17.09.2024

Statistical Methods in AI (CS7.403)

Lecture-13: Neural Networks-1

Ravi Kiran (ravi.kiran@iiit.ac.in)

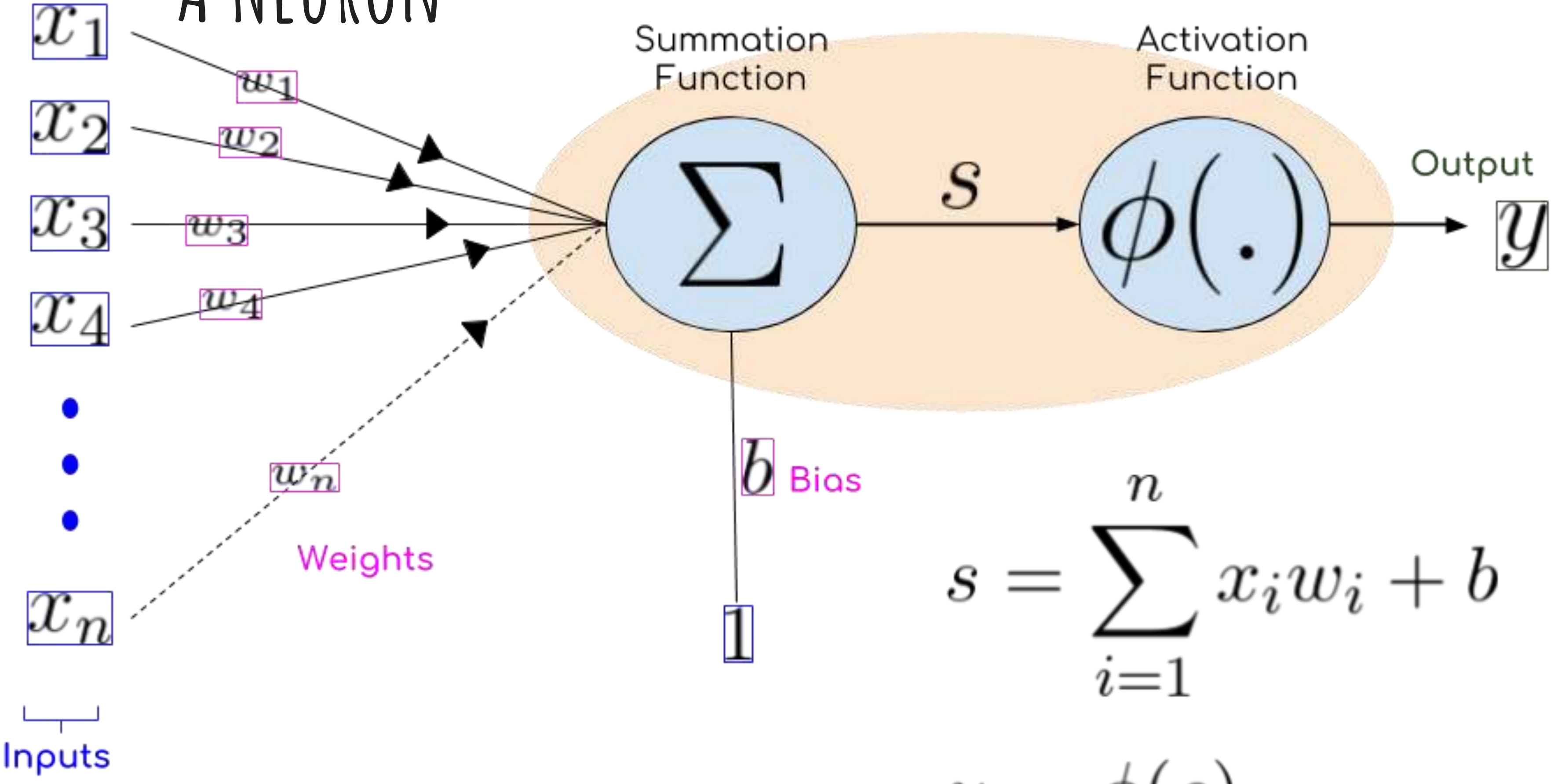
https://ravika.github.io

@vikataravi

Center for Visual Information Technology (CVIT)

IIIT Hyderabad

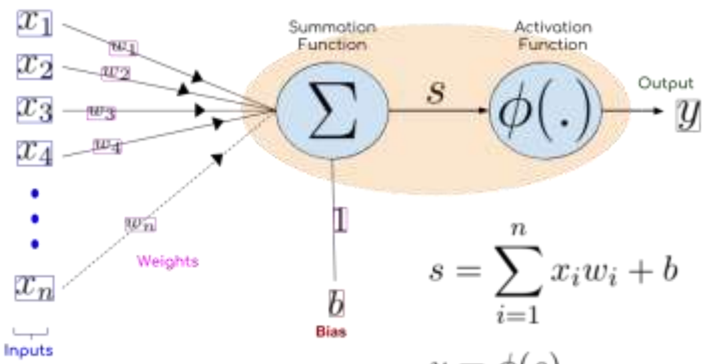
A Neuron



$$s = \sum_{i=1}^{n} x_i w_i + b$$

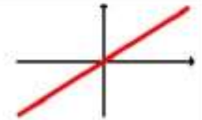
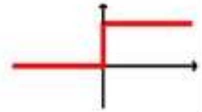
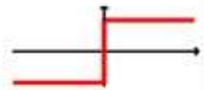
$$y = \phi(s)$$

ACTIVATION FUNCTIONS

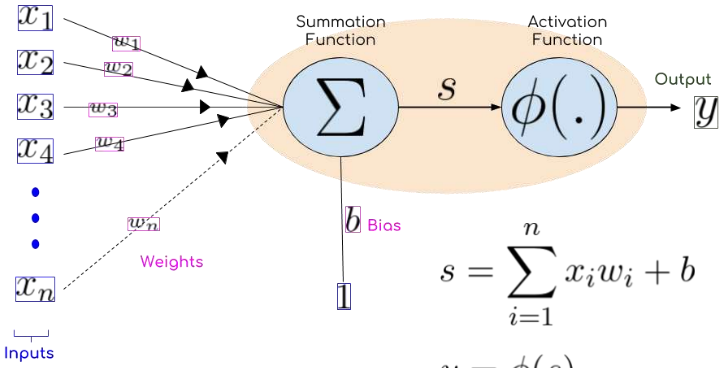


$$s = \sum_{i=1}^{n} x_i w_i + b$$

$$y = \phi(s)$$

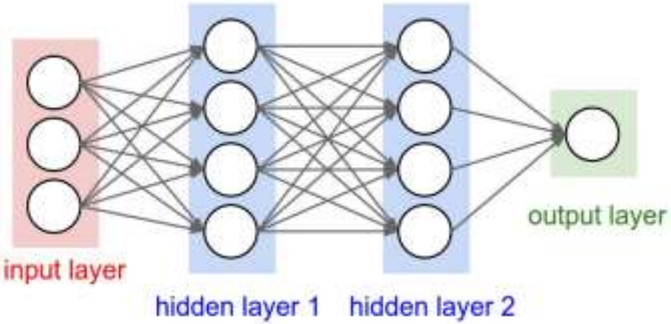
Activation Function	Equation	Example	1D Graph
Linear	$\phi(z) = z$	Adaline, linear regression	
Unit Step (Heaviside Function)	$\phi(z) = \begin{cases} 0 & z < 0 \\ 0.5 & z = 0 \\ 1 & z > 0 \end{cases}$	Perceptron variant	
Sign (signum)	$\phi(z) = \begin{cases} -1 & z < 0 \\ 0 & z = 0 \\ 1 & z > 0 \end{cases}$	Perceptron variant	
Piece-wise Linear	$\phi(z) = \begin{cases} 0 & z \le -\frac{1}{2} \\ z + \frac{1}{2} & -\frac{1}{2} \le z \le \frac{1}{2} \\ 1 & z \ge \frac{1}{2} \end{cases}$	Support vector machine	
Logistic (sigmoid)	$\phi(z) = \frac{1}{1 + e^{-z}}$	Logistic regression,	
Hyperbolic Tangent (tanh)	$\phi(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$		
ReLU	$\phi(z) = \begin{cases} 0 & z < 0 \\ z & z > 0 \end{cases}$		

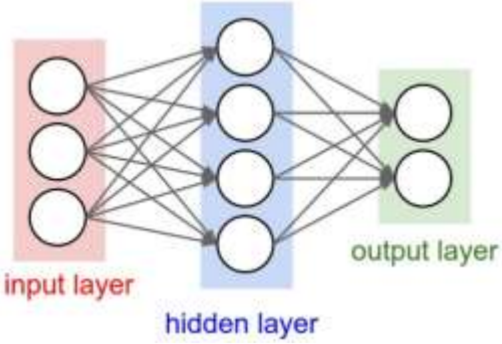
Why Use Only One Neuron ?



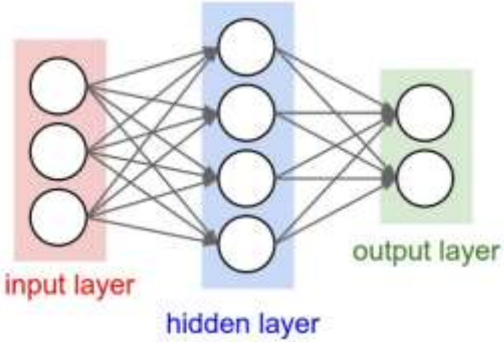
$$s = \sum_{i=1}^{n} x_i w_i + b$$

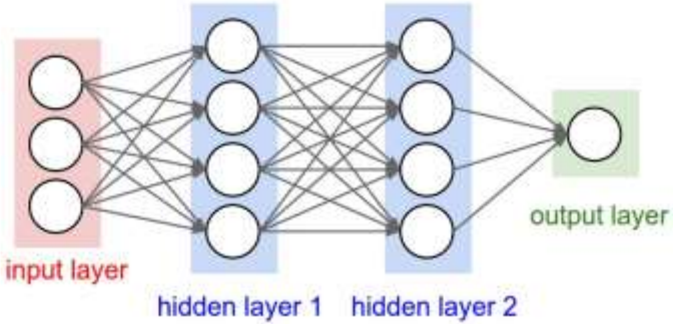
$$y = \phi(s)$$

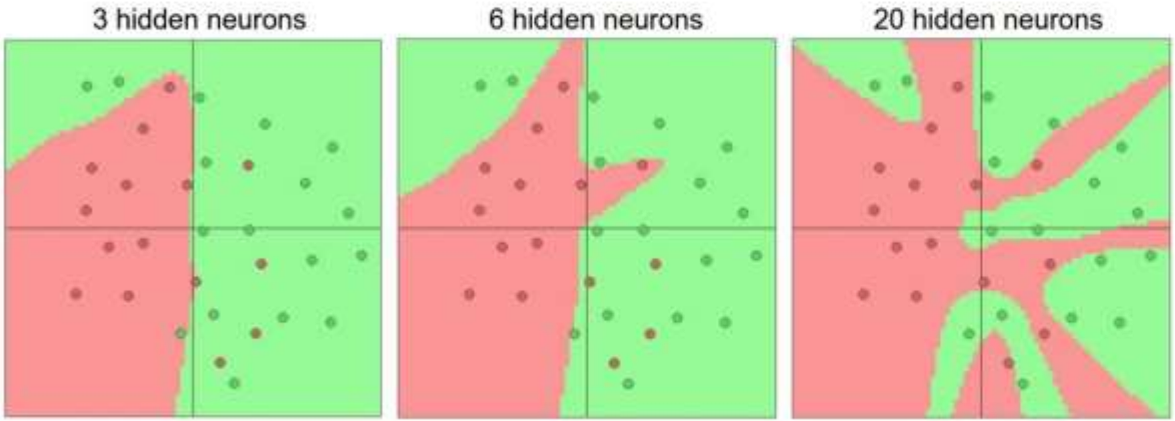




Why Use Only One Neuron ?



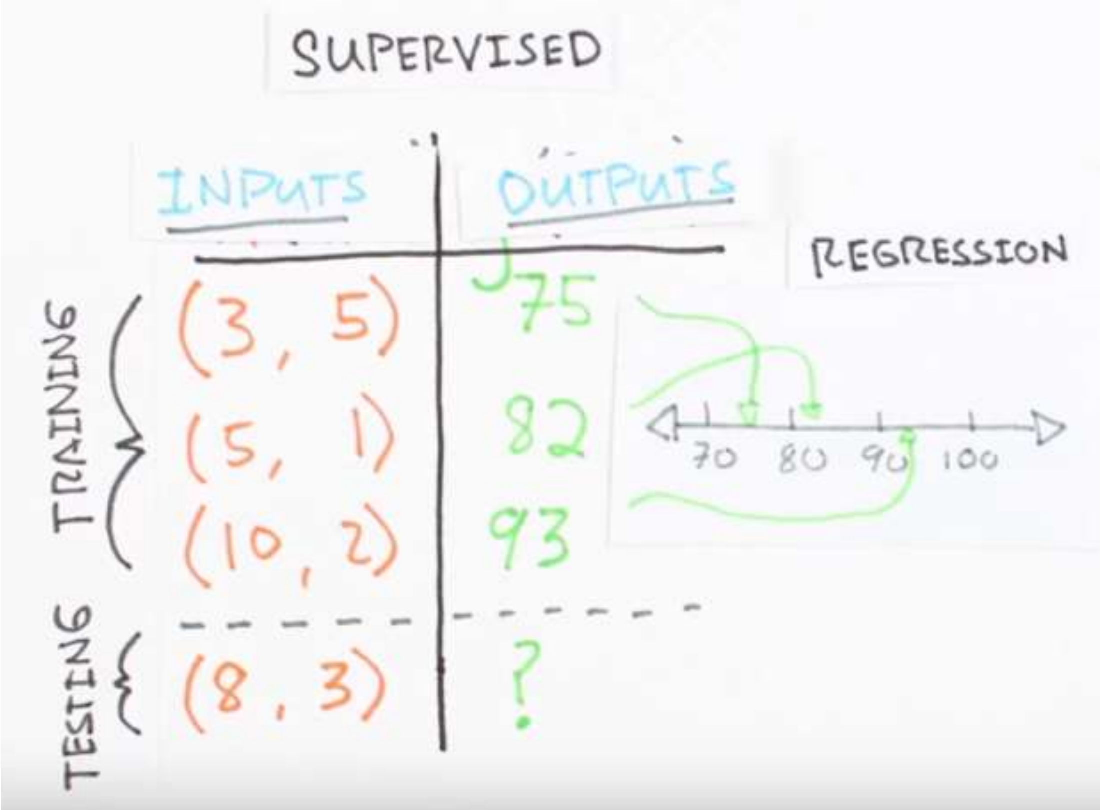




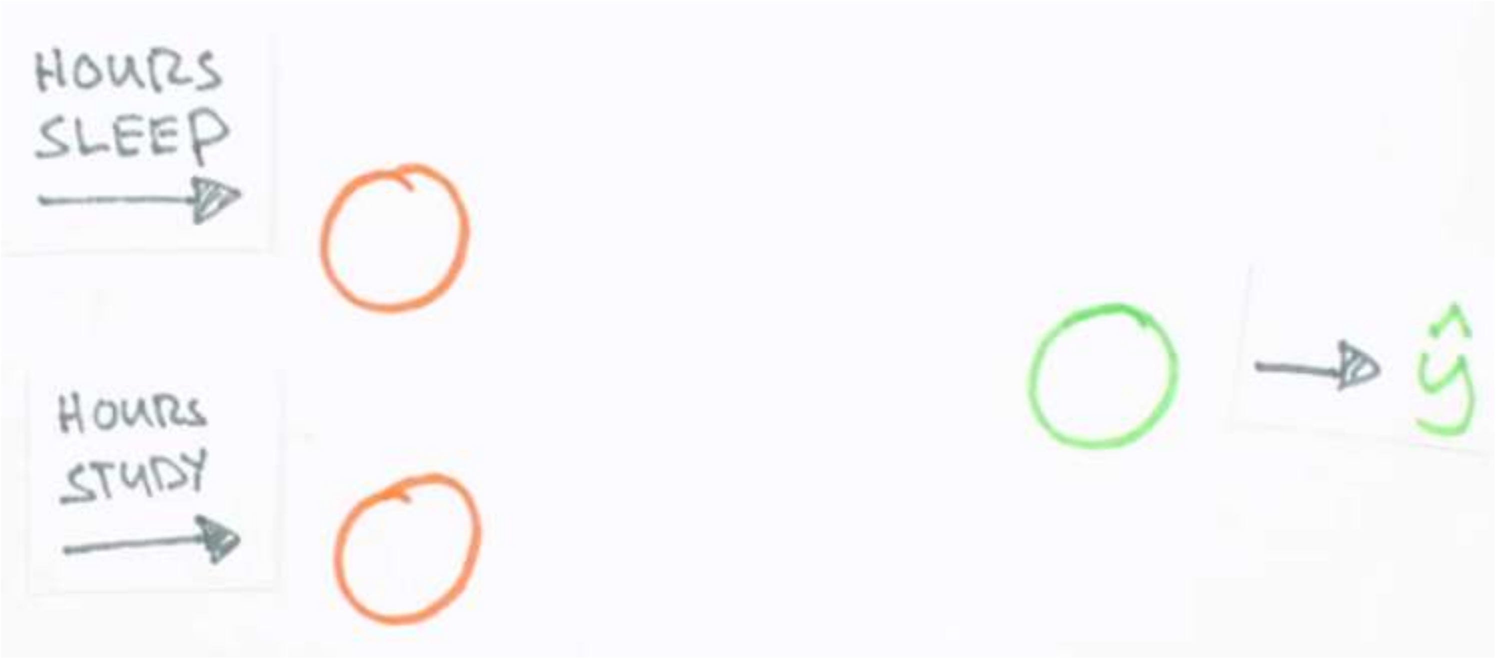
Multi-Neuron Networks

	X (Hours sleep, hours study)	y (Score on test)
TRAINING	(3, 5)	75
	(5, 1)	82
	(10, 2)	93
TESTING	(8, 3)	?

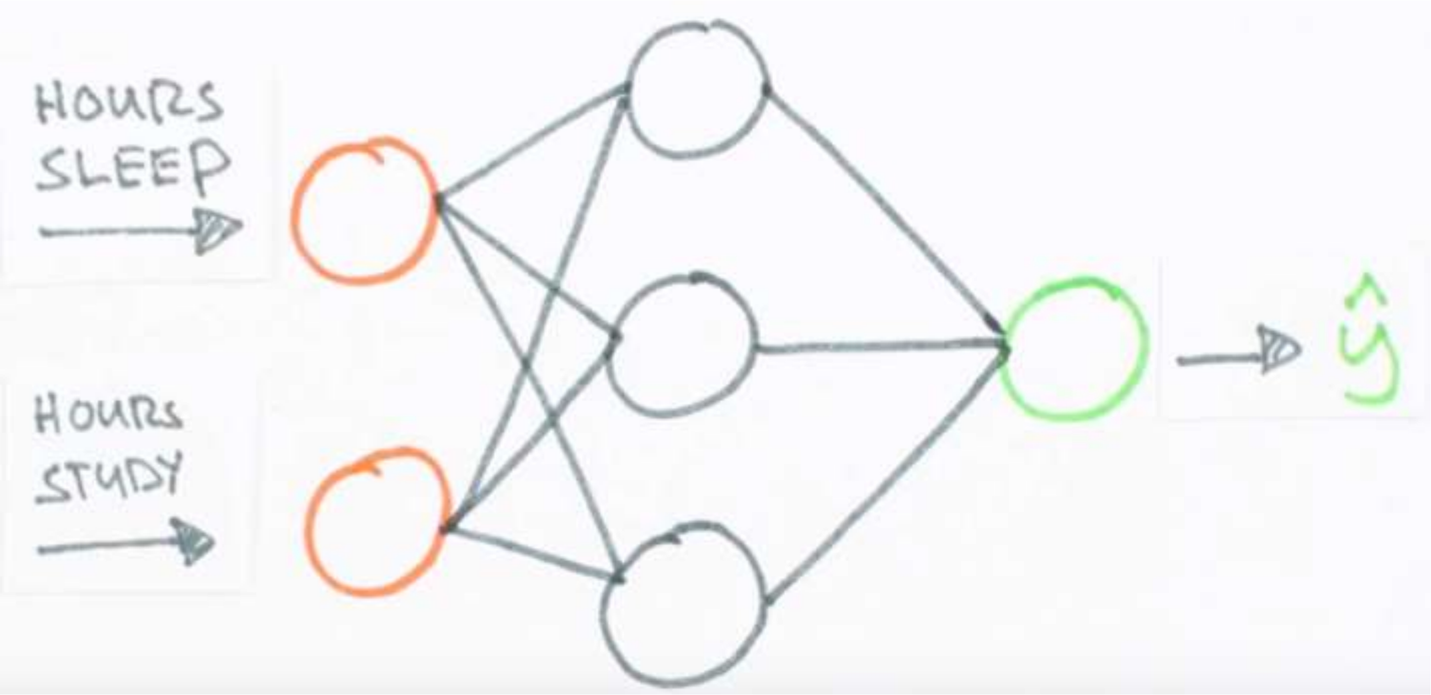
Multi-Neuron Networks



Multi-Neuron Networks :: Architecture

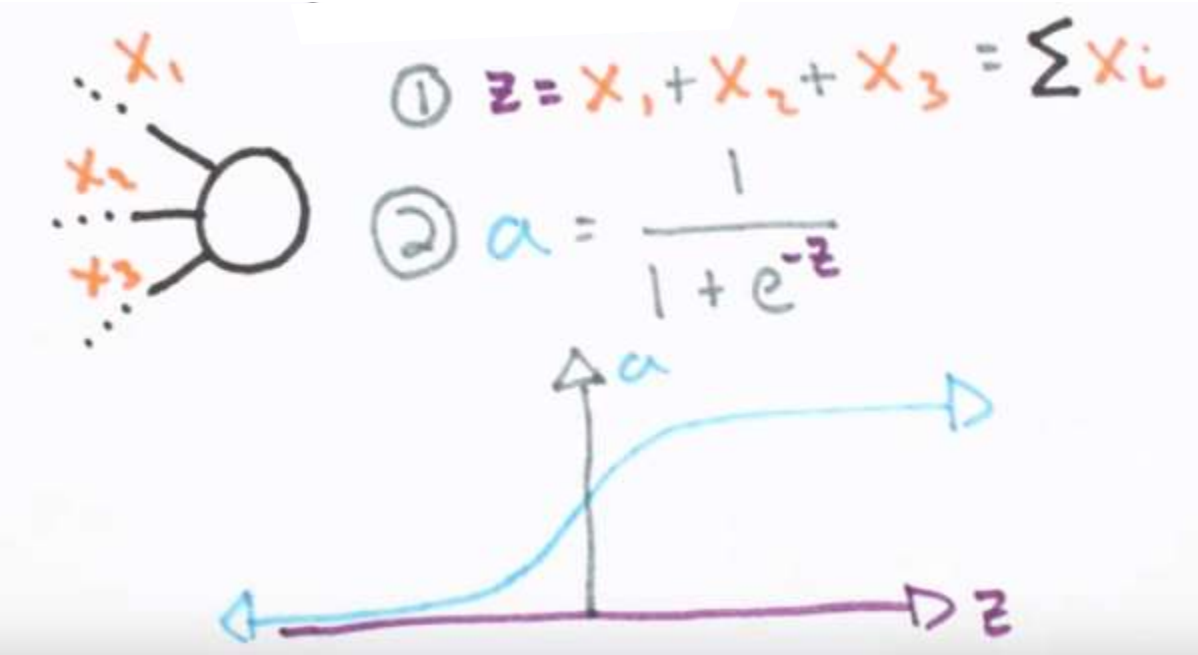


Multi-Neuron Networks :: Architecture



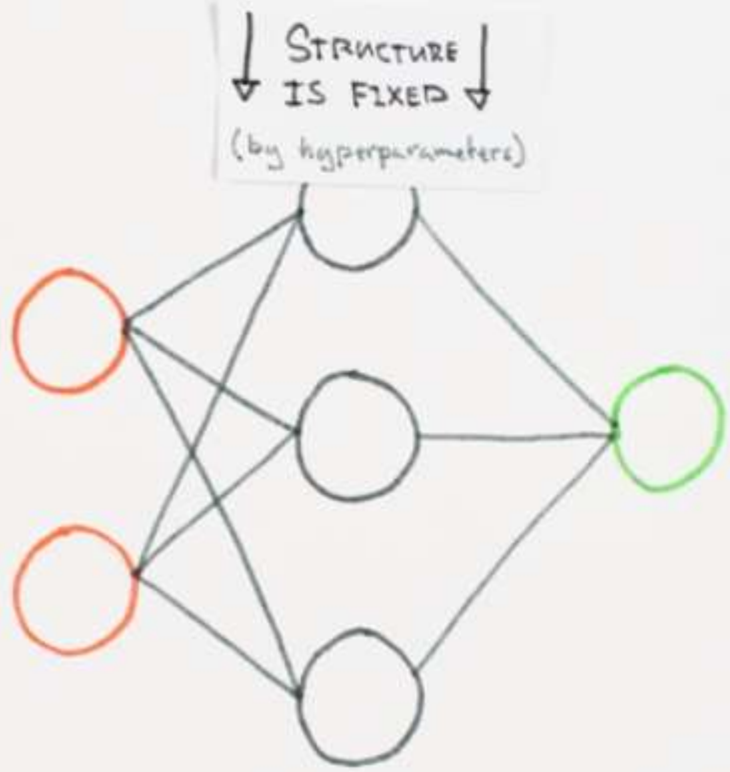
Multi-Neuron Networks :: Architecture

Neuron



Multi-Neuron Networks :: Architecture

```python
class Neural_Network(object):
    def __init__(self):
        #Define Hyperparameters
        self.inputLayerSize = 2
        self.outputLayerSize = 1
        self.hiddenLayerSize = 3

        #Weights (parameters)
        self.W1 = np.random.randn(self.inputLayerSize,self.hiddenLayerSize)
        self.W2 = np.random.randn(self.hiddenLayerSize,self.outputLayerSize)
```



