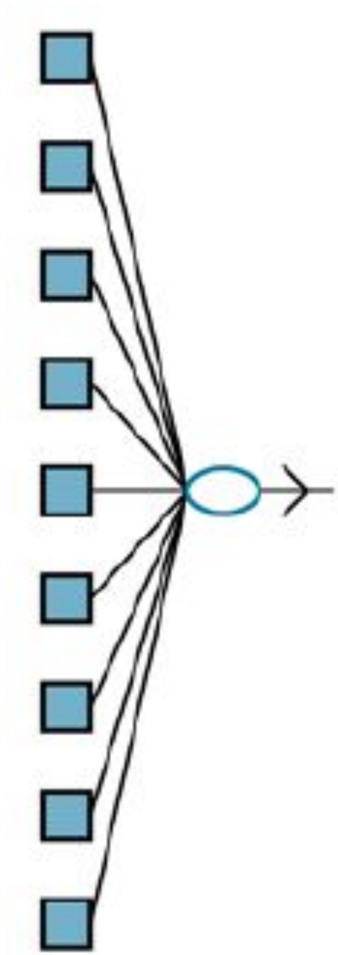
CSE 471: MACHINE LEARNING

Deep Learning

Outline

- Feedforward neural network
  - Multi layer perceptron (MLP)
- Convolutional neural network
- Recurrent neural network
-

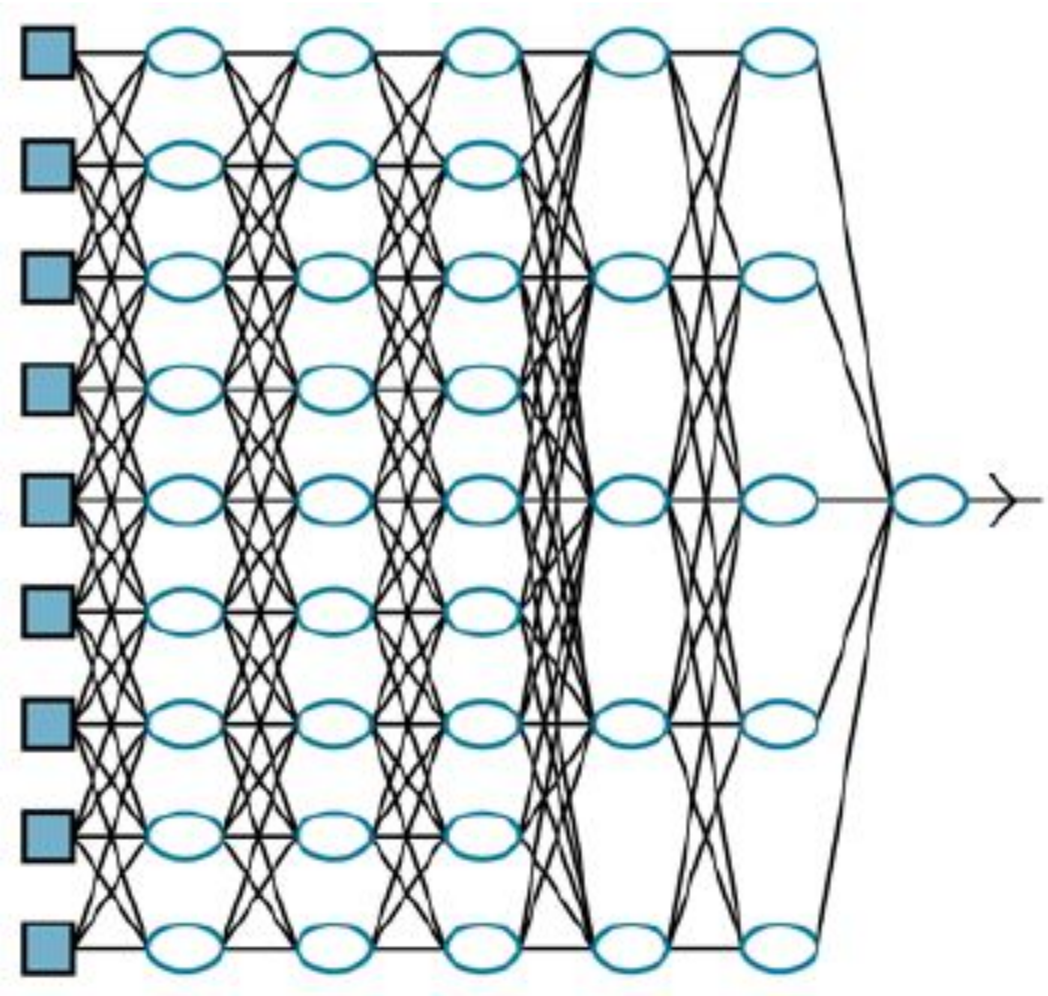
Perceptron



Multi layer perceptron

Deep learning network

A deep learning network has longer computation paths, allowing each variable to interact with all the others.



