Global Optimality Conditions for Deep Linear Neural Networks

Chulhee (Charlie) Yun

joint work with Prof. Suvrit Sra and Prof. Ali Jadbabaie

chulheey@mit.edu

LIDS Student Conference, Feb 1, 2018

Background

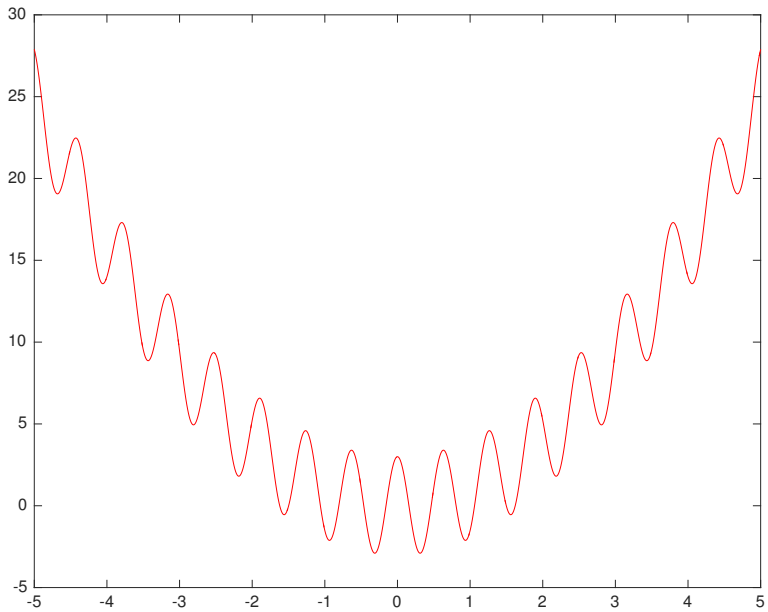
- Training a deep neural networks is minimizing the empirical risk of the network.

Background

- Training a deep neural networks is minimizing the empirical risk of the network.
- For a typical NN, the empirical risk is a nonconvex function!

Background

- Training a deep neural networks is minimizing the empirical risk of the network.
- For a typical NN, the empirical risk is a nonconvex function!
- Nonconvex optimization could end up at a bad (or spurious) local minimum.



Background

- Many success stories of NN $\Rightarrow$ despite nonconvexity, nice properties might exist!

Background

- Many success stories of NN $\Rightarrow$ despite nonconvexity, nice properties might exist!
- Empirical evidence on loss surface: putting more hidden nodes makes local minima close to global minima.

Background

- Many success stories of NN $\Rightarrow$ despite nonconvexity, nice properties might exist!
- Empirical evidence on loss surface: putting more hidden nodes makes local minima close to global minima.
- Theory side: no complete understanding yet.

Background

- Many success stories of NN $\Rightarrow$ despite nonconvexity, nice properties might exist!
- Empirical evidence on loss surface: putting more hidden nodes makes local minima close to global minima.
- Theory side: no complete understanding yet.
- Simplified model: *linear* neural networks $\Rightarrow$ nonconvex, but has nice properties!

Problem Setting: Deep Neural Networks

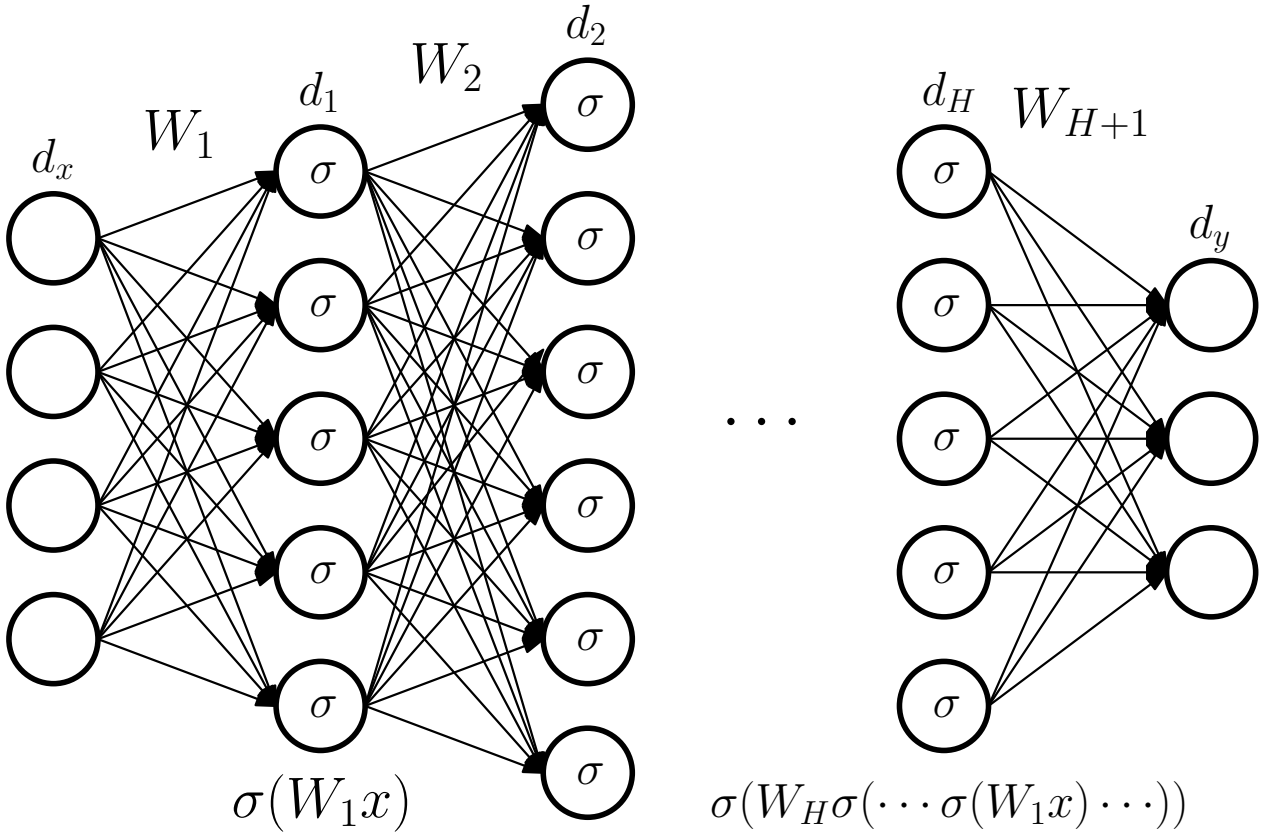
- m data points, input dimension d_x, output dimension d_y.
- Data matrix $X \in \mathbb{R}^{d_x \times m}$, label matrix $Y \in \mathbb{R}^{d_y \times m}$