Multi-Neuron Networks :: training

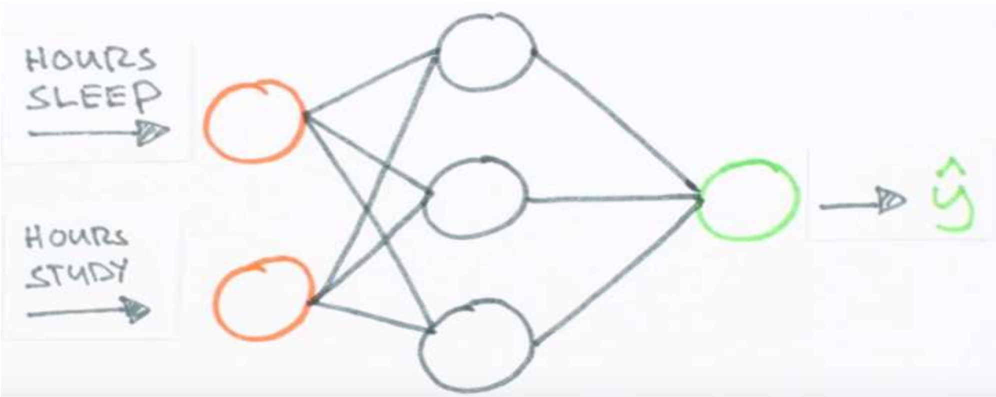
Initialize network with random weights

While [not converged]

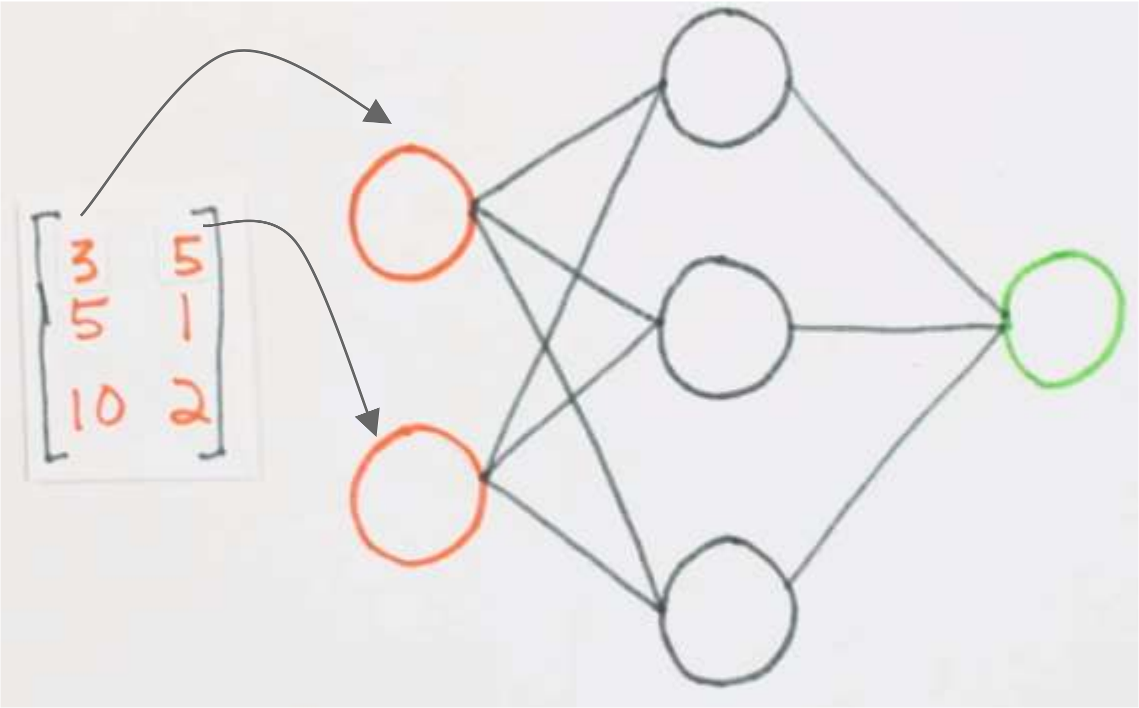
- Do forward prop
- Do backprop and determine change in weights
- Update ALL weights in ALL layers

Multi-Neuron Networks :: forward propagation

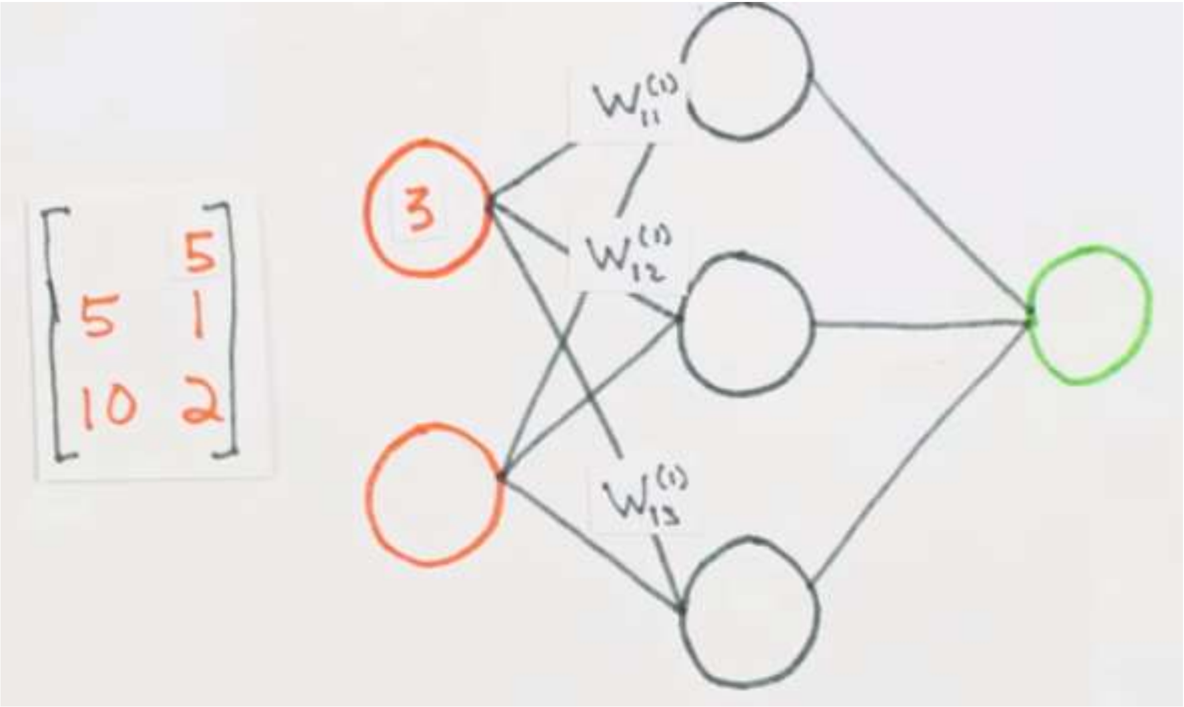
```python
class Neural_Network(object):
    def __init__(self):
        #Define Hyperparameters
        self.inputLayerSize = 2
        self.outputLayerSize = 1
        self.hiddenLayerSize = 3

        #Weights (parameters)
        self.W1 = np.random.randn(self.inputLayerSize,self.hiddenLayerSize)
        self.W2 = np.random.randn(self.hiddenLayerSize,self.outputLayerSize)

    def forward(self, X):
        #Propogate inputs though network
```



Multi-Neuron Networks :: FORWARD PROPAGATION

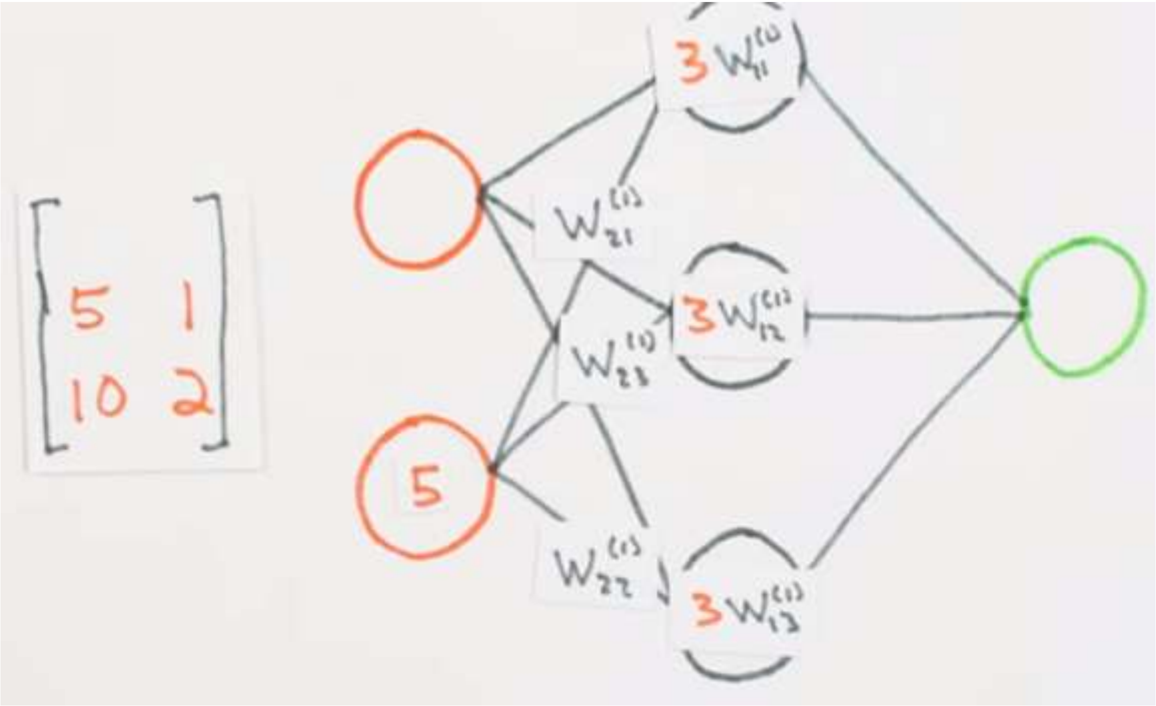


Multi-Neuron Networks :: FORWARD PROPAGATION

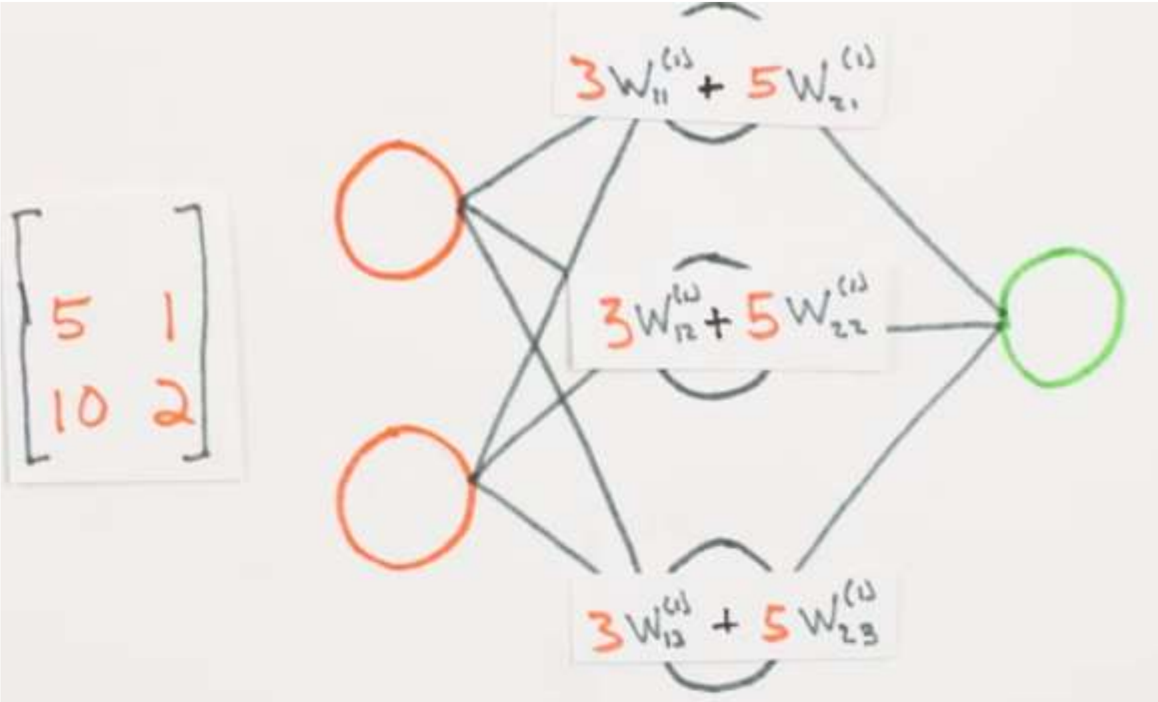


Note: No biases, for simplicity

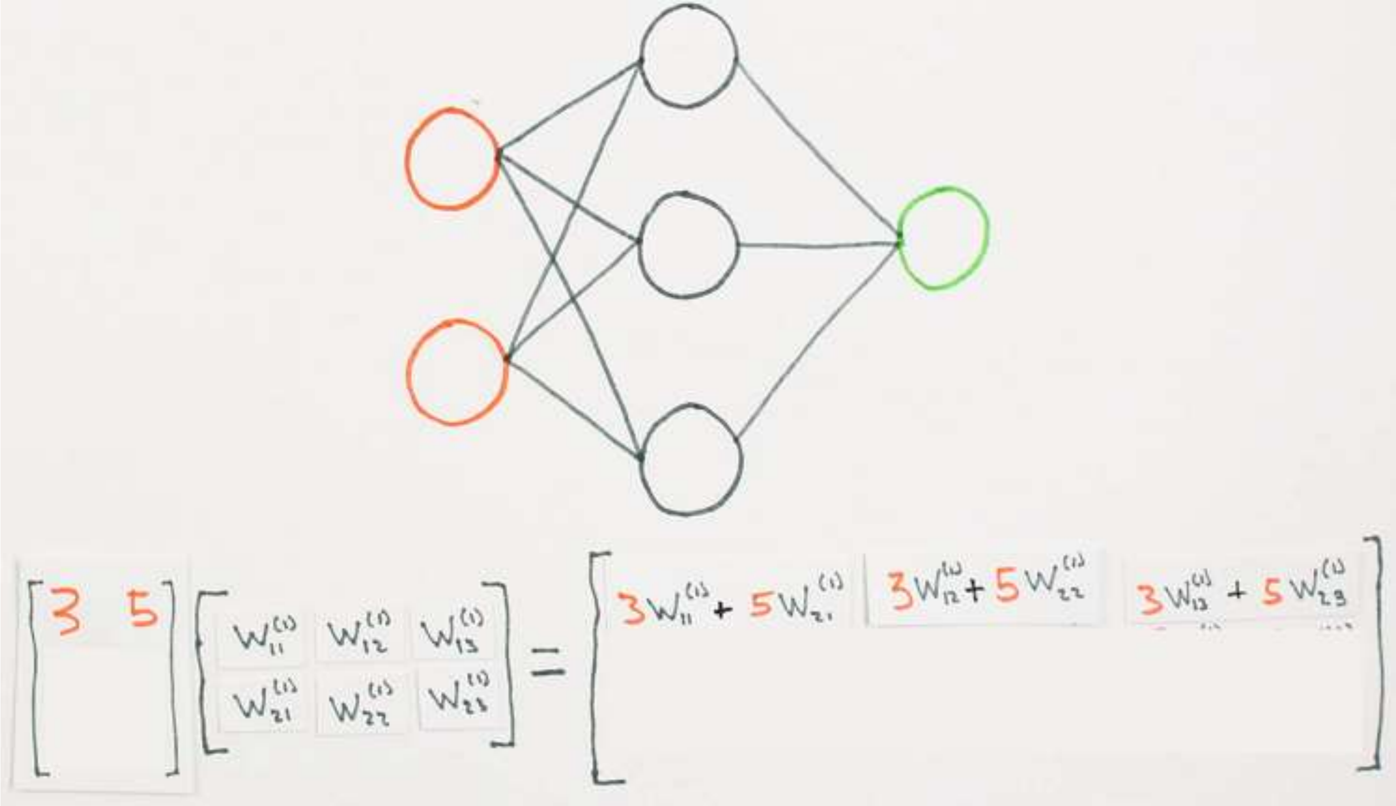
Multi-Neuron Networks :: FORWARD PROPAGATION



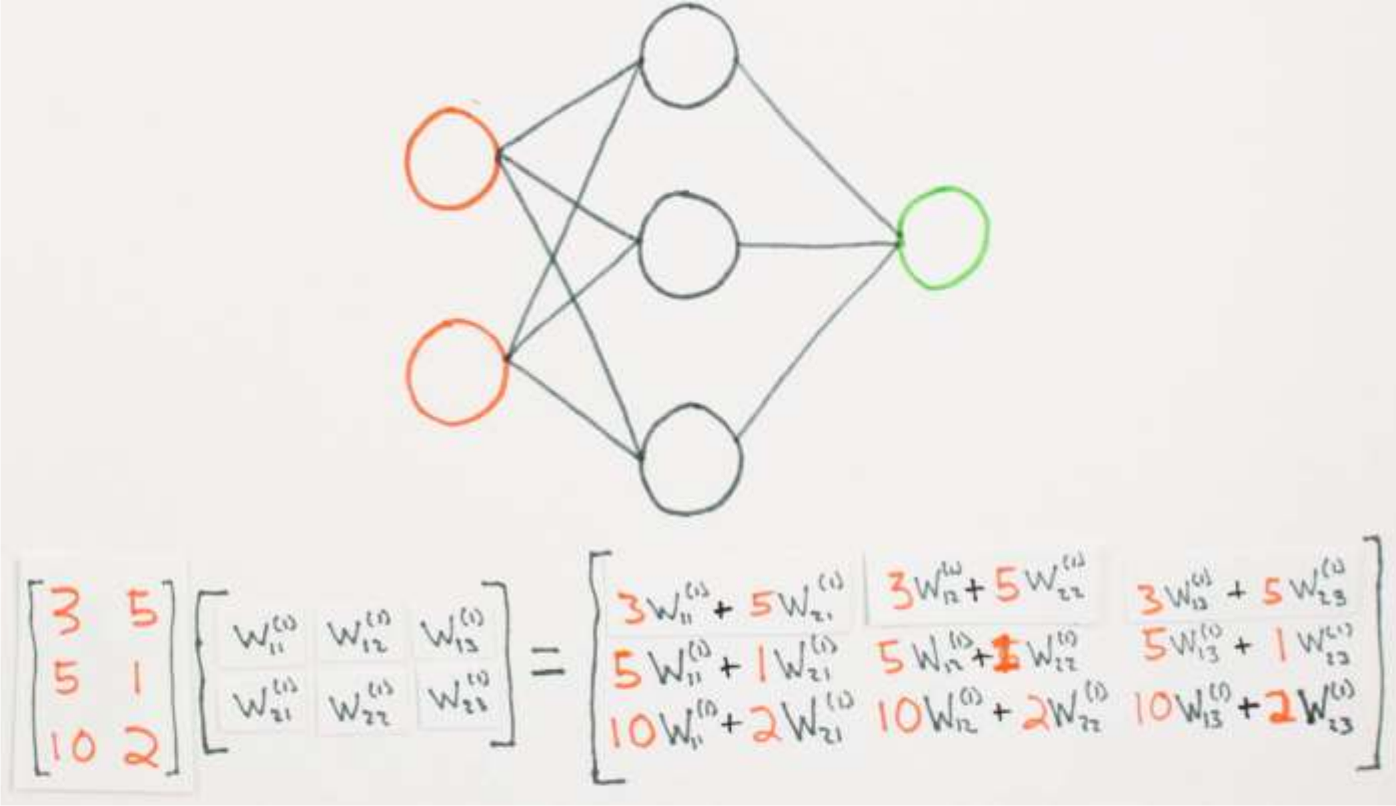
Multi-Neuron Networks :: FORWARD PROPAGATION



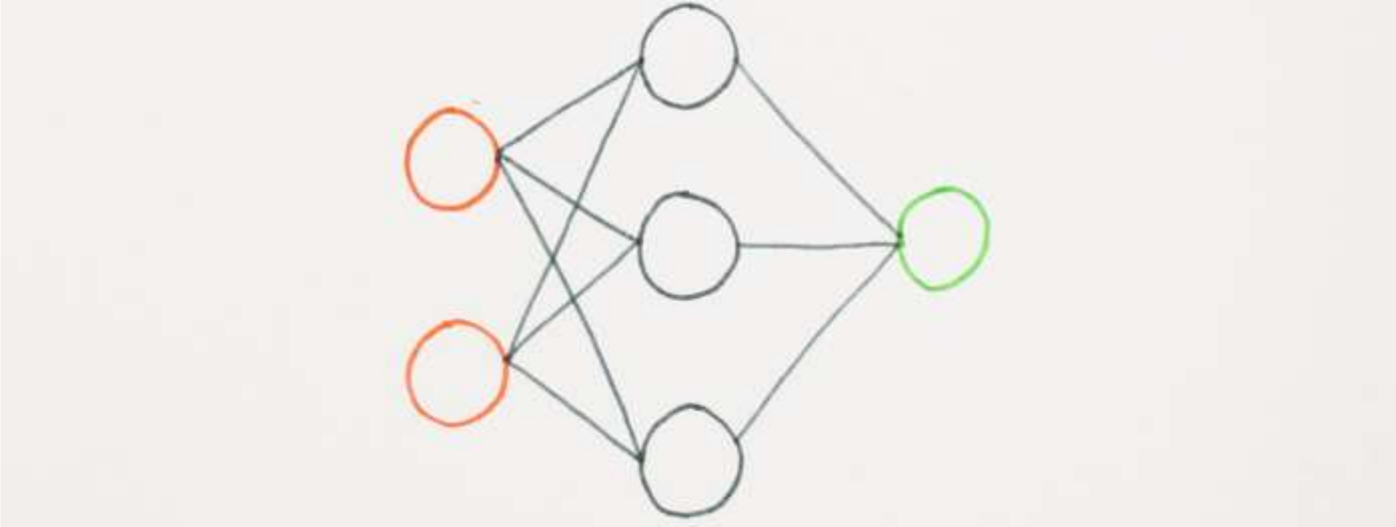
Multi-Neuron Networks :: FORWARD PROPAGATION



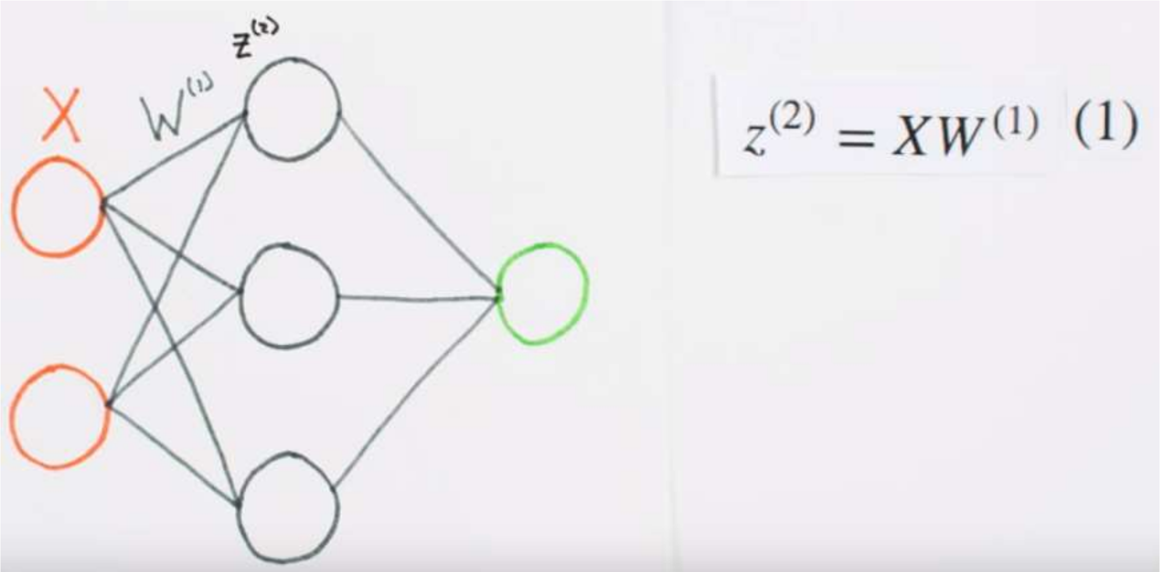
Multi-Neuron Networks :: forward propagation



