CS489/698: Intro to ML

Lecture 04: Multi-layer Perceptron

UNIVERSITY OF WATERLOO

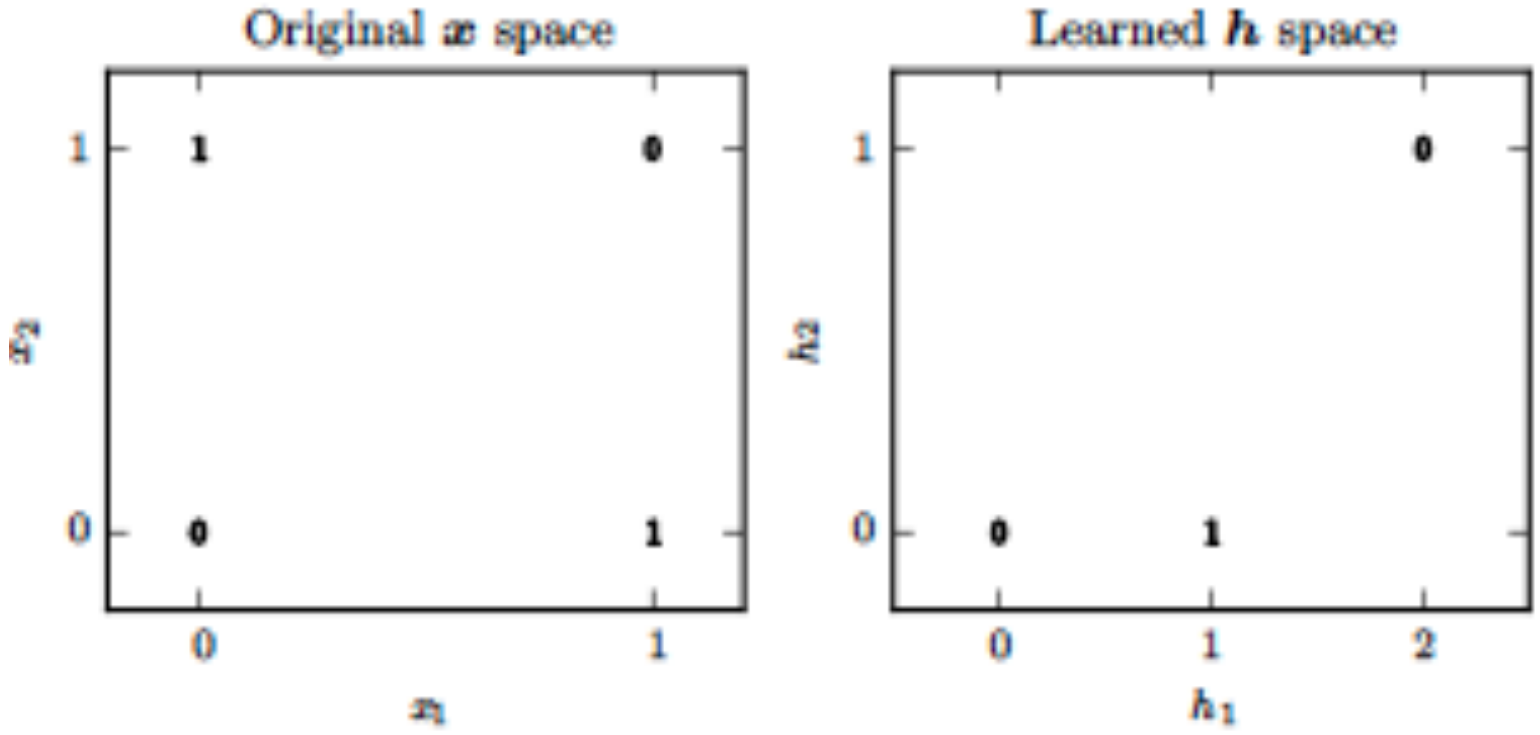
Outline

- Failure of Perceptron
- Neural Network
- Backpropagation
- Code

Outline

- Failure of Perceptron
- Neural Network
- Backpropagation
- Code

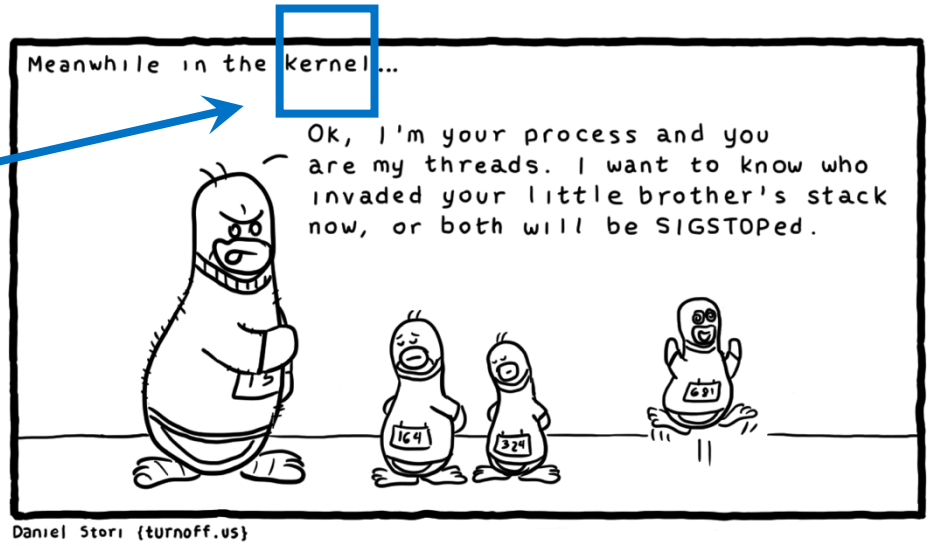
The XOR problem



- X = { (0,0), (0,1), (1,0), (1,1) }, y = {-1, 1, 1, -1}
- $f^*(\mathbf{x}) = \text{sign}[\, f_{xor} - ½ \,]$, $f_{xor}(\mathbf{x}) = (x_1 \,\&\, -x_2) \mid (-x_1 \,\&\, x_2)$

Fixing the problem

- Our model (hyperplanes) underfits the data (xor func)
- Fix representation, richer model
- Fix model, richer representation

