Intro to Neural Networks and Deep Learning

Jack Lanchantin
Dr. Yanjun Qi

UVA CS 6316
University of Virginia

Neurons

1-Layer Neural Network

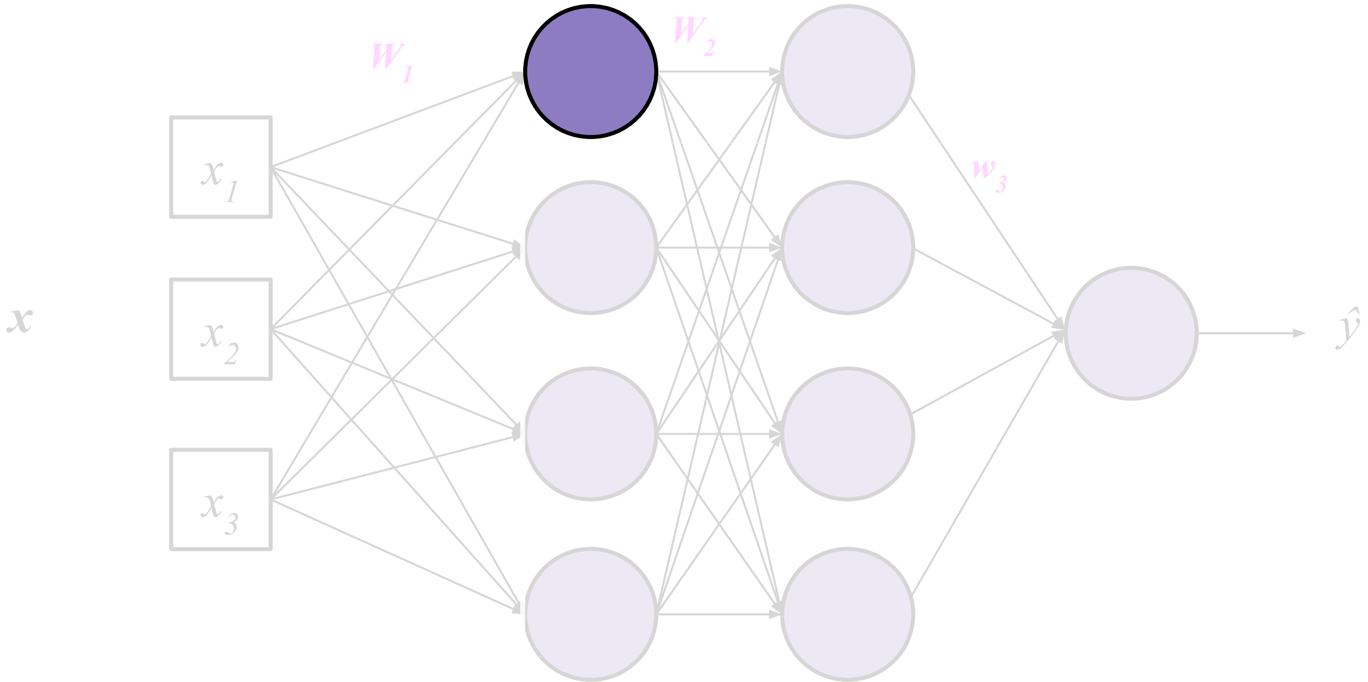
Multi-layer Neural Network

Loss Functions

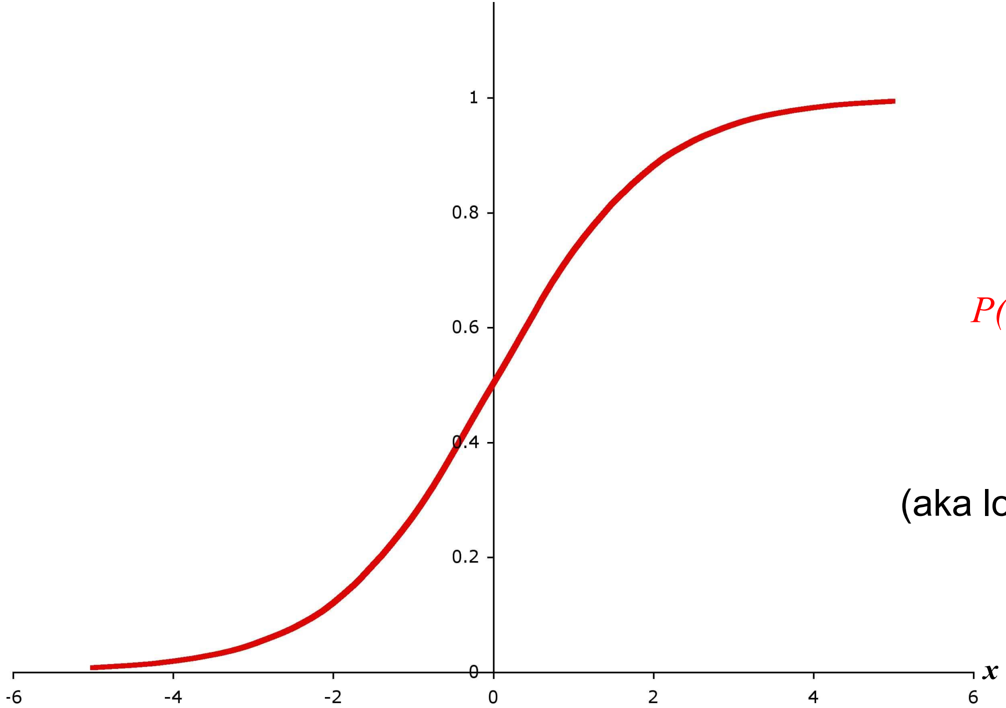
Backpropagation

Nonlinearity Functions

NNs in Practice


x
x1
x2
x3
W1
W2
w3
ŷ

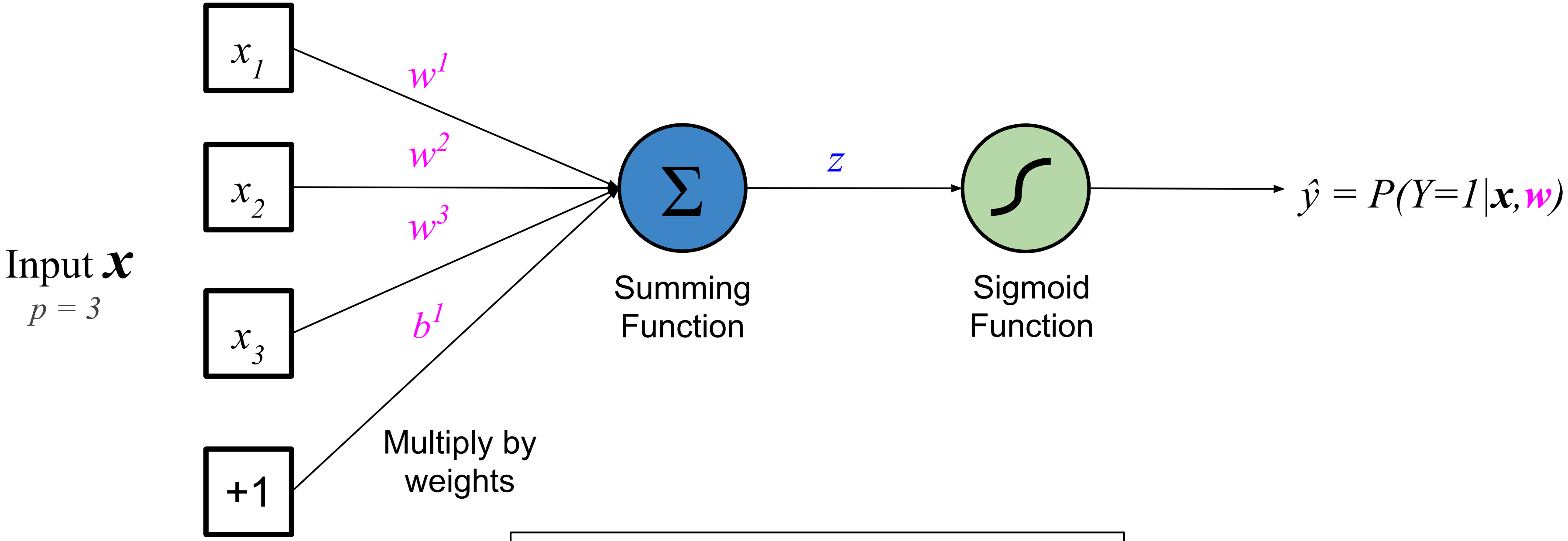
Logistic Regression



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

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

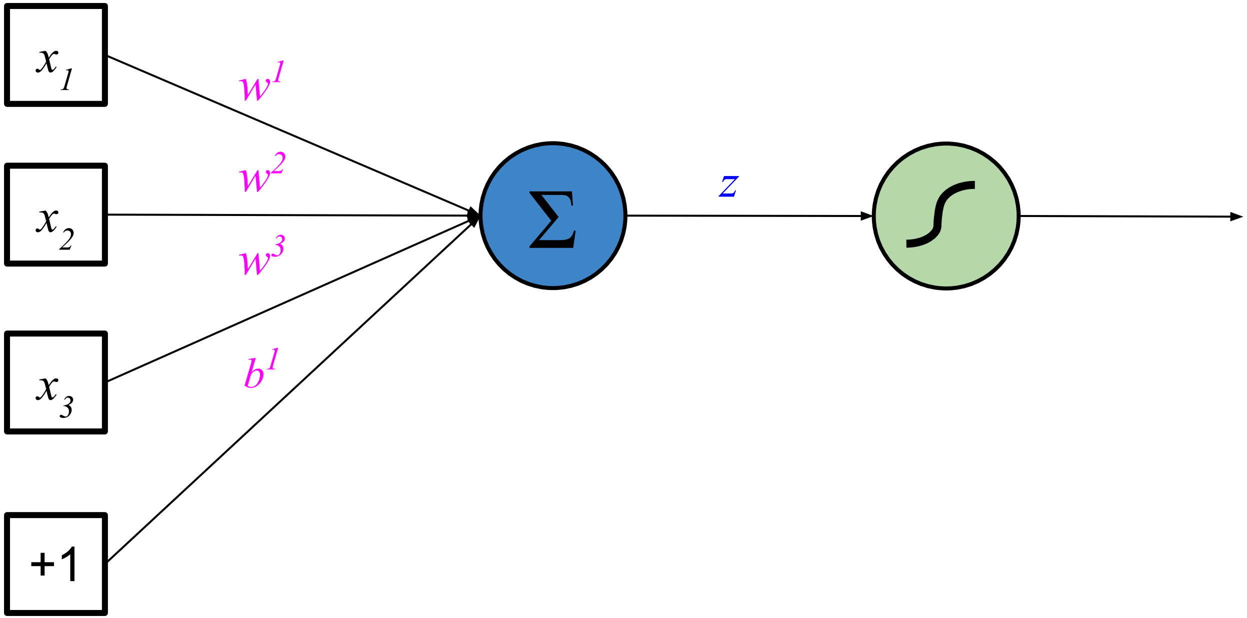
Expanded Logistic Regression



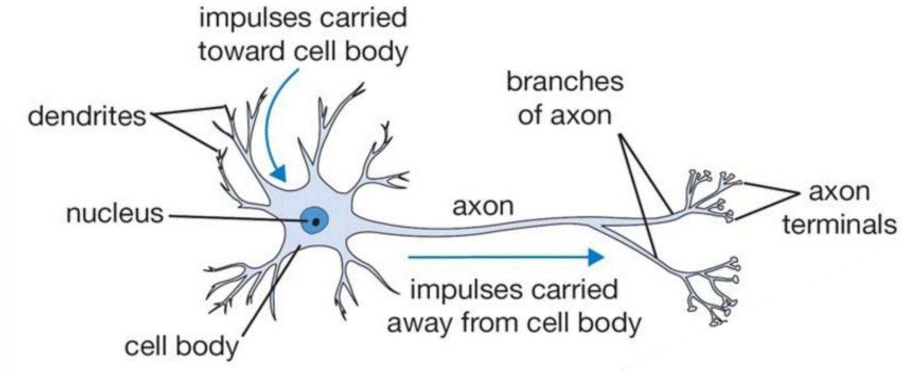
$$\underset{1\text{x}1}{z} = \underset{1\text{x}p}{\boldsymbol{w}^T} \underset{p\text{x}1}{\boldsymbol{x}} + \boldsymbol{b}$$

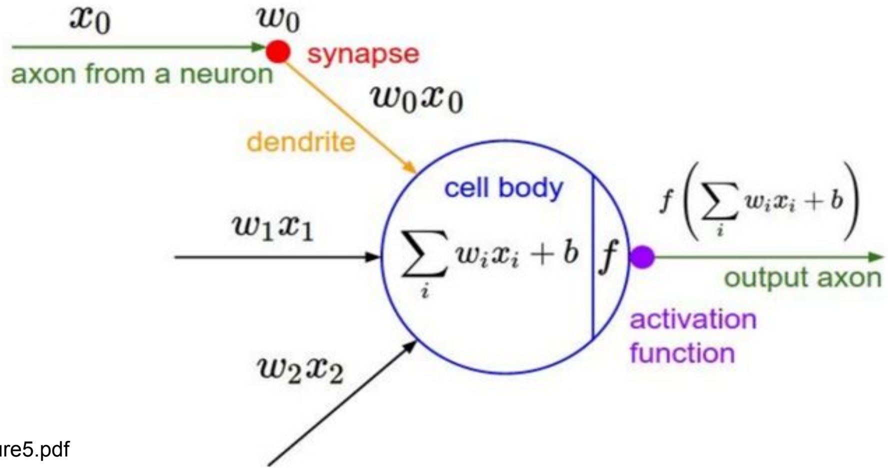
$$y = sigmoid(z) = \frac{e^z}{1 + e^z}$$

“Neuron”