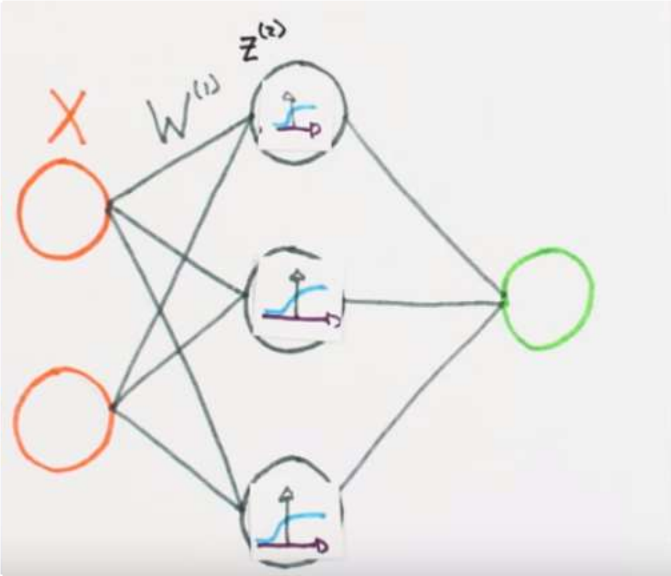
Multi-Neuron Networks :: FORWARD PROPAGATION



Multi-Neuron Networks :: FORWARD PROPAGATION

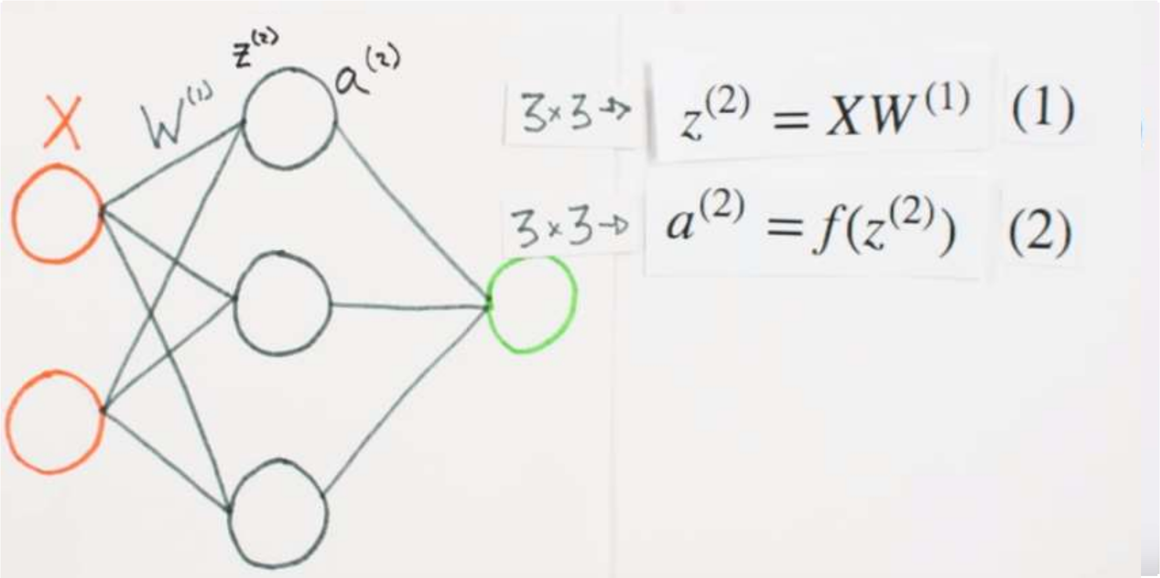


Multi-Neuron Networks :: Forward Propagation

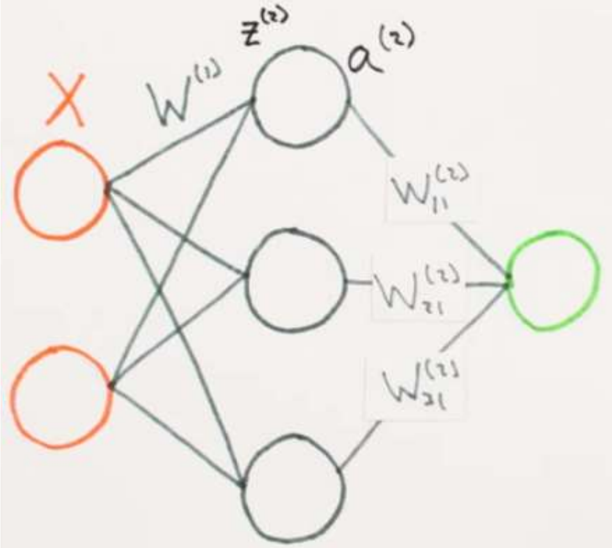


$$z^{(2)} = XW^{(1)} \quad (1)$$

Multi-Neuron Networks :: FORWARD PROPAGATION



Multi-Neuron Networks :: FORWARD PROPAGATION

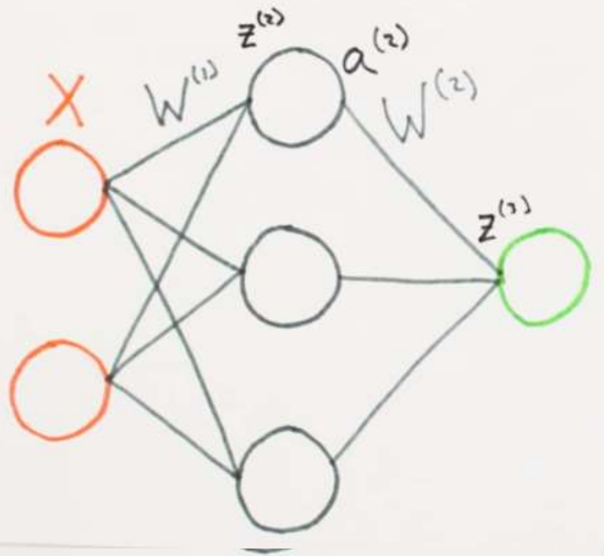


$$z^{(2)} = XW^{(1)} \quad (1)$$

$$a^{(2)} = f(z^{(2)}) \quad (2)$$

$$z^{(3)} = a^{(2)}W^{(2)} \quad (3)$$

Multi-Neuron Networks :: FORWARD PROPAGATION



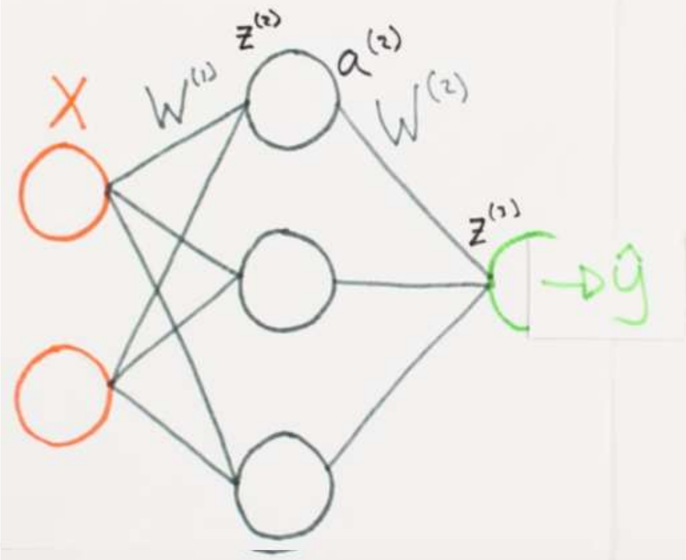
$$z^{(2)} = XW^{(1)} \quad (1)$$

$$a^{(2)} = f(z^{(2)}) \quad (2)$$

(3x1) (3x3) (3x1)

$$z^{(3)} = a^{(2)}W^{(2)} \quad (3)$$

Multi-Neuron Networks :: forward propagation



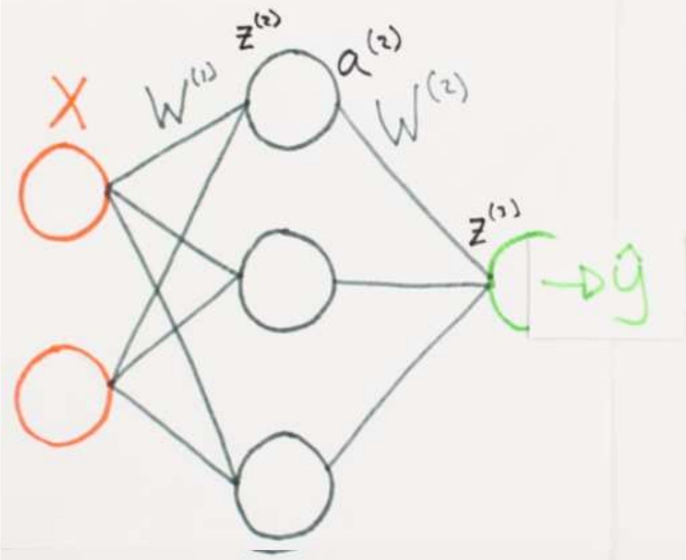
$$z^{(2)} = XW^{(1)} \quad (1)$$

$$a^{(2)} = f(z^{(2)}) \quad (2)$$

$$z^{(3)} = a^{(2)}W^{(2)} \quad (3)$$

$$\hat{y} = f(z^{(3)}) \quad (4)$$

Multi-Neuron Networks :: forward propagation



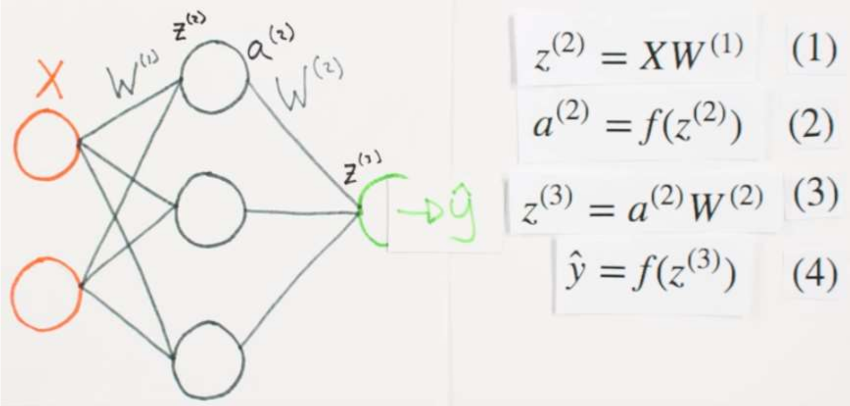
$$z^{(2)} = XW^{(1)} \qquad (1)$$

$$a^{(2)} = f(z^{(2)}) \qquad (2)$$

$$z^{(3)} = a^{(2)}W^{(2)} \qquad (3)$$

$$\hat{y} = f(z^{(3)}) \qquad (4)$$

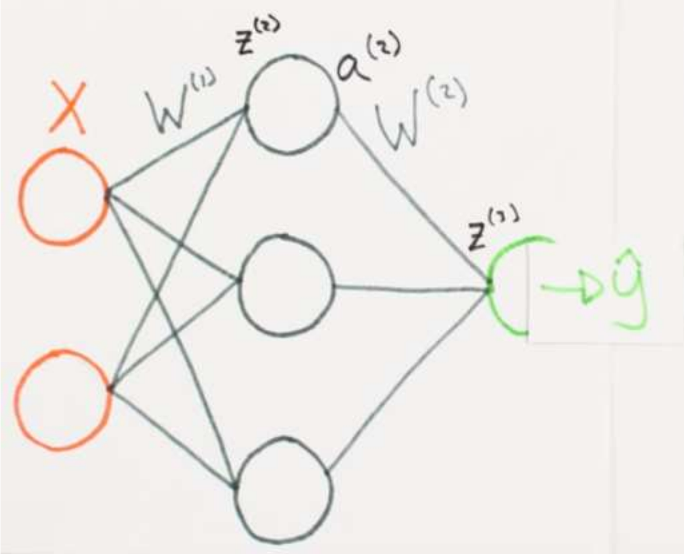
Multi-Neuron Networks :: Forward Propagation




```python
def forward(self, X):
    #Propogate inputs though network
    self.z2 = np.dot(X, self.W1) # z2 = X * W1
    self.a2 = self.sigmoid(self.z2) # a2 = sigmoid(z2)
    self.z3 = np.dot(self.a2, self.W2) # z3 = a2 * W2
    yHat = self.sigmoid(self.z3) # yHat = sigmoid(z3)
    return yHat

def sigmoid(self, z):
    #Apply sigmoid activation function to scalar, vector, or matrix
    return 1/(1+np.exp(-z))
```

Multi-Neuron Networks :: forward propagation

```
In [7]: NN = Neural_Network()

In [8]: yHat = NN.forward(X)

In [9]: yHat
Out[9]: array([[ 0.59470263],
               [ 0.58177822],
               [ 0.50641742]])

n [10]: y
ut[10]: array([[ 0.75],
               [ 0.82],
               [ 0.93]])
```



$$z^{(2)} = XW^{(1)} \quad (1)$$

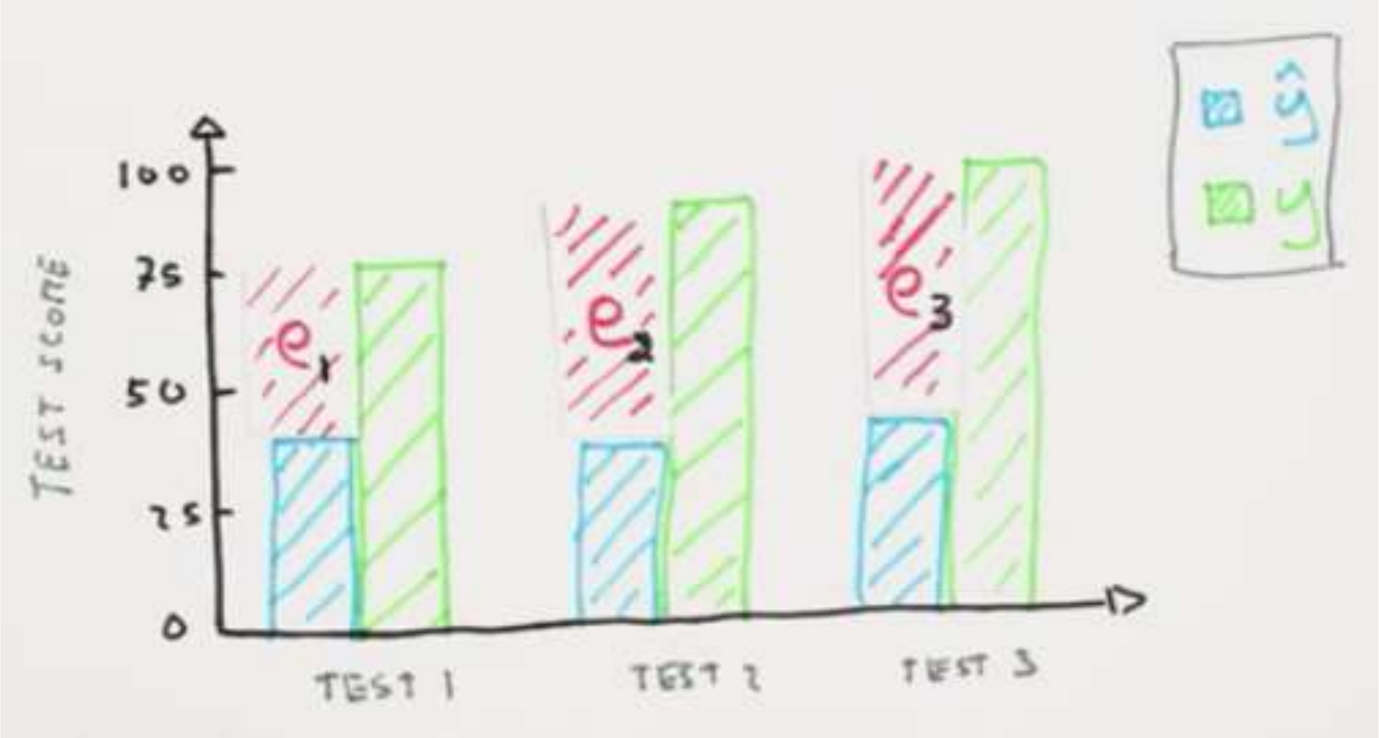
$$a^{(2)} = f(z^{(2)}) \quad (2)$$

$$z^{(3)} = a^{(2)}W^{(2)} \quad (3)$$

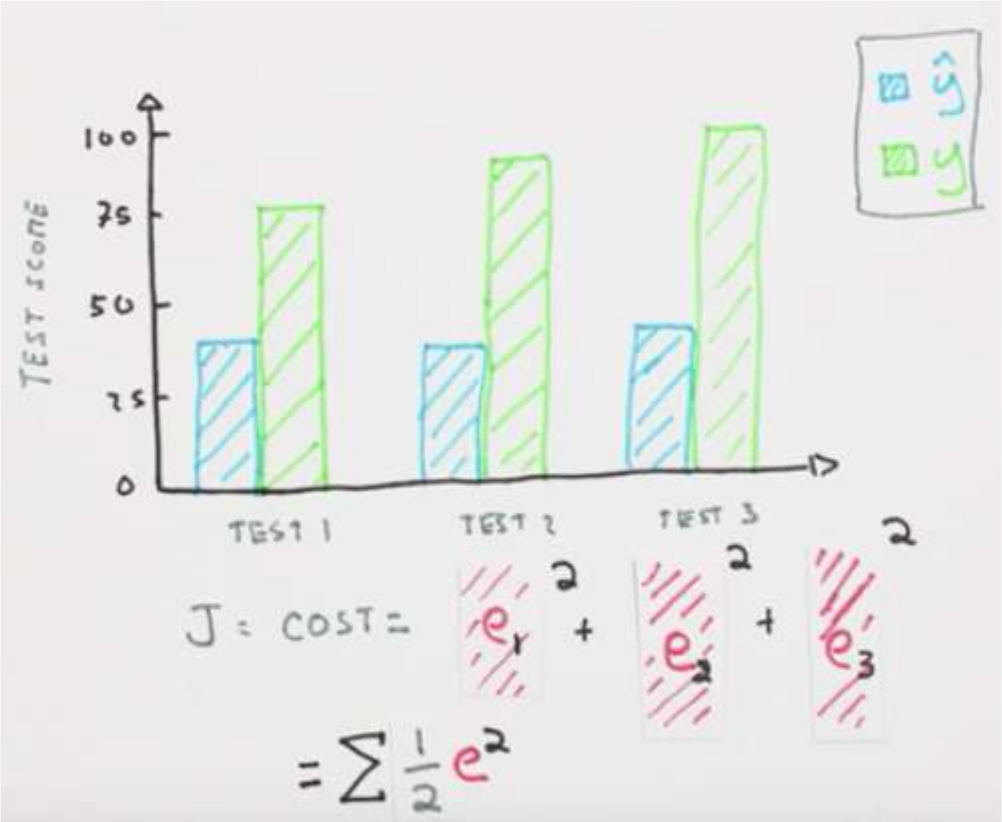
$$\hat{y} = f(z^{(3)}) \quad (4)$$

Multi-Neuron Networks :: Gradient Descent

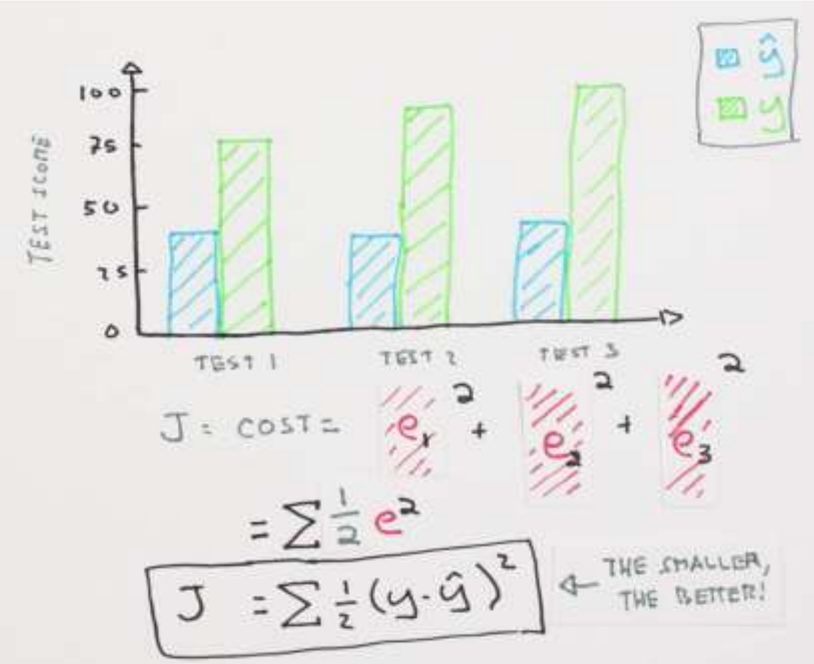
Multi-Neuron Networks :: Gradient Descent



Multi-Neuron Networks :: Gradient Descent

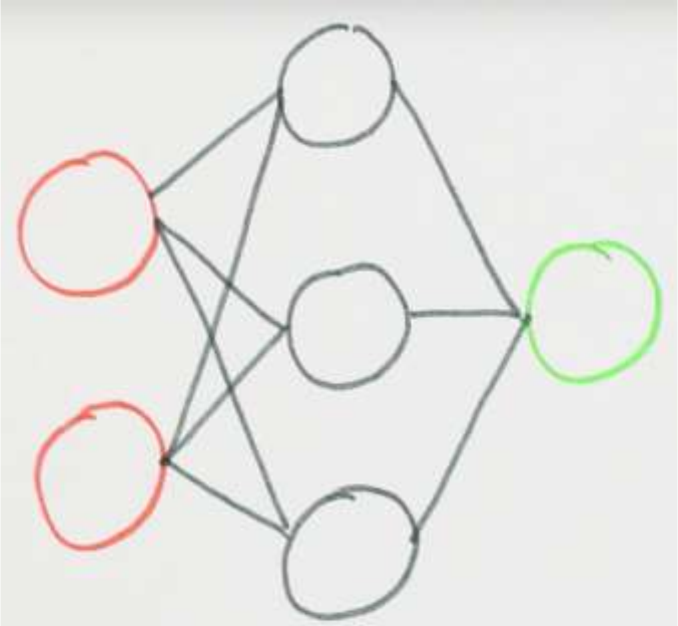


Multi-Neuron Networks :: Gradient Descent




```python
def costFunction(self, X, y):
    #Compute cost for given X,y, use weights already stored in class.
    self.yHat = self.forward(X)
    J = 0.5*sum((y-self.yHat)**2)
    return J
```

