

Back Propagation

Machine Learning
CS4824/ECE4424
Bert Huang
Virginia Tech

Outline

- Logistic regression and perceptron as neural networks
- Likelihood gradient for 2-layered neural network
- General recipe for back propagation

Back Propagation

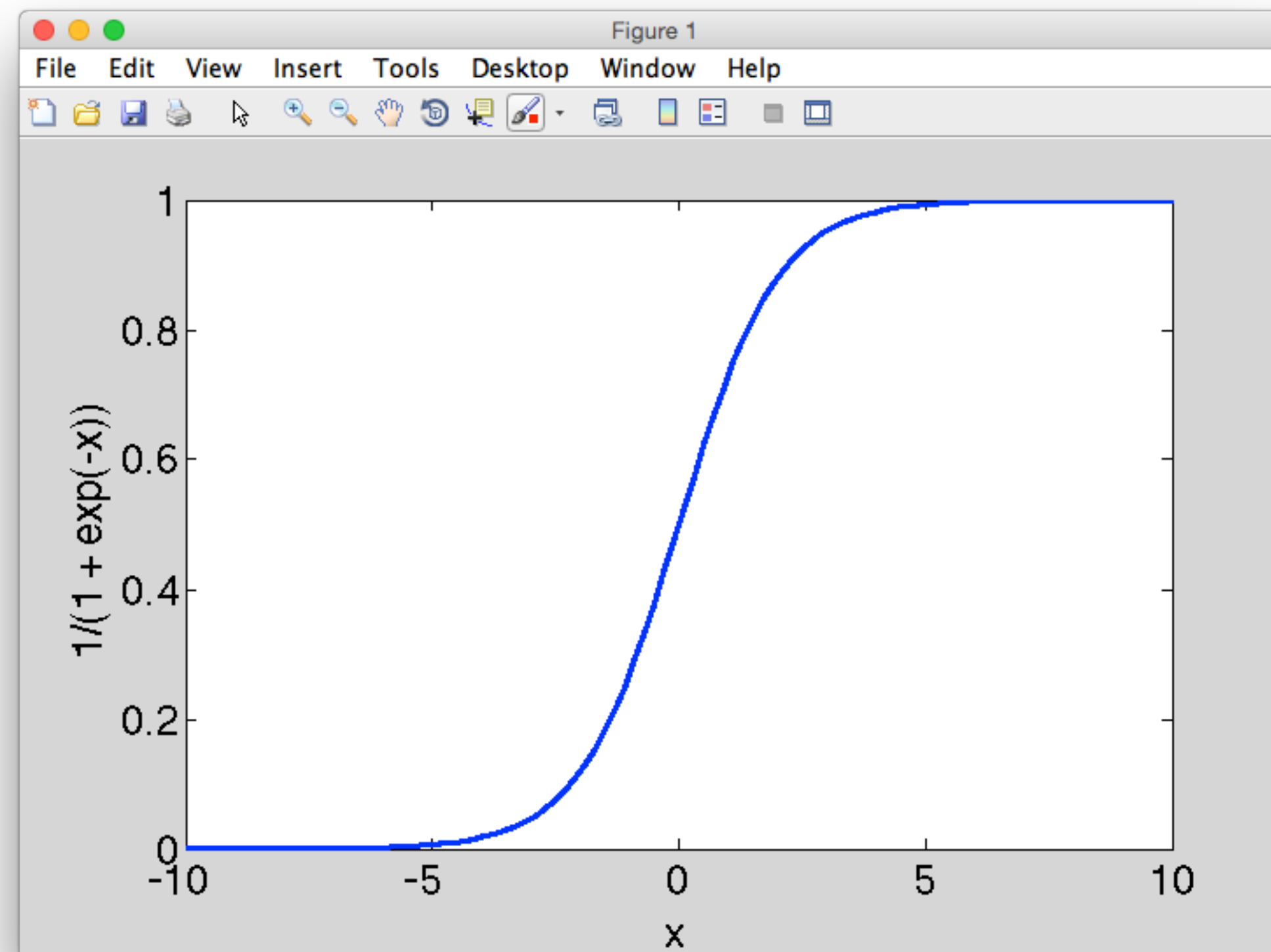
- Back propagation:
 - Compute hidden unit activations: **forward propagation**
 - Compute gradient at output layer: error
 - Propagate error back one layer at a time
- Chain rule via dynamic programming