Feedforward Network

$$a_j = g_j\left(\sum_i w_{i,j} a_i\right) \equiv g_j(in_j)$$

- g_j – Nonlinear activation function
- There is an intercept *b. Think of it as weight given to a fixed +1 activation node.*

$$a_j = g_j(\mathbf{w}^\top \mathbf{x})$$

Nonlinearity (Sigmoid)

$$\sigma(x) = 1/(1 + e^{-x})$$

Nonlinearity

Sigmoid

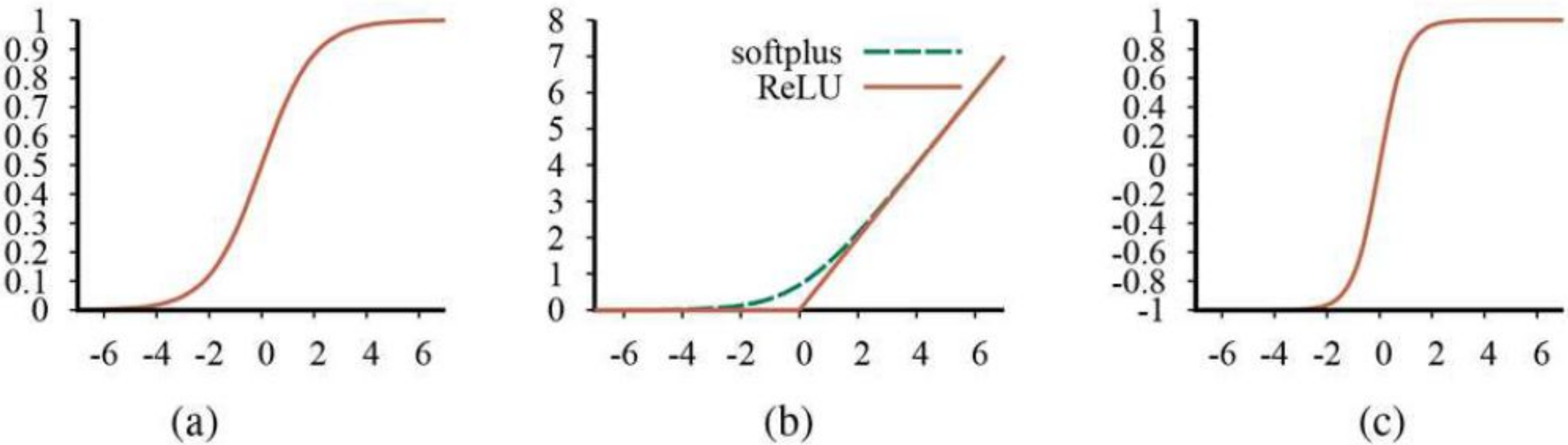
$$\sigma(x) = 1/(1 + e^{-x})$$

ReLU(x) = max(*0*, *x*)

softplus(x) = log(1 + e^{x})