Problem Setting: Deep Neural Networks

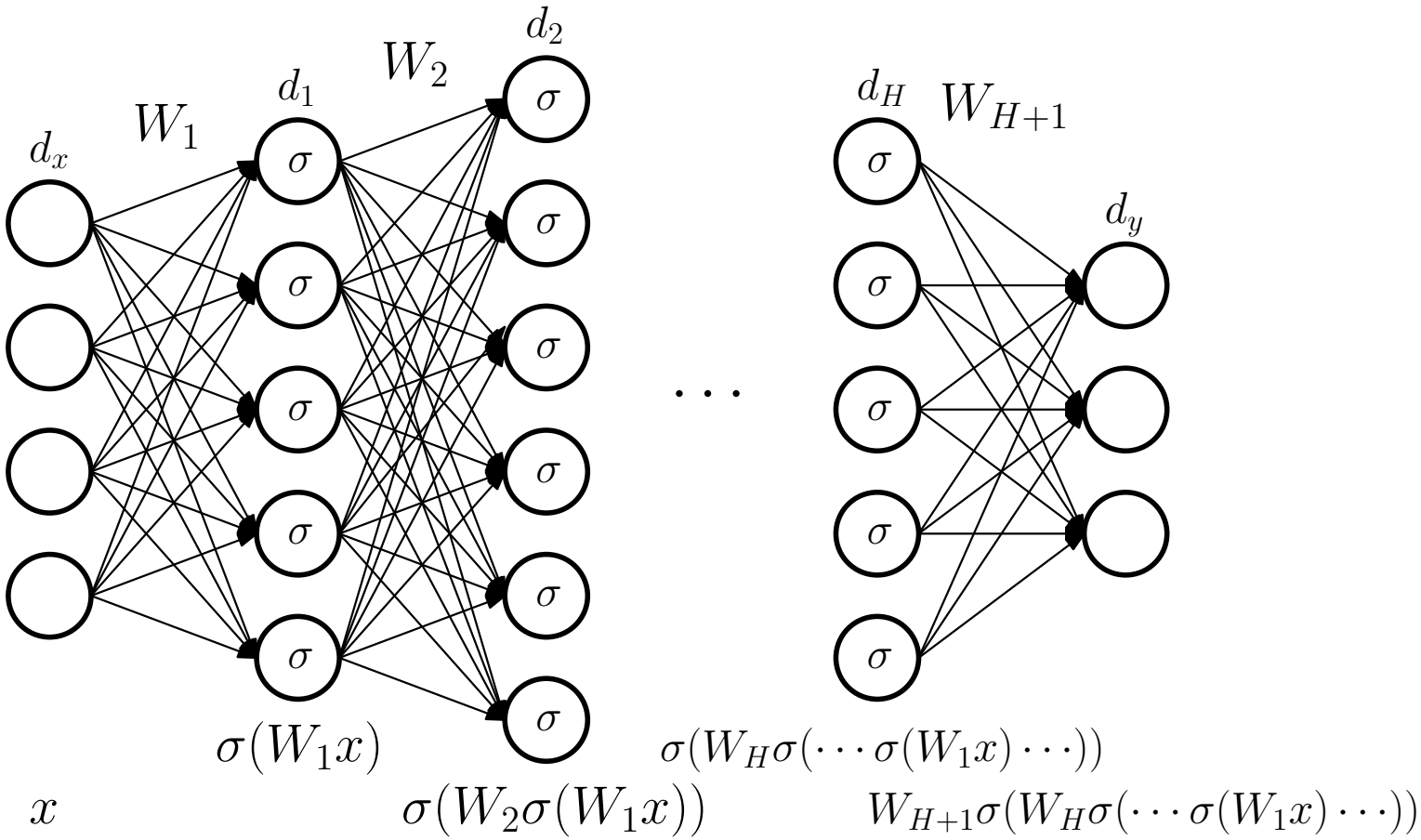
- m data points, input dimension d_x, output dimension d_y.
- Data matrix $X \in \mathbb{R}^{d_x \times m}$, label matrix $Y \in \mathbb{R}^{d_y \times m}$

- Goal: tune model parameters to fit the data.