Plan

- Start by computing gradients by hand (we'll get messy!)
- See common patterns for organizing back-propagation algorithm
- Show general matrix form for feed-forward neural networks

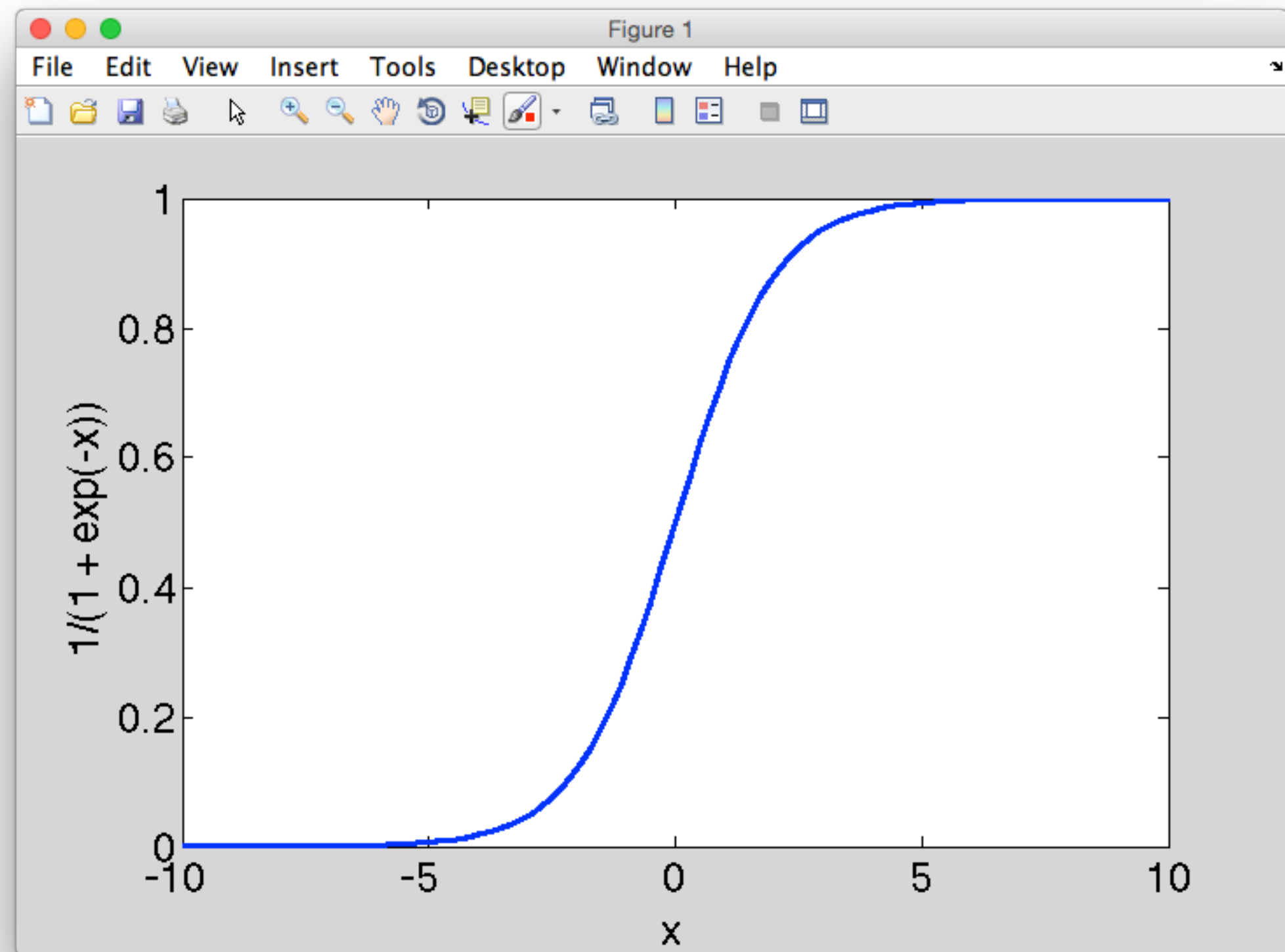
Logistic Squashing Function

$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$



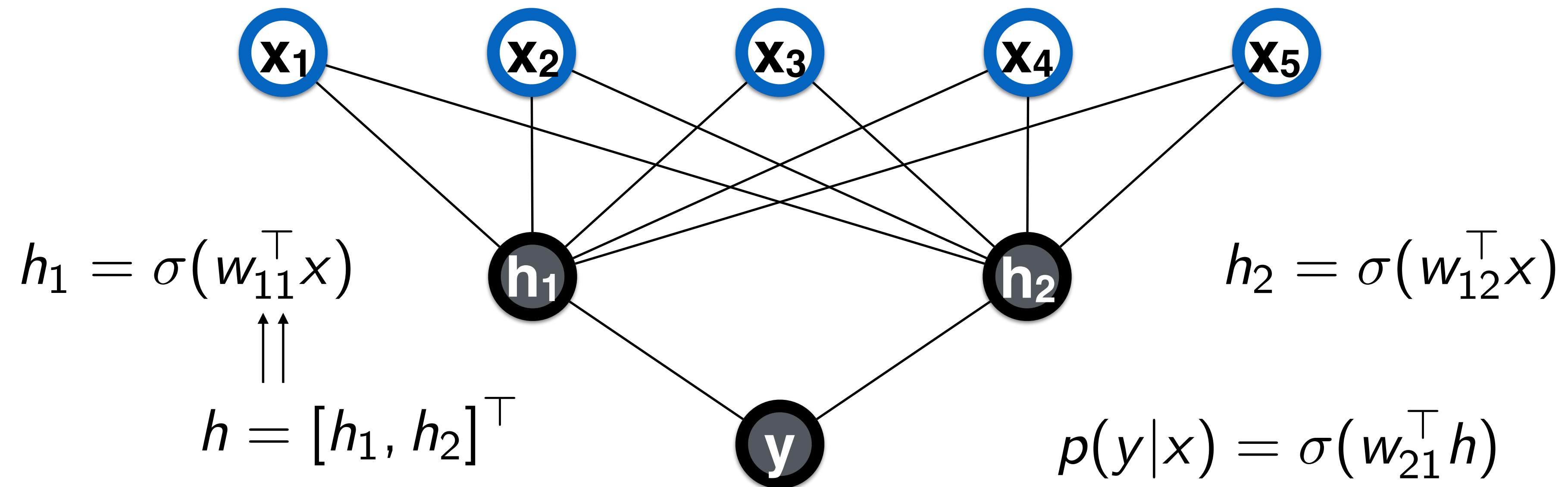
Logistic Squashing Function

$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$



$$\frac{d \sigma(x)}{d x} = \sigma(x)(1 - \sigma(x))$$

Multi-Layered Perceptron



$$p(y|x) = \sigma \left(w_{21}^\top \left[\sigma(w_{11}^\top x), \sigma(w_{12}^\top x) \right]^\top \right)$$

Gradients