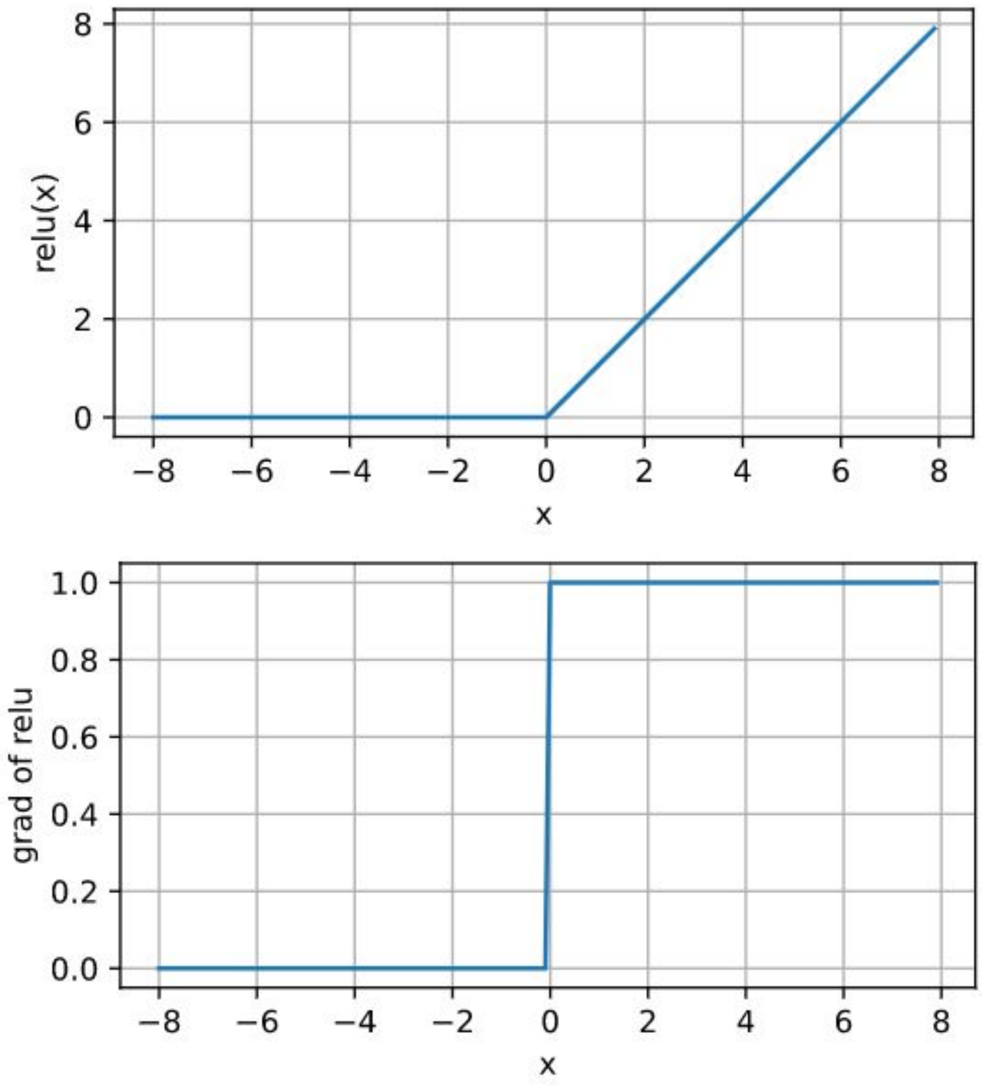
$$\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1} = 2\sigma(2x) - 1$$

