Back Propagation

Machine Learning
CSx824/ECEx242
Bert Huang
Virginia Tech

Outline

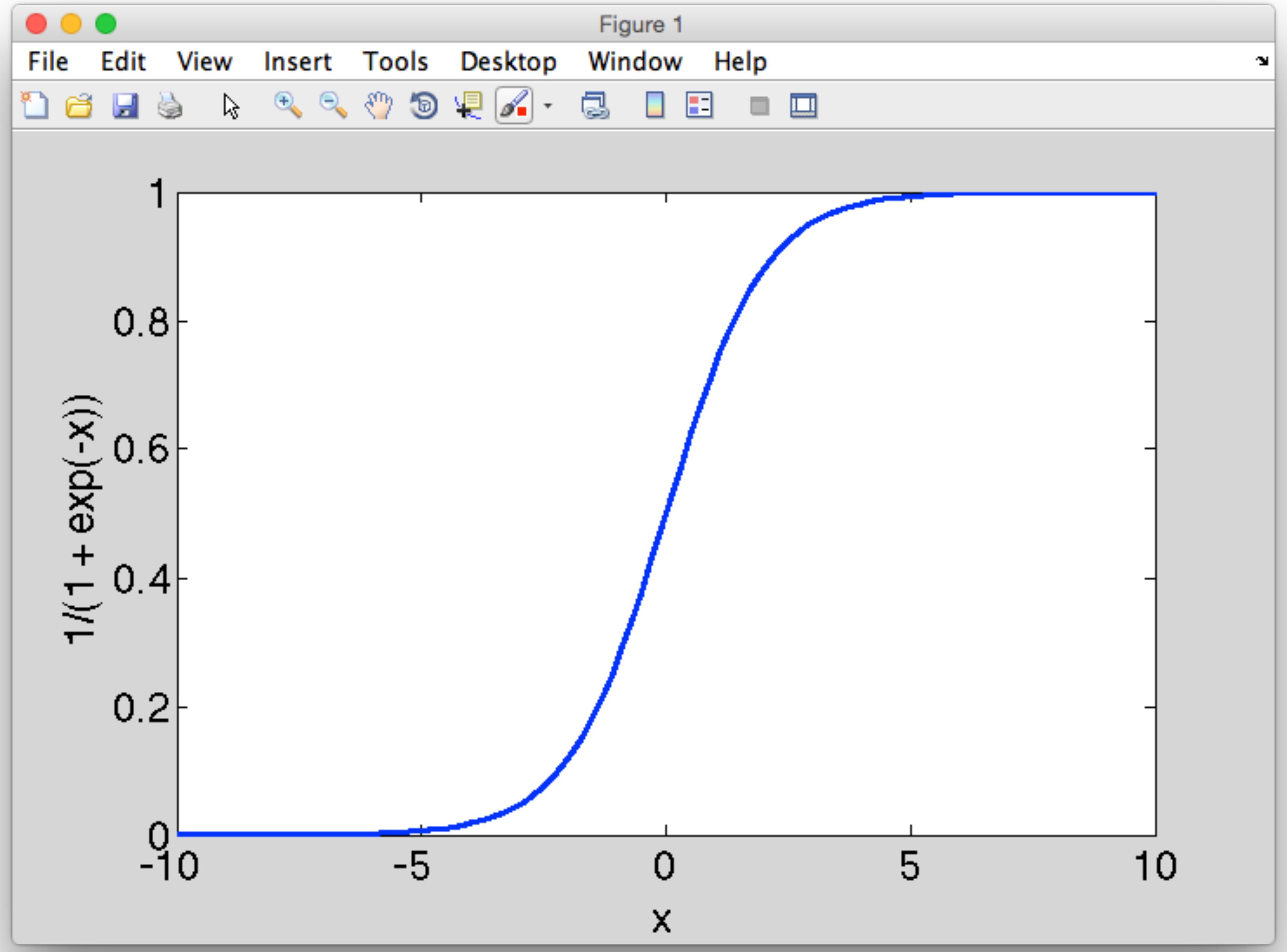
- Logistic regression and perceptron as neural networks
- Likelihood gradient for 2-layered neural network
- General recipe for back propagation

