UVA CS 6316: Machine Learning

Lecture 15: Neural Network / Deep Learning Basics

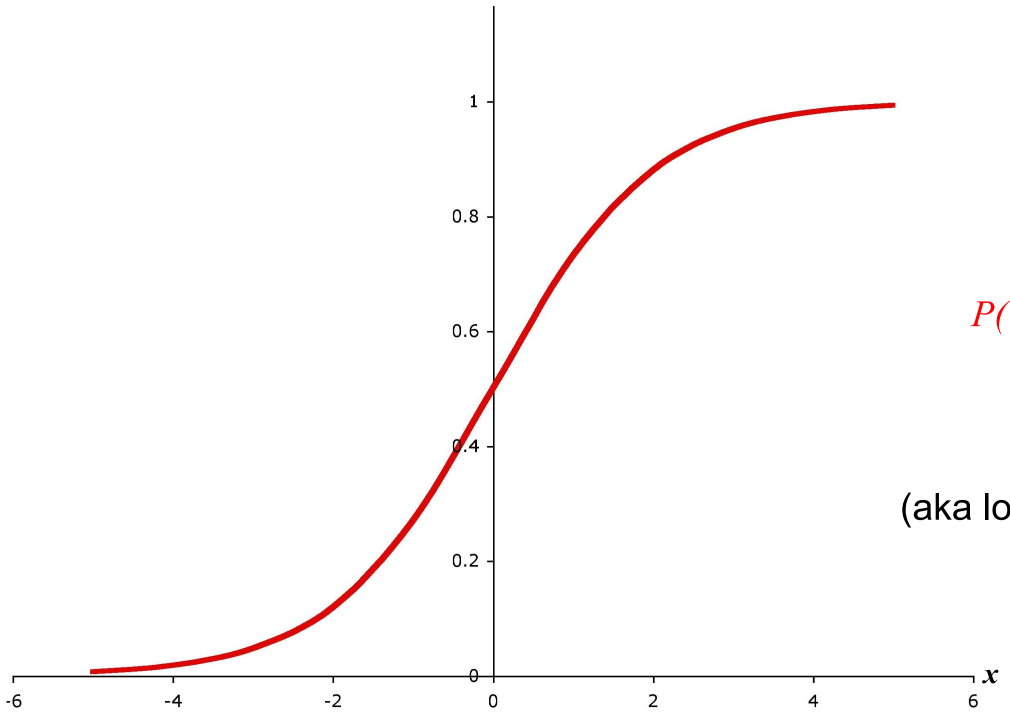
Intro to Neural Networks and Deep Learning

Jack Lanchantin, Dr. Yanjun Qi

University of Virginia

Neurons

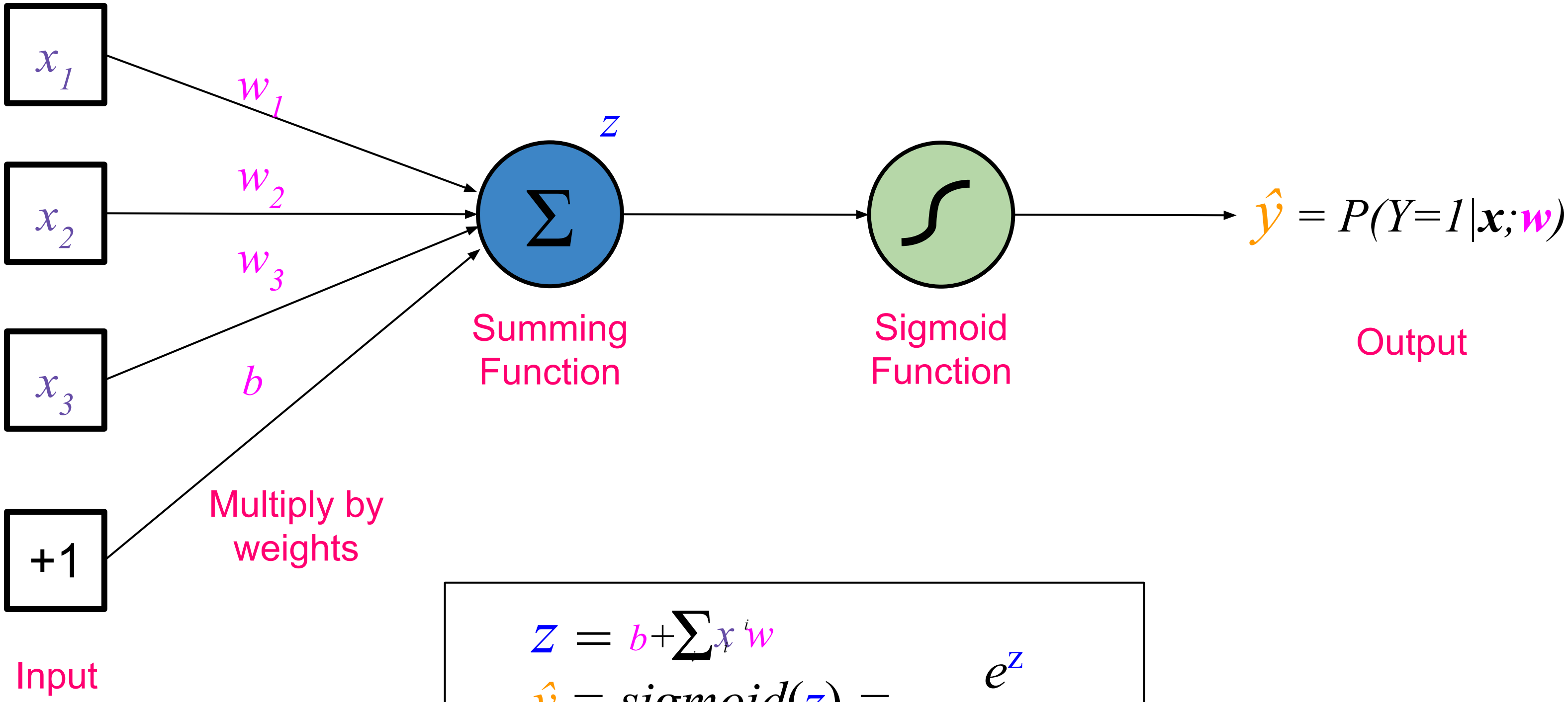
Logistic Regression



$$\color{red}{P(Y=1|\boldsymbol{x})} = \frac{e^{\boldsymbol{wx}+b}}{1+e^{\boldsymbol{wx}+b}}$$

Sigmoid Function
(aka logistic, logit, “S”, soft-step)

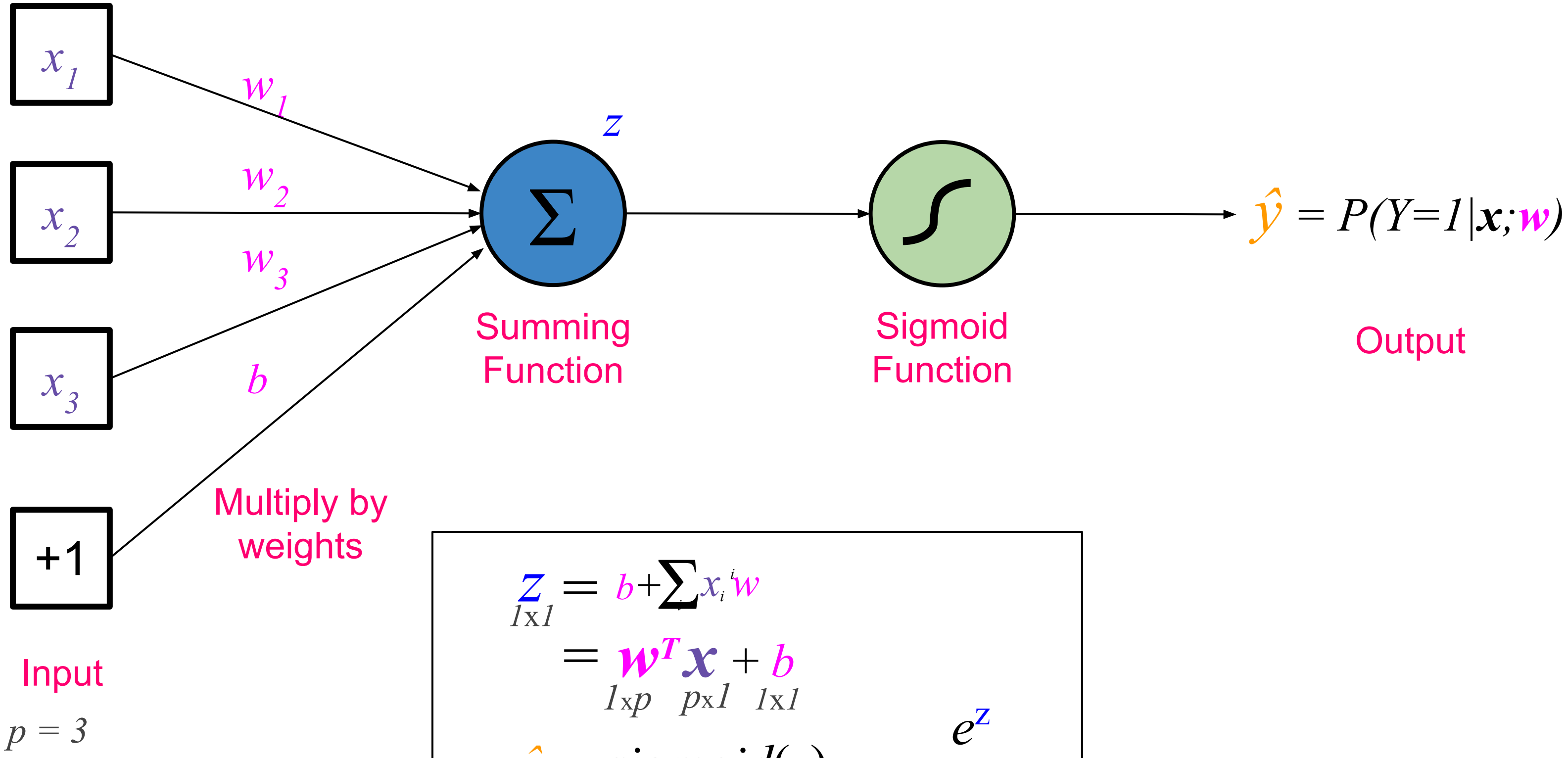
Logistic Regression



$$z = b + \sum x_i w_i$$

$$\hat{y} = sigmoid(z) = \frac{e^{z}}{1 + e^{z}}$$

Logistic Regression

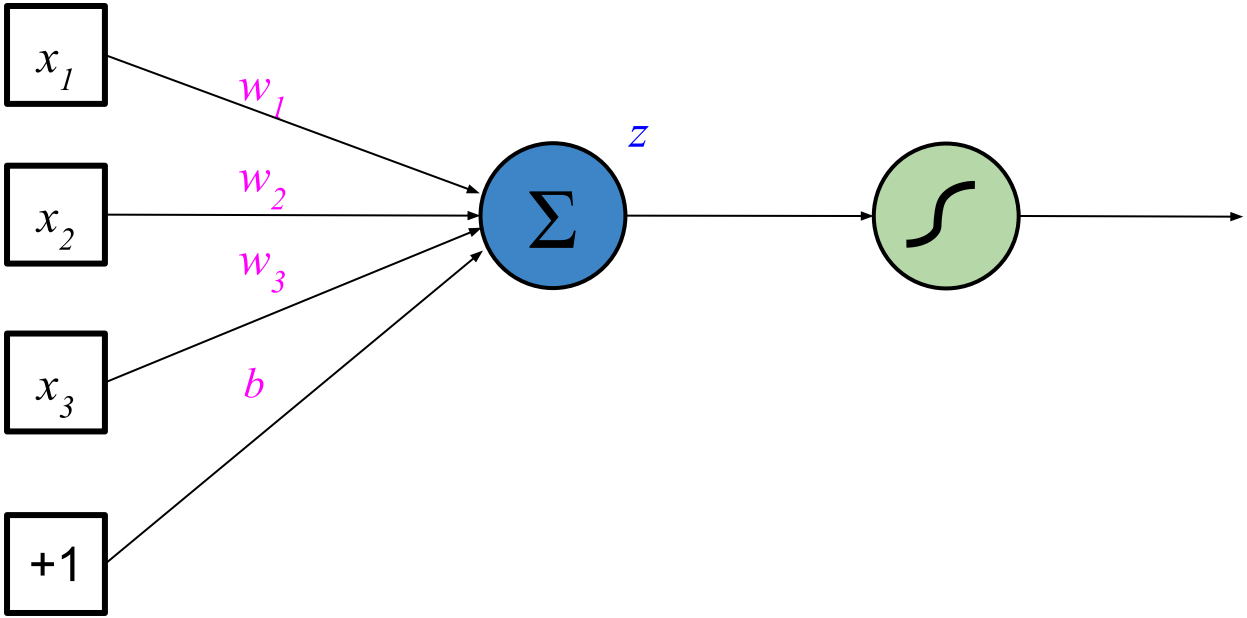


$$\underset{1\times1}{z} = b + \sum_i x_i w_i$$

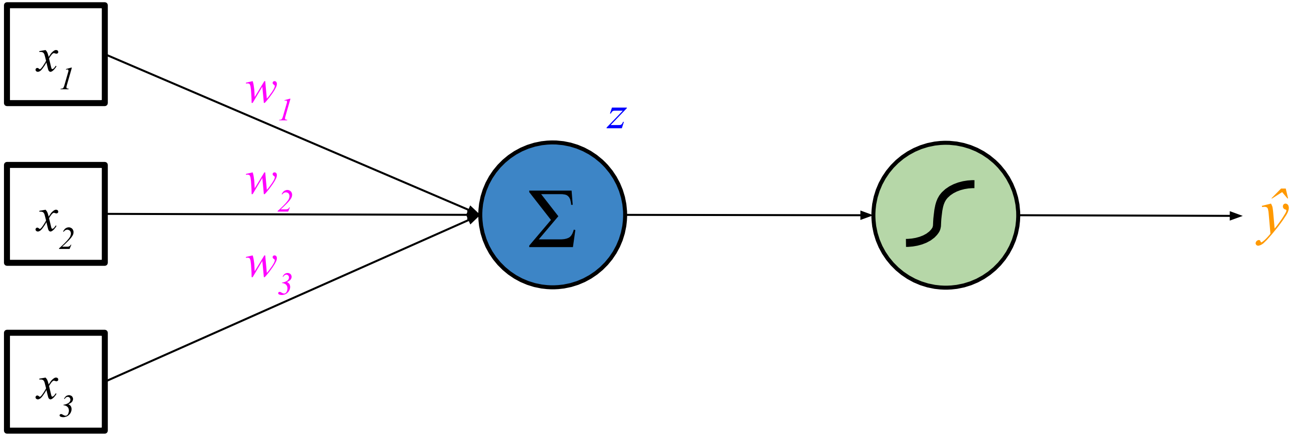
$$= \underset{1\times p}{\boldsymbol{w}^T}\,\underset{p\times1}{\boldsymbol{x}} + \underset{1\times1}{b}$$

$$\hat{y} = sigmoid(z) = \frac{e^z}{1 + e^z}$$

“Neuron” or “Perceptron”

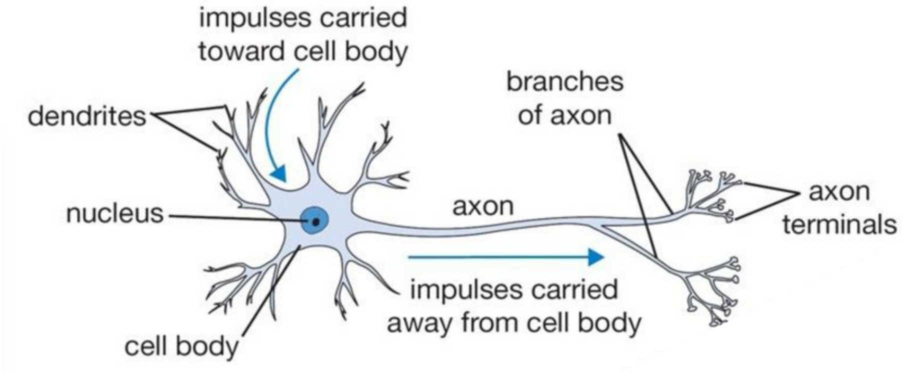


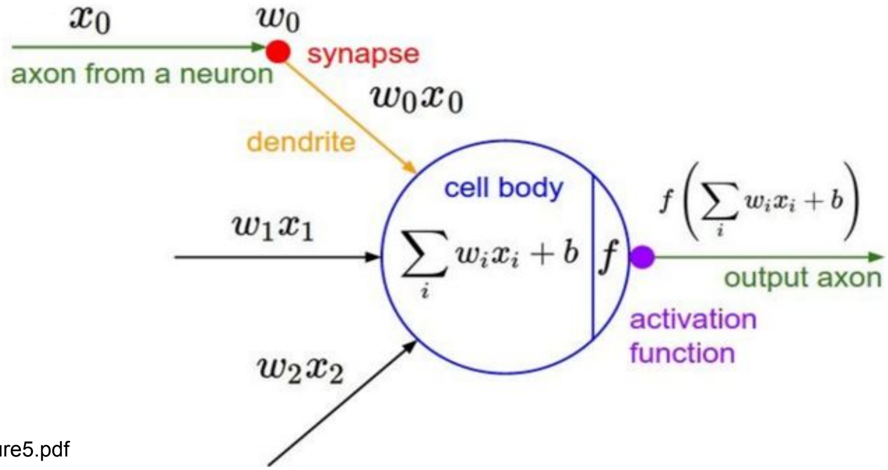
“Neuron” or “Perceptron”



From here on, we leave out bias for simplicity

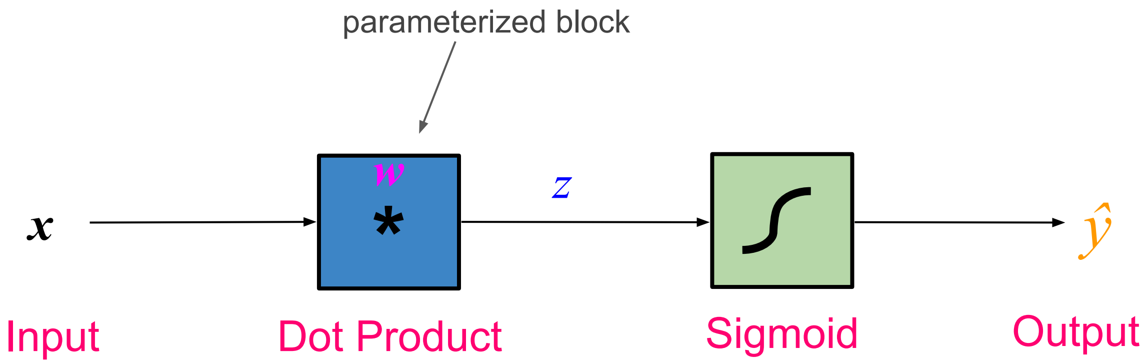
Neurons





http://cs231n.stanford.edu/slides/winter1516_lecture5.pdf

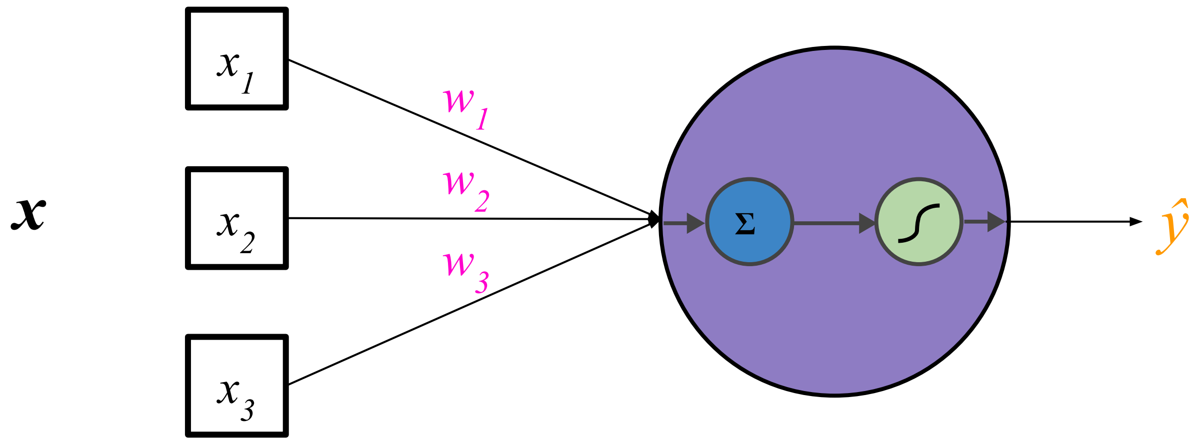
“Block View” of a Neuron



$$\underset{1\times 1}{z} = \underset{1\times p}{\boldsymbol{w}^T} \underset{p\times 1}{\boldsymbol{x}}$$