Nonlinearity

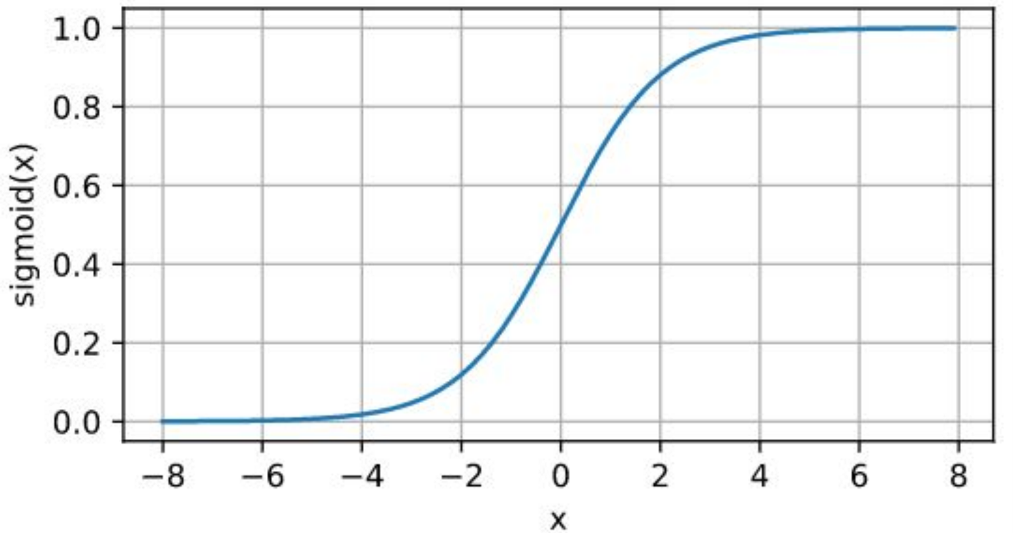


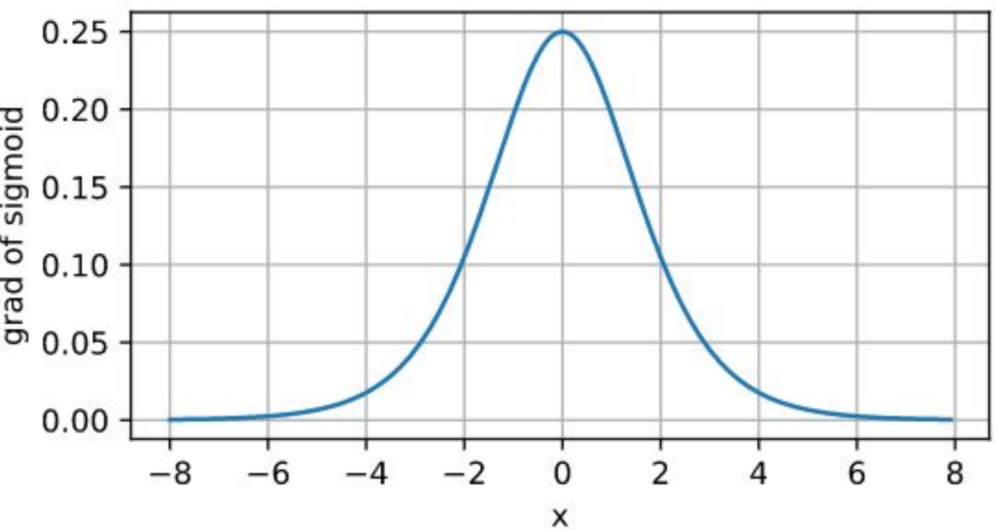
Activation functions commonly used in deep learning systems: (a) the logistic or sigmoid function; (b) the ReLU function and the softplus function; (c) the tanh function.

