Artificial Neural Network

Pritam Prakash Shete
Computer Division, BARC
Centre for Excellence in Basic Sciences

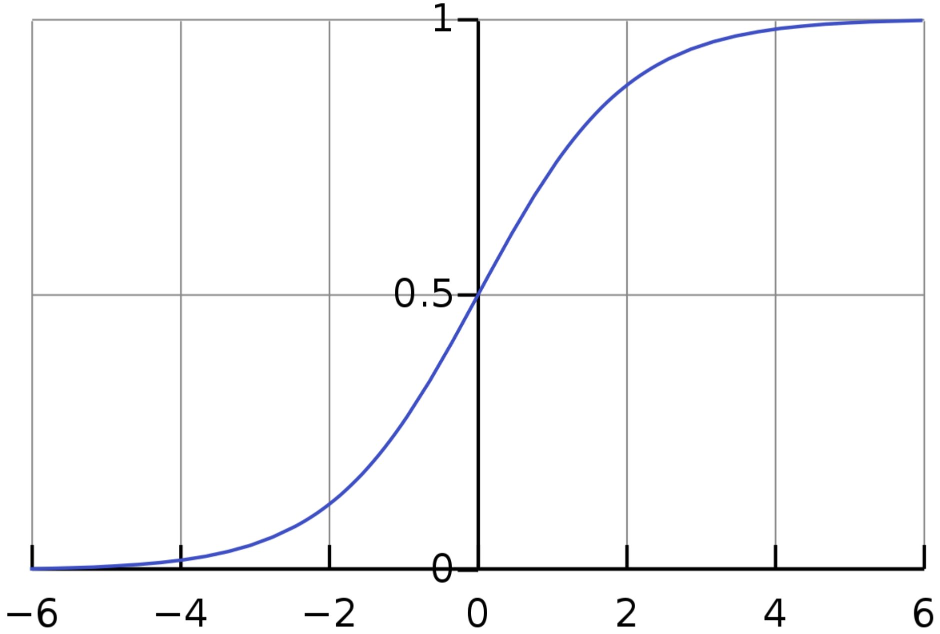
Topics

- Linear regression
- Logistic regression
- Artificial neural network
- Gradient descent
- Back propagation
- Activation functions

Linear Regression

- Weights – W
  - Vector
  - W ε R
  - Dimension – n_X
- Bias – b
  - Scalar
  - b ε R
- $y = W^T X + b$

Sigmoid Function



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

Sigmoid Function

- σ(Z) ~ 1 – For Z >> 0
- σ(Z) ~ 0 – For Z << 0
- σ(Z) = 0.5 – For Z = 0

Z	**σ(Z)**
-2	0.12
-1.5	0.18
-1	0.27
-0.5	0.38
0	0.50
0.5	0.62
1	0.73
1.5	0.82
2	0.88

Logistic Regression

- Input – X
  - Vector
  - X ε R
  - Dimension – n_X
- Output – ŷ
  - Scalar
  - 0 <= ŷ <= 1.0

Linear Regression

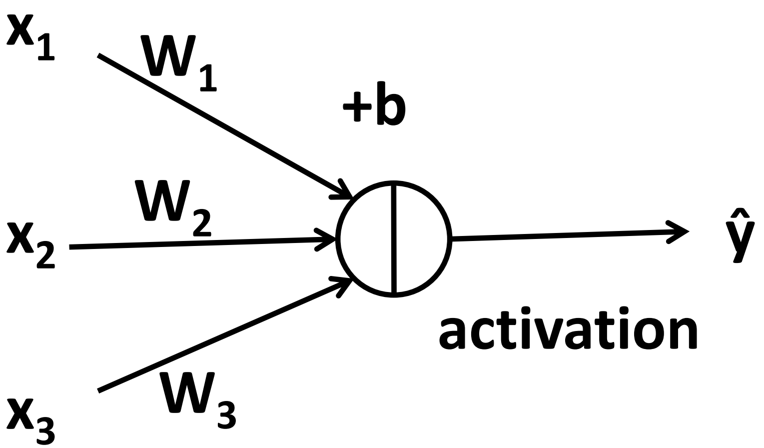
- Weights – W
  - Vector
  - W ε R
  - Dimension – n_X
- Bias – b
  - Scalar
  - b ε R
- $y = W^T X + b$