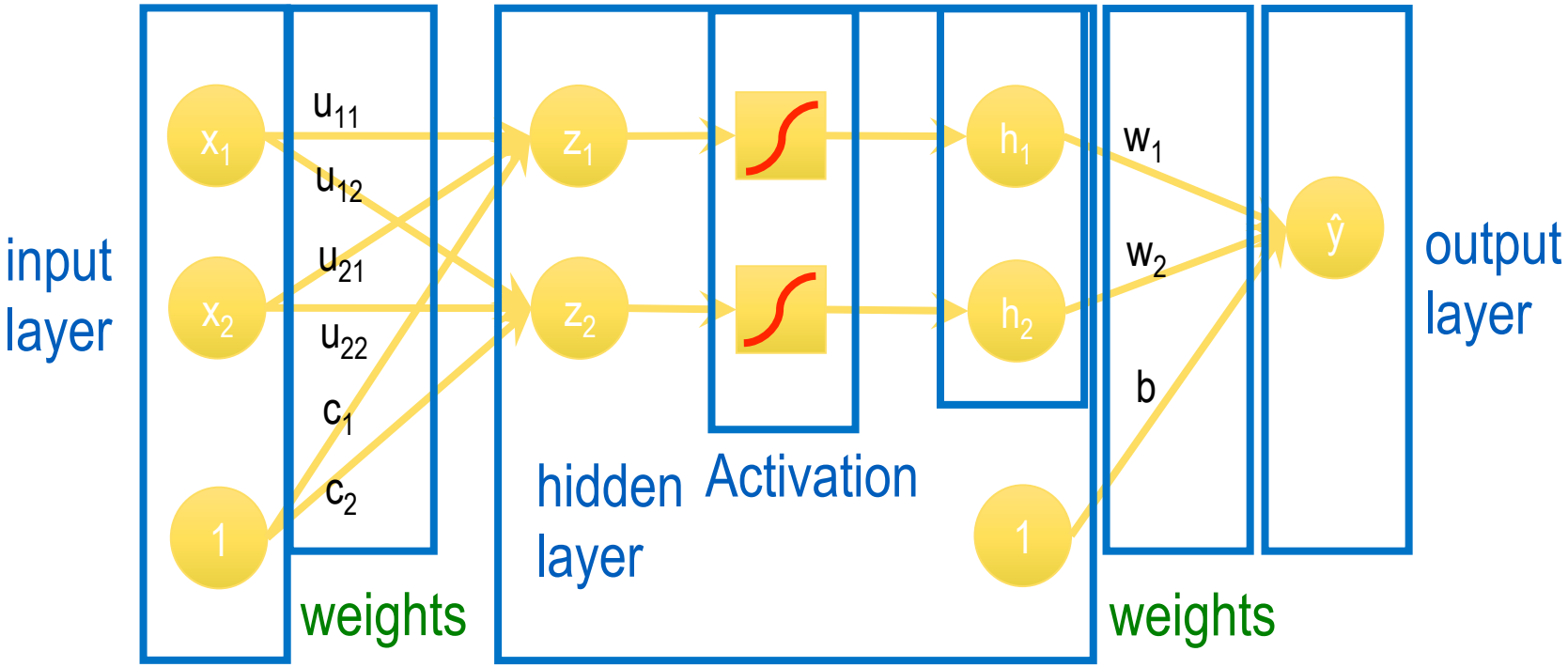


- NN: still use hyperplane, but learn representation simultaneously

Outline

- Failure of Perceptron
- Neural Network
- Backpropagation
- Code

Two-layer perceptron



$$\mathbf{z} = U\mathbf{x} + \mathbf{c}$$ 1st linear layer

makes all the difference! $$\mathbf{h} = f(\mathbf{z})$$ nonlinear transform

2nd linear layer $$\hat{y} = \langle \mathbf{h}, \mathbf{w} \rangle + b$$

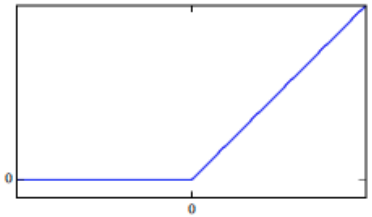
UNIVERSITY OF WATERLOO

Does it work?

$$U = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad \mathbf{c} = \begin{bmatrix} 0 \\ -1 \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} 2 \\ -4 \end{bmatrix} \quad b = -1$$

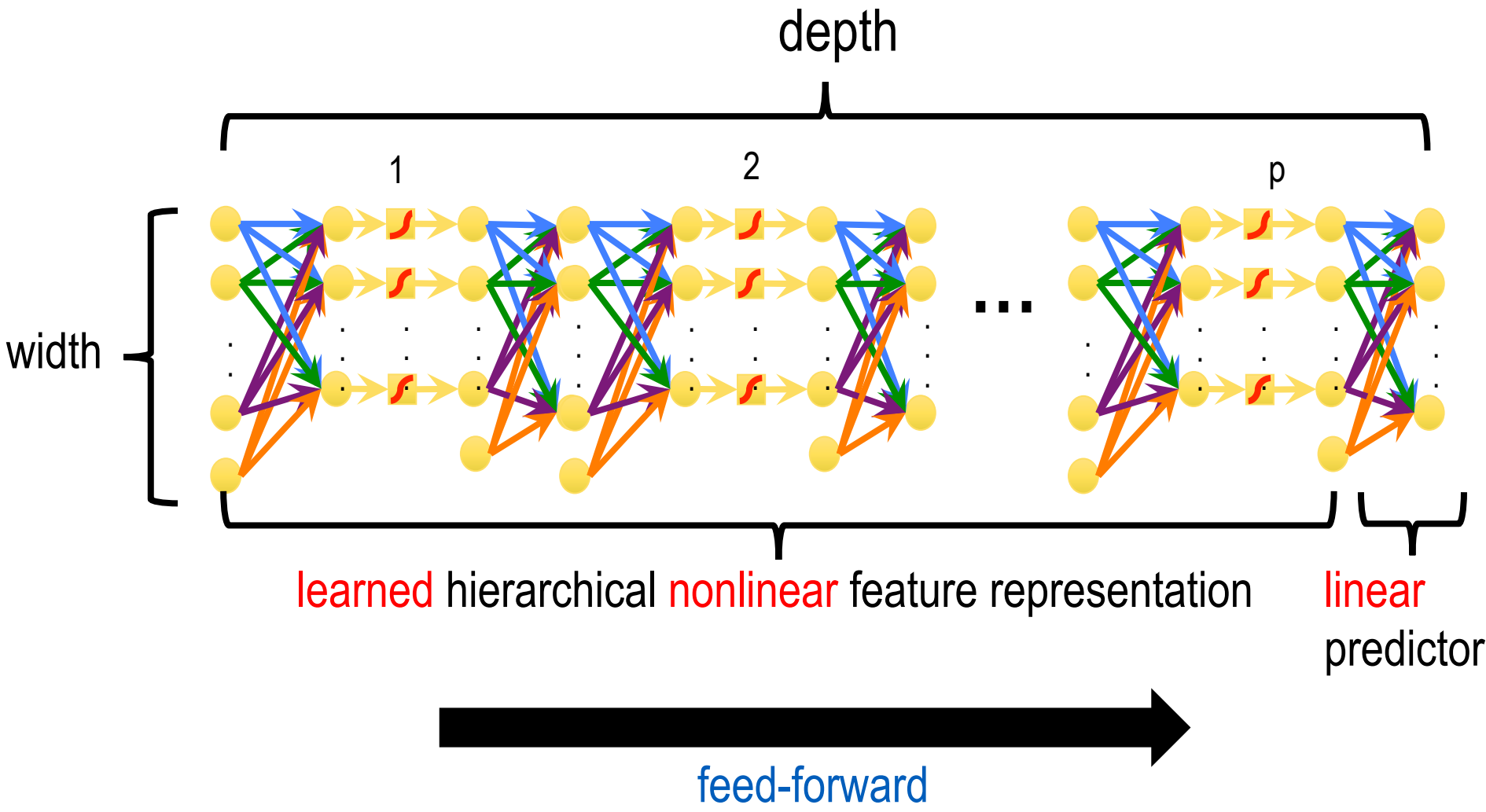
Rectified Linear Unit (ReLU)

$$f(t) = t_+ := \max(t, 0)$$



- $\mathbf{x}_1 = (0,0)$, $y_1 = -1 \rightarrow \mathbf{z}_1 = (0,-1)$, $\mathbf{h}_1 = (0,0) \rightarrow \hat{y}_1 = -1$ ✔
- $\mathbf{x}_2 = (0,1)$, $y_2 = 1 \rightarrow \mathbf{z}_2 = (1,0)$, $\mathbf{h}_2 = (1,0) \rightarrow \hat{y}_2 = 1$ ✔
- $\mathbf{x}_3 = (1,0)$, $y_3 = 1 \rightarrow \mathbf{z}_3 = (1,0)$, $\mathbf{h}_3 = (1,0) \rightarrow \hat{y}_3 = 1$ ✔
- $\mathbf{x}_4 = (1,1)$, $y_4 = -1 \rightarrow \mathbf{z}_4 = (2,1)$, $\mathbf{h}_4 = (2,1) \rightarrow \hat{y}_4 = -1$ ✔

Multi-layer perceptron (stacked)