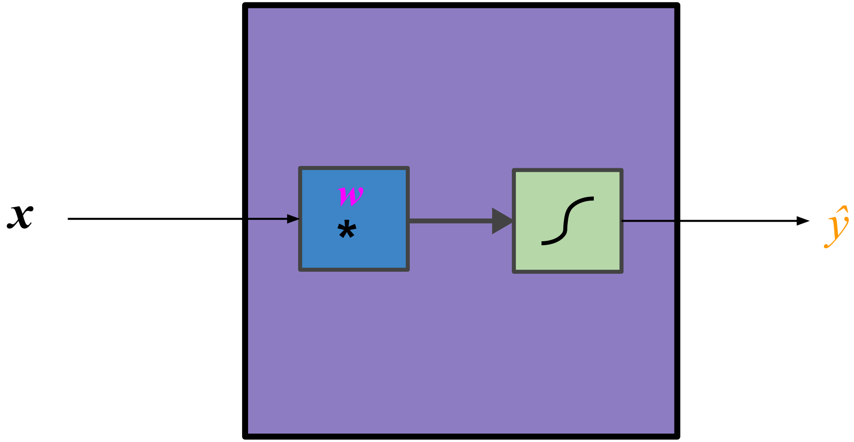
$$\hat{y} = sigmoid(z) = \frac{e^z}{1 + e^z}$$

Neuron Representation



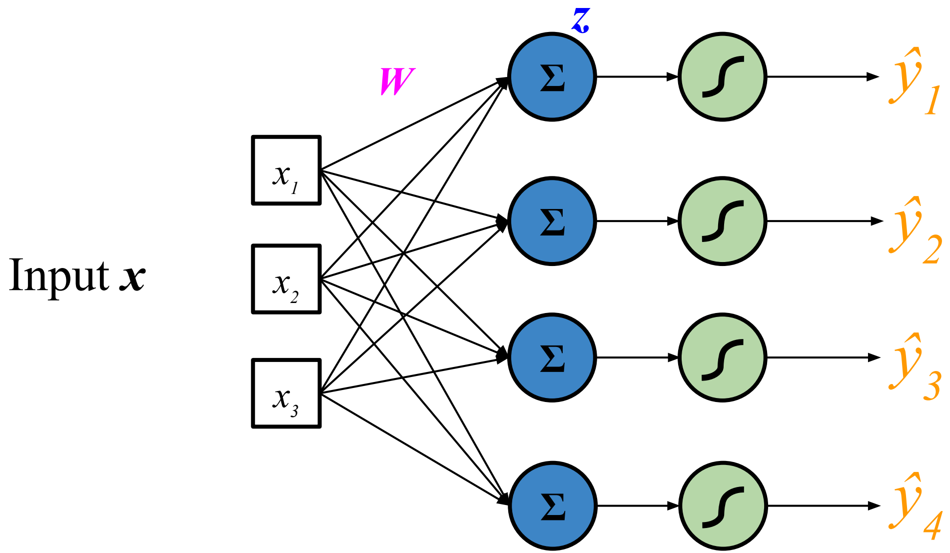
The linear transformation and nonlinearity together is typically considered a single neuron

Neuron Representation

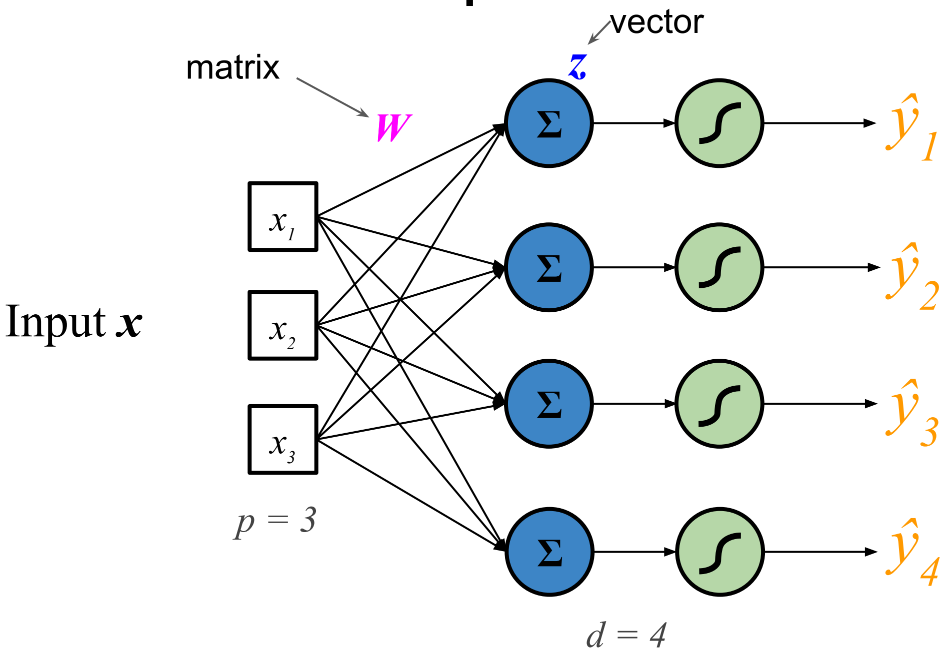


Neural Networks

Multiple Neurons

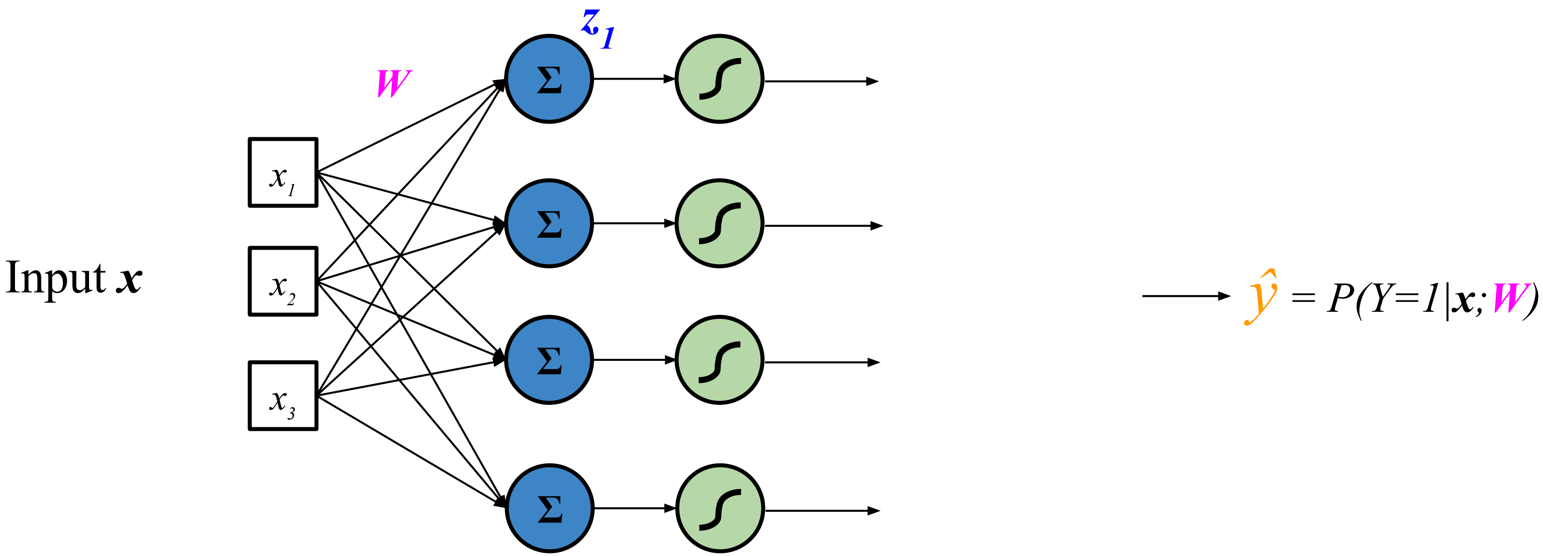


Multiple Neurons



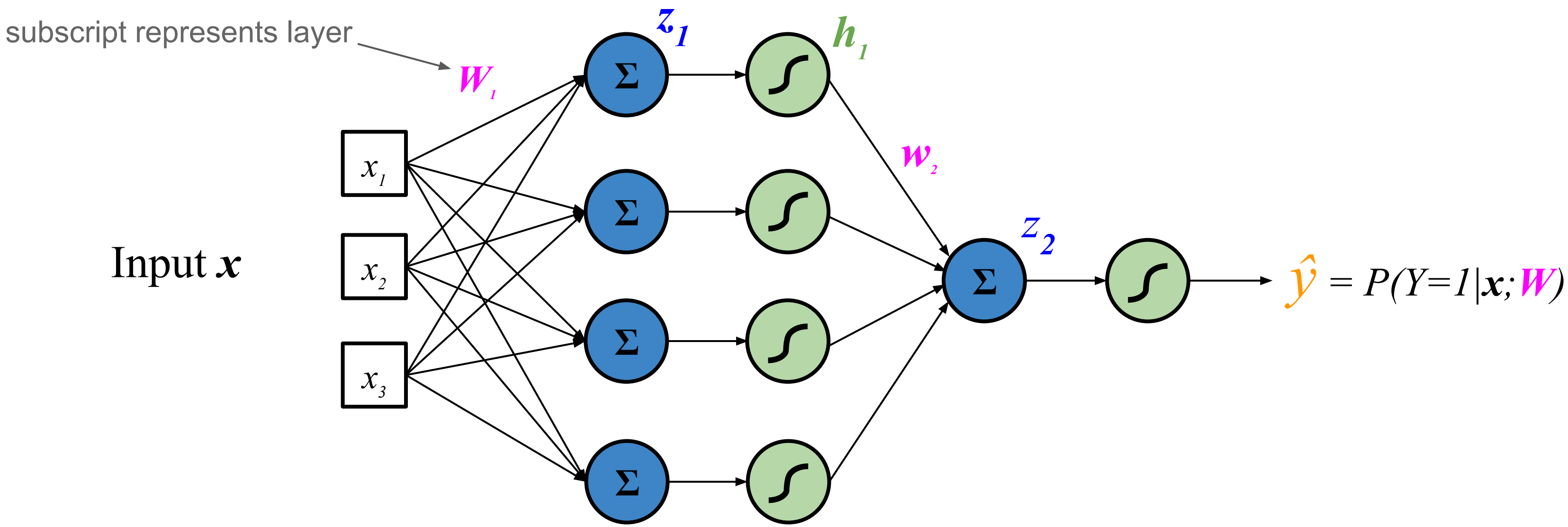
$$\underset{d\times 1}{z} = \underset{d\times p}{W^T} \underset{p\times 1}{x}$$

$$\underset{d\times 1}{\hat{y}} = sigmoid(\underset{d\times 1}{z}) = \frac{e^z}{1 + e^z}$$

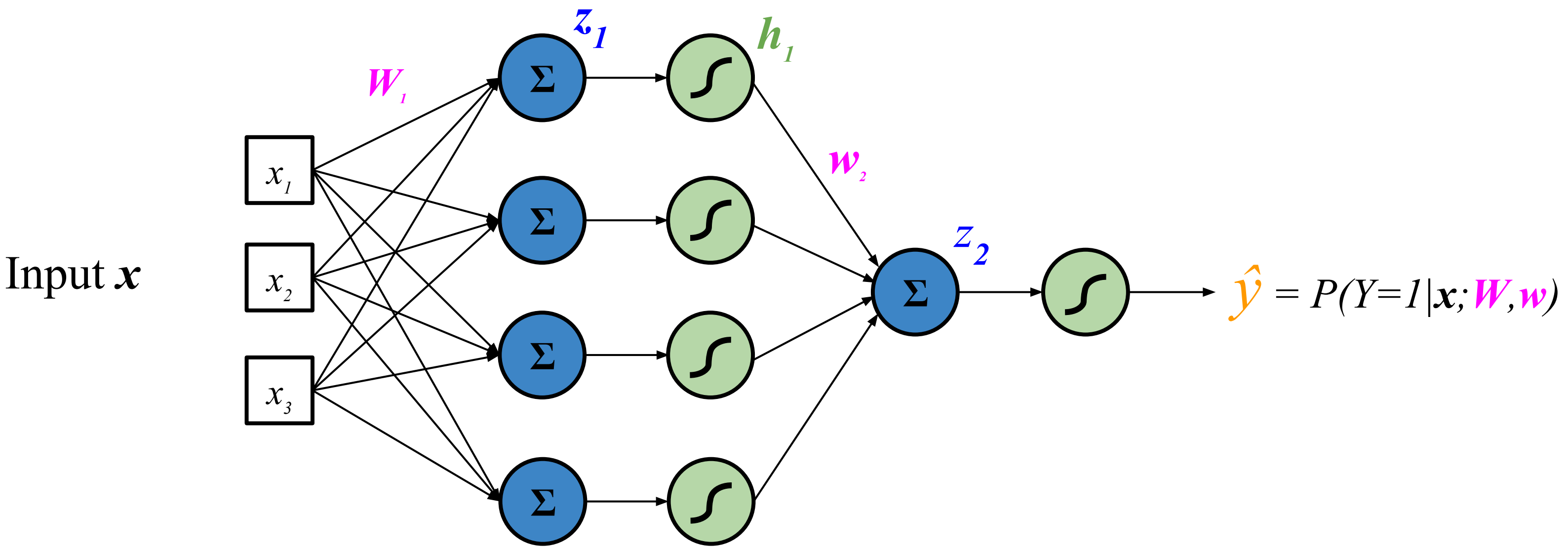


Input $\boldsymbol{x}$
x_1
x_2
x_3
W
z_1
$\hat{y} = P(Y=1|\mathbf{x};\boldsymbol{W})$

Neural Network (1 Hidden Layer)



Neural Network (1 Hidden Layer)



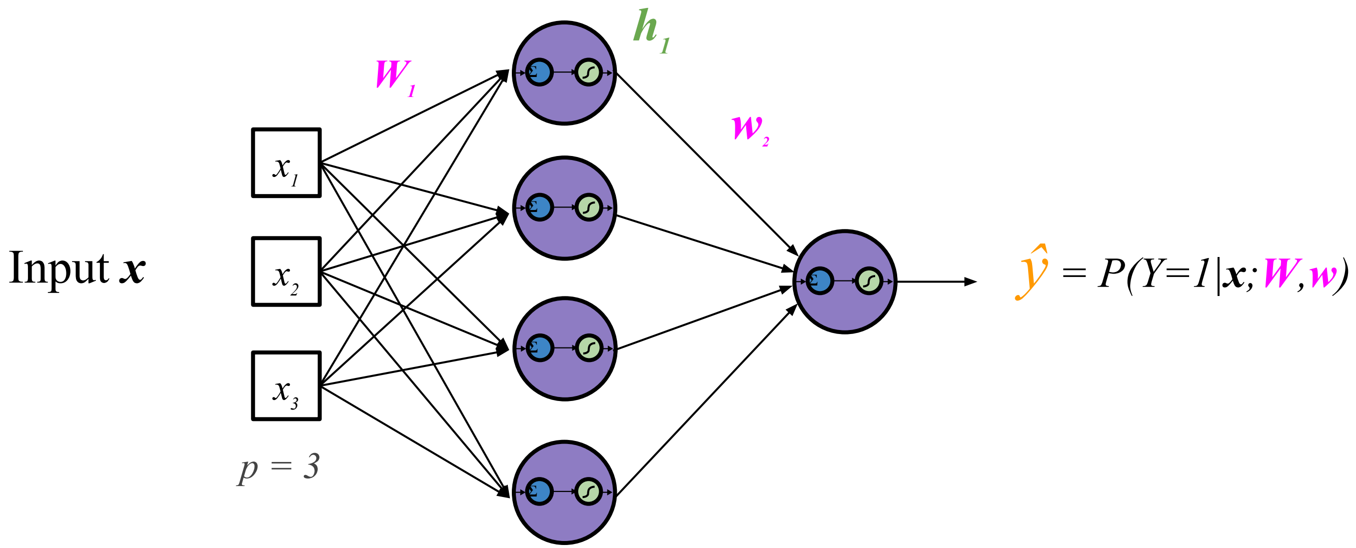
$$\underset{d\times 1}{z_1} = \underset{d\times p}{W_1^T}\ \underset{p\times 1}{x}$$

$$\underset{d\times 1}{h_1} = sigmoid(\underset{d\times 1}{z_1})$$

$$\underset{1\times 1}{z_2} = \underset{1\times p}{w_2^T}\ \underset{p\times 1}{h_1}$$

$$\underset{1\times 1}{\hat{y}} = sigmoid(\underset{1\times 1}{z_2})$$

Neural Network (1 Hidden Layer)



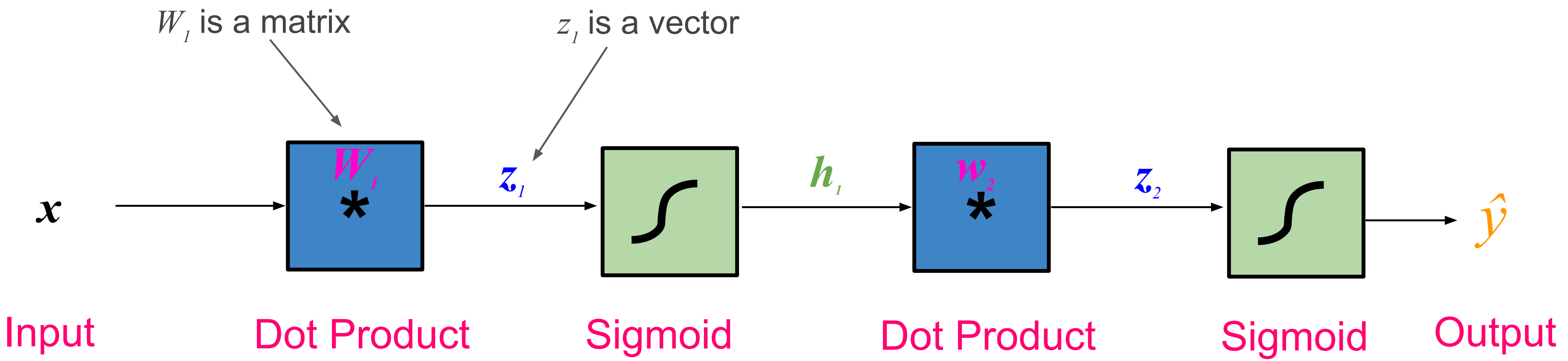
$$\underset{dx1}{\boldsymbol{z}_1} = \underset{dxp}{\boldsymbol{W}_1^T}\ \underset{px1}{\boldsymbol{x}}$$

$$\underset{dx1}{\boldsymbol{h}_1} = sigmoid(\underset{dx1}{\boldsymbol{z}_1})$$

$$\underset{1x1}{z_2} = \underset{1xp}{\boldsymbol{w}_2^T}\ \underset{px1}{\boldsymbol{h}_1}$$

$$\underset{1x1}{\hat{y}} = sigmoid(\underset{1x1}{z_2})$$

Neural Network (1 Hidden Layer)