Multi-Neuron Networks :: Gradient Descent

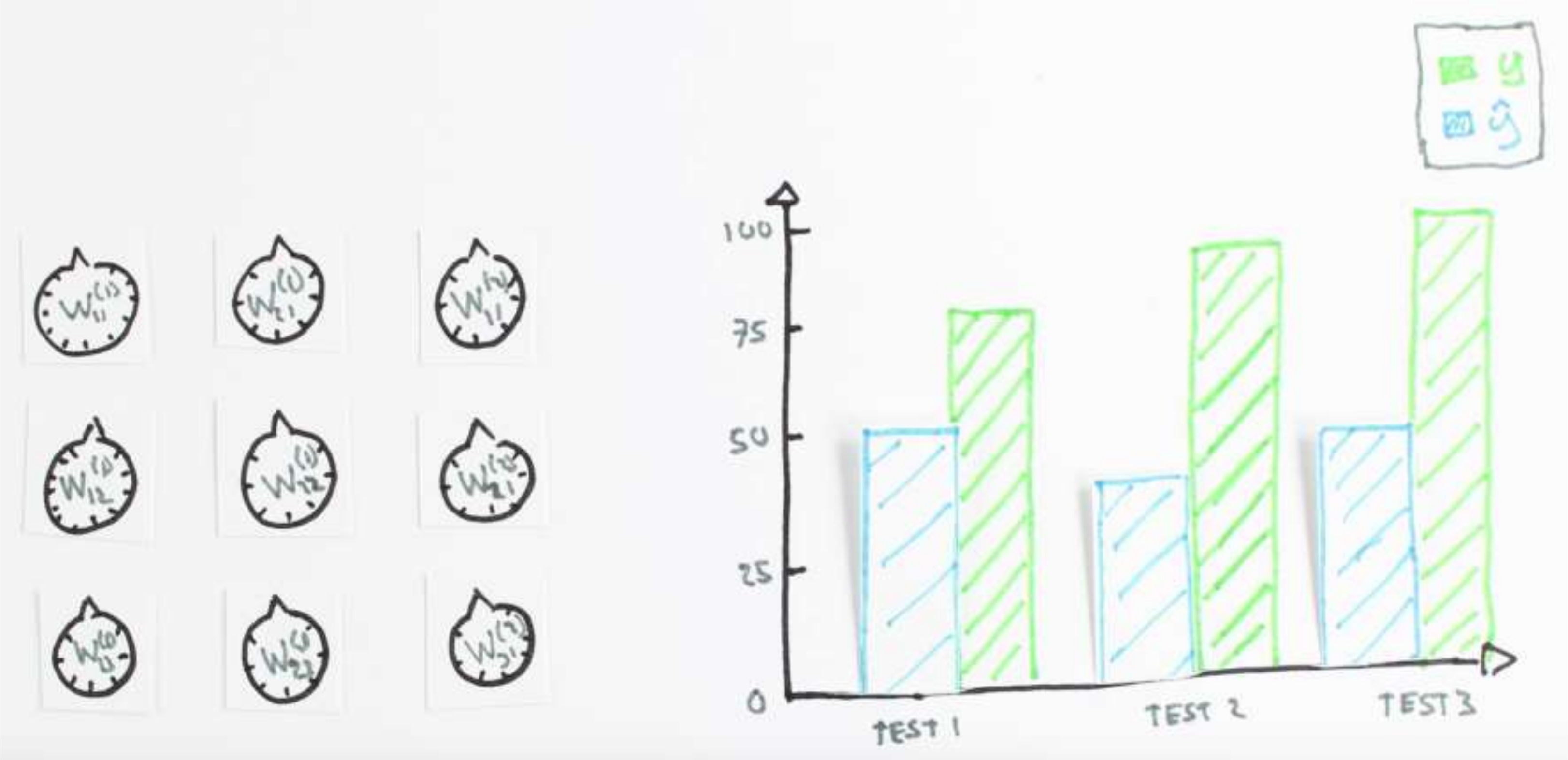


Training a Network

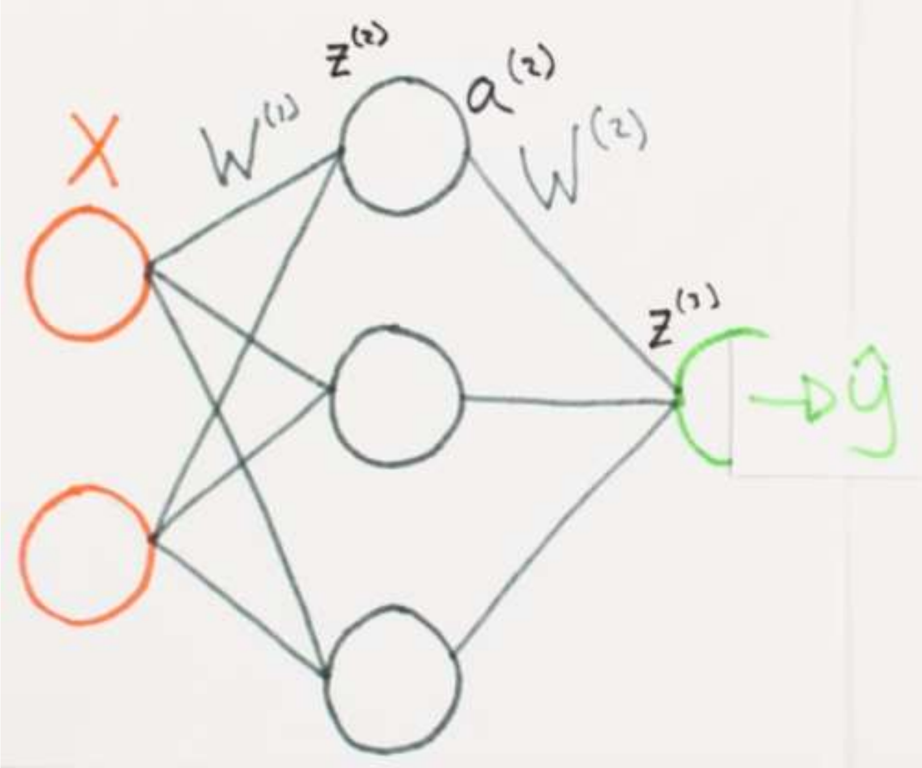
=

Minimizing a Cost Function

Multi-Neuron Networks :: Gradient Descent



Multi-Neuron Networks :: Gradient Descent



$$z^{(2)} = XW^{(1)} \quad (1)$$

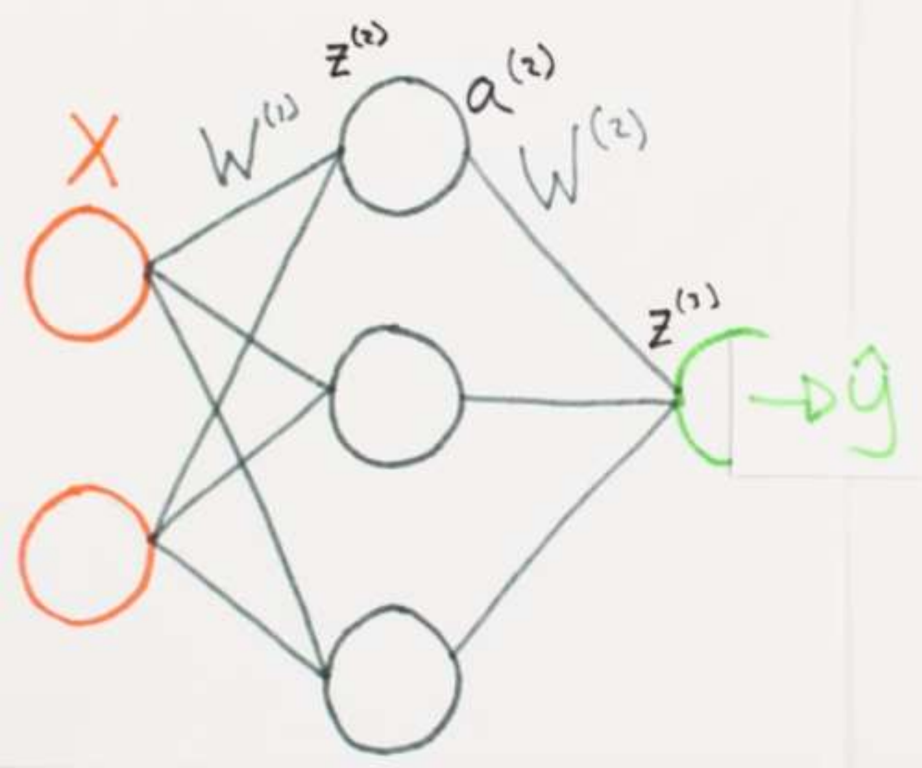
$$a^{(2)} = f(z^{(2)}) \quad (2)$$

$$z^{(3)} = a^{(2)}W^{(2)} \quad (3)$$

$$\hat{y} = f(z^{(3)}) \quad (4)$$

$$J = \sum \frac{1}{2}(y - \hat{y})^2 \quad (5)$$

Multi-Neuron Networks :: Gradient Descent



$$z^{(2)} = XW^{(1)} \quad (1)$$

$$a^{(2)} = f(z^{(2)}) \quad (2)$$

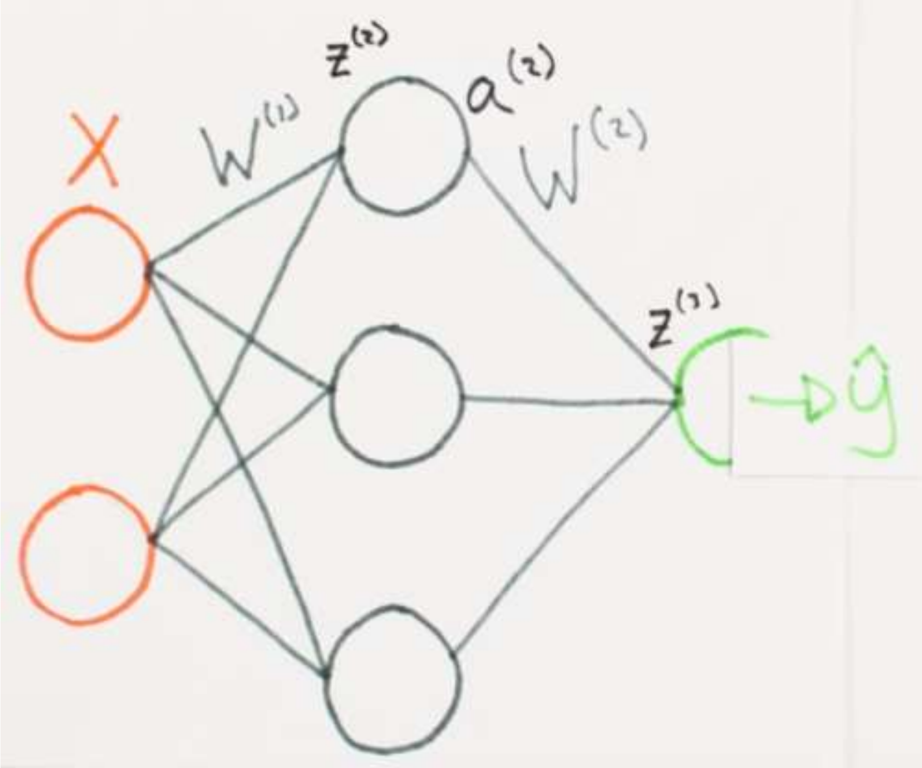
$$z^{(3)} = a^{(2)}W^{(2)} \quad (3)$$

$$\hat{y} = f(z^{(3)}) \quad (4)$$

$$J = \sum \frac{1}{2}(y - \hat{y})^2 \quad (5)$$

$$\boxed{J = \sum \frac{1}{2}(y - f(f(XW^{(1)})W^{(2)}))^2}$$

Multi-Neuron Networks :: Gradient Descent

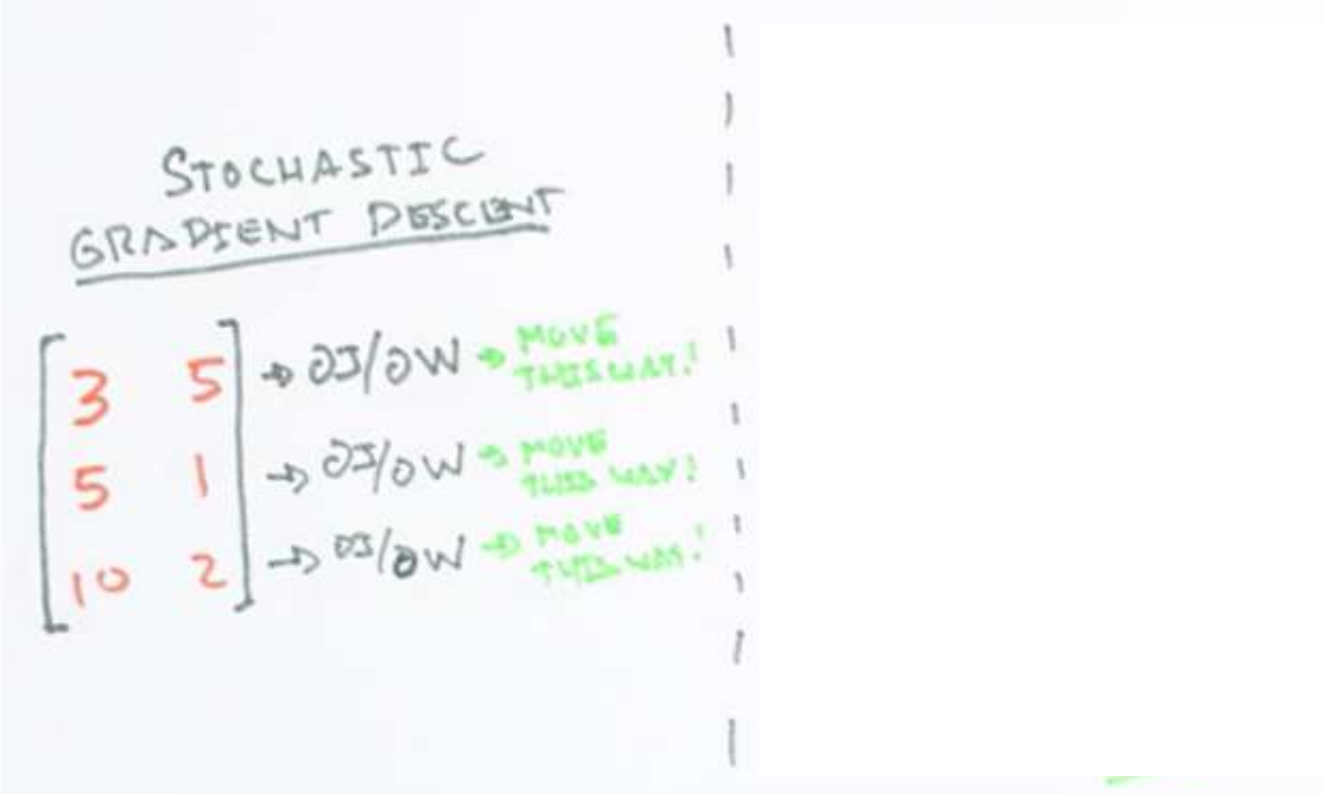


$$J = \sum \frac{1}{2}\left(y - f\left(f\left(XW^{(1)}\right)W^{(2)}\right)\right)^2$$

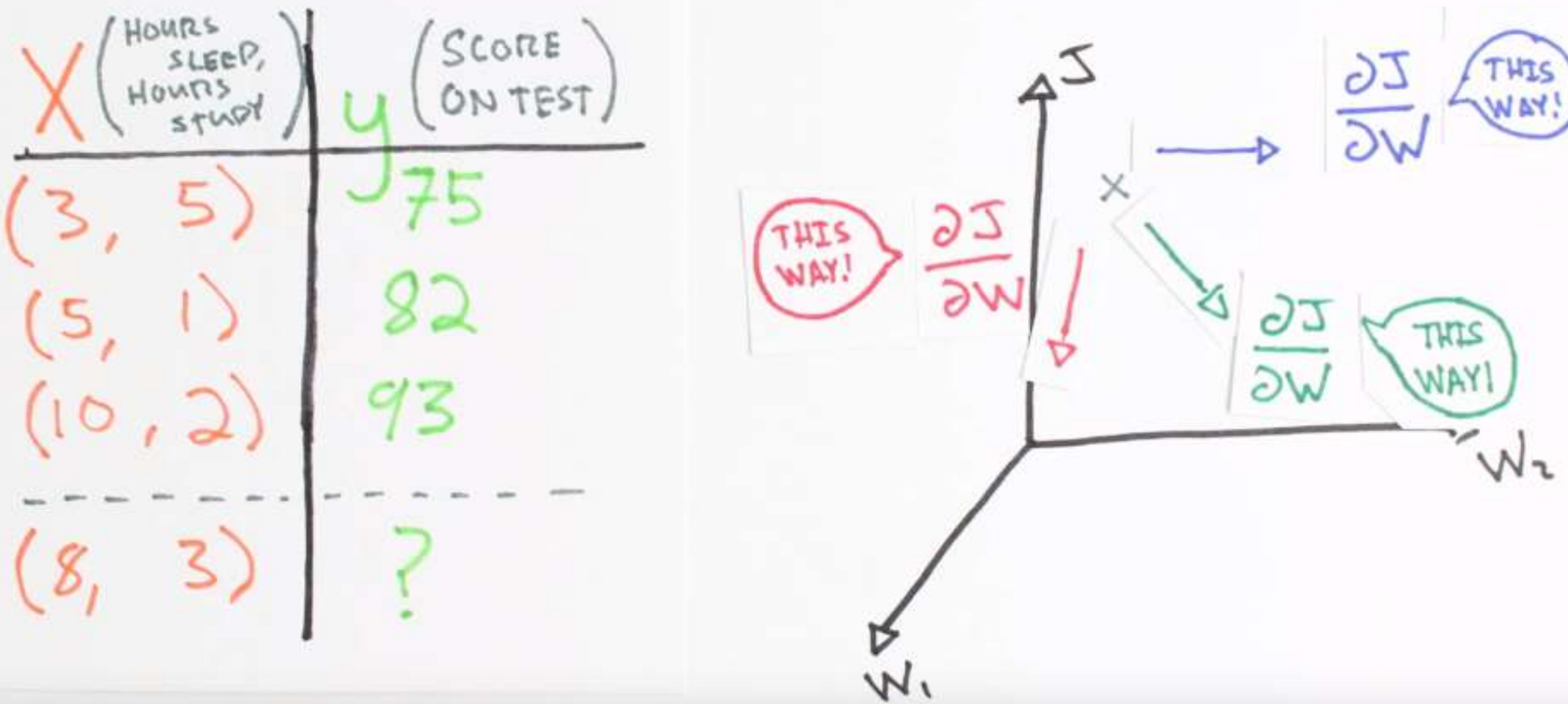
HOW DOES THIS CHANGE IF I CHANGE THESE?

$$\frac{\partial J}{\partial W}$$

Multi-Neuron Networks :: Gradient Descent

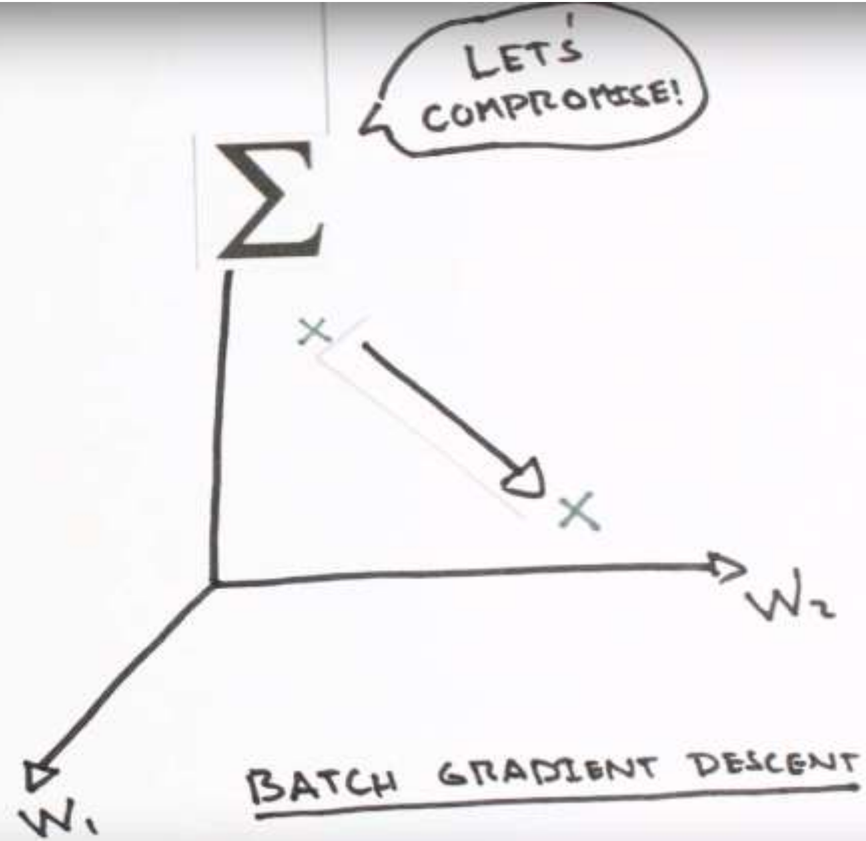


Multi-Neuron Networks :: Gradient Descent

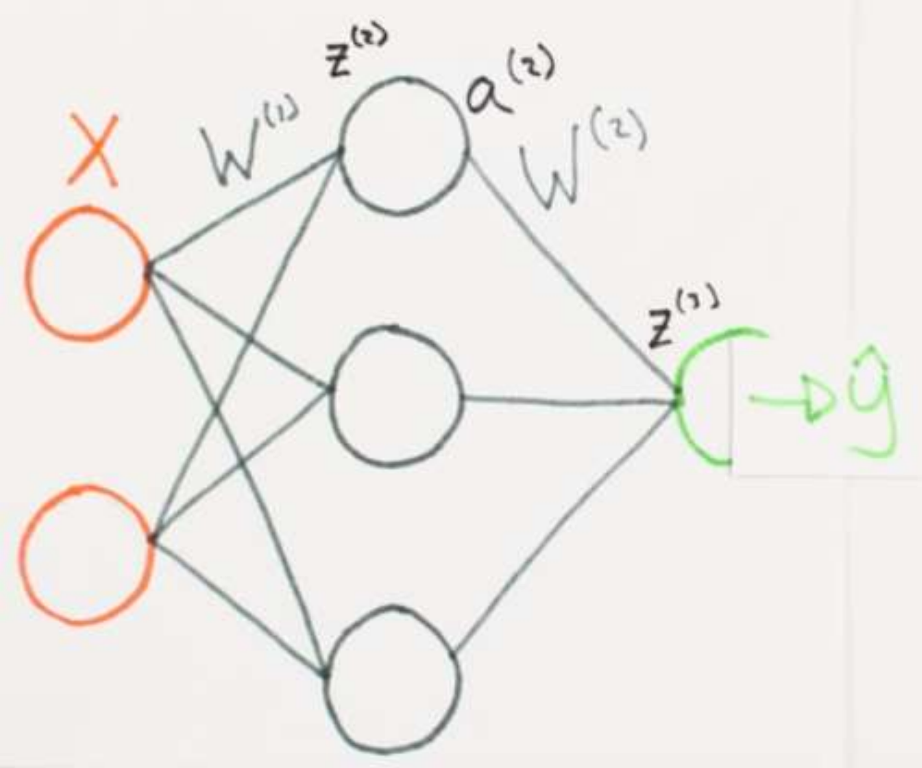


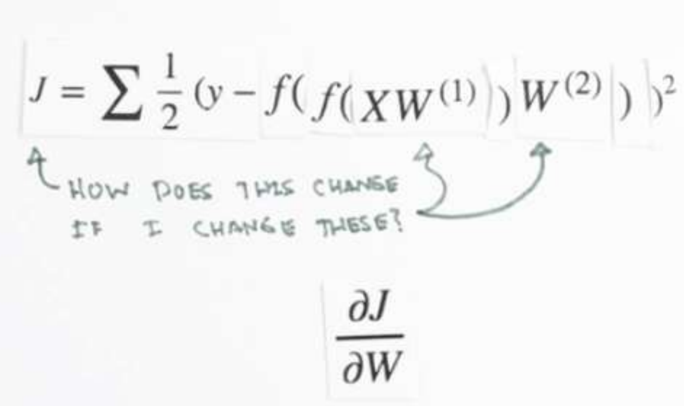
Multi-Neuron Networks :: Gradient Descent

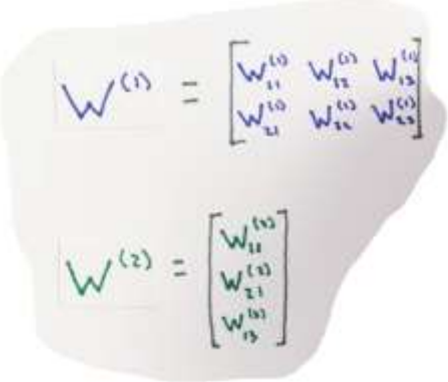
X (Hours sleep, Hours study)	y (Score on test)
(3, 5)	75
(5, 1)	82
(10, 2)	93
(8, 3)	?