$$p(y|x) = \sigma \left(w_{21}^\top \left[\sigma(w_{11}^\top x), \sigma(w_{12}^\top x) \right]^\top \right) \quad \nabla_{w_{21}} \mathbb{ll} = \sum_{i=1}^n \frac{1}{p(y_i|x_i)} \times \nabla_{w_{21}} p(y_i|x_i)$$

$$p(y|x) = \sigma(w_{21}^\top h) \quad \nabla_{w_{21}} \mathbb{ll} = \sum_{i=1}^n \frac{1}{p(y_i|x_i)} \times \nabla_{w_{21}} \sigma(y_i w_{21}^\top h)$$

$$\mathbb{ll}(W) = \sum_{i=1}^n \log p(y_i|x_i) \quad \nabla_{w_{21}} \mathbb{ll} = \sum_{i=1}^n \frac{\sigma(y_i w_{21}^\top h)(1 - \sigma(y_i w_{21}^\top h))}{\sigma(y_i w_{21}^\top h)} \times \nabla_{w_{21}} y_i w_{21}^\top h$$

$$\nabla_{w_{21}} \mathbb{ll} = \sum_{i=1}^n (1 - \sigma(y_i w_{21}^\top h)) \times \nabla_{w_{21}} y_i w_{21}^\top h$$

