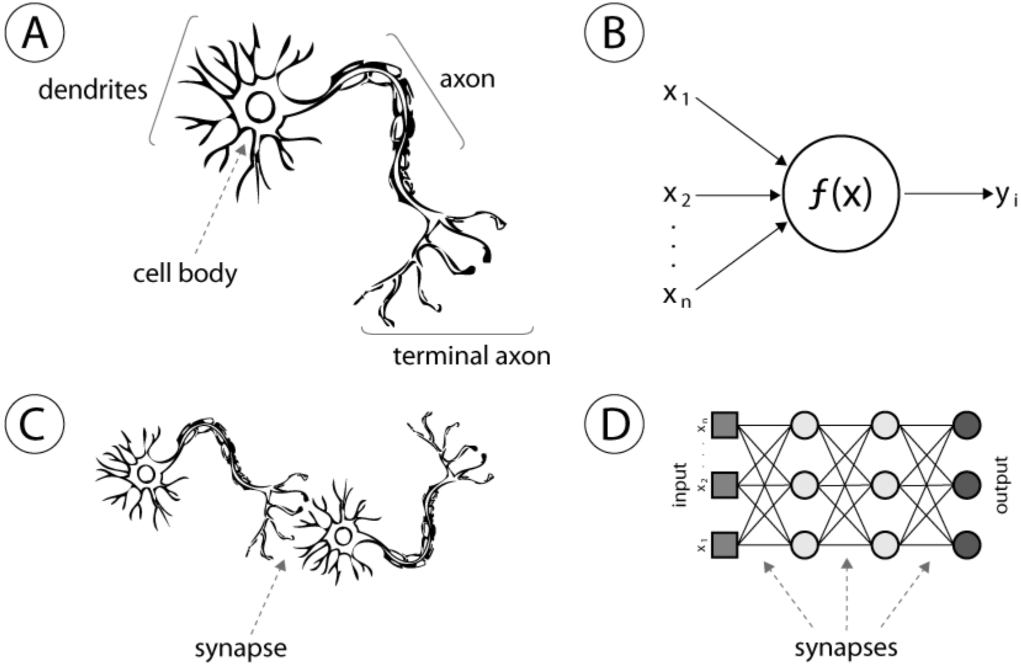
Deep learning

Hung-Hsuan Chen

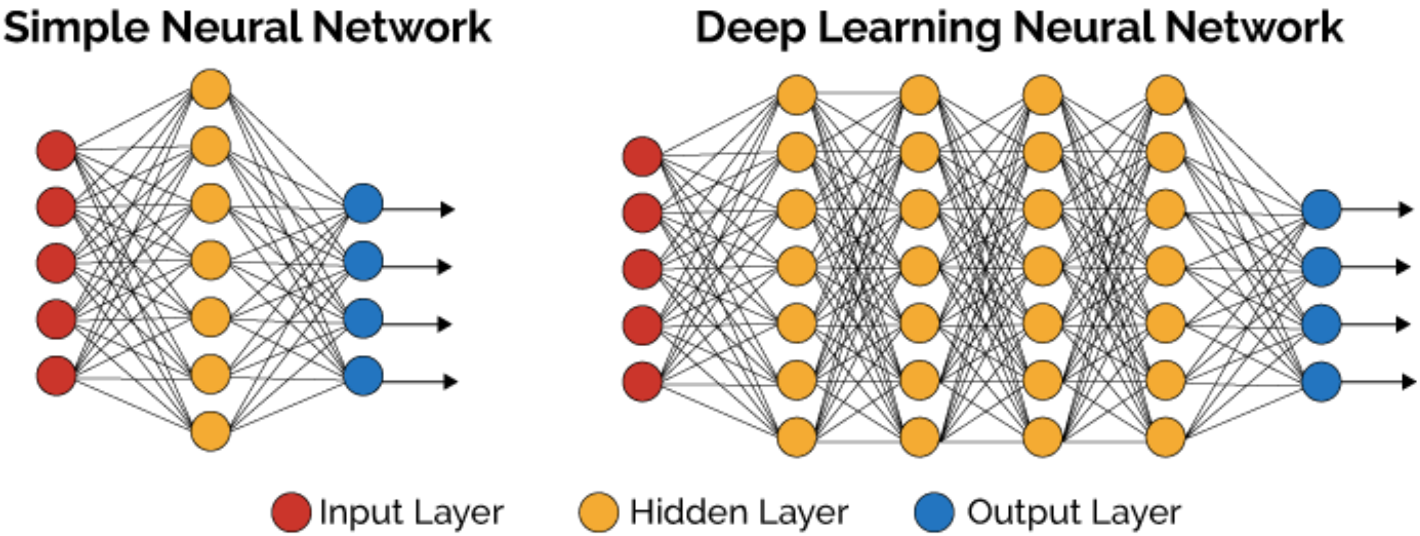
Biological neural network vs artificial neural network (ANN)



A ***synapse*** is a structure that permits a ***neuron*** to pass an electrical or chemical signal to another neuron.

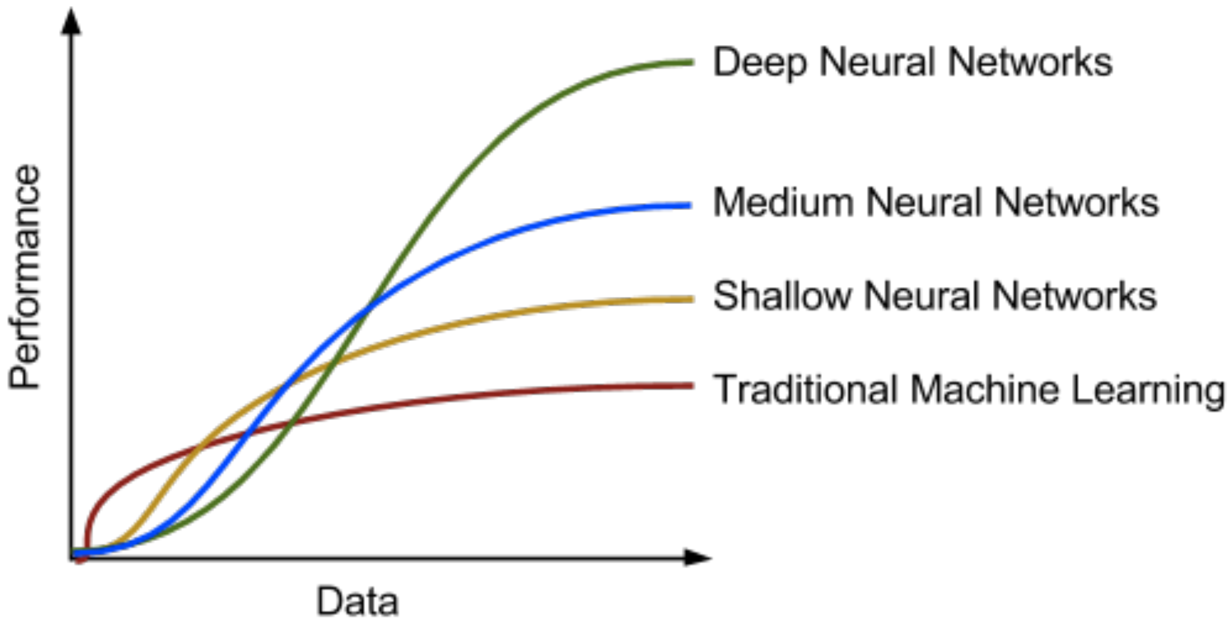
A: human neuron
B: artificial neuron
C: biological synapse
D: ANN synapse

Simple ANN vs deep learning



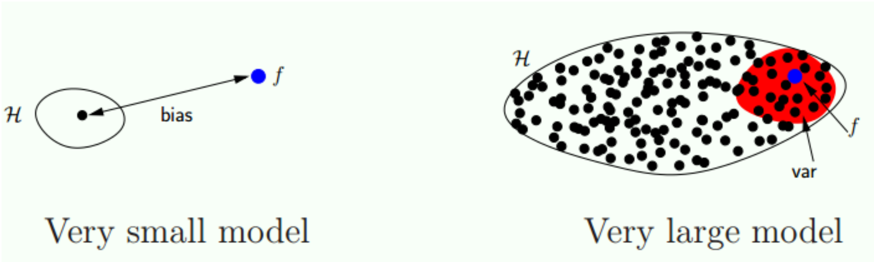
- Simple neural network is also called single layer neural network
  - Single layer ➔ single hidden layer

Why deep learning?



Deep learning and big data

- Small model: tend to have high bias
- Deep learning (large model) and small data: tend to have high variance
- Deep learning (large model) and big data: small bias and small variance
  - Computation cost is an issue



Learning representations

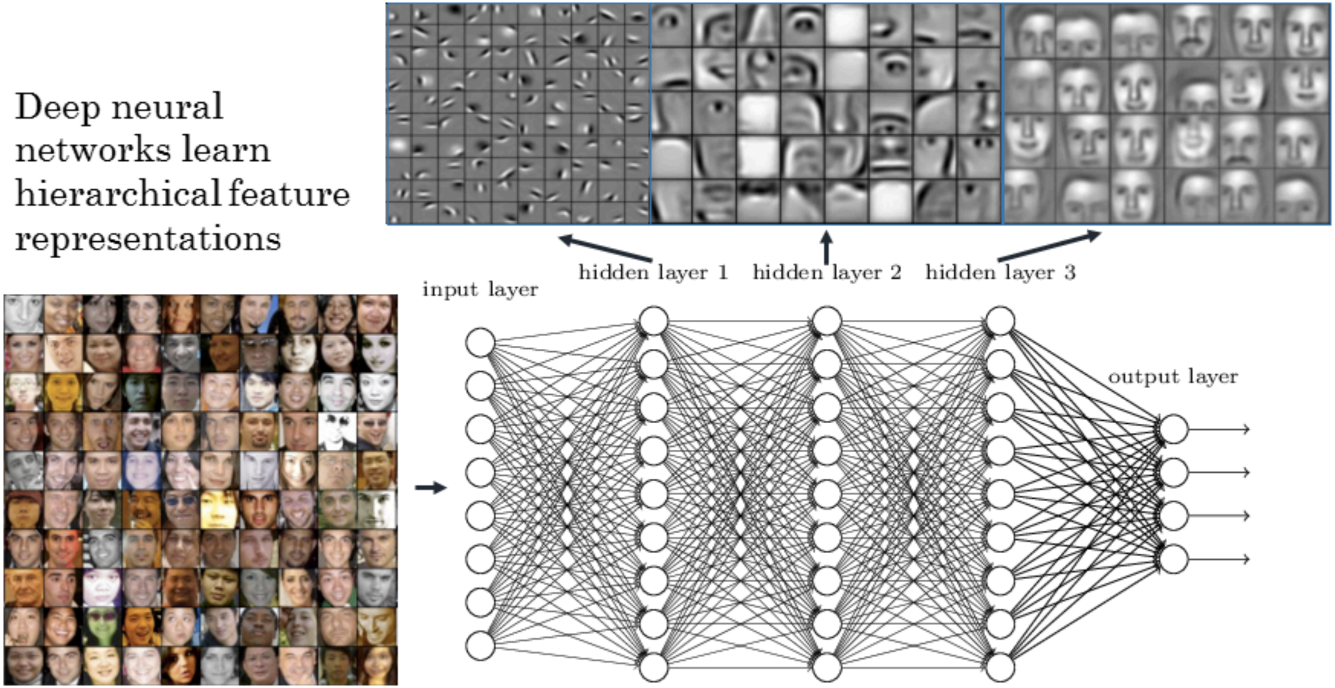
- Deep learning is part of a broader family of machine learning methods based on ***learning data representations***
- E.g., in CV, traditionally handcraft features (e.g., SIFT, HOG, etc.) are used as the features, but in deep learning, "raw pixels" used as the features

Traditional Machine Learning Flow

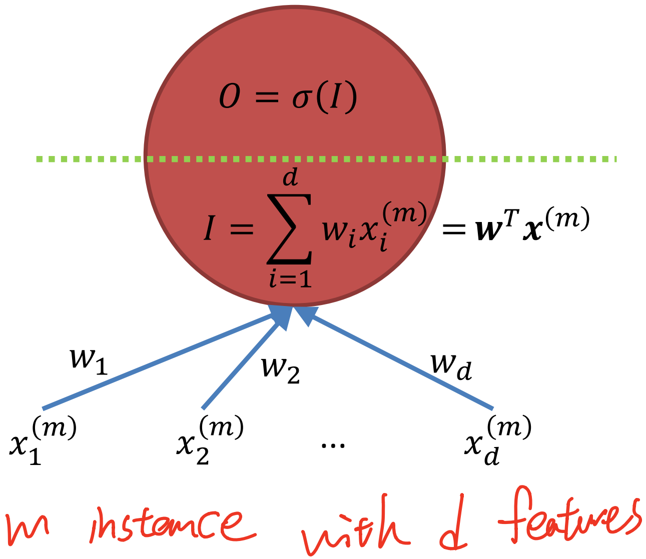
Deep Learning Flow

An example of learning data representation

Deep neural networks learn hierarchical feature representations



Computation of a neuron



- Function: $O = f\left(\boldsymbol{x}^{(m)}\right) = \sigma\left(\boldsymbol{w}^T \boldsymbol{x}^{(m)}\right)$
  - We ignore the bias term
  - σ: a non-linear activation function
    - E.g., $\sigma(z) = \frac{1}{1+\exp(-z)}$
- How to find $\boldsymbol{w}$ such that the loss is minimized?
  - (Stochastic) gradient descent
- Logistic regression

Review: gradient of the logistic function

- Lemma

  If $f(z) = \frac{1}{1+\exp(-z)}$, then $f'(z) = f(z)\big(1 - f(z)\big)$

- Proof

$$f'(z) = -\left(\frac{1}{1 + exp(-z)}\right)^2 (-\exp(-z))$$

$$= \frac{1 + \exp(-z) - 1}{(1 + \exp(-z))^2} = \frac{1}{1 + \exp(-z)}\left(1 - \frac{1}{(1 + \exp(-z))}\right)$$

$$= f(z)\big(1 - f(z)\big)$$

Review: logistic regression model training (by SGD)

- Predicting function (ignore the bias term)

$$\hat{y}^{(m)} = f\left(\boldsymbol{x}^{(m)}\right) = \sigma\left(\boldsymbol{w}^T \boldsymbol{x}^{(m)}\right)$$

- Loss for the m^{th} instance:

$$loss \coloneqq -y^{(m)} \log \hat{y}^{(m)} - \left(1 - y^{(m)}\right) \log\left(1 - \hat{y}^{(m)}\right)$$