Logistic Regression

- Weights – W
  - Vector
  - W ε R
  - Dimension – n_X
- Bias – b
  - Scalar
  - b ε R
- ~~$y = W^TX + b$~~

Logistic Regression

- Weights – W
  - Vector
  - W ε R
  - Dimension – n_X
- Bias – b
  - Scalar
  - b ε R
- $Z = W^T X + b$
- $\hat{y} = \sigma(Z)$ – Activation (sigmoid) function

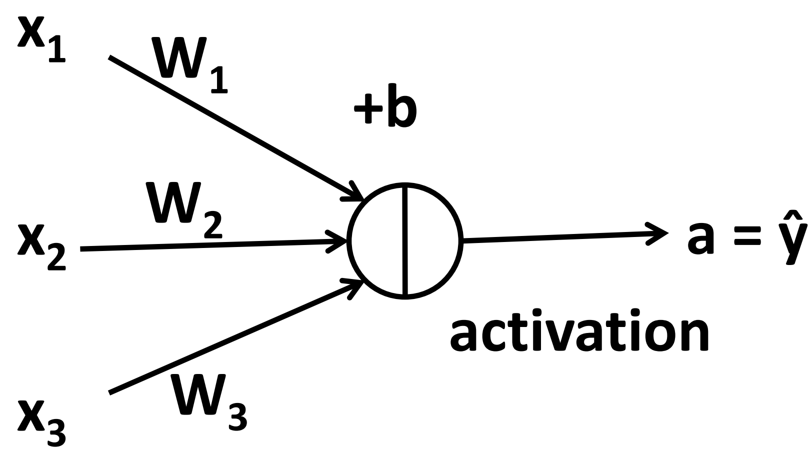


Logistic Regression

- Weights – W
  - Vector
  - W ε R
  - Dimension – n_X
- Bias – b
  - Scalar
  - b ε R
- $Z = W^T X + b$
- $\hat{y} = P(y=1 \mid X)$

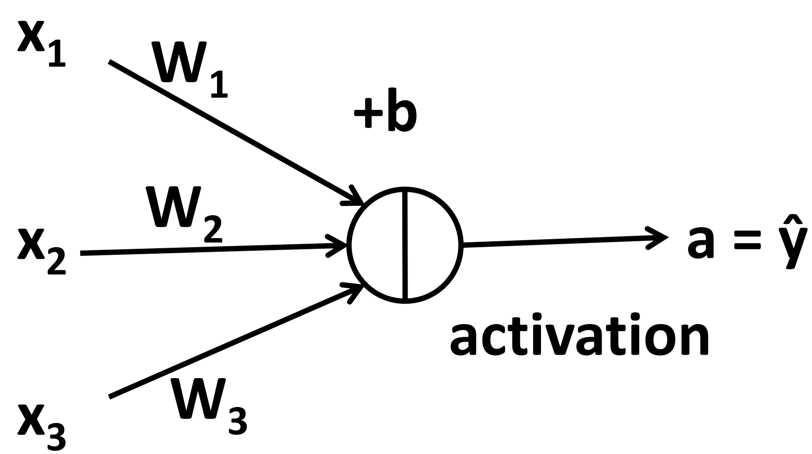
Artificial Neural Network

- $Z = W^T X + b$
  - Linear regression



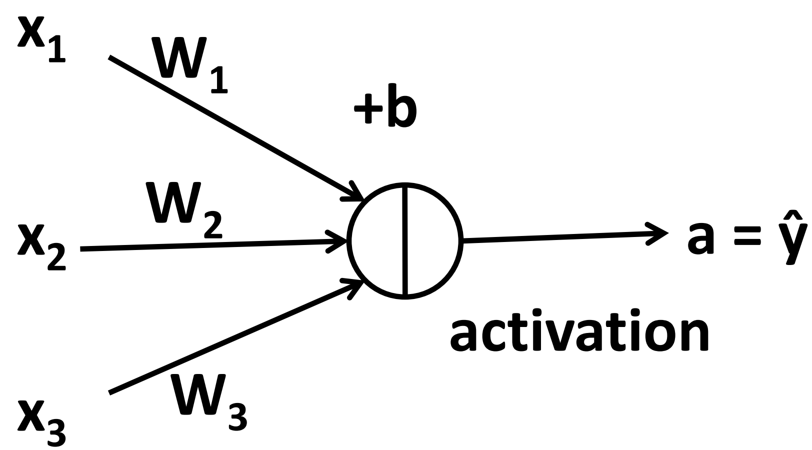
Artificial Neural Network

- $Z = W^TX + b$
  - Linear regression
- $\hat{y} = \sigma(Z) = a$
  - Activation function
  - e.g. sigmoid function