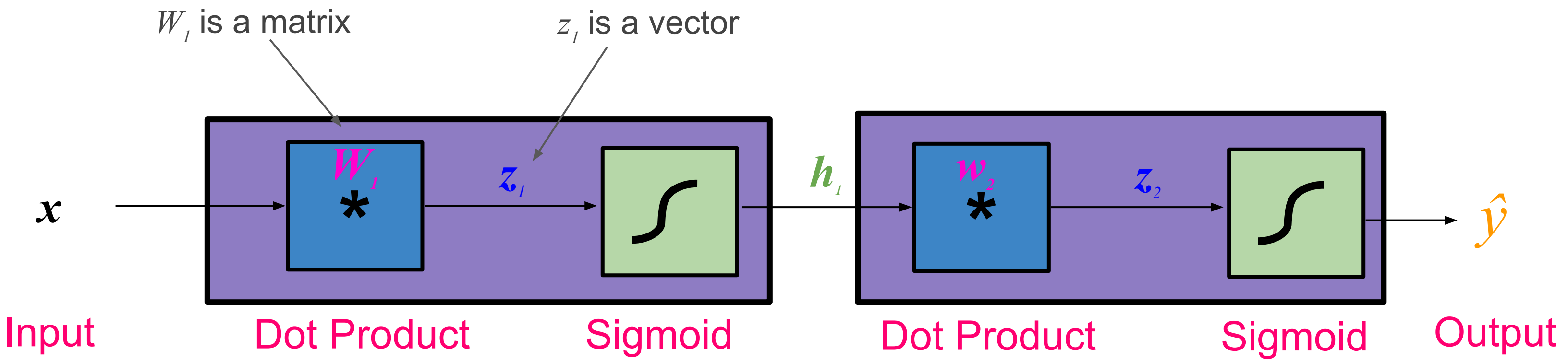
$$\underset{dx1}{z_1} = \underset{dxp}{W_1^T}\ \underset{px1}{x}$$

$$\underset{dx1}{h_1} = sigmoid(\underset{dx1}{z_1})$$

$$\underset{1x1}{z_2} = \underset{1xp}{w_2^T}\ \underset{px1}{h_1}$$

$$\underset{1x1}{\hat{y}} = sigmoid(\underset{1x1}{z_2})$$

Neural Network (1 Hidden Layer)



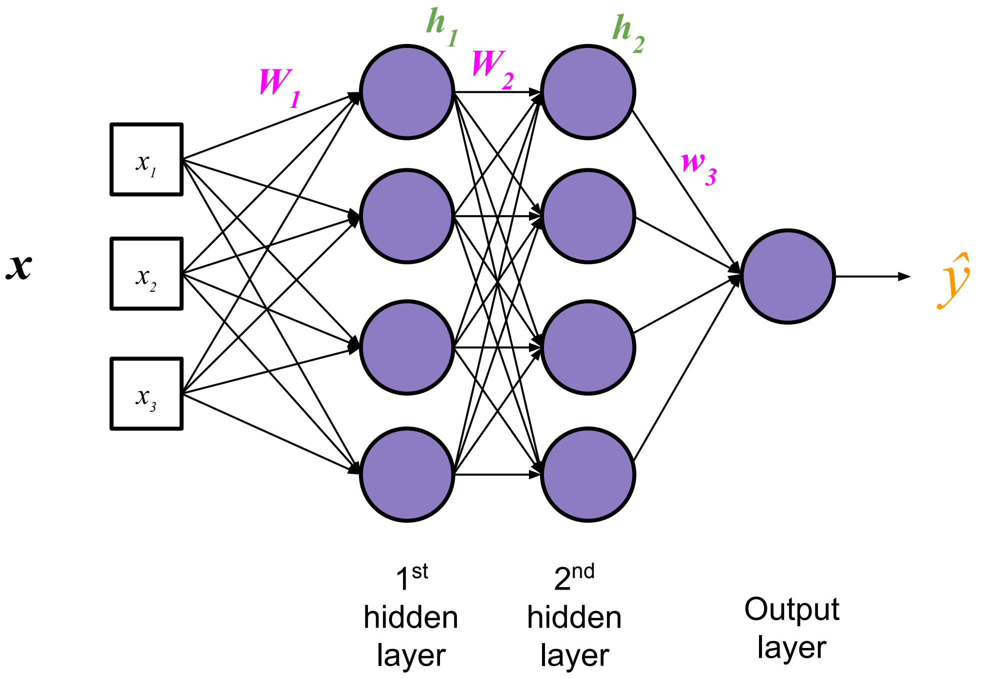
$$\underset{d\times 1}{z_1} = \underset{d\times p}{W_1}^T \underset{p\times 1}{x}$$

$$\underset{d\times 1}{h_1} = sigmoid(\underset{d\times 1}{z_1})$$

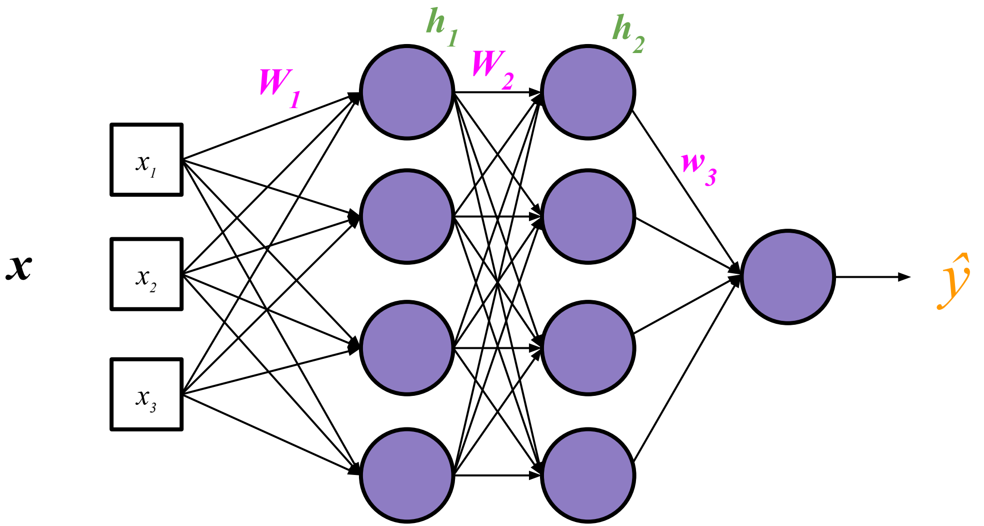
$$\underset{1\times 1}{z_2} = \underset{1\times p}{w_2}^T \underset{p\times 1}{h_1}$$

$$\underset{1\times 1}{\hat{y}} = sigmoid(\underset{1\times 1}{z_2})$$

Neural Network (2 Hidden Layers)



Neural Network (2 Hidden Layers)



$$\boldsymbol{z_1} = \boldsymbol{W_1^T x}$$

hidden layer 1 output → $$\boldsymbol{h_1} = sigmoid(\boldsymbol{z_1})$$

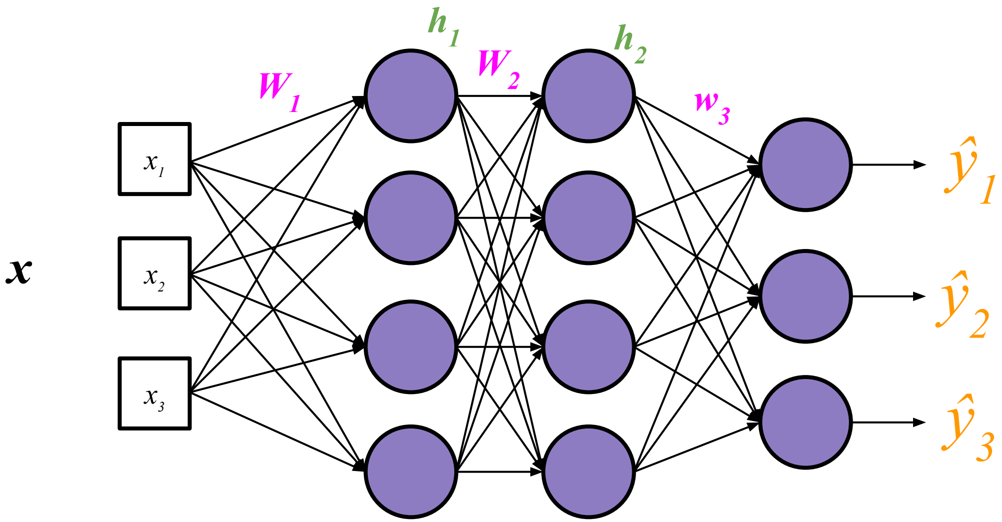
$$\boldsymbol{z_2} = \boldsymbol{W_2^T h_1}$$

hidden layer 2 output → $$\boldsymbol{h_2} = sigmoid(\boldsymbol{z_2})$$

$$z_3 = \boldsymbol{w_3^T h_2}$$

$$\hat{y} = sigmoid(z_3)$$

Multi-Class Neural Network (2 Hidden Layers)



$$z_1 = W_1^T x$$

hidden layer 1 output → $h_1 = sigmoid(z_1)$

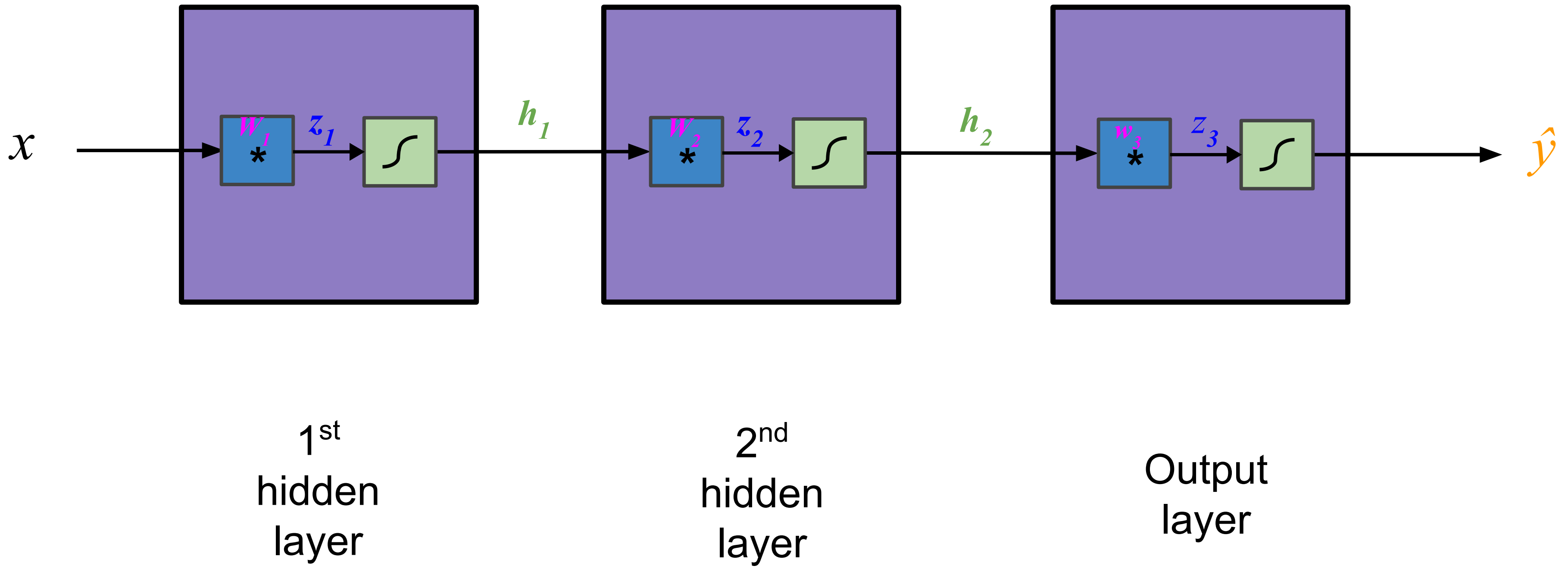
$$z_2 = W_2^T h_1$$

hidden layer 2 output → $h_2 = sigmoid(z_2)$

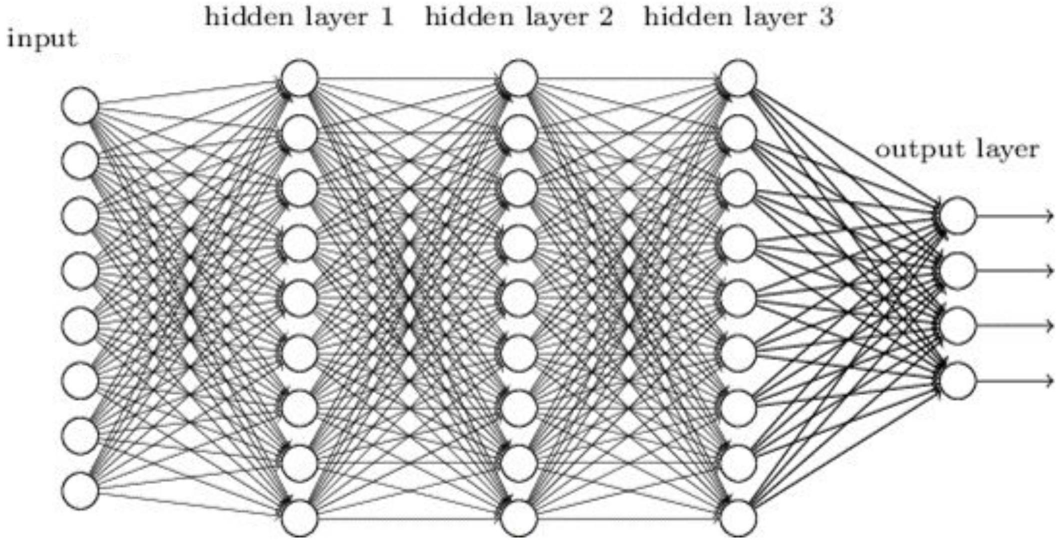
$$z_3 = w_3^T h_2$$

$$\hat{y} = sigmoid(z_3)$$

“Block View” Of Multi-Layer Neural Network



“Deep” Neural Networks (i.e. > 1 hidden layer)



Loss Functions

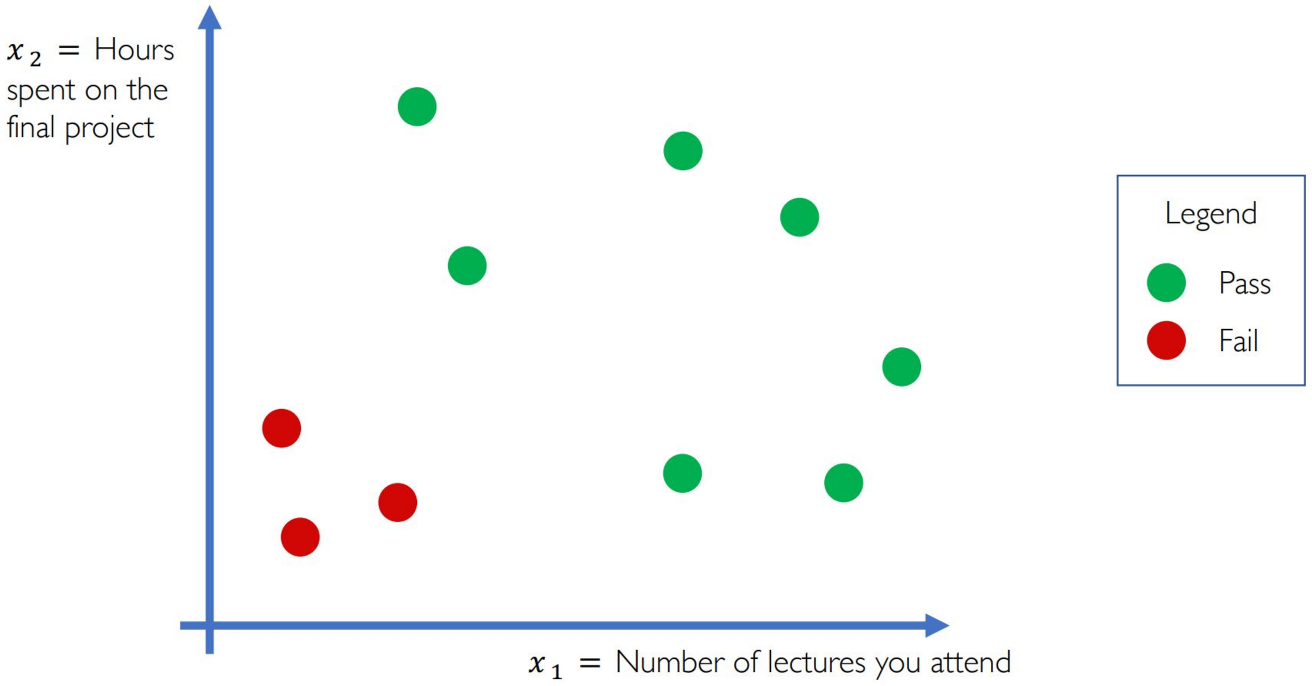
Toy Example: Will I pass this class?

Let's start with a simple two feature model

x_1 = Number of lectures you attend

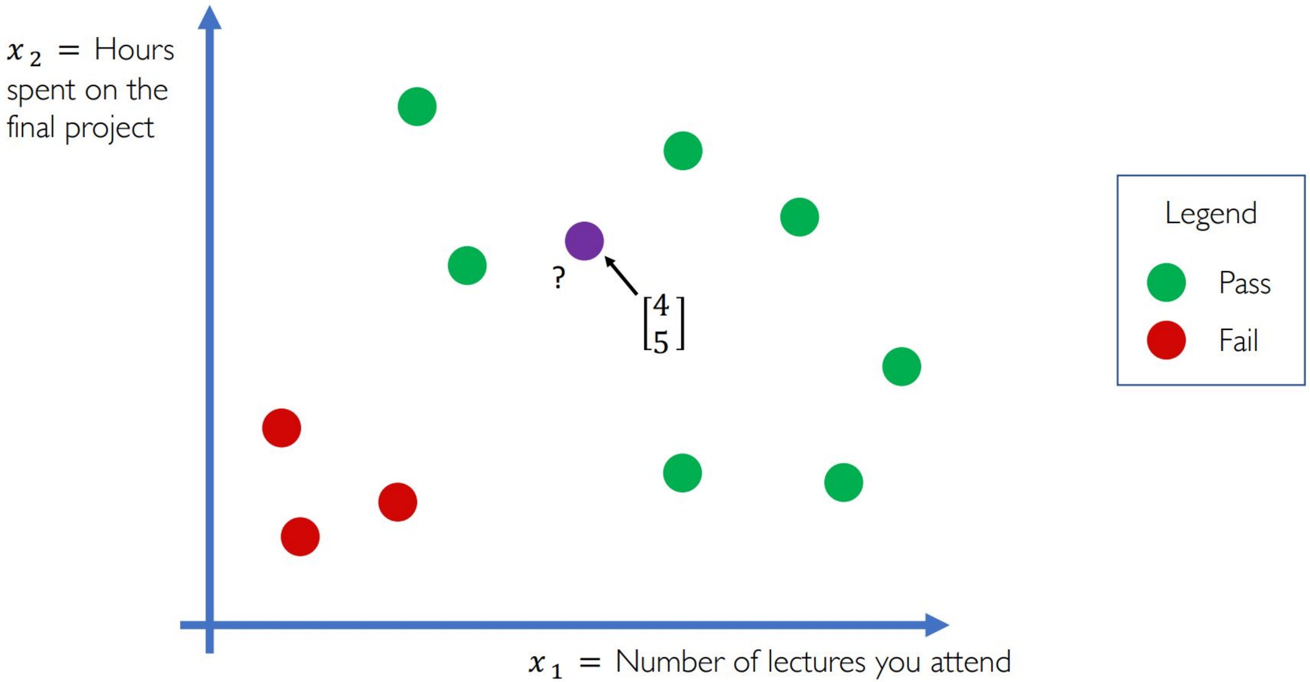
x_2 = Hours spent on the final project

Toy Example: Will I pass this class?