Gradients

$$p(y|x) = \sigma \left(w_{21}^\top \left[\sigma(w_{11}^\top x), \sigma(w_{12}^\top x) \right]^\top \right)$$

$$p(y|x) = \sigma(w_{21}^\top h)$$

$$\ell(W) = \sum_{i=1}^n \log p(y_i|x_i)$$

$$\nabla_{w_{21}} \ell = \sum_{i=1}^n (1 - \sigma(y_i w_{21}^\top h)) \times \nabla_{w_{21}} y_i w_{21}^\top h$$

$$\nabla_{w_{21}} \ell = \sum_{i=1}^n (1 - \sigma(y_i w_{21}^\top h)) y_i h$$

$$\nabla_{w_{21}} \ell = \sum_{i=1}^n \begin{cases} (1 - \sigma(w_{21}^\top h)) h & \text{if } y_i = +1 \\ (-1 + \sigma(-w_{21}^\top h)) h & \text{if } y_i = -1 \end{cases}$$

$$\nabla_{w_{21}} \ell = \sum_{i=1}^n (I(y_i = 1) - \sigma(w_{21}^\top h)) h$$

Gradients

$$p(y|x) = \sigma \left(w_{21}^\top \left[\sigma(w_{11}^\top x), \sigma(w_{12}^\top x) \right]^\top \right) \quad \nabla_{w_{11}} \text{ll} = \sum_{i=1}^n \frac{1}{p(y_i|x_i)} \times \nabla_{w_{11}} p(y_i|x_i)$$

$$p(y|x) = \sigma(w_{21}^\top h) \quad \nabla_{w_{11}} \text{ll} = \sum_{i=1}^n (I(y_i = 1) - \sigma(w_{21}^\top h)) \nabla_{w_{11}} w_{21}^\top h$$

$$\text{ll}(W) = \sum_{i=1}^n \log p(y_i|x_i) \quad \nabla_{w_{11}} \text{ll} = \sum_{i=1}^n (I(y_i = 1) - \sigma(w_{21}^\top h)) w_{21}^\top (\nabla_{w_{11}} h)$$

$$\nabla_{w_{11}} \text{ll} = \sum_{i=1}^n (I(y_i = 1) - \sigma(w_{21}^\top h)) w_{21} [1] \nabla_{w_{11}} \sigma(w_{11}^\top x_i)$$

Gradients

$$p(y|x) = \sigma \left(w_{21}^\top \left[\sigma(w_{11}^\top x), \sigma(w_{12}^\top x) \right]^\top \right) \quad \nabla_{w_{11}} \text{ll} = \sum_{i=1}^n \frac{1}{p(y_i|x_i)} \times \nabla_{w_{11}} p(y_i|x_i)$$

$$p(y|x) = \sigma(w_{21}^\top h) \quad \nabla_{w_{11}} \text{ll} = \sum_{i=1}^n (I(y_i = 1) - \sigma(w_{21}^\top h)) \nabla_{w_{11}} w_{21}^\top h$$

$$\text{ll}(W) = \sum_{i=1}^n \log p(y_i|x_i) \quad \nabla_{w_{11}} \text{ll} = \sum_{i=1}^n (I(y_i = 1) - \sigma(w_{21}^\top h)) w_{21}^\top (\nabla_{w_{11}} h)$$

$$\nabla_{w_{11}} \text{ll} = \sum_{i=1}^n (I(y_i = 1) - \sigma(w_{21}^\top h)) w_{21} [1] \nabla_{w_{11}} \sigma(w_{11}^\top x_i)$$

$$\nabla_{w_{11}} \text{ll} = \sum_{i=1}^n (I(y_i = 1) - \sigma(w_{21}^\top h)) w_{21} [1] \sigma(w_{11}^\top x_i) (1 - \sigma(w_{11}^\top x_i)) x_i$$