Problem Setting: Deep Neural Networks

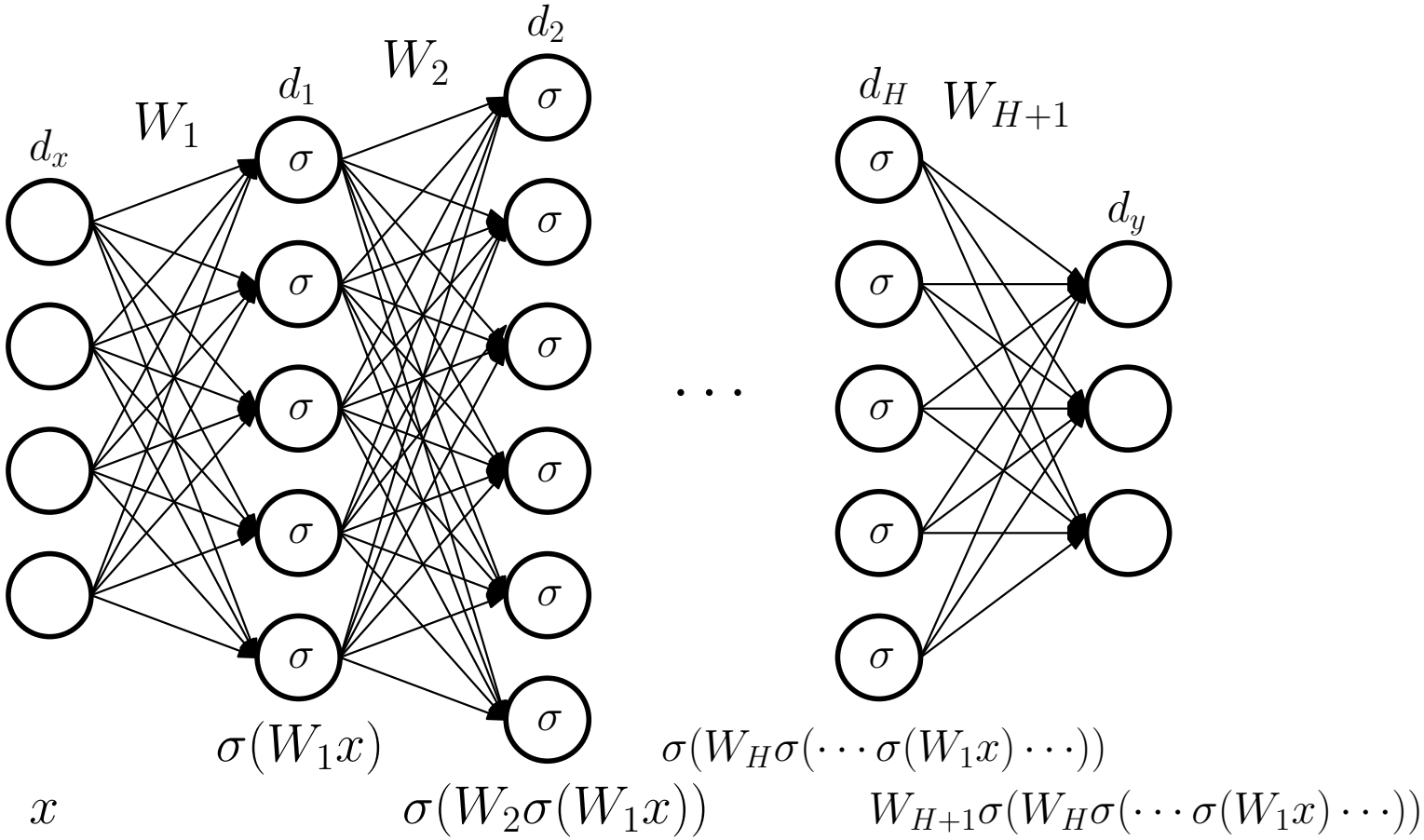


Problem Setting: Deep Neural Networks



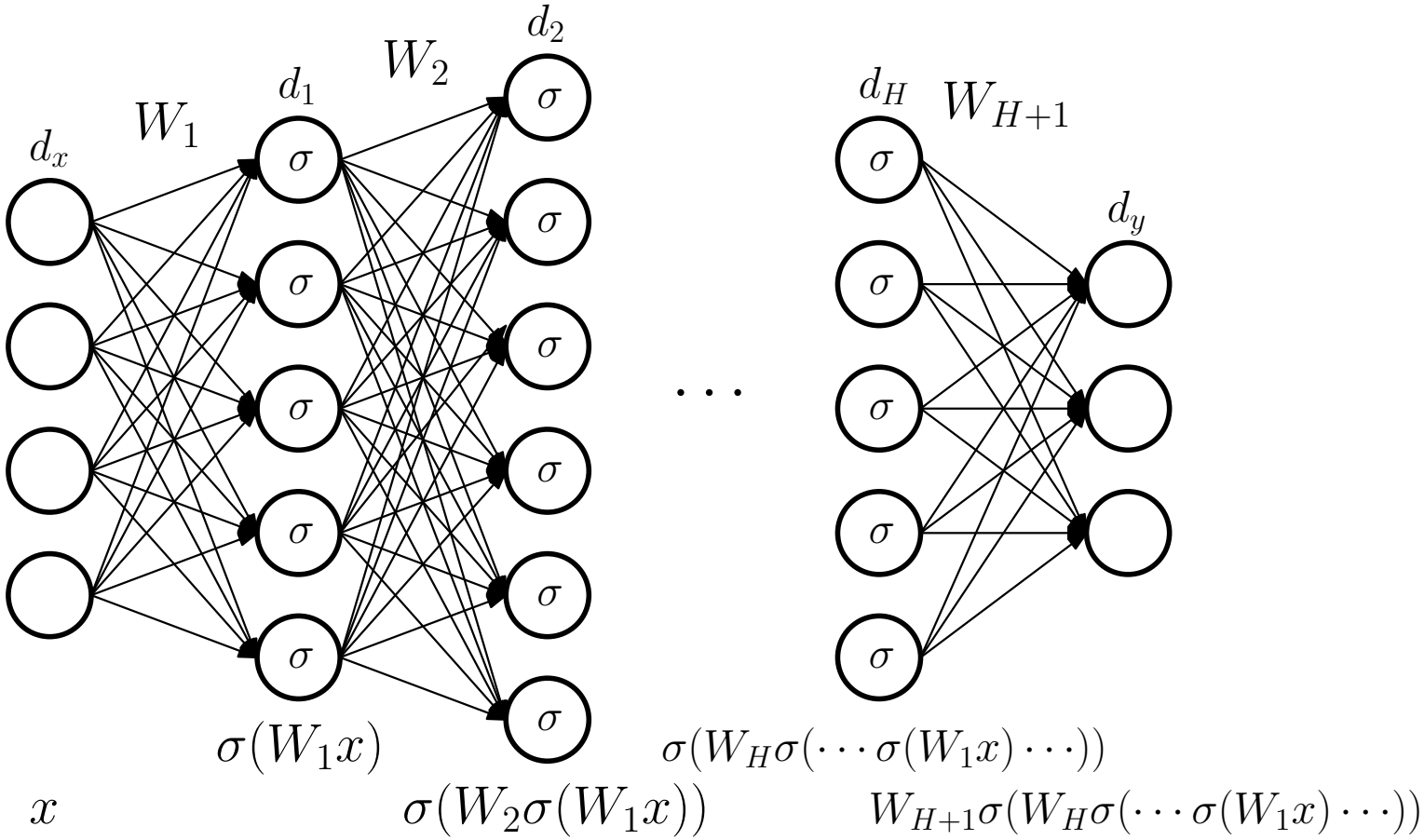
- H hidden layers of dimensions $d_1, d_2, \ldots, d_H$