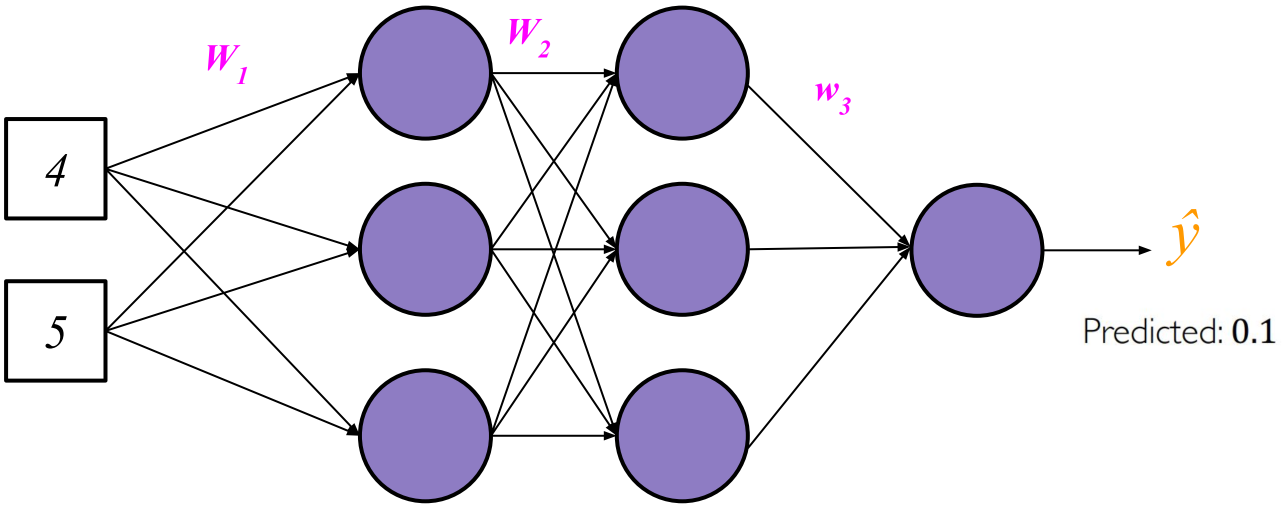
http://introtodeeplearning.com/2019/materials/2019_6S191_L1.pdf

Toy Example: Will I pass this class?



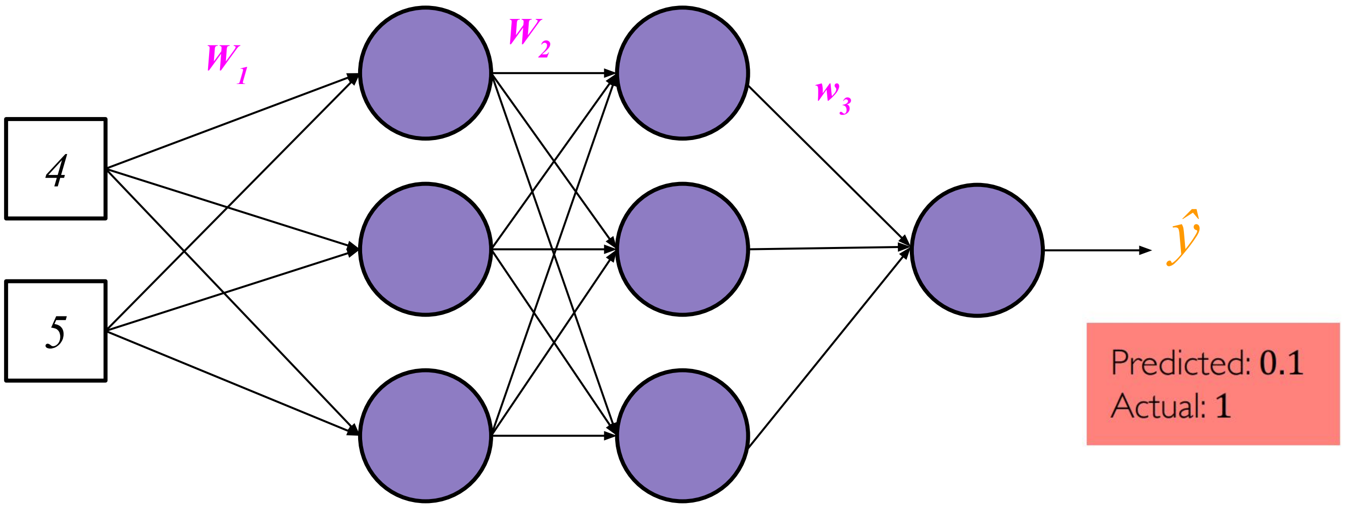
http://introtodeeplearning.com/2019/materials/2019_6S191_L1.pdf

Toy Example: Will I pass this class?



http://introtodeeplearning.com/2019/materials/2019_6S191_L1.pdf

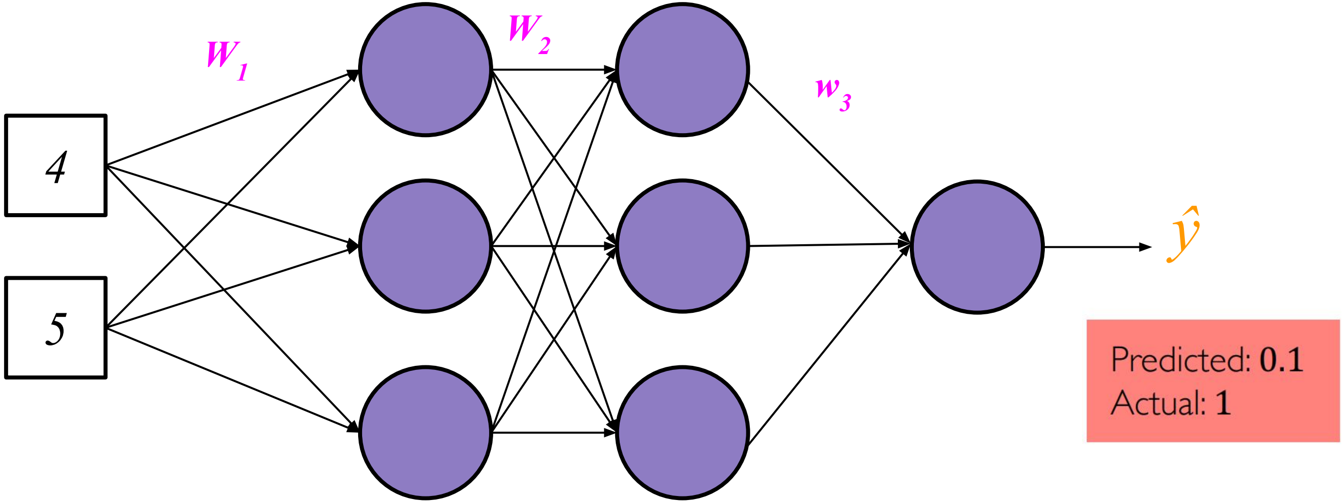
Toy Example: Will I pass this class?



http://introtodeeplearning.com/2019/materials/2019_6S191_L1.pdf

Toy Example: Will I pass this class?

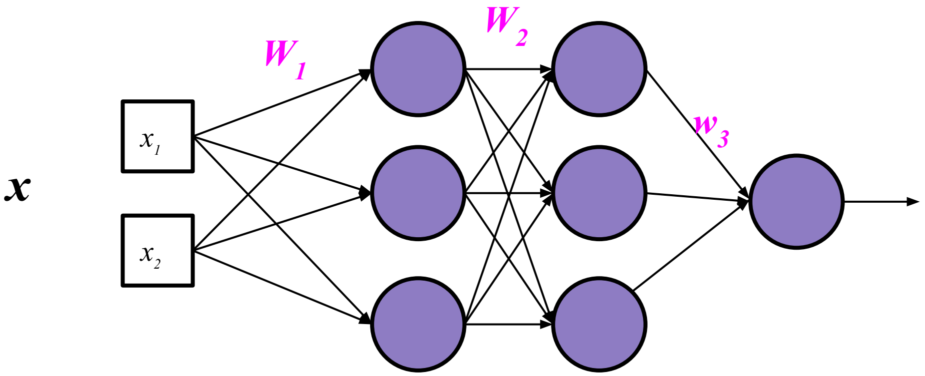
*The **loss** of our network measures the cost incurred from incorrect predictions*



$$\mathcal{L}\Big(\underbrace{f\big(x^{(i)}; \boldsymbol{W}\big)}_{\text{Predicted}}, \underbrace{y^{(i)}}_{\text{Actual}}\Big)$$

Binary Classification Loss

Binary Cross Entropy Loss



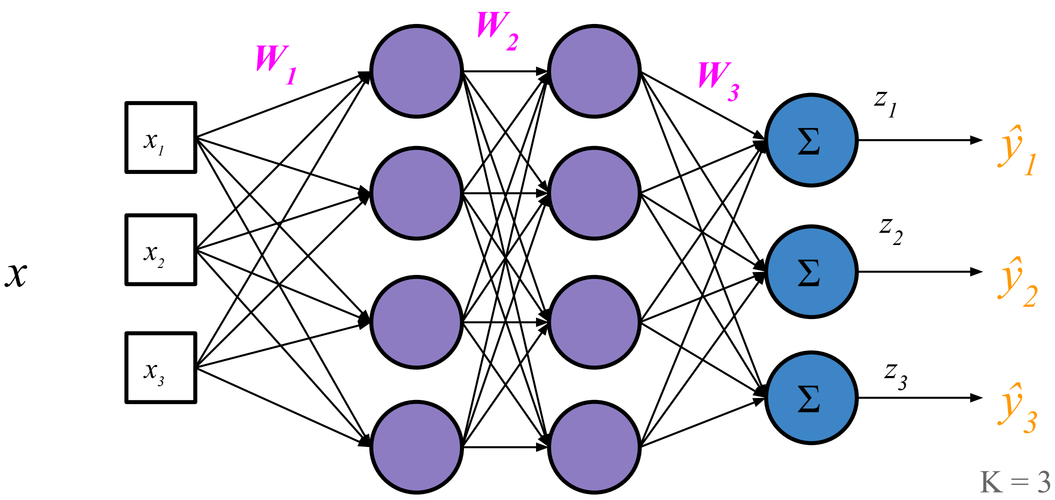
$\hat{y} = P(Y=1|X,W)$

$$
\begin{aligned}
E = \text{loss} &= -\log P(\mathrm{Y} = \hat{y} \mid \mathbf{X} = \boldsymbol{x}) \\
&= -y\log(\hat{y}) - (1-y)\log(1-\hat{y})
\end{aligned}
$$

true output

Multi-Class Classification Loss

Cross Entropy Loss



$$\hat{y}_i = \frac{e^{z_i}}{\sum_j e^{z_j}} = P(y_i = 1 \mid \boldsymbol{x})$$

"Softmax" function.
Normalizing function which converts each class output to a probability.

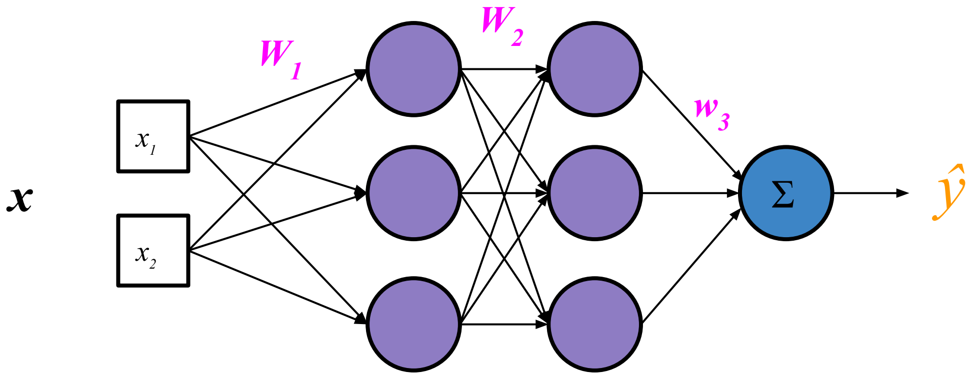
$$\boldsymbol{y} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad \hat{\boldsymbol{y}} = \begin{pmatrix} 0.1 \\ 0.7 \\ 0.2 \end{pmatrix}$$

$$\boxed{E = \text{loss} = -\sum_{j=1...K} y_j \log \hat{y}_j}$$

"0" for all except true class

Regression Loss

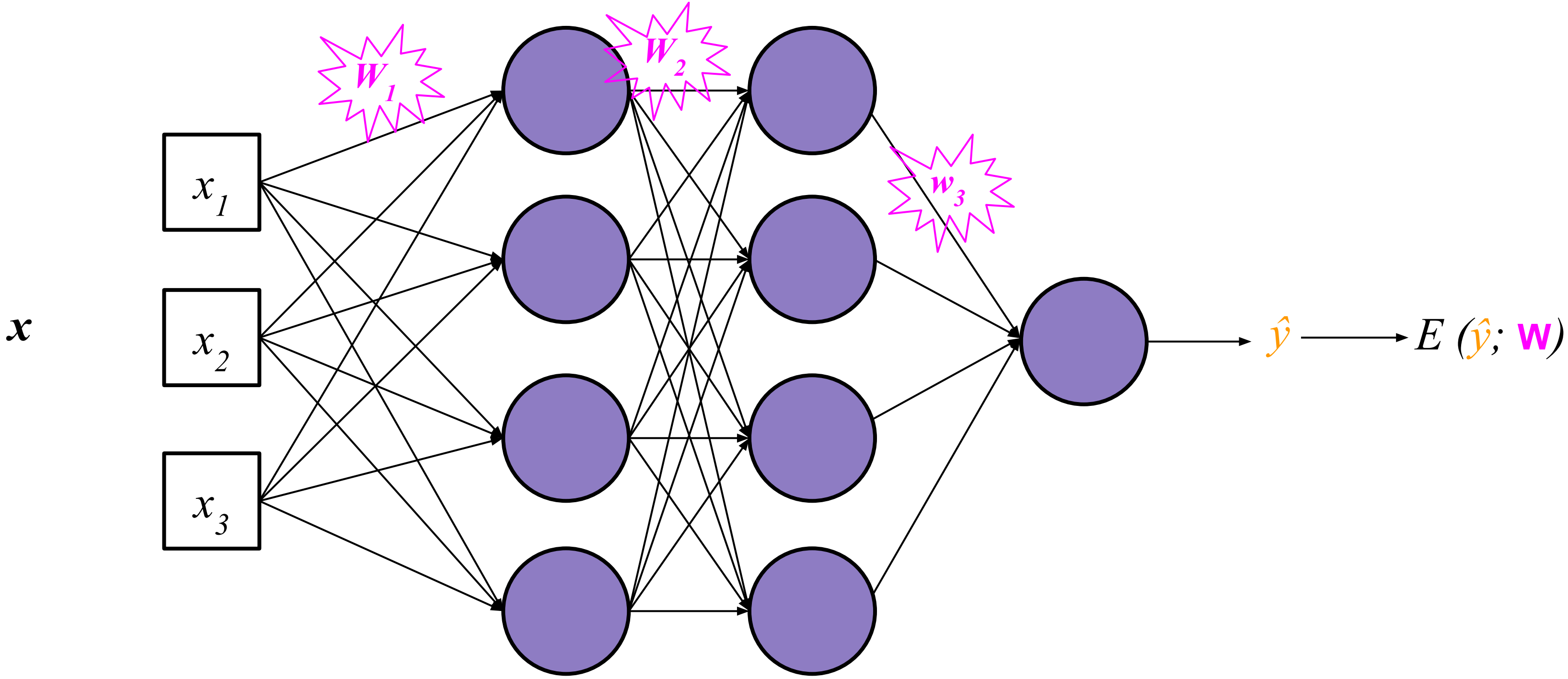
Mean Squared Error Loss



$$E = \text{loss} = \frac{1}{2}(y - \hat{y})^2$$

true output

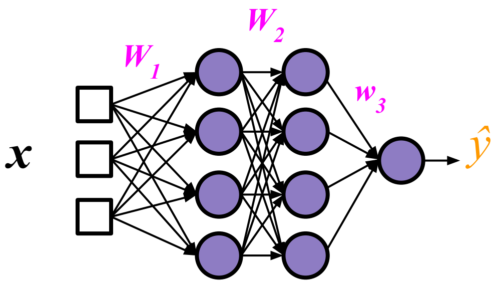
Backpropagation



We want to find the network weights that ***achieve the lowest loss***

$$\mathbf{W}^* = \underset{\mathbf{W}}{\operatorname{argmin}}\ E(\hat{y}\,;\,\mathbf{W})$$

Training Neural Networks



How do we learn the optimal weights W_L for our task??

- **Gradient descent:**

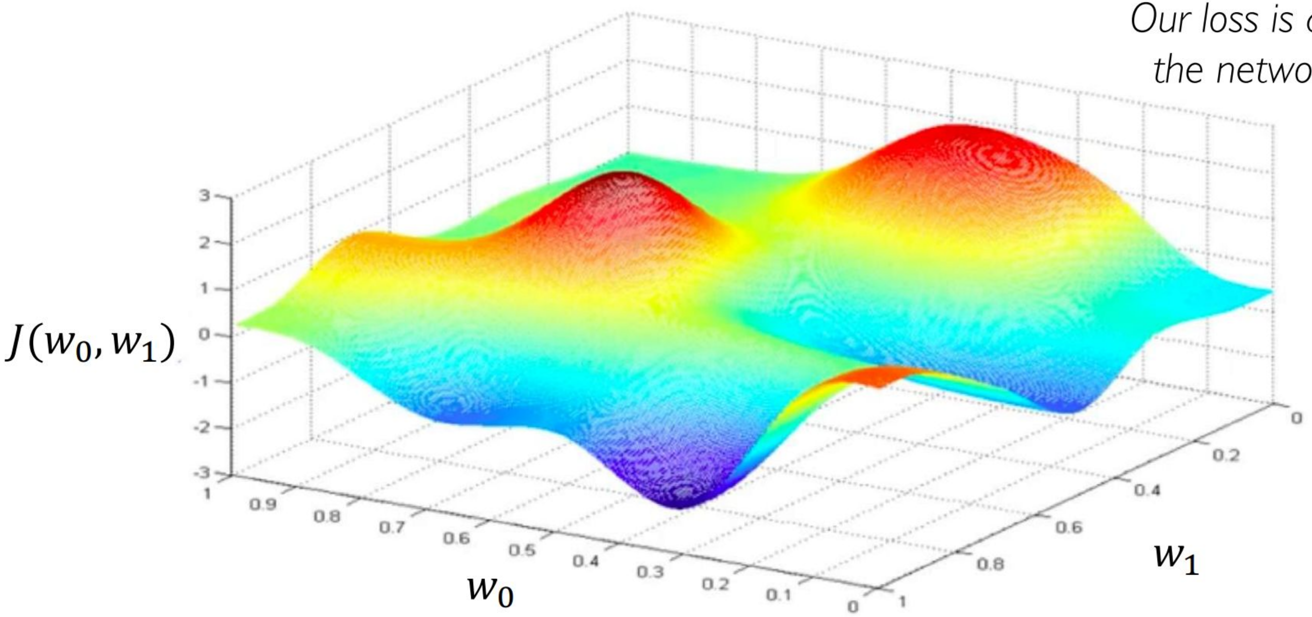
$$W_L(t+1) = W_L(t) - \eta \frac{\partial E}{\partial W_L(t)}$$

LeCun et. al. *Efficient Backpropagation*. 1998

Gradient Descent

$$\mathbf{W}^* = \underset{\mathbf{W}}{\operatorname{argmin}}\ E(\hat{y}\ ;\ \mathbf{W})$$