- Weight update rule

$$w_i \leftarrow w_i - \eta \frac{\partial loss}{\partial w_i}$$

Review: gradient of logistic regression

$$\frac{\partial loss}{\partial w_i}$$

$$= \frac{\partial -y^{(m)} \log\left(\sigma\left(\boldsymbol{w}^T\boldsymbol{x}^{(m)}\right)\right)}{\partial \log\left(\sigma\left(\boldsymbol{w}^T\boldsymbol{x}^{(m)}\right)\right)} \cdot \frac{\partial \log\left(\sigma\left(\boldsymbol{w}^T\boldsymbol{x}^{(m)}\right)\right)}{\partial\sigma\left(\boldsymbol{w}^T\boldsymbol{x}^{(m)}\right)} \cdot \frac{\partial\sigma\left(\boldsymbol{w}^T\boldsymbol{x}^{(m)}\right)}{\partial\boldsymbol{w}^T\boldsymbol{x}^{(m)}} \cdot \frac{\partial\boldsymbol{w}^T\boldsymbol{x}^{(m)}}{\cancel{\partial i}\ \partial w_i}$$

$$+ \frac{\partial\left(1-y^{(m)}\right) \log\left(1-\sigma\left(\boldsymbol{w}^T\boldsymbol{x}^{(m)}\right)\right)}{\partial \log\left(1-\sigma\left(\boldsymbol{w}^T\boldsymbol{x}^{(m)}\right)\right)} \cdot \frac{\partial \log\left(1-\sigma\left(\boldsymbol{w}^T\boldsymbol{x}^{(m)}\right)\right)}{\partial\left(1-\sigma\left(\boldsymbol{w}^T\boldsymbol{x}^{(m)}\right)\right)} \cdot \frac{\partial\left(1-\sigma\left(\boldsymbol{w}^T\boldsymbol{x}^{(m)}\right)\right)}{\partial\sigma\left(\boldsymbol{w}^T\boldsymbol{x}^{(m)}\right)}$$

$$\cdot \frac{\partial\sigma\left(\boldsymbol{w}^T\boldsymbol{x}^{(m)}\right)}{\partial\boldsymbol{w}^T\boldsymbol{x}^{(m)}} \cdot \frac{\partial\boldsymbol{w}^T\boldsymbol{x}^{(m)}}{\partial(w_i)}$$

$$= y^{(m)} \cdot \frac{1}{\sigma\left(\boldsymbol{w}^T\boldsymbol{x}^{(m)}\right)} \cdot \sigma\left(\boldsymbol{w}^T\boldsymbol{x}^{(m)}\right) \cdot \left(1-\sigma\left(\boldsymbol{w}^T\boldsymbol{x}^{(m)}\right)\right) \cdot x_i^{(m)}$$

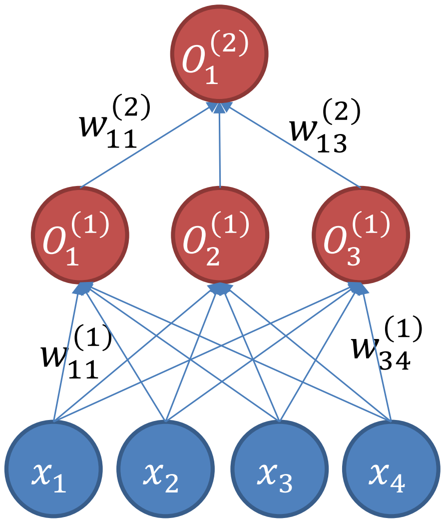
$$+\left(1-y^{(m)}\right) \cdot \frac{1}{1-\sigma\left(\boldsymbol{w}^T\boldsymbol{x}^{(m)}\right)} \cdot (-1) \cdot \sigma\left(\boldsymbol{w}^T\boldsymbol{x}^{(m)}\right) \cdot \left(1-\sigma\left(\boldsymbol{w}^T\boldsymbol{x}^{(m)}\right)\right) \cdot x_i^{(m)}$$

$$= x_i^{(m)}\left(y^{(m)} - \hat{y}^{(m)}\right)$$

Feedforward neural network

Back-prop: adjust weights to minimize error

Forward: predict $f\left(\boldsymbol{x}^{(m)}\right)$ based on current weights



$$\boldsymbol{o}^{(\ell)} = \left[o_1^{(\ell)} \quad \dots \quad o_I^{(\ell)}\right]^T$$

$$\boldsymbol{W}^{(\ell)} = \begin{bmatrix} w_{11}^{(\ell)} & \cdots & w_{1J}^{(\ell)} \\ \vdots & \ddots & \vdots \\ w_{I1}^{(\ell)} & \cdots & w_{IJ}^{(\ell)} \end{bmatrix}$$

$$\boldsymbol{o}^{(2)} = \left[o_1^{(2)}\right]^T$$

$$\boldsymbol{W}^{(2)} = \left[w_{11}^{(2)} \quad w_{12}^{(2)} \quad w_{13}^{(2)}\right]$$

$$\boldsymbol{o}^{(1)} = \left[o_1^{(1)} \quad o_2^{(1)} \quad o_3^{(1)}\right]^T$$

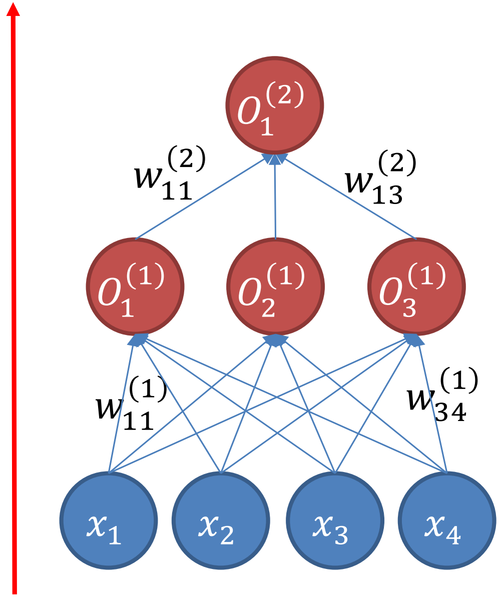
$$\boldsymbol{W}^{(1)} = \begin{bmatrix} w_{11}^{(1)} & \cdots & w_{14}^{(1)} \\ \vdots & \ddots & \vdots \\ w_{31}^{(1)} & \cdots & w_{34}^{(1)} \end{bmatrix}$$

The initial weights ($\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$) are randomly assigned

Forward (predict)

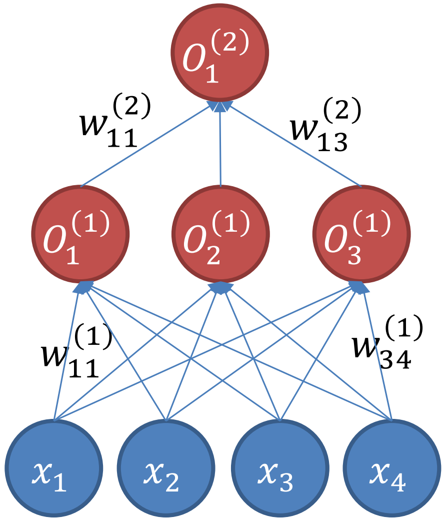
- Forward (predict)
  - $\boldsymbol{I}^{(1)} = \boldsymbol{W}^{(1)}\boldsymbol{x}$
  - $\boldsymbol{O}^{(1)} = \sigma\left(\boldsymbol{I}^{(1)}\right)$
  - $\boldsymbol{I}^{(2)} = \boldsymbol{W}^{(2)}\boldsymbol{O}^{(1)}$
  - $\boldsymbol{O}^{(2)} = \sigma\left(\boldsymbol{I}^{(2)}\right)$
- Initial weights $w_{ij}^{(\ell)}$: randomly assign
  - E.g., $w_{ij}^{(\ell)} \leftarrow N(0, 1)$

Forward: predict $f\left(\boldsymbol{x}^{(m)}\right)$ based on current weights



Forward example

- Input feature and target
  - $\boldsymbol{x} = [1 \quad -1 \quad 3 \quad -2]^T, y = 1$
- Initial weights (random assign)
  - $\boldsymbol{W}^{(2)} = [-1 \quad 1 \quad -1]$
  - $\boldsymbol{W}^{(1)} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & -1 & 0.5 & -1 \\ -1 & -1 & 0 & 0.5 \end{bmatrix}$
- Forward
  - $\boldsymbol{I}^{(1)} = \boldsymbol{W}^{(1)}\boldsymbol{x} = [4 \quad 0.5 \quad 1]^T$
  - $\boldsymbol{O}^{(1)} = \sigma\left(\boldsymbol{I}^{(1)}\right) = [0.98 \quad 0.62 \quad 0.73]^T$
  - $\boldsymbol{I}^{(2)} = \boldsymbol{W}^{(2)}\boldsymbol{O}^{(1)} = [-1.09]$
  - $\boldsymbol{O}^{(2)} = \sigma\left(\boldsymbol{I}^{(2)}\right) = [0.25]$

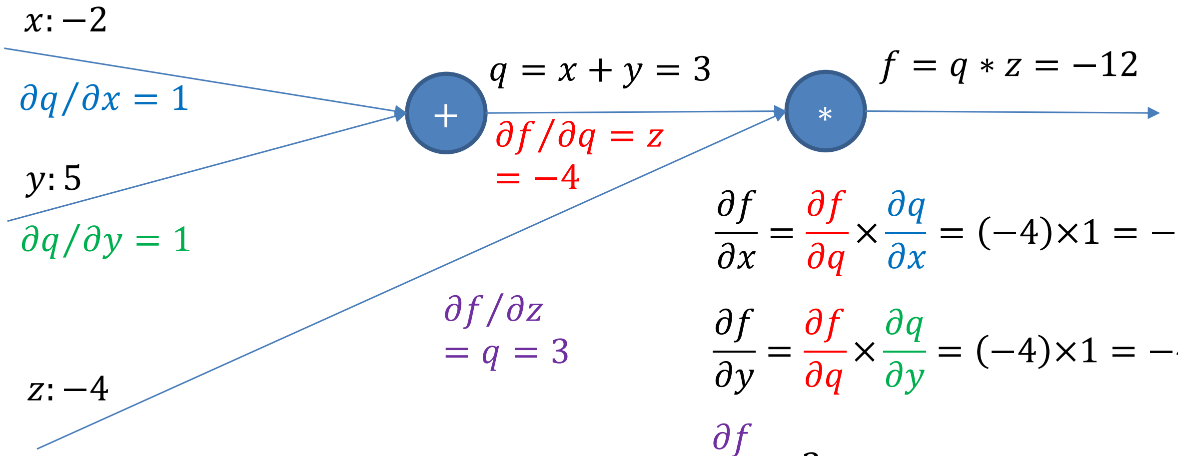


Back-propagation

Slides are taken from Prof. Fei-Fei Li

Computational graph

- $f(x, y, z) = (x + y)z$
  - E.g., $x = -2, y = 5, z = -4$



$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} \times \frac{\partial q}{\partial x} = (-4) \times 1 = -4$$

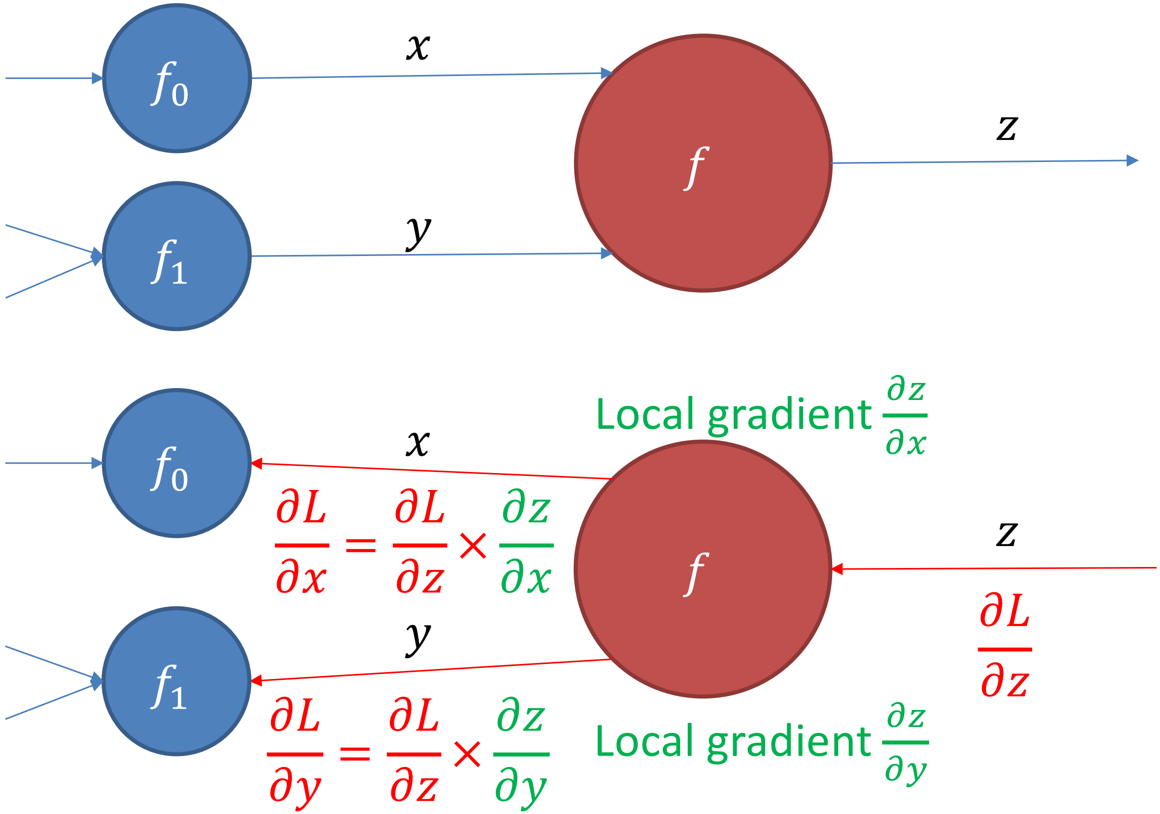
$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial q} \times \frac{\partial q}{\partial y} = (-4) \times 1 = -4$$

$$\frac{\partial f}{\partial z} = 3$$

Local gradient and chain rule

- Given $f(x, y) = z$ and the global loss is L
  1. Compute local gradients $\frac{\partial z}{\partial x}$ and $\frac{\partial z}{\partial y}$
  2. The value of $\frac{\partial L}{\partial z}$ is computed by other components in the computational graph
  3. Based on chain rule, we can get $\frac{\partial L}{\partial x}$ and $\frac{\partial L}{\partial y}$ by $\frac{\partial L}{\partial x} = \frac{\partial L}{\partial z} \times \frac{\partial z}{\partial x}$, and $\frac{\partial L}{\partial y} = \frac{\partial L}{\partial z} \times \frac{\partial z}{\partial y}$

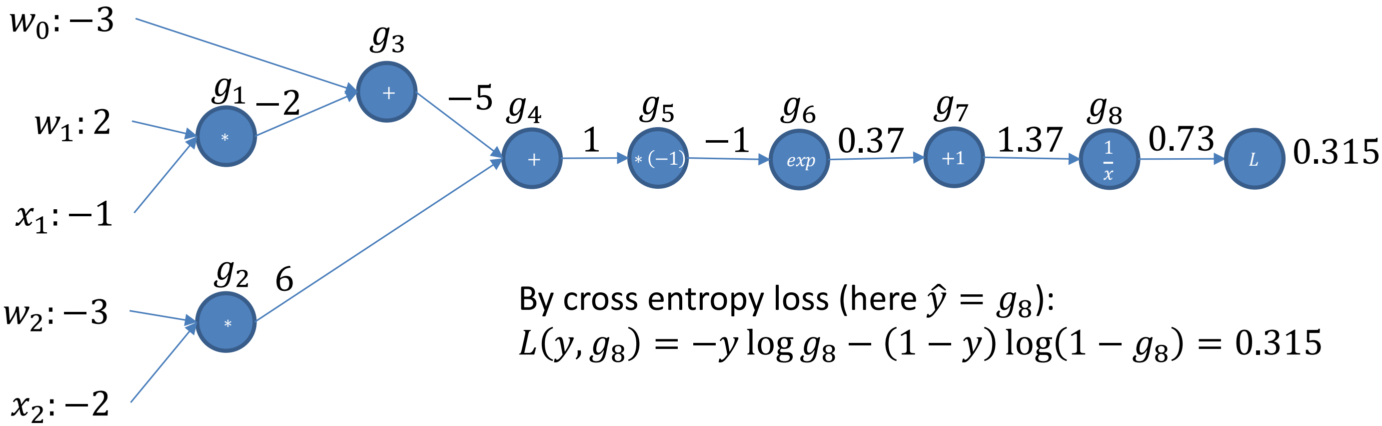
Compute local gradient for every operator and apply chain rule