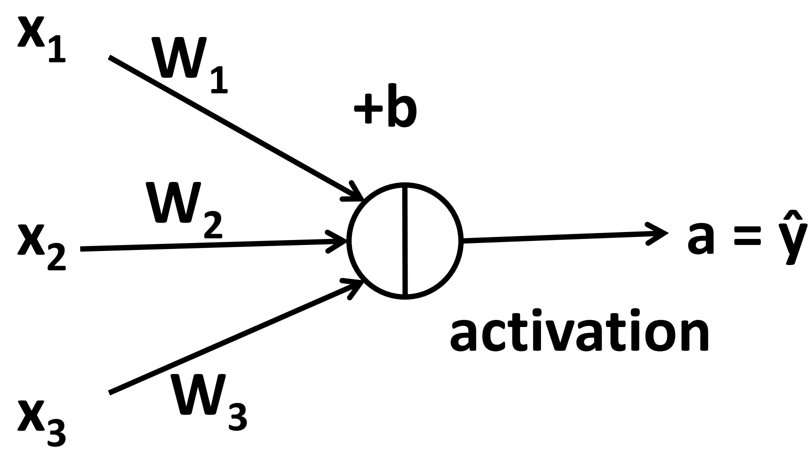
Artificial Neural Network

- $Z = W^TX + b$
  - Linear regression
- $\hat{y} = \sigma(Z) = a$
  - Activation function
  - e.g. sigmoid function
- Loss function
  - $L(\hat{y}^{(i)}, y^{(i)})$ – One i^{th} sample



Artificial Neural Network

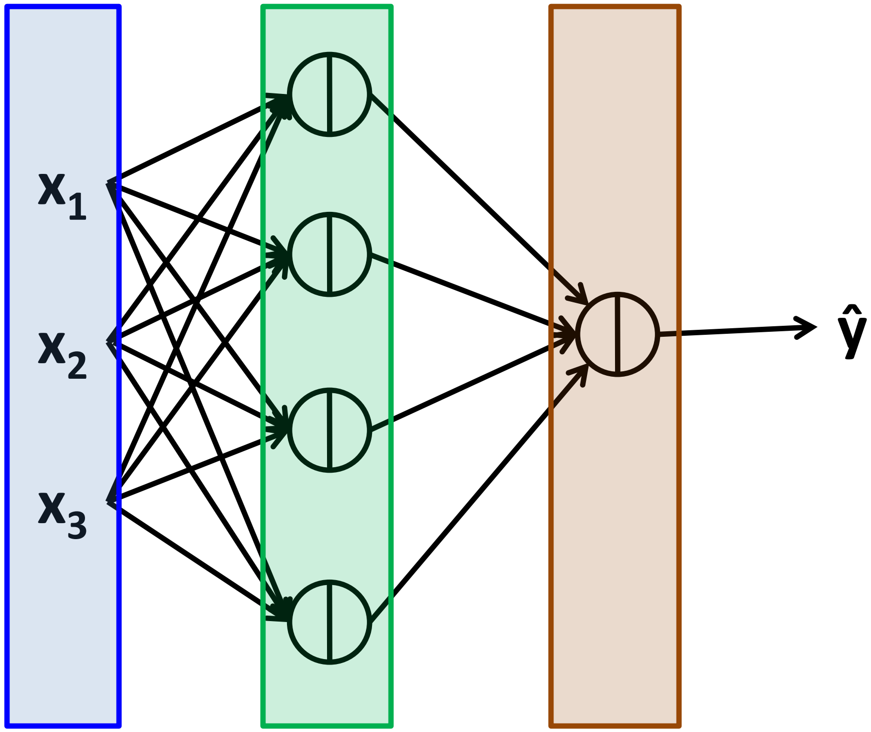
- $Z = W^T X + b$
  - Linear regression
- $\hat{y} = \sigma(Z) = a$
  - Activation function
  - e.g. sigmoid function



- Loss function
  - $L(\hat{y}^{(i)}, y^{(i)})$ – One i^{th} sample
- Cost function
  - $J(W, b)$ – Average of loss function for all samples