UNIVERSITY OF WATERLOO

Activation function

- Sigmoid

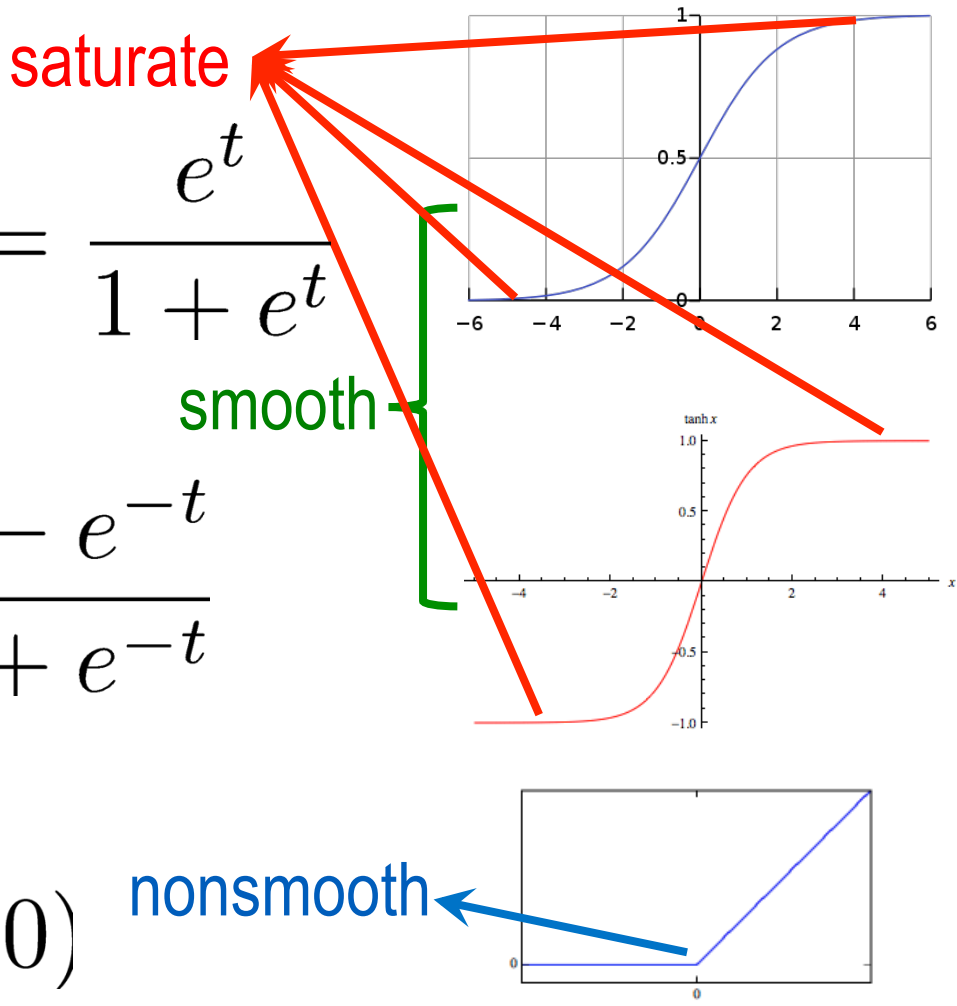
$$f(t) = \sigma(t) = \frac{1}{1+e^{-t}} = \frac{e^t}{1+e^t}$$

- Tanh

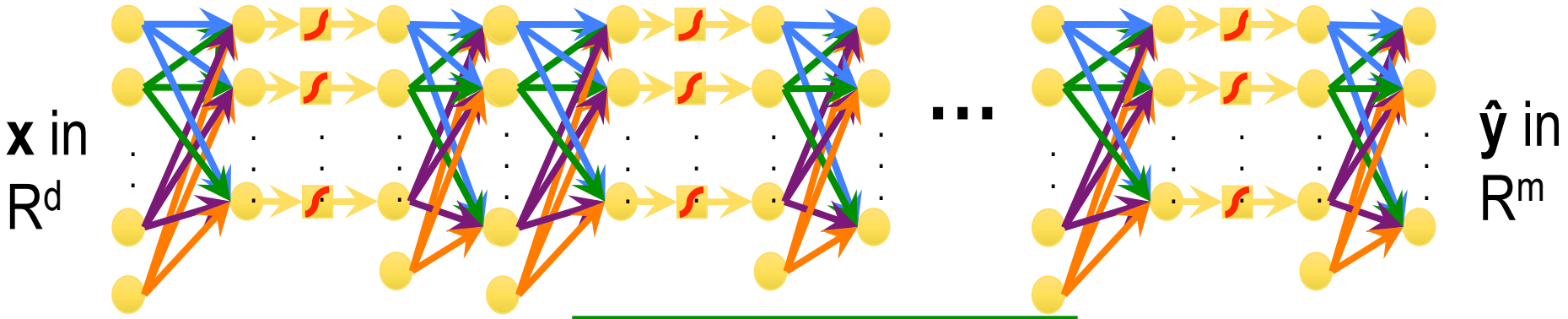
$$f(t) = \tanh(t) = \frac{e^t - e^{-t}}{e^t + e^{-t}}$$

- Rectified Linear

$$f(t) = t_+ := \max(t, 0)$$



Weights training



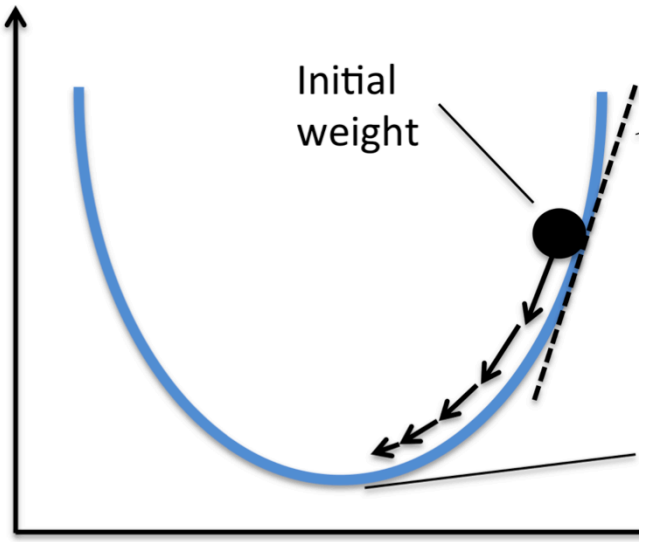
$$\hat{y} = q(x; \Theta)$$

- Need a loss ℓ to measure diff. between pred $\hat{y}$ and truth y
  - E.g., $(\hat{y}-y)^2$; more later
- Need a training set $\{(x_1,y_1), \ldots, (x_n,y_n)\}$ to train weights Θ

Gradient Descent

$$\min_{\Theta} L(\Theta) := \frac{1}{n} \sum_{i=1}^{n} \ell\left[\mathbf{q}(\mathbf{x}_i; \Theta), \mathbf{y}_i\right]$$

$$\Theta_{t+1} \leftarrow \Theta_t - \eta_t \nabla L(\Theta_t)$$



Step size (learning rate)

- const., if L is smooth
- diminishing, otherwise

Outline

- Failure of Perceptron
- Neural Network
- Backpropagation
- Code

Backpropogation

- A fancy name for the chain rule for derivatives:

$$f(x) = g[\, h(x) \,] \rightarrow f'(x) = g'[\, h(x) \,] * h'(x)$$

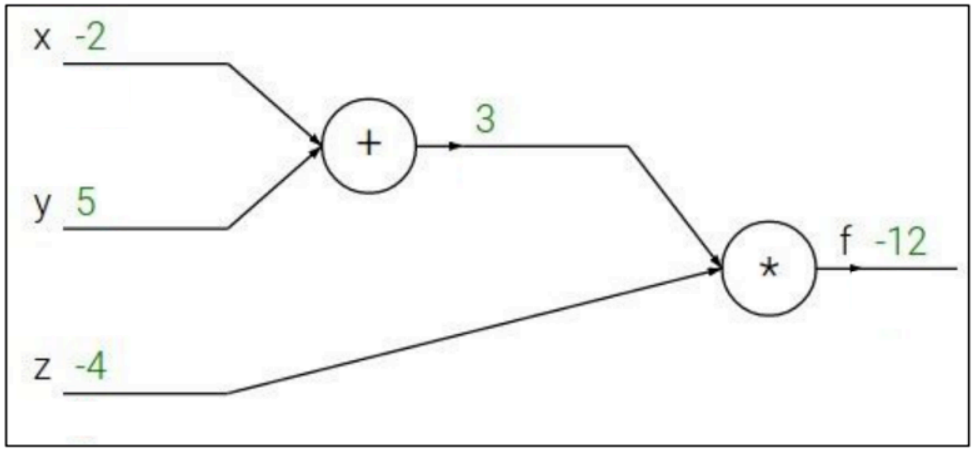
- Efficiently computes the derivative in NN