Example: logistic function

$$f(\boldsymbol{w}, \boldsymbol{x}) = \frac{1}{1 + e^{-(w_0 + w_1 x_1 + w_2 x_2)}}$$

Observing $(x_1, x_2, y) = (-1, -2, 1)$



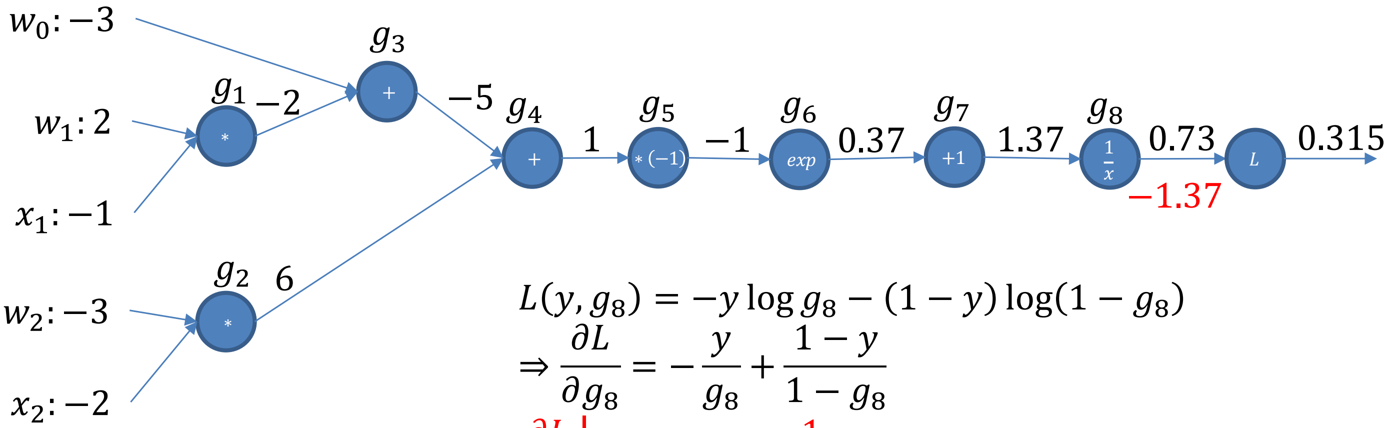
By cross entropy loss (here $\hat{y} = g_8$):

$L(y, g_8) = -y \log g_8 - (1 - y) \log(1 - g_8) = 0.315$

Example: logistic function

$$f(\boldsymbol{w}, \boldsymbol{x}) = \frac{1}{1 + e^{-(w_0 + w_1 x_1 + w_2 x_2)}}$$

Observing $(x_1, x_2, y) = (-1, -2, 1)$



$$L(y, g_8) = -y \log g_8 - (1 - y) \log(1 - g_8)$$

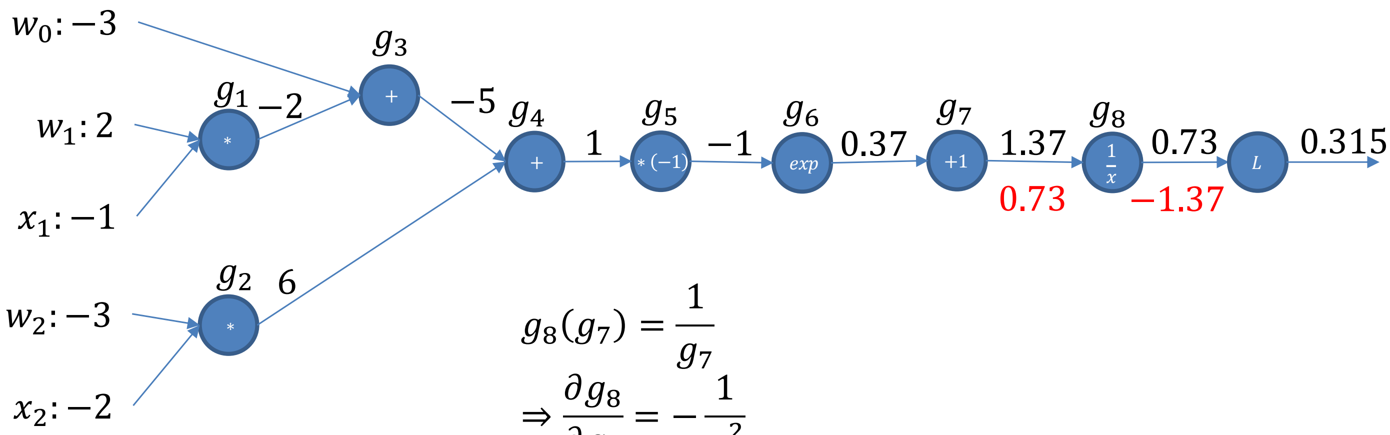
$$\Rightarrow \frac{\partial L}{\partial g_8} = -\frac{y}{g_8} + \frac{1 - y}{1 - g_8}$$

$$\left.\frac{\partial L}{\partial g_8}\right|_{g_8 = 0.73} = -\frac{1}{0.73} = -1.37$$

Example: logistic function

$$f(\boldsymbol{w}, \boldsymbol{x}) = \frac{1}{1 + e^{-(w_0 + w_1 x_1 + w_2 x_2)}}$$

Observing $(x_1, x_2, y) = (-1, -2, 1)$



$$g_8(g_7) = \frac{1}{g_7}$$

$$\Rightarrow \frac{\partial g_8}{\partial g_7} = -\frac{1}{g_7^2}$$

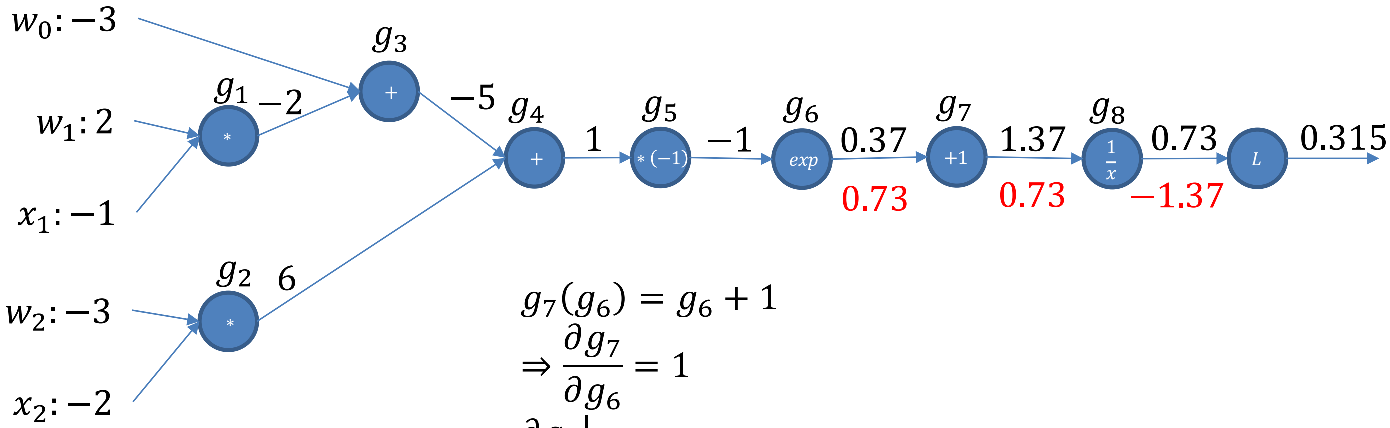
$$\left.\frac{\partial g_8}{\partial g_7}\right|_{g_7 = 1.37} = -\frac{1}{1.37^2} = -0.53$$

$$\frac{\partial L}{\partial g_7} = \frac{\partial L}{\partial g_8} \times \frac{\partial g_8}{\partial g_7} = -1.37 \times -0.53 = 0.73$$

Example: logistic function

$$f(\boldsymbol{w}, \boldsymbol{x}) = \frac{1}{1 + e^{-(w_0 + w_1 x_1 + w_2 x_2)}}$$

Observing $(x_1, x_2, y) = (-1, -2, 1)$



$$g_7(g_6) = g_6 + 1$$

$$\Rightarrow \frac{\partial g_7}{\partial g_6} = 1$$

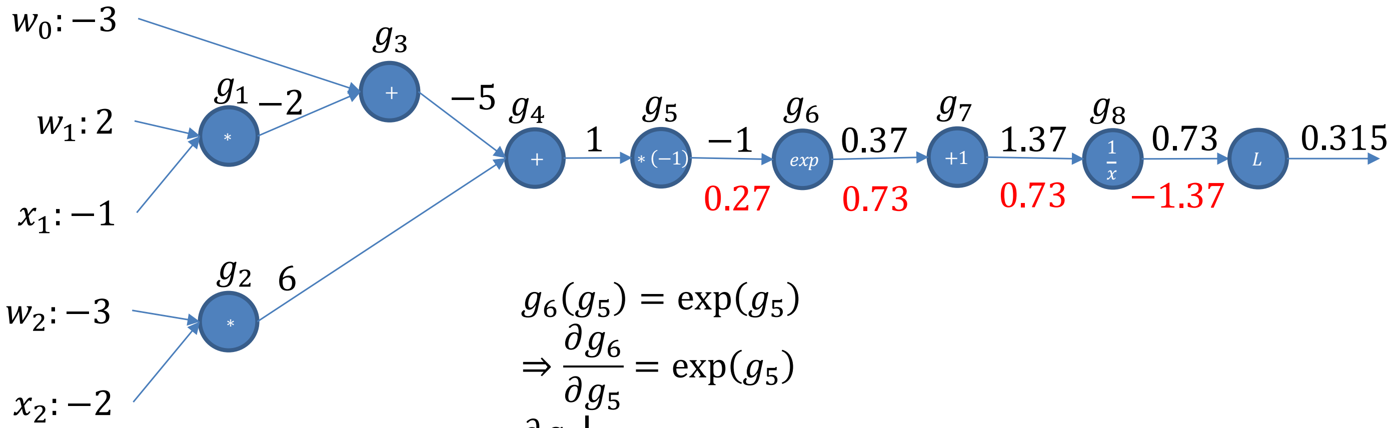
$$\left.\frac{\partial g_7}{\partial g_6}\right|_{g_6 = 0.37} = 1$$

$$\frac{\partial L}{\partial g_6} = \frac{\partial L}{\partial g_7} \times \frac{\partial g_7}{\partial g_6} = 0.73 \times 1 = 0.73$$

Example: logistic function

$$f(\boldsymbol{w}, \boldsymbol{x}) = \frac{1}{1 + e^{-(w_0 + w_1 x_1 + w_2 x_2)}}$$

Observing $(x_1, x_2, y) = (-1, -2, 1)$



$$g_6(g_5) = \exp(g_5)$$

$$\Rightarrow \frac{\partial g_6}{\partial g_5} = \exp(g_5)$$

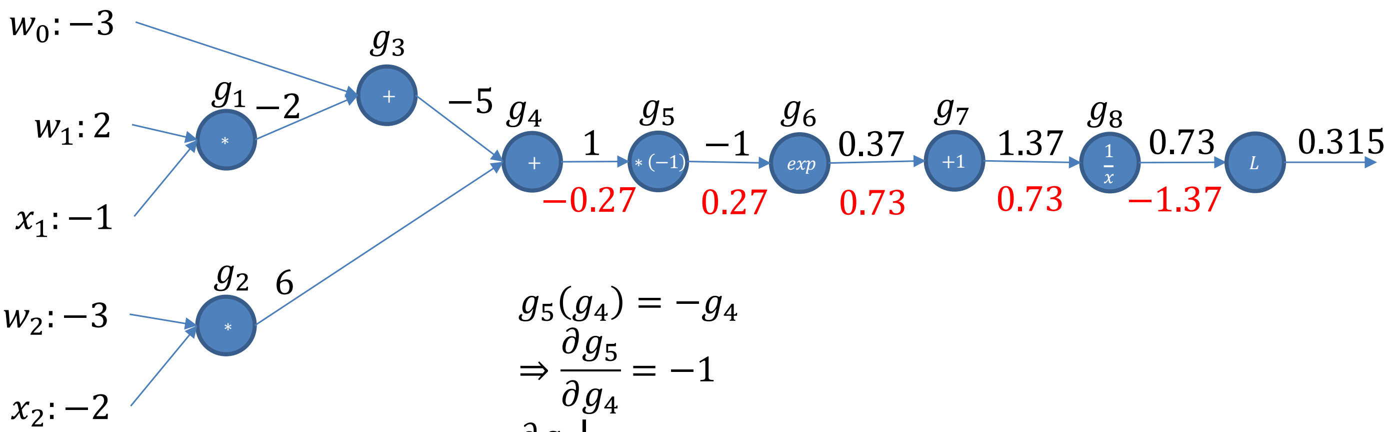
$$\left.\frac{\partial g_6}{\partial g_5}\right|_{g_5=-1} = \exp(-1) = 0.37$$

$$\frac{\partial L}{\partial g_5} = \frac{\partial L}{\partial g_6} \times \frac{\partial g_6}{\partial g_5} = 0.73 \times 0.37 = 0.27$$

Example: logistic function

$$f(\boldsymbol{w}, \boldsymbol{x}) = \frac{1}{1 + e^{-(w_0 + w_1 x_1 + w_2 x_2)}}$$

Observing $(x_1, x_2, y) = (-1, -2, 1)$



$$g_5(g_4) = -g_4$$

$$\Rightarrow \frac{\partial g_5}{\partial g_4} = -1$$

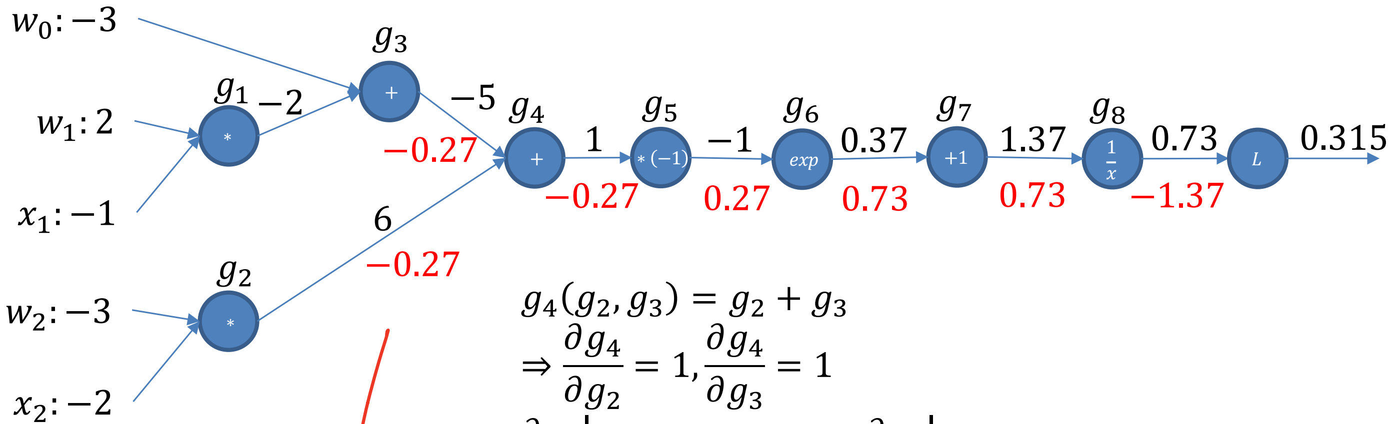
$$\left.\frac{\partial g_5}{\partial g_4}\right|_{g_4=1} = -1$$

$$\frac{\partial L}{\partial g_4} = \frac{\partial L}{\partial g_5} \times \frac{\partial g_5}{\partial g_4} = 0.27 \times (-1) = -0.27$$