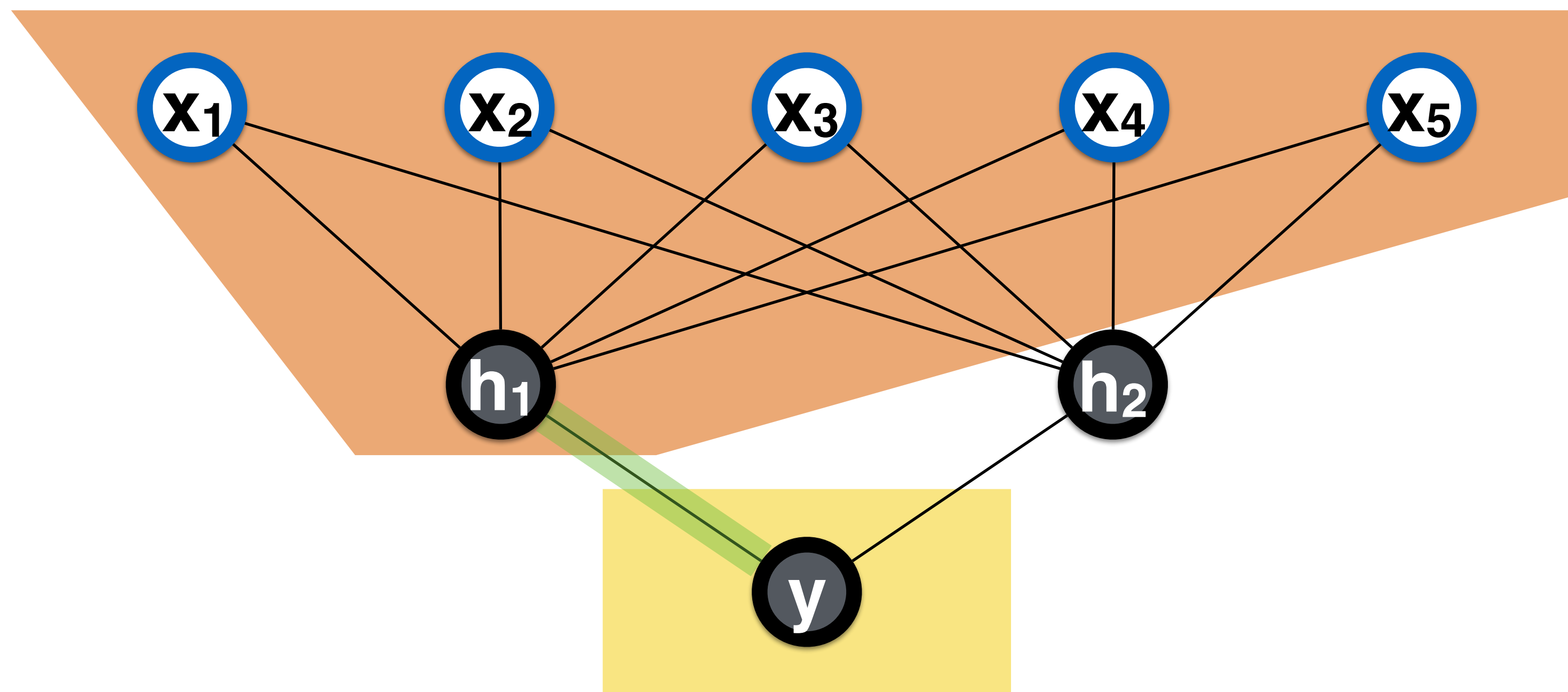
$$\ell(W) = \sum_{i=1}^n \log p(y_i|x_i) = \sum_{i=1}^n \log \sigma \left(y_i w_{21}^\top [\sigma(w_{11}^\top x_i), \sigma(w_{12}^\top x_i)]^\top \right)$$

$$\nabla_{w_{11}} \ell = \sum_{i=1}^n \left(\log \sigma(y_i w_{21}^\top h) \right) w_{21}^\top h \sigma(w_{11}^\top x_i) (1 - \sigma(w_{11}^\top x_i)) x_i$$