- Two passes; complexity = O(size(NN))
  - forward pass: compute function value sequentially
  - backward pass: compute derivative sequentially

Backprop through a Computation Graph

$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4



http://web.stanford.edu/class/cs224n/lectures/cs224n-2017-lecture5.pdf

Recursive differentiation

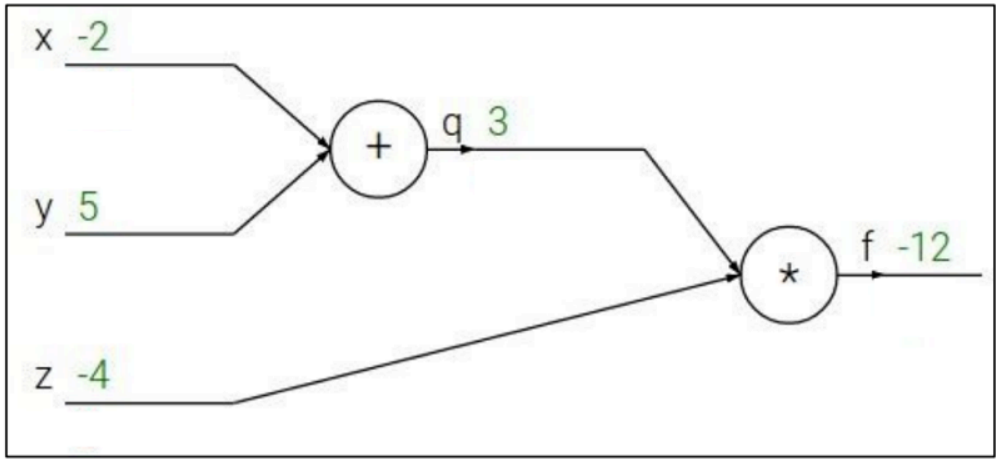
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



Recursive differentiation

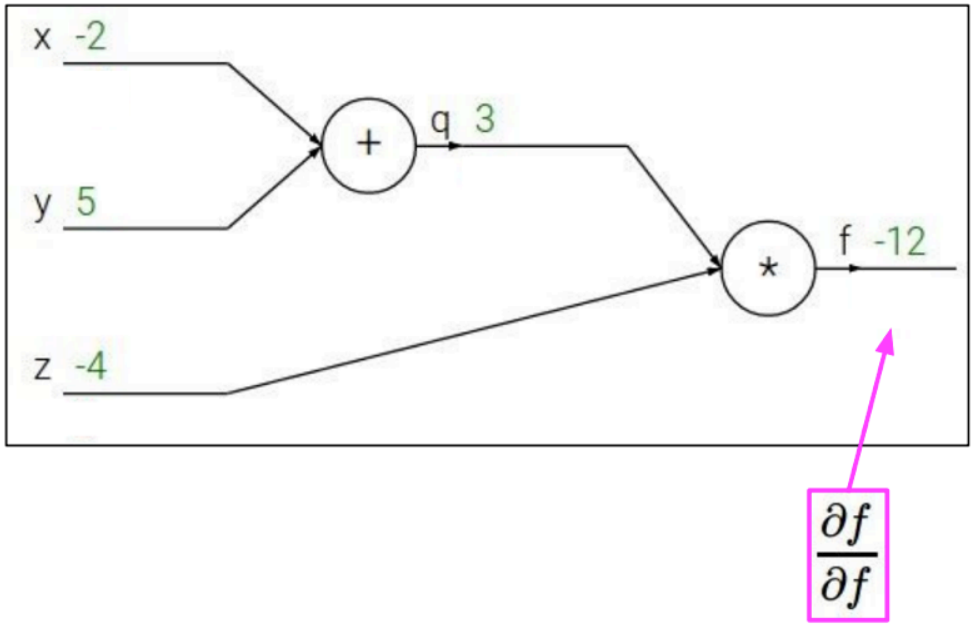
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



Recursive differentiation

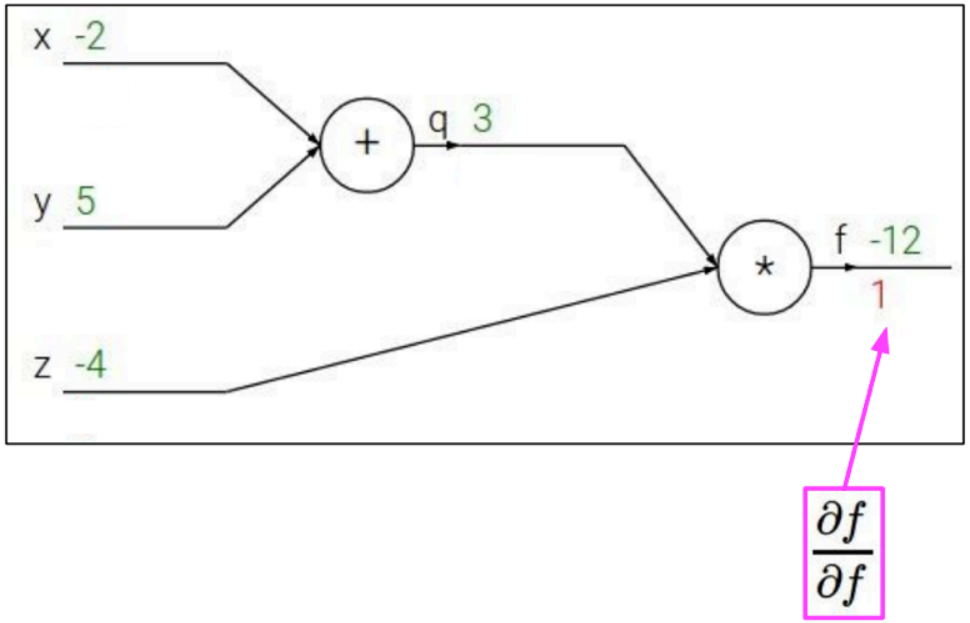
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



Recursive differentiation

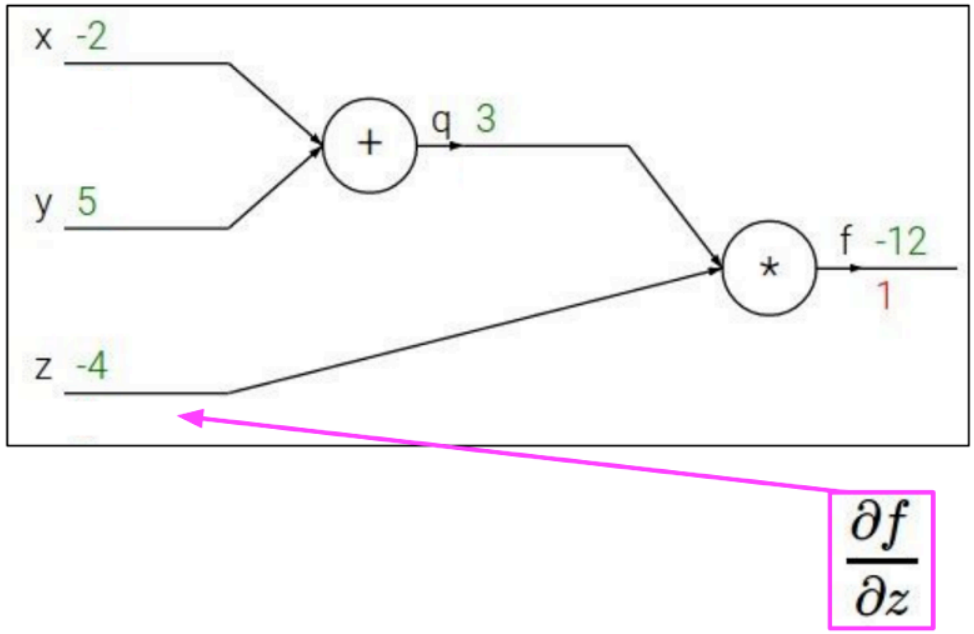
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