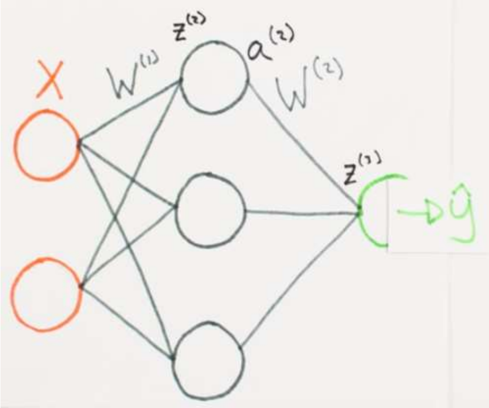
Multi-Neuron Networks :: Backpropagation







Multi-Neuron Networks :: Backpropagation



$$J = \sum \frac{1}{2}(y - f(f(XW^{(1)})W^{(2)}))^2$$

HOW DOES THIS CHANGE IF I CHANGE THESE?

$$\frac{\partial J}{\partial W}$$

$$W^{(1)} = \begin{bmatrix} W_{11}^{(1)} & W_{12}^{(1)} & W_{13}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} & W_{23}^{(1)} \end{bmatrix} \qquad \frac{\partial J}{\partial W^{(1)}} = \begin{bmatrix} \frac{\partial J}{\partial W_{11}^{(1)}} & \frac{\partial J}{\partial W_{12}^{(1)}} & \frac{\partial J}{\partial W_{13}^{(1)}} \\ \frac{\partial J}{\partial W_{21}^{(1)}} & \frac{\partial J}{\partial W_{22}^{(1)}} & \frac{\partial J}{\partial W_{23}^{(1)}} \end{bmatrix}$$

$$W^{(2)} = \begin{bmatrix} W_{11}^{(2)} \\ W_{21}^{(2)} \\ W_{13}^{(2)} \end{bmatrix} \qquad \frac{\partial J}{\partial W^{(2)}} = \begin{bmatrix} \partial J/\partial W_{11}^{(2)} \\ \partial J/\partial W_{21}^{(2)} \\ \partial J/\partial W_{31}^{(2)} \end{bmatrix}$$

Multi-Neuron Networks :: Backpropagation

$$J = \sum \frac{1}{2}(y - \hat{y})^2$$

Adds cost from each example

$$\frac{\partial J}{\partial W^{(2)}} = \frac{\partial \sum \frac{1}{2}(y - \hat{y})^2}{\partial W^{(2)}}$$

$$z^{(2)} = XW^{(1)} \quad (1)$$

$$a^{(2)} = f(z^{(2)}) \quad (2)$$

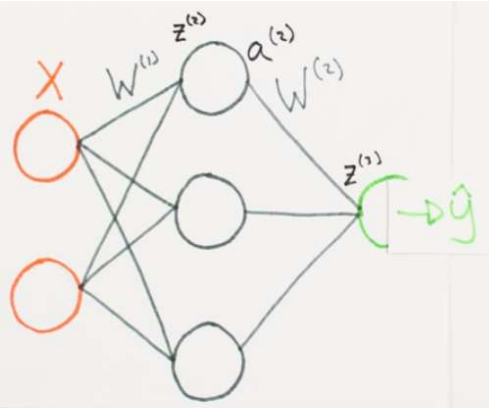
$$z^{(3)} = a^{(2)}W^{(2)} \quad (3)$$

$$\hat{y} = f(z^{(3)}) \quad (4)$$

$$J = \sum \frac{1}{2}(y - f(f(XW^{(1)})W^{(2)}))^2$$

How does this change if I change these?

$$\frac{\partial J}{\partial W}$$



$$W^{(1)} = \begin{bmatrix} W_{11}^{(1)} & W_{12}^{(1)} & W_{13}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} & W_{23}^{(1)} \end{bmatrix} \qquad \frac{\partial J}{\partial W^{(1)}} = \begin{bmatrix} \frac{\partial J}{\partial W_{11}^{(1)}} & \frac{\partial J}{\partial W_{12}^{(1)}} & \frac{\partial J}{\partial W_{13}^{(1)}} \\ \frac{\partial J}{\partial W_{21}^{(1)}} & \frac{\partial J}{\partial W_{22}^{(1)}} & \frac{\partial J}{\partial W_{23}^{(1)}} \end{bmatrix}$$

$$W^{(2)} = \begin{bmatrix} W_{11}^{(2)} \\ W_{21}^{(2)} \\ W_{13}^{(2)} \end{bmatrix} \qquad \frac{\partial J}{\partial W^{(2)}} = \begin{bmatrix} \partial J / \partial W_{11}^{(2)} \\ \partial J / \partial W_{21}^{(2)} \\ \partial J / \partial W_{31}^{(2)} \end{bmatrix}$$

Multi-Neuron Networks :: Backpropagation

$$J = \sum \frac{1}{2}(y - \hat{y})^2$$

ADDS COST FROM EACH EXAMPLE

$$\frac{\partial J}{\partial W^{(2)}} = \frac{\partial \sum \frac{1}{2}(y - \hat{y})^2}{\partial W^{(2)}}$$

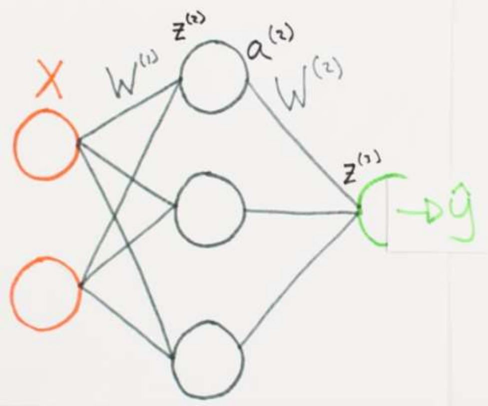
$$\boxed{\frac{\partial J}{\partial W^{(2)}} = \sum \frac{\partial \frac{1}{2}(y - \hat{y})^2}{\partial W^{(2)}}}$$

$$z^{(2)} = XW^{(1)} \quad (1)$$
$$a^{(2)} = f(z^{(2)}) \quad (2)$$
$$z^{(3)} = a^{(2)}W^{(2)} \quad (3)$$
$$\hat{y} = f(z^{(3)}) \quad (4)$$

$$J = \sum \frac{1}{2}(y - f(f(XW^{(1)})W^{(2)}))^2$$

HOW DOES THIS CHANGE IF I CHANGE THESE?

$$\frac{\partial J}{\partial W}$$



$$W^{(1)} = \begin{bmatrix} W_{11}^{(1)} & W_{12}^{(1)} & W_{13}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} & W_{23}^{(1)} \end{bmatrix} \qquad \frac{\partial J}{\partial W^{(1)}} = \begin{bmatrix} \frac{\partial J}{\partial W_{11}^{(1)}} & \frac{\partial J}{\partial W_{12}^{(1)}} & \frac{\partial J}{\partial W_{13}^{(1)}} \\ \frac{\partial J}{\partial W_{21}^{(1)}} & \frac{\partial J}{\partial W_{22}^{(1)}} & \frac{\partial J}{\partial W_{23}^{(1)}} \end{bmatrix}$$

$$W^{(2)} = \begin{bmatrix} W_{11}^{(2)} \\ W_{21}^{(2)} \\ W_{13}^{(2)} \end{bmatrix} \qquad \frac{\partial J}{\partial W^{(2)}} = \begin{bmatrix} \partial J/\partial W_{11}^{(2)} \\ \partial J/\partial W_{21}^{(2)} \\ \partial J/\partial W_{31}^{(2)} \end{bmatrix}$$

Multi-Neuron Networks :: Backpropagation

$$J = \sum \frac{1}{2}(y - \hat{y})^2$$

ADDS COST FROM EACH EXAMPLE

$$\frac{\partial J}{\partial W^{(2)}} = \frac{\partial \sum \frac{1}{2}(y - \hat{y})^2}{\partial W^{(2)}}$$

$$\frac{\partial J}{\partial W^{(2)}} = \sum \frac{\partial \frac{1}{2}(y - \hat{y})^2}{\partial W^{(2)}}$$

$$\boxed{\frac{\partial \frac{1}{2}(y - \hat{y})^2}{\partial W^{(2)}}}$$

$$z^{(2)} = XW^{(1)} \quad (1)$$

$$a^{(2)} = f(z^{(2)}) \quad (2)$$

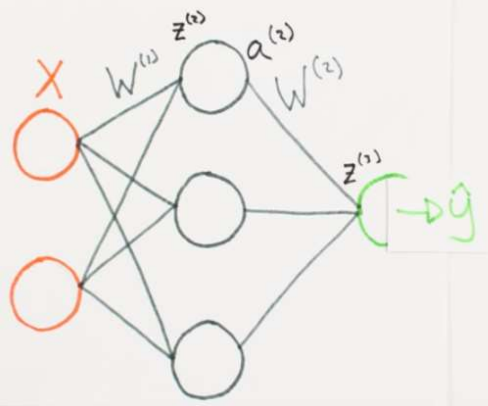
$$z^{(3)} = a^{(2)}W^{(2)} \quad (3)$$

$$\hat{y} = f(z^{(3)}) \quad (4)$$

$$J = \sum \frac{1}{2}(y - f(f(XW^{(1)})W^{(2)}))^2$$

HOW DOES THIS CHANGE IF I CHANGE THESE?

$$\frac{\partial J}{\partial W}$$



$$W^{(1)} = \begin{bmatrix} W_{11}^{(1)} & W_{12}^{(1)} & W_{13}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} & W_{23}^{(1)} \end{bmatrix}$$

$$\frac{\partial J}{\partial W^{(1)}} = \begin{bmatrix} \frac{\partial J}{\partial W_{11}^{(1)}} & \frac{\partial J}{\partial W_{12}^{(1)}} & \frac{\partial J}{\partial W_{13}^{(1)}} \\ \frac{\partial J}{\partial W_{21}^{(1)}} & \frac{\partial J}{\partial W_{22}^{(1)}} & \frac{\partial J}{\partial W_{23}^{(1)}} \end{bmatrix}$$

$$W^{(2)} = \begin{bmatrix} W_{11}^{(2)} \\ W_{21}^{(2)} \\ W_{13}^{(2)} \end{bmatrix}$$

$$\frac{\partial J}{\partial W^{(2)}} = \begin{bmatrix} \partial J/\partial W_{11}^{(2)} \\ \partial J/\partial W_{21}^{(2)} \\ \partial J/\partial W_{31}^{(2)} \end{bmatrix}$$

Multi-Neuron Networks :: Backpropagation

$$J = \sum \frac{1}{2}(y - \hat{y})^2$$

Adds cost from each example

$$\frac{\partial J}{\partial W^{(2)}} = \frac{\partial \sum \frac{1}{2}(y - \hat{y})^2}{\partial W^{(2)}}$$

$$\frac{\partial J}{\partial W^{(2)}} = \sum \frac{\partial \frac{1}{2}(y - \hat{y})^2}{\partial W^{(2)}}$$

$$\frac{\partial \frac{1}{2}(y - \hat{y})^2}{\partial W^{(2)}}$$

$$\boxed{\frac{\partial J}{\partial W^{(2)}} = -(y - \hat{y})\frac{\partial \hat{y}}{\partial W^{(2)}}}$$

$$z^{(2)} = XW^{(1)} \quad (1)$$

$$a^{(2)} = f(z^{(2)}) \quad (2)$$

$$z^{(3)} = a^{(2)}W^{(2)} \quad (3)$$

$$\hat{y} = f(z^{(3)}) \quad (4)$$

$$J = \sum \frac{1}{2}(y - f(f(XW^{(1)})W^{(2)}))^2$$

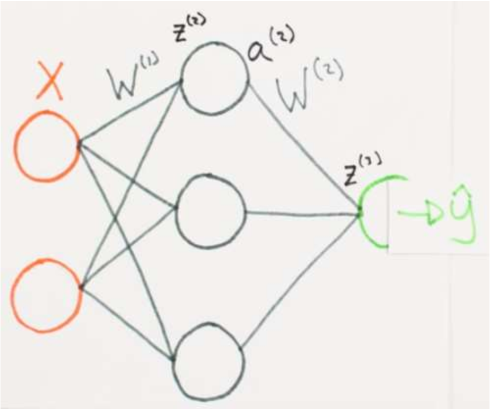
How does this change if I change these?

$$\frac{\partial J}{\partial W}$$

CHAIN RULE

ex $\frac{d}{dx}(3x + 2x^2)^2 = 2(3x + 2x^2)(3 + 6x)$

$$\boxed{\frac{dz}{dx} = \frac{dz}{dy} \cdot \frac{dy}{dx}}$$



$$W^{(1)} = \begin{bmatrix} W_{11}^{(1)} & W_{12}^{(1)} & W_{13}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} & W_{23}^{(1)} \end{bmatrix} \qquad \frac{\partial J}{\partial W^{(1)}} = \begin{bmatrix} \frac{\partial J}{\partial W_{11}^{(1)}} & \frac{\partial J}{\partial W_{12}^{(1)}} & \frac{\partial J}{\partial W_{13}^{(1)}} \\ \frac{\partial J}{\partial W_{21}^{(1)}} & \frac{\partial J}{\partial W_{22}^{(1)}} & \frac{\partial J}{\partial W_{23}^{(1)}} \end{bmatrix}$$

$$W^{(2)} = \begin{bmatrix} W_{11}^{(2)} \\ W_{21}^{(2)} \\ W_{31}^{(2)} \end{bmatrix} \qquad \frac{\partial J}{\partial W^{(2)}} = \begin{bmatrix} \partial J/\partial W_{11}^{(2)} \\ \partial J/\partial W_{21}^{(2)} \\ \partial J/\partial W_{31}^{(2)} \end{bmatrix}$$

Multi-Neuron Networks :: Backpropagation

$$J = \sum \frac{1}{2}(y - \hat{y})^2$$

ADDS COST FROM EACH EXAMPLE

$$\frac{\partial J}{\partial W^{(2)}} = \frac{\partial \sum \frac{1}{2}(y-\hat{y})^2}{\partial W^{(2)}}$$

$$\frac{\partial J}{\partial W^{(2)}} = \sum \frac{\partial \frac{1}{2}(y-\hat{y})^2}{\partial W^{(2)}}$$

$$\frac{\partial \frac{1}{2}(y-\hat{y})^2}{\partial W^{(2)}}$$

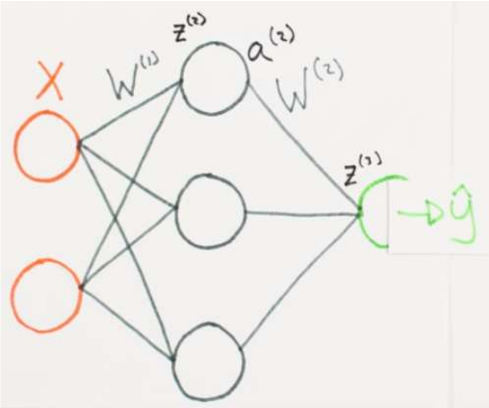
$$\boxed{\frac{\partial J}{\partial W^{(2)}} = -(y-\hat{y})\frac{\partial \hat{y}}{\partial W^{(2)}}}$$

$$z^{(2)} = XW^{(1)} \quad (1)$$
$$a^{(2)} = f(z^{(2)}) \quad (2)$$
$$z^{(3)} = a^{(2)}W^{(2)} \quad (3)$$
$$\hat{y} = f(z^{(3)}) \quad (4)$$

$$J = \sum \frac{1}{2}(y - f(f(XW^{(1)})W^{(2)}))^2$$

HOW DOES THIS CHANGE IF I CHANGE THESE?

$$\frac{\partial J}{\partial W}$$



CHAIN RULE

ex $\frac{d}{dx}(3x+2x^2)^2 = 2(3x+2x^2)(3+6x)$

$$\boxed{\frac{dz}{dx} = \frac{dz}{dy}\cdot\frac{dy}{dx}}$$

$$W^{(1)} = \begin{bmatrix} W_{11}^{(1)} & W_{12}^{(1)} & W_{13}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} & W_{23}^{(1)} \end{bmatrix} \qquad \frac{\partial J}{\partial W^{(1)}} = \begin{bmatrix} \frac{\partial J}{\partial W_{11}^{(1)}} & \frac{\partial J}{\partial W_{12}^{(1)}} & \frac{\partial J}{\partial W_{13}^{(1)}} \\ \frac{\partial J}{\partial W_{21}^{(1)}} & \frac{\partial J}{\partial W_{22}^{(1)}} & \frac{\partial J}{\partial W_{23}^{(1)}} \end{bmatrix}$$

$$W^{(2)} = \begin{bmatrix} W_{11}^{(2)} \\ W_{21}^{(2)} \\ W_{31}^{(2)} \end{bmatrix} \qquad \frac{\partial J}{\partial W^{(2)}} = \begin{bmatrix} \partial J/\partial W_{11}^{(2)} \\ \partial J/\partial W_{21}^{(2)} \\ \partial J/\partial W_{31}^{(2)} \end{bmatrix}$$

Multi-Neuron Networks :: Backpropagation

$$J = \sum \frac{1}{2}(y-\hat{y})^2$$

ADDS COST FROM EACH EXAMPLE

$$\frac{\partial J}{\partial W^{(2)}} = \frac{\partial \sum \frac{1}{2}(y-\hat{y})^2}{\partial W^{(2)}}$$

$$\frac{\partial J}{\partial W^{(2)}} = \sum \frac{\partial \frac{1}{2}(y-\hat{y})^2}{\partial W^{(2)}}$$

$$\frac{\partial \frac{1}{2}(y-\hat{y})^2}{\partial W^{(2)}}$$

$$\frac{\partial J}{\partial W^{(2)}} = -(y-\hat{y})\frac{\partial \hat{y}}{\partial W^{(2)}}$$

$$\hat{y} = f(z^{(3)})$$

$$\boxed{\frac{\partial J}{\partial W^{(2)}} = -(y-\hat{y})\frac{\partial \hat{y}}{\partial z^{(3)}}\frac{\partial z^{(3)}}{\partial W^{(2)}}}$$

$$z^{(2)} = XW^{(1)} \quad (1)$$

$$a^{(2)} = f(z^{(2)}) \quad (2)$$

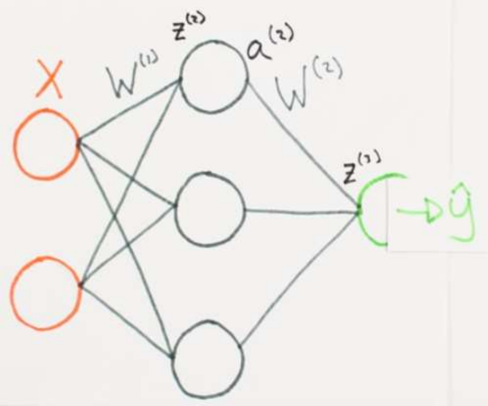
$$z^{(3)} = a^{(2)}W^{(2)} \quad (3)$$

$$\hat{y} = f(z^{(3)}) \quad (4)$$

$$J = \sum \frac{1}{2}(y - f(f(XW^{(1)})W^{(2)}))^2$$

HOW DOES THIS CHANGE IF I CHANGE THESE?

$$\frac{\partial J}{\partial W}$$



CHAIN RULE

ex $\frac{d}{dx}(3x+2x^2)^2 = 2(3x+2x^2)(3+6x)$

$$\boxed{\frac{dz}{dx} = \frac{dz}{dy}\cdot\frac{dy}{dx}}$$

Multi-Neuron Networks :: Backpropagation

$$J = \sum \frac{1}{2}(y - \hat{y})^2$$

ADDS COST FROM EACH EXAMPLE

$$\frac{\partial J}{\partial W^{(2)}} = \frac{\partial \sum \frac{1}{2}(y - \hat{y})^2}{\partial W^{(2)}}$$

$$\frac{\partial J}{\partial W^{(2)}} = \sum \frac{\partial \frac{1}{2}(y - \hat{y})^2}{\partial W^{(2)}}$$

$$\frac{\partial \frac{1}{2}(y - \hat{y})^2}{\partial W^{(2)}}$$

$$\frac{\partial J}{\partial W^{(2)}} = -(y - \hat{y})\frac{\partial \hat{y}}{\partial W^{(2)}}$$

$$\hat{y} = f(z^{(3)})$$

$$\frac{\partial J}{\partial W^{(2)}} = -(y - \hat{y})\frac{\partial \hat{y}}{\partial z^{(3)}}\frac{\partial z^{(3)}}{\partial W^{(2)}}$$

$$\frac{\partial J}{\partial W^{(2)}} = -(y - \hat{y})f'(z^{(3)})\frac{\partial z^{(3)}}{\partial W^{(2)}}$$

$$z^{(2)} = XW^{(1)} \quad (1)$$

$$a^{(2)} = f(z^{(2)}) \quad (2)$$

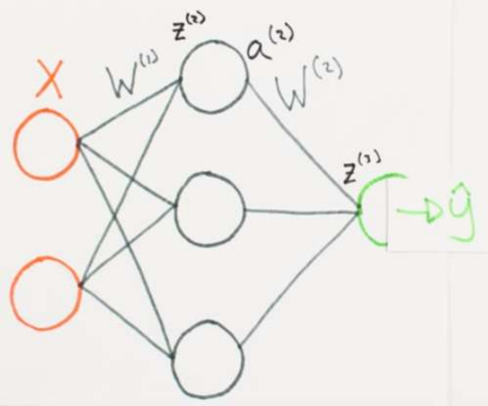
$$z^{(3)} = a^{(2)}W^{(2)} \quad (3)$$

$$\hat{y} = f(z^{(3)}) \quad (4)$$

$$J = \sum \frac{1}{2}(y - f(f(XW^{(1)})W^{(2)}))^2$$

HOW DOES THIS CHANGE IF I CHANGE THESE?

$$\frac{\partial J}{\partial W}$$



$$W^{(1)} = \begin{bmatrix} W_{11}^{(1)} & W_{12}^{(1)} & W_{13}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} & W_{23}^{(1)} \end{bmatrix} \qquad \frac{\partial J}{\partial W^{(1)}} = \begin{bmatrix} \frac{\partial J}{\partial W_{11}^{(1)}} & \frac{\partial J}{\partial W_{12}^{(1)}} & \frac{\partial J}{\partial W_{13}^{(1)}} \\ \frac{\partial J}{\partial W_{21}^{(1)}} & \frac{\partial J}{\partial W_{22}^{(1)}} & \frac{\partial J}{\partial W_{23}^{(1)}} \end{bmatrix}$$

$$W^{(2)} = \begin{bmatrix} W_{11}^{(2)} \\ W_{21}^{(2)} \\ W_{31}^{(2)} \end{bmatrix} \qquad \frac{\partial J}{\partial W^{(2)}} = \begin{bmatrix} \partial J/\partial W_{11}^{(2)} \\ \partial J/\partial W_{21}^{(2)} \\ \partial J/\partial W_{31}^{(2)} \end{bmatrix}$$

Multi-Neuron Networks :: Backpropagation

$$J = \sum \frac{1}{2}(y-\hat{y})^2$$

Adds cost from each example

$$\frac{\partial J}{\partial W^{(2)}} = \frac{\partial \sum \frac{1}{2}(y-\hat{y})^2}{\partial W^{(2)}}$$

$$\frac{\partial J}{\partial W^{(2)}} = \sum \frac{\partial \frac{1}{2}(y-\hat{y})^2}{\partial W^{(2)}}$$

$$\frac{\partial \frac{1}{2}(y-\hat{y})^2}{\partial W^{(2)}}$$

$$\frac{\partial J}{\partial W^{(2)}} = -(y-\hat{y})\frac{\partial \hat{y}}{\partial W^{(2)}}$$

$$\hat{y} = f(z^{(3)})$$

$$\frac{\partial J}{\partial W^{(2)}} = -(y-\hat{y})\frac{\partial \hat{y}}{\partial z^{(3)}}\frac{\partial z^{(3)}}{\partial W^{(2)}}$$

$$\frac{\partial J}{\partial W^{(2)}} = -(y-\hat{y})f'(z^{(3)})\frac{\partial z^{(3)}}{\partial W^{(2)}}$$

$$\frac{\partial J}{\partial W^{(2)}} = (a^{(2)})^T \delta^{(3)} \qquad (6)$$

$$\delta^{(3)} = -(y-\hat{y})f'(z^{(3)})$$

Backprop error

$$z^{(2)} = XW^{(1)} \qquad (1)$$

$$a^{(2)} = f(z^{(2)}) \qquad (2)$$

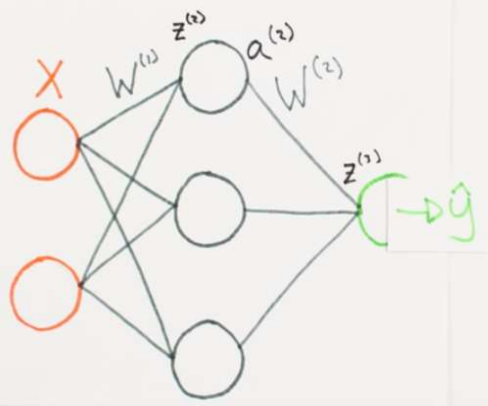
$$z^{(3)} = a^{(2)}W^{(2)} \qquad (3)$$

$$\hat{y} = f(z^{(3)}) \qquad (4)$$

$$J = \sum \frac{1}{2}(y - f(f(XW^{(1)})W^{(2)}))^2$$

HOW DOES THIS CHANGE IF I CHANGE THESE?

$$\frac{\partial J}{\partial W}$$



$$W^{(1)} = \begin{bmatrix} W_{11}^{(1)} & W_{12}^{(1)} & W_{13}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} & W_{23}^{(1)} \end{bmatrix} \qquad \frac{\partial J}{\partial W^{(1)}} = \begin{bmatrix} \frac{\partial J}{\partial W_{11}^{(1)}} & \frac{\partial J}{\partial W_{12}^{(1)}} & \frac{\partial J}{\partial W_{13}^{(1)}} \\ \frac{\partial J}{\partial W_{21}^{(1)}} & \frac{\partial J}{\partial W_{22}^{(1)}} & \frac{\partial J}{\partial W_{23}^{(1)}} \end{bmatrix}$$

$$W^{(2)} = \begin{bmatrix} W_{11}^{(2)} \\ W_{21}^{(2)} \\ W_{31}^{(2)} \end{bmatrix} \qquad \frac{\partial J}{\partial W^{(2)}} = \begin{bmatrix} \partial J/\partial W_{11}^{(2)} \\ \partial J/\partial W_{21}^{(2)} \\ \partial J/\partial W_{31}^{(2)} \end{bmatrix}$$