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

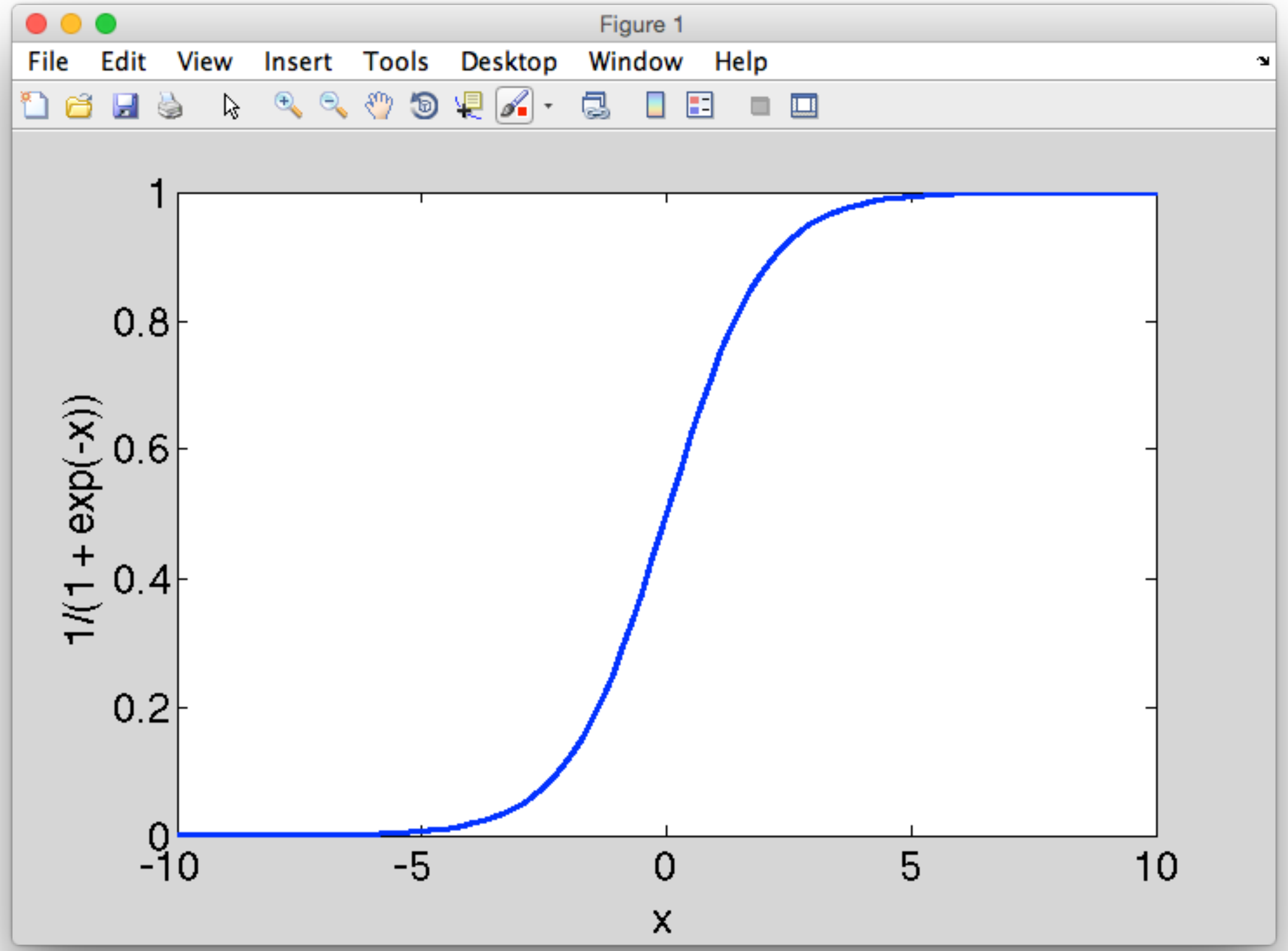
Logistic Squashing Function

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



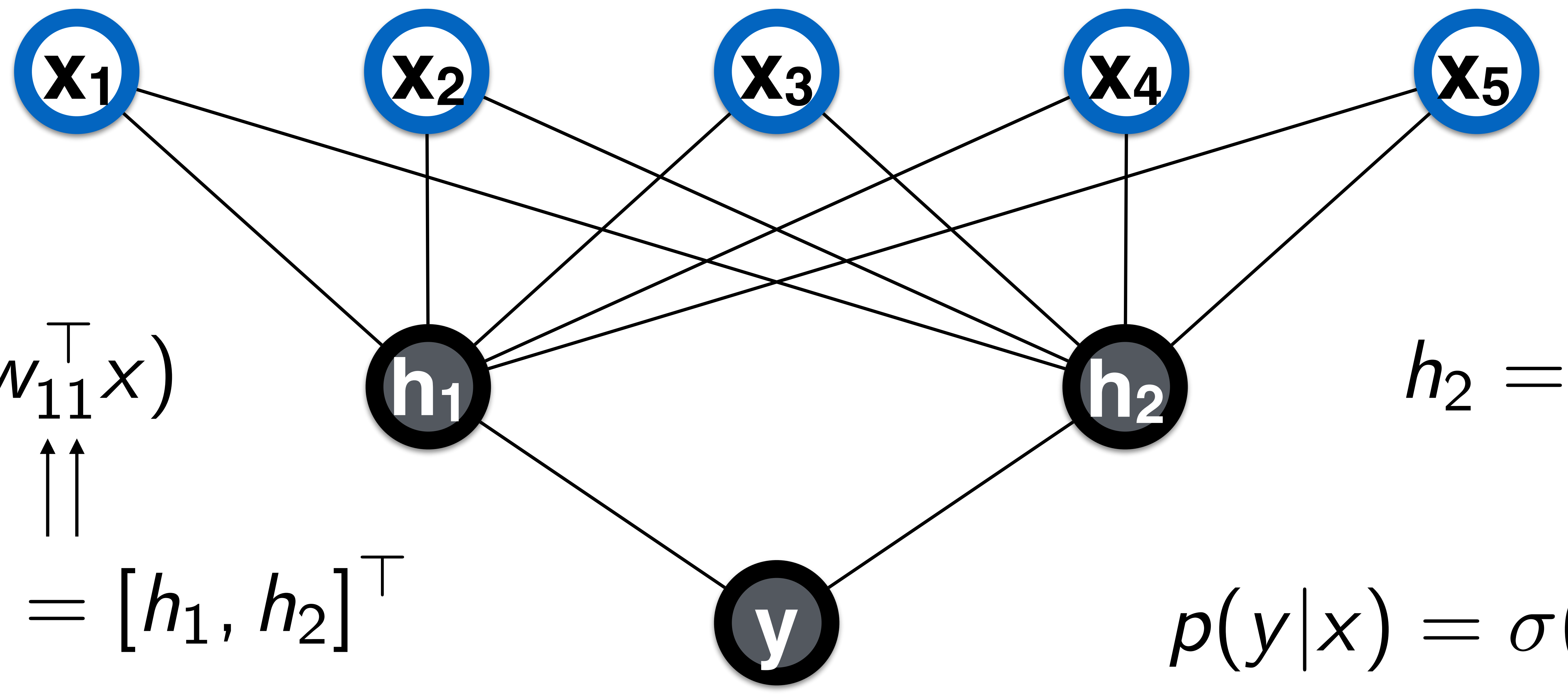
Logistic Squashing Function

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



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

Multi-Layered Perceptron



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

$h_2 = \sigma(w_{12}^\top x)$

$\uparrow\uparrow$

$h = [h_1, h_2]^\top$

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

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

Gradients

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

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

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

$$\nabla_{w_{21}} \mathrm{ll} = \sum_{i=1}^{n} \frac{1}{p(y_i|x_i)} \times \nabla_{w_{21}} p(y_i|x_i)$$

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

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

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

Gradients

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

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

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

$$\nabla_{w_{11}} \text{ll} = \sum_{i=1}^{n} \frac{1}{p(y_i|x_i)} \times \nabla_{w_{11}} p(y_i|x_i)$$

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

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

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

Gradients

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

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

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

$$\nabla_{w_{11}} \mathrm{ll} = \sum_{i=1}^{n} \frac{1}{p(y_i|x_i)} \times \nabla_{w_{11}} p(y_i|x_i)$$