Example: logistic function

$$f(\boldsymbol{w}, \boldsymbol{x}) = \frac{1}{1 + e^{-(w_0 + w_1 x_1 + w_2 x_2)}}$$

Observing $(x_1, x_2, y) = (-1, -2, 1)$



$$g_4(g_2, g_3) = g_2 + g_3$$

$$\Rightarrow \frac{\partial g_4}{\partial g_2} = 1, \frac{\partial g_4}{\partial g_3} = 1$$

$$\left.\frac{\partial g_4}{\partial g_2}\right|_{(g_2,g_3)=(6,-5)} = 1, \left.\frac{\partial g_4}{\partial g_3}\right|_{(g_2,g_3)=(6,-5)} = 1$$

有2个input

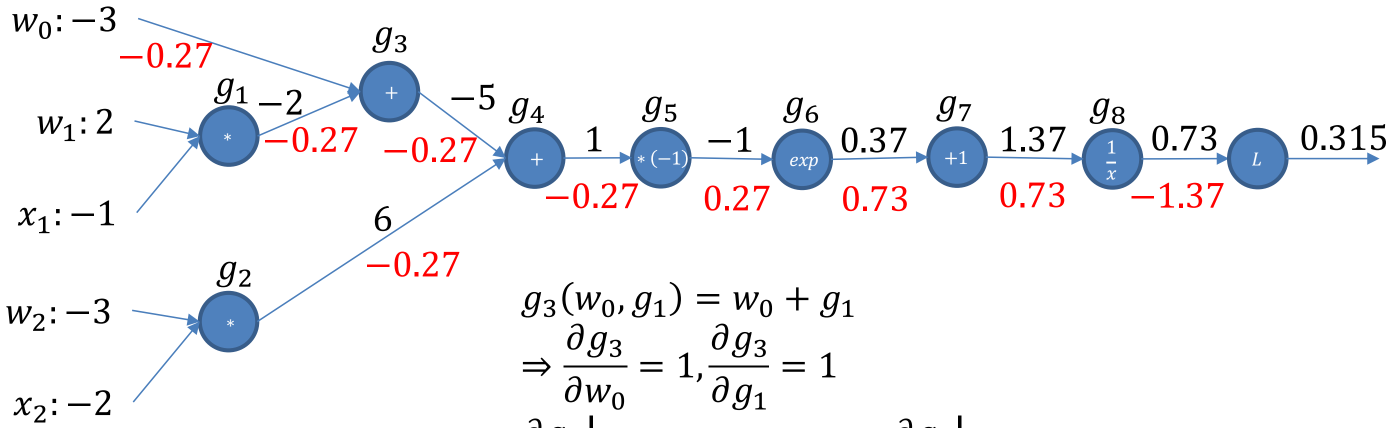
$$\frac{\partial L}{\partial g_2} = \frac{\partial L}{\partial g_4} \times \frac{\partial g_4}{\partial g_2} = -0.27 \times 1 = -0.27$$

$$\frac{\partial L}{\partial g_3} = \frac{\partial L}{\partial g_4} \times \frac{\partial g_4}{\partial g_3} = -0.27 \times 1 = -0.27$$

Example: logistic function

$$f(\boldsymbol{w}, \boldsymbol{x}) = \frac{1}{1 + e^{-(w_0 + w_1 x_1 + w_2 x_2)}}$$

Observing $(x_1, x_2, y) = (-1, -2, 1)$



$$g_3(w_0, g_1) = w_0 + g_1$$

$$\Rightarrow \frac{\partial g_3}{\partial w_0} = 1, \frac{\partial g_3}{\partial g_1} = 1$$

$$\left.\frac{\partial g_3}{\partial w_0}\right|_{(w_0, g_1) = (-3, -2)} = 1, \left.\frac{\partial g_3}{\partial g_1}\right|_{(w_0, g_1) = (-3, -2)} = 1$$

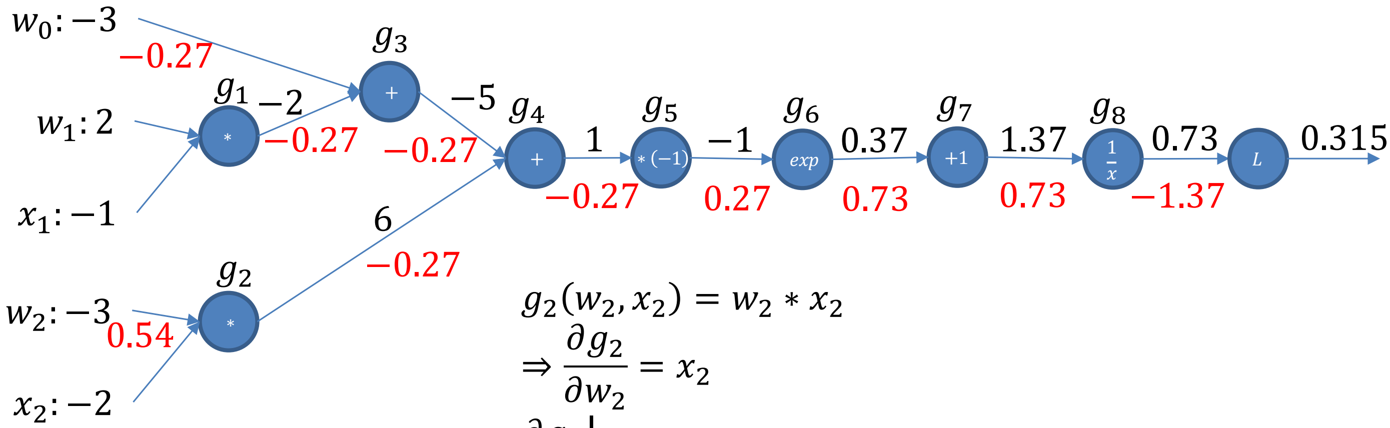
$$\frac{\partial L}{\partial w_0} = \frac{\partial L}{\partial g_3} \times \frac{\partial g_3}{\partial w_0} = -0.27 \times 1 = -0.27$$

$$\frac{\partial L}{\partial g_1} = \frac{\partial L}{\partial g_3} \times \frac{\partial g_3}{\partial g_1} = -0.27 \times 1 = -0.27$$

Example: logistic function

$$f(\boldsymbol{w}, \boldsymbol{x}) = \frac{1}{1 + e^{-(w_0 + w_1 x_1 + w_2 x_2)}}$$

Observing $(x_1, x_2, y) = (-1, -2, 1)$



$$g_2(w_2, x_2) = w_2 * x_2$$

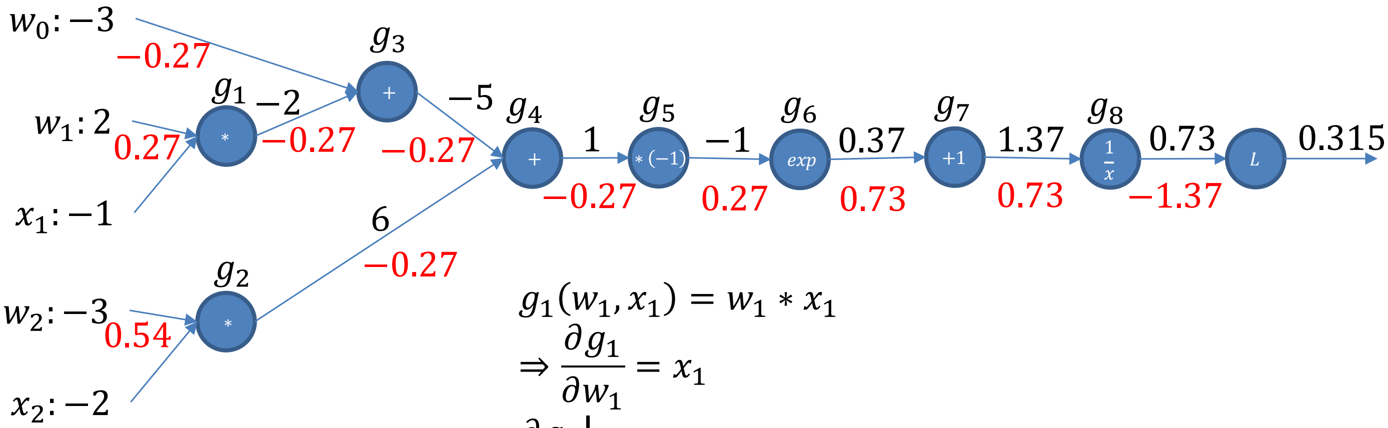
$$\Rightarrow \frac{\partial g_2}{\partial w_2} = x_2$$

$$\left.\frac{\partial g_2}{\partial w_2}\right|_{(w_2, x_2) = (-3, -2)} = -2$$

$$\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial g_2} \times \frac{\partial g_2}{\partial w_2} = -0.27 \times (-2) = 0.54$$

Example: logistic function

$$f(\boldsymbol{w}, \boldsymbol{x}) = \frac{1}{1 + e^{-(w_0 + w_1 x_1 + w_2 x_2)}}$$ Observing $(x_1, x_2, y) = (-1, -2, 1)$



$$g_1(w_1, x_1) = w_1 * x_1$$

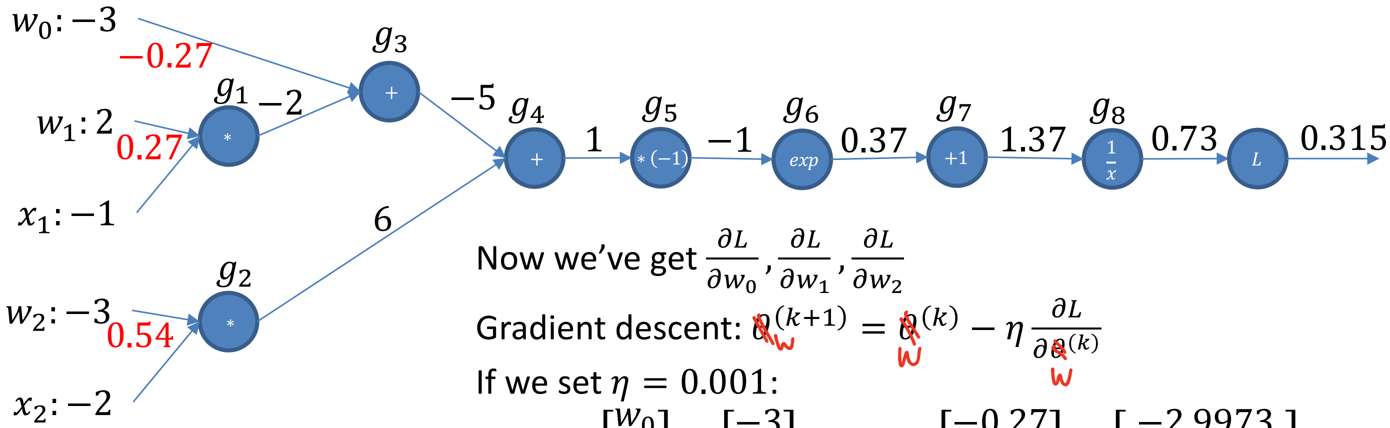
$$\Rightarrow \frac{\partial g_1}{\partial w_1} = x_1$$

$$\left.\frac{\partial g_1}{\partial w_1}\right|_{(w_1, x_1) = (2, -1)} = -1$$

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial g_1} \times \frac{\partial g_1}{\partial w_1} = -0.27 \times (-1) = 0.27$$

Example: logistic function

$$f(\boldsymbol{w}, \boldsymbol{x}) = \frac{1}{1 + e^{-(w_0 + w_1 x_1 + w_2 x_2)}}$$ Observing $(x_1, x_2, y) = (-1, -2, 1)$



Now we've get $\frac{\partial L}{\partial w_0}, \frac{\partial L}{\partial w_1}, \frac{\partial L}{\partial w_2}$

Gradient descent: $\cancel{\theta}\,w^{(k+1)} = \cancel{\theta}\,w^{(k)} - \eta \frac{\partial L}{\partial \cancel{\theta}\,w^{(k)}}$

If we set $\eta = 0.001$:

$$\begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} -3 \\ 2 \\ -3 \end{bmatrix} - 0.001 \begin{bmatrix} -0.27 \\ 0.27 \\ 0.54 \end{bmatrix} = \begin{bmatrix} -2.9973 \\ 1.9973 \\ -3.00054 \end{bmatrix}$$

Compare the old $\boldsymbol{w}$ and new $\boldsymbol{w}$:

$f(w_0 = -3, w_1 = 2, w_2 = -3, \boldsymbol{x}) = 0.7311$

$f(w_0 = -2.9973, w_1 = 1.9973, w_2 = -3.00054, \boldsymbol{x}) = 0.7323$, indeed getting closer to 1

$\sigma(x)$

$\Delta \times \sigma(1) \times [1-\sigma(1)]$

$\frac{\partial C}{\partial \sigma} = \Delta$

使 compute easy 了。

$c = a + b$

$\therefore \frac{\partial c}{\partial a} = 1, \quad \frac{\partial c}{\partial b} = 1$

Patterns in backward flow

add gate: gradient distributor

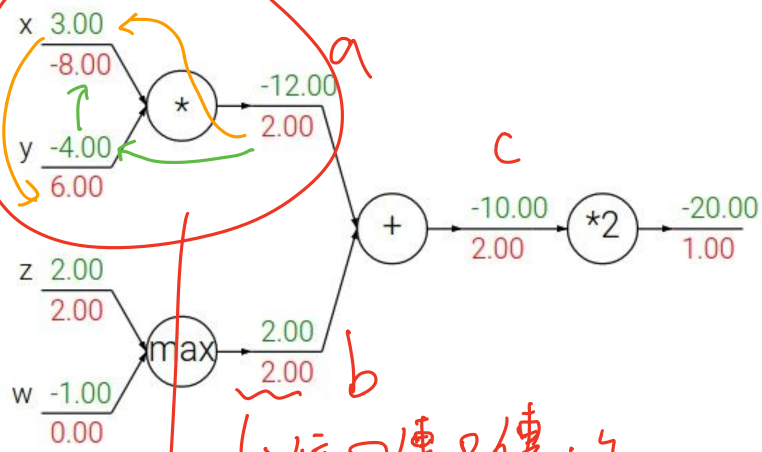
max gate: gradient router

mul gate: gradient switcher

sigmoid gate: $\sigma'(z) = \sigma(z)\big(1 - \sigma(z)\big)$

$b = \max(z, w)$

$$\frac{\partial b}{\partial z} = \begin{cases} 1 & \text{if } z > w \\ 0 & \text{if } z < w \end{cases}$$