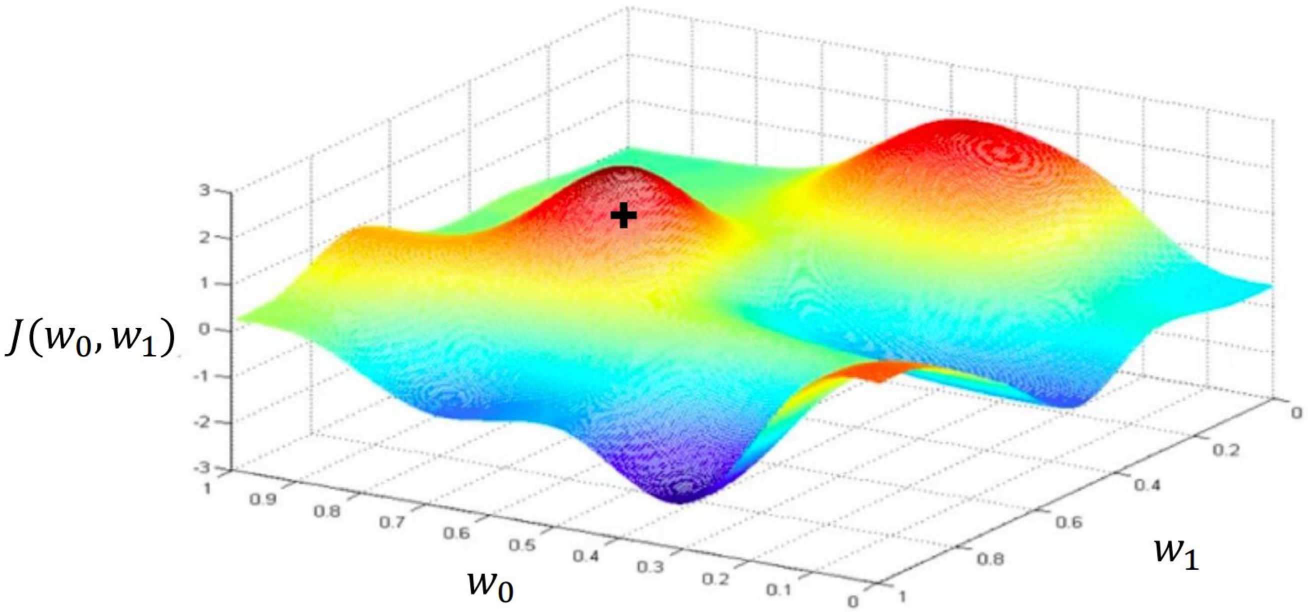
Remember:
Our loss is a function of the network weights!



http://introtodeeplearning.com/2019/materials/2019_6S191_L1.pdf

Gradient Descent

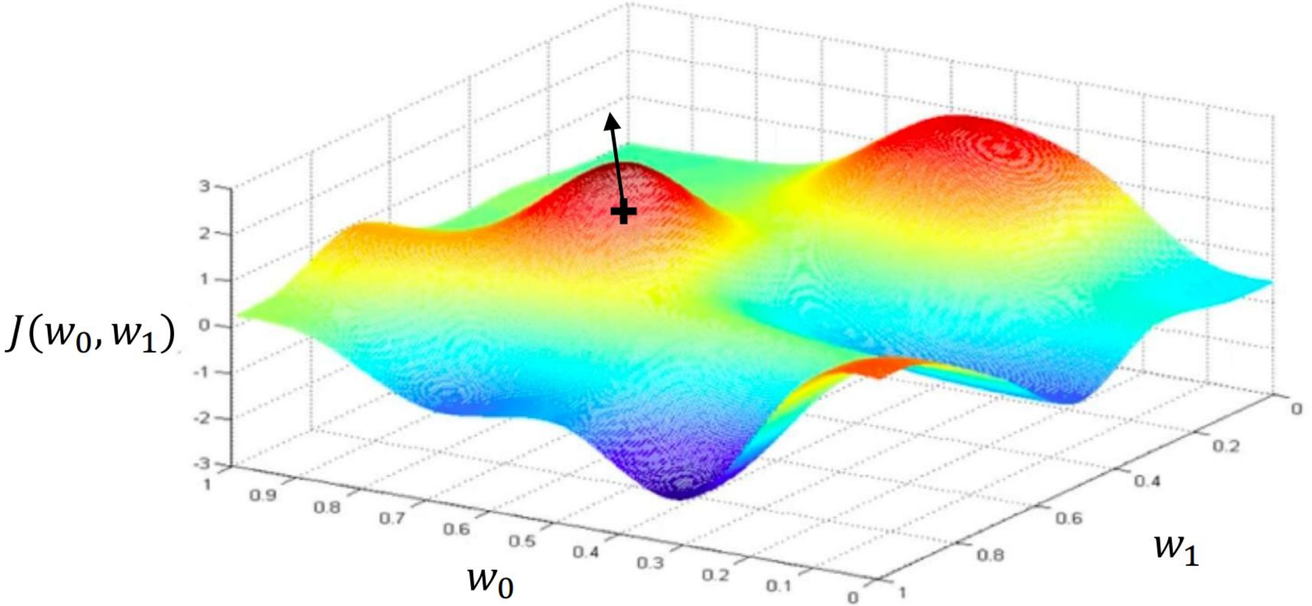
Randomly pick an initial (w_0, w_1)



http://introtodeeplearning.com/2019/materials/2019_6S191_L1.pdf

Gradient Descent

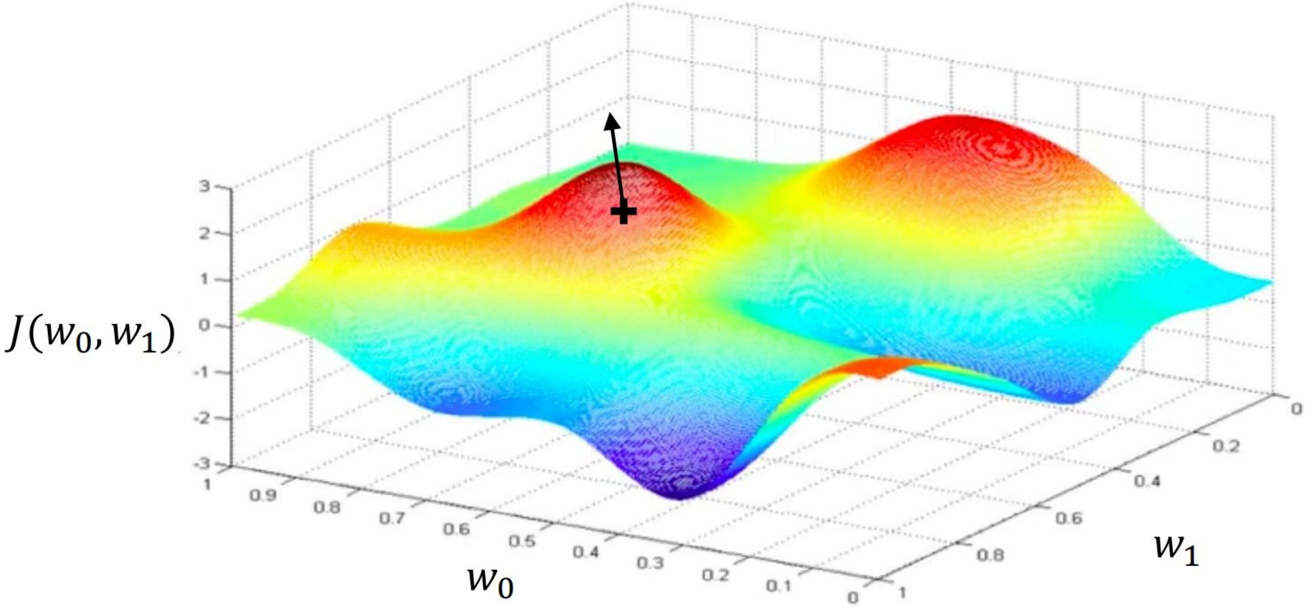
Compute gradient, $\dfrac{\partial E}{\partial W(t)}$



http://introtodeeplearning.com/2019/materials/2019_6S191_L1.pdf

Gradient Descent

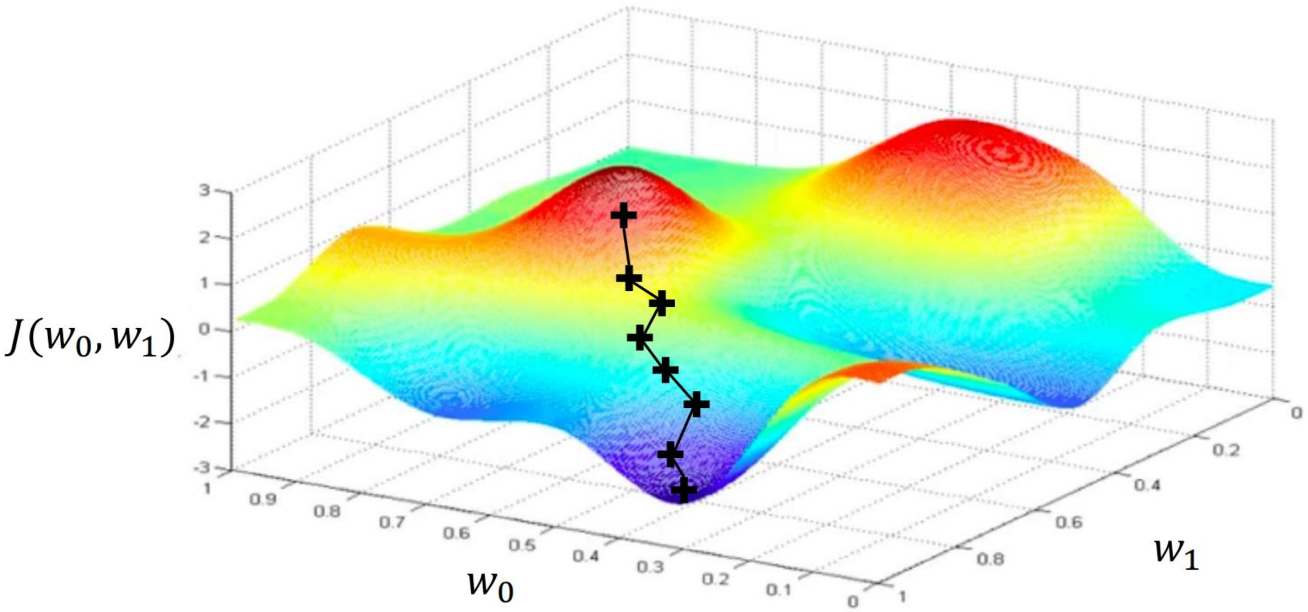
Compute gradient, $\dfrac{\partial E}{\partial W(t)}$



http://introtodeeplearning.com/2019/materials/2019_6S191_L1.pdf

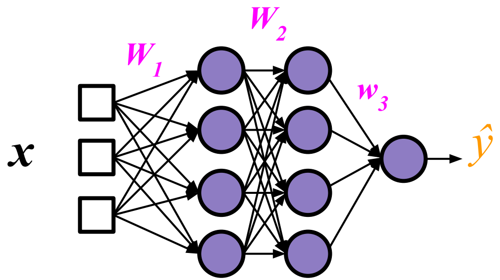
Gradient Descent

Repeat until convergence



http://introtodeeplearning.com/2019/materials/2019_6S191_L1.pdf

Training Neural Networks



But how do we get gradients of lower layers (e.g. W_1) ?

- **Backpropagation!**
  - Repeated application of chain rule of calculus
  - Locally minimize the objective
  - Requires all “blocks” of the network to be differentiable

LeCun et. al. *Efficient Backpropagation*. 1998

Backpropagation Intro

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

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

http://cs231n.stanford.edu/slides/winter1516_lecture5.pdf

Backpropagation Intro

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

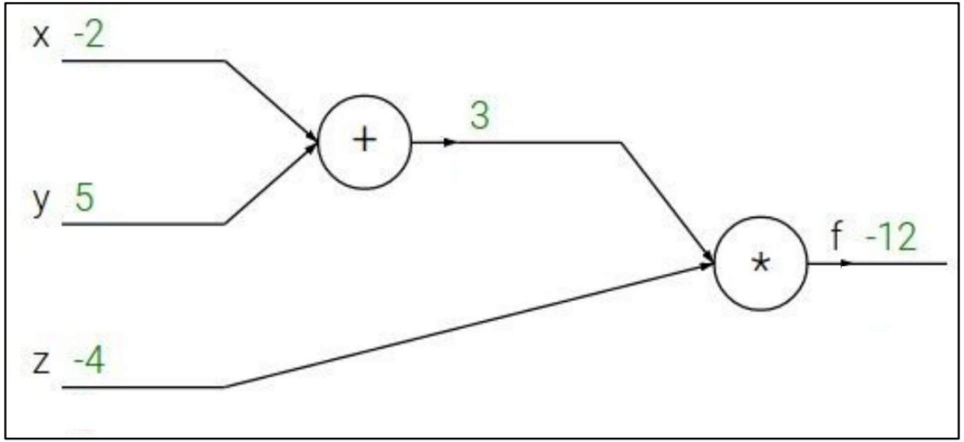
e.g. x = -2, y = 5, z = -4

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

Backpropagation Intro

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

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



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

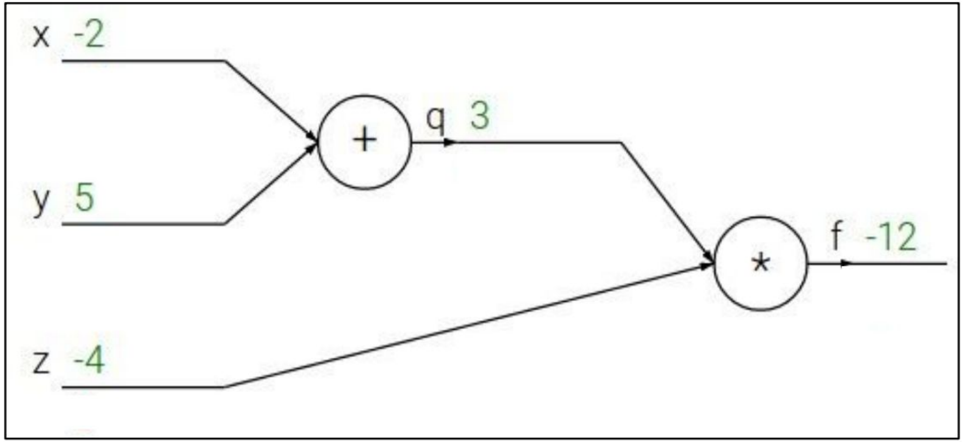
http://cs231n.stanford.edu/slides/winter1516_lecture5.pdf