Recursive differentiation

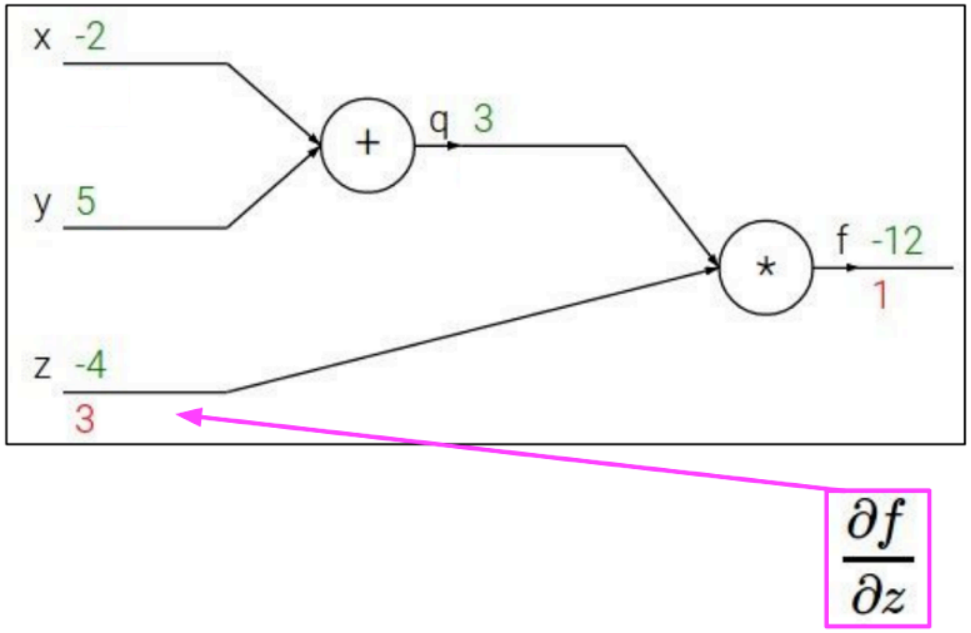
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



Recursive differentiation

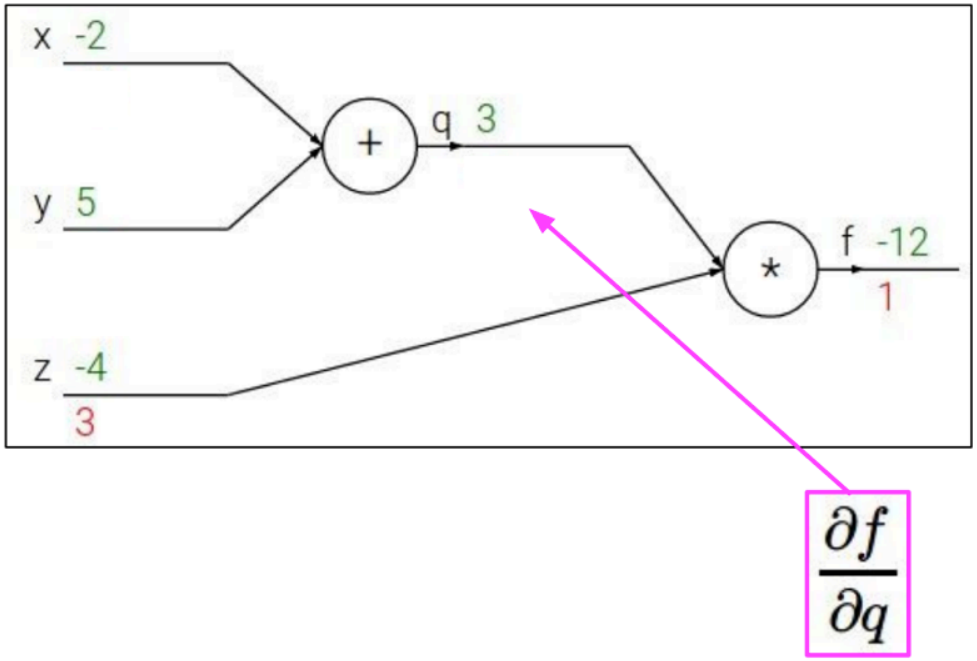
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



Recursive differentiation

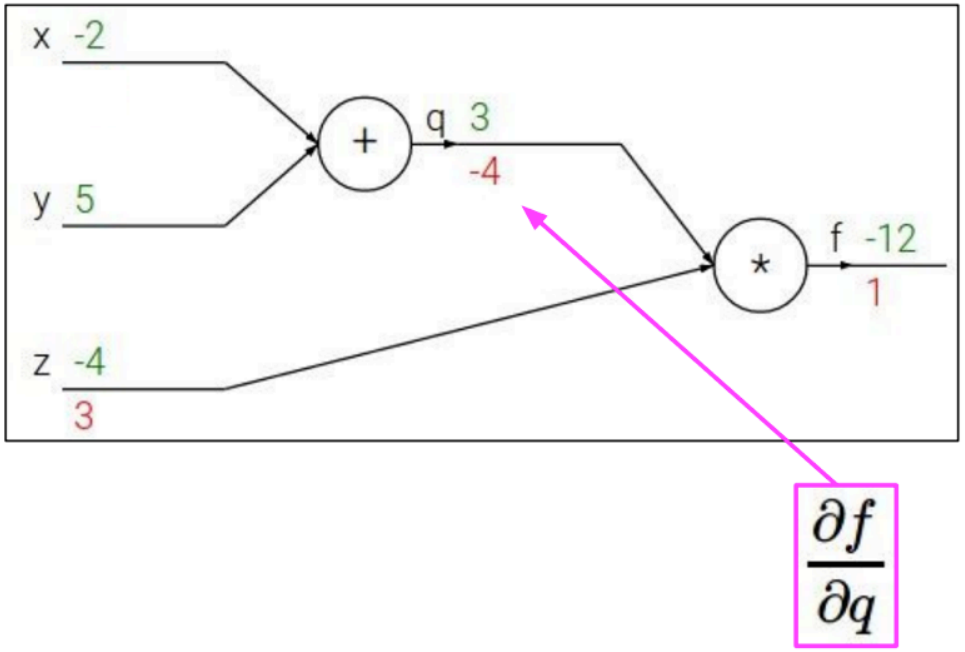
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



Recursive differentiation

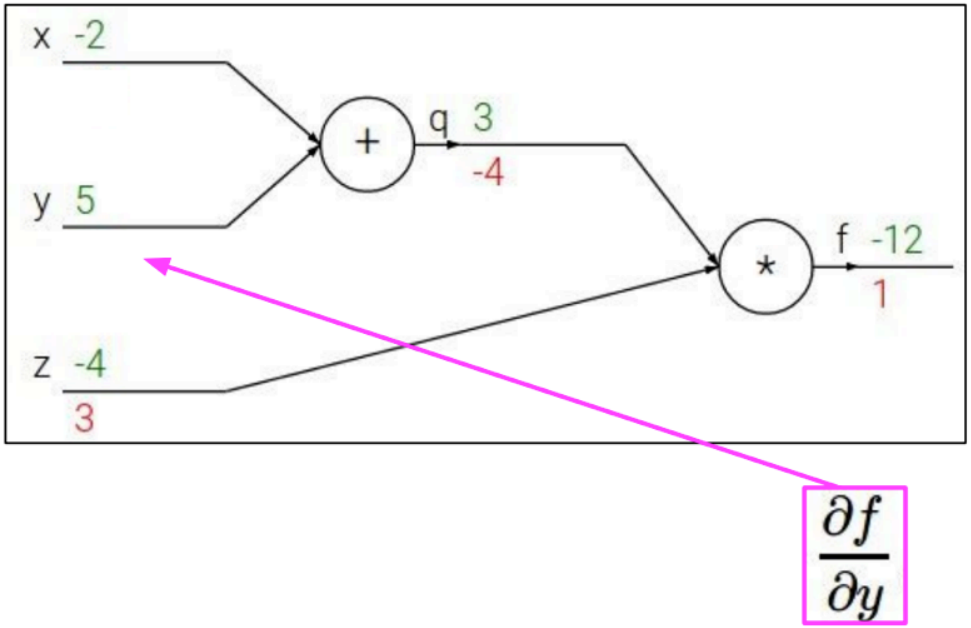
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