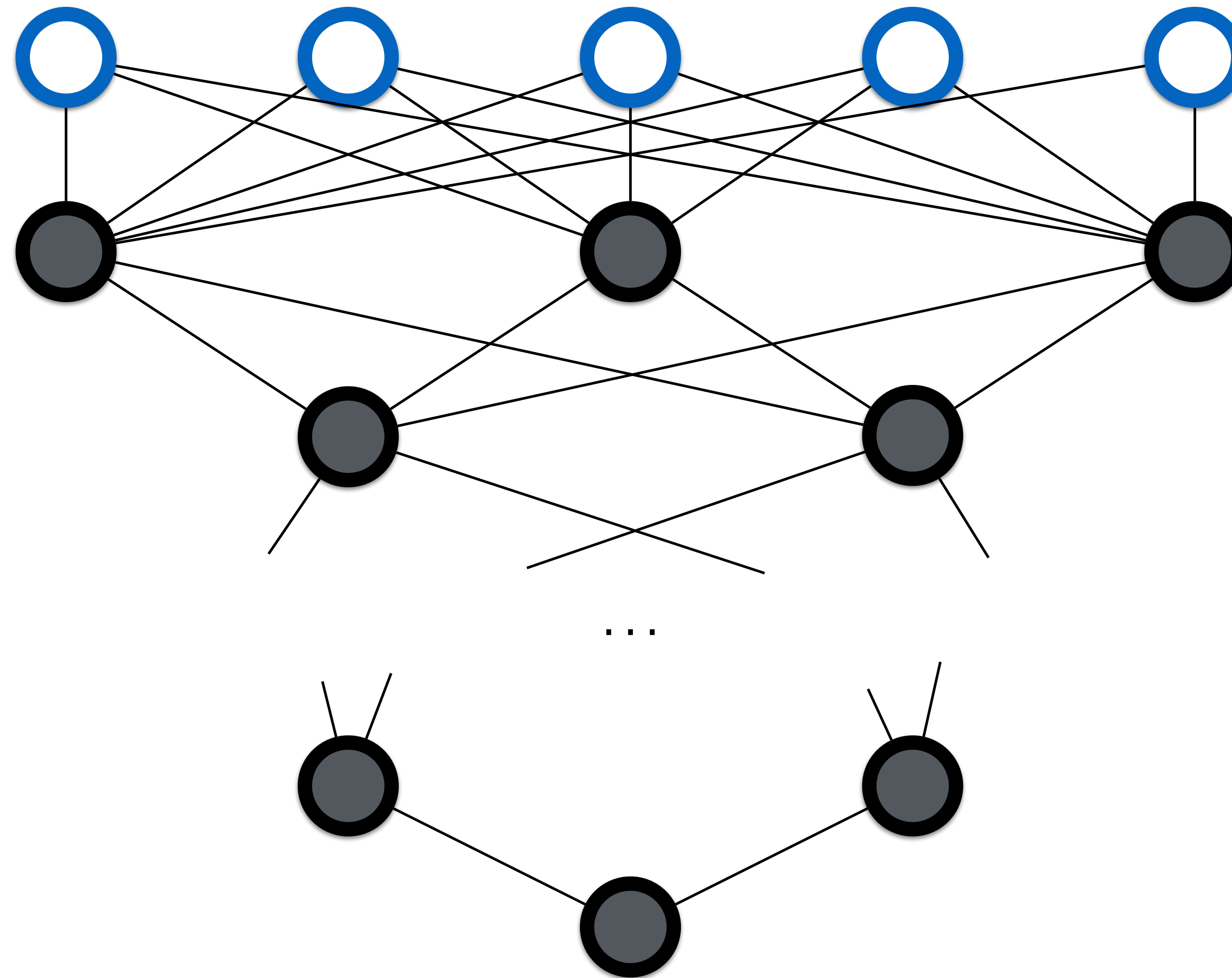
$h_1 = \sigma(w_{11}^\top x)$

$$\nabla_{w_{11}} \mathcal{L} = \sum_{i=1}^n \underbrace{(I(y_i = 1) - \sigma(w_{21}^\top h))}_{\text{raw error}} \underbrace{w_{21} [1]}_{\text{blame for error}} \underbrace{\sigma(w_{11}^\top x_i) (1 - \sigma(w_{11}^\top x_i)) x_i}_{\text{gradient of blamed error}}$$