Backpropagation Intro

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

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

$$\boxed{q = x + y}$$



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

http://cs231n.stanford.edu/slides/winter1516_lecture5.pdf

Backpropagation Intro

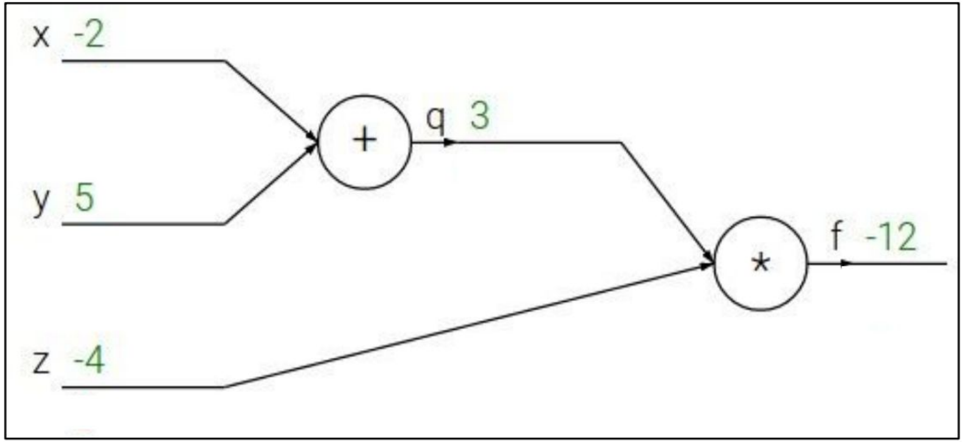
$$f(x, y, z) = (x + y)z$$

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

$$q = x + y$$

$$f = qz$$

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



http://cs231n.stanford.edu/slides/winter1516_lecture5.pdf

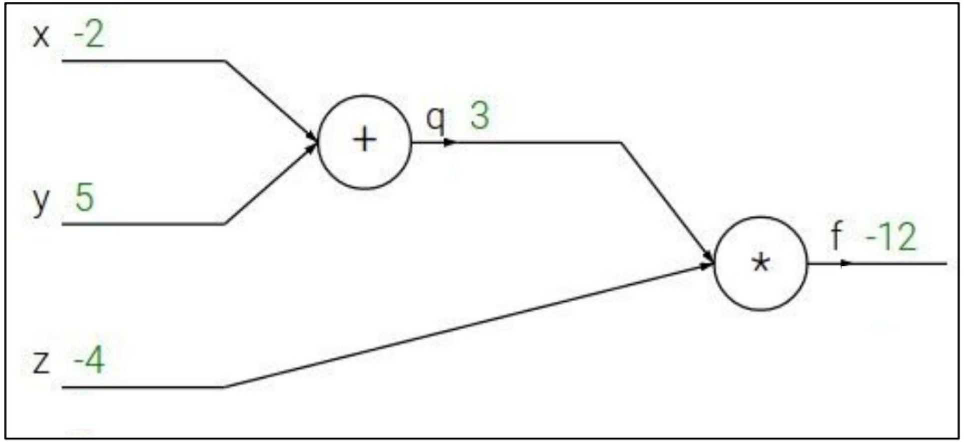
Backpropagation Intro

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

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

http://cs231n.stanford.edu/slides/winter1516_lecture5.pdf

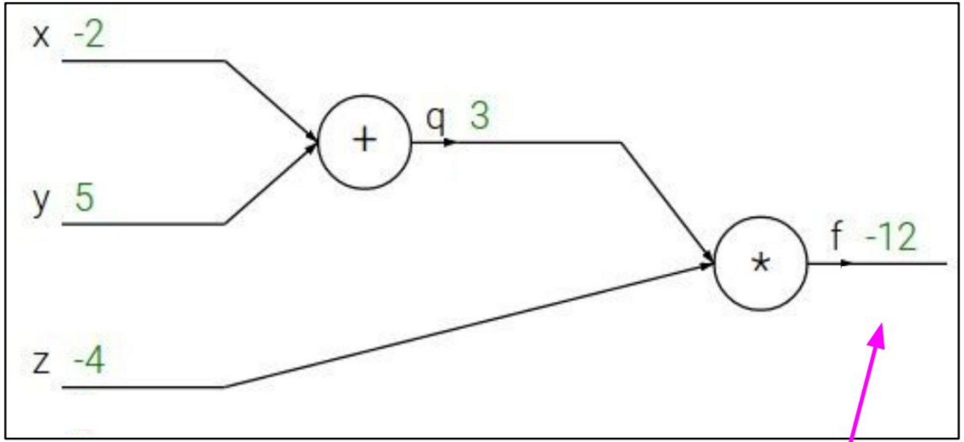
Backpropagation Intro

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

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

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

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



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

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

Backpropagation Intro

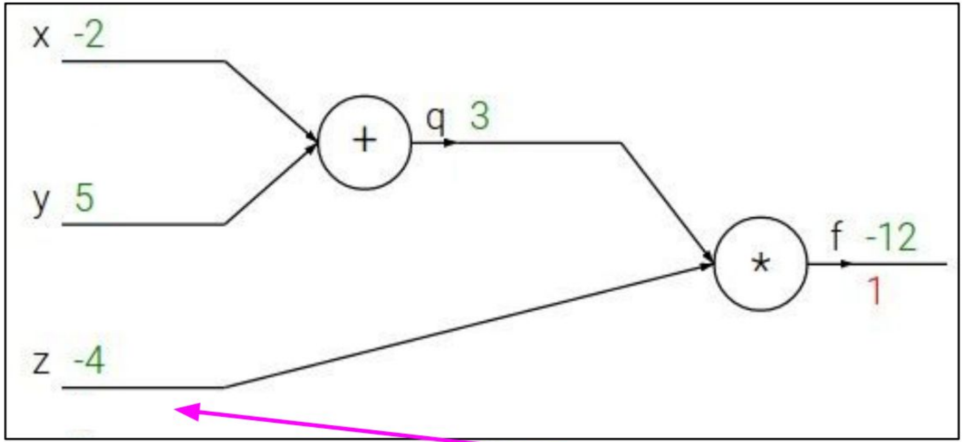
$$f(x, y, z) = (x + y)z$$

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



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

http://cs231n.stanford.edu/slides/winter1516_lecture5.pdf

Backpropagation Intro

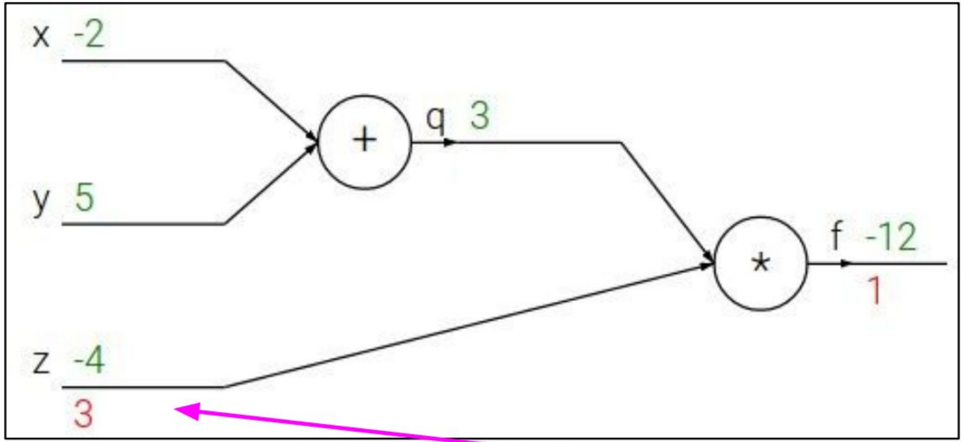
$$f(x,y,z) = (x+y)z$$

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



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

http://cs231n.stanford.edu/slides/winter1516_lecture5.pdf

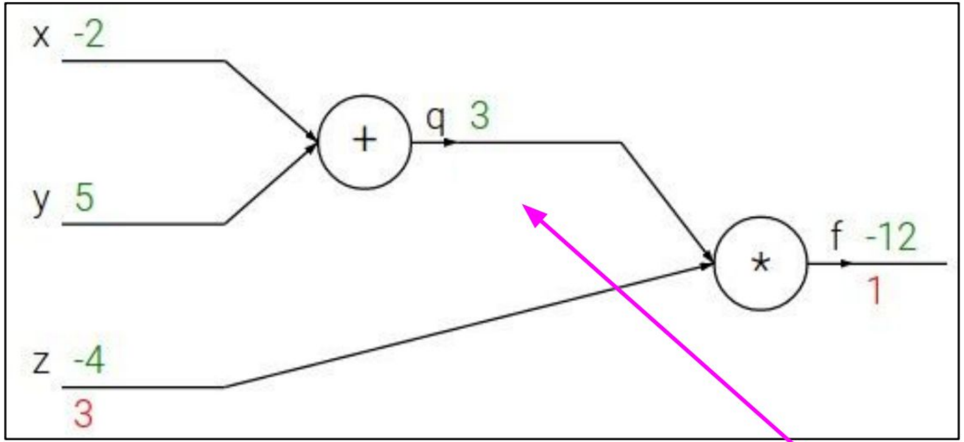
Backpropagation Intro

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

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

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

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



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

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

http://cs231n.stanford.edu/slides/winter1516_lecture5.pdf

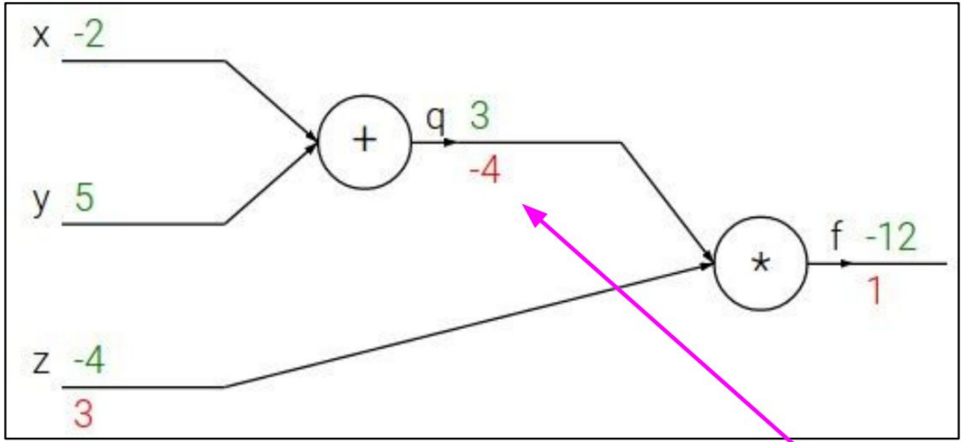
Backpropagation Intro

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

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

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

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



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

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

Backpropagation Intro

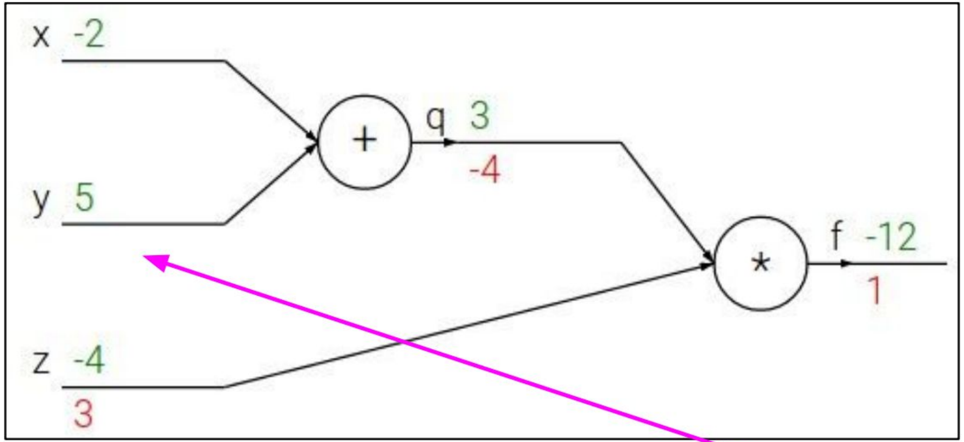
$$f(x, y, z) = (x + y)z$$

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



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

http://cs231n.stanford.edu/slides/winter1516_lecture5.pdf

Backpropagation Intro

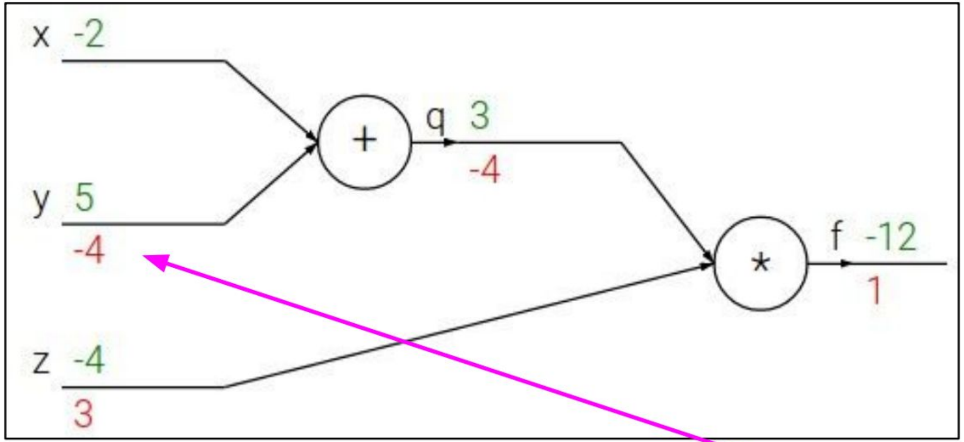
$$f(x,y,z) = (x+y)z$$

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:

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

Backpropagation Intro

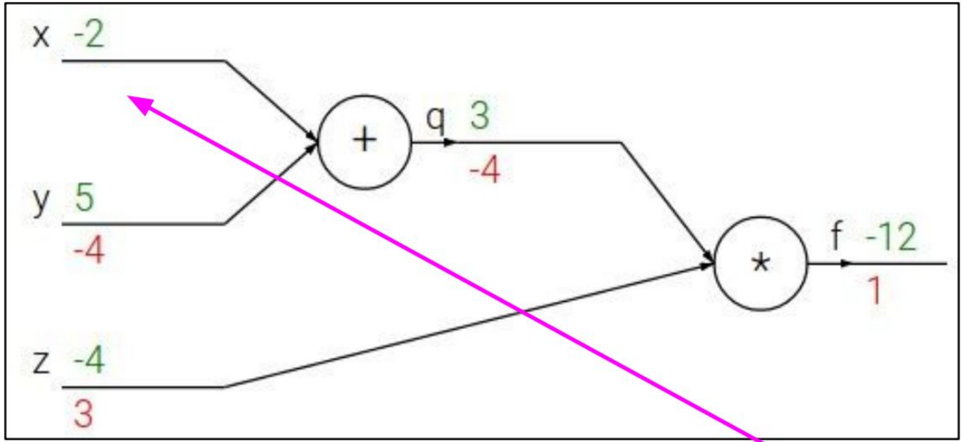
$$f(x, y, z) = (x + y)z$$

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



$$\frac{\partial f}{\partial x}$$

http://cs231n.stanford.edu/slides/winter1516_lecture5.pdf

Backpropagation Intro

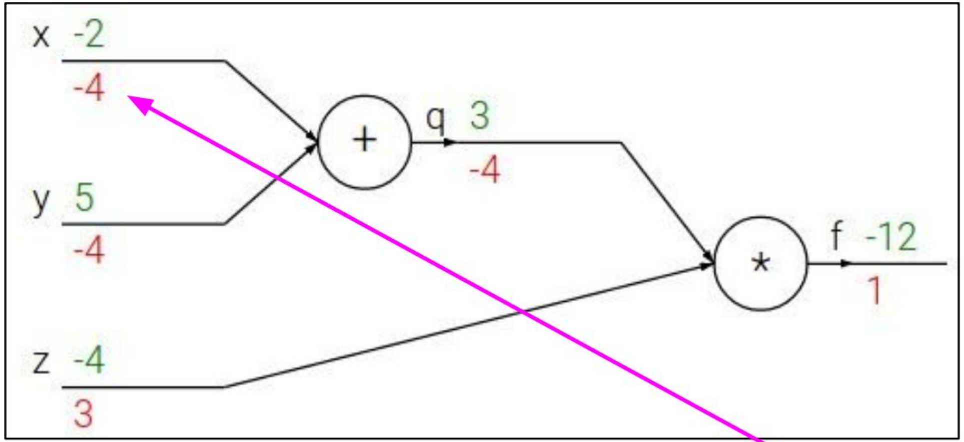
$$f(x,y,z) = (x+y)z$$

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



$$\frac{\partial f}{\partial x}$$

Chain rule:

$$\boxed{\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q}\frac{\partial q}{\partial x}}$$

Backpropagation Intro

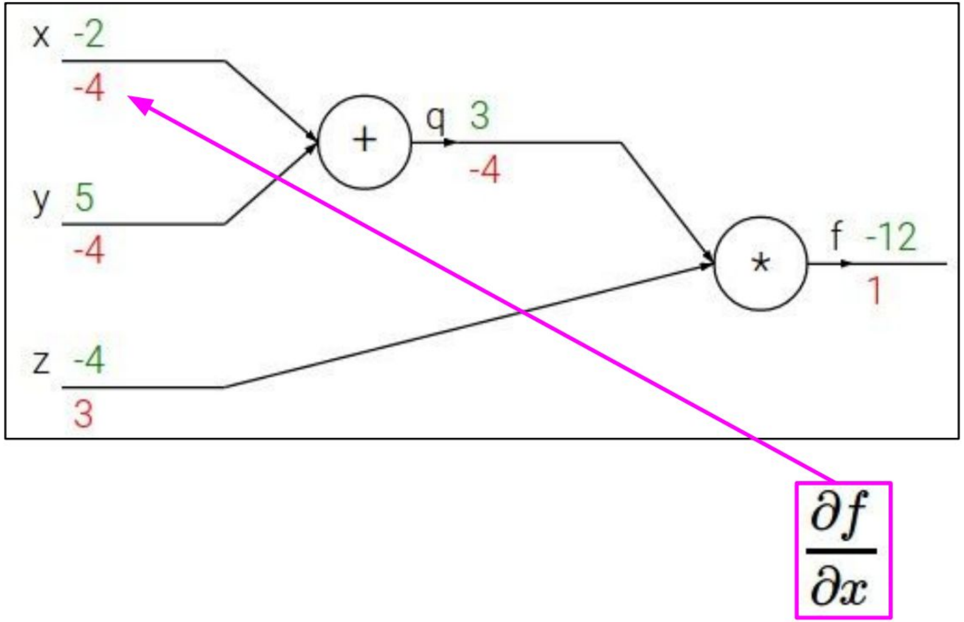
$$f(x, y, z) = (x + y)z$$

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

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

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

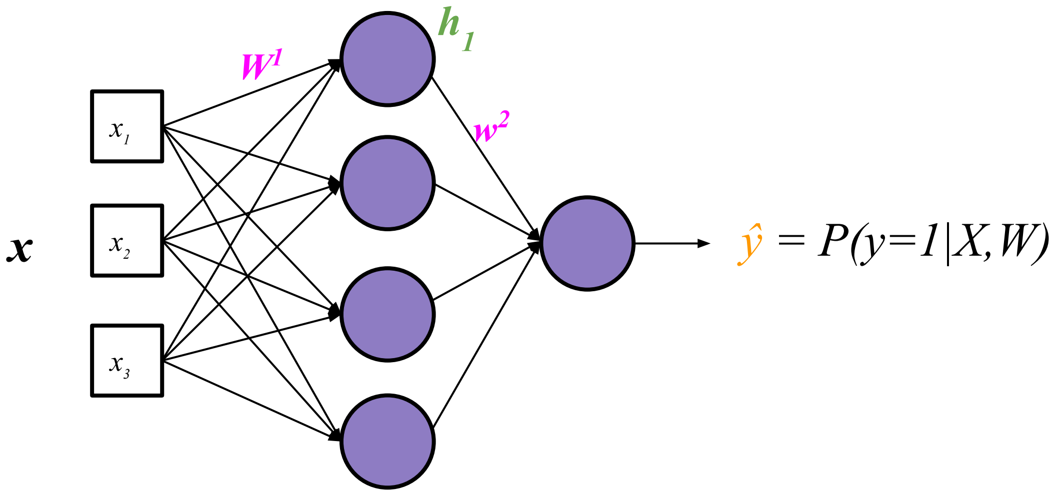
Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



Tells us: by increasing x by a scale of 1, we decrease f by a scale of 4

Backpropagation

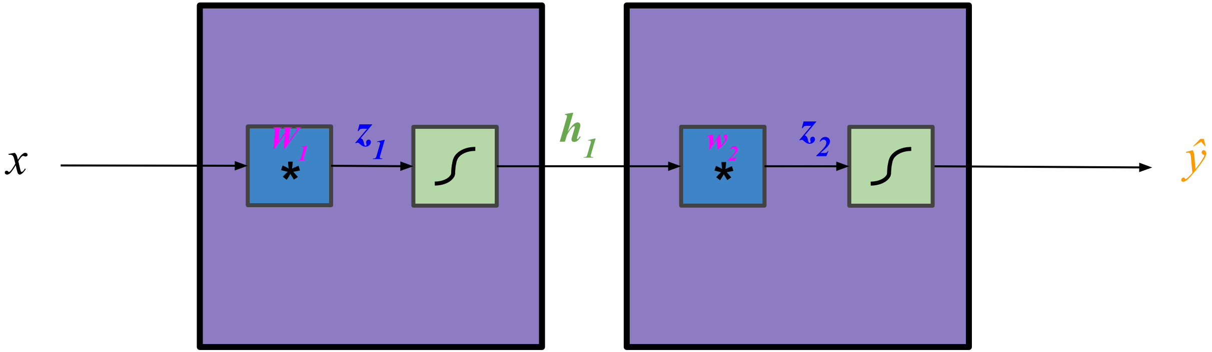
(binary classification example)



Example on 1-hidden layer network for binary classification

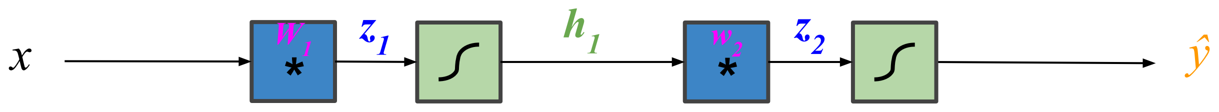
Backpropagation

(binary classification example)



Backpropagation

(binary classification example)



$$E = \text{loss} = -y\ln(\hat{y}) - (1-y)\ln(1-\hat{y})$$

Gradient Descent to Minimize loss:

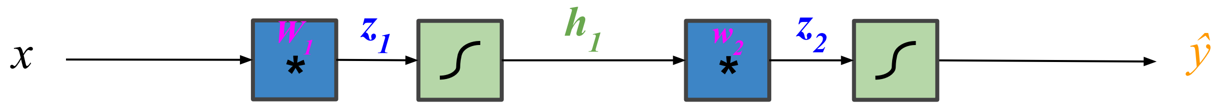
$$\mathbf{w}_2(t+1) = \mathbf{w}_2(t) - \eta \boxed{\frac{\partial E}{\partial \mathbf{w}_2(t)}}$$

$$W_1(t+1) = W_1(t) - \eta \boxed{\frac{\partial E}{\partial W_1(t)}}$$

Need to find these!

Backpropagation

(binary classification example)



$$E = f_4 = -y\ln(\hat{y}) - (1-y)\ln(1-\hat{y})$$

$$\hat{y} = f_3 = \frac{e^{z_2}}{1+e^{z_2}}$$

$$z_2 = f_3 = \boldsymbol{w}_2^T \boldsymbol{h}_1$$

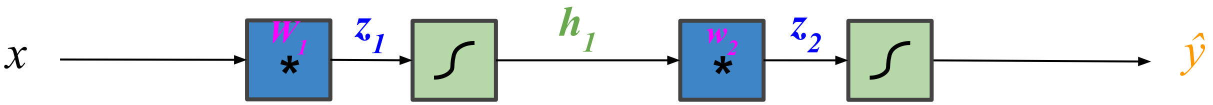
$$\boldsymbol{h}_1 = f_2 = \frac{e^{z_1}}{1+e^{z_1}}$$

$$\boldsymbol{z}_1 = f_1 = W_1^T \boldsymbol{x}$$

$$E = f_4(f_3(f_2(f_1(x))))$$

Backpropagation

(binary classification example)



$$E = f_4 = -y\ln(\hat{y}) - (1-y)\ln(1-\hat{y})$$

$$\hat{y} = f_3 = \frac{e^{z_2}}{1+e^{z_2}}$$

$$z_2 = f_3 = \boxed{\boldsymbol{w}_2^T}\boldsymbol{h}_1$$

$$\boldsymbol{h}_1 = f_2 = \frac{e^{z_1}}{1+e^{z_1}}$$

$$\boldsymbol{z}_1 = f_1 = \boxed{W_1^T}\boldsymbol{x}$$

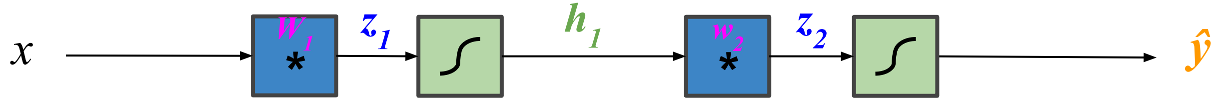
$$\boxed{\frac{\partial E}{\partial \mathbf{w}_2} = ??}$$

$$\boxed{\frac{\partial E}{\partial \mathbf{W}_1} = ??}$$

$$E = f_4(f_3(f_2(f_1(x))))$$

Backpropagation

(binary classification example)



$$E = f_4 = -y\ln(\hat{y}) - (1-y)\ln(1-\hat{y})$$

$$\hat{y} = f_3 = \frac{e^{z_2}}{1+e^{z_2}}$$

$$z_2 = f_3 = \boxed{\boldsymbol{w}_2^T}\boldsymbol{h}_1$$

$$\boldsymbol{h}_1 = f_2 = \frac{e^{z_1}}{1+e^{z_1}}$$

$$\boldsymbol{z}_1 = f_1 = \boxed{W_1^T}\boldsymbol{x}$$

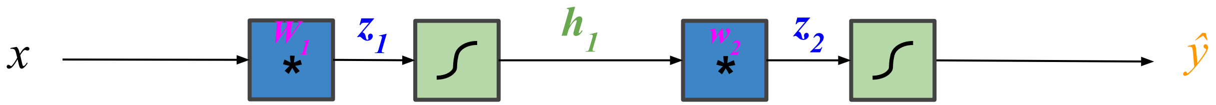
$$\boxed{\frac{\partial E}{\partial \mathbf{w}_2} = ??}$$

$$\boxed{\frac{\partial E}{\partial \mathbf{W}_1} = ??}$$

$$E = f_4(f_3(f_2(f_1(x))))$$ → Exploit the chain rule!