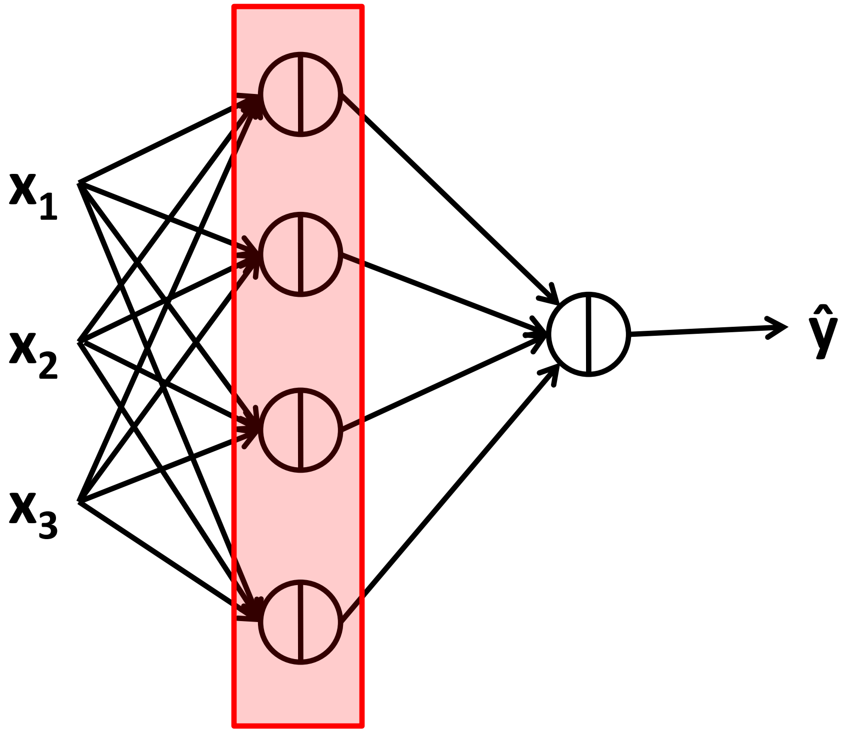
Two Layer Neural Network

- Input layer
- Hidden layer – Layer 1
- Output layer – Layer 2



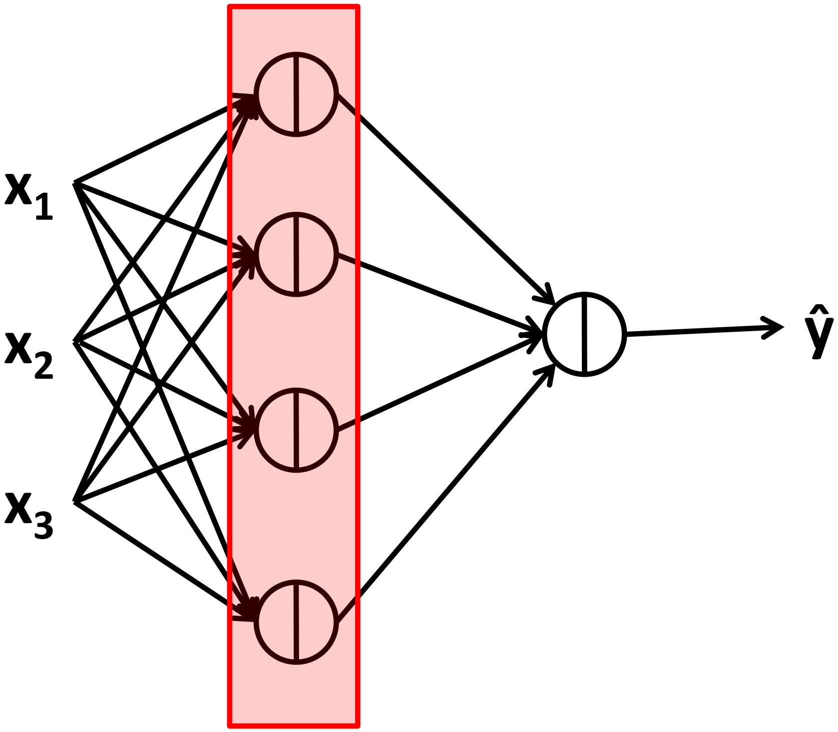
Two Layer Neural Network

- Layer 1
  - $Z^{[1]} = W^{[1]}X + b^{[1]}$