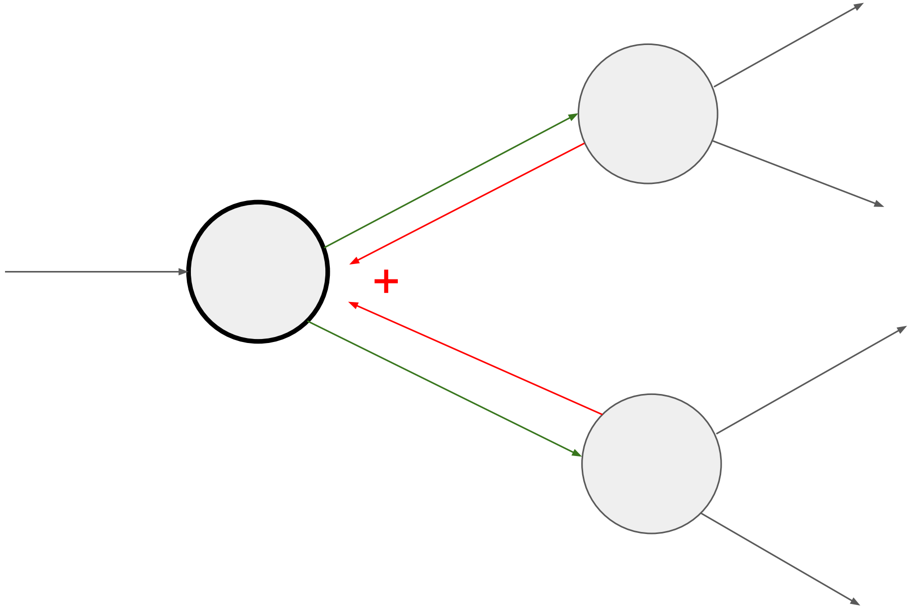
↳ 往回傳只傳1的

$$\frac{\partial b}{\partial w} = \begin{cases} 0 & \text{if } z > w \\ 1 & \text{if } z < w \end{cases}$$

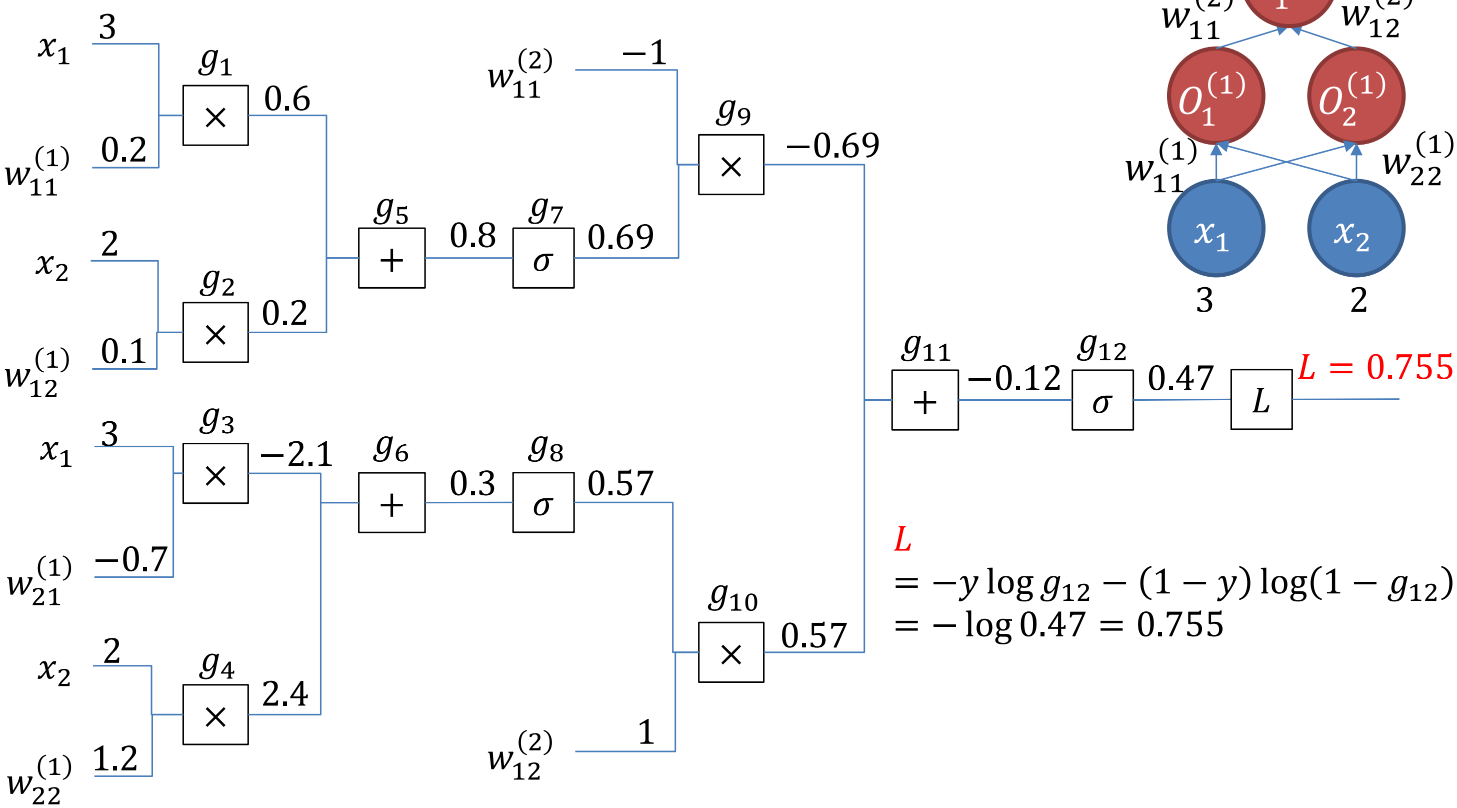
$a = x \cdot y$

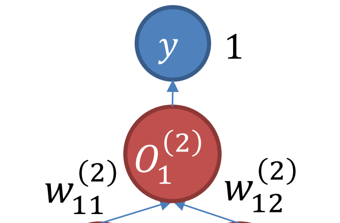
$\frac{\partial f}{\partial x} = y, \quad \frac{\partial f}{\partial y} = x$

Gradients add at branches



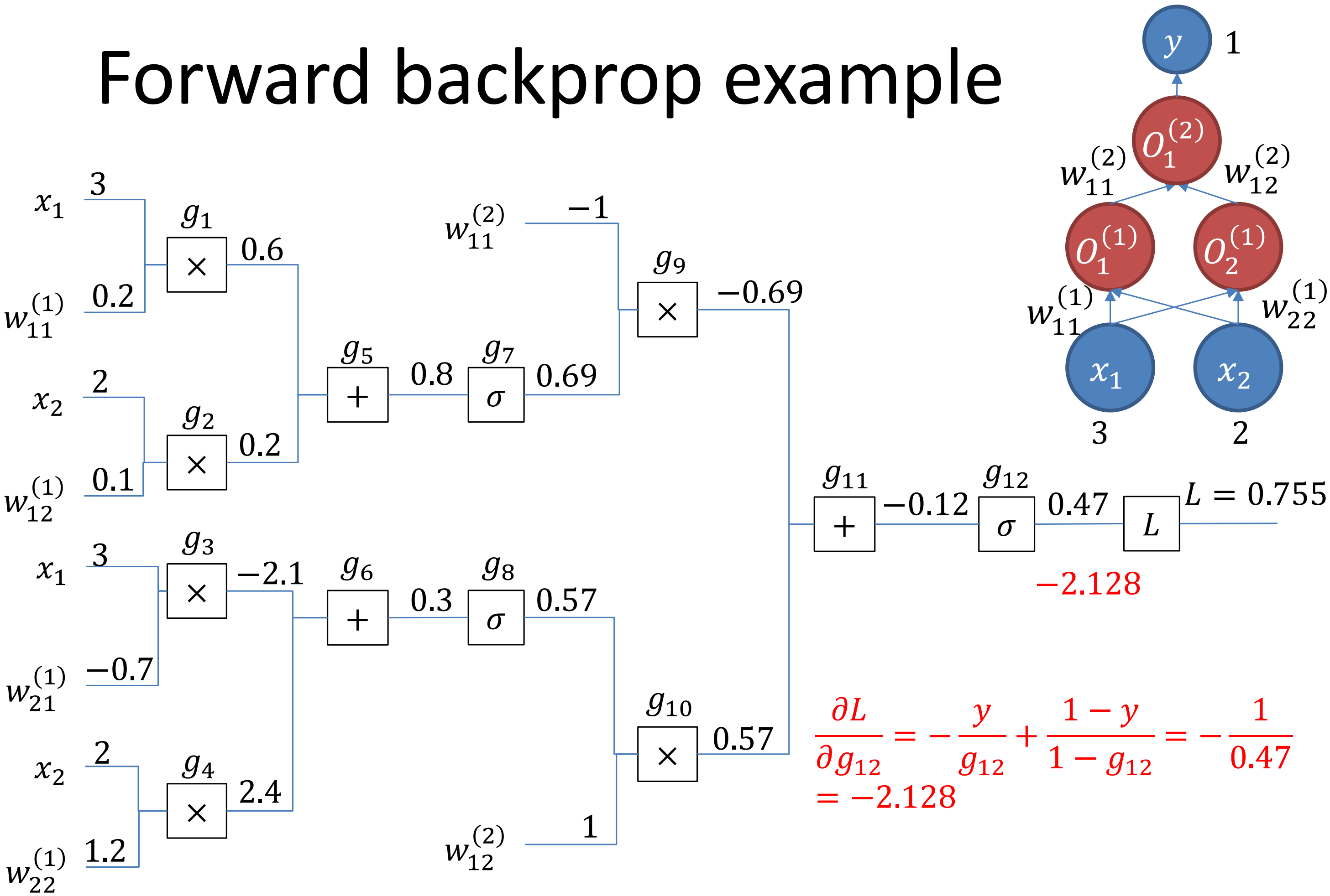
Forward backprop example





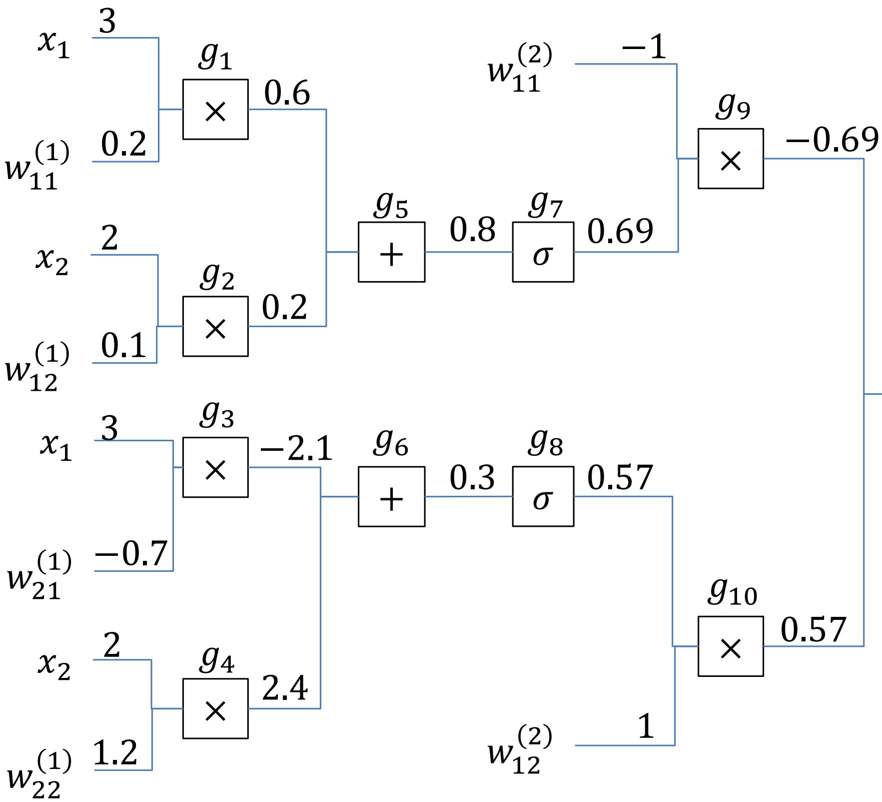
$$L = -y \log g_{12} - (1-y)\log(1-g_{12}) = -\log 0.47 = 0.755$$

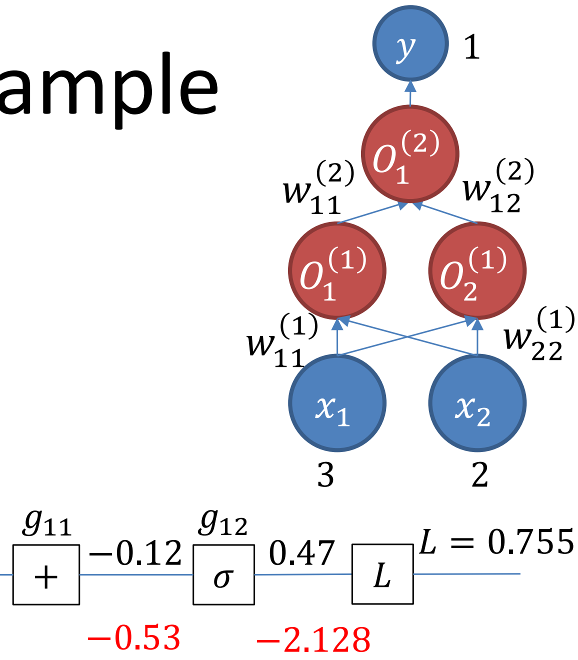
Forward backprop example



$$\frac{\partial L}{\partial g_{12}} = -\frac{y}{g_{12}} + \frac{1-y}{1-g_{12}} = -\frac{1}{0.47}$$
$$= -2.128$$

Forward backprop example





$$g_{12} = \sigma(g_{11})$$

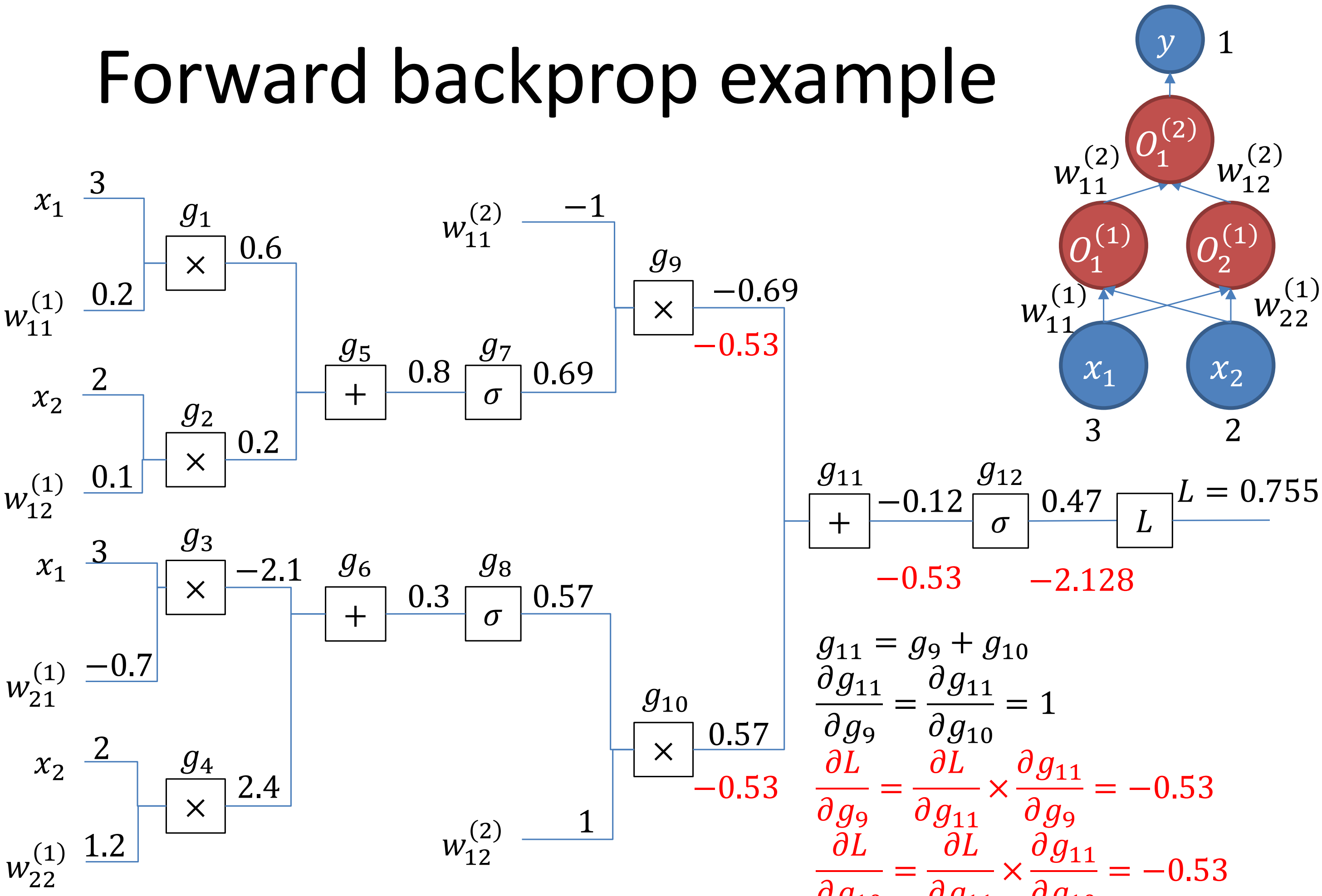
$$\frac{\partial g_{12}}{\partial g_{11}} = \sigma(g_{11})\big(1 - \sigma(g_{11})\big)$$