Backpropagation

(binary classification example)



$$E = -y\ln(\hat{y}) - (1-y)\ln(1-\hat{y})$$

$$\hat{y} = \frac{e^{z_2}}{1+e^{z_2}}$$

$$z_2 = \boxed{\boldsymbol{w}_2^T}\boldsymbol{h}_1$$

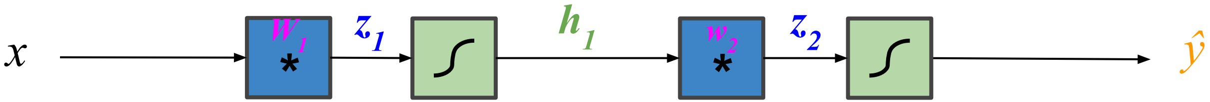
$$\boldsymbol{h}_1 = \frac{e^{z_1}}{1+e^{z_1}}$$

$$\boldsymbol{z}_1 = W_1^T\boldsymbol{x}$$

$$\boxed{\frac{\partial E}{\partial \boldsymbol{w}_2}} =$$

Backpropagation

(binary classification example)

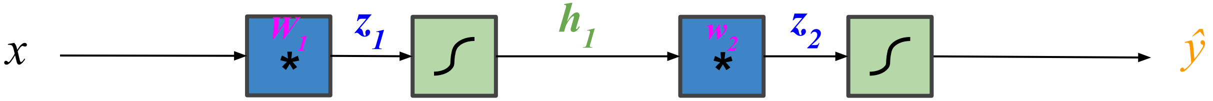


$$
\begin{aligned}
E &= -\,y\ln(\hat{y}) \\
&\quad -(1-y)\ln(1-\hat{y}) \\
\hat{y} &= \frac{e^{z_2}}{1+e^{z_2}} \\
z_2 &= \boxed{\boldsymbol{w}_2^T}\boldsymbol{h}_1 \\
\boldsymbol{h}_1 &= \frac{e^{z_1}}{1+e^{z_1}} \\
\boldsymbol{z}_1 &= W_1^T\boldsymbol{x}
\end{aligned}
$$

$$
\boxed{\frac{\partial E}{\partial \boldsymbol{w}_2}} = \overbrace{\frac{\partial E}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z_2} \cdot \frac{\partial z_2}{\partial \boldsymbol{w}_2}}^{\text{chain rule}}
$$

Backpropagation

(binary classification example)



$$\boxed{\begin{aligned} E = & -y\ln(\hat{y}) \\ & -(1-y)\ln(1-\hat{y}) \end{aligned}}$$

$$\hat{y} = \frac{e^{z_2}}{1+e^{z_2}}$$

$$z_2 = \boldsymbol{w}_2^T \boldsymbol{h}_1$$

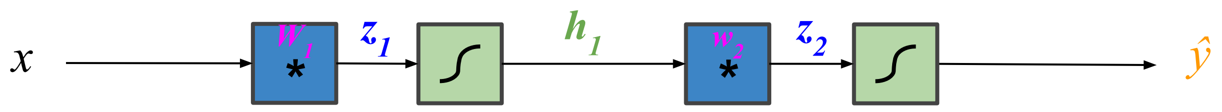
$$\boldsymbol{h}_1 = \frac{e^{\boldsymbol{z}_1}}{1+e^{\boldsymbol{z}_1}}$$

$$\boldsymbol{z}_1 = W_1^T \boldsymbol{x}$$

$$\begin{aligned} \frac{\partial E}{\partial \boldsymbol{w}_2} &= \boxed{\frac{\partial E}{\partial \hat{y}}} \cdot \frac{\partial \hat{y}}{\partial z_2} \cdot \frac{\partial z_2}{\partial \boldsymbol{w}_2} \\ &= \boxed{\left(\frac{\hat{y}-y}{\hat{y}(1-\hat{y})}\right)} \cdot \end{aligned}$$

Backpropagation

(binary classification example)



$$E = -y\ln(\hat{y}) - (1-y)\ln(1-\hat{y})$$

$$\boxed{\hat{y} = \frac{e^{z_2}}{1+e^{z_2}}}$$

$$z_2 = \boldsymbol{w}_2^T \boldsymbol{h}_1$$

$$\boldsymbol{h}_1 = \frac{e^{z_1}}{1+e^{z_1}}$$

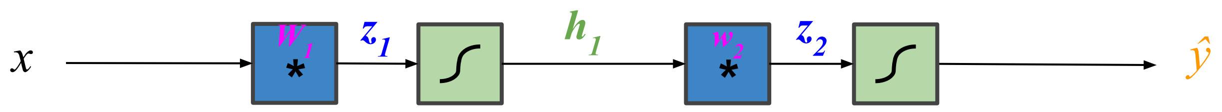
$$\boldsymbol{z}_1 = W_1^T \boldsymbol{x}$$

$$\frac{\partial E}{\partial \boldsymbol{w}_2} = \frac{\partial E}{\partial \hat{y}} \cdot \boxed{\frac{\partial \hat{y}}{\partial z_2}} \cdot \frac{\partial z_2}{\partial \boldsymbol{w}_2}$$

$$= \left(\frac{\hat{y}-y}{\hat{y}(1-\hat{y})}\right) \cdot \boxed{\left(\frac{e^{z_2}}{1+e^{z_2}}\left(1-\frac{e^{z_2}}{1+e^{z_2}}\right)\right)} \cdot$$

Backpropagation

(binary classification example)



$$E = -y \ln(\hat{y}) - (1-y)\ln(1-\hat{y})$$

$$\hat{y} = \frac{e^{z_2}}{1+e^{z_2}}$$

$$\boxed{z_2 = \boldsymbol{w}_2^T \boldsymbol{h}_1}$$

$$\boldsymbol{h}_1 = \frac{e^{z_1}}{1+e^{z_1}}$$

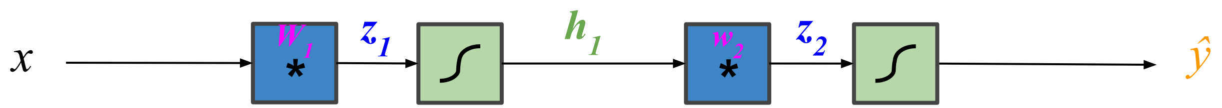
$$\boldsymbol{z}_1 = W_1^T \boldsymbol{x}$$

$$\frac{\partial E}{\partial \boldsymbol{w}_2} = \frac{\partial E}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z_2} \cdot \boxed{\frac{\partial z_2}{\partial \boldsymbol{w}_2}}$$

$$= \left(\frac{\hat{y}-y}{\hat{y}(1-\hat{y})}\right) \cdot \left(\frac{e^{z_2}}{1+e^{z_2}}\left(1 - \frac{e^{z_2}}{1+e^{z_2}}\right)\right) \cdot \boxed{(\boldsymbol{h}_1)}$$

Backpropagation

(binary classification example)

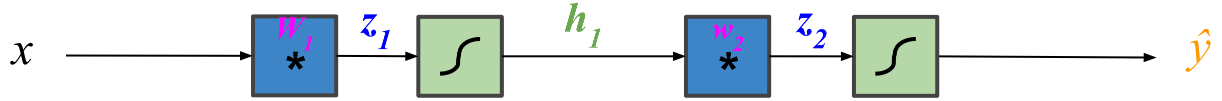


$$
\begin{aligned}
E &= -\,y\ln(\hat{y}) \\
&\quad -(1-y)\ln(1-\hat{y}) \\
\hat{y} &= \frac{e^{z_2}}{1+e^{z_2}} \\
z_2 &= \boldsymbol{w}_2^T \boldsymbol{h}_1 \\
\boldsymbol{h}_1 &= \frac{e^{\boldsymbol{z}_1}}{1+e^{\boldsymbol{z}_1}} \\
\boldsymbol{z}_1 &= W_1^T \boldsymbol{x}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial E}{\partial \boldsymbol{w}_2} &= \frac{\partial E}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z_2} \cdot \frac{\partial z_2}{\partial \boldsymbol{w}_2} \\
&= \left(\frac{\hat{y}-y}{\hat{y}(1-\hat{y})}\right) \cdot \left(\frac{e^{z_2}}{1+e^{z_2}}\left(1-\frac{e^{z_2}}{1+e^{z_2}}\right)\right) \cdot (\boldsymbol{h}_1) \\
&= \left(\frac{\hat{y}-y}{\hat{y}(1-\hat{y})}\right) \cdot (\hat{y}(1-\hat{y})) \cdot (\boldsymbol{h}_1)
\end{aligned}
$$

Backpropagation

(binary classification example)



$$E = -y\ln(\hat{y}) - (1-y)\ln(1-\hat{y})$$

$$\hat{y} = \frac{e^{z_2}}{1+e^{z_2}}$$

$$z_2 = \boldsymbol{w}_2^T \boldsymbol{h}_1$$

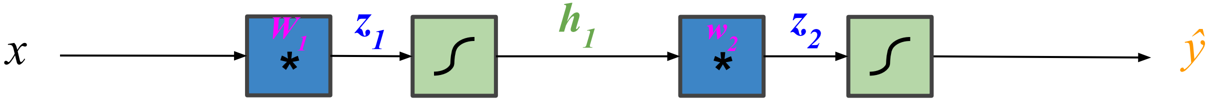
$$\boldsymbol{h}_1 = \frac{e^{z_1}}{1+e^{z_1}}$$

$$\boldsymbol{z}_1 = \boxed{W_1^T} \boldsymbol{x}$$

$$\boxed{\frac{\partial E}{\partial \boldsymbol{W}_1}} =$$

Backpropagation

(binary classification example)

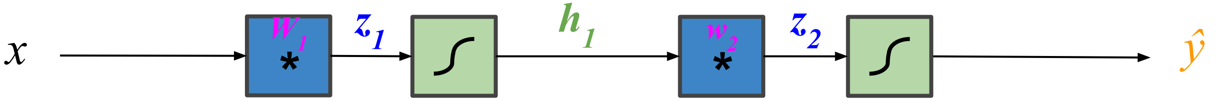


$$
\begin{aligned}
E &= -\,y\ln(\hat{y}) \\
&\quad -(1-y)\ln(1-\hat{y}) \\
\hat{y} &= \frac{e^{z_2}}{1+e^{z_2}} \\
z_2 &= \boldsymbol{w}_2^T \boldsymbol{h}_1 \\
\boldsymbol{h}_1 &= \frac{e^{\boldsymbol{z}_1}}{1+e^{\boldsymbol{z}_1}} \\
\boldsymbol{z}_1 &= \boxed{W_1^T}\,\boldsymbol{x}
\end{aligned}
$$

$$
\boxed{\frac{\partial E}{\partial \boldsymbol{W}_1}} = \frac{\partial E}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z_2} \cdot \frac{\partial z_2}{\partial \boldsymbol{h}_1} \cdot \frac{\partial \boldsymbol{h}_1}{\partial \boldsymbol{z}_1} \cdot \frac{\partial \boldsymbol{z}_1}{\partial W_1}
$$

Backpropagation

(binary classification example)



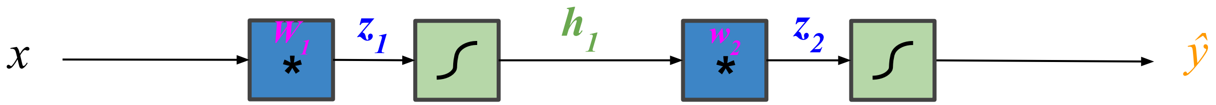
$$
\begin{aligned}
E &= -\,y\ln(\hat{y}) \\
&\quad -(1-y)\ln(1-\hat{y}) \\
\hat{y} &= \frac{e^{z_2}}{1+e^{z_2}} \\
z_2 &= \boldsymbol{w}_2^T \boldsymbol{h}_1 \\
\boldsymbol{h}_1 &= \frac{e^{z_1}}{1+e^{z_1}} \\
\boldsymbol{z}_1 &= \boxed{W_1^T}\boldsymbol{x}
\end{aligned}
$$

$$
\boxed{\frac{\partial E}{\partial \boldsymbol{W}_1}} = \frac{\partial E}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z_2} \cdot \frac{\partial z_2}{\partial \boldsymbol{h}_1} \cdot \frac{\partial \boldsymbol{h}_1}{\partial \boldsymbol{z}_1} \cdot \frac{\partial \boldsymbol{z}_1}{\partial W_1}
$$

$$
= \left(\frac{\hat{y}-y}{\hat{y}(1-\hat{y})}\right) \cdot (\hat{y}(1-\hat{y})) \cdot (\boldsymbol{w}) \cdot (\boldsymbol{h}_1(1-\boldsymbol{h}_1)) \cdot (x)
$$

Backpropagation

(binary classification example)



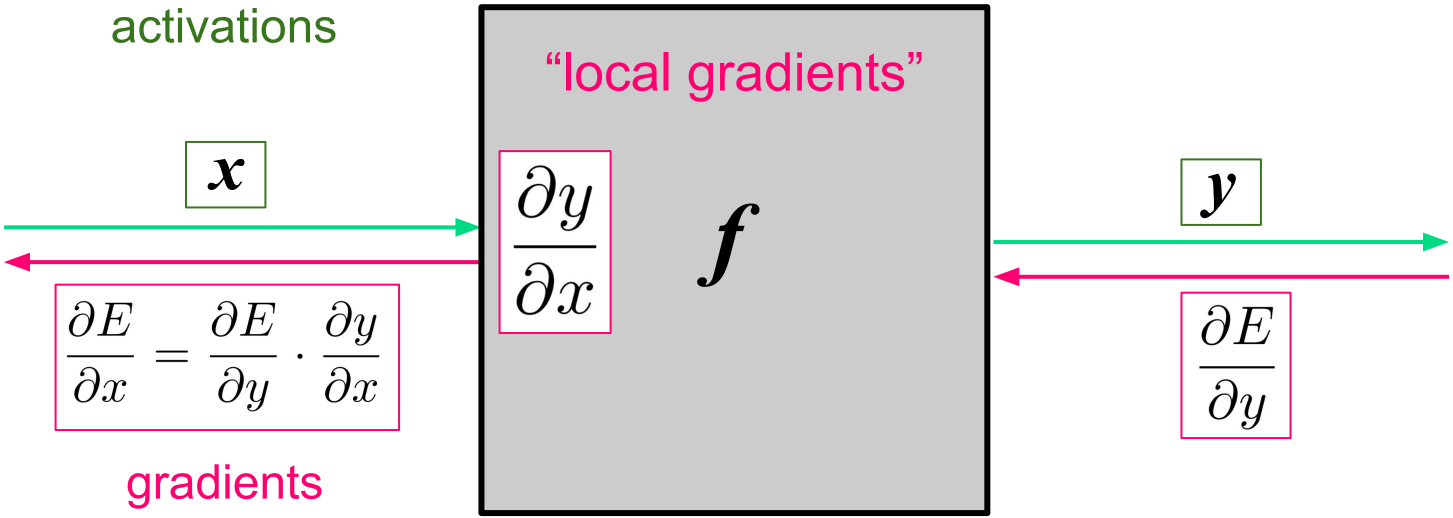
$$
\begin{aligned}
E &= -\,y\ln(\hat{y}) \\
&\quad -(1-y)\ln(1-\hat{y}) \\
\hat{y} &= \frac{e^{z_2}}{1+e^{z_2}} \\
z_2 &= \boldsymbol{w}_2^T \boldsymbol{h}_1 \\
\boldsymbol{h}_1 &= \frac{e^{z_1}}{1+e^{z_1}} \\
\boldsymbol{z}_1 &= W_1^T \boldsymbol{x}
\end{aligned}
$$

already computed!

$$
\frac{\partial E}{\partial \boldsymbol{W}_1} = \boxed{\frac{\partial E}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z_2}} \cdot \frac{\partial z_2}{\partial \boldsymbol{h}_1} \cdot \frac{\partial \boldsymbol{h}_1}{\partial \boldsymbol{z}_1} \cdot \frac{\partial \boldsymbol{z}_1}{\partial W_1}
$$

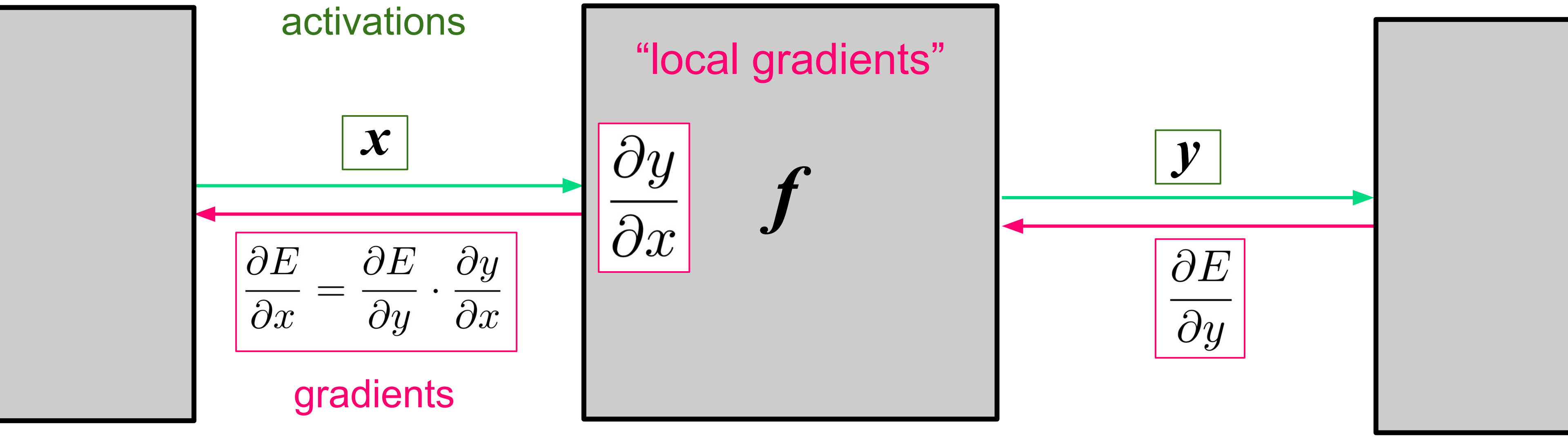
$$
= \left(\frac{\hat{y}-y}{\hat{y}(1-\hat{y})}\right) \cdot (\hat{y}(1-\hat{y})) \cdot (\boldsymbol{w}) \cdot (\boldsymbol{h}_1(1-\boldsymbol{h}_1)) \cdot (x)
$$