Problem Setting: Deep Neural Networks



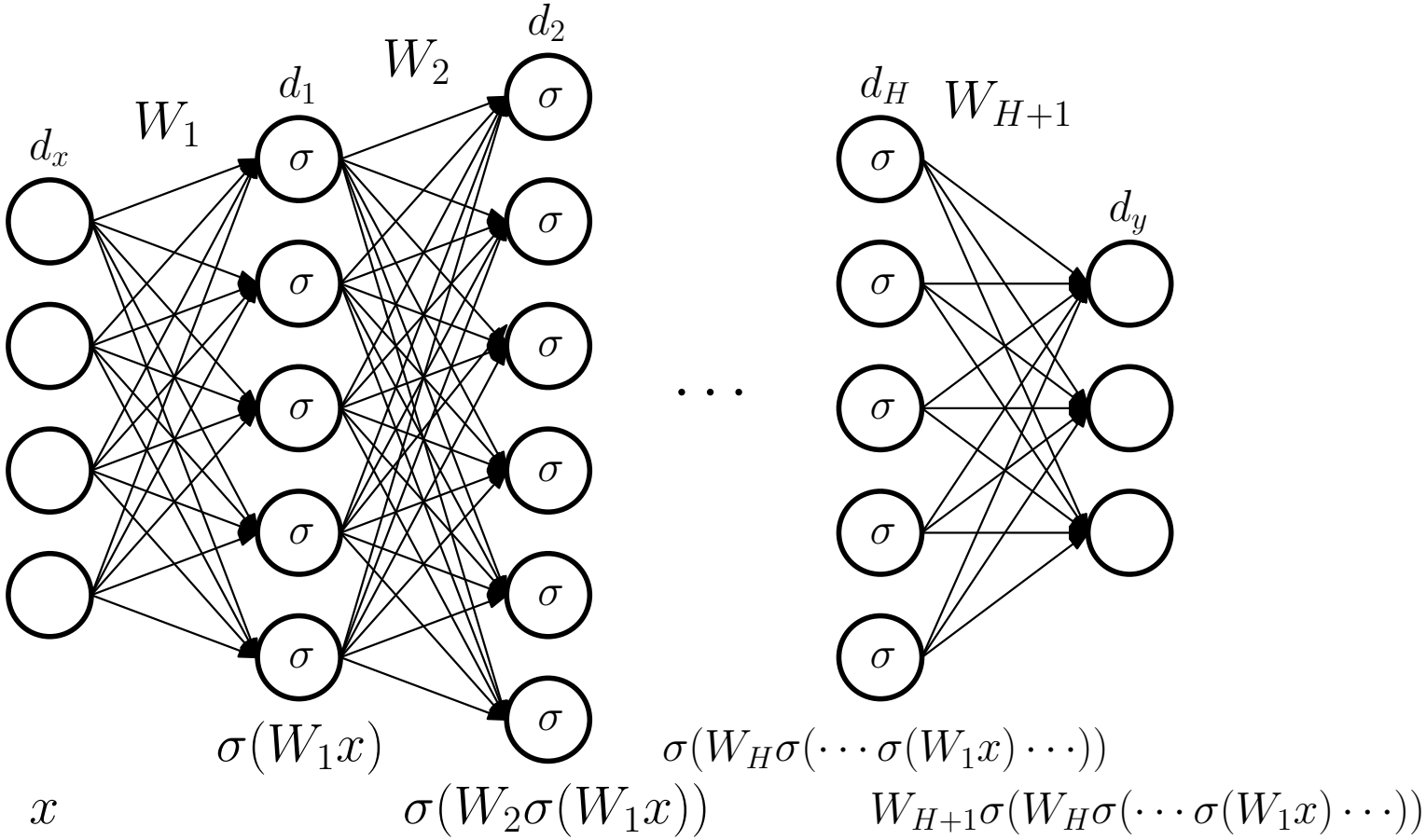
- H hidden layers of dimensions $d_1, d_2, \ldots, d_H$
- Parameter matrices $W_i \in \mathbb{R}^{d_i \times d_{i-1}}$ $(i = 1, \ldots, H+1)$

Problem Setting: Deep Neural Networks



- H hidden layers of dimensions $d_1, d_2, \ldots, d_H$
- Parameter matrices $W_i \in \mathbb{R}^{d_i \times d_{i-1}}$ $(i = 1, \ldots, H+1)$
- Nonlinear activation functions σ at hidden nodes

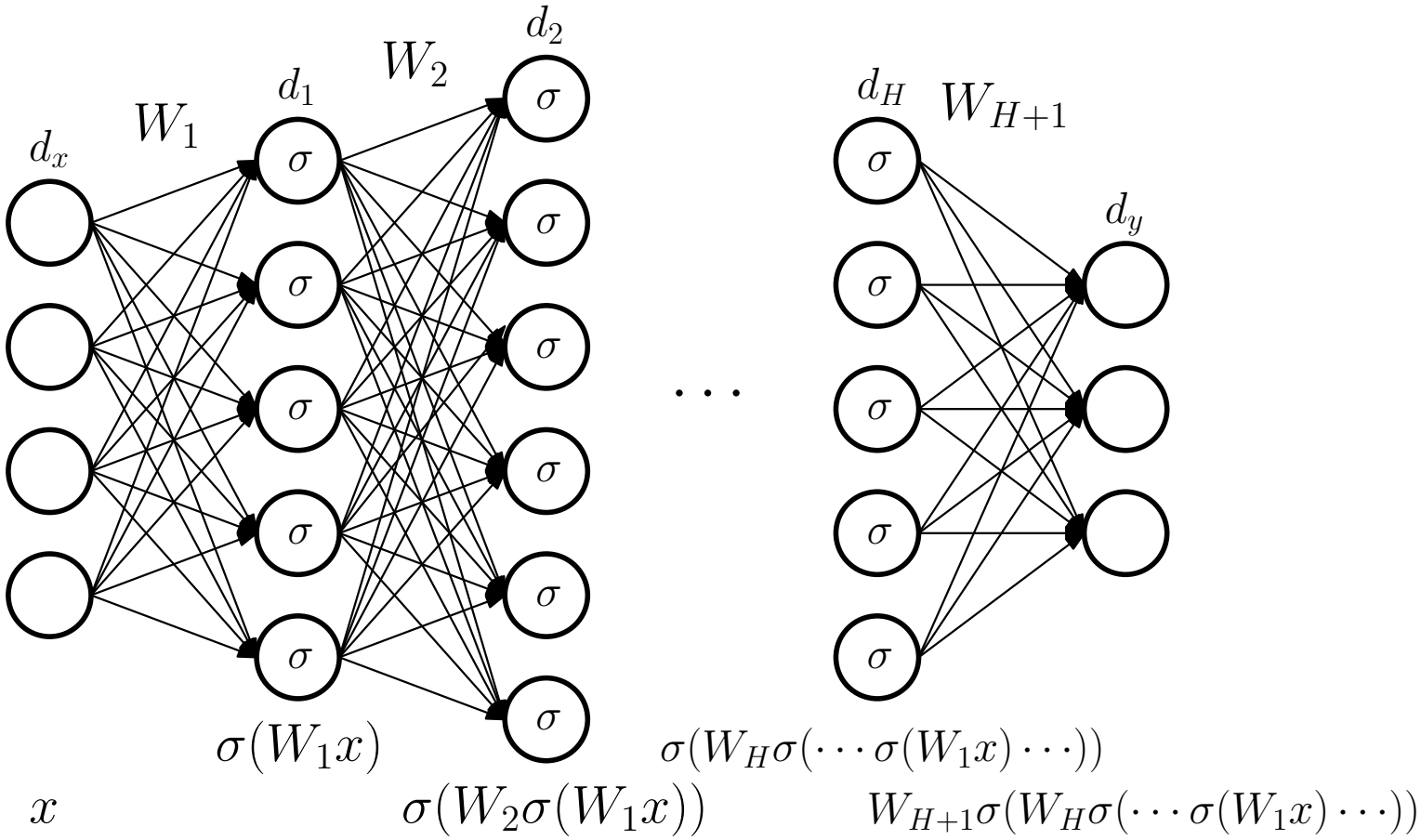
Problem Setting: Deep Neural Networks



- For given $x \in \mathbb{R}^{d_x}$, output $\hat{y} \in \mathbb{R}^{d_y}$ of network:

$$\hat{y} = W_{H+1}\sigma(W_H\sigma(\cdots\sigma(W_1 x)\cdots)).$$

Problem Setting: Deep Neural Networks



- Minimize empirical risk, over parameters $(W_i)_{i=1}^{H+1}$.