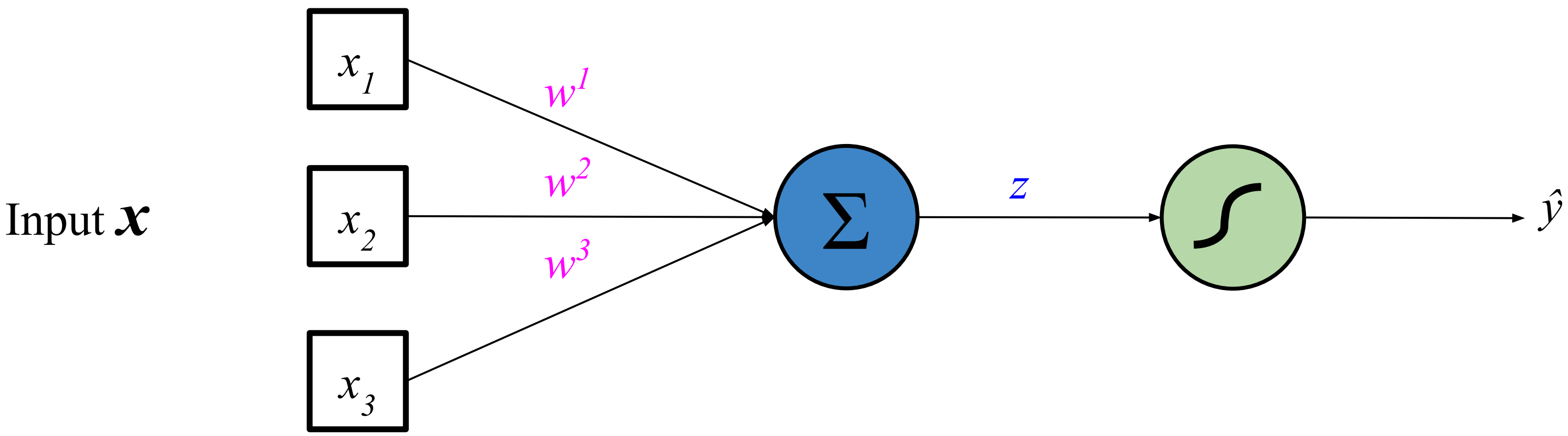
Neurons





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

Neuron

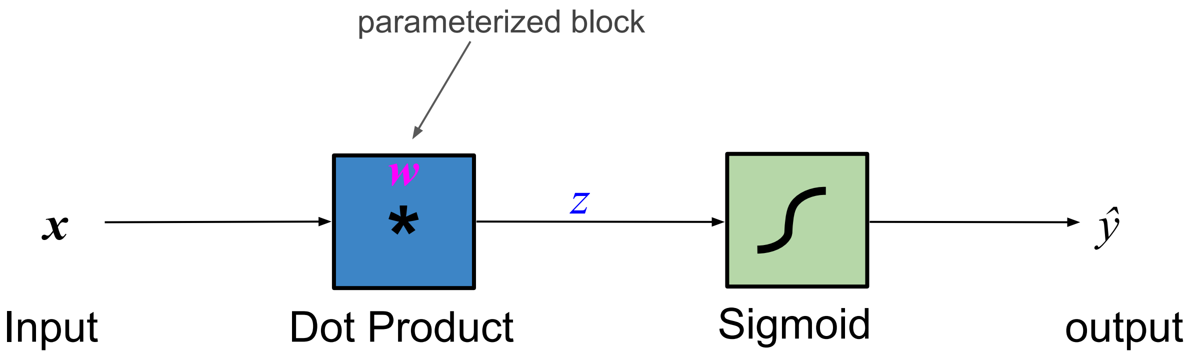


From here on, we leave out bias for simplicity

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

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

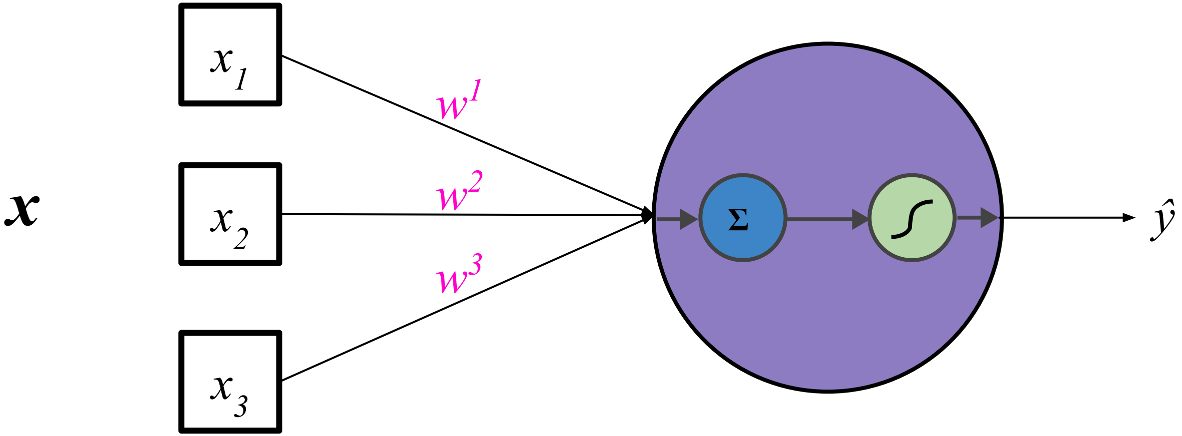
“Block View” of a Neuron



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

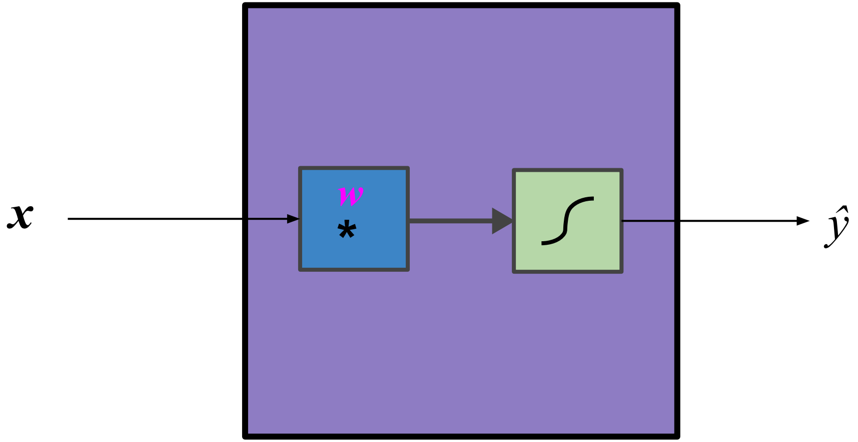
$$\hat{y} = sigmoid(z) = \frac{e^z}{1 + e^z}$$

Neuron Representation



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

Neuron Representation



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

Neurons

1-Layer Neural Network

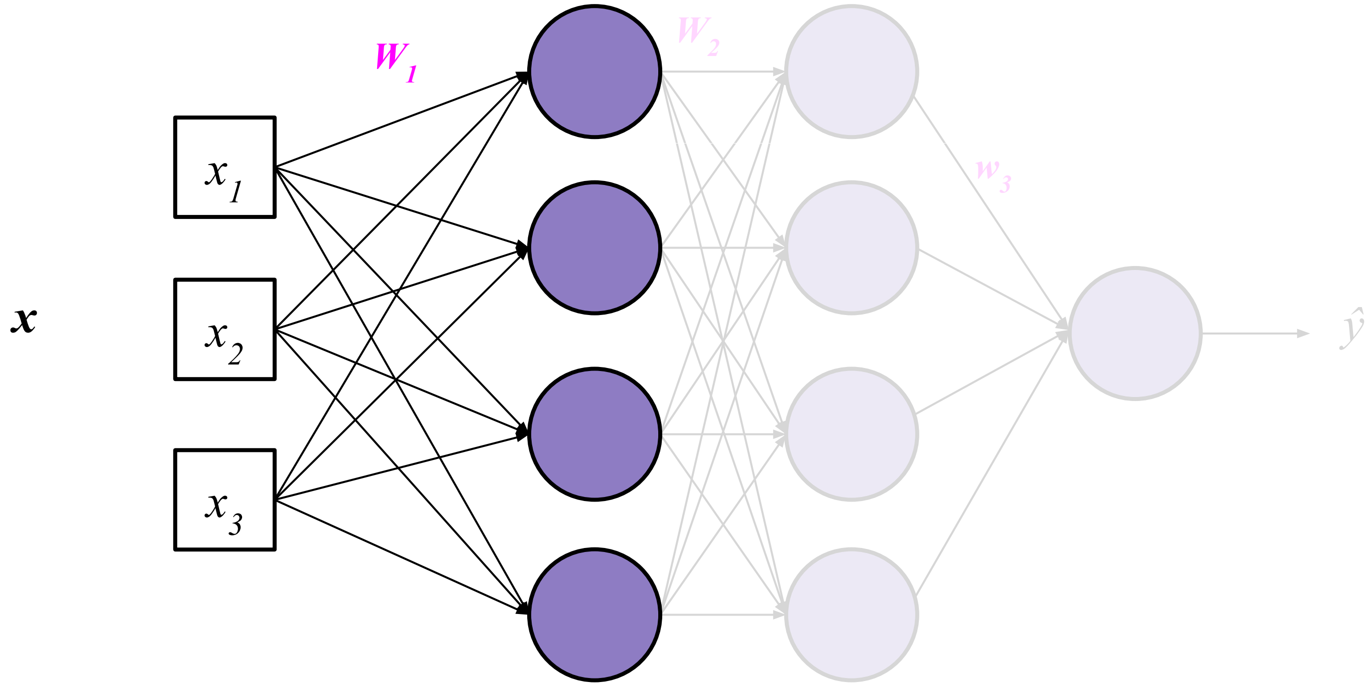
Multi-layer Neural Network

Loss Functions

Backpropagation

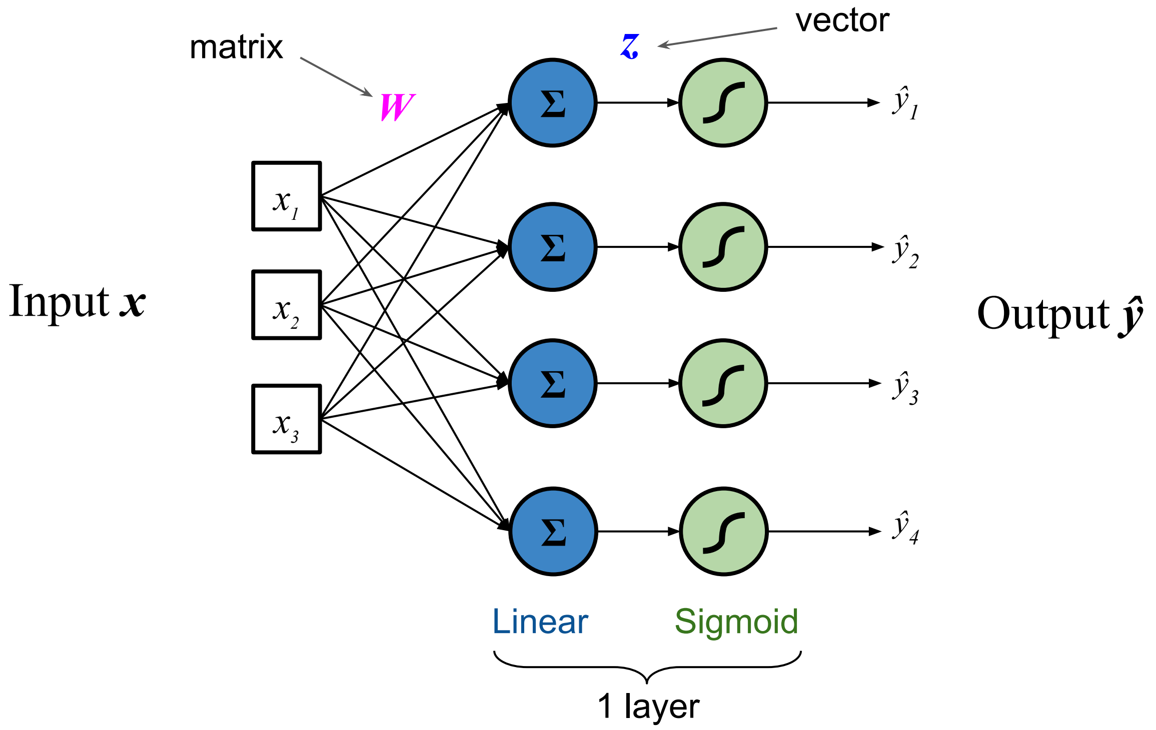
Nonlinearity Functions

NNs in Practice

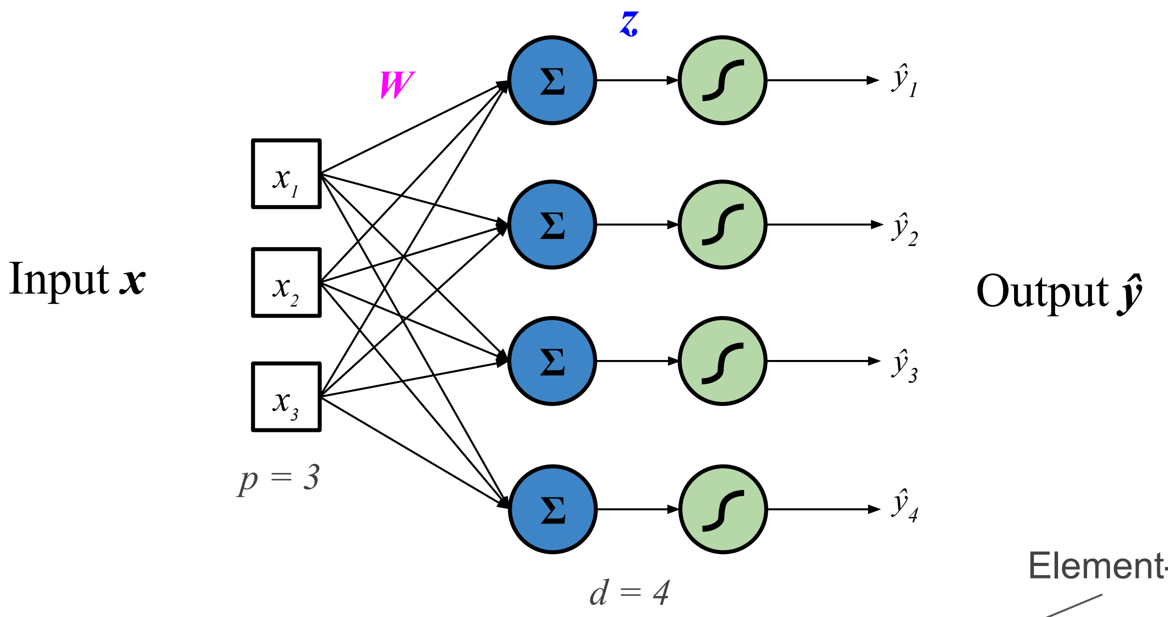


$\boldsymbol{x}$
x_1
x_2
x_3
W_1
W_2
w_3
$\hat{y}$

1-Layer Neural Network (with 4 neurons)



1-Layer Neural Network (with 4 neurons)