$\log \sigma(y_i w_{21}^\top h)$
 $w_{21}^\top h$
 $h_1 = \sigma(w_{11}^\top x)$



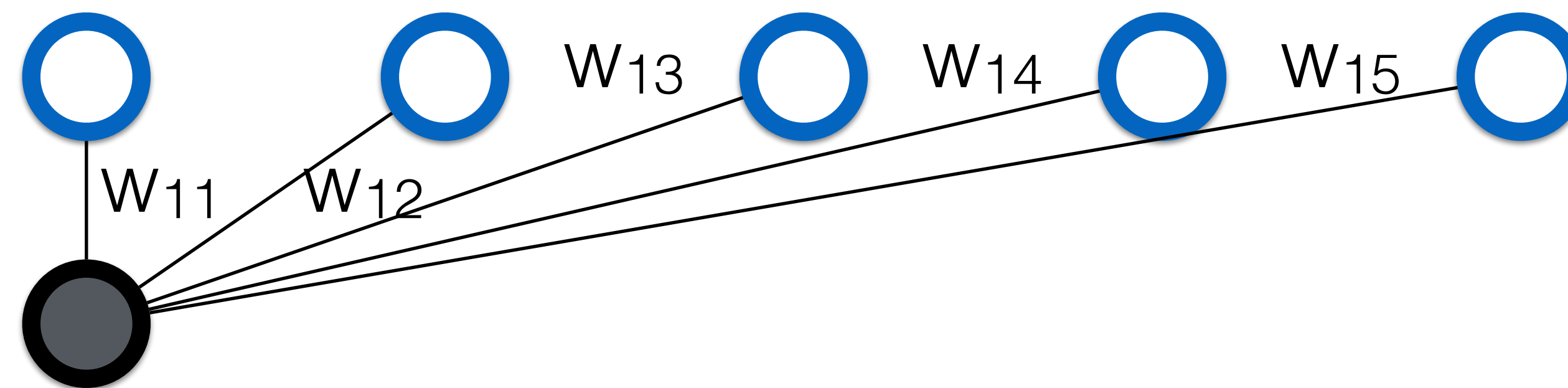
Matrix Form



x

$$h_1 = s(W_1 x)$$

Matrix Form



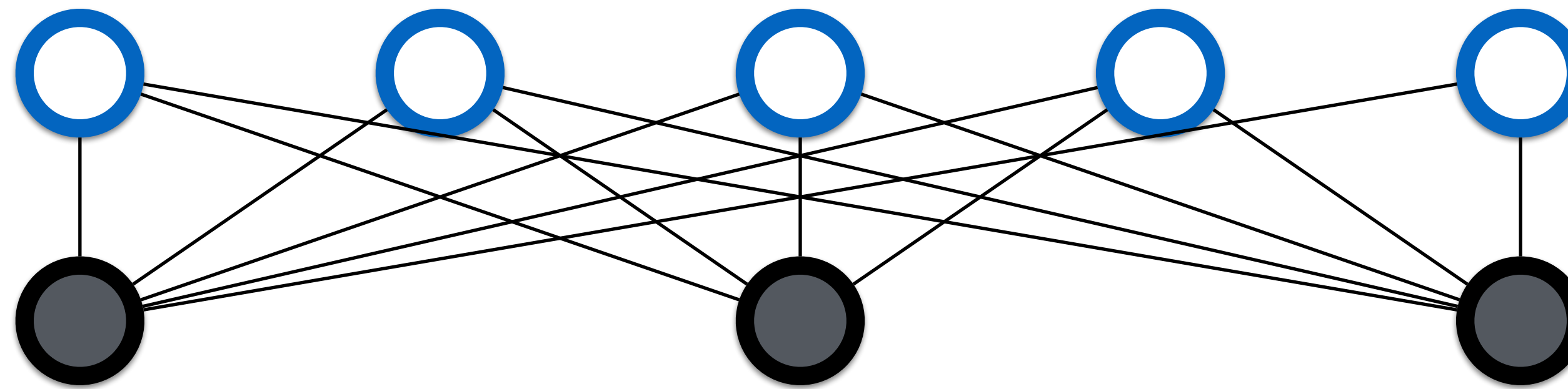
$$h_1 = s(W_1 x)$$

$$s(v) = [s(v_1), s(v_2), s(v_3), \dots]^T$$

$W_1 =$

W_{11}	W_{12}	W_{13}	W_{14}	W_{15}

Matrix Form



x

$$h_1 = s(W_1 x)$$

$$s(v) = [s(v_1), s(v_2), s(v_3), \dots]^T$$

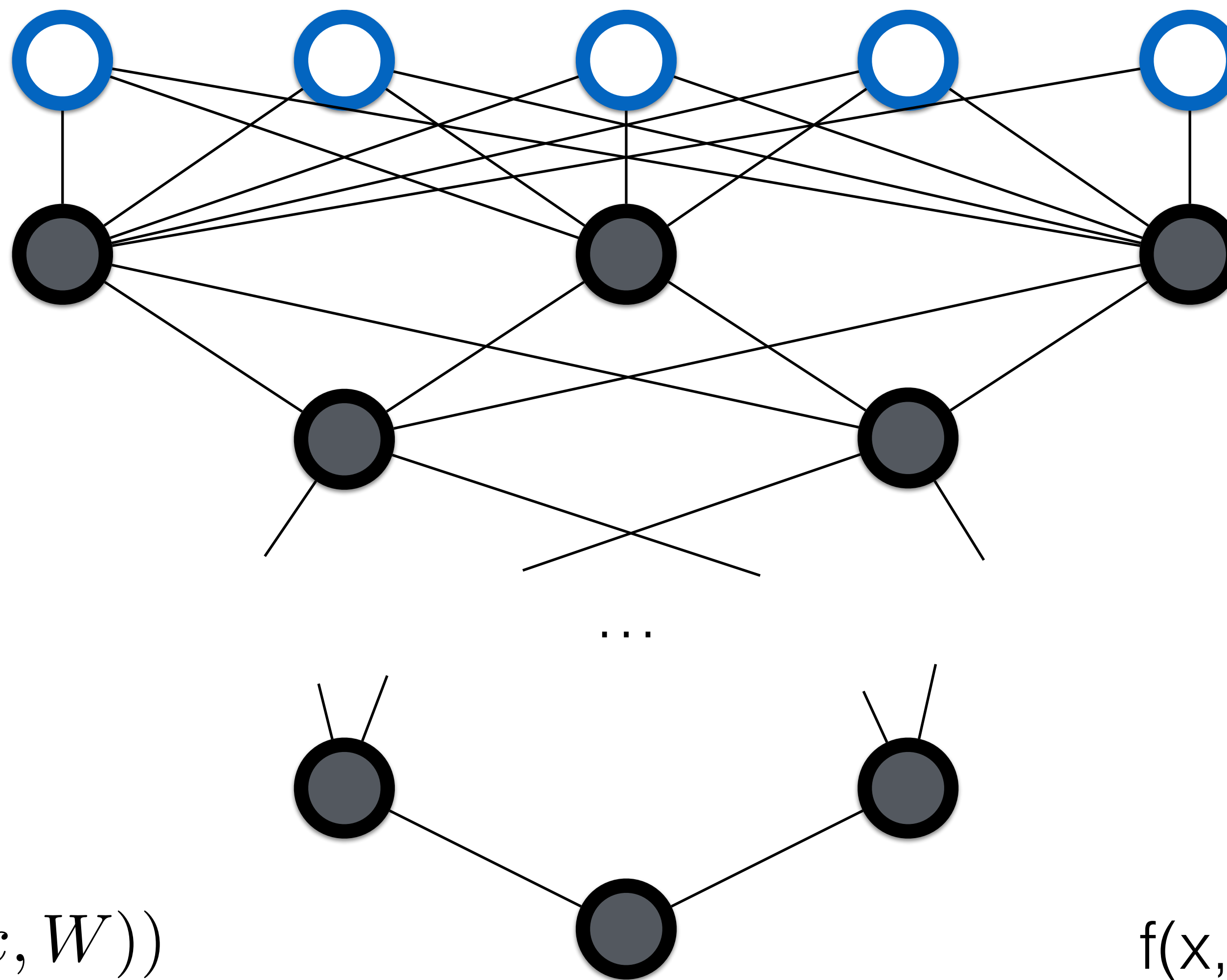
$W_1 =$

W_{11}	W_{12}	W_{13}	W_{14}	W_{15}
W_{21}	W_{22}	W_{23}	W_{24}	W_{25}
W_{31}	W_{32}	W_{33}	W_{34}	W_{35}

of output units

of input units

Matrix Form



$$J(W) = \ell(f(x, W))$$

$$f(x, W) = s(W_m h_{m-1})$$

Matrix Gradient Recipe

$$h_1 = s(W_1 x)$$

$$\nabla_{W_1} J = \delta_1 x^\top$$

$$h_2 = s(W_2 h_1)$$

$$\nabla_{W_i} J = \delta_i h_{i-1}^\top$$

...

$$\delta_i = (W_{i+1}^\top \delta_{i+1}) \odot s'(W_i h_{i-1})$$