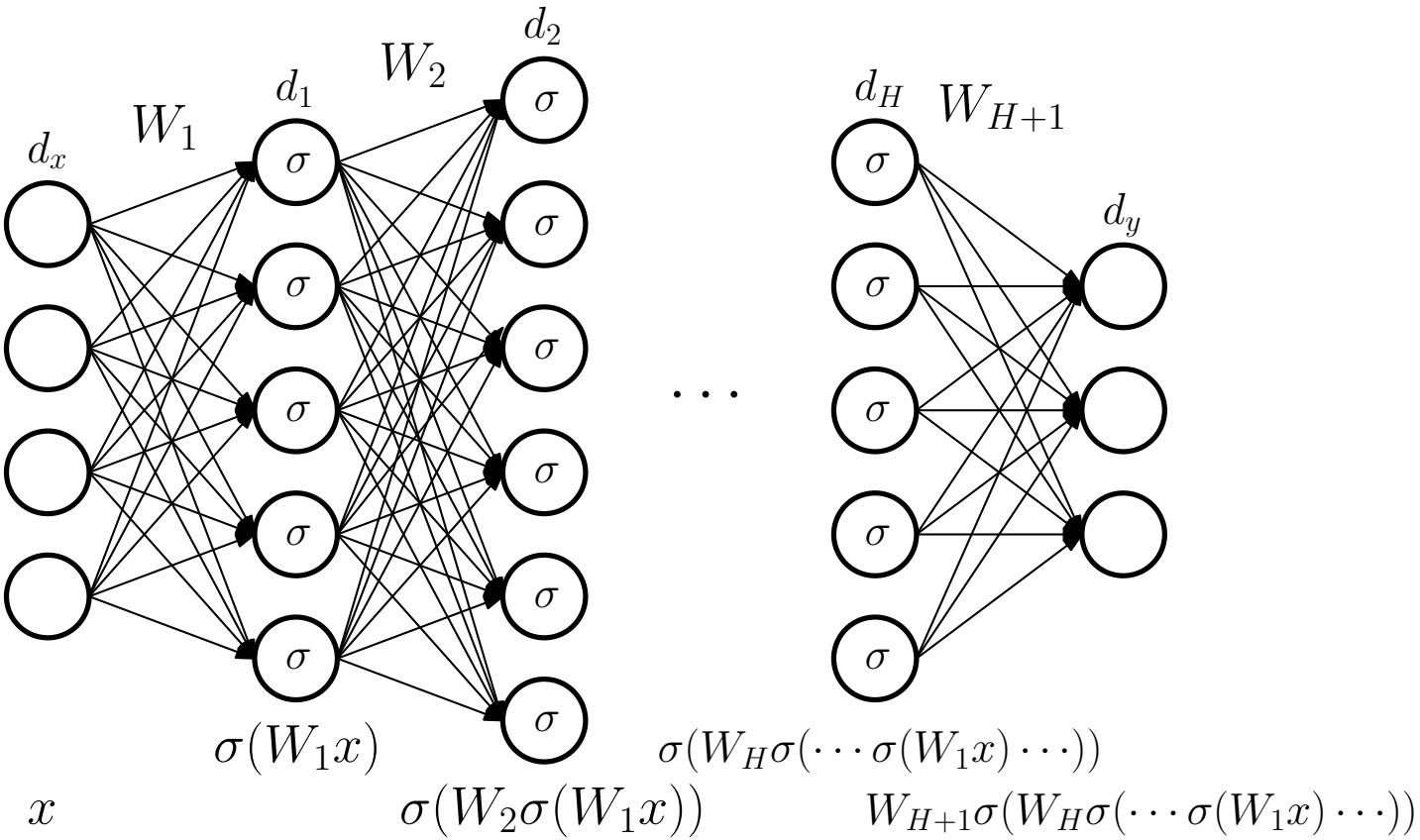
$$L((W_i)_{i=1}^{H+1}) := \frac{1}{2}\left\| W_{H+1}\sigma(W_H\sigma(\cdots\sigma(W_1 X)\cdots)) - Y \right\|_{\mathrm{F}}^2$$

Problem Setting: Deep Linear Neural Networks

- Deep *linear* neural networks: no nonlinear activation function σ.

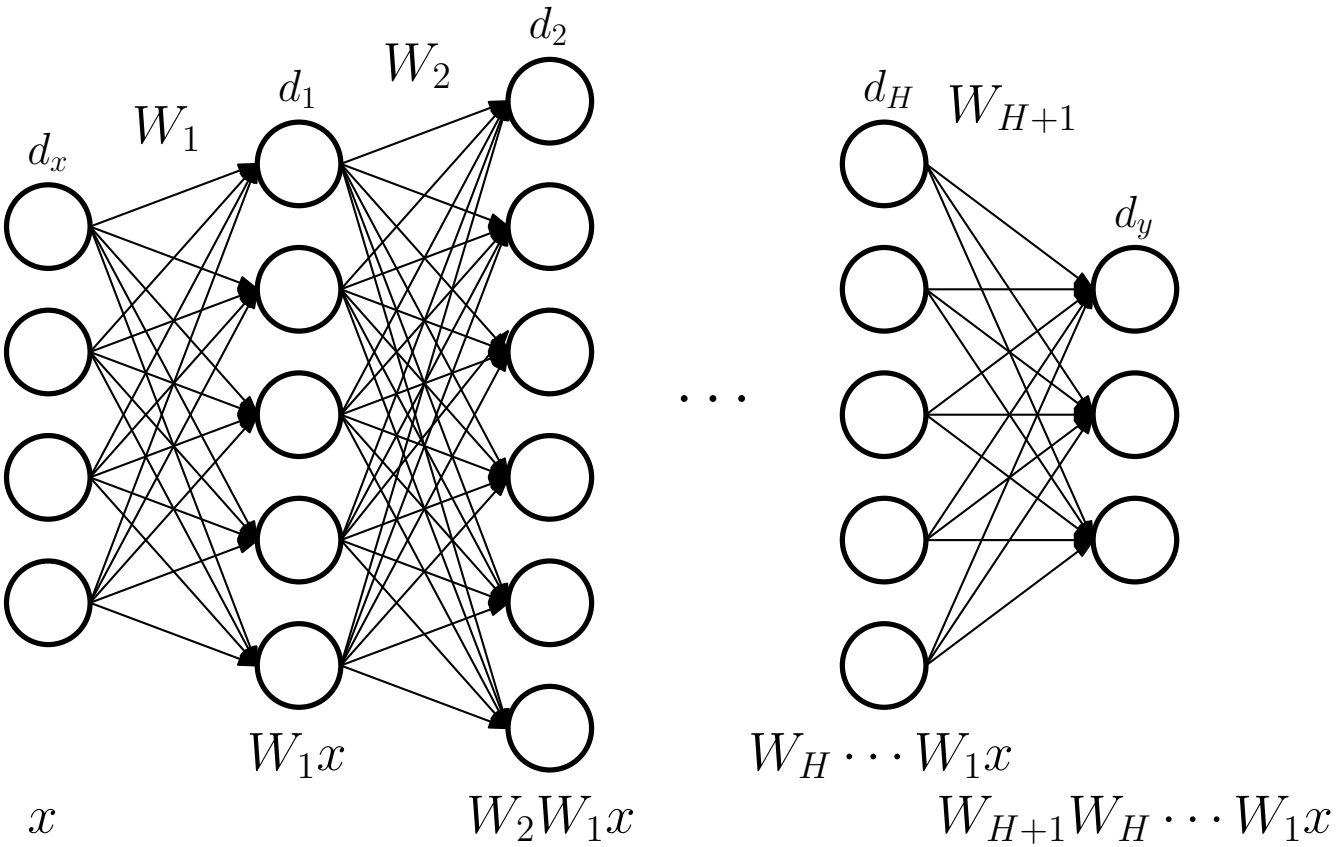
Problem Setting: Deep Linear Neural Networks

