep Matrix Factorization: Guess and Check (cont')

**Theorem**

Let $\epsilon_{\text{train}}, \epsilon_{\text{gen}} > 0$. Assume ground truth $W^*$ has rank 1 and activation $\sigma$ is linear ($\sigma(\alpha) = \alpha$). Then, under mild conditions, $\exists c > 0$ such that for any depth $L$:

$$\mathcal{P}\big(\mathcal{L}_{\text{gen}}(W) < c\epsilon_{\text{train}} \,\big|\, \mathcal{L}_{\text{train}}(W) < \epsilon_{\text{train}}\big) \;\geq\; 1 - O\big(\tfrac{1}{L}\big).$$

**Proof Sketch**

# Deep Matrix Factorization: Guess and Check (cont')

**Theorem**

Let $\epsilon_{\text{train}}, \epsilon_{\text{gen}} > 0$. Assume ground truth $W^*$ has rank 1 and activation $\sigma$ is linear ($\sigma(\alpha) = \alpha$). Then, under mild conditions, $\exists c > 0$ such that for any depth $L$:

$$\mathcal{P}\big(\mathcal{L}_{\text{gen}}(W) < c\epsilon_{\text{train}} \,\big|\, \mathcal{L}_{\text{train}}(W) < \epsilon_{\text{train}}\big) \;\geq\; 1 - O\big(\tfrac{1}{L}\big).$$

**Proof Sketch**

- Write $W = W_L\, W_{L-1:2}\, W_1$ with $W_{L-1:2} := W_{L-1} \cdots W_2$

# Deep Matrix Factorization: Guess and Check (cont')

**Theorem**

Let $\epsilon_{\text{train}}, \epsilon_{\text{gen}} > 0$. Assume ground truth $W^*$ has rank 1 and activation $\sigma$ is linear ($\sigma(\alpha) = \alpha$). Then, under mild conditions, $\exists c > 0$ such that for any depth $L$:

$$\mathcal{P}\big(\mathcal{L}_{\text{gen}}(W) < c\epsilon_{\text{train}} \,\big|\, \mathcal{L}_{\text{train}}(W) < \epsilon_{\text{train}}\big) \;\geq\; 1 - O\big(\tfrac{1}{L}\big).$$

**Proof Sketch**

- Write $W = W_L\, W_{L-1:2}\, W_1$ with $W_{L-1:2} := W_{L-1} \cdots W_2$
- By bounds on spectrum of Gaussian products: $W_{L-1:2}$ is close to a rank 1 matrix w.p. $1 - e^{-\Omega(L)}$

# Deep Matrix Factorization: Guess and Check (cont')

**Theorem**

Let $\epsilon_{\text{train}}, \epsilon_{\text{gen}} > 0$. Assume ground truth $W^*$ has rank 1 and activation $\sigma$ is linear ($\sigma(\alpha) = \alpha$). Then, under mild conditions, $\exists c > 0$ such that for any depth $L$:

$$\mathcal{P}\big(\mathcal{L}_{\text{gen}}(W) < c\epsilon_{\text{train}} \,\big|\, \mathcal{L}_{\text{train}}(W) < \epsilon_{\text{train}}\big) \;\geq\; 1 - O\big(\tfrac{1}{L}\big).$$

**Proof Sketch**

- Write $W = W_L\, W_{L-1:2}\, W_1$ with $W_{L-1:2} := W_{L-1} \cdots W_2$
- By bounds on spectrum of Gaussian products: $W_{L-1:2}$ is close to a rank 1 matrix w.p. $1 - e^{-\Omega(L)}$ $\Rightarrow$ same holds for $W$

# Deep Matrix Factorization: Guess and Check (cont')

**Theorem**

Let $\epsilon_{\text{train}}, \epsilon_{\text{gen}} > 0$. Assume ground truth $W^*$ has rank 1 and activation $\sigma$ is linear ($\sigma(\alpha) = \alpha$). Then, under mild conditions, $\exists c > 0$ such that for any depth $L$:

$$\mathcal{P}\big(\mathcal{L}_{\text{gen}}(W) < c\epsilon_{\text{train}} \,\big|\, \mathcal{L}_{\text{train}}(W) < \epsilon_{\text{train}}\big) \;\geq\; 1 - O\big(\tfrac{1}{L}\big).$$