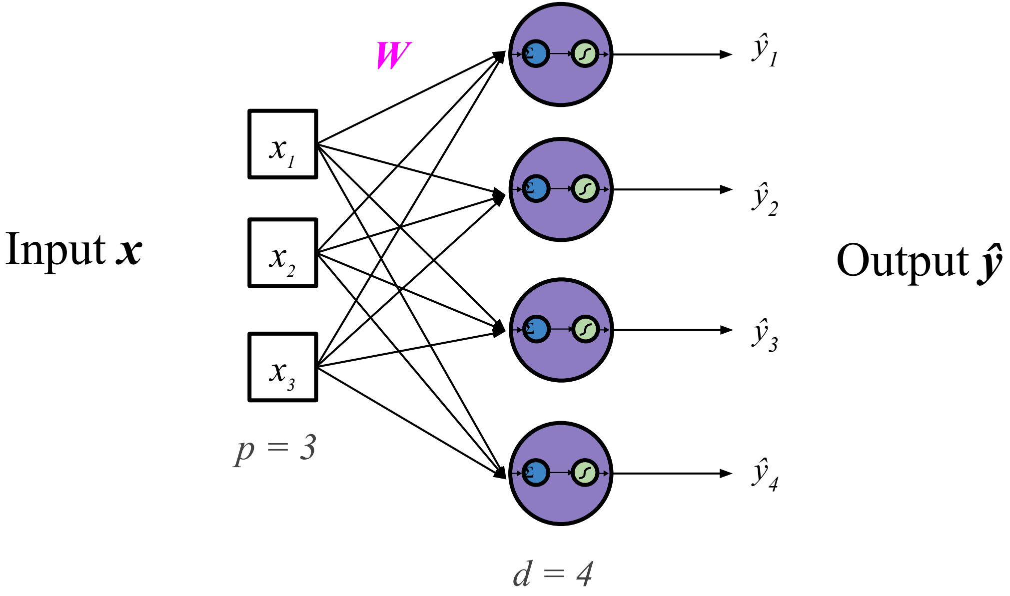


Element-wise on vector z

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

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

1-Layer Neural Network (with 4 neurons)

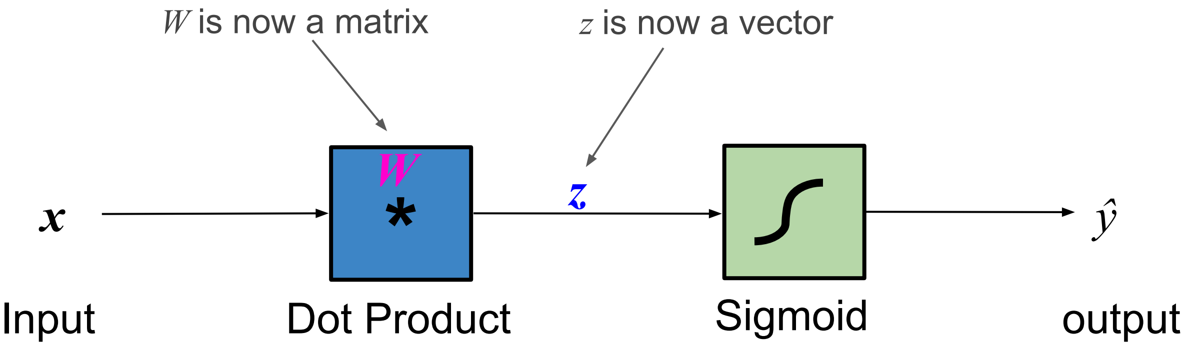


Element-wise on vector z

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

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

“Block View” of a Neural Network



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

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

Neurons

1-Layer Neural Network

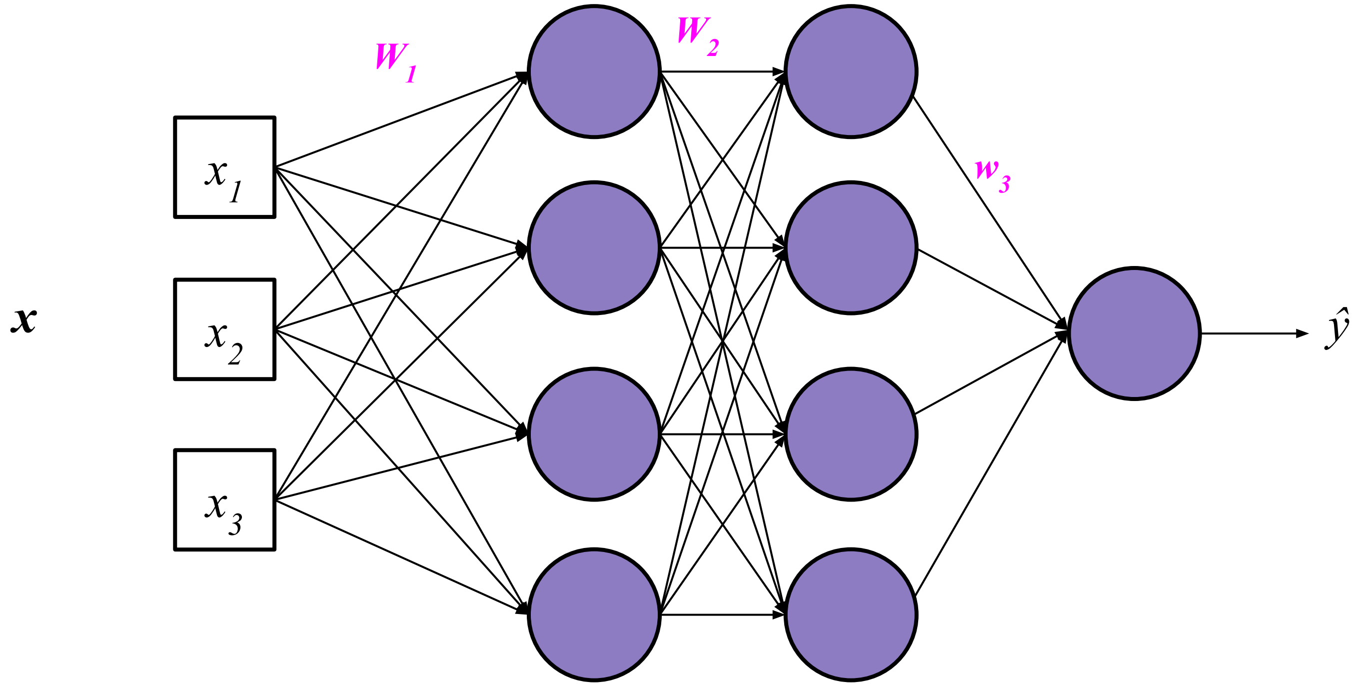
Multi-layer Neural Network

Loss Functions

Backpropagation

Nonlinearity Functions

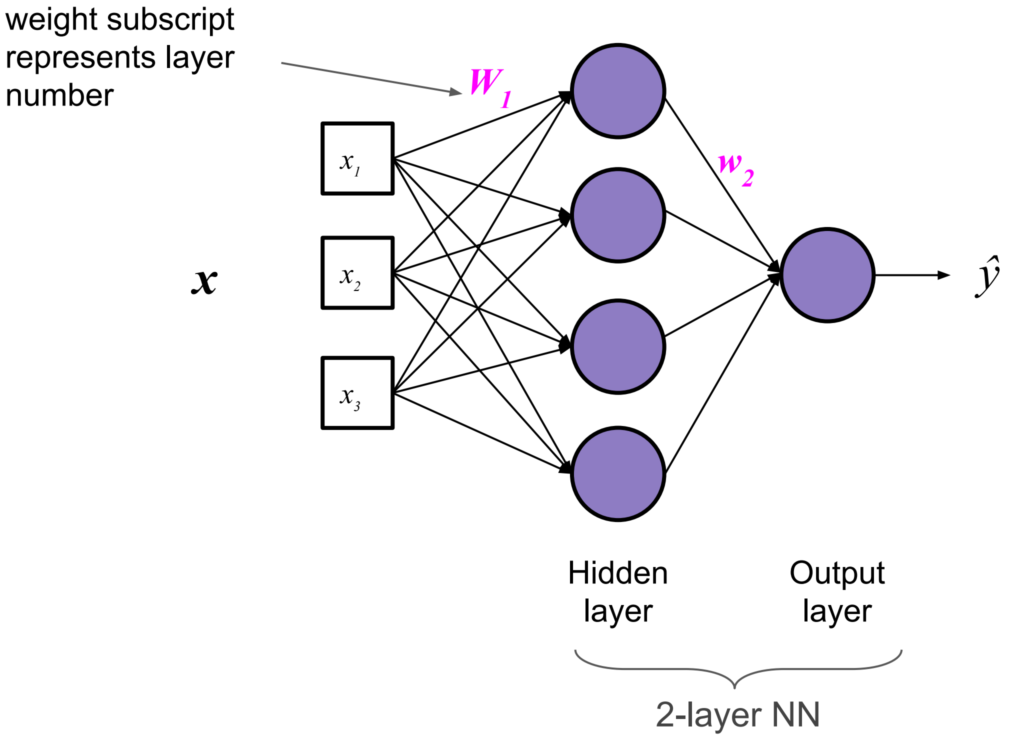
NNs in Practice



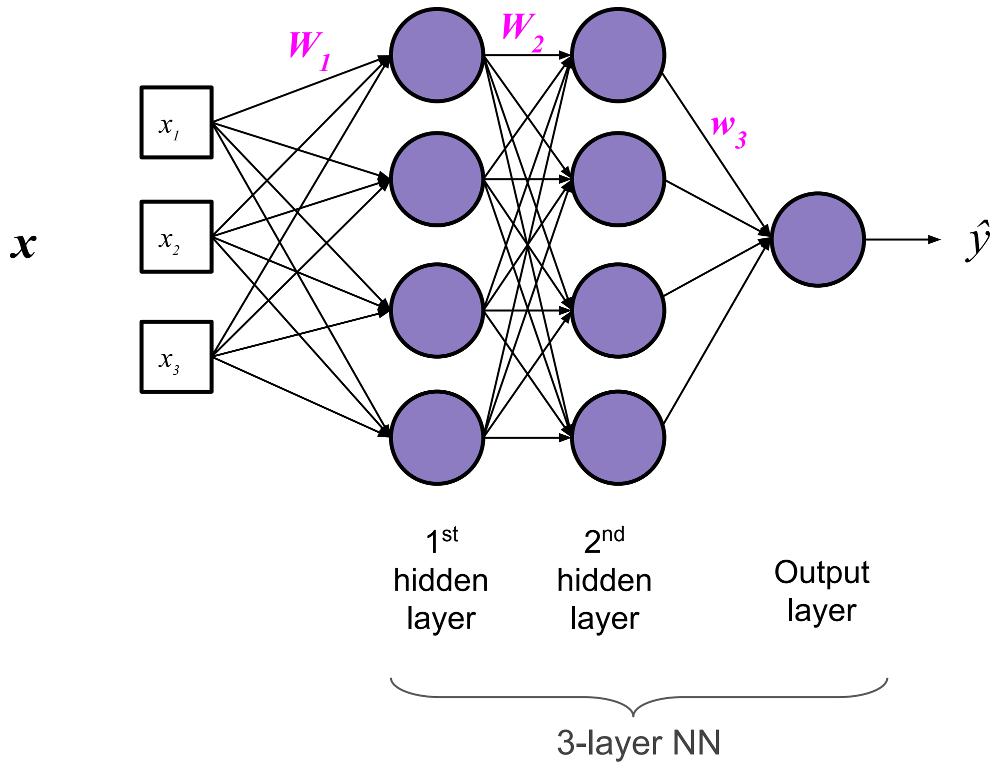
$\boldsymbol{x}$
x_1
x_2
x_3
W_1
W_2
w_3
$\hat{y}$

Multi-Layer Neural Network

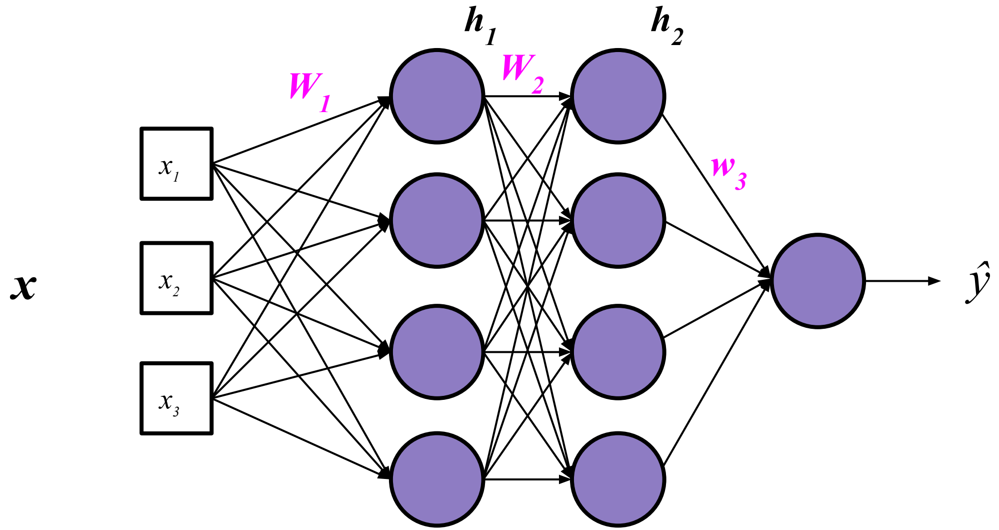
(Multi-Layer Perceptron (**MLP**) Network)



Multi-Layer Neural Network (MLP)



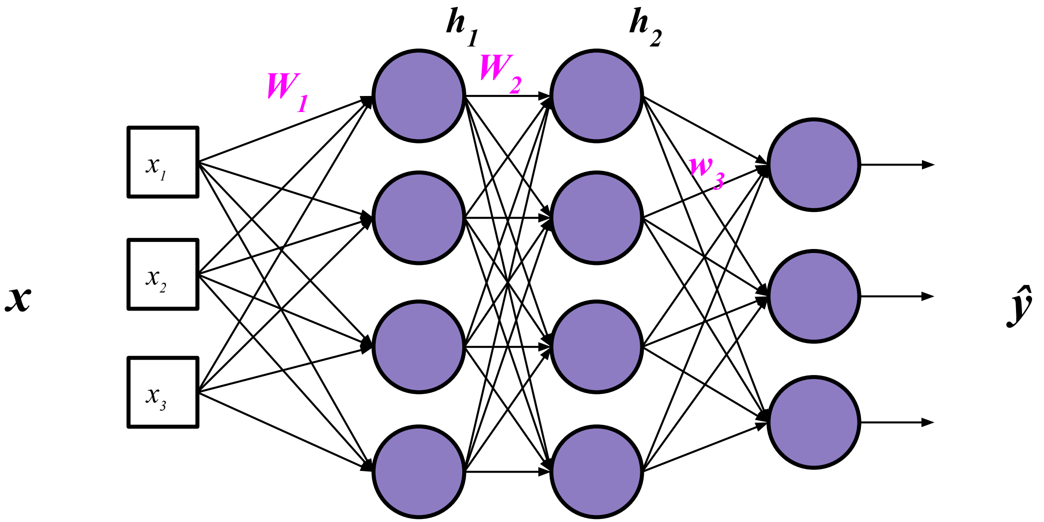
Multi-Layer Neural Network (MLP)



hidden layer 1 output

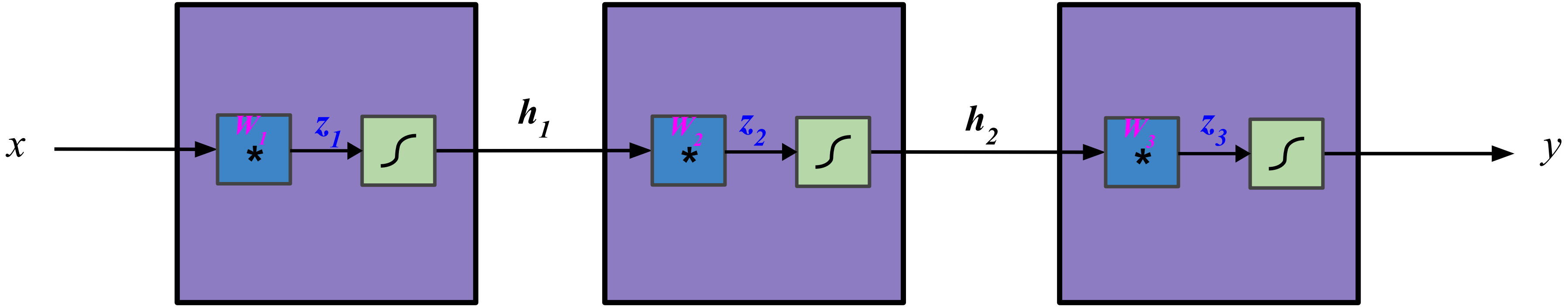
$$\boldsymbol{z}_1 = \boldsymbol{W}_1^T \boldsymbol{x}$$
$$\boldsymbol{h}_1 = sigmoid(\boldsymbol{z}_1)$$
$$\boldsymbol{z}_2 = \boldsymbol{W}_2^T \boldsymbol{h}_1$$
$$\boldsymbol{h}_2 = sigmoid(\boldsymbol{z}_2)$$
$$\boldsymbol{z}_3 = \boldsymbol{w}_3^T \boldsymbol{h}_2$$
$$\hat{y} = sigmoid(\boldsymbol{z}_3)$$