- Deep *linear* neural networks: no nonlinear activation function σ.

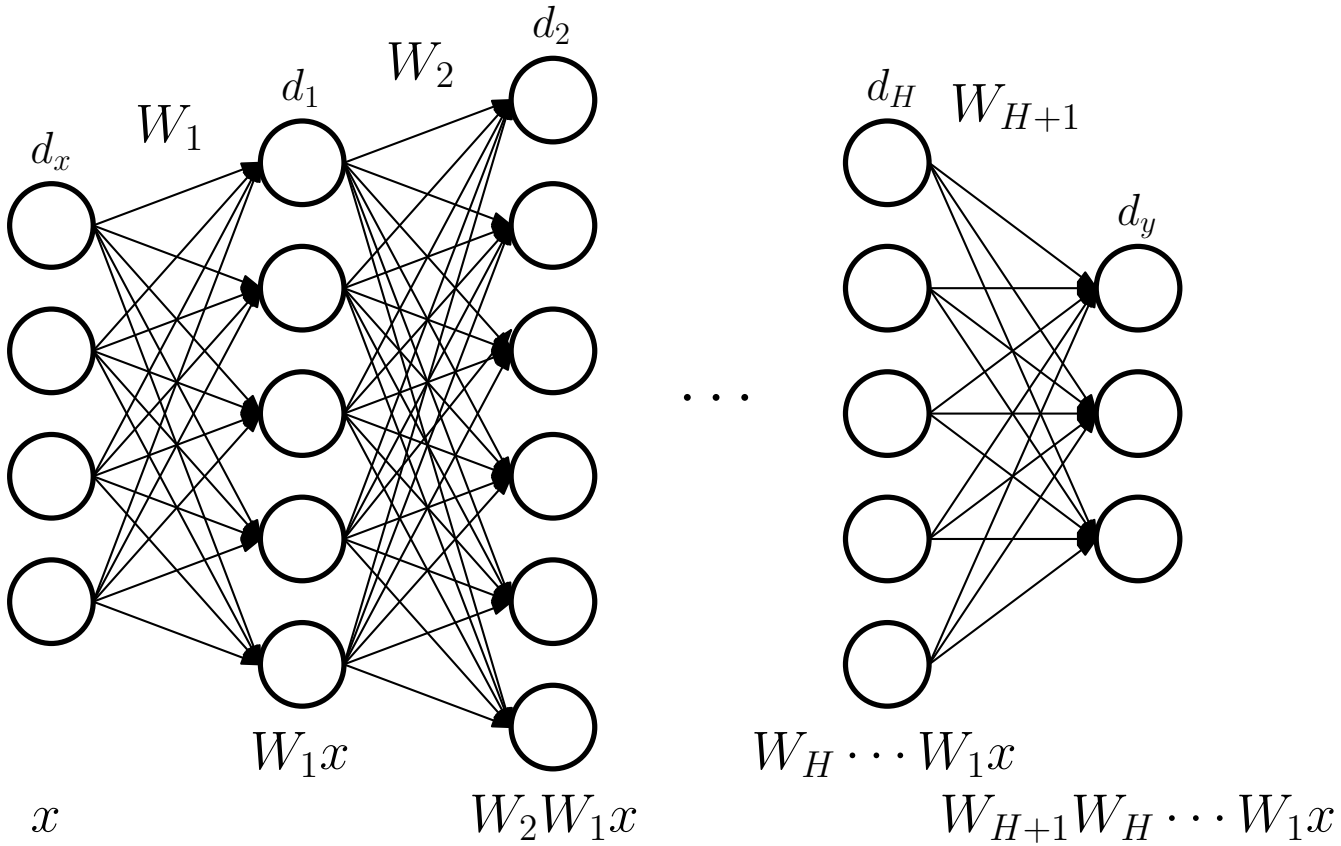


Problem Setting: Deep Linear Neural Networks

- Deep *linear* neural networks: no nonlinear activation function σ.



Problem Setting: Deep Linear Neural Networks



- Minimize empirical risk, over parameters $(W_i)_{i=1}^{H+1}$.

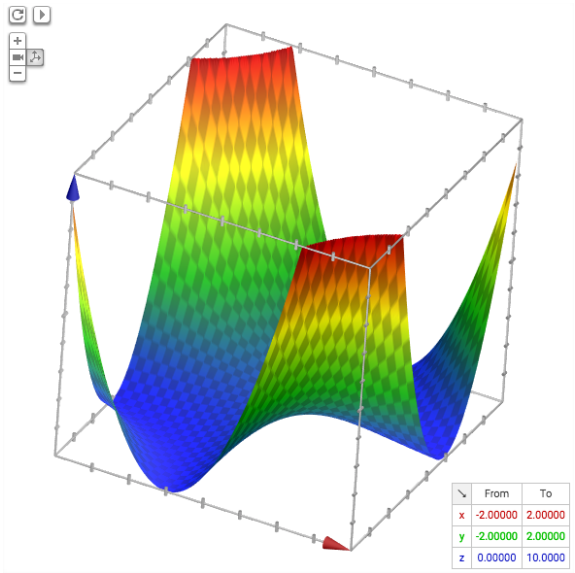
$$L((W_i)_{i=1}^{H+1}) := \frac{1}{2} \| W_{H+1} W_H \cdots W_1 X - Y \|_{\mathsf{F}}^2$$

Why Linear Neural Networks?