**Proof Sketch**

- Write $W = W_L\, W_{L-1:2}\, W_1$ with $W_{L-1:2} := W_{L-1} \cdots W_2$
- By bounds on spectrum of Gaussian products: $W_{L-1:2}$ is close to a rank 1 matrix w.p. $1 - e^{-\Omega(L)} \Rightarrow$ same holds for $W$
- Under mild assumption on measurement matrices $A_i$, the event $\{\mathcal{L}_{\text{train}}(W) < \epsilon_{\text{train}} \wedge\ \mathcal{L}_{\text{gen}}(W) \geq \epsilon_{\text{train}} c\}$ holds w.p. $O(1/L)$

# Deep Matrix Factorization: Guess and Check (cont')

**Theorem**

Let $\epsilon_{\text{train}}, \epsilon_{\text{gen}} > 0$. Assume ground truth $W^*$ has rank 1 and activation $\sigma$ is linear ($\sigma(\alpha) = \alpha$). Then, under mild conditions, $\exists c > 0$ such that for any depth $L$:

$$\mathcal{P}\big(\mathcal{L}_{\text{gen}}(W) < c\epsilon_{\text{train}} \,\big|\, \mathcal{L}_{\text{train}}(W) < \epsilon_{\text{train}}\big) \;\geq\; 1 - O\big(\tfrac{1}{L}\big).$$

**Proof Sketch**

- Write $W = W_L\, W_{L-1:2}\, W_1$ with $W_{L-1:2} := W_{L-1} \cdots W_2$
- By bounds on spectrum of Gaussian products: $W_{L-1:2}$ is close to a rank 1 matrix w.p. $1 - e^{-\Omega(L)} \Rightarrow$ same holds for $W$
- Under mild assumption on measurement matrices $A_i$, the event $\{\mathcal{L}_{\text{train}}(W) < \epsilon_{\text{train}} \wedge\ \mathcal{L}_{\text{gen}}(W) \geq \epsilon_{\text{train}} c\}$ holds w.p. $O(1/L)$
- Since $\mathcal{P}\big(\mathcal{L}_{\text{train}}(W) < \epsilon_{\text{train}}\big) = \Omega(1)$:

$$\mathcal{P}\big(\mathcal{L}_{\text{gen}} < \epsilon_{\text{train}} c \mid \mathcal{L}_{\text{train}} < \epsilon_{\text{train}}\big) \geq 1 - O\big(\tfrac{1}{L}\big)$$

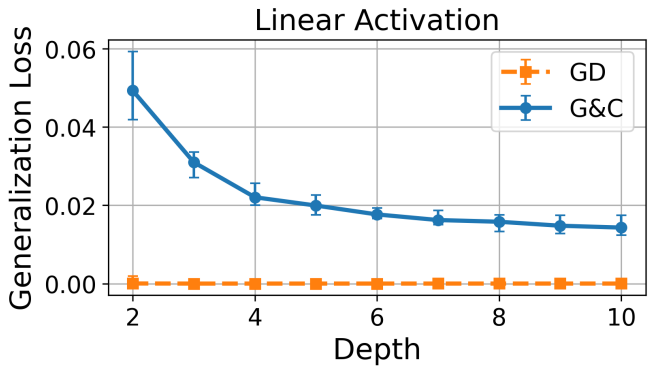
# Deep Matrix Factorization: Empirical Demonstrations

# Deep Matrix Factorization: Empirical Demonstrations

Experiments validate theory:

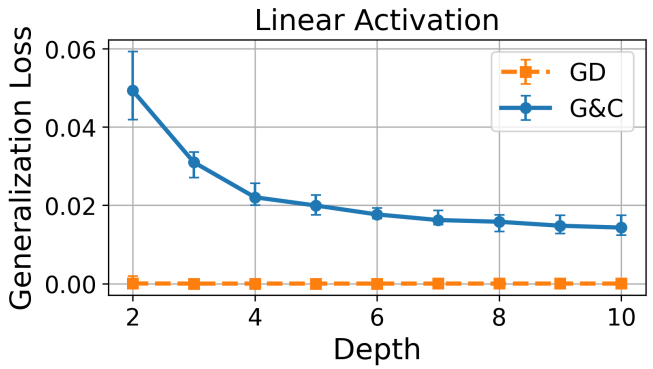
# Deep Matrix Factorization: Empirical Demonstrations