Multi-Neuron Networks :: Backpropagation

$$J = \sum \frac{1}{2}(y - \hat{y})^2$$

ADDS COST FROM EACH EXAMPLE

$$\frac{\partial J}{\partial W^{(2)}} = \frac{\partial \sum \frac{1}{2}(y - \hat{y})^2}{\partial W^{(2)}}$$

$$\frac{\partial J}{\partial W^{(2)}} = \sum \frac{\partial \frac{1}{2}(y - \hat{y})^2}{\partial W^{(2)}}$$

$$\frac{\partial \frac{1}{2}(y - \hat{y})^2}{\partial W^{(2)}}$$

$$\frac{\partial J}{\partial W^{(2)}} = -(y - \hat{y})\frac{\partial \hat{y}}{\partial W^{(2)}}$$

$$\hat{y} = f(z^{(3)})$$

$$\frac{\partial J}{\partial W^{(2)}} = -(y - \hat{y})\frac{\partial \hat{y}}{\partial z^{(3)}}\frac{\partial z^{(3)}}{\partial W^{(2)}}$$

$$\frac{\partial J}{\partial W^{(2)}} = -(y - \hat{y})f'(z^{(3)})\frac{\partial z^{(3)}}{\partial W^{(2)}}$$

$$\frac{\partial J}{\partial W^{(2)}} = (a^{(2)})^T \delta^{(3)} \qquad (6)$$

$$\delta^{(3)} = -(y - \hat{y})f'(z^{(3)})$$

Backprop error

$$z^{(2)} = XW^{(1)} \qquad (1)$$

$$a^{(2)} = f(z^{(2)}) \qquad (2)$$

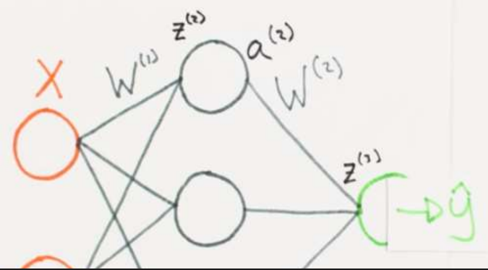
$$z^{(3)} = a^{(2)}W^{(2)} \qquad (3)$$

$$\hat{y} = f(z^{(3)}) \qquad (4)$$

$$J = \sum \frac{1}{2}(y - f(f(XW^{(1)})W^{(2)}))^2$$

HOW DOES THIS CHANGE IF I CHANGE THESE?

$$\frac{\partial J}{\partial W}$$




```
def sigmoid(self, z):
    #Apply sigmoid activation function to scalar, vector, or matrix
    return 1/(1+np.exp(-z))

def sigmoidPrime(self,z):
    #Gradient of sigmoid
    return np.exp(-z)/((1+np.exp(-z))**2)

# backpropagation
def costFunctionPrime(self, X, y):
    #Compute derivative with respect to W1 and W2 for a given X and y:
    self.yHat = self.forward(X)

    delta3 = np.multiply(-(y-self.yHat), self.sigmoidPrime(self.z3))
    dJdW2 = np.dot(self.a2.T, delta3)
```

Multi-Neuron Networks :: Backpropagation

$$J = \sum \frac{1}{2}(y - \hat{y})^2$$

$$\frac{\partial J}{\partial W^{(1)}} = \frac{\partial \sum \frac{1}{2}(y - \hat{y})^2}{\partial W^{(1)}}$$

$$z^{(2)} = XW^{(1)} \quad (1)$$

$$a^{(2)} = f(z^{(2)}) \quad (2)$$

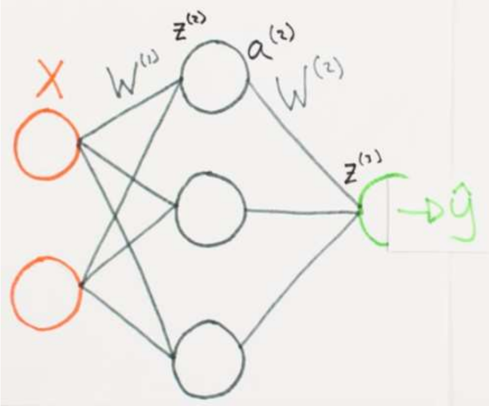
$$z^{(3)} = a^{(2)}W^{(2)} \quad (3)$$

$$\hat{y} = f(z^{(3)}) \quad (4)$$

$$J = \sum \frac{1}{2}(y - f(f(XW^{(1)})W^{(2)}))^2$$

HOW DOES THIS CHANGE IF I CHANGE THESE?

$$\frac{\partial J}{\partial W}$$



$$W^{(1)} = \begin{bmatrix} W_{11}^{(1)} & W_{12}^{(1)} & W_{13}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} & W_{23}^{(1)} \end{bmatrix} \qquad \frac{\partial J}{\partial W^{(1)}} = \begin{bmatrix} \frac{\partial J}{\partial W_{11}^{(1)}} & \frac{\partial J}{\partial W_{12}^{(1)}} & \frac{\partial J}{\partial W_{13}^{(1)}} \\ \frac{\partial J}{\partial W_{21}^{(1)}} & \frac{\partial J}{\partial W_{22}^{(1)}} & \frac{\partial J}{\partial W_{23}^{(1)}} \end{bmatrix}$$

$$W^{(2)} = \begin{bmatrix} W_{11}^{(2)} \\ W_{21}^{(2)} \\ W_{13}^{(2)} \end{bmatrix} \qquad \frac{\partial J}{\partial W^{(2)}} = \begin{bmatrix} \partial J/\partial W_{11}^{(2)} \\ \partial J/\partial W_{21}^{(2)} \\ \partial J/\partial W_{31}^{(2)} \end{bmatrix}$$

Multi-Neuron Networks :: Backpropagation

$$J = \sum \frac{1}{2}(y - \hat{y})^2$$

$$\frac{\partial J}{\partial W^{(1)}} = \frac{\partial \frac{1}{2}(y - \hat{y})^2}{\partial W^{(1)}}$$

$$\frac{\partial J}{\partial W^{(1)}} = -(y - \hat{y})\frac{\partial \hat{y}}{\partial z^{(3)}}\frac{\partial z^{(3)}}{\partial W^{(1)}}$$

$$\frac{\partial J}{\partial W^{(1)}} = -(y - \hat{y})f'(z^{(3)})\frac{\partial z^{(3)}}{\partial W^{(1)}}$$

$$z^{(2)} = XW^{(1)} \quad (1)$$

$$a^{(2)} = f(z^{(2)}) \quad (2)$$

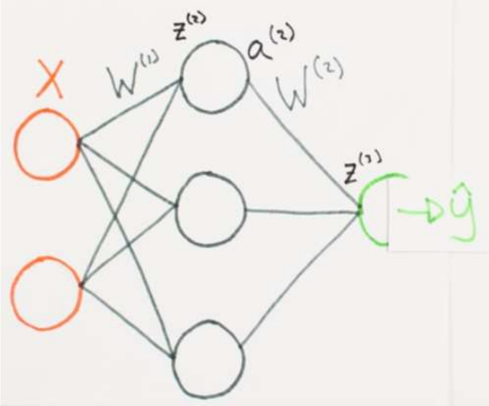
$$z^{(3)} = a^{(2)}W^{(2)} \quad (3)$$

$$\hat{y} = f(z^{(3)}) \quad (4)$$

$$J = \sum \frac{1}{2}(y - f(f(XW^{(1)})W^{(2)}))^2$$

HOW DOES THIS CHANGE IF I CHANGE THESE?

$$\frac{\partial J}{\partial W}$$



$$W^{(1)} = \begin{bmatrix} W_{11}^{(1)} & W_{12}^{(1)} & W_{13}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} & W_{23}^{(1)} \end{bmatrix}$$

$$\frac{\partial J}{\partial W^{(1)}} = \begin{bmatrix} \frac{\partial J}{\partial W_{11}^{(1)}} & \frac{\partial J}{\partial W_{12}^{(1)}} & \frac{\partial J}{\partial W_{13}^{(1)}} \\ \frac{\partial J}{\partial W_{21}^{(1)}} & \frac{\partial J}{\partial W_{22}^{(1)}} & \frac{\partial J}{\partial W_{23}^{(1)}} \end{bmatrix}$$

$$W^{(2)} = \begin{bmatrix} W_{11}^{(2)} \\ W_{21}^{(2)} \\ W_{31}^{(2)} \end{bmatrix}$$

$$\frac{\partial J}{\partial W^{(2)}} = \begin{bmatrix} \partial J / \partial W_{11}^{(2)} \\ \partial J / \partial W_{21}^{(2)} \\ \partial J / \partial W_{31}^{(2)} \end{bmatrix}$$

Multi-Neuron Networks :: Backpropagation

$$J = \sum \frac{1}{2}(y - \hat{y})^2$$

$$\frac{\partial J}{\partial W^{(1)}} = \frac{\partial \sum \frac{1}{2}(y-\hat{y})^2}{\partial W^{(1)}}$$

$$\frac{\partial J}{\partial W^{(1)}} = -(y-\hat{y})\frac{\partial \hat{y}}{\partial z^{(3)}}\frac{\partial z^{(3)}}{\partial W^{(1)}}$$

$$\frac{\partial J}{\partial W^{(1)}} = -(y-\hat{y})f'(z^{(3)})\frac{\partial z^{(3)}}{\partial W^{(1)}}$$

$$\delta^{(3)} = -(y-\hat{y})f'(z^{(3)})$$

$$\frac{\partial J}{\partial W^{(1)}} = \delta^{(3)}\frac{\partial z^{(3)}}{\partial W^{(1)}}$$

$$z^{(2)} = XW^{(1)} \quad (1)$$

$$a^{(2)} = f(z^{(2)}) \quad (2)$$

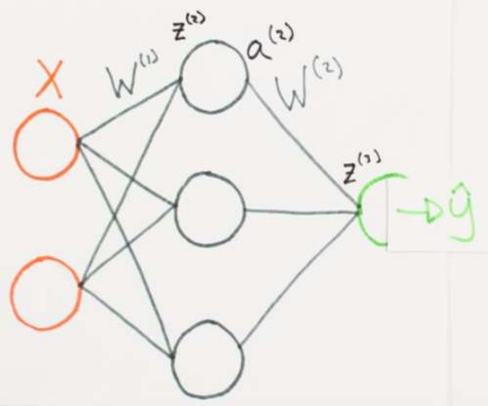
$$z^{(3)} = a^{(2)}W^{(2)} \quad (3)$$

$$\hat{y} = f(z^{(3)}) \quad (4)$$

$$J = \sum \frac{1}{2}(y - f(f(XW^{(1)})W^{(2)}))^2$$

HOW DOES THIS CHANGE IF I CHANGE THESE?

$$\frac{\partial J}{\partial W}$$



$$W^{(1)} = \begin{bmatrix} W_{11}^{(1)} & W_{12}^{(1)} & W_{13}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} & W_{23}^{(1)} \end{bmatrix}$$

$$\frac{\partial J}{\partial W^{(1)}} = \begin{bmatrix} \frac{\partial J}{\partial W_{11}^{(1)}} & \frac{\partial J}{\partial W_{12}^{(1)}} & \frac{\partial J}{\partial W_{13}^{(1)}} \\ \frac{\partial J}{\partial W_{21}^{(1)}} & \frac{\partial J}{\partial W_{22}^{(1)}} & \frac{\partial J}{\partial W_{23}^{(1)}} \end{bmatrix}$$

$$W^{(2)} = \begin{bmatrix} W_{11}^{(2)} \\ W_{21}^{(2)} \\ W_{31}^{(2)} \end{bmatrix}$$

$$\frac{\partial J}{\partial W^{(2)}} = \begin{bmatrix} \partial J/\partial W_{11}^{(2)} \\ \partial J/\partial W_{21}^{(2)} \\ \partial J/\partial W_{31}^{(2)} \end{bmatrix}$$

Multi-Neuron Networks :: Backpropagation

$$J = \sum \frac{1}{2}(y - f(f(XW^{(1)})W^{(2)}))^2$$

HOW DOES THIS CHANGE IF I CHANGE THESE?

$$\frac{\partial J}{\partial W}$$

$$J = \sum \frac{1}{2}(y - \hat{y})^2$$

$$\frac{\partial J}{\partial W^{(1)}} = \frac{\partial \sum \frac{1}{2}(y - \hat{y})^2}{\partial W^{(1)}}$$

$$\frac{\partial J}{\partial W^{(1)}} = -(y - \hat{y})\frac{\partial \hat{y}}{\partial z^{(3)}}\frac{\partial z^{(3)}}{\partial W^{(1)}}$$

$$\frac{\partial J}{\partial W^{(1)}} = -(y - \hat{y})f'(z^{(3)})\frac{\partial z^{(3)}}{\partial W^{(1)}}$$

$$\delta^{(3)} = -(y - \hat{y})f'(z^{(3)})$$

$$\frac{\partial J}{\partial W^{(1)}} = \delta^{(3)}\frac{\partial z^{(3)}}{\partial W^{(1)}}$$

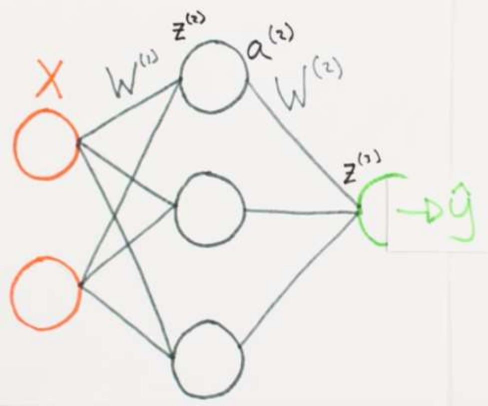
$$\frac{\partial J}{\partial W^{(1)}} = \delta^{(3)}\frac{\partial z^{(3)}}{\partial a^{(2)}}\frac{\partial a^{(2)}}{\partial W^{(1)}}$$

$$z^{(2)} = XW^{(1)} \quad (1)$$

$$a^{(2)} = f(z^{(2)}) \quad (2)$$

$$z^{(3)} = a^{(2)}W^{(2)} \quad (3)$$

$$\hat{y} = f(z^{(3)}) \quad (4)$$



$$W^{(1)} = \begin{bmatrix} W_{11}^{(1)} & W_{12}^{(1)} & W_{13}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} & W_{23}^{(1)} \end{bmatrix} \qquad \frac{\partial J}{\partial W^{(1)}} = \begin{bmatrix} \frac{\partial J}{\partial W_{11}^{(1)}} & \frac{\partial J}{\partial W_{12}^{(1)}} & \frac{\partial J}{\partial W_{13}^{(1)}} \\ \frac{\partial J}{\partial W_{21}^{(1)}} & \frac{\partial J}{\partial W_{22}^{(1)}} & \frac{\partial J}{\partial W_{23}^{(1)}} \end{bmatrix}$$

$$W^{(2)} = \begin{bmatrix} W_{11}^{(2)} \\ W_{21}^{(2)} \\ W_{13}^{(2)} \end{bmatrix} \qquad \frac{\partial J}{\partial W^{(2)}} = \begin{bmatrix} \partial J/\partial W_{11}^{(2)} \\ \partial J/\partial W_{21}^{(2)} \\ \partial J/\partial W_{31}^{(2)} \end{bmatrix}$$

Multi-Neuron Networks :: Backpropagation

$$J = \sum \frac{1}{2}(y - \hat{y})^2$$

$$\frac{\partial J}{\partial W^{(1)}} = \frac{\partial \sum \frac{1}{2}(y - \hat{y})^2}{\partial W^{(1)}}$$

$$\frac{\partial J}{\partial W^{(1)}} = -(y - \hat{y})\frac{\partial \hat{y}}{\partial z^{(3)}}\frac{\partial z^{(3)}}{\partial W^{(1)}}$$

$$\frac{\partial J}{\partial W^{(1)}} = -(y - \hat{y})f'(z^{(3)})\frac{\partial z^{(3)}}{\partial W^{(1)}}$$

$$\frac{\partial J}{\partial W^{(1)}} = \delta^{(3)}\frac{\partial z^{(3)}}{\partial W^{(1)}}$$

$$\frac{\partial J}{\partial W^{(1)}} = \delta^{(3)}\frac{\partial z^{(3)}}{\partial a^{(2)}}\frac{\partial a^{(2)}}{\partial W^{(1)}}$$

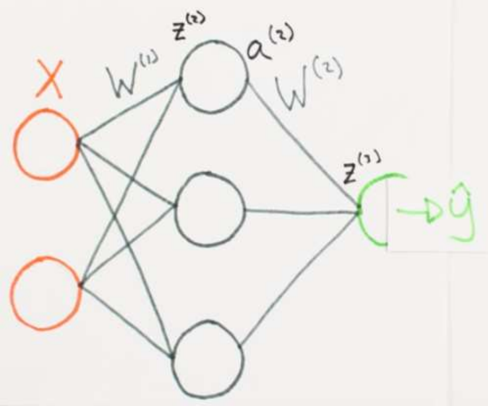
$$\frac{\partial J}{\partial W^{(1)}} = \delta^{(3)}(W^{(2)})^T\frac{\partial a^{(2)}}{\partial W^{(1)}}$$

$$z^{(2)} = XW^{(1)} \quad (1)$$
$$a^{(2)} = f(z^{(2)}) \quad (2)$$
$$z^{(3)} = a^{(2)}W^{(2)} \quad (3)$$
$$\hat{y} = f(z^{(3)}) \quad (4)$$

$$J = \sum \frac{1}{2}(y - f(f(XW^{(1)})W^{(2)}))^2$$

HOW DOES THIS CHANGE IF I CHANGE THESE?

$$\frac{\partial J}{\partial W}$$



$$W^{(1)} = \begin{bmatrix} W_{11}^{(1)} & W_{12}^{(1)} & W_{13}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} & W_{23}^{(1)} \end{bmatrix} \qquad \frac{\partial J}{\partial W^{(1)}} = \begin{bmatrix} \frac{\partial J}{\partial W_{11}^{(1)}} & \frac{\partial J}{\partial W_{12}^{(1)}} & \frac{\partial J}{\partial W_{13}^{(1)}} \\ \frac{\partial J}{\partial W_{21}^{(1)}} & \frac{\partial J}{\partial W_{22}^{(1)}} & \frac{\partial J}{\partial W_{23}^{(1)}} \end{bmatrix}$$

$$W^{(2)} = \begin{bmatrix} W_{11}^{(2)} \\ W_{21}^{(2)} \\ W_{31}^{(2)} \end{bmatrix} \qquad \frac{\partial J}{\partial W^{(2)}} = \begin{bmatrix} \partial J/\partial W_{11}^{(2)} \\ \partial J/\partial W_{21}^{(2)} \\ \partial J/\partial W_{31}^{(2)} \end{bmatrix}$$

Multi-Neuron Networks :: Backpropagation

$$J = \sum \frac{1}{2}(y - \hat{y})^2$$

$$\frac{\partial J}{\partial W^{(1)}} = \frac{\partial \sum \frac{1}{2}(y - \hat{y})^2}{\partial W^{(1)}}$$

$$\frac{\partial J}{\partial W^{(1)}} = -(y - \hat{y})\frac{\partial \hat{y}}{\partial z^{(3)}} \frac{\partial z^{(3)}}{\partial W^{(1)}}$$

$$\frac{\partial J}{\partial W^{(1)}} = -(y - \hat{y}) f'(z^{(3)}) \frac{\partial z^{(3)}}{\partial W^{(1)}}$$

$$\frac{\partial J}{\partial W^{(1)}} = \delta^{(3)} \frac{\partial z^{(3)}}{\partial W^{(1)}}$$

$$\frac{\partial J}{\partial W^{(1)}} = \delta^{(3)} \frac{\partial z^{(3)}}{\partial a^{(2)}} \frac{\partial a^{(2)}}{\partial W^{(1)}}$$

$$\frac{\partial J}{\partial W^{(1)}} = \delta^{(3)} (W^{(2)})^T \frac{\partial a^{(2)}}{\partial W^{(1)}}$$

$$\frac{\partial J}{\partial W^{(1)}} = \delta^{(3)} (W^{(2)})^T \frac{\partial a^{(2)}}{\partial z^{(2)}} \frac{\partial z^{(2)}}{\partial W^{(1)}}$$

$$\frac{\partial J}{\partial W^{(1)}} = \delta^{(3)} (W^{(2)})^T f'(z^{(2)}) \frac{\partial z^{(2)}}{\partial W^{(1)}}$$

$$z^{(2)} = XW^{(1)} \quad (1)$$

$$a^{(2)} = f(z^{(2)}) \quad (2)$$

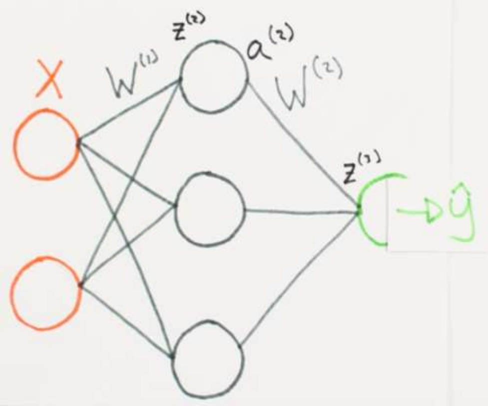
$$z^{(3)} = a^{(2)}W^{(2)} \quad (3)$$

$$\hat{y} = f(z^{(3)}) \quad (4)$$

$$J = \sum \frac{1}{2}(y - f(f(XW^{(1)})W^{(2)}))^2$$

HOW DOES THIS CHANGE IF I CHANGE THESE?

$$\frac{\partial J}{\partial W}$$



$$W^{(1)} = \begin{bmatrix} W_{11}^{(1)} & W_{12}^{(1)} & W_{13}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} & W_{23}^{(1)} \end{bmatrix}$$

$$\frac{\partial J}{\partial W^{(1)}} = \begin{bmatrix} \frac{\partial J}{\partial W_{11}^{(1)}} & \frac{\partial J}{\partial W_{12}^{(1)}} & \frac{\partial J}{\partial W_{13}^{(1)}} \\ \frac{\partial J}{\partial W_{21}^{(1)}} & \frac{\partial J}{\partial W_{22}^{(1)}} & \frac{\partial J}{\partial W_{23}^{(1)}} \end{bmatrix}$$

$$W^{(2)} = \begin{bmatrix} W_{11}^{(2)} \\ W_{21}^{(2)} \\ W_{13}^{(2)} \end{bmatrix}$$

$$\frac{\partial J}{\partial W^{(2)}} = \begin{bmatrix} \partial J / \partial W_{11}^{(2)} \\ \partial J / \partial W_{21}^{(2)} \\ \partial J / \partial W_{31}^{(2)} \end{bmatrix}$$

Multi-Neuron Networks :: Backpropagation

$$J = \sum \frac{1}{2}(y - \hat{y})^2$$

$$\frac{\partial J}{\partial W^{(1)}} = \frac{\partial \frac{1}{2}(y - \hat{y})^2}{\partial W^{(1)}}$$

$$\frac{\partial J}{\partial W^{(1)}} = -(y - \hat{y})\frac{\partial \hat{y}}{\partial z^{(3)}}\frac{\partial z^{(3)}}{\partial W^{(1)}}$$

$$\frac{\partial J}{\partial W^{(1)}} = -(y - \hat{y})f'(z^{(3)})\frac{\partial z^{(3)}}{\partial W^{(1)}}$$

$$\frac{\partial J}{\partial W^{(1)}} = \delta^{(3)}\frac{\partial z^{(3)}}{\partial W^{(1)}}$$

$$\frac{\partial J}{\partial W^{(1)}} = \delta^{(3)}\frac{\partial z^{(3)}}{\partial a^{(2)}}\frac{\partial a^{(2)}}{\partial W^{(1)}}$$

$$\frac{\partial J}{\partial W^{(1)}} = \delta^{(3)}(W^{(2)})^T\frac{\partial a^{(2)}}{\partial W^{(1)}}$$

$$\frac{\partial J}{\partial W^{(1)}} = \delta^{(3)}(W^{(2)})^T\frac{\partial a^{(2)}}{\partial z^{(2)}}\frac{\partial z^{(2)}}{\partial W^{(1)}}$$

$$\frac{\partial J}{\partial W^{(1)}} = \delta^{(3)}(W^{(2)})^T f'(z^{(2)})\frac{\partial z^{(2)}}{\partial W^{(1)}}$$

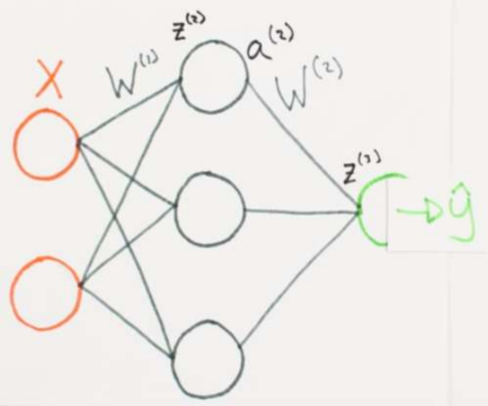
$$\frac{\partial J}{\partial W^{(1)}} = X^T\overbrace{\delta^{(3)}(W^{(2)})^T f'(z^{(2)})}^{\delta^{(2)}}$$

$$z^{(2)} = XW^{(1)} \qquad (1)$$
$$a^{(2)} = f(z^{(2)}) \qquad (2)$$
$$z^{(3)} = a^{(2)}W^{(2)} \qquad (3)$$
$$\hat{y} = f(z^{(3)}) \qquad (4)$$

$$J = \sum \frac{1}{2}(y - f(f(XW^{(1)})W^{(2)}))^2$$

HOW DOES THIS CHANGE
IF I CHANGE THESE?