Multi-Class Output MLP



$$
\begin{aligned}
z_1 &= W_1^T x \\
h_1 &= sigmoid(z_1) \\
z_2 &= W_2^T h_1 \\
h_2 &= sigmoid(z_2) \\
z_3 &= W_3^T h_2 \\
\hat{y} &= sigmoid(z_2)
\end{aligned}
$$

"Block View" Of MLP

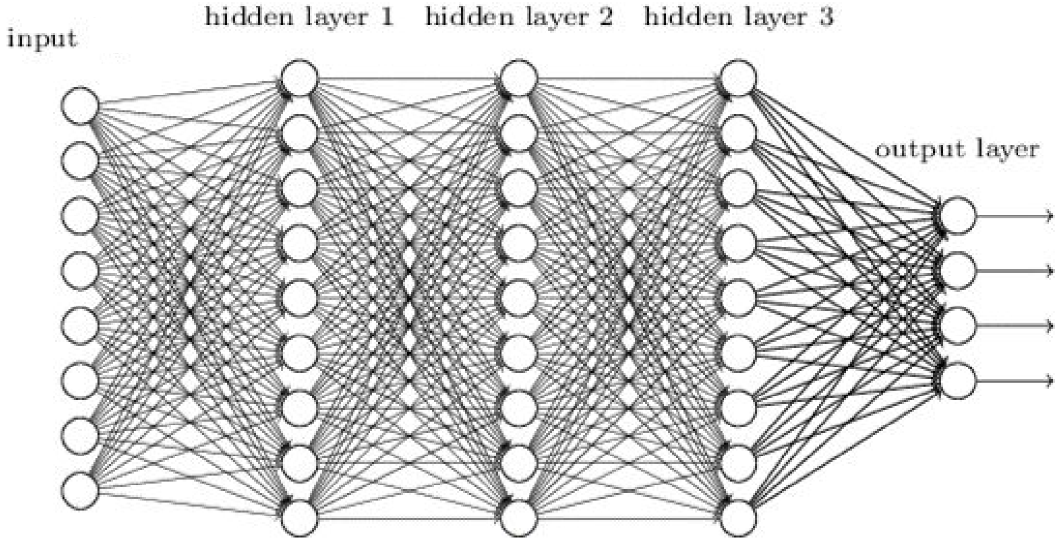


1st
hidden
layer

2nd
hidden
layer

Output
layer

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



Researchers have successfully used 1000 layers to train an object classifier

Neurons

1-Layer Neural Network

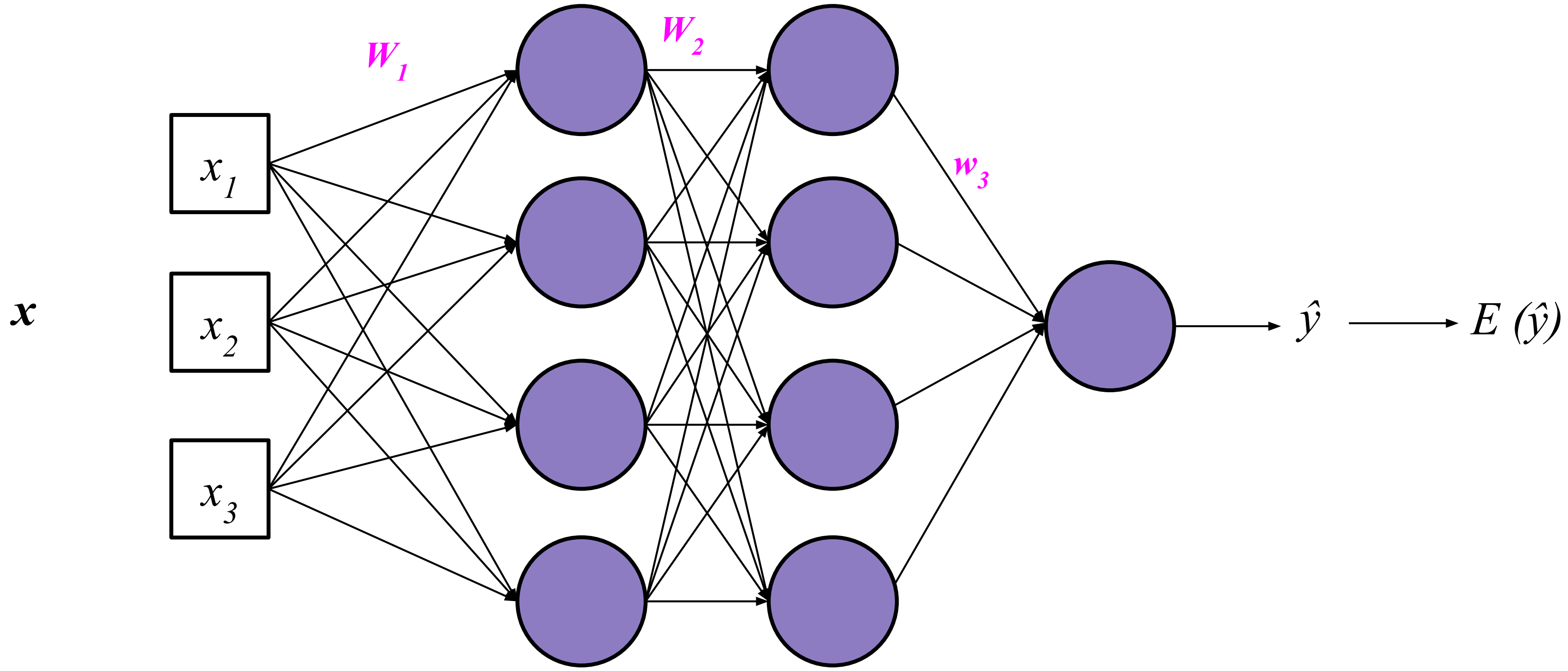
Multi-layer Neural Network

Loss Functions

Backpropagation

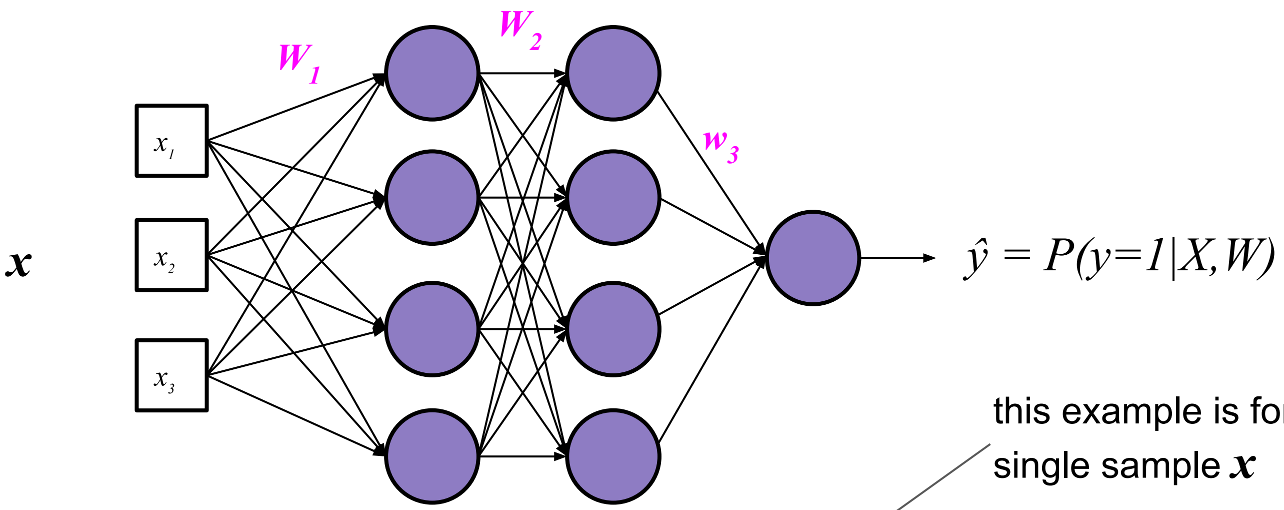
Nonlinearity Functions

NNs in Practice



$\boldsymbol{x}$
x_1
x_2
x_3
W_1
W_2
w_3
$\hat{y}$
$E(\hat{y})$

Binary Classification Loss

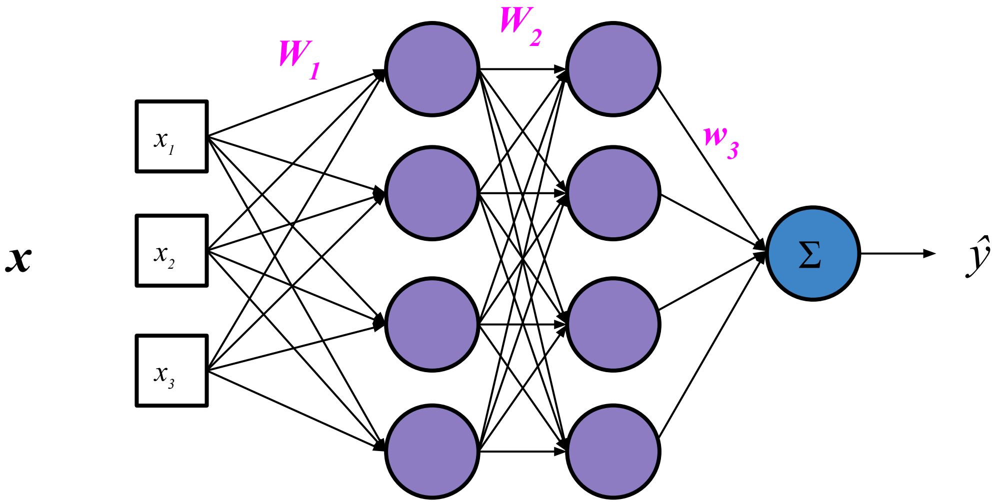


this example is for a single sample $\boldsymbol{x}$

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

true output

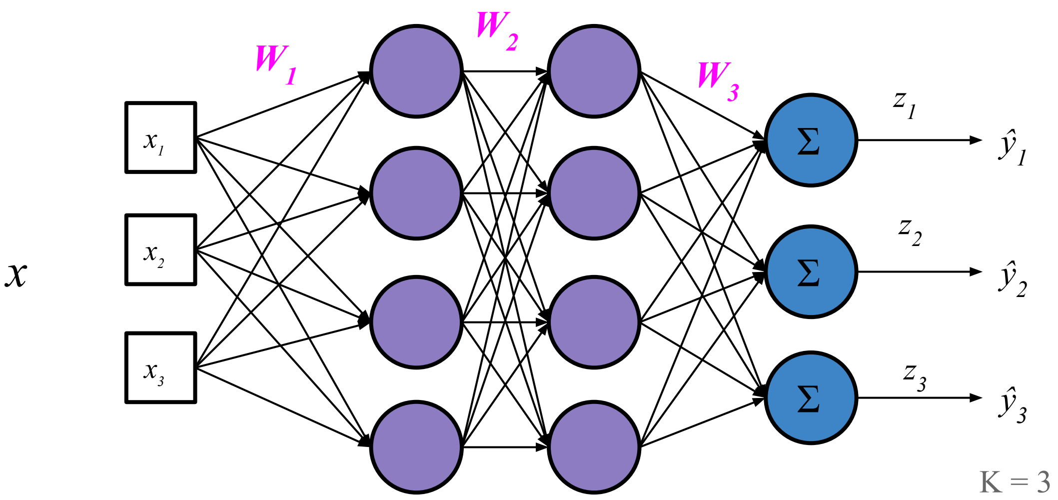
Regression Loss



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

true output

Multi-Class Classification Loss



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

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

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

"0" for all except true class

Neurons

1-Layer Neural Network

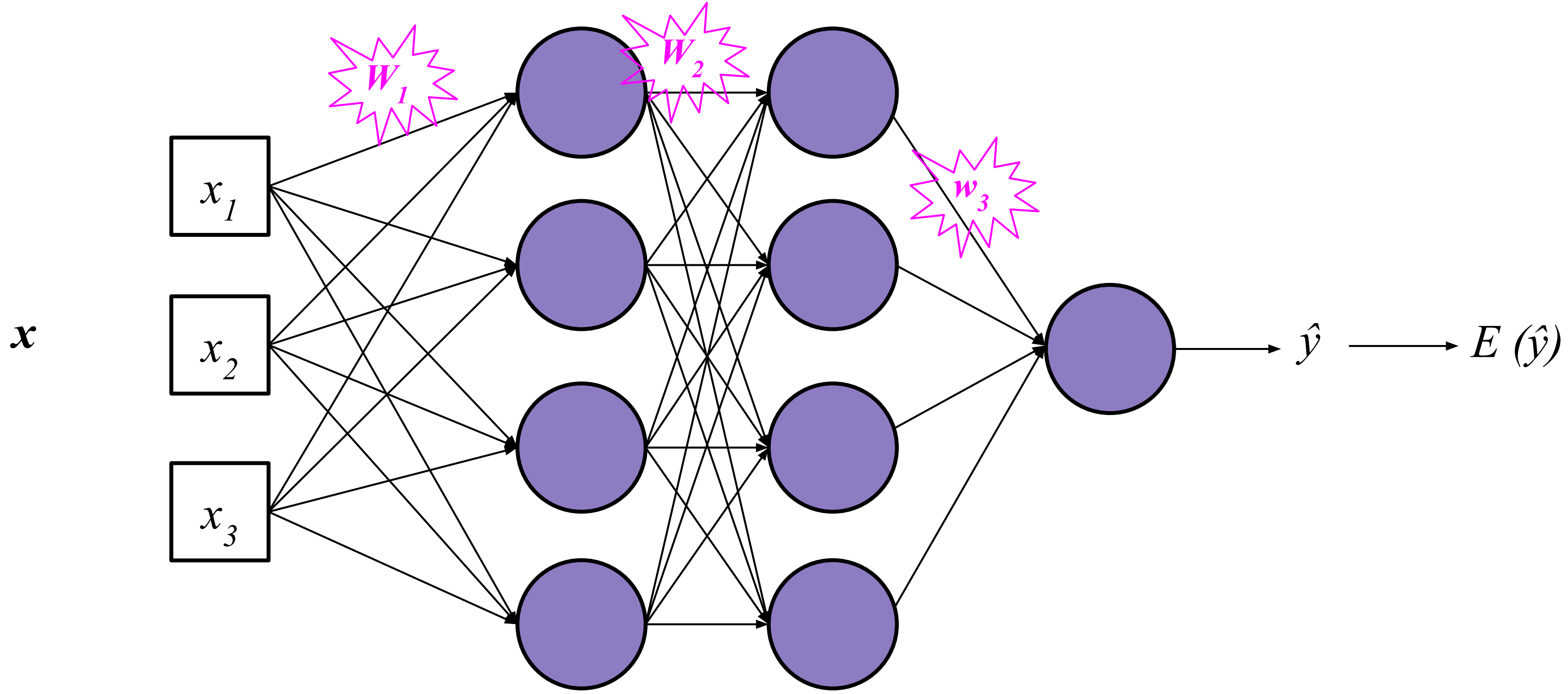
Multi-layer Neural Network

Loss Functions

Backpropagation

Nonlinearity Functions

NNs in Practice



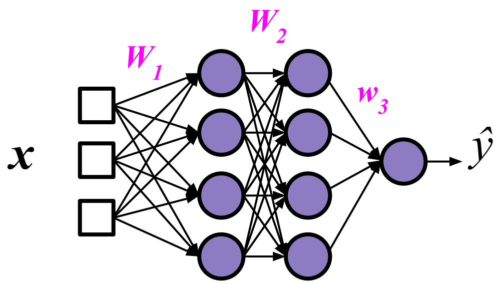
$\boldsymbol{x}$
x_1
x_2
x_3
W_1
W_2
w_3
$\hat{y}$
$E\,(\hat{y})$