$$= (0.47)(0.53) = 0.2491$$

$$\frac{\partial L}{\partial g_{11}} = \frac{\partial L}{\partial g_{12}} \times \frac{\partial g_{12}}{\partial g_{11}}$$

$$= -2.128 \times 0.2491 = -0.53$$

Forward backprop example



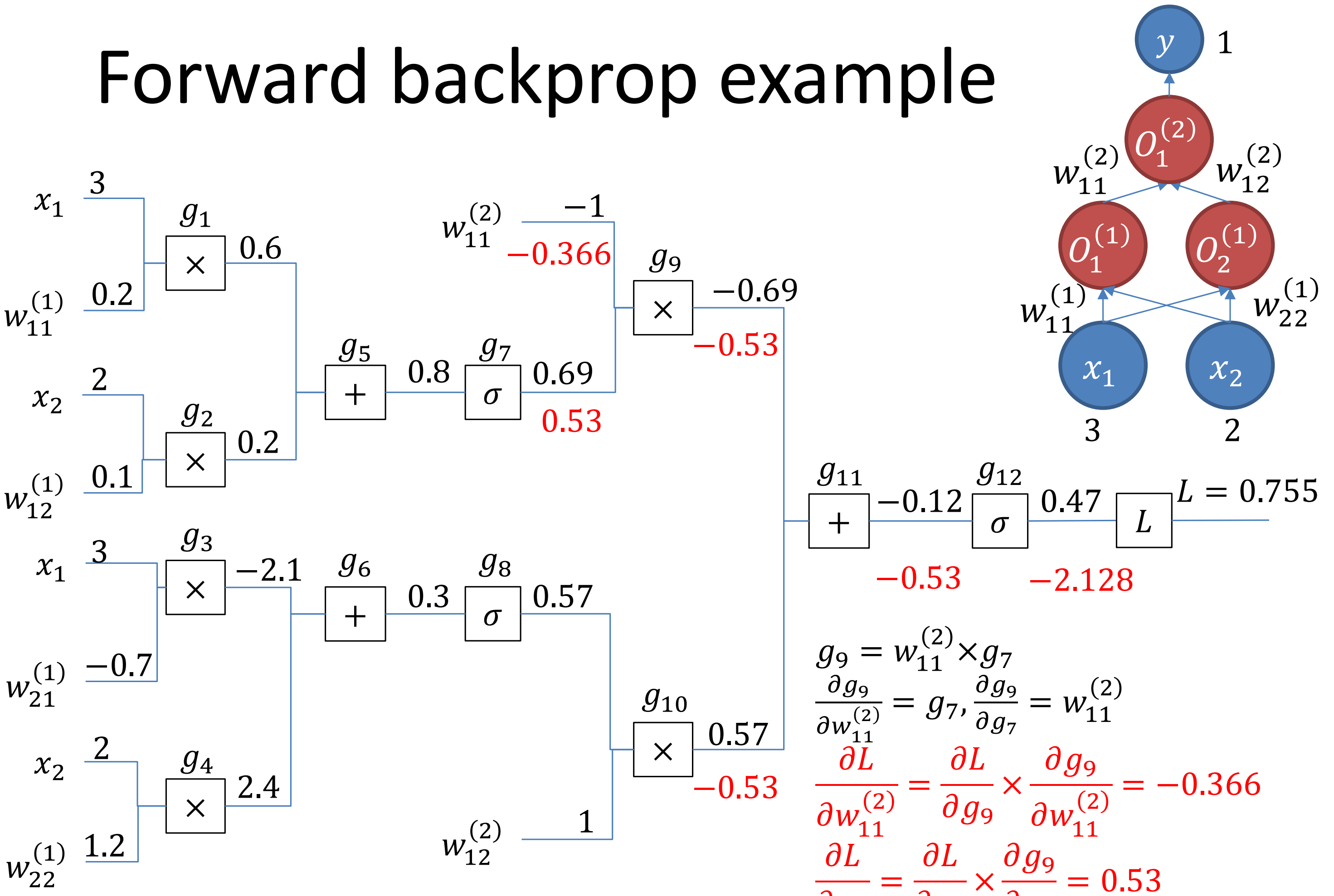
$$g_{11} = g_9 + g_{10}$$

$$\frac{\partial g_{11}}{\partial g_9} = \frac{\partial g_{11}}{\partial g_{10}} = 1$$

$$\frac{\partial L}{\partial g_9} = \frac{\partial L}{\partial g_{11}} \times \frac{\partial g_{11}}{\partial g_9} = -0.53$$

$$\frac{\partial L}{\partial g_{10}} = \frac{\partial L}{\partial g_{11}} \times \frac{\partial g_{11}}{\partial g_{10}} = -0.53$$

Forward backprop example



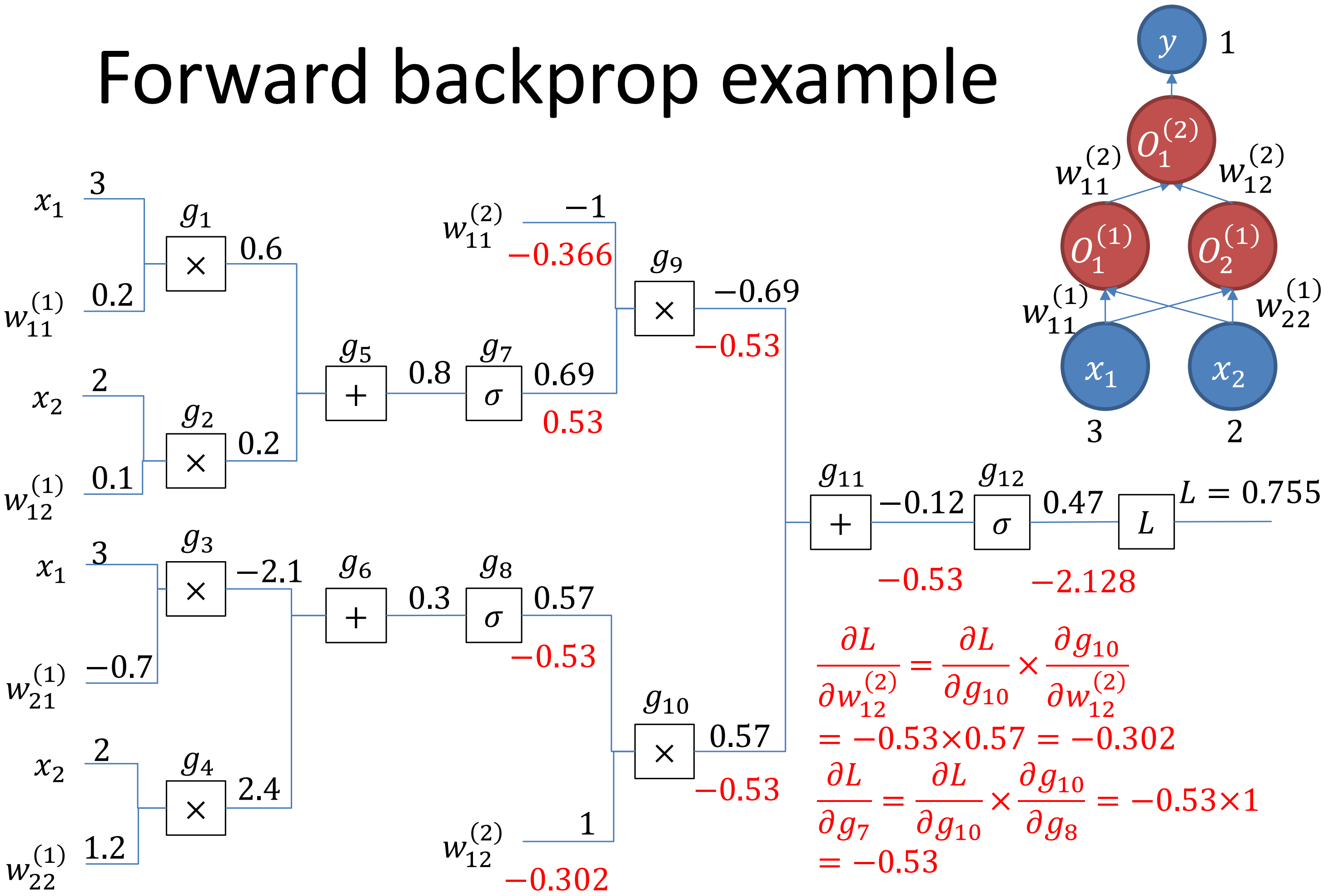
$$g_9 = w_{11}^{(2)} \times g_7$$

$$\frac{\partial g_9}{\partial w_{11}^{(2)}} = g_7, \frac{\partial g_9}{\partial g_7} = w_{11}^{(2)}$$

$$\frac{\partial L}{\partial w_{11}^{(2)}} = \frac{\partial L}{\partial g_9} \times \frac{\partial g_9}{\partial w_{11}^{(2)}} = -0.366$$

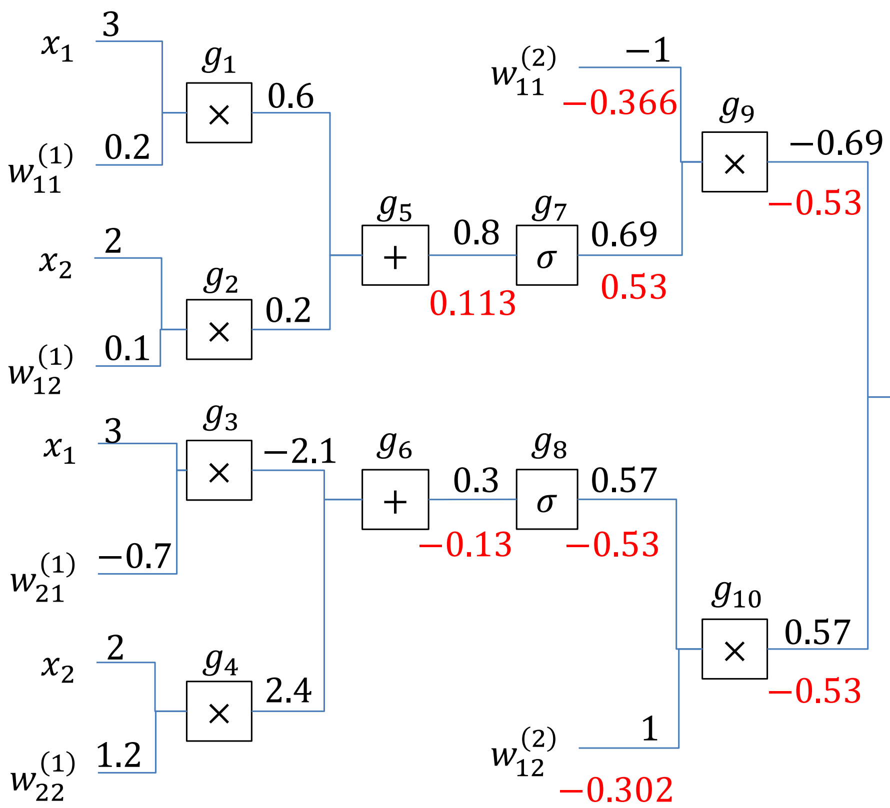
$$\frac{\partial L}{\partial g_7} = \frac{\partial L}{\partial g_9} \times \frac{\partial g_9}{\partial g_7} = 0.53$$

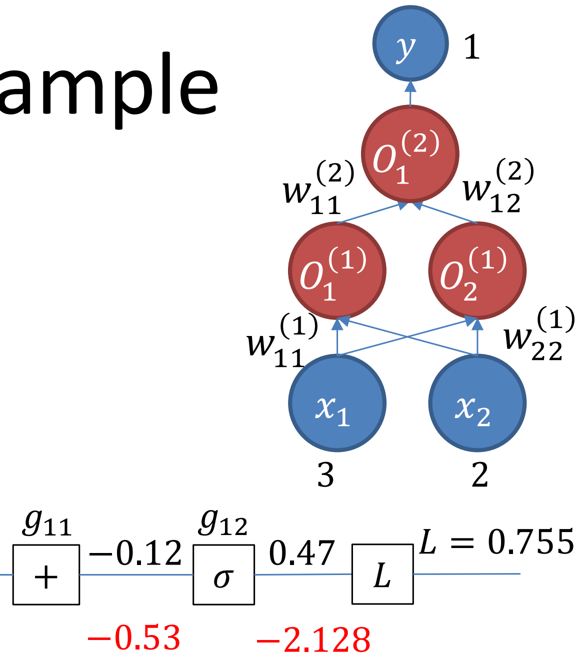
Forward backprop example



$$\frac{\partial L}{\partial w_{12}^{(2)}} = \frac{\partial L}{\partial g_{10}} \times \frac{\partial g_{10}}{\partial w_{12}^{(2)}}$$
$$= -0.53 \times 0.57 = -0.302$$
$$\frac{\partial L}{\partial g_7} = \frac{\partial L}{\partial g_{10}} \times \frac{\partial g_{10}}{\partial g_8} = -0.53 \times 1$$
$$= -0.53$$