“Local-ness” of Backpropagation



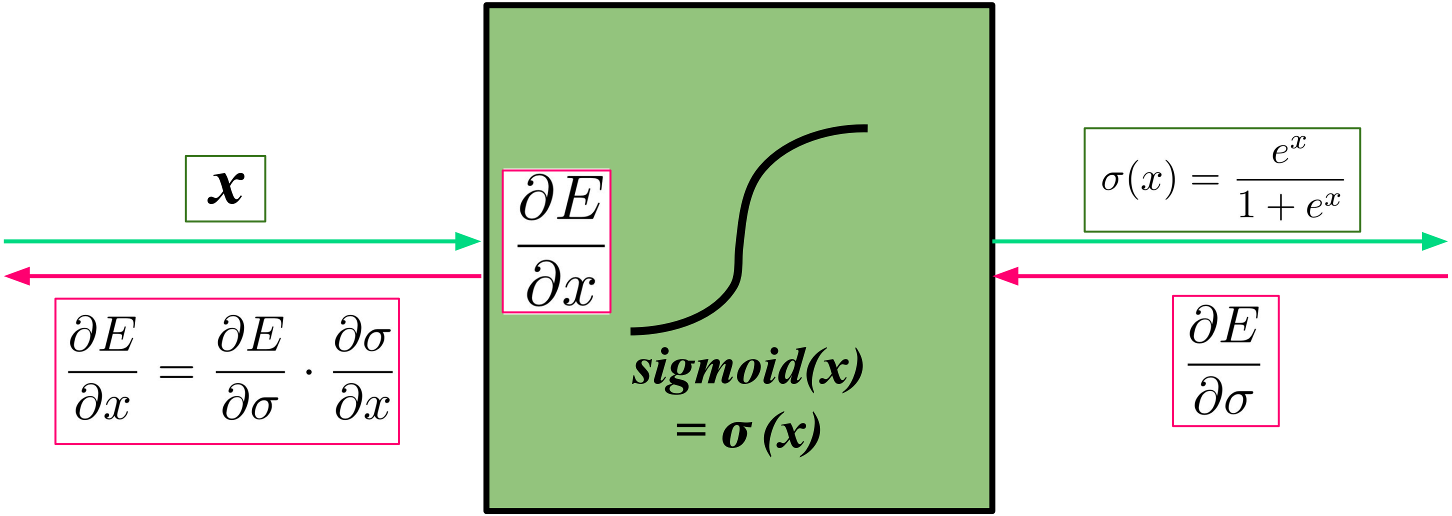
http://cs231n.stanford.edu/slides/winter1516_lecture5.pdf

"Local-ness" of Backpropagation



http://cs231n.stanford.edu/slides/winter1516_lecture5.pdf

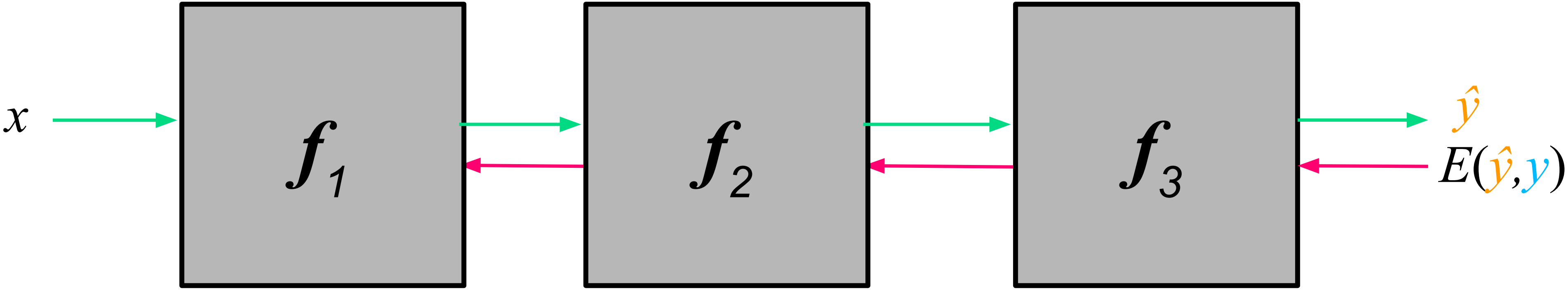
Example: Sigmoid Block



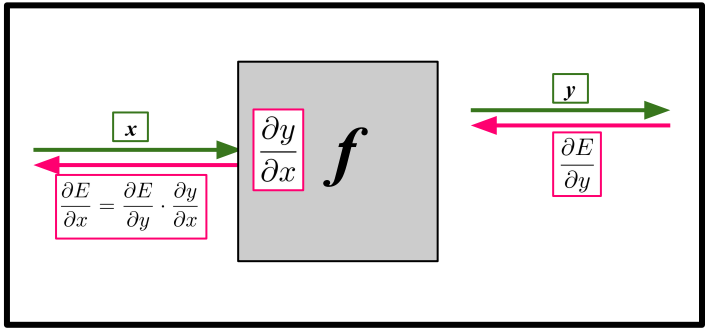
http://cs231n.stanford.edu/slides/winter1516_lecture5.pdf

Deep Learning:

Layers of Differentiable Parameterized Functions (with nonlinearities)

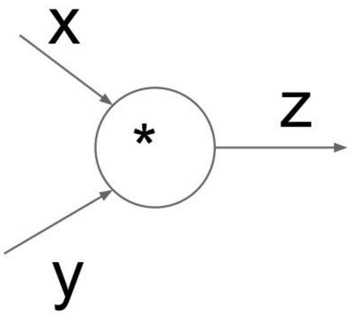


Building Deep Neural Nets



http://cs231n.stanford.edu/slides/winter1516_lecture5.pdf

Block Example Implementation



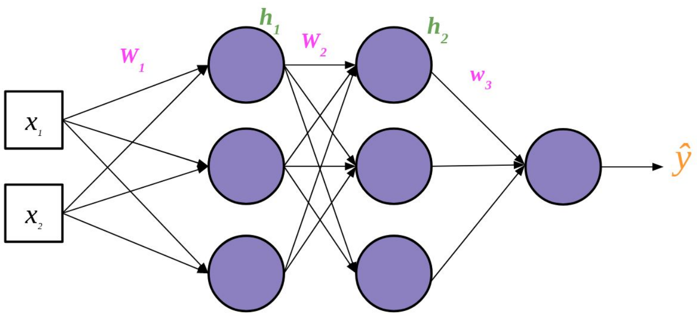

```
class MultiplyGate(object):
    def forward(x,y):
        z = x*y
        self.x = x # must keep these around!
        self.y = y
        return z
    def backward(dz):
        dx = self.y * dz # [dz/dx * dL/dz]
        dy = self.x * dz # [dz/dy * dL/dz]
        return [dx, dy]
```

(x,y,z are scalars)

Deep Learning Frameworks

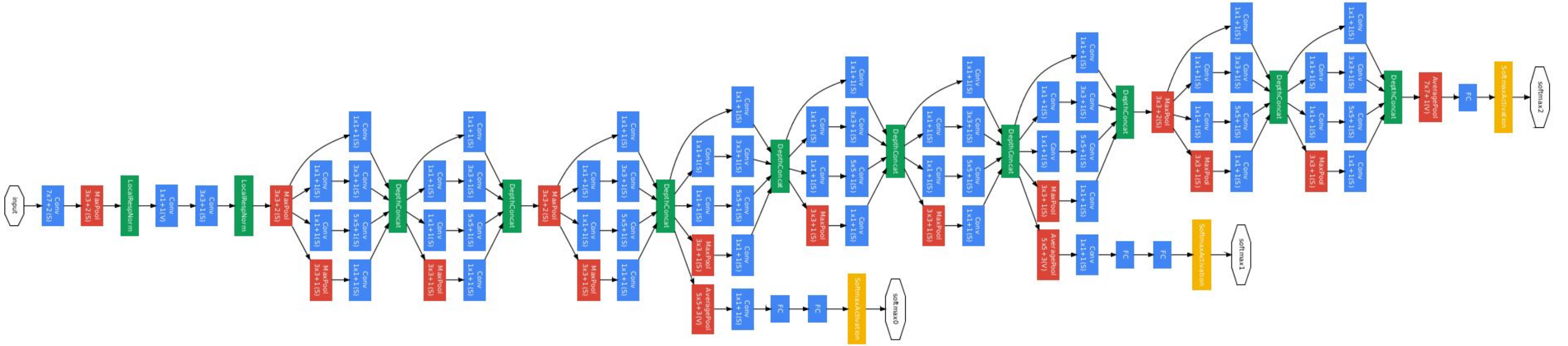
Chainer
mxnet
Caffe2
Microsoft CNTK
TensorFlow
Keras
GLUON
PYTORCH
theano

Pytorch Sample Code



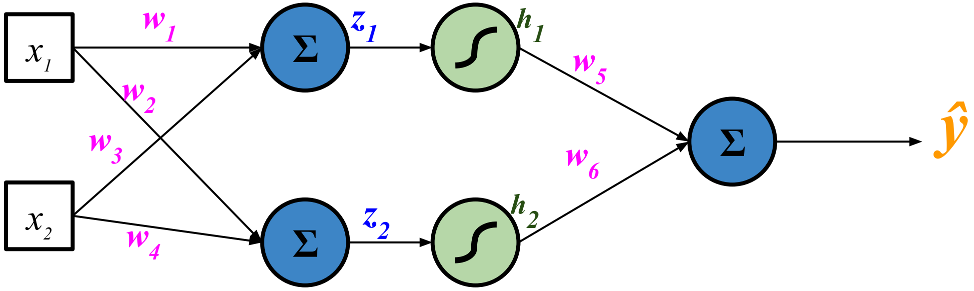

```
import torch.nn as nn
import torch.nn.functional as F

class ThreeLayerNet(torch.nn.Module):
    def __init__(self, d_in, d_hidden, d_out):
        super().__init__()
        self.W1 = nn.Linear(d_in,d_hidden)
        self.W2 = nn.Linear(d_hidden,d_hidden)
        self.w3 = nn.Linear(d_hidden,d_out)
        self.nonlinear = nn.Sigmoid()

    def forward(self, x):
        h1 = self.nonlinear(self.W1(x))
        h2 = self.nonlinear(self.W2(h1))
        y_hat = self.nonlinear(self.w3(h2))
        return y_hat

model = ThreeLayerNet(2,3,1)
```

Building Deep Neural Nets



“GoogLeNet” for Object Classification

Backprop Demo



f_4	$E = (y - \hat{y})^2$
f_3	$\hat{y} = h_1 w_5 + h_2 w_6$
f_2	$h_1 = \dfrac{exp(z_1)}{1 + exp(z_1)}$ $h_2 = \dfrac{exp(z_2)}{1 + exp(z_2)}$
f_1	$z_1 = x_1 w_1 + x_2 w_3$ $z_2 = x_1 w_2 + x_2 w_4$

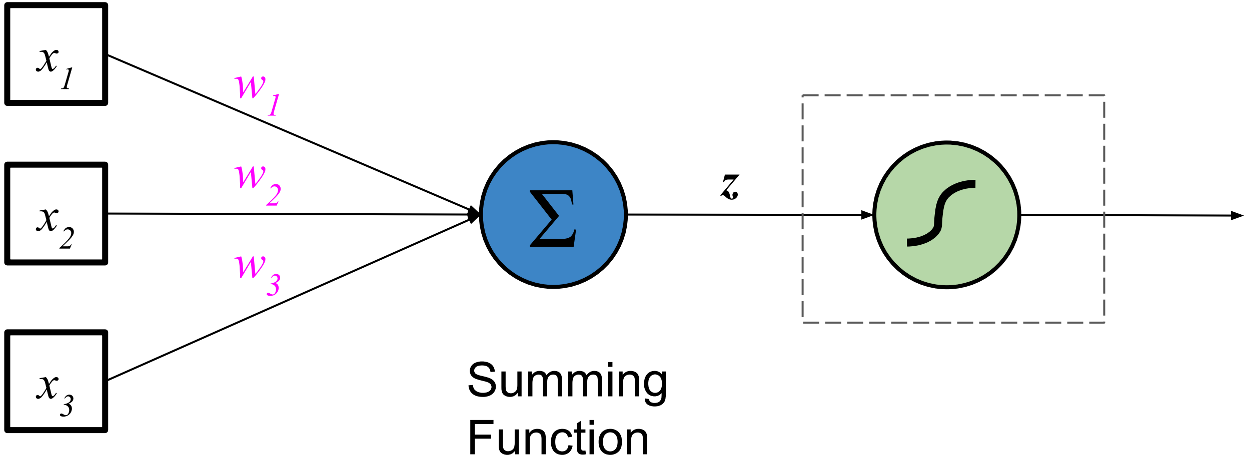
$$w(t+1) = w(t) - \eta \frac{\partial E}{\partial w(t)}$$

$$\boxed{\frac{\partial E}{\partial w_i} = ??}$$

Nonlinearity Functions

Nonlinearity Functions

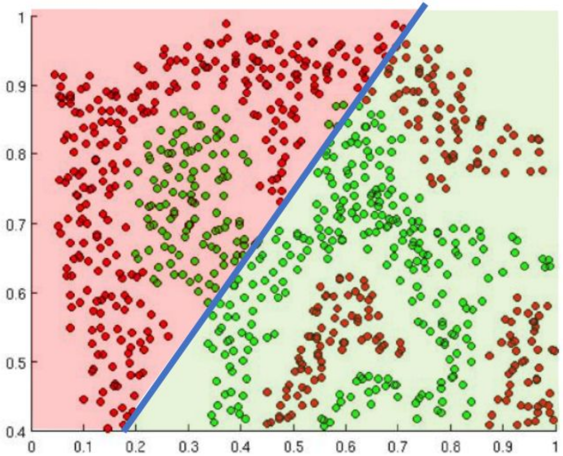
(i.e. transfer or activation functions)



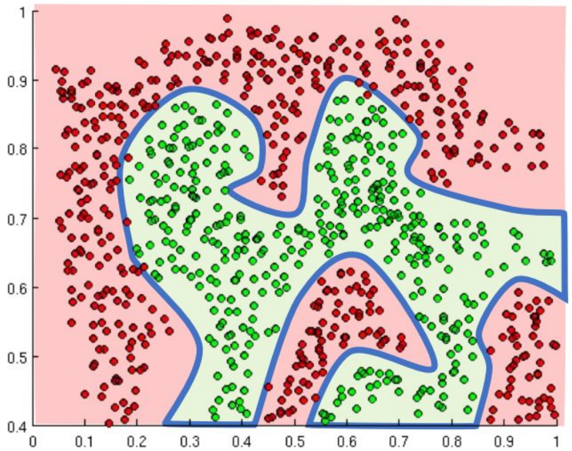
Nonlinearity Functions

(i.e. transfer or activation functions)

The purpose of activation functions is to ***introduce non-linearities*** *into the network*



Linear Activation functions produce linear decisions no matter the network size

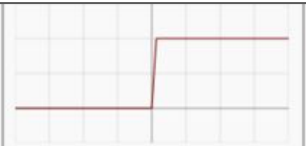
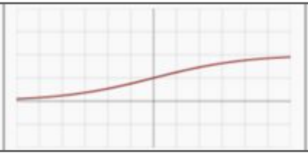


Non-linearities allow us to approximate arbitrarily complex functions

http://introtodeeplearning.com/2019/materials/2019_6S191_L1.pdf

Nonlinearity Functions

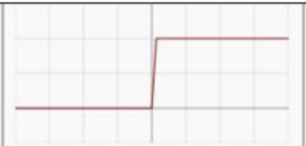
(i.e. transfer or activation functions)

Name	Plot	Equation	Derivative (w.r.t x)
Binary step		$f(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ 1 & \text{for} \quad x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for} \quad x \neq 0 \\ ? & \text{for} \quad x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
TanH		$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
Rectifier (ReLU)[9]		$f(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ x & \text{for} \quad x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ 1 & \text{for} \quad x \geq 0 \end{cases}$

https://en.wikipedia.org/wiki/Activation_function#Comparison_of_activation_functions

Nonlinearity Functions

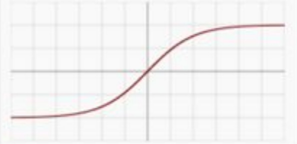
(i.e. transfer or activation functions)

Name	Plot	Equation	Derivative (w.r.t x)
Binary step		$f(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ 1 & \text{for} \quad x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for} \quad x \neq 0 \\ ? & \text{for} \quad x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
TanH		$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
Rectifier (ReLU)[9]		$f(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ x & \text{for} \quad x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ 1 & \text{for} \quad x \geq 0 \end{cases}$

https://en.wikipedia.org/wiki/Activation_function#Comparison_of_activation_functions

Nonlinearity Functions

(i.e. transfer or activation functions)

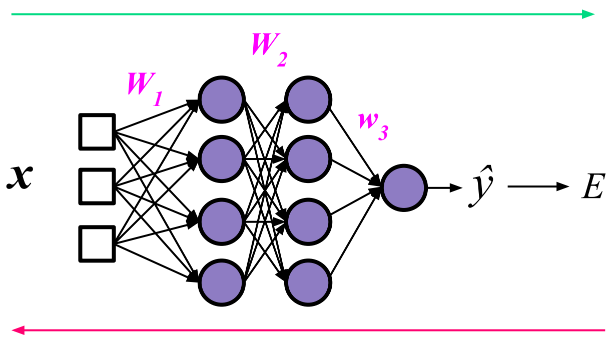
Name	Plot	Equation	Derivative (w.r.t x)
Binary step		$f(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ 1 & \text{for} \quad x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for} \quad x \neq 0 \\ ? & \text{for} \quad x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
TanH		$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
Rectifier (ReLU)[9]		$f(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ x & \text{for} \quad x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ 1 & \text{for} \quad x \geq 0 \end{cases}$

usually works best in practice

https://en.wikipedia.org/wiki/Activation_function#Comparison_of_activation_functions

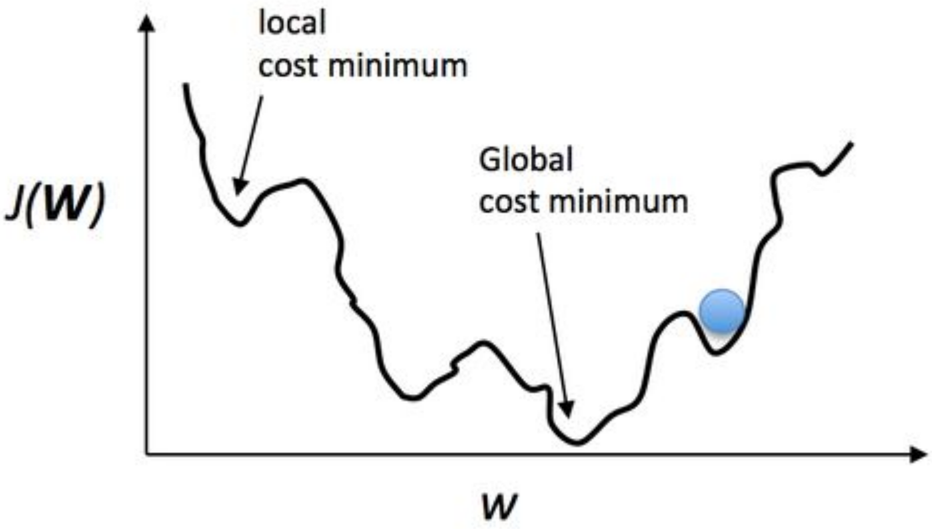
Neural Nets in Practice

Neural Net Pipeline



1. Initialize weights
2. For each batch of input x samples S:
   a. Run the network "Forward" on S to compute outputs and loss
   b. Run the network "Backward" using outputs and loss to compute gradients
   c. Update weights using SGD (or a similar method)
3. Repeat step 2 until loss convergence

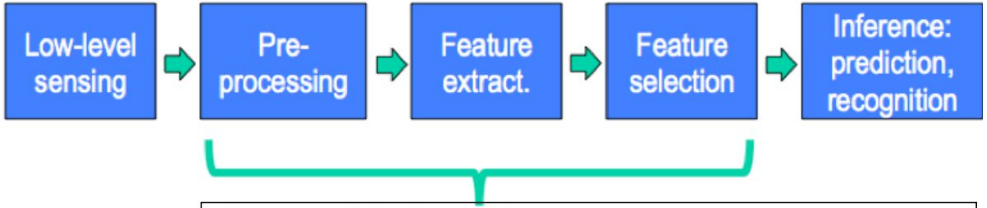
Non-Convexity of Neural Nets



In very high dimensions, there exists many local minimum which are about the same.

Pascanu, et. al. *On the saddle point problem for non-convex optimization 2014*

Advantage of Neural Nets



Feature Engineering
- ✓ Most critical for accuracy
- ✓ Account for most of the computation for testing
- ✓ Most time-consuming in development cycle
- ✓ Often hand-craft and task dependent in practice

Feature Learning
- ✓ Easily adaptable to new similar tasks
- ✓ Layerwise representation
- ✓ Layer-by-layer unsupervised training
- ✓ Layer-by-layer supervised training

As long as it's fully differentiable, we can train the model to automatically learn features for us.

Summary

Neurons

Neural Networks

Loss Functions and Backpropagation