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

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

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

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

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

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

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

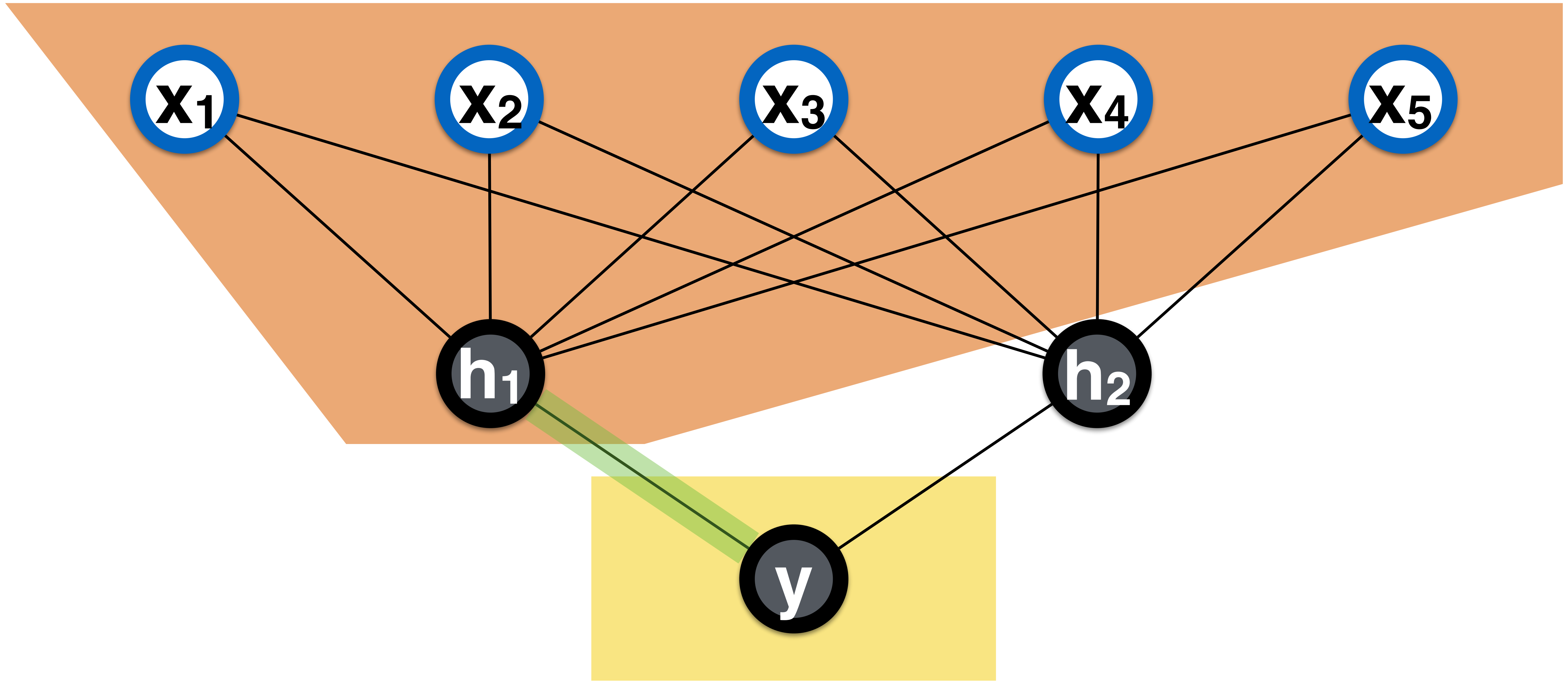
$\log \sigma(w_{21}^\top h)$ $\quad w_{21}^\top h$ $\quad h_1 = \sigma(w_{11}^\top x)$