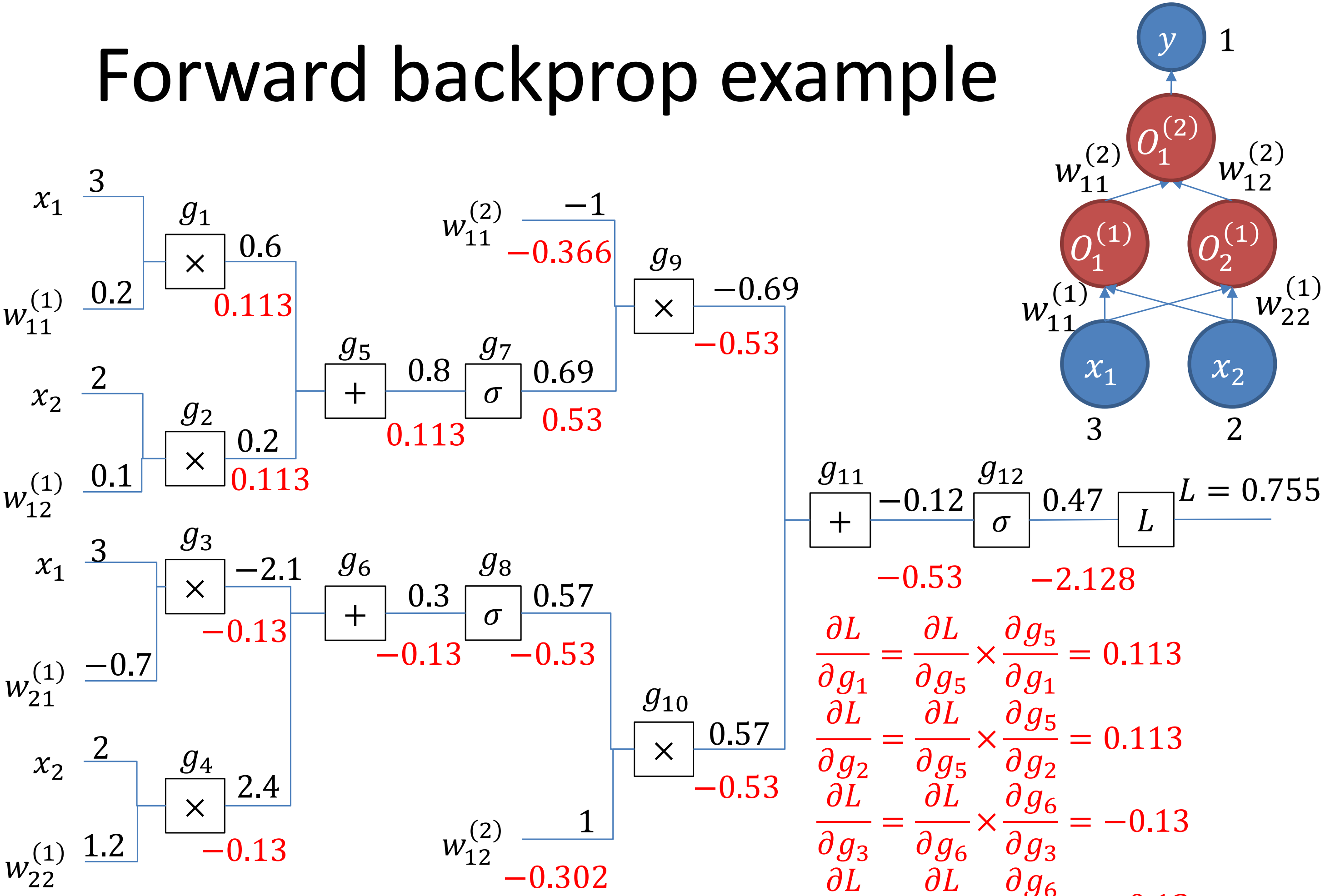
Forward backprop example





$$\frac{\partial L}{\partial g_5} = \frac{\partial L}{\partial g_7} \times \frac{\partial g_7}{\partial g_5}$$
$$= 0.53 \times \big(0.69(1 - 0.69)\big) = 0.113$$
$$\frac{\partial L}{\partial g_6} = \frac{\partial L}{\partial g_8} \times \frac{\partial g_8}{\partial g_6}$$
$$= -0.53 \times \big(0.57(1 - 0.57)\big)$$
$$= -0.13$$

Forward backprop example



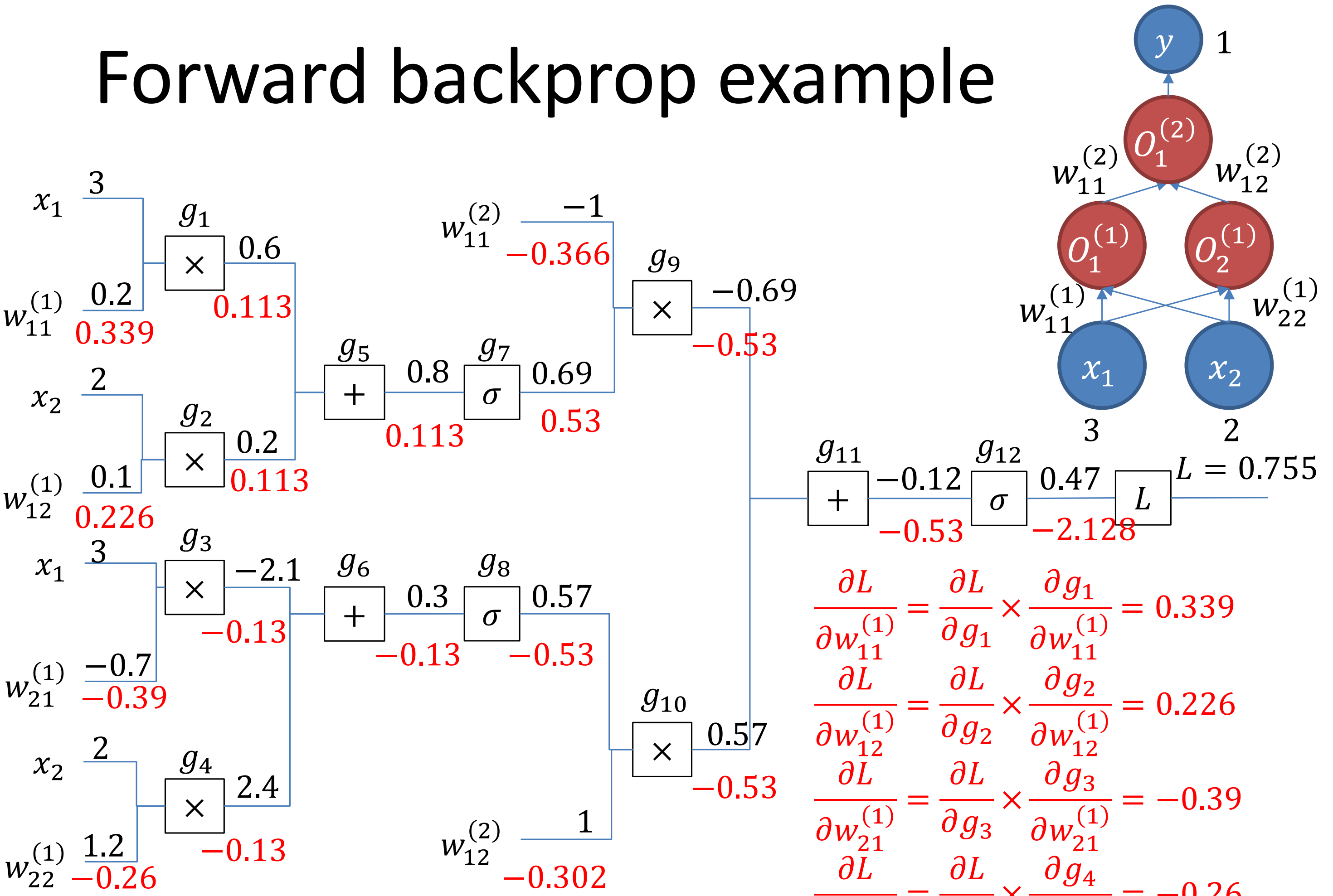
$$\frac{\partial L}{\partial g_1} = \frac{\partial L}{\partial g_5} \times \frac{\partial g_5}{\partial g_1} = 0.113$$

$$\frac{\partial L}{\partial g_2} = \frac{\partial L}{\partial g_5} \times \frac{\partial g_5}{\partial g_2} = 0.113$$

$$\frac{\partial L}{\partial g_3} = \frac{\partial L}{\partial g_6} \times \frac{\partial g_6}{\partial g_3} = -0.13$$

$$\frac{\partial L}{\partial g_4} = \frac{\partial L}{\partial g_6} \times \frac{\partial g_6}{\partial g_4} = -0.13$$

Forward backprop example



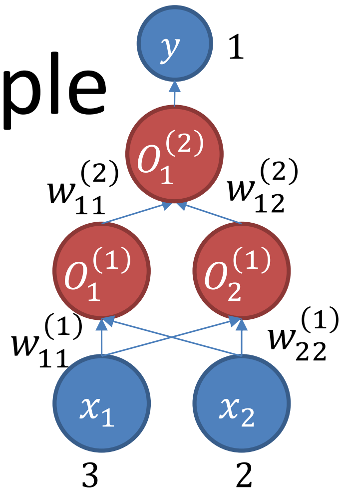
$$\frac{\partial L}{\partial w_{11}^{(1)}} = \frac{\partial L}{\partial g_1} \times \frac{\partial g_1}{\partial w_{11}^{(1)}} = 0.339$$

$$\frac{\partial L}{\partial w_{12}^{(1)}} = \frac{\partial L}{\partial g_2} \times \frac{\partial g_2}{\partial w_{12}^{(1)}} = 0.226$$

$$\frac{\partial L}{\partial w_{21}^{(1)}} = \frac{\partial L}{\partial g_3} \times \frac{\partial g_3}{\partial w_{21}^{(1)}} = -0.39$$

$$\frac{\partial L}{\partial w_{22}^{(1)}} = \frac{\partial L}{\partial g_4} \times \frac{\partial g_4}{\partial w_{22}^{(1)}} = -0.26$$

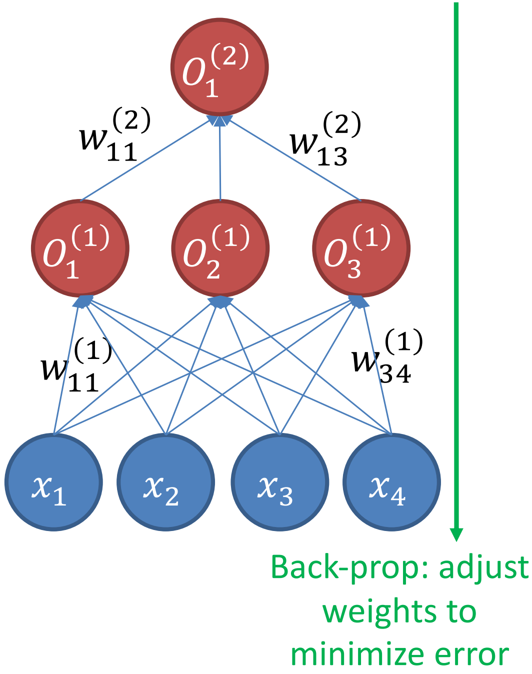
Forward backprop example



- Use original $\boldsymbol{w}^{(1)}, \boldsymbol{w}^{(2)}$:
  - $\hat{y} = 0.4711$
- Use the new $\boldsymbol{w}^{(1)}, \boldsymbol{w}^{(2)}$ computed by gradient descent ($\eta = 0.001$)
  - $\boldsymbol{w}^{(\ell)} = \boldsymbol{w}^{(\ell)} - \eta \nabla_{\boldsymbol{w}^{(\ell)}} L$
  - $\hat{y} = 0.4714$
- The new $\hat{y}$ indeed closer to 1

Back-prop update

- Back-prop
  - $\frac{\partial loss}{\partial \boldsymbol{I}^{(2)}} = \frac{\partial loss}{\partial \boldsymbol{O}^{(2)}} \frac{\partial \boldsymbol{O}^{(2)}}{\partial \boldsymbol{I}^{(2)}}$
  - $\frac{\partial loss}{\partial \boldsymbol{W}^{(2)}} = \frac{\partial loss}{\partial \boldsymbol{I}^{(2)}} \frac{\partial \boldsymbol{I}^{(2)}}{\partial \boldsymbol{W}^{(2)}}$
  - $\frac{\partial loss}{\partial \boldsymbol{I}^{(1)}} = \frac{\partial loss}{\partial \boldsymbol{I}^{(2)}} \frac{\partial \boldsymbol{I}^{(2)}}{\partial \boldsymbol{O}^{(1)}} \frac{\partial \boldsymbol{O}^{(1)}}{\partial \boldsymbol{I}^{(1)}}$
  - $\frac{\partial loss}{\partial \boldsymbol{W}^{(1)}} = \frac{\partial loss}{\partial \boldsymbol{I}^{(1)}} \frac{\partial \boldsymbol{I}^{(1)}}{\partial \boldsymbol{W}^{(1)}}$



- $\boldsymbol{I}^{(1)} = \boldsymbol{W}^{(1)}\boldsymbol{x}$
- $\boldsymbol{O}^{(1)} = \sigma\left(\boldsymbol{I}^{(1)}\right)$
- $\boldsymbol{I}^{(2)} = \boldsymbol{W}^{(2)}\boldsymbol{O}^{(1)}$
- $\boldsymbol{O}^{(2)} = \sigma\left(\boldsymbol{I}^{(2)}\right)$

Red: computed in higher layers
Blue: depends on the activation function
Green: depends on the loss function

Weight updates

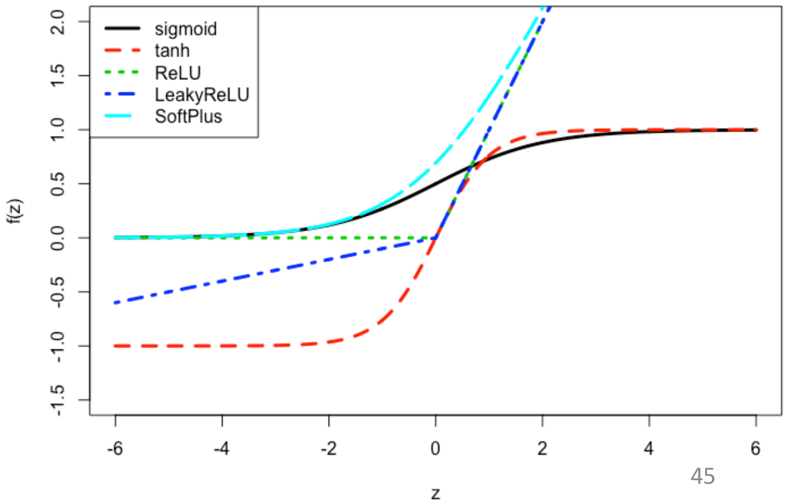
- Update $\boldsymbol{W}^{(\ell)}$ by (stochastic) gradient descent
  - Compute $\frac{\partial loss}{\partial \boldsymbol{W}^{(\ell)}}$ for all $\boldsymbol{W}^{(\ell)}$
  - $\boldsymbol{W}^{(\ell)} \leftarrow \boldsymbol{W}^{(\ell)} - \eta \frac{\partial loss}{\partial \boldsymbol{W}^{(\ell)}}$

Loss functions

- K-nary multi-class classification: cross entropy loss
  - $Loss = -\sum_{k=1}^{K} I(y_k = 1)\log \hat{y}$
    - Example: 3 classes
      - $p(\hat{y}_i = k) = [1/4 \quad 1/4 \quad 1/2]$
      - $p(y_i = k) = [0 \quad 1 \quad 0]$
      - $-\sum_k p(y_i = k)\log[p(\hat{y}_i = k)] = -\left[0\log\frac{1}{4} + 1\log\frac{1}{4} + 0\log\frac{1}{2}\right] = 2$
  - When $K = 2 \Rightarrow$ back to binary classification ($y \in \{0,1\}$)
    - $Loss = -y\log\hat{y} - (1-y)\log(1-\hat{y})$
- Regression
  - $Loss = \frac{1}{2}(y - \hat{y})^2$
- Multi-label classification (assuming L possible labels $(\ell_1, \ldots, \ell_L)$ for each instance)
  - $Loss = \sum_{i=1}^{L}\left(-\ell_i \log\hat{\ell}_i - (1-\ell_i)\log\left(1-\hat{\ell}_i\right)\right)$

Common activation functions

- Logistic (sigmoid) function
  - $f(z) = \frac{1}{1+\exp(-z)}$
- Hyperbolic (tanh) function
  - $f(z) = \frac{2}{1+\exp(-2z)} - 1$
- Rectified Linear Unit (ReLU)
  - $f(z) = \max(0, z)$
- Soft-plus
  - $f(z) = \log(1 + \exp(z))$
- Leaky ReLU
  - $f(z) = \begin{cases} z \text{ if } z \geq 0 \\ \alpha z \text{ if } z < 0 \end{cases}$
    - α is small and positive (e.g., 0.01)

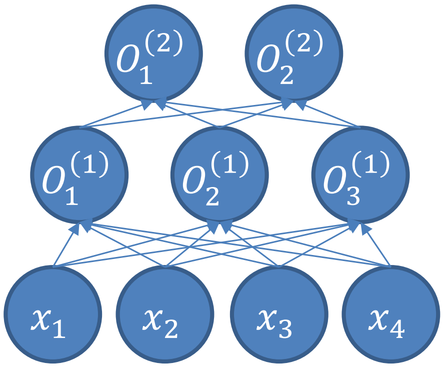


Activation function for the output layer

- For the final layer, select the activation function based on the type of your task
  - Regression: return one value based on linear activation function
    - $f(z) = z$
  - K-ary multi-class classification: return a vector based on softmax function
    - $f(\boldsymbol{z}) = \left[\frac{e^{z_1}}{\sum_j e^{z_j}}, \frac{e^{z_2}}{\sum_j e^{z_j}}, \dots, \frac{e^{z_K}}{\sum_j e^{z_j}}\right]$ or softmax
      - $\text{Sum}\big(f(\boldsymbol{z})\big) = 1$
  - Multi-label classification: return a vector based on sigmoid function
    - $f(\boldsymbol{z}) = \left[\frac{1}{1+\exp(-z_1)}, \frac{1}{1+\exp(-z_2)}, \dots, \frac{1}{1+\exp(-z_K)}\right]$
      - $\text{Sum}\big(f(\boldsymbol{z})\big) \neq 1$

Why non-linear activation function?