ReLU



Sigmoid

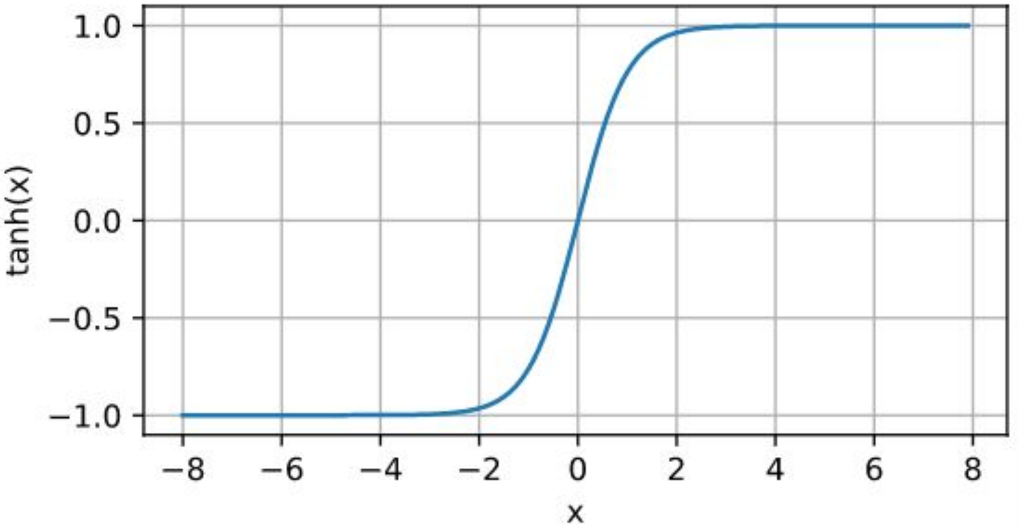




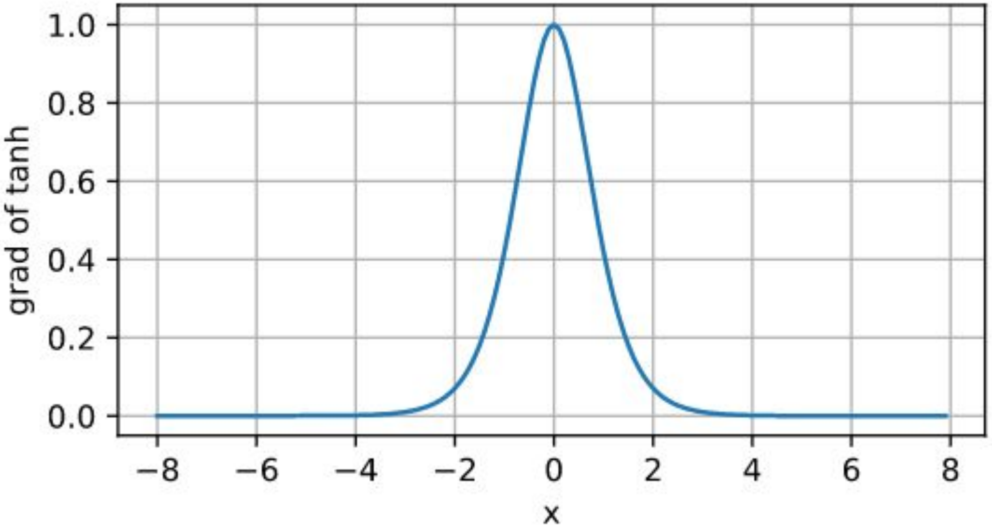
$$\frac{d}{dx}\text{sigmoid}(x) = \frac{\exp(-x)}{(1+\exp(-x))^2}$$