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

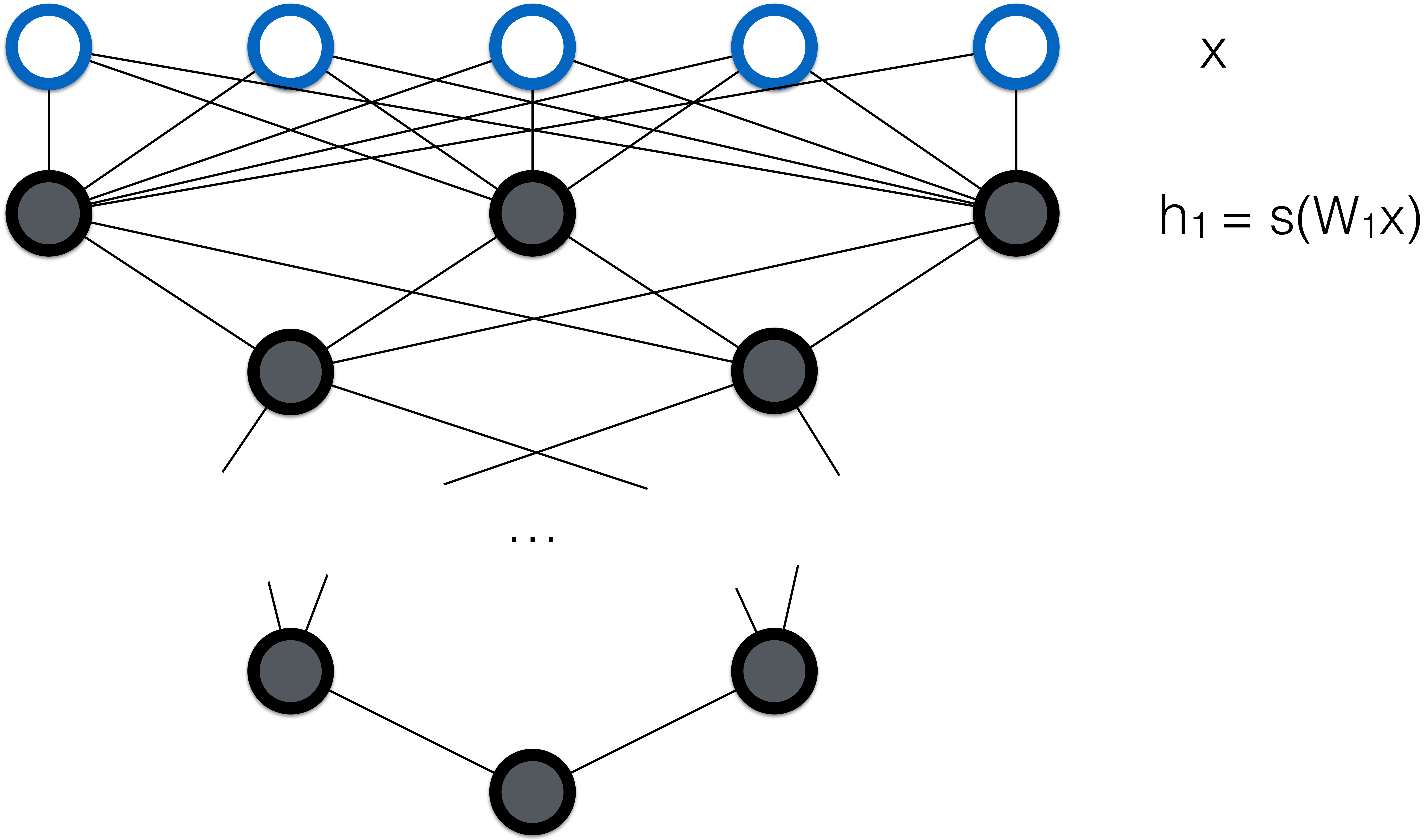
raw error

gradient of blamed error

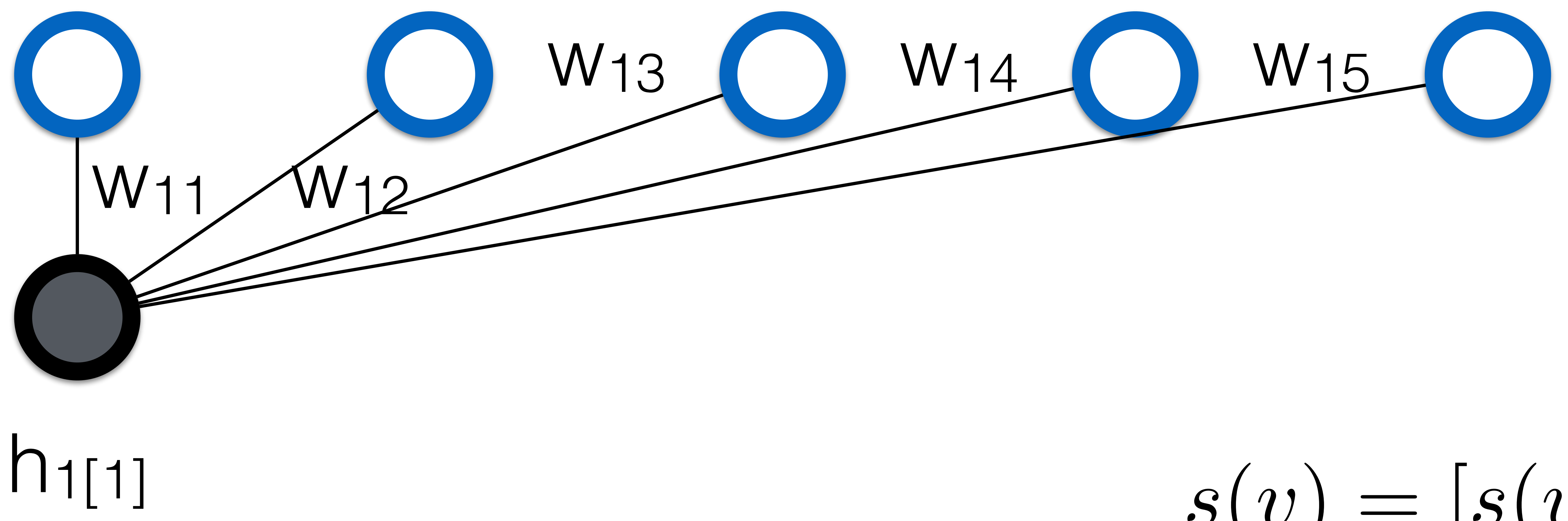
blame for error



Matrix Form



Matrix Form



x

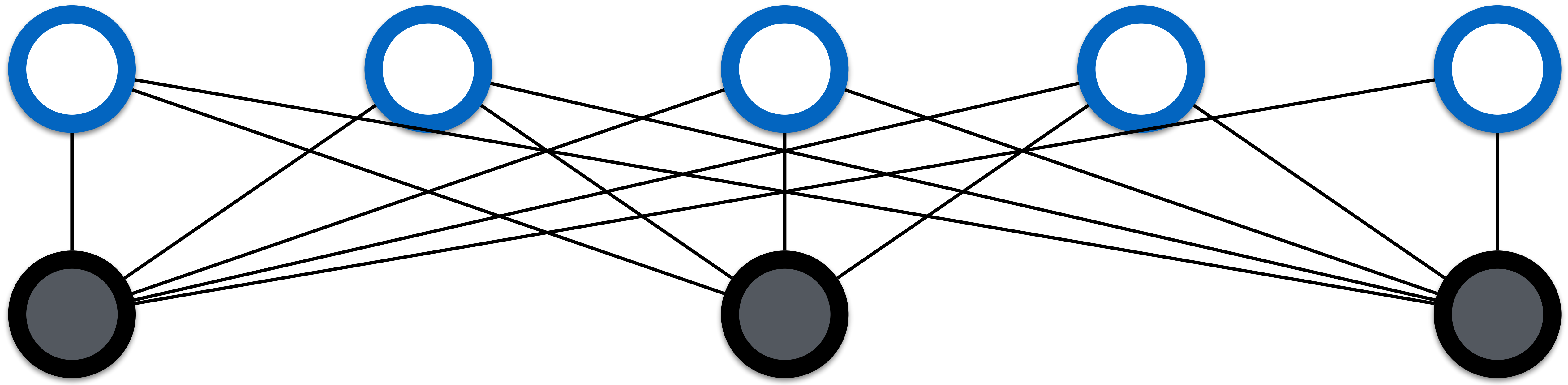
$h_1 = s(W_1 x)$

$$s(v) = [s(v_1), s(v_2), s(v_3), ...]^\top$$

$W_1 =$

w_{11}	w_{12}	w_{13}	w_{14}	w_{15}

Matrix Form



x

$h_1 = s(W_1 x)$

$$s(v) = [s(v_1), s(v_2), s(v_3), \ldots]^\top$$