Recursive differentiation

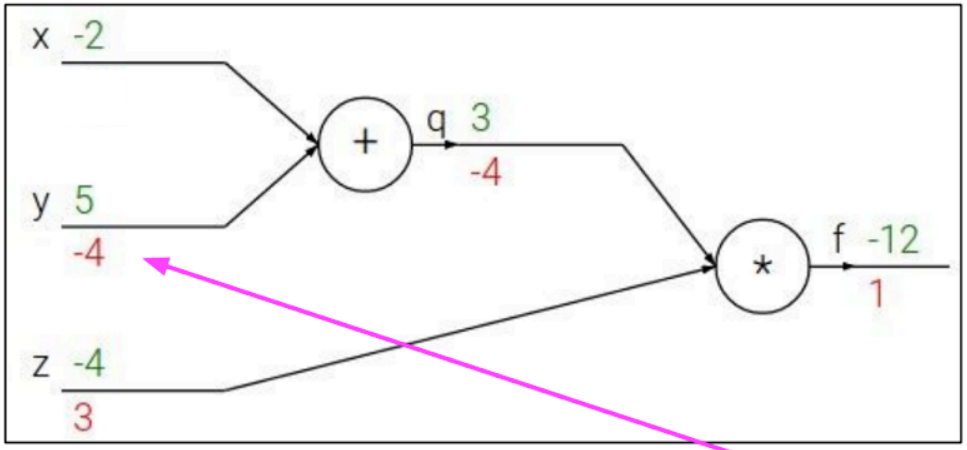
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



$\frac{\partial f}{\partial y}$

Chain rule:

$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial y}$$

Recursive differentiation

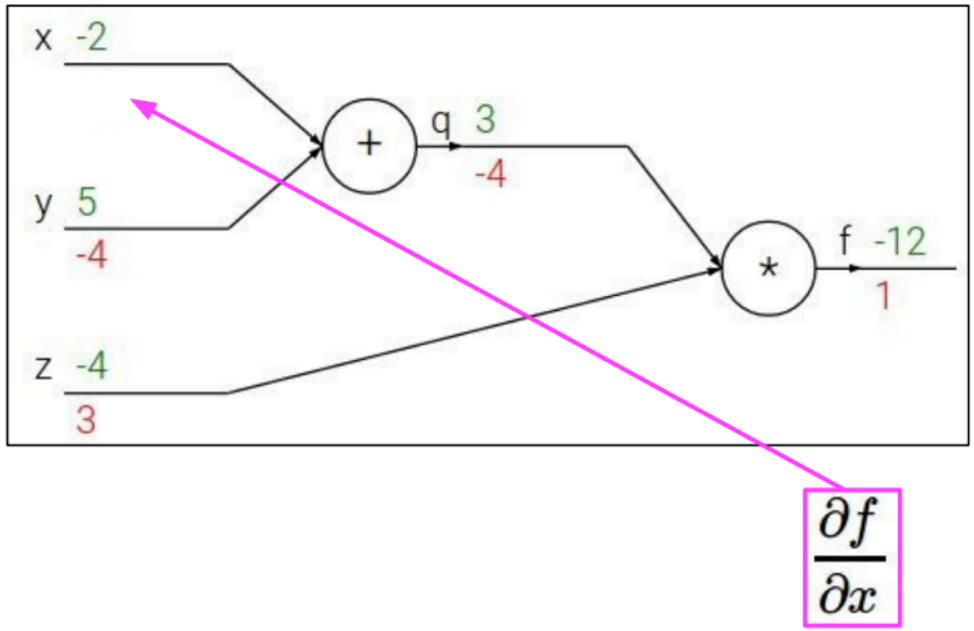
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

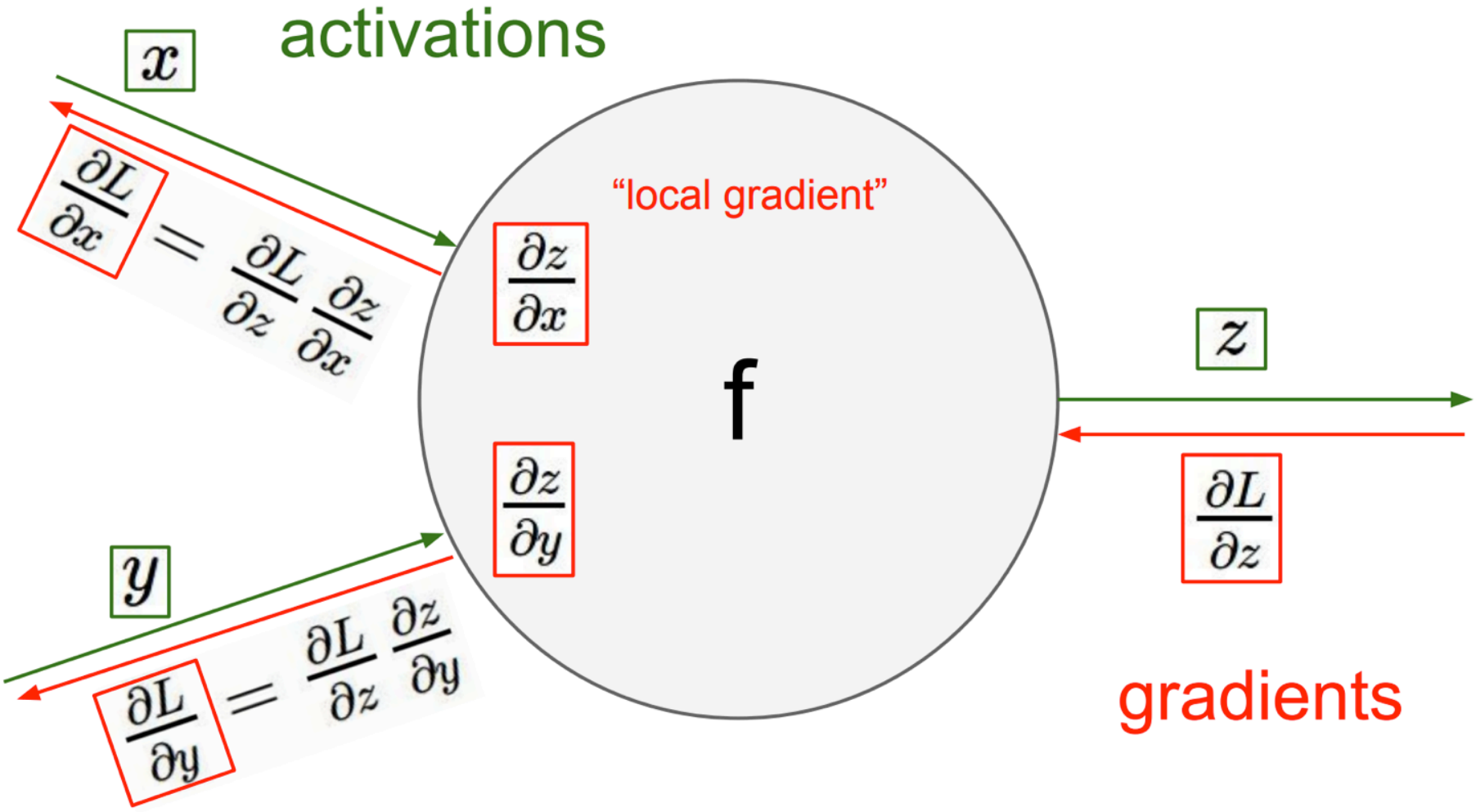
$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$