Training Neural Networks

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

- **Gradient descent:**

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



But how do we get gradients of lower layers?

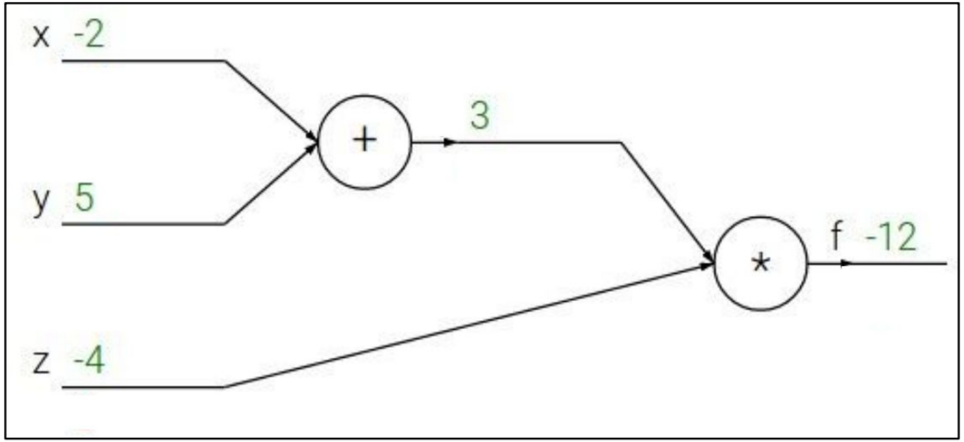
- **Backpropagation!**
  - Repeated application of chain rule of calculus
  - Locally minimize the objective
  - Requires all “blocks” of the network to be differentiable

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

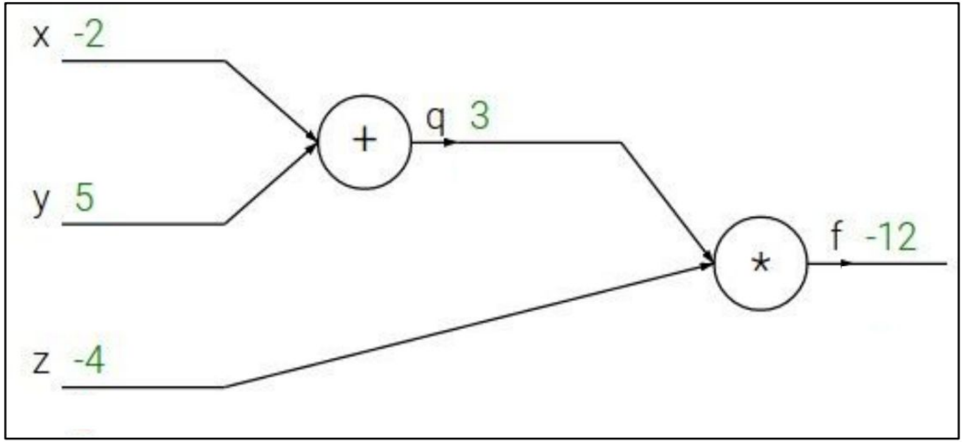
Backpropagation Intro

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

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

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

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



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

Backpropagation Intro

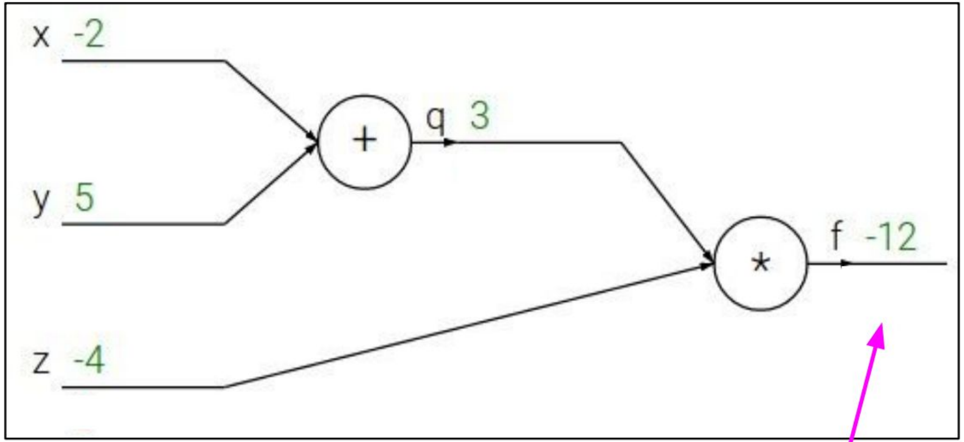
$$f(x, y, z) = (x + y)z$$

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



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

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

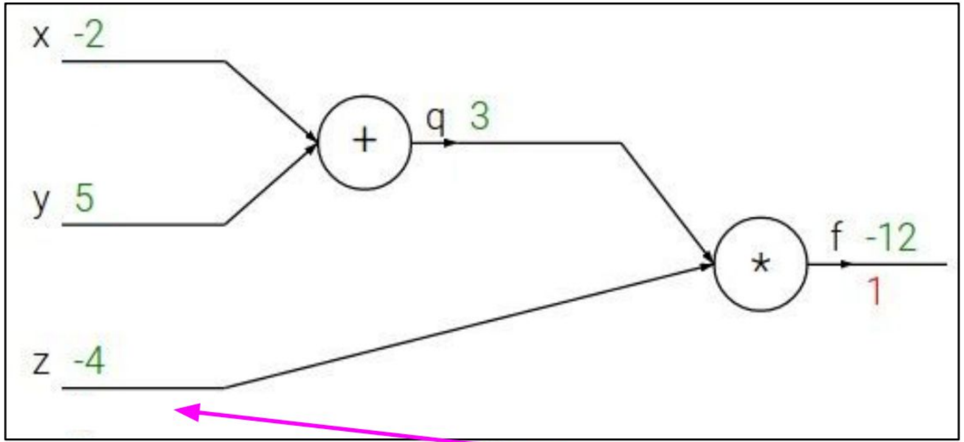
Backpropagation Intro

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

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

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

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



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

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

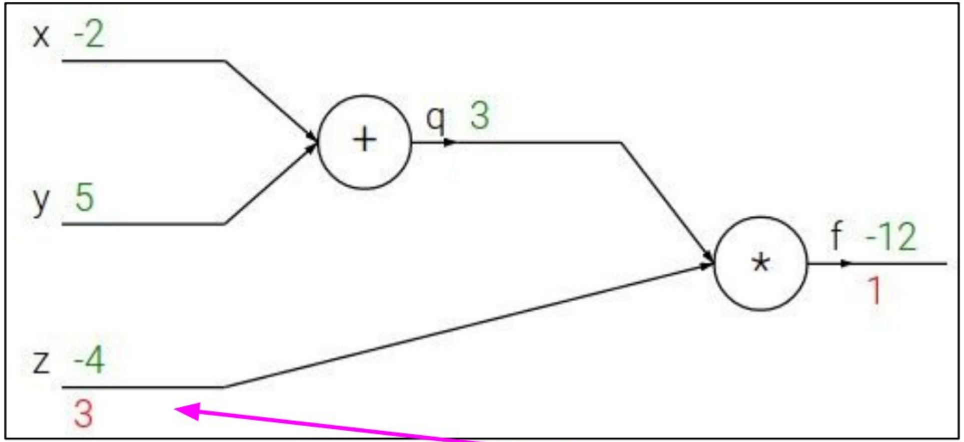
$$f(x, y, z) = (x + y)z$$

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



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

Backpropagation Intro

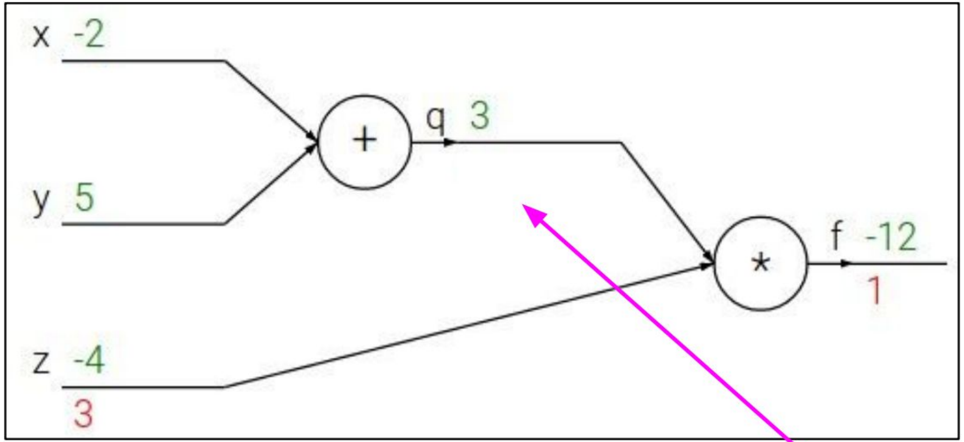
$$f(x, y, z) = (x + y)z$$

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



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

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

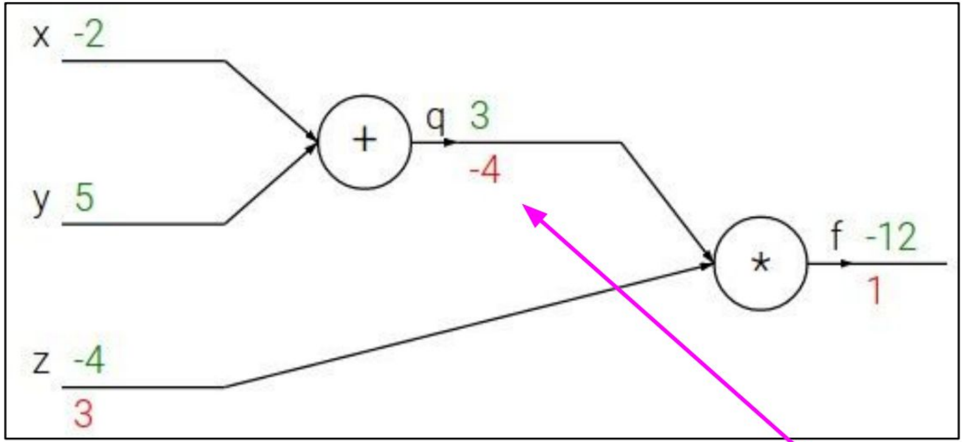
Backpropagation Intro

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

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

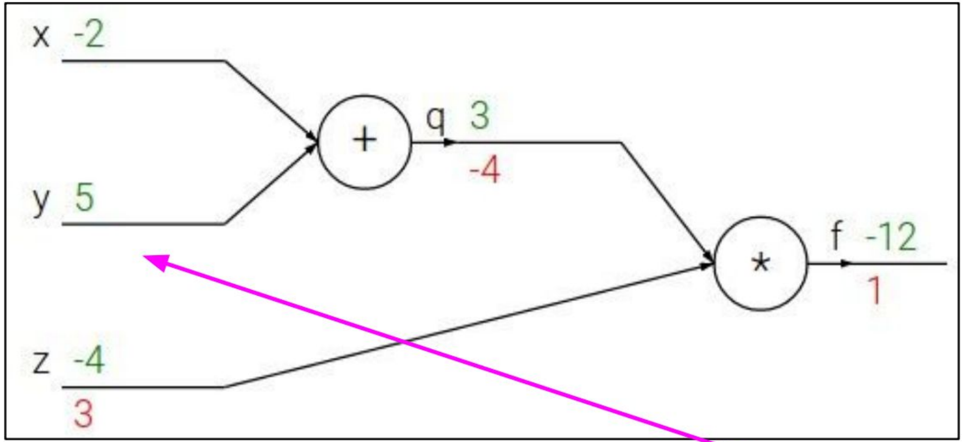
$$f(x, y, z) = (x + y)z$$

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



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

Backpropagation Intro

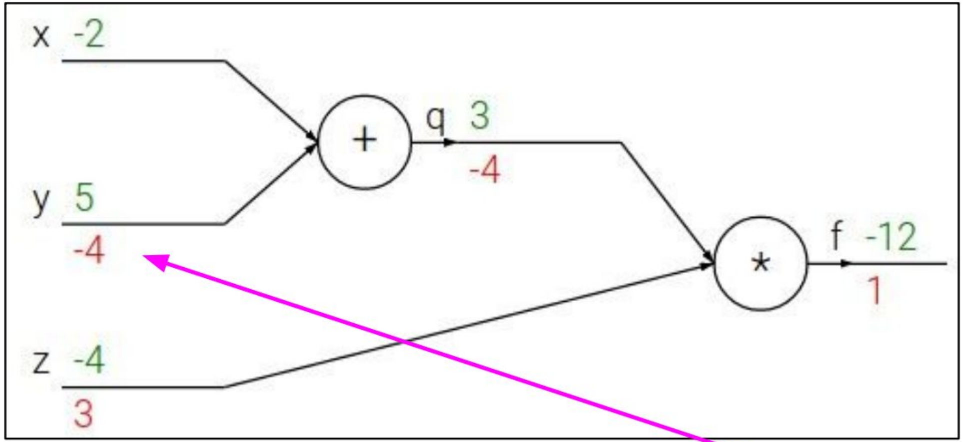
$$f(x,y,z) = (x+y)z$$

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:

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

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

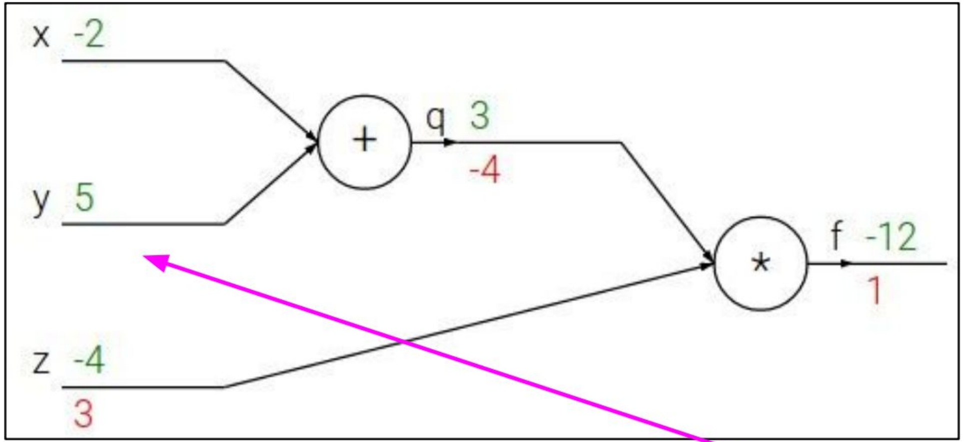
Backpropagation Intro

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

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

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

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



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

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

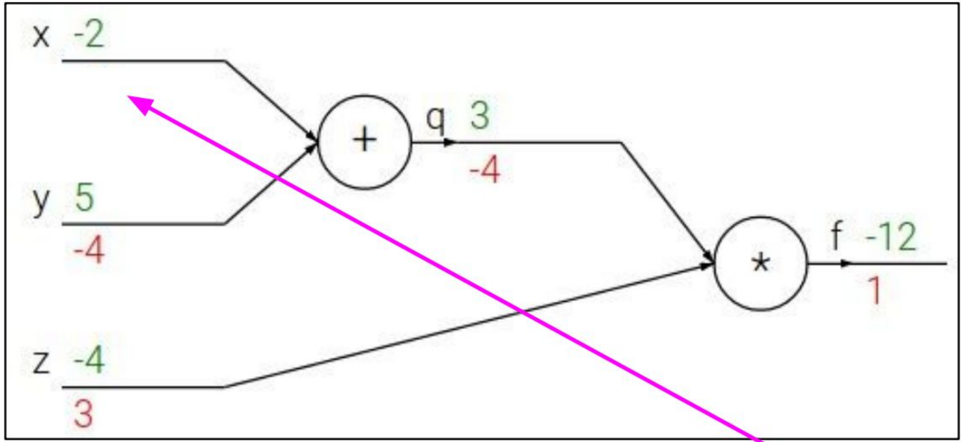
$$f(x, y, z) = (x + y)z$$

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

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

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

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

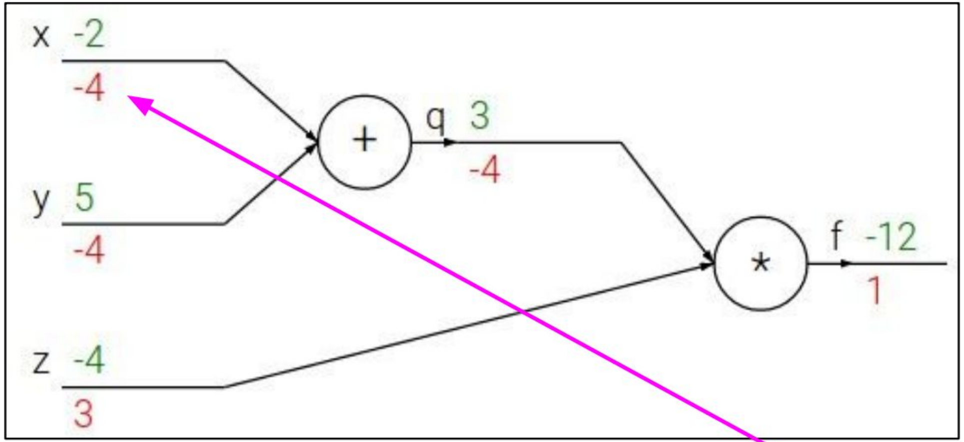
$$f(x,y,z) = (x+y)z$$

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

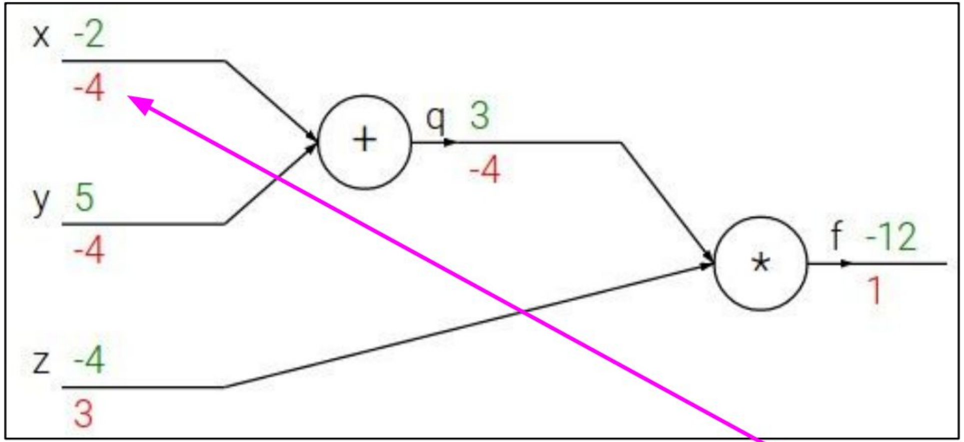
$$f(x, y, z) = (x + y)z$$

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



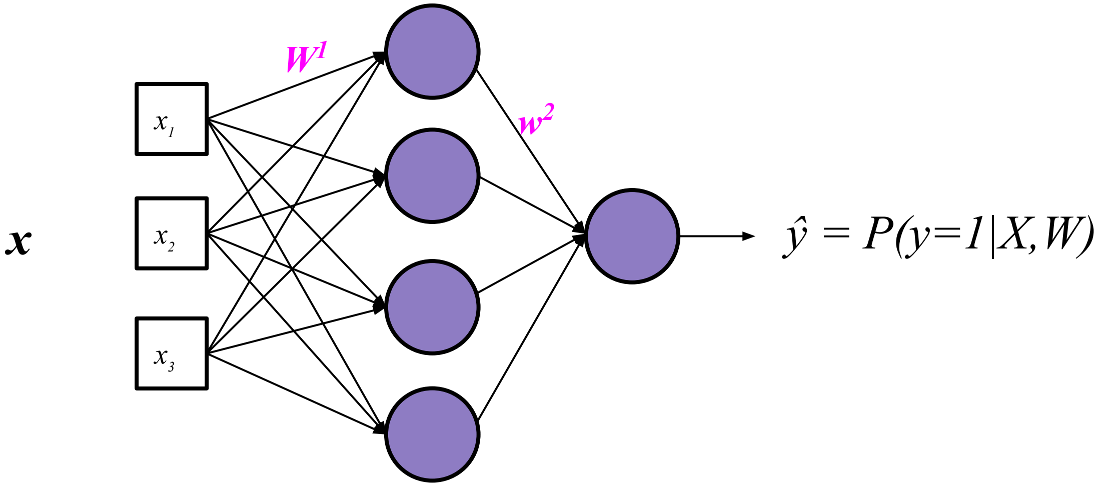
$$\boxed{\frac{\partial f}{\partial x}}$$

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

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

Backpropagation

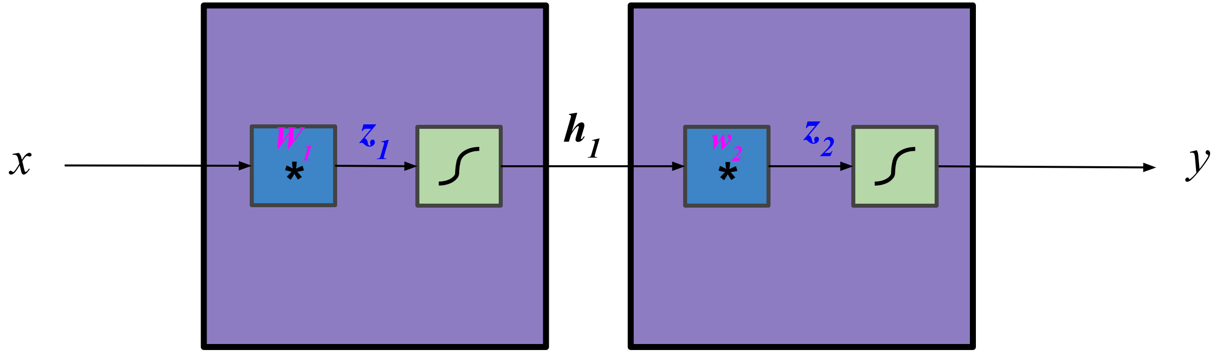
(binary classification example)



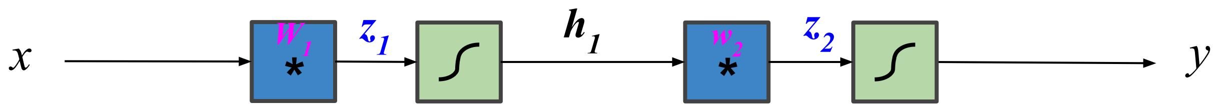
Example on 1-hidden layer NN for binary classification

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:

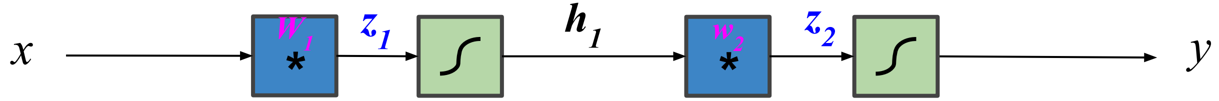
$$\mathbf{w}_2(t+1) = \mathbf{w}_2(t) - \eta \boxed{\frac{\partial E}{\partial \mathbf{w}_2(t)}}$$

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

Need to find these!

Backpropagation

(binary classification example)



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

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

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

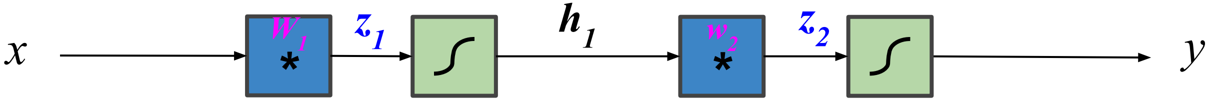
$$\boldsymbol{h}_1 = f_2 = \frac{e^{z_1}}{1+e^{z_1}}$$

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

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

Backpropagation

(binary classification example)



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

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

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

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

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

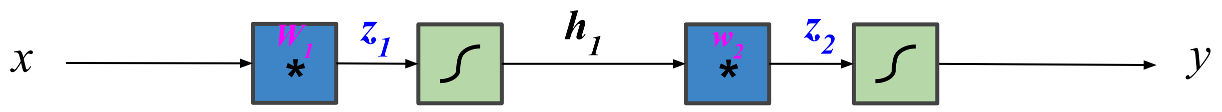
$$\boxed{\frac{\partial E}{\partial \mathbf{w}_2} = ??}$$

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

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

Backpropagation

(binary classification example)



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

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

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

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

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

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

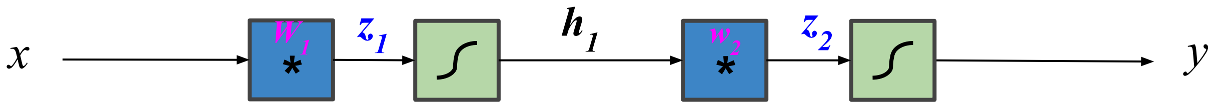
$$\boxed{\frac{\partial E}{\partial \mathbf{W}_1} = ??}$$

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

Exploit the chain rule!

Backpropagation

(binary classification example)

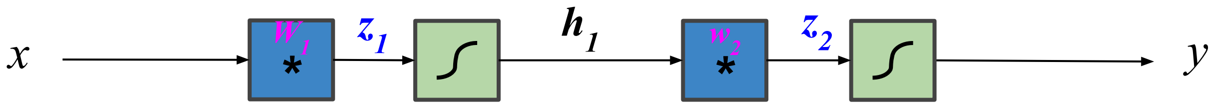


$$
\begin{aligned}
E &= -y\ln(\hat{y}) \\
&\quad -(1-y)\ln(1-\hat{y}) \\
\hat{y} &= \frac{e^{z_2}}{1+e^{z_2}} \\
z_2 &= \boxed{\boldsymbol{w}_2^T}\boldsymbol{h}_1 \\
\boldsymbol{h}_1 &= \frac{e^{z_1}}{1+e^{z_1}} \\
\boldsymbol{z}_1 &= W_1^T\boldsymbol{x}
\end{aligned}
$$

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

Backpropagation

(binary classification example)



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

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

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

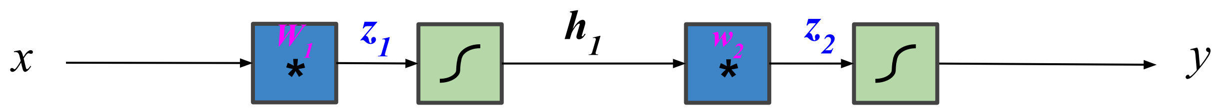
$$\boldsymbol{h}_1 = \frac{e^{z_1}}{1+e^{z_1}}$$

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

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

Backpropagation

(binary classification example)



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

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

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

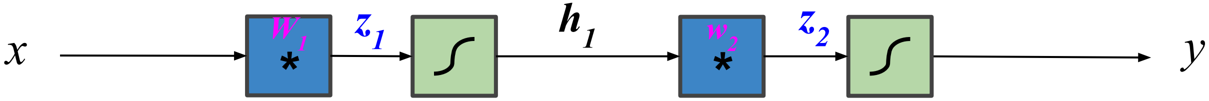
$$
\boldsymbol{h}_1 = \frac{e^{z_1}}{1+e^{z_1}}
$$

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

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

Backpropagation

(binary classification example)



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

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

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

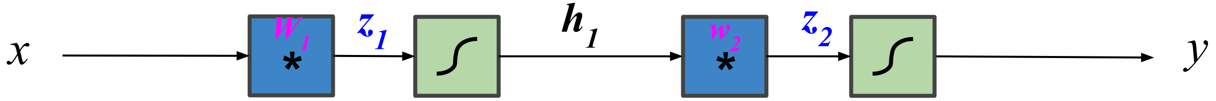
$$\boldsymbol{h}_1 = \frac{e^{\boldsymbol{z}_1}}{1+e^{\boldsymbol{z}_1}}$$