$$= \text{sigmoid}(x)\,(1 - \text{sigmoid}(x))$$

Tanh

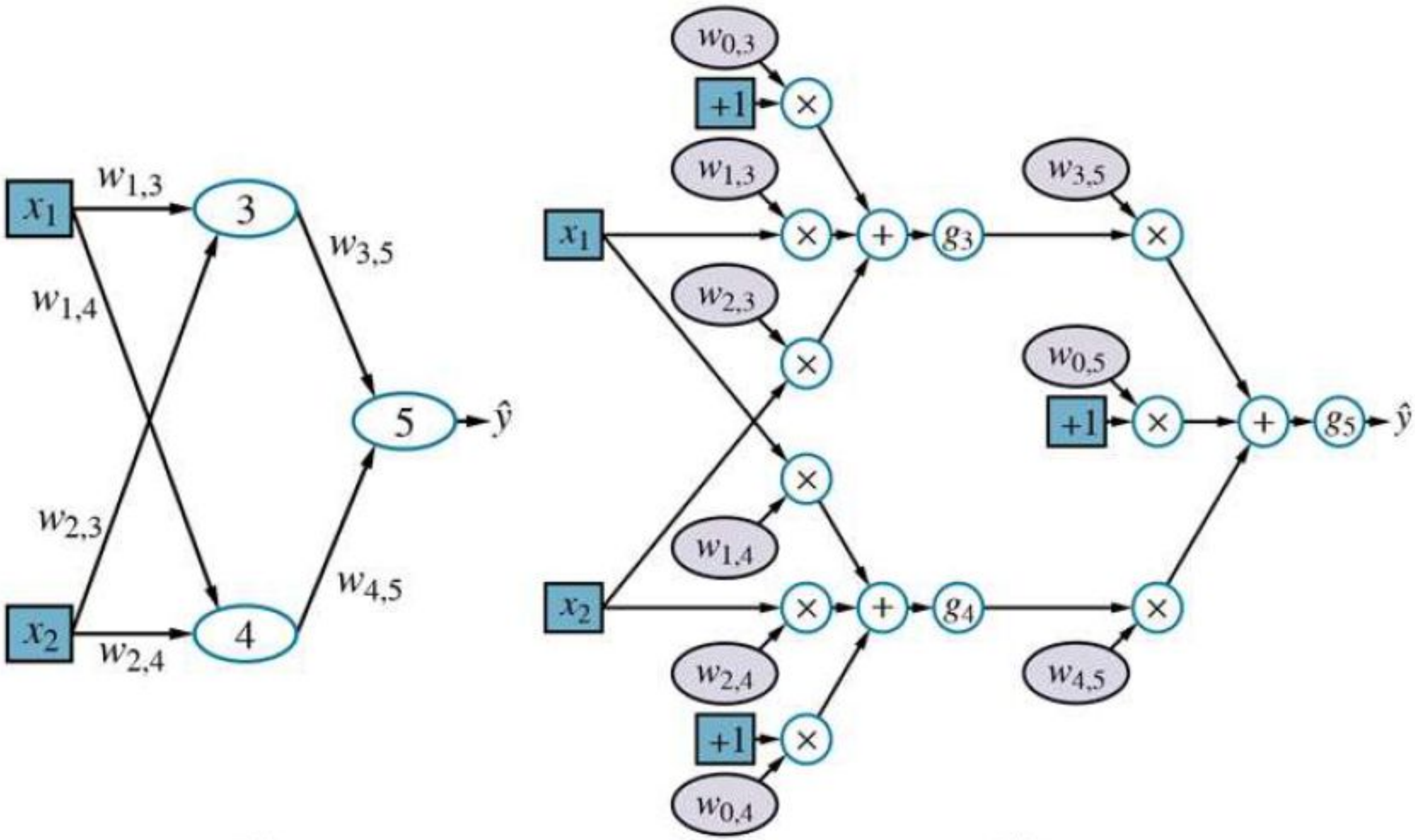


$$\tanh(x) = \frac{1 - \exp(-2x)}{1 + \exp(-2x)}.$$



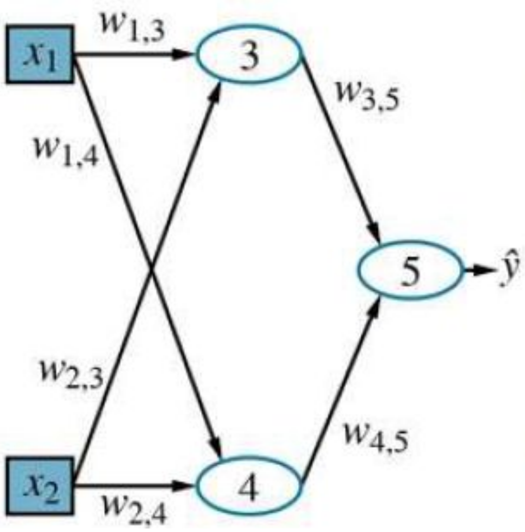
$$\frac{d}{dx}\tanh(x) = 1 - \tanh^2(x)$$

Computation Graph



(a) (b)

Forward computation



$$\begin{aligned}\hat{y} &= g_5(in_5) = g_5(w_{0,5} + w_{3,5}a_3 + w_{4,5}a_4) \\ &= g_5(w_{0,5} + w_{3,5}g_3(in_3) + w_{4,5}g_4(in_4)) \\ &= g_5(w_{0,5} + w_{3,5}g_3(w_{0,3} + w_{1,3}x_1 + w_{2,3}x_2) \\ &\qquad + w_{4,5}g_4(w_{0,4} + w_{1,4}x_1 + w_{2,4}x_2))\end{aligned}$$

$$\boxed{h_{\mathbf{w}}(\mathbf{x}) = \mathbf{g}^{(2)}(\mathbf{W}^{(2)}\mathbf{g}^{(1)}(\mathbf{W}^{(1)}\mathbf{x}))}$$

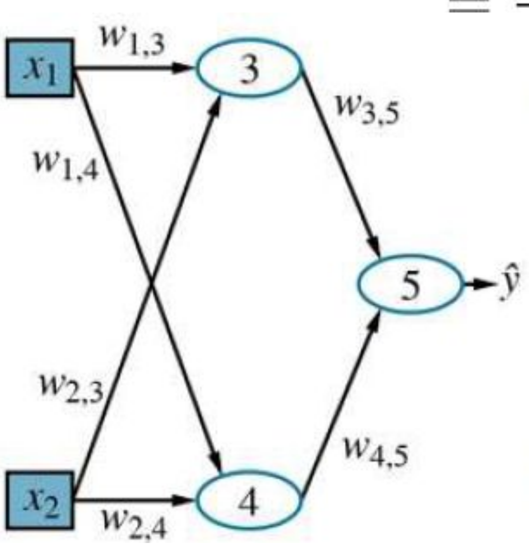
Gradients

$$\frac{\partial}{\partial w_{3,5}} Loss(h_{\mathbf{w}}) = \frac{\partial}{\partial w_{3,5}}(y-\hat{y})^2 = -2(y-\hat{y})\frac{\partial \hat{y}}{\partial w_{3,5}}$$