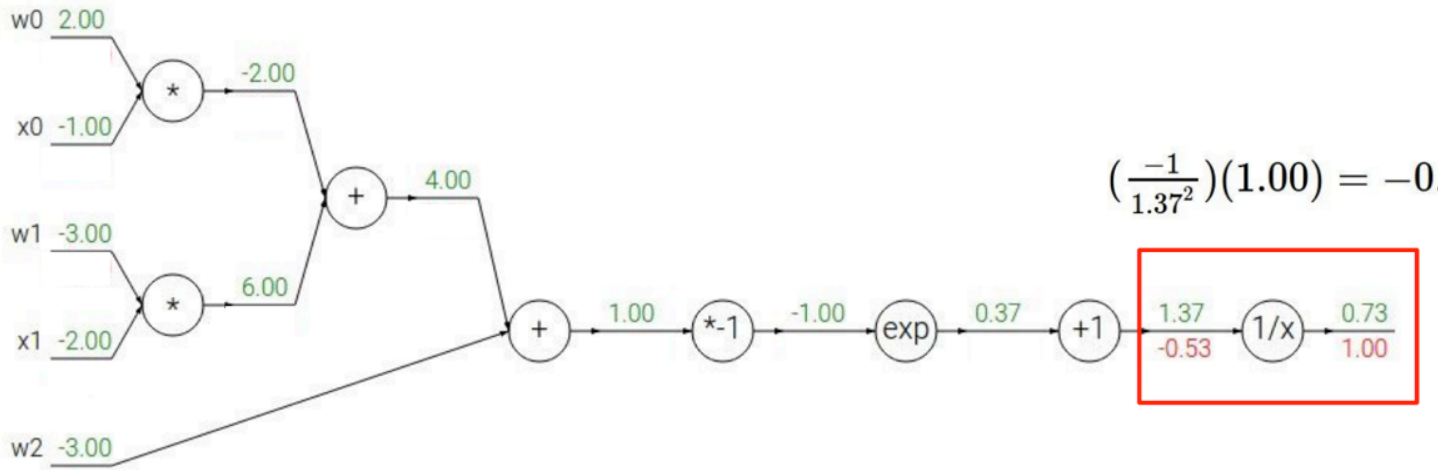


Node in Graph



Logistic Regression Computation Graph

Another example: $$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



$$(\frac{-1}{1.37^2})(1.00) = -0.53$$

$f(x) = e^x \rightarrow \frac{df}{dx} = e^x$

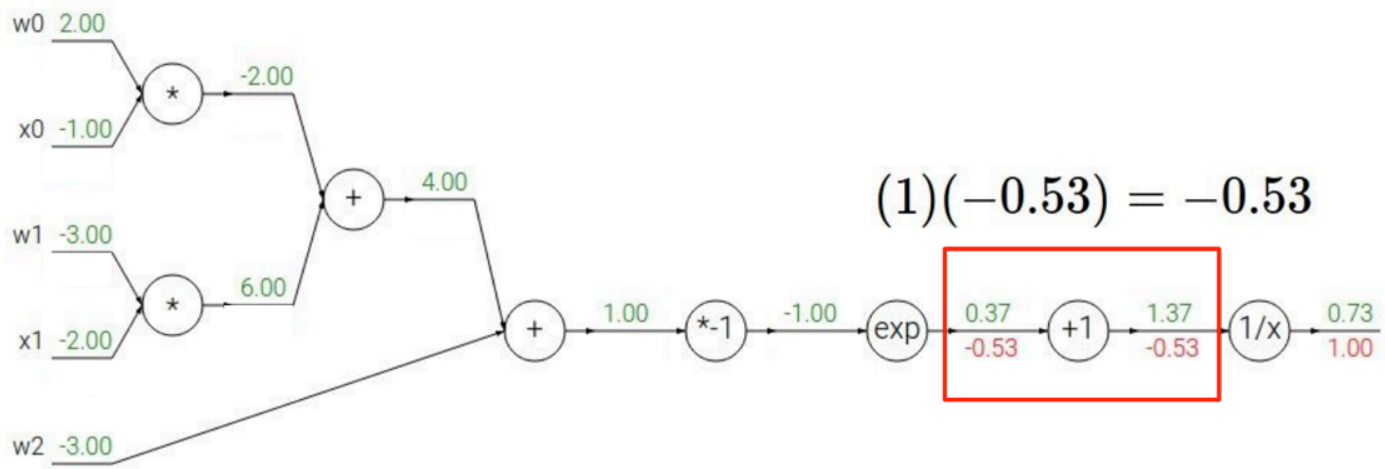
$f_a(x) = ax \rightarrow \frac{df}{dx} = a$

$f(x) = \frac{1}{x} \rightarrow \frac{df}{dx} = -1/x^2$

$f_c(x) = c + x \rightarrow \frac{df}{dx} = 1$

Logistic Regression Computation Graph

Another example: $f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$



$(1)(-0.53) = -0.53$

$f(x) = e^x \rightarrow \frac{df}{dx} = e^x$

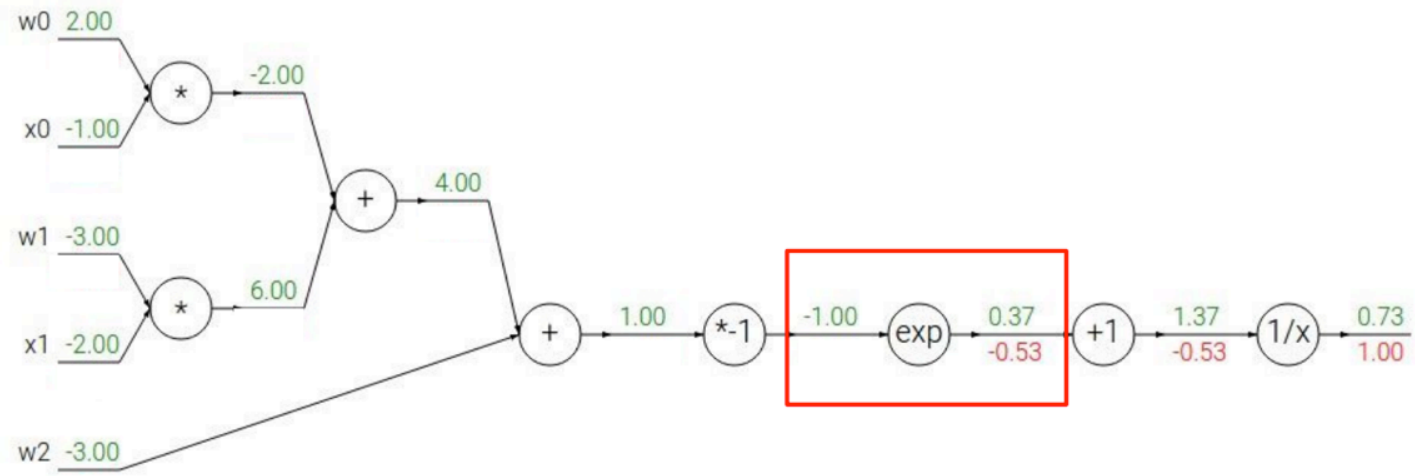
$f_a(x) = ax \rightarrow \frac{df}{dx} = a$