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

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

Backpropagation

(binary classification example)



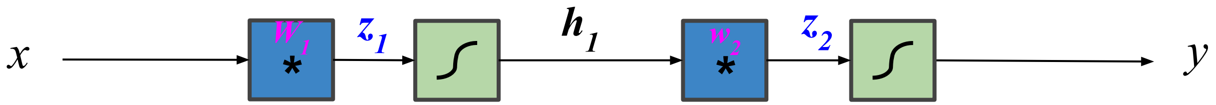
$$\boxed{\begin{aligned} E &= -y\ln(\hat{y}) \\ &\quad -(1-y)\ln(1-\hat{y}) \\ \hat{y} &= \frac{e^{z_2}}{1+e^{z_2}} \\ z_2 &= \boldsymbol{w}_2^T \boldsymbol{h}_1 \\ \boldsymbol{h}_1 &= \frac{e^{z_1}}{1+e^{z_1}} \\ \boldsymbol{z}_1 &= W_1^T \boldsymbol{x} \end{aligned}}$$

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

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

Backpropagation

(binary classification example)

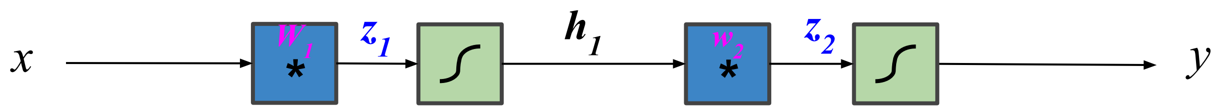


$$\begin{aligned}
E &= -y\ln(\hat{y}) \\
&\quad - (1-y)\ln(1-\hat{y}) \\
\hat{y} &= \frac{e^{z_2}}{1+e^{z_2}} \\
z_2 &= \boldsymbol{w}_2^T \cdot \boldsymbol{h}_1 \\
\boldsymbol{h}_1 &= \frac{e^{z_1}}{1+e^{z_1}} \\
\boldsymbol{z}_1 &= W_1^T \cdot \boldsymbol{x}
\end{aligned}$$

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

Backpropagation

(binary classification example)



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

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

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

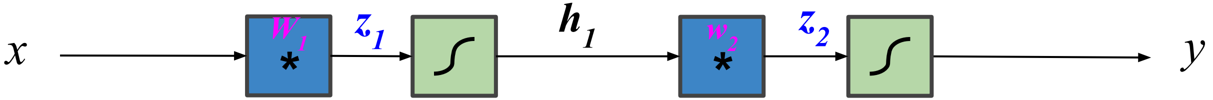
$$\boldsymbol{h}_1 = \frac{e^{z_1}}{1+e^{z_1}}$$

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

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

Backpropagation

(binary classification example)

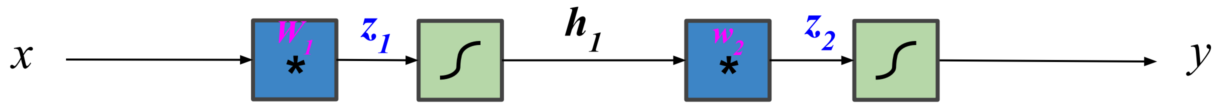


$$\begin{aligned} E &= -\,y\ln(\hat{y}) \\ &\quad -(1-y)\ln(1-\hat{y}) \\ \hat{y} &= \frac{e^{z_2}}{1+e^{z_2}} \\ z_2 &= \boldsymbol{w}_2^T\boldsymbol{h}_1 \\ \boldsymbol{h}_1 &= \frac{e^{z_1}}{1+e^{z_1}} \\ \boldsymbol{z}_1 &= \boxed{W_1^T}\boldsymbol{x} \end{aligned}$$

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

Backpropagation

(binary classification example)

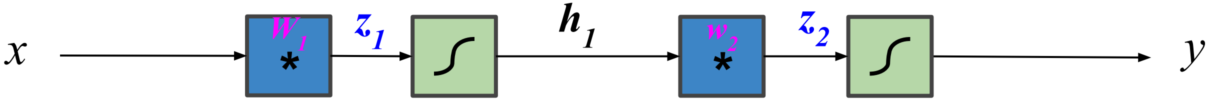


$$
\begin{aligned}
E &= -\,y\ln(\hat{y}) \\
&\quad -(1-y)\ln(1-\hat{y}) \\
\hat{y} &= \frac{e^{z_2}}{1+e^{z_2}} \\
z_2 &= \boldsymbol{w}_2^T \boldsymbol{h}_1 \\
\boldsymbol{h}_1 &= \frac{e^{z_1}}{1+e^{z_1}} \\
\boldsymbol{z}_1 &= W_1^T \boldsymbol{x}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial E}{\partial \boldsymbol{W}_1} &= \frac{\partial E}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z_2} \cdot \frac{\partial z_2}{\partial \boldsymbol{h}_1} \cdot \frac{\partial \boldsymbol{h}_1}{\partial \boldsymbol{z}_1} \cdot \frac{\partial \boldsymbol{z}_1}{\partial W_1} \\
&= \left(\frac{\hat{y}-y}{\hat{y}(1-\hat{y})} \right) \cdot (\hat{y}(1-\hat{y})) \cdot (\boldsymbol{w}) \cdot (\boldsymbol{h}_1(1-\boldsymbol{h}_1)) \cdot (x)
\end{aligned}
$$

Backpropagation

(binary classification example)

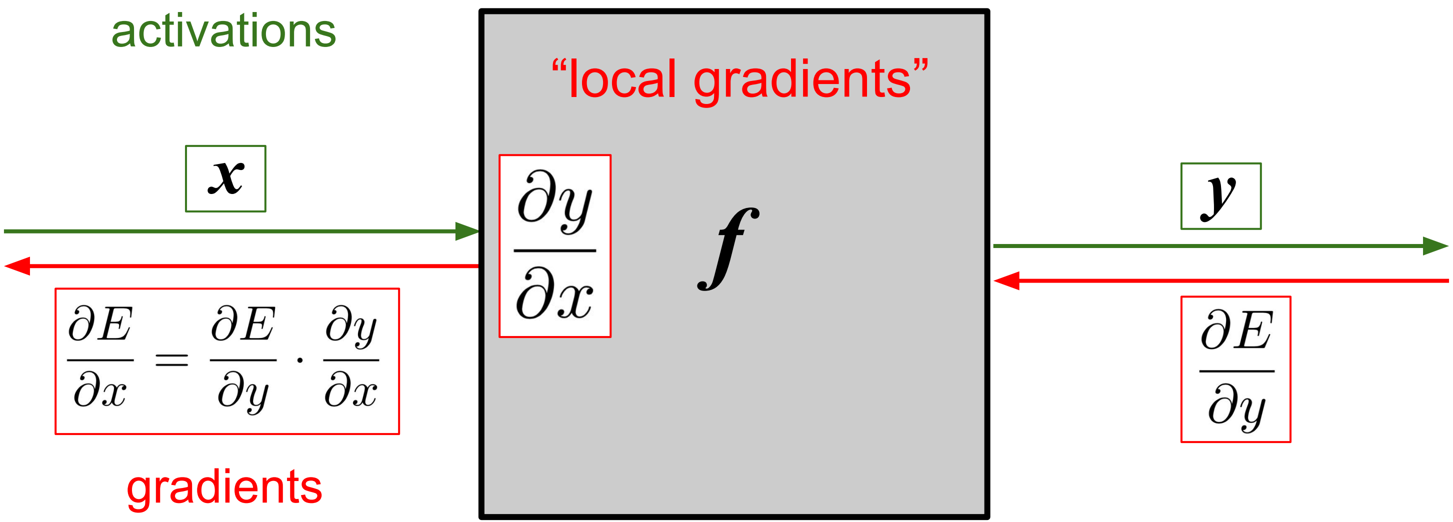


$$
\begin{aligned}
E = &-y\ln(\hat{y}) \\
&-(1-y)\ln(1-\hat{y}) \\
\hat{y} = &\frac{e^{z_2}}{1+e^{z_2}} \\
z_2 = &\boldsymbol{w}_2^T\boldsymbol{h}_1 \\
\boldsymbol{h}_1 = &\frac{e^{z_1}}{1+e^{z_1}} \\
\boldsymbol{z}_1 = &W_1^T\boldsymbol{x}
\end{aligned}
$$

already computed

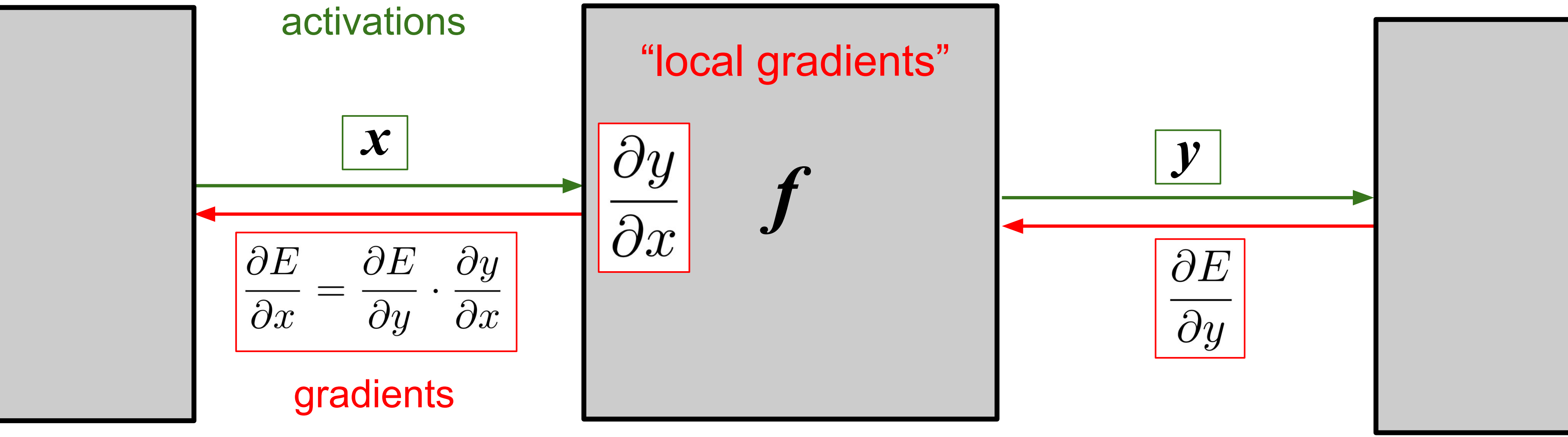
$$
\begin{aligned}
\frac{\partial E}{\partial \boldsymbol{W}_1} &= \boxed{\frac{\partial E}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z_2}} \cdot \frac{\partial z_2}{\partial \boldsymbol{h}_1} \cdot \frac{\partial \boldsymbol{h}_1}{\partial \boldsymbol{z}_1} \cdot \frac{\partial \boldsymbol{z}_1}{\partial W_1} \\
&= \left(\frac{\hat{y}-y}{\hat{y}(1-\hat{y})}\right) \cdot (\hat{y}(1-\hat{y})) \cdot (\boldsymbol{w}) \cdot (\boldsymbol{h}_1(1-\boldsymbol{h}_1)) \cdot (x)
\end{aligned}
$$

“Local-ness” of Backpropagation



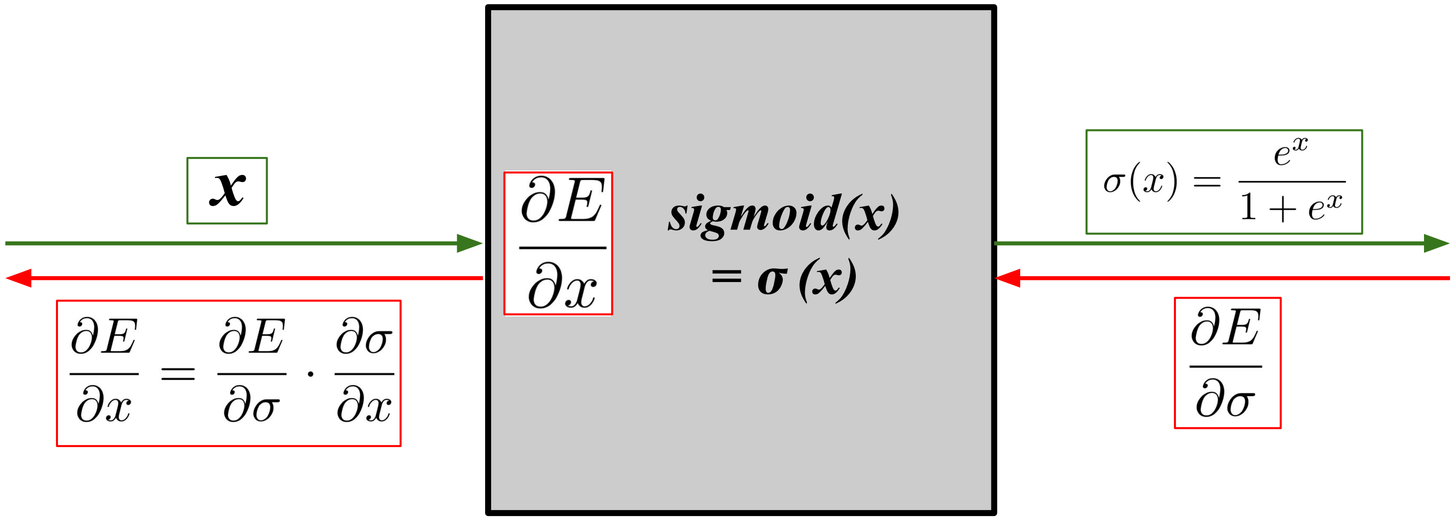
http://cs231n.stanford.edu/slides/winter1516_lecture5.pdf