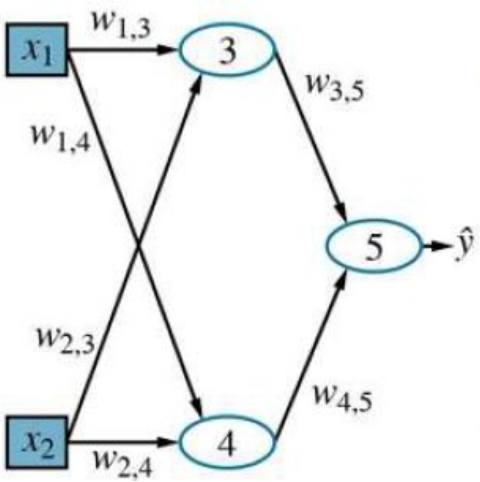
$$= -2(y-\hat{y})\frac{\partial}{\partial w_{3,5}} g_5(in_5) = -2(y-\hat{y})g_5'(in_5)\frac{\partial}{\partial w_{3,5}} in_5$$

$$= -2(y-\hat{y})g_5'(in_5)\frac{\partial}{\partial w_{3,5}}\left(w_{0,5} + w_{3,5}a_3 + w_{4,5}a_4\right)$$

$$= -2(y-\hat{y})g_5'(in_5)a_3 .$$



Gradients



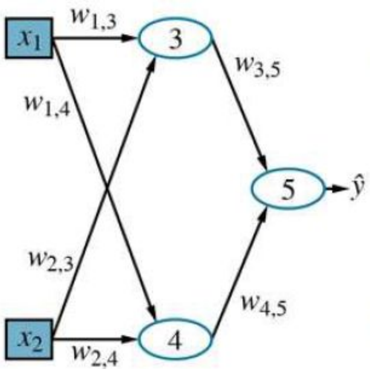
$$\frac{\partial}{\partial w_{3,5}} Loss(h_{\mathbf{w}}) = -2(y - \hat{y})g_5'(in_5)a_3$$

If we define $\Delta_5 = 2(\hat{y} - y)g_5'(in_5)$ as a sort of "perceived error" at the point where unit 5 receives its input, then the gradient with respect to $w_{3,5}$ is just $\Delta_5 a_3$. This makes perfect sense: if Δ_5 is positive, that means $\hat{y}$ is too big (recall that g' is always nonnegative); if a_3 is also positive, then increasing $w_{3,5}$ will only make things worse, whereas if a_3 is negative, then increasing $w_{3,5}$ will reduce the error. The magnitude of a_3 also matters: if a_3 is small for this training example, then $w_{3,5}$ didn't play a major role in producing the error and doesn't need to be changed much.

Gradients

$$\frac{\partial}{\partial w_{1,3}} Loss(h_{\mathbf{w}}) = -2(y-\hat{y})g_5'(in_5)\frac{\partial}{\partial w_{1,3}}\left(w_{0,5}+w_{3,5}a_3+w_{4,5}a_4\right)$$

$$= -2(y-\hat{y})g_5'(in_5)w_{3,5}\frac{\partial}{\partial w_{1,3}}a_3$$



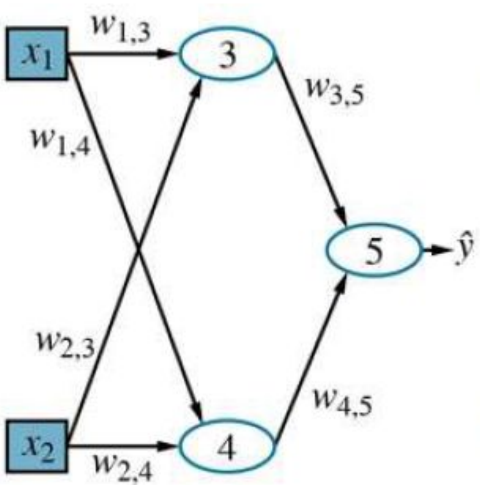
$$= -2(y-\hat{y})g_5'(in_5)w_{3,5}\frac{\partial}{\partial w_{1,3}}g_3(in_3)$$

$$= -2(y-\hat{y})g_5'(in_5)w_{3,5}g_3'(in_3)\frac{\partial}{\partial w_{1,3}}in_3$$

$$= -2(y-\hat{y})g_5'(in_5)w_{3,5}g_3'(in_3)\frac{\partial}{\partial w_{1,3}}\left(w_{0,3}+w_{1,3}x_1+w_{2,3}x_2\right)$$