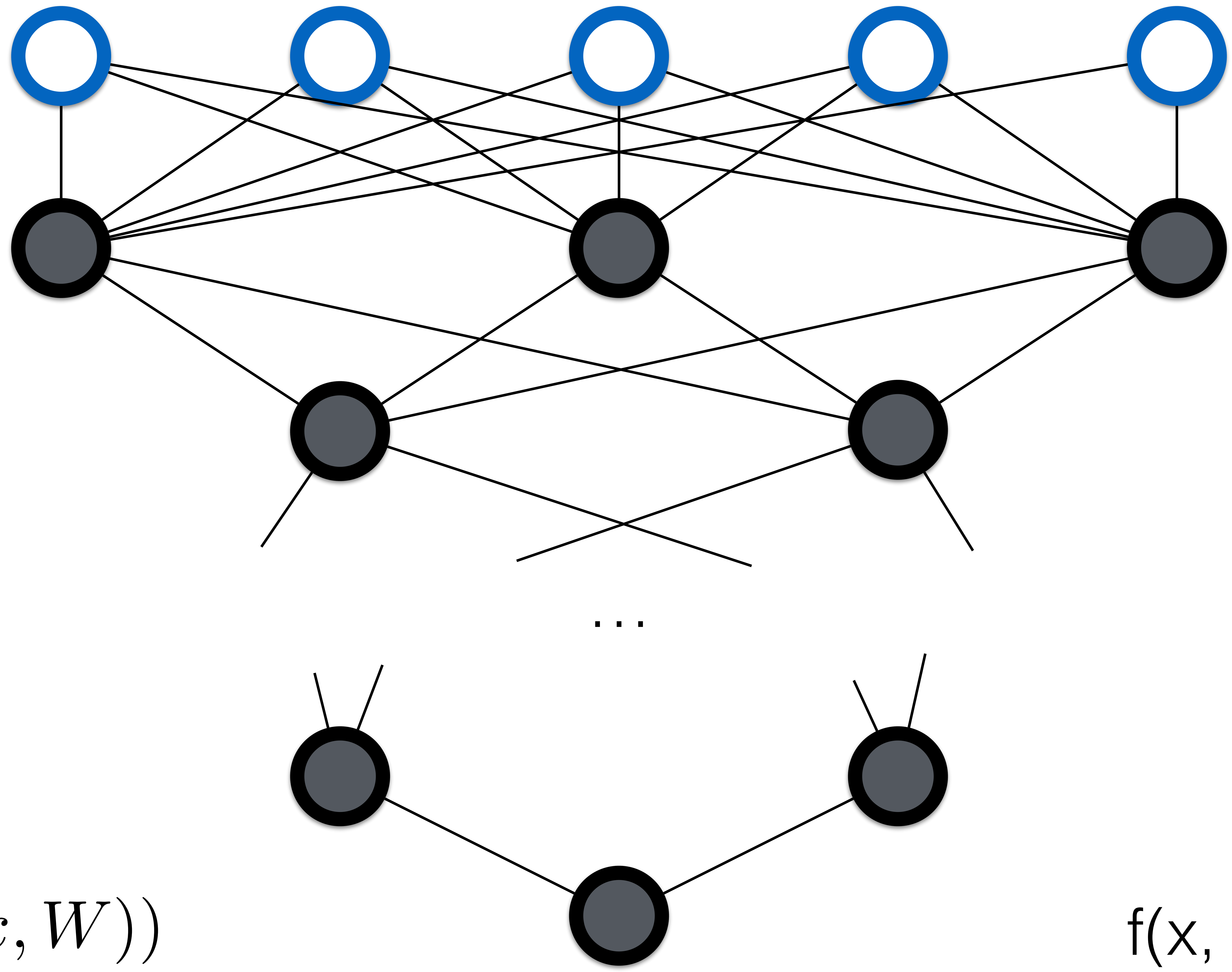
$W_1 =$

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



x

$h_1 = s(W_1 x)$

$h_2 = s(W_2 h_1)$

…

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

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

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

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

$h_1 = s(W_1 x)$

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

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

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

Feed Forward Propagation

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

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

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

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

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