- If no (non-linear) activation function, multi-layer reduces to single layer



$\boldsymbol{W}^{(2)} \in R^{2\times 3}$

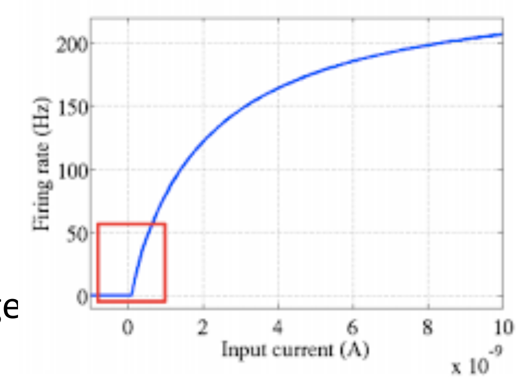
$\boldsymbol{W}^{(1)} \in R^{3\times 4}$

$$\begin{aligned}\boldsymbol{O}^{(2)} &= \boldsymbol{W}^{(2)}\boldsymbol{O}^{(1)}\\ &= \boldsymbol{W}^{(2)}\left(\boldsymbol{W}^{(1)}\boldsymbol{x}\right)\\ &= \left(\boldsymbol{W}^{(2)}\boldsymbol{W}^{(1)}\right)\boldsymbol{x}\end{aligned}$$

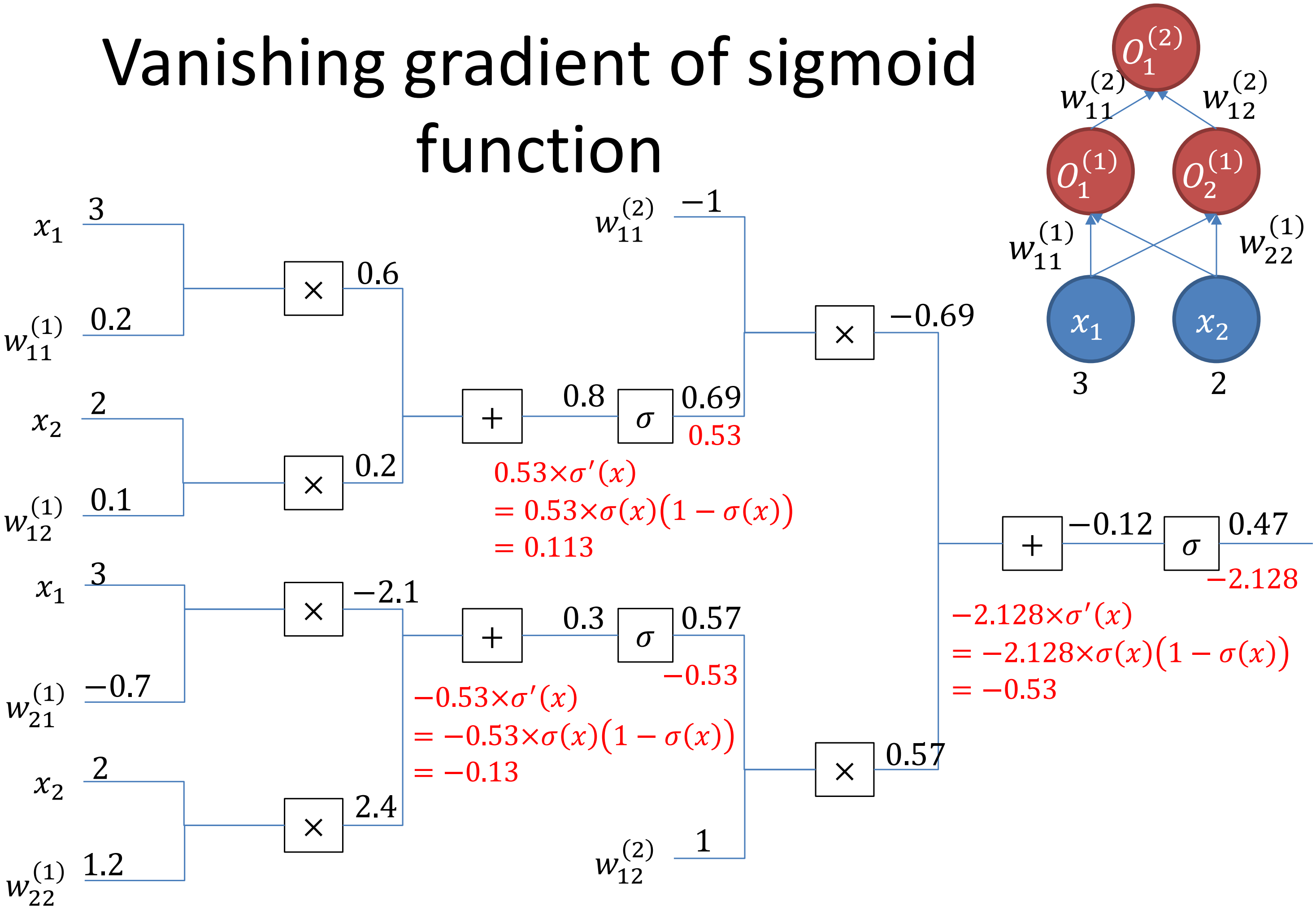
$$\Rightarrow \left(\boldsymbol{W}^{(2)}\boldsymbol{W}^{(1)}\right) \in R^{2\times 4}$$

ReLU: "default" activation function for the hidden layers

- Vanishing gradient problem in sigmoid and tanh
  - When z is outside $[-4, 4]$, $f'_{sigmoid}(z) \approx 0$ and $f'_{tanh}(z) \approx 0$ ➔ new information cannot be back-propagated
- ReLU's sparse neuron outputs may prevent overfitting
  - When $z < 0, f_{ReLU}(z) = 0$ ➔ many neuron outputs are zero ➔ model becomes smaller
- ReLU's computation cost is small
  - No exponential computation
- ReLU's may resemble biological neurons
  - Firing rate > 0 only when input current is large enough (see the figure)

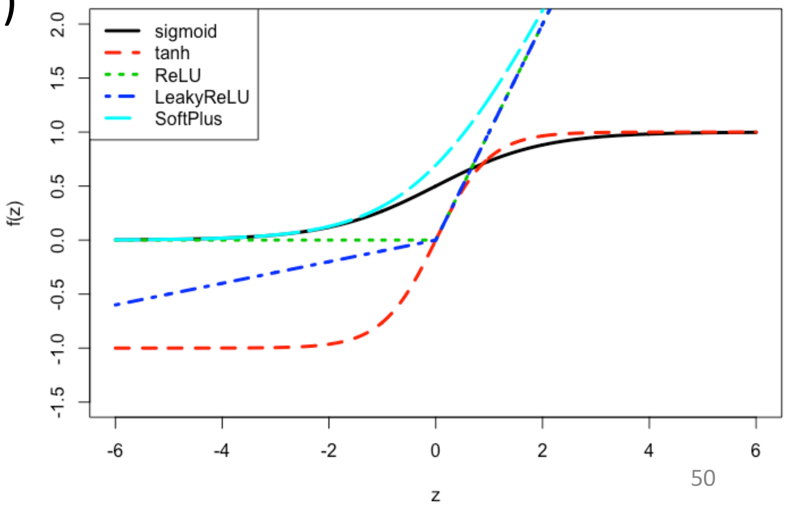


Vanishing gradient of sigmoid function



Derivative of activation functions

- Logistic (sigmoid) function
  - $f'(z) = f(z)\big(1 - f(z)\big)$
- Hyperbolic (tanh) function
  - $f'^{(z)} = 1 - \big(f(z)\big)^2$
- Rectified Linear Unit (ReLU)
  - $f(z) = \begin{cases} 1 & \text{if } z > 0 \\ 0 & \text{else} \end{cases}$
- Soft-plus
  - $f'(z) = \frac{1}{1+\exp(-z)}$
- Leaky ReLU
  - $f'(z) = \begin{cases} 1 \text{ if } z \geq 0 \\ \alpha \text{ if } z < 0 \end{cases}$
    - α is small and positive (e.g., 0.01)



Example: write your own activation operation

- Define autograd function (in PyTorch) by writing forward and backward for Tensors

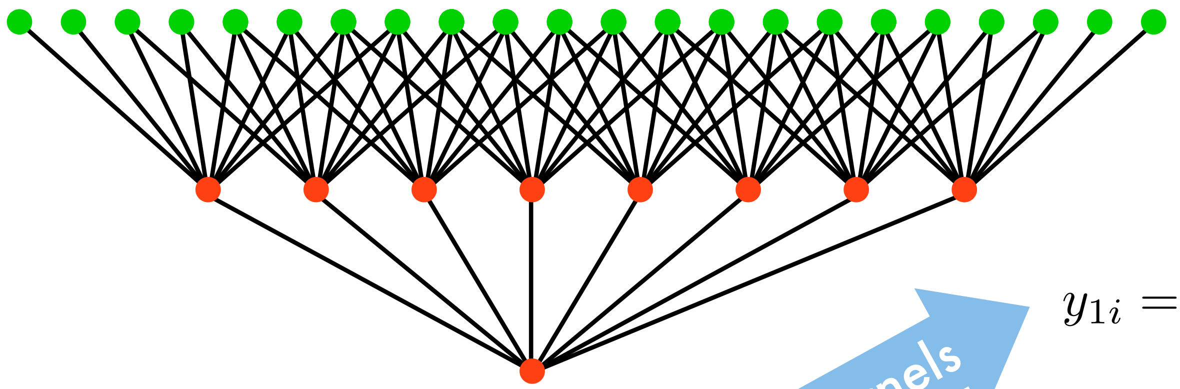
```python
class ReLU(torch.autograd.Function):
    def forward(self, x):
        self.save_for_backward(x)
        return x.clamp(min=0)

    def backward(self, grad_y):
        x, = self.saved_tensors
        grad_input = grad_y.clone()
        grad_input[x < 0] = 0
        return grad_input
```

- You may define your customized activation functions

- PyTorch performs chain rules for you if all operators on the graph are well-defined:
  - `loss.backward()`

Nonlinearities via Layers



$$y_{1i}(x) = \sigma(\langle w_{1i}, x \rangle)$$
$$y_2(x) = \sigma(\langle w_2, y_1 \rangle)$$

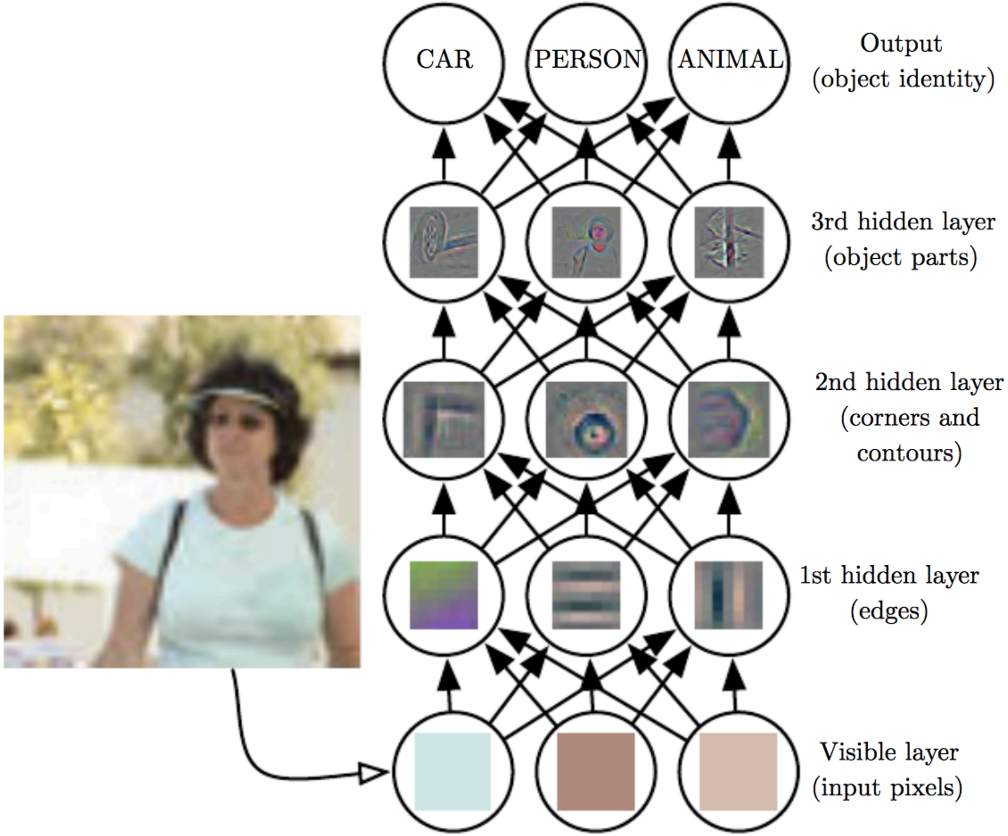
Kernels

$$y_{1i} = k(x_i, x)$$

Deep Nets

optimize
all weights

Learning representations



Popular deep learning framework

- Most popular
  - Tensorflow (Keras), PyTorch
- Others
  - MXNet, Caffe and Caffe2, Theano, Torch

Other successful deep learning architecture

- Convolutional Neural Network (CNN)
- Recurrent Neural Network (RNN)

Conclusion

- Supervised DNN is composed by many hidden layers
  - Learning representations
  - Interaction between features to form high-dimensional features
  - Non-linearity property
- Training a supervised DNN is based on SGD
  - View few (e.g., 32) instances in on batch
  - Chain rule and backpropagation are the tricks to perform derivative