$f(x) = \frac{1}{x} \rightarrow \frac{df}{dx} = -1/x^2$

$f_c(x) = c + x \rightarrow \frac{df}{dx} = 1$

Logistic Regression Computation Graph

Another example: $$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



$$f(x) = e^x \rightarrow \frac{df}{dx} = e^x$$

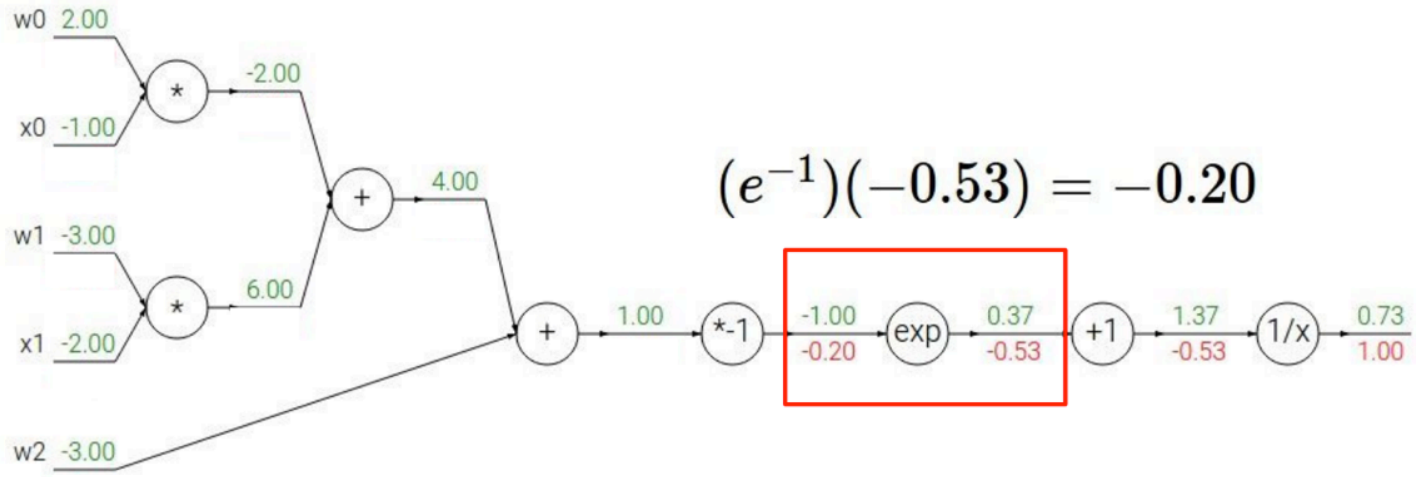
$$f_a(x) = ax \rightarrow \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \rightarrow \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \rightarrow \frac{df}{dx} = 1$$

Logistic Regression Computation Graph

Another example: $$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



$$(e^{-1})(-0.53) = -0.20$$

$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

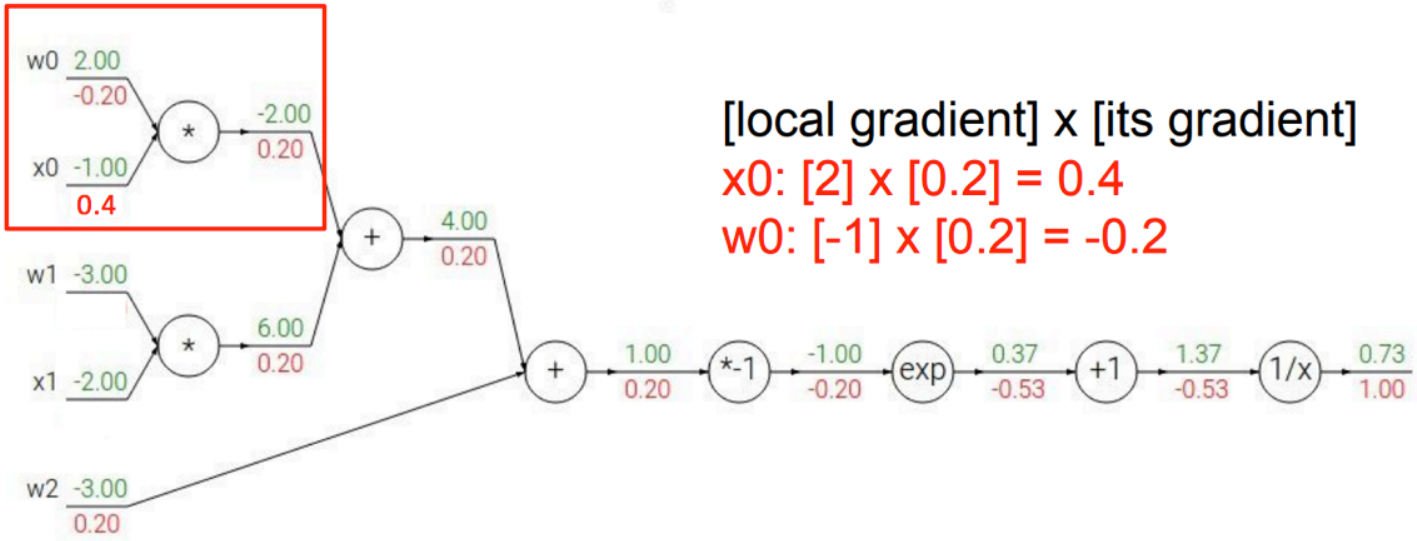
$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

Local Gradient

Another example: $$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



[local gradient] x [its gradient]
x0: [2] x [0.2] = 0.4
w0: [-1] x [0.2] = -0.2

$$f(x) = e^x \rightarrow \frac{df}{dx} = e^x$$

$$f_a(x) = ax \rightarrow \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \rightarrow \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \rightarrow \frac{df}{dx} = 1$$

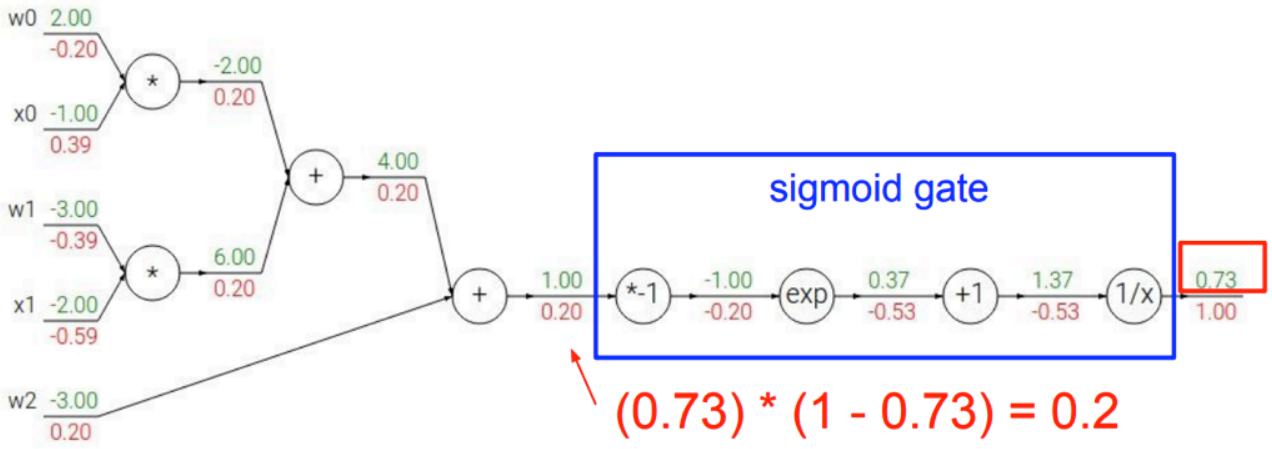
Skipping to the end

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$

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

sigmoid function

$$\frac{d\sigma(x)}{dx} = \frac{e^{-x}}{(1+e^{-x})^2} = \left(\frac{1+e^{-x}-1}{1+e^{-x}}\right)\left(\frac{1}{1+e^{-x}}\right) = (1-\sigma(x))\,\sigma(x)$$



Outline

- Failure of Perceptron
- Neural Network
- Backpropagation
- Code

Code Basics

- shape => the size of every dimension
  - i.e.
    a = np.array([[1, 2], [3, 4], [5, 7]])
    print a.shape => (3, 2)

- dimensions => total number of dimensions, as in 2D, 3D or nD.

Types of Nodes in Computation Graph (so far)

- Constants
- Variables
- Placeholders
- Unary Operations
- Binary Operations