“Local-ness” of Backpropagation



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

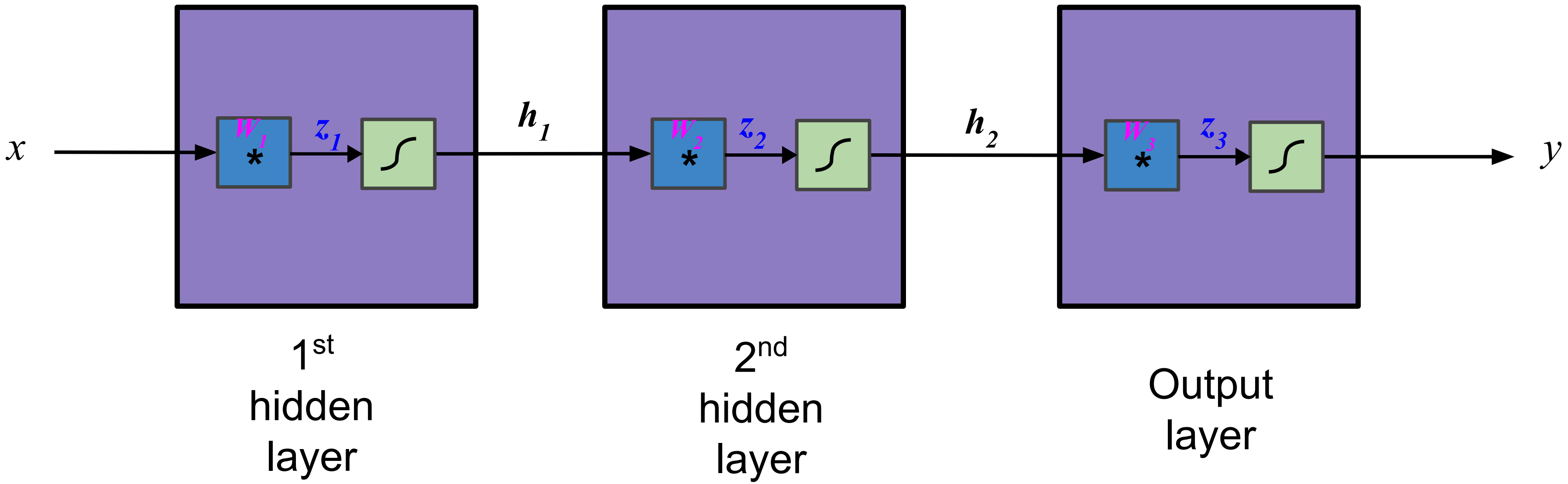
Example: Sigmoid Block



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

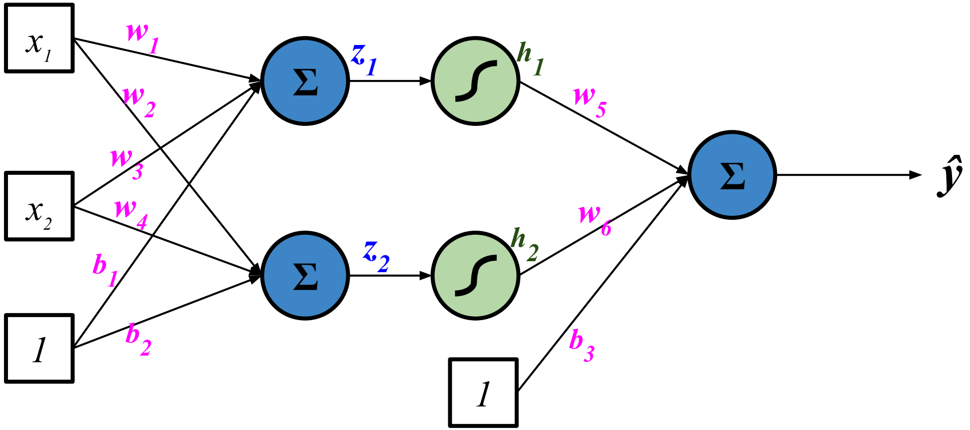
Deep Learning =

Concatenation of Differentiable Parameterized Layers (linear & nonlinearity functions)



Want to find optimal weights W to minimize some loss function E!

Backprop Whiteboard Demo



f_1	$z_1 = x_1 w_1 + x_2 w_3 + b_1$ $z_2 = x_1 w_2 + x_2 w_4 + b_2$
f_2	$h_1 = \frac{exp(z_1)}{1 + exp(z_1)}$ $h_2 = \frac{exp(z_2)}{1 + exp(z_2)}$
f_3	$\hat{y} = h_1 w_5 + h_2 w_6 + b_3$
f_4	$E = (y - \hat{y})^2$

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

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

Neurons

1-Layer Neural Network

Multi-layer Neural Network

Loss Functions

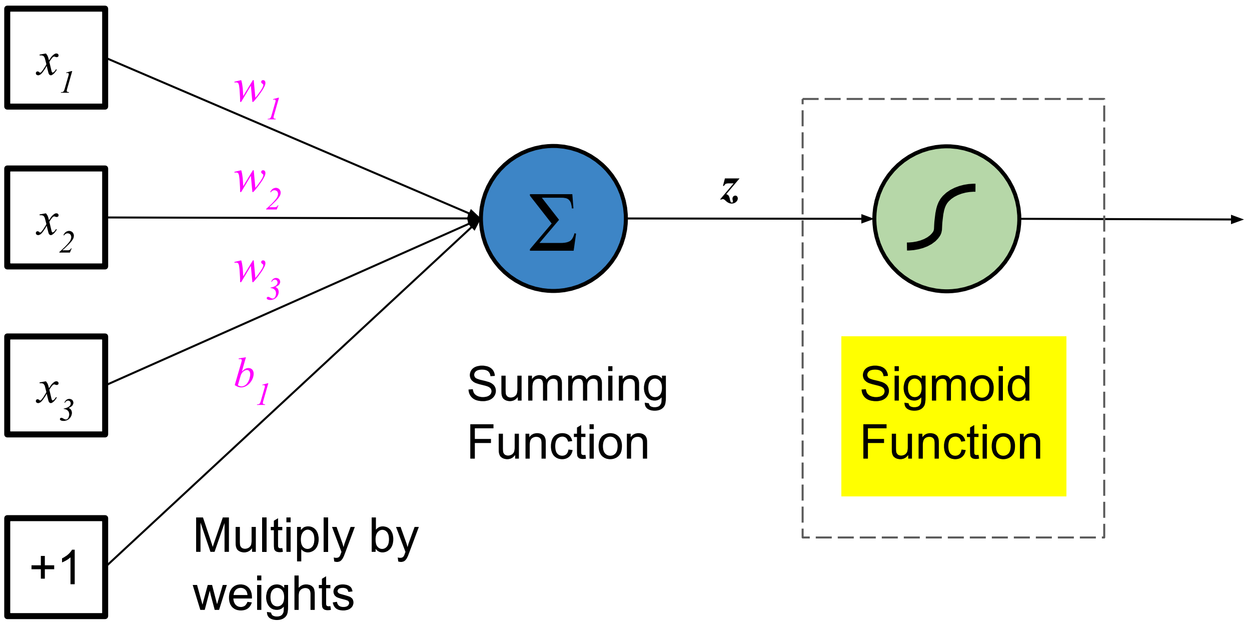
Backpropagation

Nonlinearity Functions

NNs in Practice

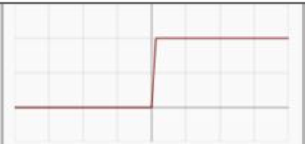
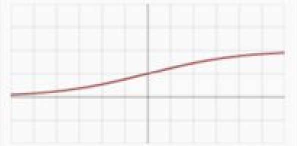
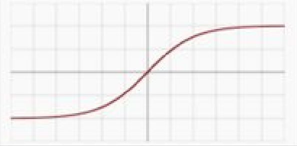
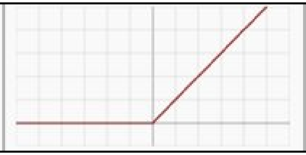
Nonlinearity Functions

(i.e. transfer or activation functions)



Nonlinearity Functions

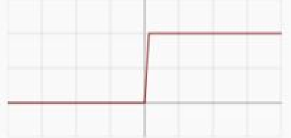
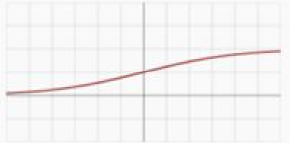
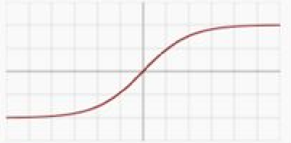
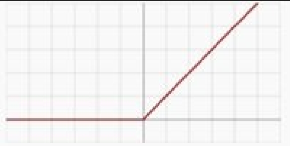
(i.e. transfer or activation functions)

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

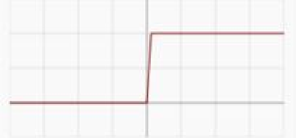
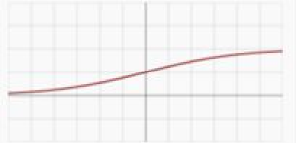
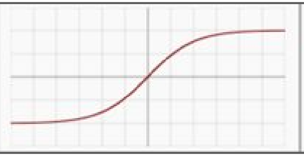
(i.e. transfer or activation functions)

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]		$f(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ x & \text{for} \quad x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ 1 & \text{for} \quad x \geq 0 \end{cases}$

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

Nonlinearity Functions

(aka 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

Neurons

1-Layer Neural Network

Multi-layer Neural Network

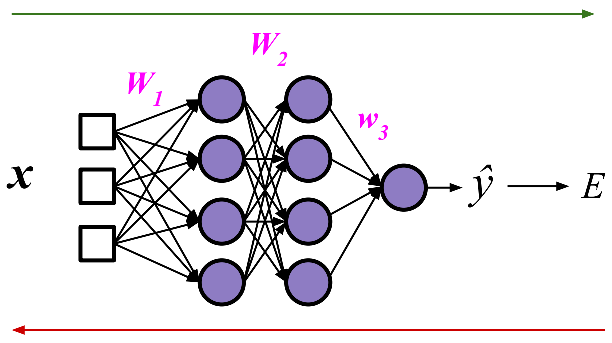
Loss Functions

Backpropagation

Nonlinearity Functions

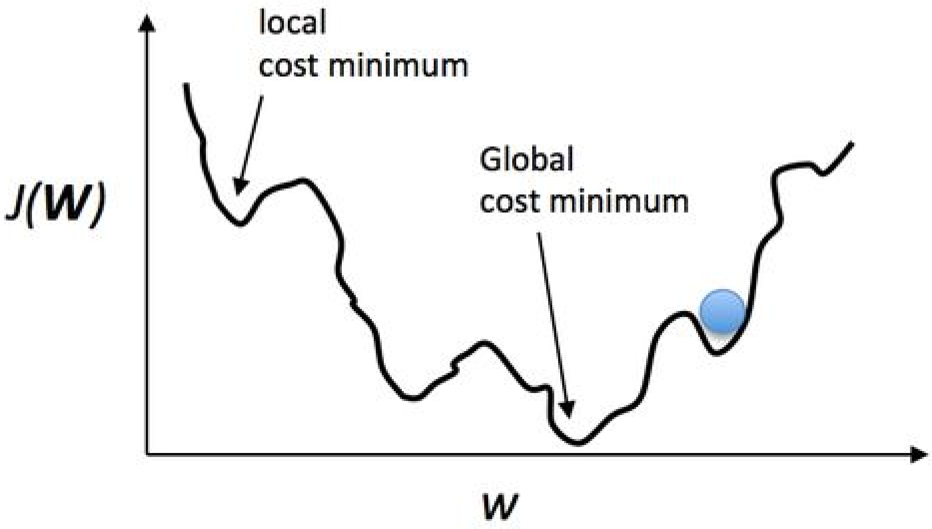
NNs in Practice

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

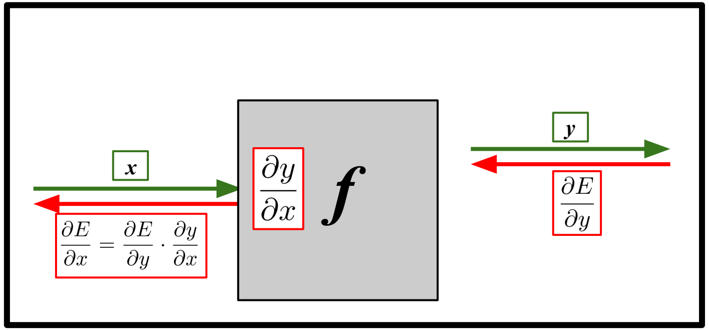
Non-Convexity of Neural Nets



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

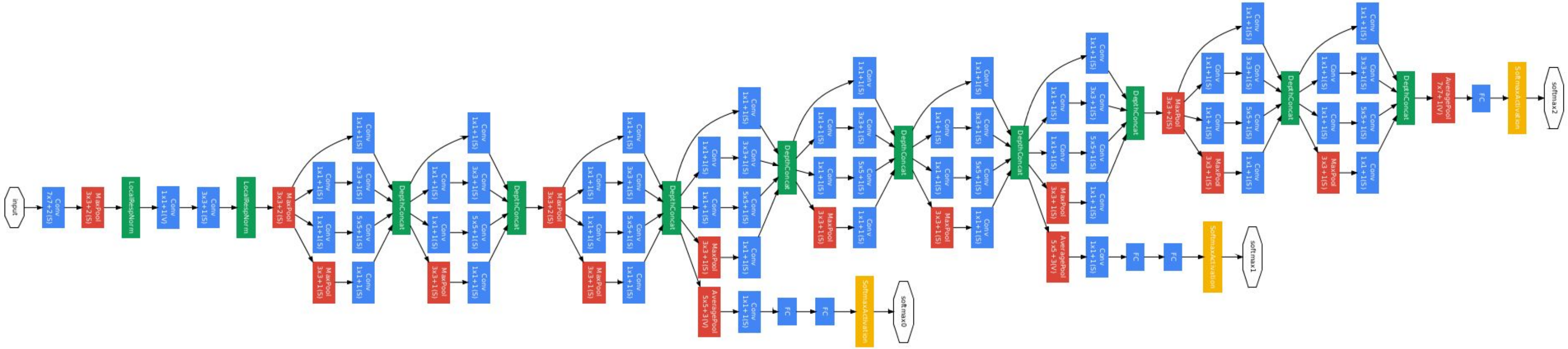
Pascanu, et. al. *On the saddle point problem for non-convex optimization 2014*

Building Deep Neural Nets



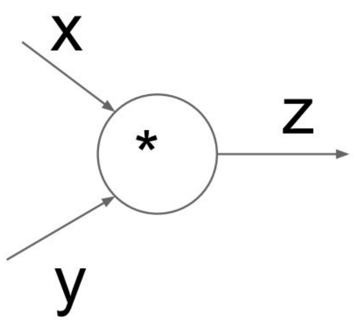
http://cs231n.stanford.edu/slides/winter1516_lecture5.pdf

Building Deep Neural Nets



“GoogLeNet” for Object Classification

Block Example Implementation

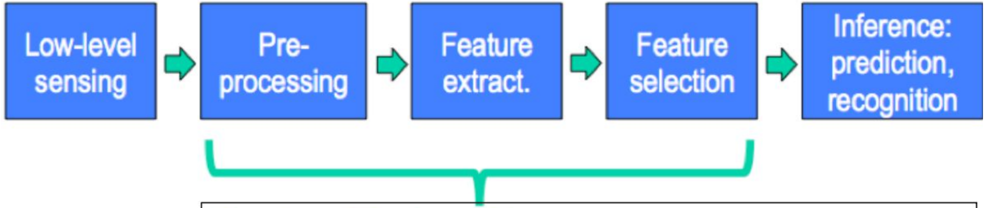



```
class MultiplyGate(object):
    def forward(x,y):
        z = x*y
        self.x = x # must keep these around!
        self.y = y
        return z
    def backward(dz):
        dx = self.y * dz # [dz/dx * dL/dz]
        dy = self.x * dz # [dz/dy * dL/dz]
        return [dx, dy]
```

(x,y,z are scalars)

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

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.