$$\frac{\partial J}{\partial W}$$



$$W^{(1)} = \begin{bmatrix} W^{(1)}_{11} & W^{(1)}_{12} & W^{(1)}_{13} \\ W^{(1)}_{21} & W^{(1)}_{22} & W^{(1)}_{23} \end{bmatrix} \qquad \frac{\partial J}{\partial W^{(1)}} = \begin{bmatrix} \frac{\partial J}{\partial W^{(1)}_{11}} & \frac{\partial J}{\partial W^{(1)}_{12}} & \frac{\partial J}{\partial W^{(1)}_{13}} \\ \frac{\partial J}{\partial W^{(1)}_{21}} & \frac{\partial J}{\partial W^{(1)}_{22}} & \frac{\partial J}{\partial W^{(1)}_{23}} \end{bmatrix}$$

$$W^{(2)} = \begin{bmatrix} W^{(2)}_{11} \\ W^{(2)}_{21} \\ W^{(2)}_{31} \end{bmatrix} \qquad \frac{\partial J}{\partial W^{(2)}} = \begin{bmatrix} \partial J/\partial W^{(2)}_{11} \\ \partial J/\partial W^{(2)}_{21} \\ \partial J/\partial W^{(2)}_{31} \end{bmatrix}$$

Multi-Neuron Networks :: Backpropagation

$$J = \sum \frac{1}{2}(y - f(f(XW^{(1)})W^{(2)}))^2$$

HOW DOES THIS CHANGE IF I CHANGE THESE?

$$\frac{\partial J}{\partial W}$$

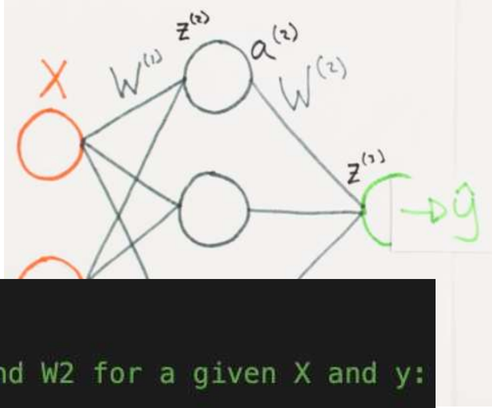
$$J = \sum \frac{1}{2}(y - \hat{y})^2$$

$$z^{(2)} = XW^{(1)} \quad (1)$$

$$a^{(2)} = f(z^{(2)}) \quad (2)$$

$$z^{(3)} = a^{(2)}W^{(2)} \quad (3)$$

$$\hat{y} = f(z^{(3)}) \quad (4)$$



$$\frac{\partial J}{\partial W^{(2)}} = (a^{(2)})^T \delta^{(3)}$$

$$\delta^{(3)} = -(y - \hat{y})f'(z^{(3)})$$

$$\frac{\partial J}{\partial W^{(1)}} = X^T \delta^{(2)}$$

$$\delta^{(2)} = \delta^{(3)}(W^{(2)})^T f'(z^{(2)})$$

```
# backpropagation
def costFunctionPrime(self, X, y):
    #Compute derivative with respect to W1 and W2 for a given X and y:
    self.yHat = self.forward(X)

    delta3 = np.multiply(-(y-self.yHat), self.sigmoidPrime(self.z3))
    dJdW2 = np.dot(self.a2.T, delta3)

    delta2 = np.dot(delta3, self.W2.T)*self.sigmoidPrime(self.z2)
    dJdW1 = np.dot(X.T, delta2)

    return dJdW1, dJdW2
```

$$\frac{\partial J}{\partial W^{(1)}_{11}} \quad \frac{\partial J}{\partial W^{(1)}_{12}}$$

Multi-Neuron Networks :: Backpropagation

$$J = \sum \frac{1}{2}(y - \hat{y})^2$$

$$\frac{\partial J}{\partial W^{(2)}} = (a^{(2)})^T \delta^{(3)}$$

$$\delta^{(3)} = -(y - \hat{y}) f'(z^{(3)})$$

$$\frac{\partial J}{\partial W^{(1)}} = X^T \delta^{(2)}$$

$$\delta^{(2)} = \delta^{(3)} (W^{(2)})^T f'(z^{(2)})$$

$$z^{(2)} = XW^{(1)} \quad (1)$$

$$a^{(2)} = f(z^{(2)}) \quad (2)$$

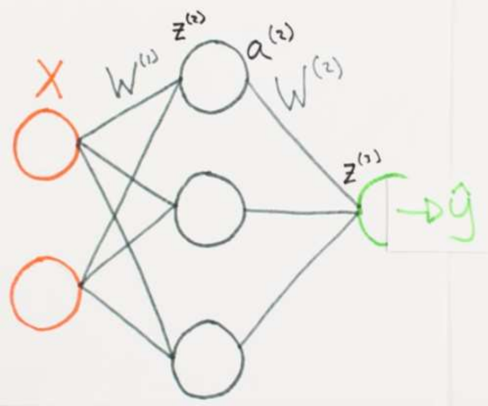
$$z^{(3)} = a^{(2)} W^{(2)} \quad (3)$$

$$\hat{y} = f(z^{(3)}) \quad (4)$$

$$J = \sum \frac{1}{2}(y - f(f(XW^{(1)})W^{(2)}))^2$$

HOW DOES THIS CHANGE IF I CHANGE THESE?

$$\frac{\partial J}{\partial W}$$




```
NN = Neural_Network()
cost1 = NN.costFunction(X, y)
print('cost1=',cost1)
dJdW1, dJdW2 = NN.costFunctionPrime(X, y)
print('dJ/dW1=',dJdW1)
print('dJ/dW2=',dJdW2)
eta = 0.01
NN.W1 = NN.W1 - eta * dJdW1
NN.W2 = NN.W2 - eta * dJdW2
cost2 = NN.costFunction(X, y)
print('cost2=',cost2)
```

```
cost1= [0.44735371]
dJ/dW1= [[-0.08913117 -0.04750461 -0.00562623]
 [-0.05862425 -0.03130539 -0.00351033]]
dJ/dW2= [[-0.4110688 ]
 [-0.37530217]
 [-0.4590466 ]]
cost2= [0.44202336]
```

$$W^{(1)} = \begin{bmatrix} W_{11}^{(1)} & W_{12}^{(1)} & W_{13}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} & W_{23}^{(1)} \end{bmatrix} \qquad \frac{\partial J}{\partial W^{(1)}} = \begin{bmatrix} \frac{\partial J}{\partial W_{11}^{(1)}} & \frac{\partial J}{\partial W_{12}^{(1)}} & \frac{\partial J}{\partial W_{13}^{(1)}} \\ \frac{\partial J}{\partial W_{21}^{(1)}} & \frac{\partial J}{\partial W_{22}^{(1)}} & \frac{\partial J}{\partial W_{23}^{(1)}} \end{bmatrix}$$

$$W^{(2)} = \begin{bmatrix} W_{11}^{(2)} \\ W_{21}^{(2)} \\ W_{31}^{(2)} \end{bmatrix} \qquad \frac{\partial J}{\partial W^{(2)}} = \begin{bmatrix} \partial J/\partial W_{11}^{(2)} \\ \partial J/\partial W_{21}^{(2)} \\ \partial J/\partial W_{31}^{(2)} \end{bmatrix}$$

Multi-Neuron Networks :: training

Initialize network with random weights

While [not converged]

Do forward prop

Do backprop and determine change in weights

Update all weights in all layers

```
NN = Neural_Network()
cost1 = NN.costFunction(X, y)
print('cost1=',cost1)
dJdW1, dJdW2 = NN.costFunctionPrime(X, y)
print('dJ/dW1=',dJdW1)
print('dJ/dW2=',dJdW2)
eta = 0.01
NN.W1 = NN.W1 - eta * dJdW1
NN.W2 = NN.W2 - eta * dJdW2
```

One Iteration

Multi-Neuron Networks :: training

Initialize network with random weights

While [not converged]

Do forward prop

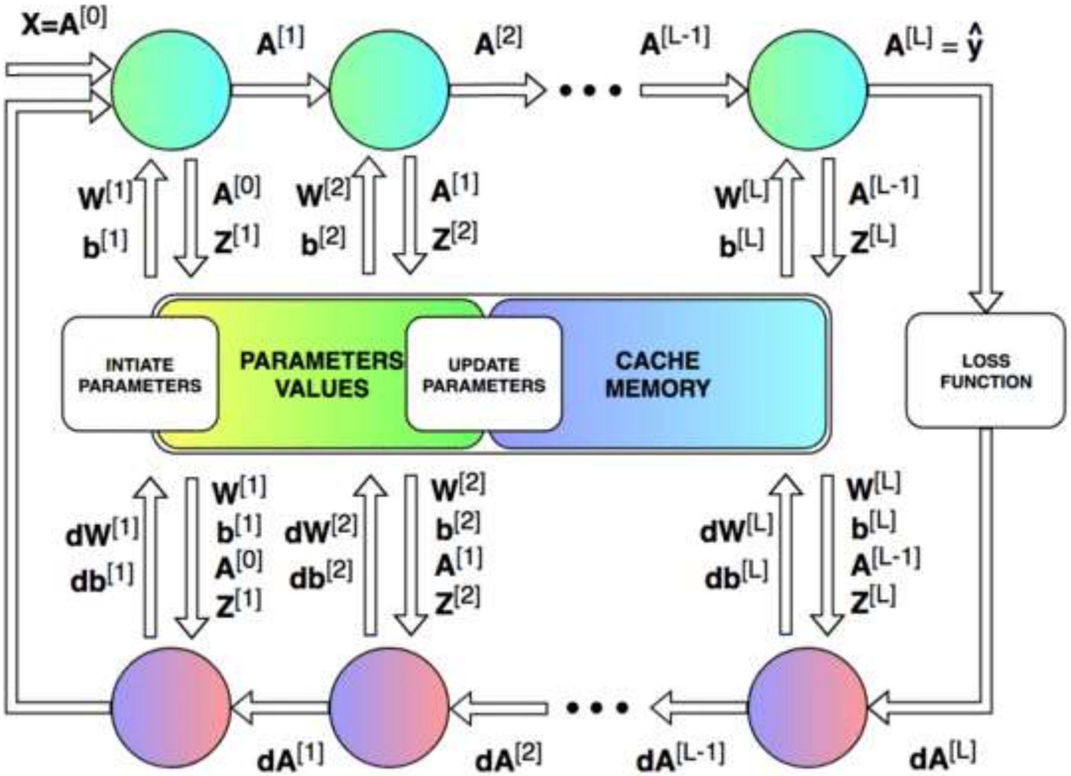
Do backprop and determine change in weights

Update All weights in All layers

$$\mathbf{w}^{(\mathbf{t+1})} = \mathbf{w}^{(\mathbf{t})} - \eta^{(t)} \nabla_{\mathbf{w}} \mathbf{J}(\mathbf{w})$$

One Iteration

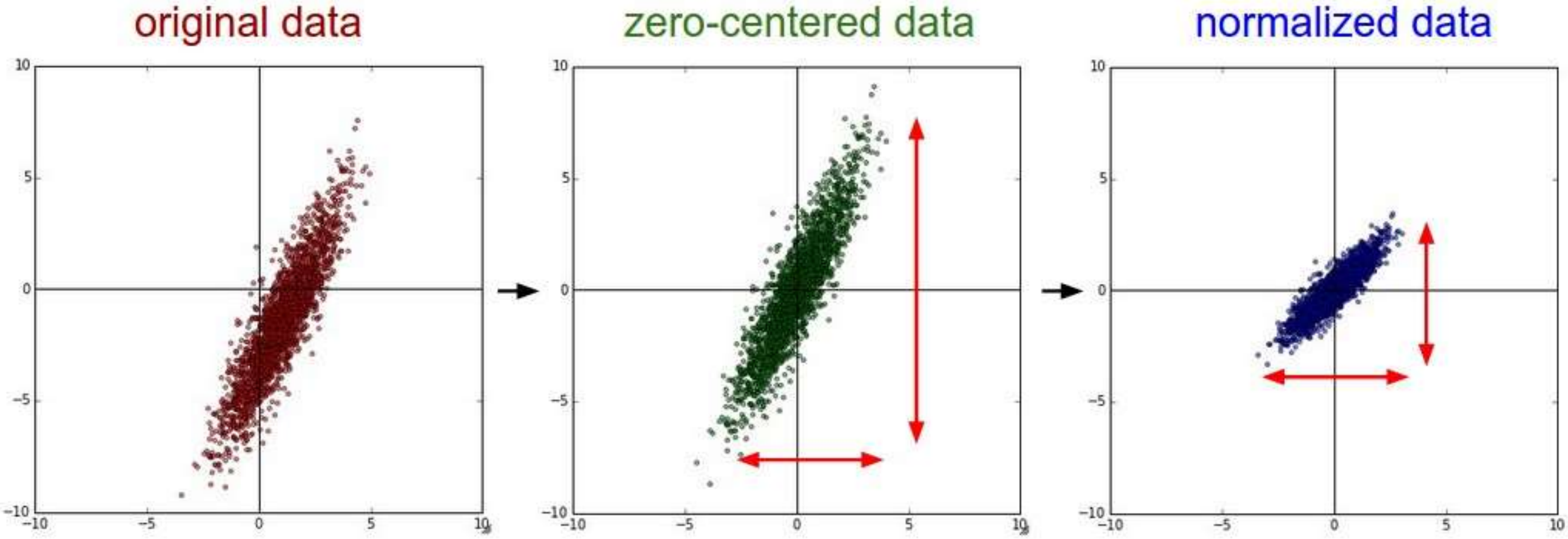
FORWARD PROPAGATION



BACKWARD PROPAGATION

Data Setup

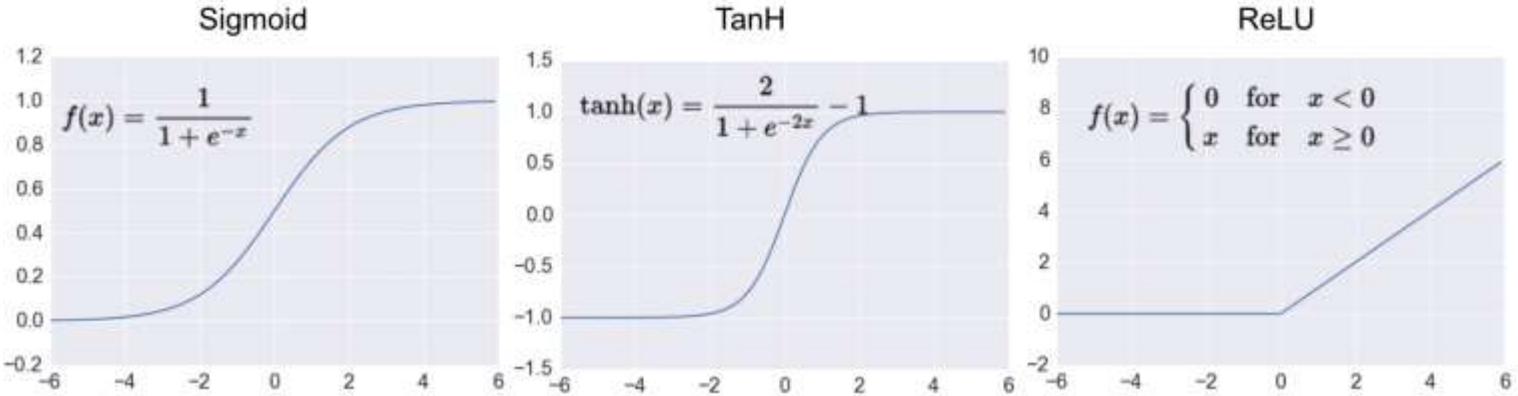
- Preprocessing:



Weight initialization

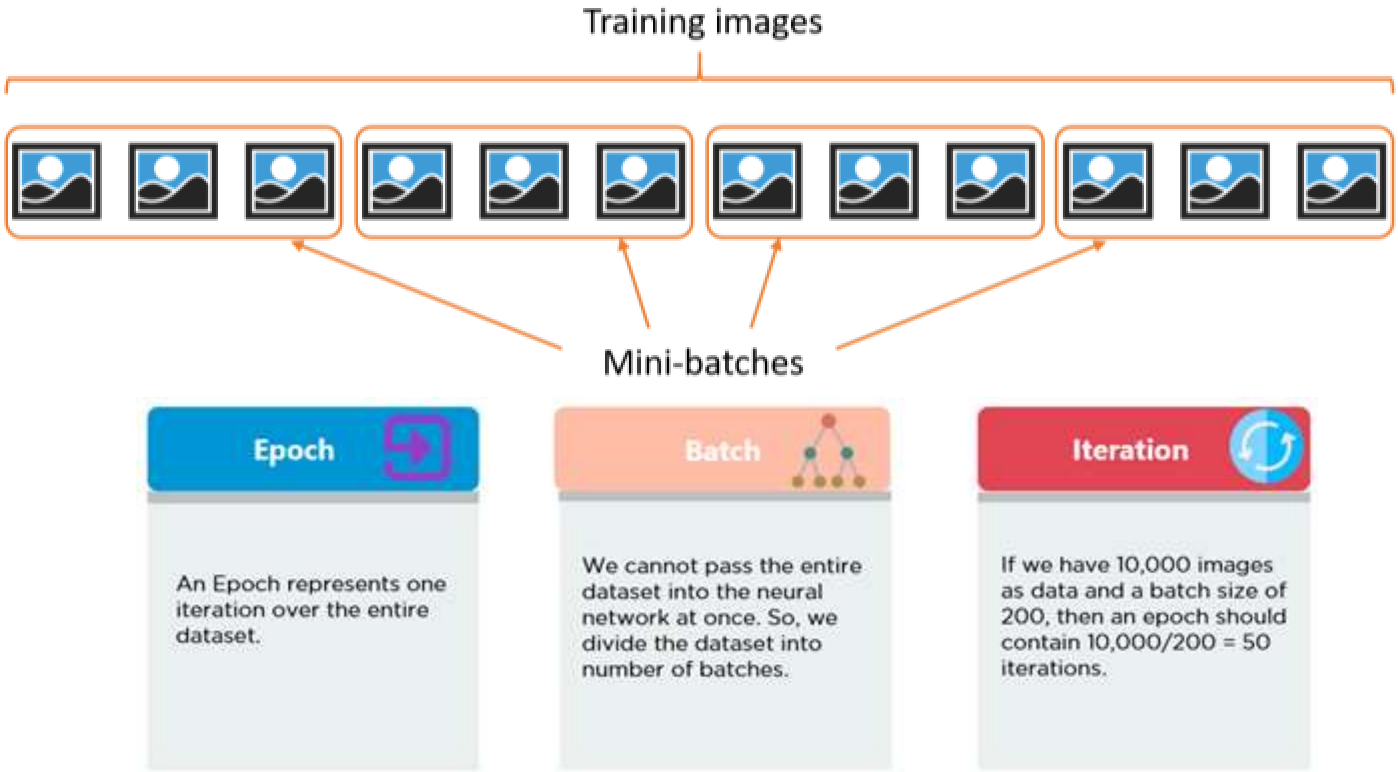
- All Zeros
- Random [0,1]
- Random [-1,1]
- `w = np.random.randn(n) * sqrt(2.0/n), n = # of inputs to neuron`

Activation functions



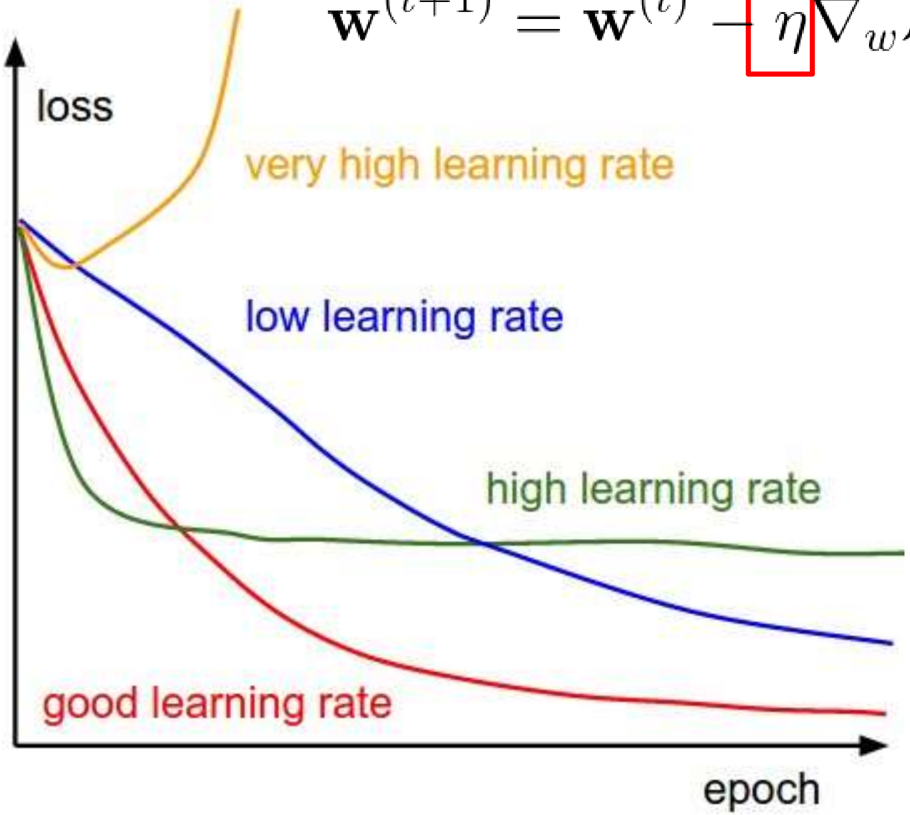
MINIBATCH VS SINGLE

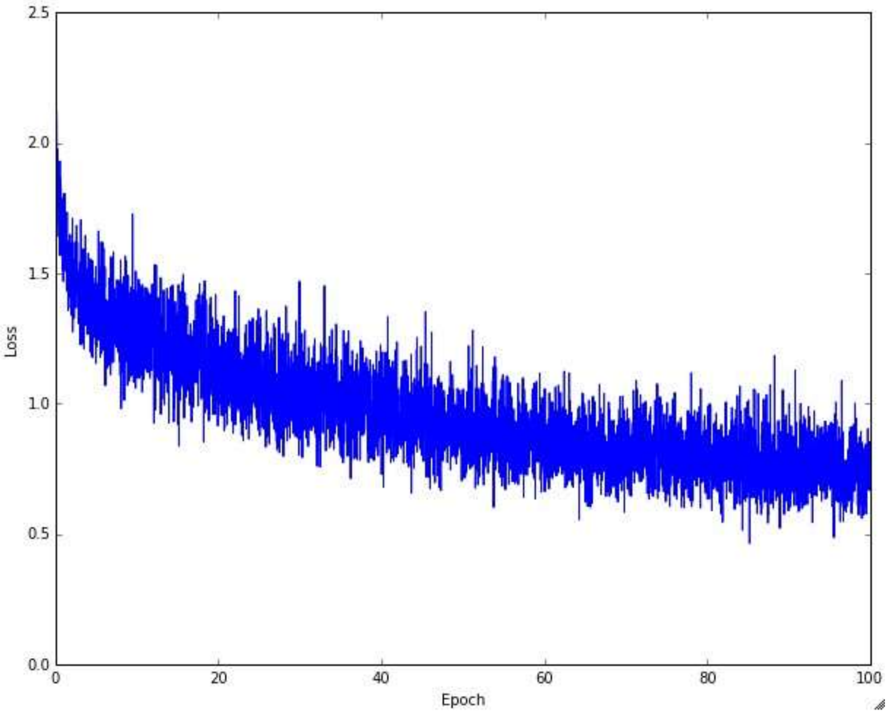
- Average error, gradients



Training – setting learning rate

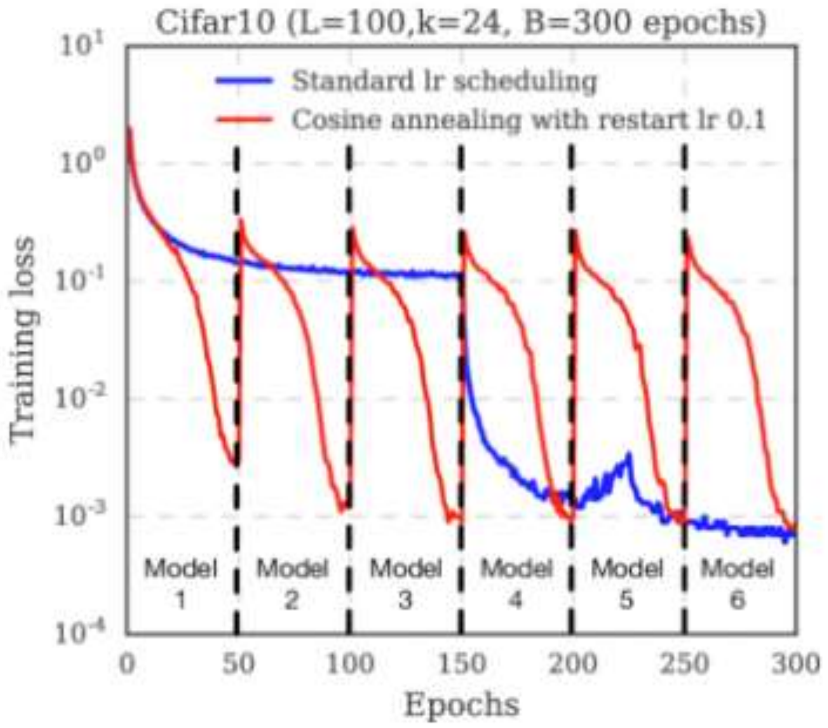
$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \boxed{\eta}\nabla_w \mathcal{L}(\mathbf{w})$$