- No better than linear least squares, and the empirical risk is still nonconvex!

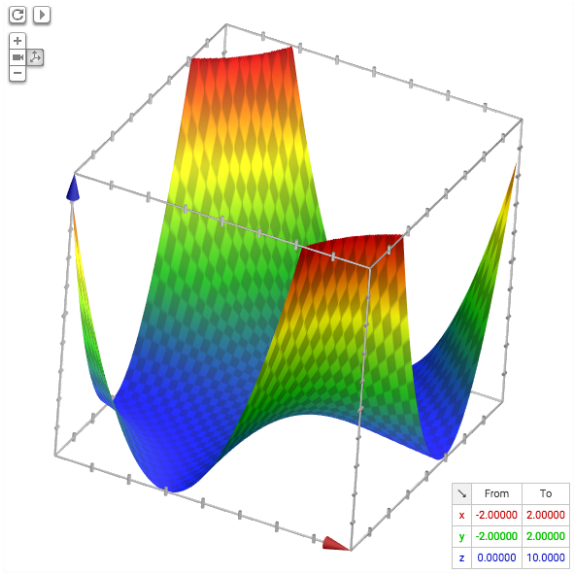
Why Linear Neural Networks?

- No better than linear least squares, and the empirical risk is still nonconvex!
- Example: $f(w_1, w_2) = (w_2 w_1 - 1)^2$ (c.f. $\|W_{H+1} W_H \cdots W_1 X - Y\|_F^2$)



Why Linear Neural Networks?

- No better than linear least squares, and the empirical risk is still nonconvex!
- Example: $f(w_1, w_2) = (w_2 w_1 - 1)^2$ (c.f. $\|W_{H+1} W_H \cdots W_1 X - Y\|_F^2$)



- However, linear nets have nice properties, which wish to extend to nonlinear nets.

Why Linear Neural Networks?

- Kawaguchi, 2016: Any critical point of empirical risk of deep linear neural networks is either a global minimum or a saddle point.

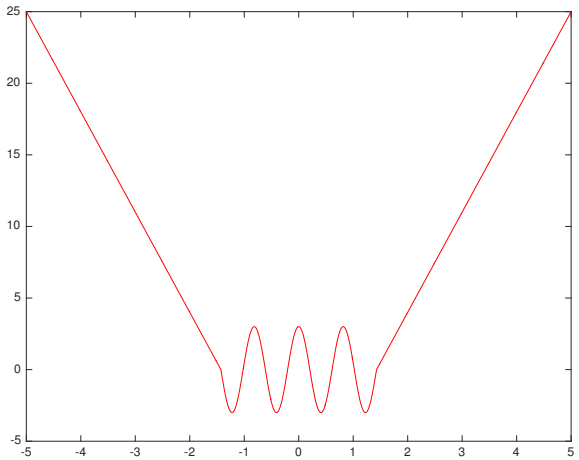
Why Linear Neural Networks?

- Kawaguchi, 2016: Any critical point of empirical risk of deep linear neural networks is either a global minimum or a saddle point.
  $\Rightarrow$ They have a nice property: "all local minima are globally optimal"

Why Linear Neural Networks?

- Kawaguchi, 2016: Any critical point of empirical risk of deep linear neural networks is either a global minimum or a saddle point.
  $\Rightarrow$ They have a nice property: "all local minima are globally optimal"

- A nonconvex function with "all local minima are globally optimal" property:



Our Contribution

- Kawaguchi, 2016: Any critical point of deep linear neural networks is either a global minimum or a saddle point.

Our Contribution

- Kawaguchi, 2016: Any critical point of deep linear neural networks is either a global minimum or a saddle point.
- Our result: prove conditions to *distinguish* global minima and saddle points.

Our Contribution

- Kawaguchi, 2016: Any critical point of deep linear neural networks is either a global minimum or a saddle point.
- Our result: prove conditions to *distinguish* global minima and saddle points.
- Better understanding of loss surface of deep linear neural networks.

Our Contribution

- Kawaguchi, 2016: Any critical point of deep linear neural networks is either a global minimum or a saddle point.
- Our result: prove conditions to *distinguish* global minima and saddle points.
- Better understanding of loss surface of deep linear neural networks.
- Efficiently checkable necessary and sufficient conditions for global optimality for a nonconvex problem. (typically not possible!)

Problem Setting: Deep Linear Neural Networks

- Assumptions (standard or removable):
  - $d_y \leq d_x$: to simplify presentation
  - $d_x \leq m$, $d_y \leq m$, matrices XX^T, YX^T full rank.
  - $YX^T(XX^T)^{-1}X$ has distinct singular values.

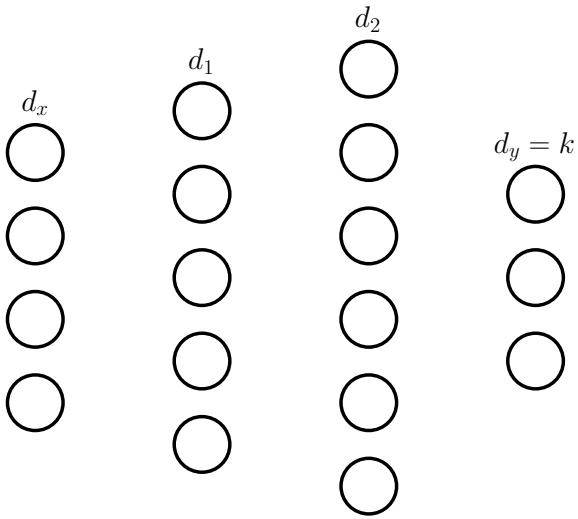
Problem Setting: Deep Linear Neural Networks

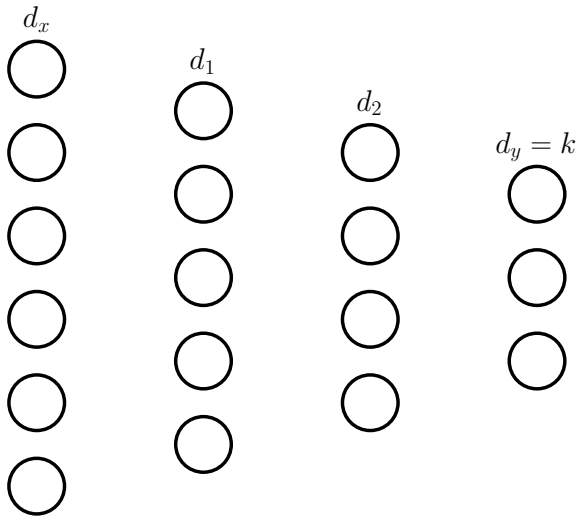
- Assumptions (standard or removable):
  - $d_y \leq d_x$: to simplify presentation
  - $d_x \leq m$, $d_y \leq m$, matrices XX^T, YX^T full rank.
  - $YX^T(XX^T)^{-1}X$ has distinct singular values.