$$h_{m-1} = s(W_{m-1} h_{m-2})$$

$$\nabla_{W_{m-1}} J = \delta_{m-1} h_{m-2}^\top$$

$$\delta_{m-1} = (W_m^\top \delta_m) \odot s'(W_{m-1} h_{m-2})$$

$$f(x, W) = s(W_m h_{m-1})$$

$$\nabla_{W_m} J = \delta_m h_{m-1}^\top$$

$$\delta_m = \ell'(f(x, W))$$

$$J(W) = \ell(f(x, W))$$

Matrix Gradient Recipe

$$h_1 = s(W_1 x)$$

$$h_i = s(W_i h_{i-1})$$

$$f(x, W) = s(W_m h_{m-1})$$

$$J(W) = \ell(f(x, W))$$

$$\delta_i = (W_{i+1}^\top \delta_{i+1}) \odot s'(W_i h_{i-1})$$

$$\delta_m = \ell'(f(x, W))$$

$$\nabla_{W_1} J = \delta_1 x^\top$$

$$\nabla_{W_i} J = \delta_i h_{i-1}^\top$$

Feed Forward
Propagation

Back Propagation

Challenges

- Local minima (non-convex)
- Overfitting

Remedies

- Regularization
- Parameter sharing: convolution
- Pre-training: initializing weights smartly
- Training data manipulation, e.g., dropout, noise, transformations
- Huge data sets