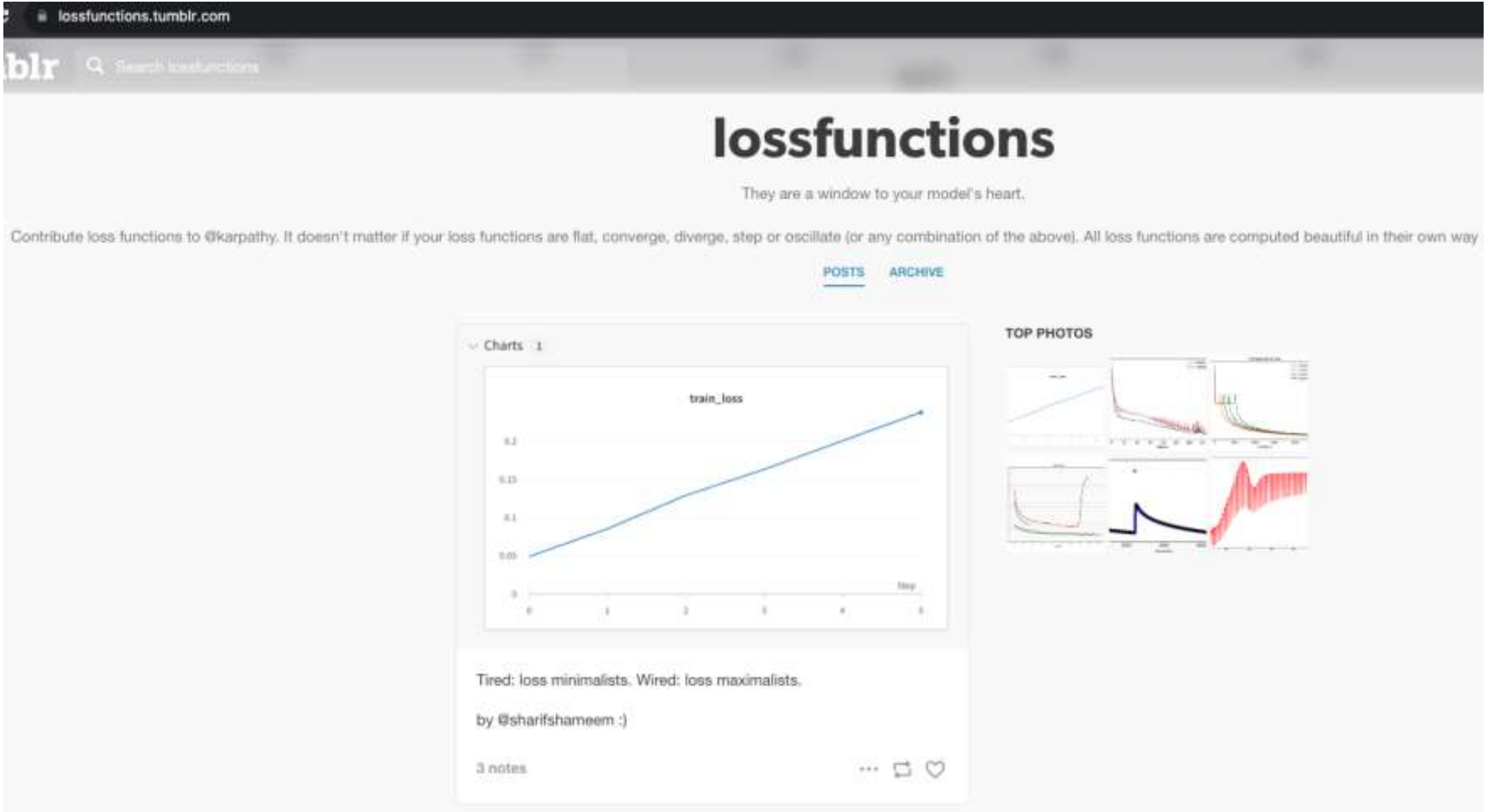


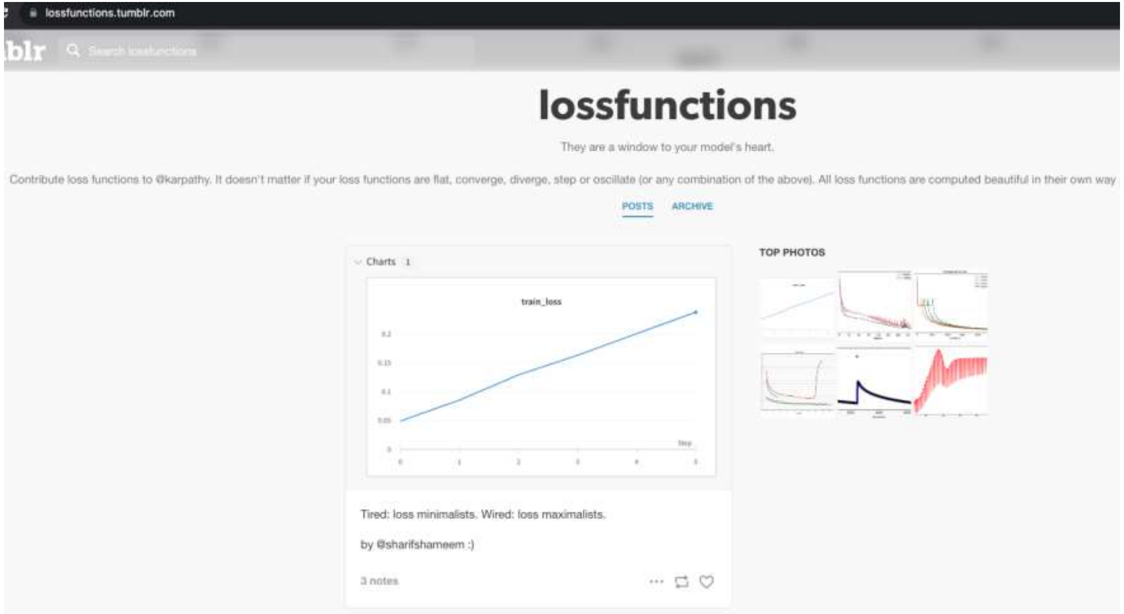


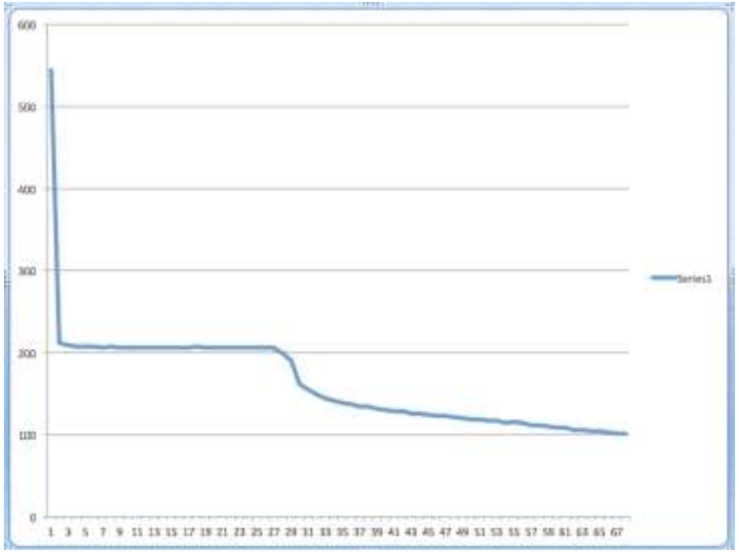
Training – setting learning rate

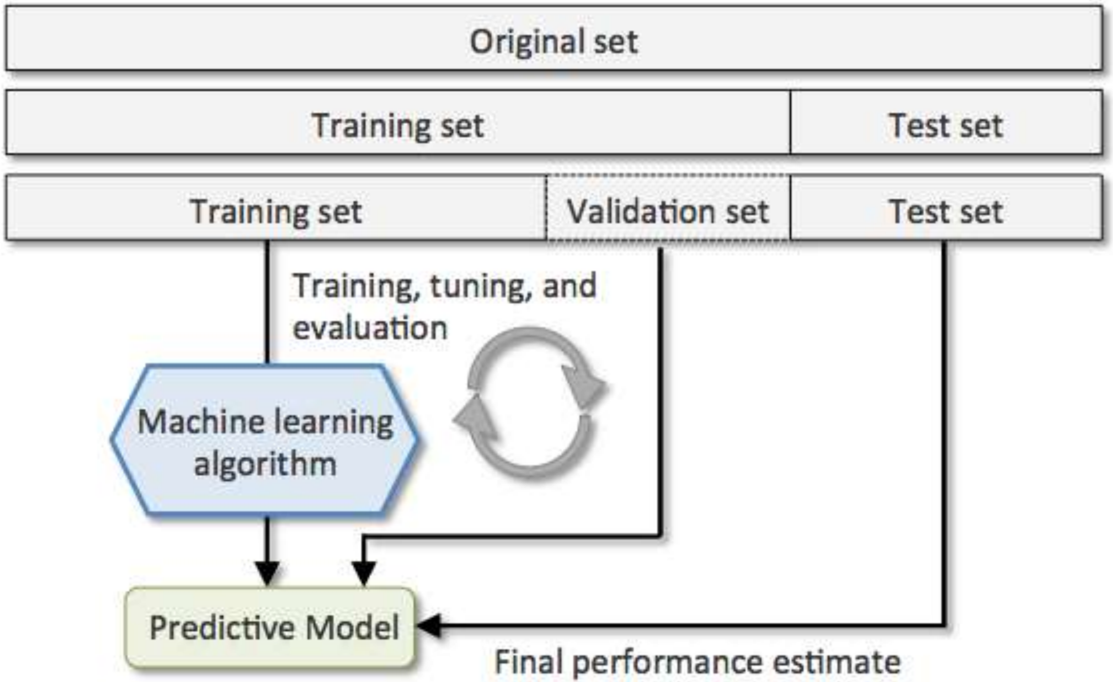
$$\mathbf{w}^{(\mathbf{t+1})} = \mathbf{w}^{(\mathbf{t})} - \boxed{\eta^{(t)}}\nabla_{\mathbf{w}}\mathbf{J}(\mathbf{w})$$



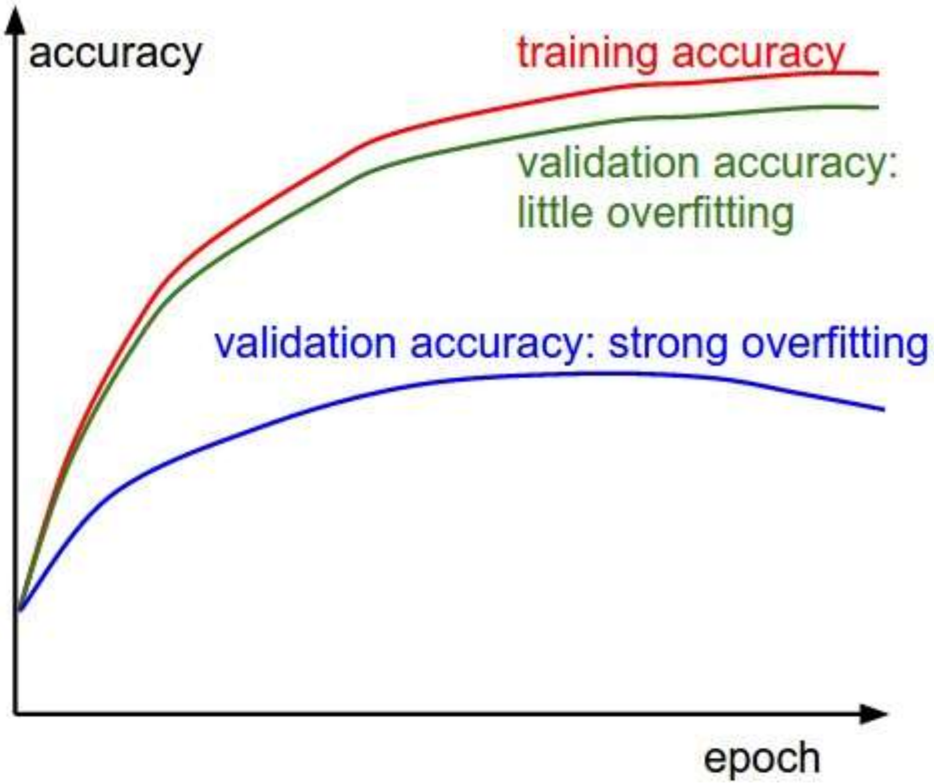

lossfunctions.tumblr.com
lossfunctions
They are a window to your model's heart.
Contribute loss functions to @karpathy. It doesn't matter if your loss functions are flat, converge, diverge, step or oscillate (or any combination of the above). All loss functions are computed beautiful in their own way
POSTS
ARCHIVE
Charts 1
train_loss
Tired: loss minimalists. Wired: loss maximalists.
by @sharifshameem :)
3 notes
TOP PHOTOS


lossfunctions.tumblr.com
lossfunctions
They are a window to your model's heart.
Contribute loss functions to @karpathy. It doesn't matter if your loss functions are flat, converge, diverge, step or oscillate (or any combination of the above). All loss functions are computed beautiful in their own way
POSTS
ARCHIVE
Charts 1
train_loss
TOP PHOTOS
Tired: loss minimalists. Wired: loss maximalists.
by @sharifshameem :)
3 notes

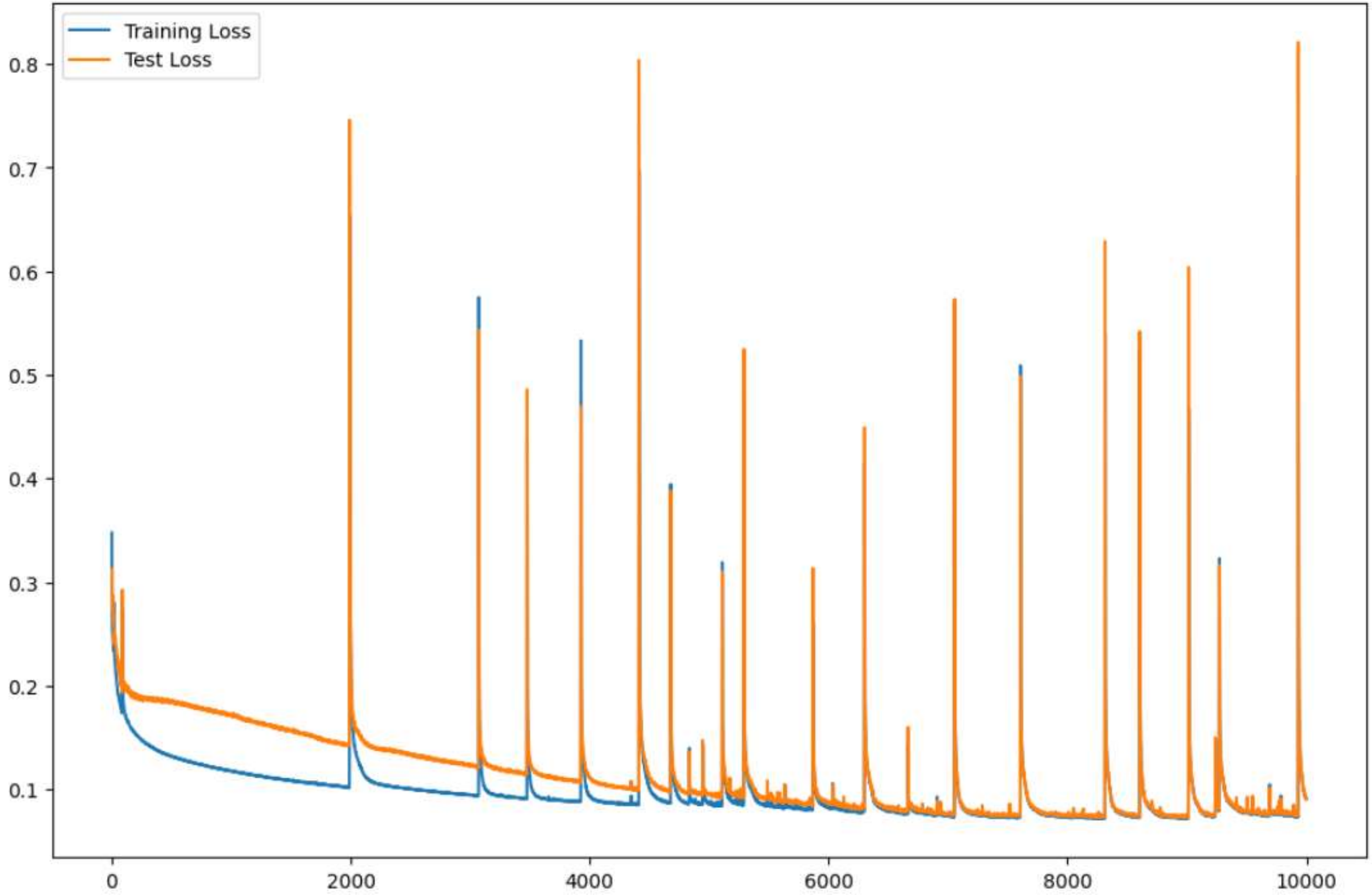

600
500
400
300
200
100
0
Series1


Original set
Training set
Test set
Training set
Validation set
Test set
Training, tuning, and evaluation
Machine learning algorithm
Predictive Model
Final performance estimate

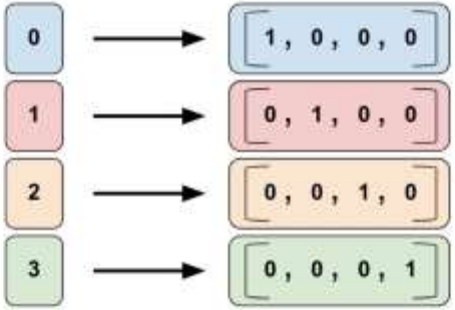
When to stop training



When to stop training

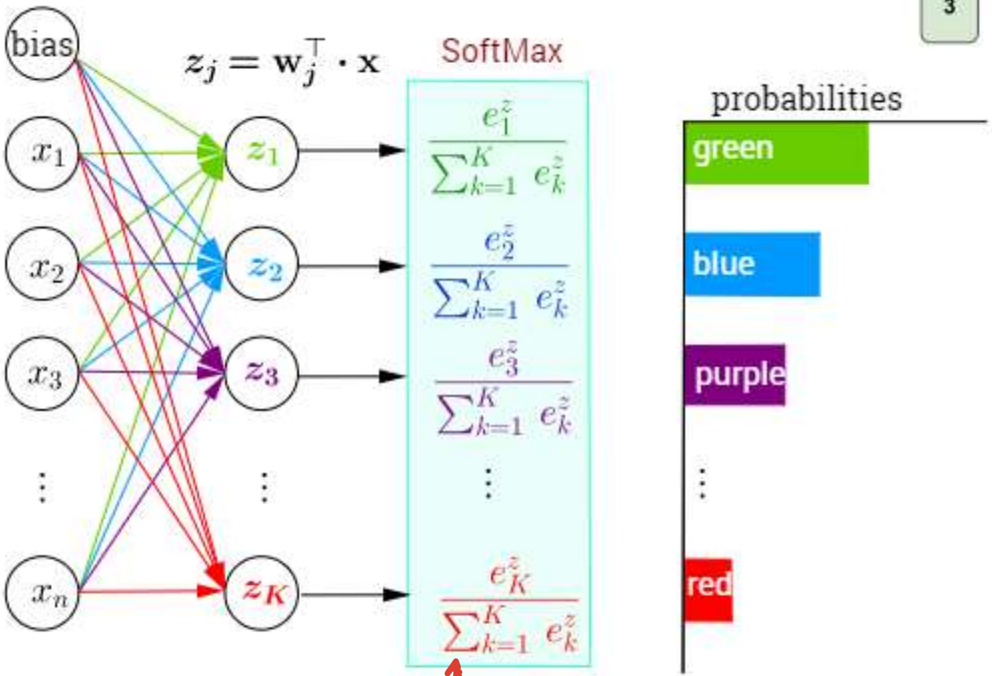

Training Loss
Test Loss
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
2000
4000
6000
8000
10000

Classification loss



Multi-Class Classification with NN and SoftMax Function

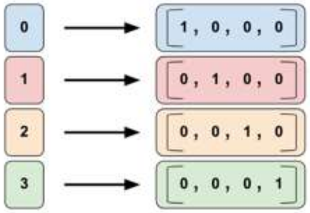
$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_K \end{bmatrix} = \begin{bmatrix} \mathbf{w}_1^\top \\ \mathbf{w}_2^\top \\ \mathbf{w}_3^\top \\ \vdots \\ \mathbf{w}_K^\top \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}$$

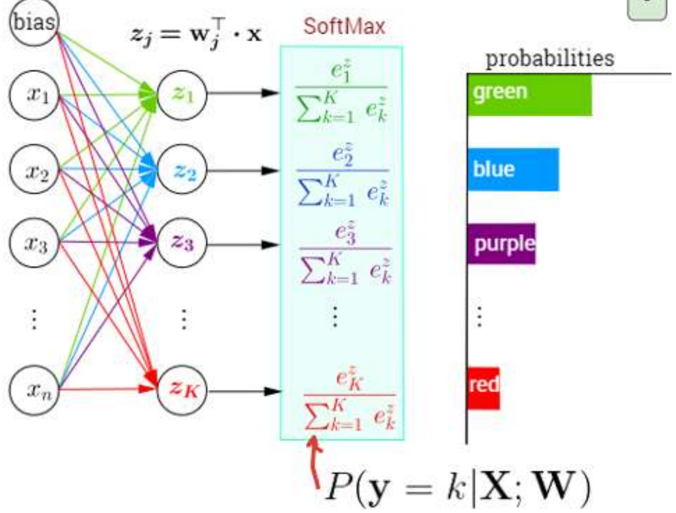


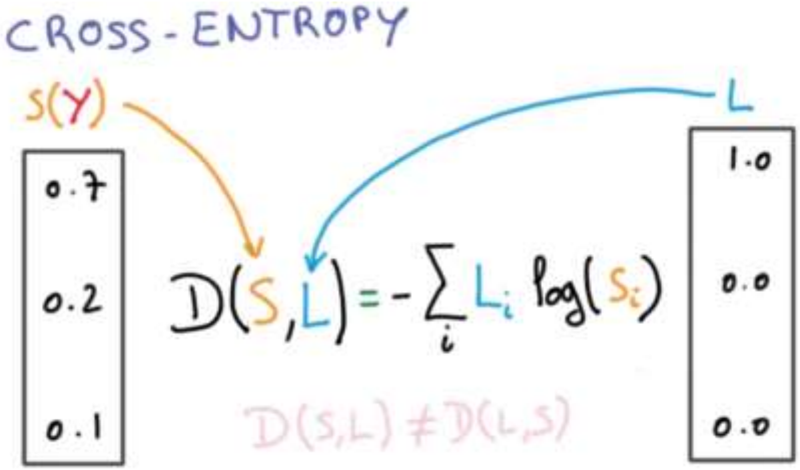
$$P(\mathbf{y} = k | \mathbf{X}; \mathbf{W})$$

Classification Loss

$$
\begin{aligned}
D_{\mathrm{KL}}(p|q) &= \sum_i p_i \log \frac{p_i}{q_i} \\
&= \sum_i (-p_i \log q_i + p_i \log p_i) \\
&= -\sum_i p_i \log q_i + \sum_i p_i \log p_i \\
&= -\sum_i p_i \log q_i - \sum_i p_i \log \frac{1}{p_i} \\
&= -\sum_i p_i \log q_i - H(p) \\
&= \sum_i p_i \log \frac{1}{q_i} - H(p)
\end{aligned}
$$







Resources

- Videos

- Example from lecture: https://www.youtube.com/watch?v=bxe2T-V8XRs&list=PLiaHhY2iBX9hdHaRr6b7XevZtgZRa1PoU (contains some technical bugs in some places which have been corrected in the lecture)

- 3Blue1Brown: https://www.youtube.com/watch?v=aircAruvnKk&list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi

- StatQuest: https://www.youtube.com/watch?v=zxagGtF9MeU&list=PLblh5JKOoLUIxGDQs4LFFD--41Vzf-ME1

- NN zero to hero: https://www.youtube.com/watch?v=VMj-3S1tku0&list=PLAqhIrjkxbuWI23v9cThsA9GvCAUhRvKZ

- https://theaisummer.com/weights-and-biases-tutorial/