Experiments validate theory:



# Deep Matrix Factorization: Empirical Demonstrations

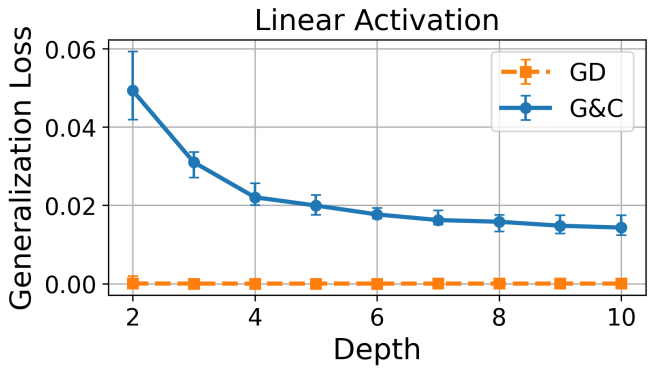
Experiments validate theory:




GD generalizes well throughout; G&C improves as depth grows

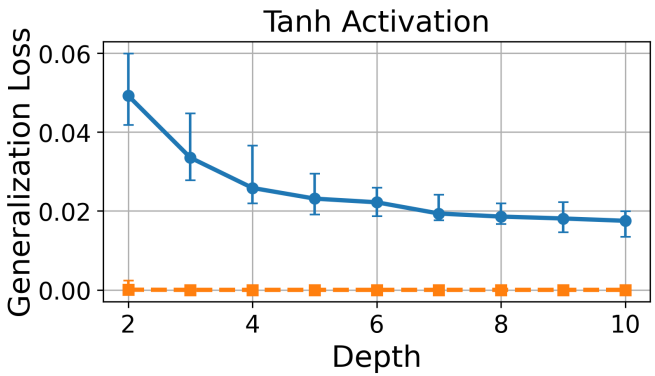
# Deep Matrix Factorization: Empirical Demonstrations

Experiments validate theory:




GD generalizes well throughout; G&C improves as depth grows

Phenomenon extends to nonlinear activations:



# Outline

# Recap

# Recap

Understanding deep learning requires addressing three Qs:

Expressiveness  Optimization  Generalization

# Recap

Understanding deep learning requires addressing three Qs:

Expressiveness  Optimization  Generalization

Two alternative hypotheses concerning generalization:

# Recap

Understanding deep learning requires addressing three Qs:

Expressiveness  Optimization  Generalization

Two alternative hypotheses concerning generalization:

- Implicit bias:

# Recap

Understanding deep learning requires addressing three Qs:

Expressiveness  Optimization  Generalization

Two alternative hypotheses concerning generalization:

- Implicit bias: among predictors fitting training set, those that generalize well are special, yet GD finds one

# Recap

Understanding deep learning requires addressing three Qs:

Expressiveness  Optimization  Generalization

Two alternative hypotheses concerning generalization:

- Implicit bias: among predictors fitting training set, those that generalize well are special, yet GD finds one
- Volume hypothesis:

# Recap

Understanding deep learning requires addressing three Qs:

Expressiveness  Optimization  Generalization

Two alternative hypotheses concerning generalization:

- Implicit bias: among predictors fitting training set, those that generalize well are special, yet GD finds one
- Volume hypothesis: most predictors fitting training set generalize well, and GD merely finds typical one

# Recap

Understanding deep learning requires addressing three Qs:

Expressiveness  Optimization  Generalization

Two alternative hypotheses concerning generalization:

- Implicit bias: among predictors fitting training set, those that generalize well are special, yet GD finds one
- Volume hypothesis: most predictors fitting training set generalize well, and GD merely finds typical one

**Q**: is GD needed for good generalization?

# Recap

Understanding deep learning requires addressing three Qs:

Expressiveness  Optimization  Generalization

Two alternative hypotheses concerning generalization:

- Implicit bias: among predictors fitting training set, those that generalize well are special, yet GD finds one
- Volume hypothesis: most predictors fitting training set generalize well, and GD merely finds typical one

**Q**: is GD needed for good generalization?

We analyze matrix factorization, a common testbed in deep learning theory

# Recap

Understanding deep learning requires addressing three Qs:

Expressiveness  Optimization  Generalization

Two alternative hypotheses concerning generalization:

- Implicit bias: among predictors fitting training set, those that generalize well are special, yet GD finds one
- Volume hypothesis: most predictors fitting training set generalize well, and GD merely finds typical one

**Q**: is GD needed for good generalization?

We analyze matrix factorization, a common testbed in deep learning theory

Even in this simple setting, there is no simple answer:

# Recap

Understanding deep learning requires addressing three Qs:

Expressiveness  Optimization  Generalization

Two alternative hypotheses concerning generalization:

- Implicit bias: among predictors fitting training set, those that generalize well are special, yet GD finds one
- Volume hypothesis: most predictors fitting training set generalize well, and GD merely finds typical one

**Q**: is GD needed for good generalization?

We analyze matrix factorization, a common testbed in deep learning theory

Even in this simple setting, there is no simple answer:

GD is needed for wide models, not needed for deep models

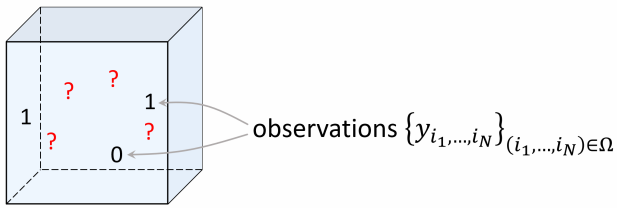
# Future Work

Recently we proved that properties of GD ensure good generalization in settings more realistic than matrix factorization:

# Future Work

Recently we proved that properties of GD ensure good generalization in settings more realistic than matrix factorization:

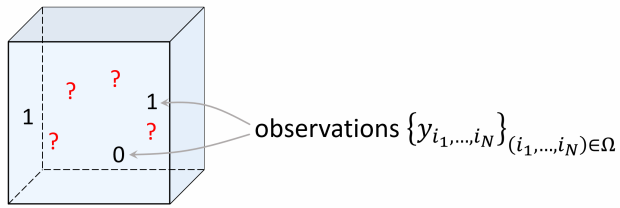
- Tensor factorizations (Razin et al., 2021, 2022)

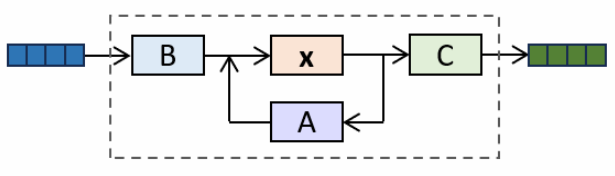


# Future Work

Recently we proved that properties of GD ensure good generalization in settings more realistic than matrix factorization:

- Tensor factorizations (Razin et al., 2021, 2022)
- Sequence-to-sequence models (Cohen-Karlik et al., 2022a,b)

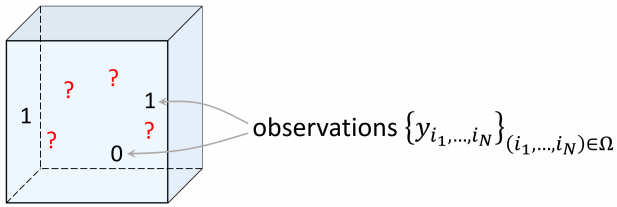


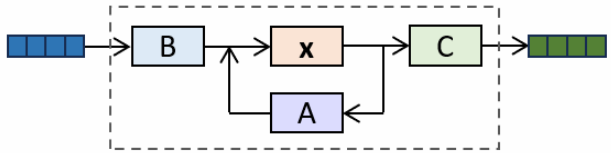



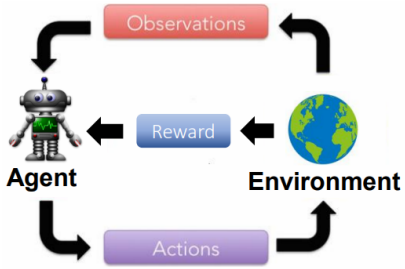
# Future Work

Recently we proved that properties of GD ensure good generalization in settings more realistic than matrix factorization:

- Tensor factorizations (Razin et al., 2021, 2022)
- Sequence-to-sequence models (Cohen-Karlik et al., 2022a,b)
- Reinforcement learning (Razin et al., 2024)



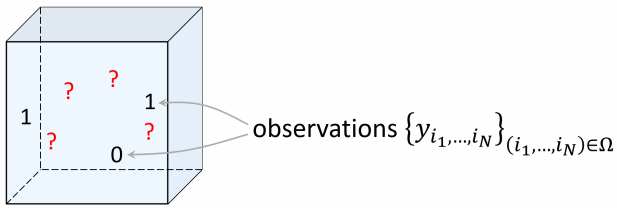



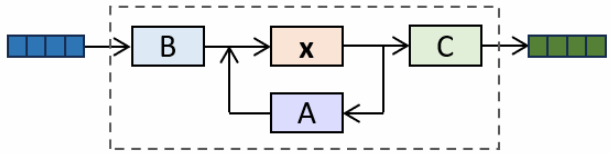



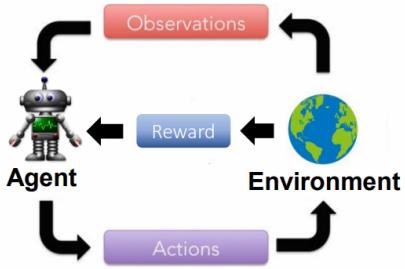
# Future Work

Recently we proved that properties of GD ensure good generalization in settings more realistic than matrix factorization:

- Tensor factorizations (Razin et al., 2021, 2022)
- Sequence-to-sequence models (Cohen-Karlik et al., 2022a,b)
- Reinforcement learning (Razin et al., 2024)



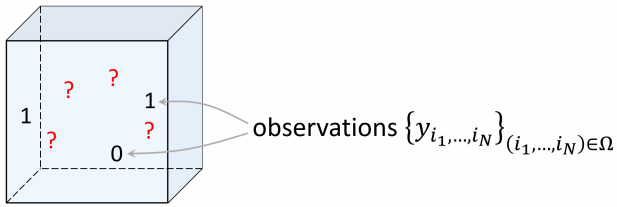

**Future Work**

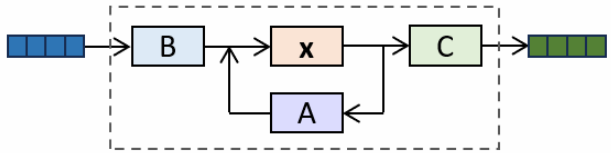
Study volume hypothesis (i.e. whether GD is needed) in these settings

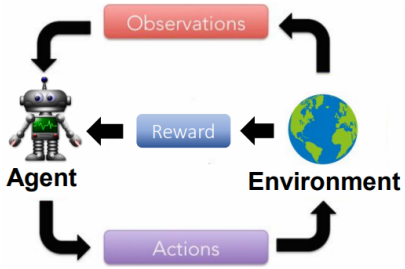
# Future Work

Recently we proved that properties of GD ensure good generalization in settings more realistic than matrix factorization:

- Tensor factorizations (Razin et al., 2021, 2022)
- Sequence-to-sequence models (Cohen-Karlik et al., 2022a,b)
- Reinforcement learning (Razin et al., 2024)




**Future Work**

Study volume hypothesis (i.e. whether GD is needed) in these settings

May indicate when accelerations of GD impede generalization

# Thank You!

**Work supported by**:
ERC Starting Grant (101164614); Apple; Google; Meta; Yandex; ISF Grant (1780/21); Blavatnik Foundation; Adelis Foundation; Tel Aviv University Center for AI and Data Science; and Amnon and Anat Shashua