- Notation:
  - k: minimum width of the network over all layers ($\min_j d_j$).
  - p: index of layer with minimum width k ($\operatorname{argmin}_j d_j$)

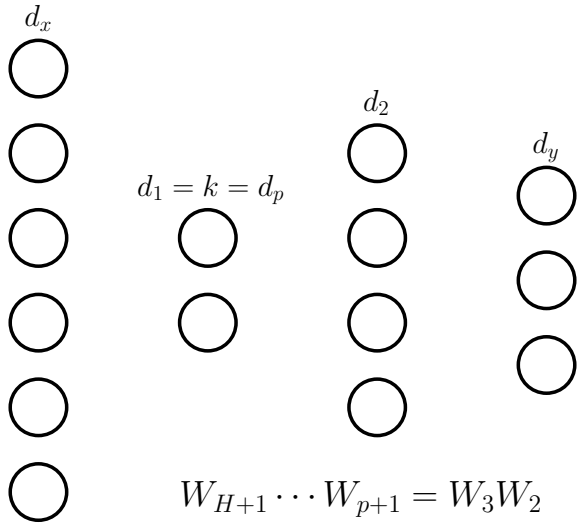
Main Theorems

- Theorem 1: All hidden layers are as wide as output layer ($k = d_y$)

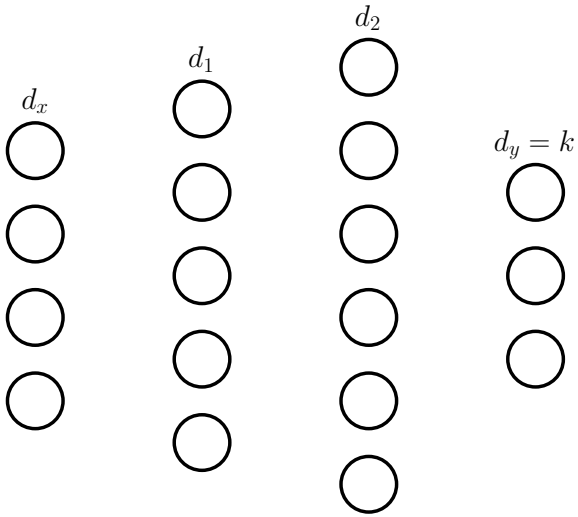


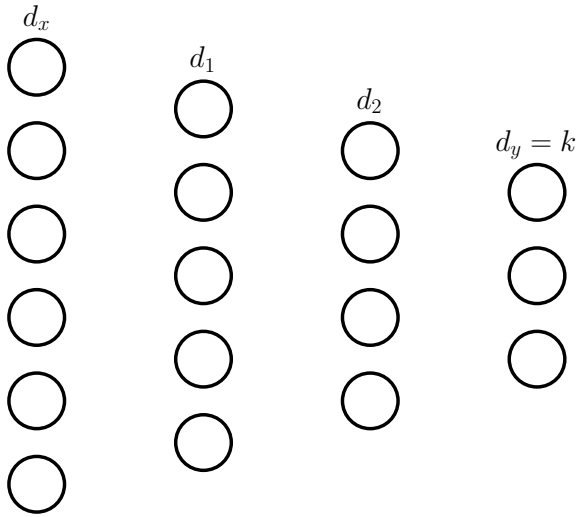


- Theorem 2: $\exists$ hidden layer narrower than output layer ($k < d_y$)



Main Theorems





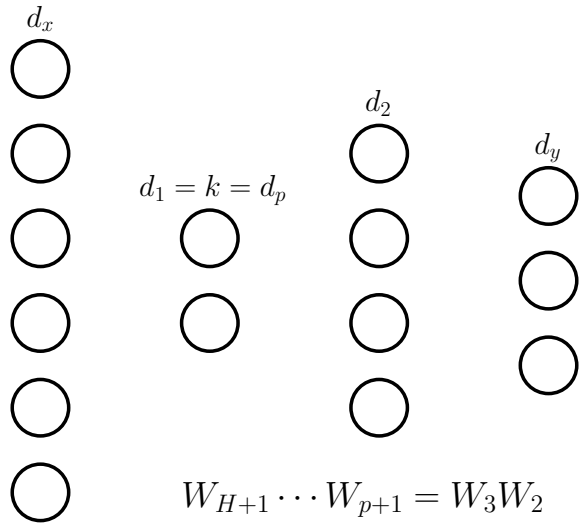
Theorem 1

If all hidden layers are at least as wide as output layer ($k = d_y$), a critical point of the empirical risk is a global minimum iff $W_{H+1} \cdots W_1 \in \mathbb{R}^{d_y \times d_x}$ is full rank. In other words, define

$$\mathcal{V}_1 = \{(W_1, \ldots, W_{H+1}) : \text{rank}(W_{H+1} \cdots W_1) = k\}.$$

Then, every critical point in $\mathcal{V}_1$ is a global minimum of $L((W_i)_{i=1}^{H+1})$, while every critical point in $\mathcal{V}_1^c$ is a saddle point of $L((W_i)_{i=1}^{H+1})$.

Main Theorems



Theorem 2

If there exists a hidden layer narrower than output ($k < d_y$), then a critical point of the empirical risk is a global minimum iff $W_{H+1} \cdots W_1$ has rank k and column spaces of $W_{H+1} \cdots W_{p+1} \in \mathbb{R}^{d_y \times k}$ and $U \in \mathbb{R}^{d_y \times k}$ are identical (U is a matrix-valued constant that only depends on X, Y and k). That is, define

$$\mathcal{V}_2 = \{(W_1, \ldots, W_{H+1}) : \quad \text{rank}(W_{H+1} \cdots W_1) = k, \\ \text{col}(W_{H+1} \cdots W_{p+1}) = \text{col}(U)\}.$$

Then, every critical point in $\mathcal{V}_2$ is a global minimum of $L((W_i)_{i=1}^{H+1})$, while every critical point in $\mathcal{V}_2^c$ is a saddle point of $L((W_i)_{i=1}^{H+1})$.

Discussion

- For proofs: Global optimality conditions for deep neural networks, to appear at ICLR 2018 (https://arxiv.org/abs/1707.02444)

Discussion

- For proofs: Global optimality conditions for deep neural networks, to appear at ICLR 2018 (https://arxiv.org/abs/1707.02444)

- Extensions: different losses, nonlinear settings, etc — coming soon!

Discussion

- For proofs: Global optimality conditions for deep neural networks, to appear at ICLR 2018 (https://arxiv.org/abs/1707.02444)
- Extensions: different losses, nonlinear settings, etc — coming soon!
- Questions?