$$= -2(y-\hat{y})g_5'(in_5)w_{3,5}g_3'(in_3)x_1.$$

Gradients

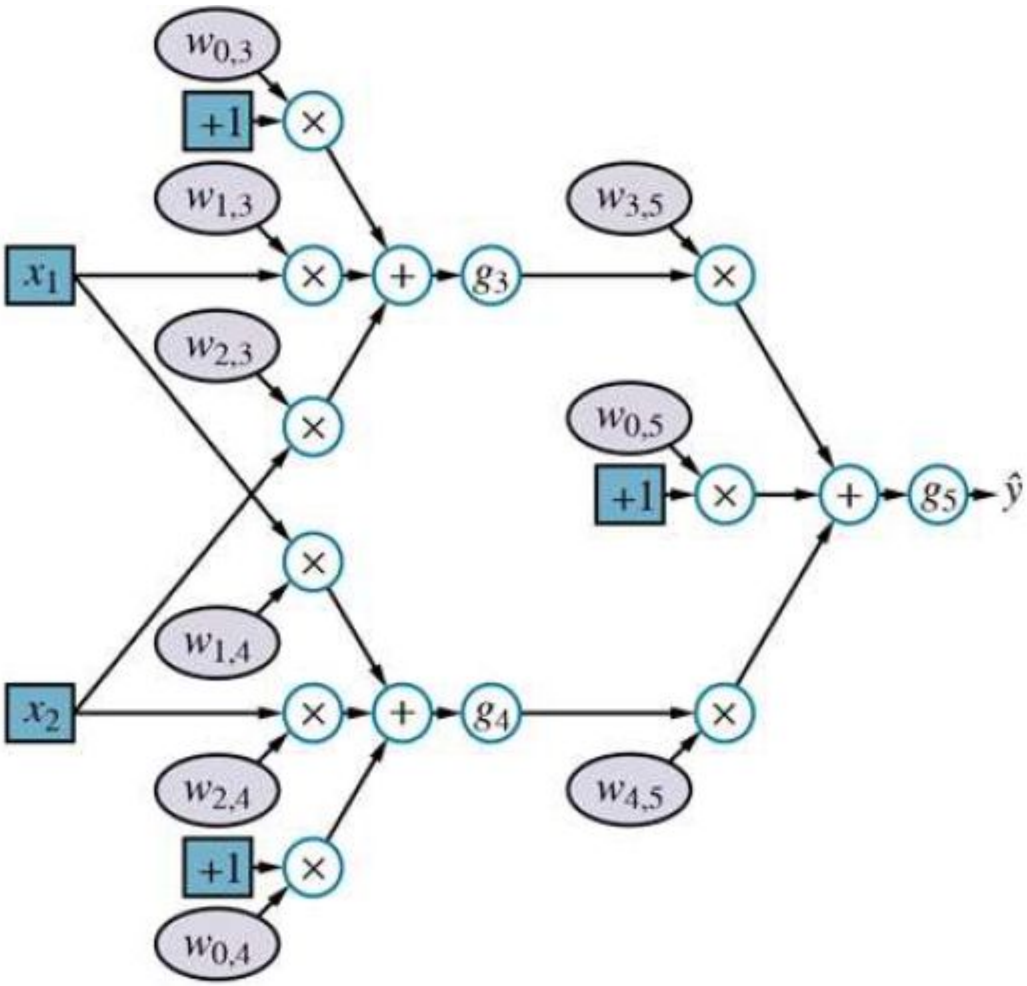


$$\frac{\partial}{\partial w_{1,3}} Loss(h_{\mathbf{w}}) = -2(y - \hat{y})g_5'(in_5)w_{3,5}g_3'(in_3)x_1$$

If we also define $\Delta_3 = \Delta_5 w_{3,5} g_3'(in_3)$, then the gradient for $w_{1,3}$ becomes just $\Delta_3 x_1$. Thus, the perceived error at the input to unit 3 is the perceived error at the input to unit 5, multiplied by information along the path from 5 back to 3. This phenomenon is completely general, and gives rise to the term **back-propagation** for the way that the error at the output is passed back through the network.

Home work

- $w_{0,3} = 1$, $w_{0,4} = -1$, $w_{0,5} = 1$
- $w_{1,3} = 2$, $w_{1,4} = 4$, $w_{2,3} = 3$
- $w_{2,4} = 1$, $w_{3,5} = 2$, $w_{4,5} = 3$
- $x_1 = 5$, $x_2 = -3$
- $y = 1$
- Non-linearity = Sigmoid
- y_hat = ?
- Calculate the gradients
  - w.r.t the w's.



Gradients

- Vanishing gradient
  - Derivatives can be very close to 0
  - Changing the weights may have negligible effects.
- Automatic differentiation
  - Systematic application of chain rule.