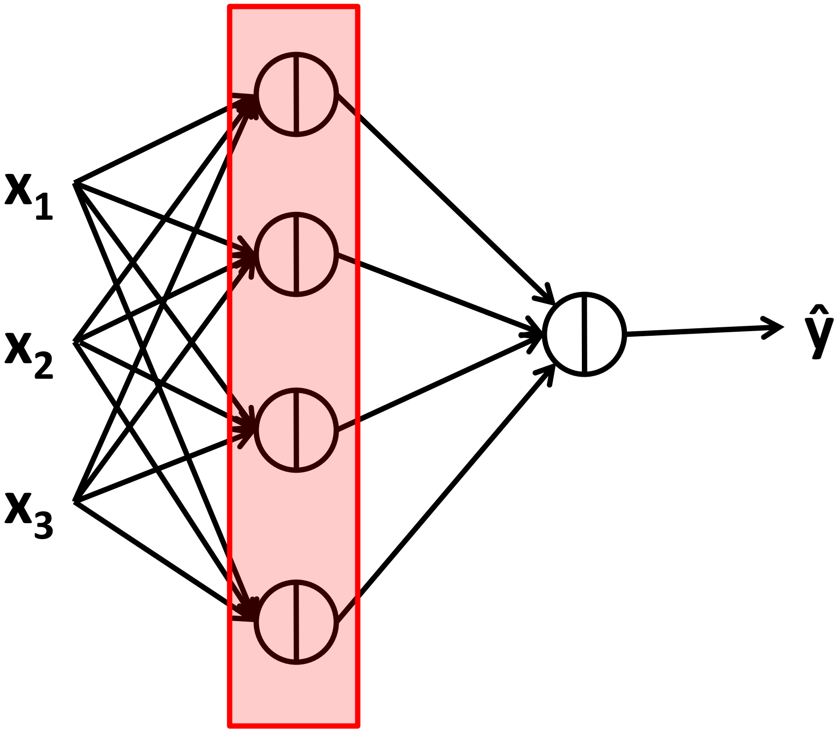
Two Layer Neural Network

- Layer 1
  - $Z^{[1]} = W^{[1]}a^{[0]} + b^{[1]}$



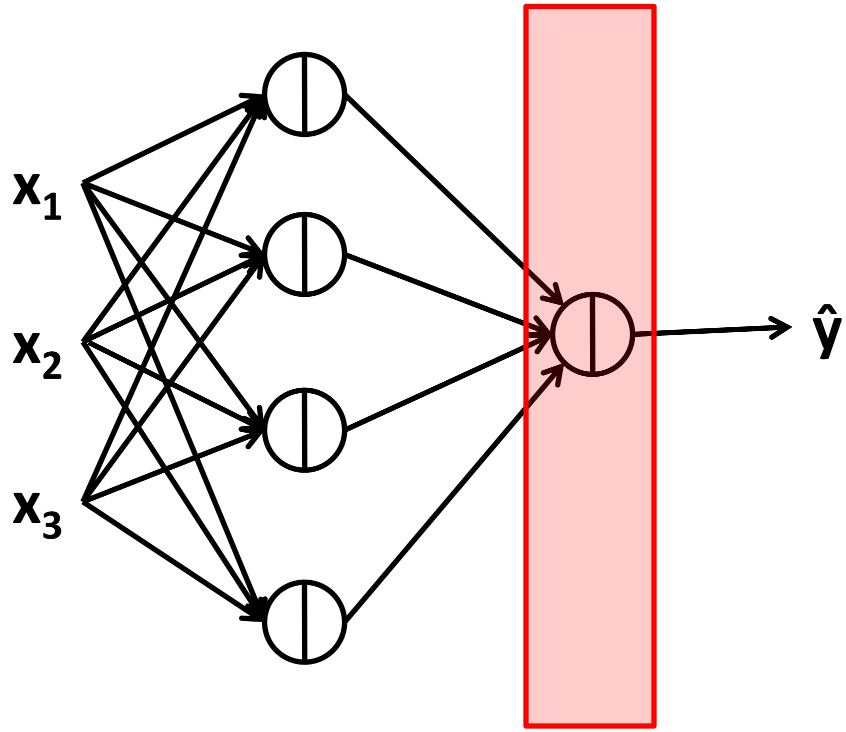
Two Layer Neural Network

- Layer 1
  - $Z^{[1]} = W^{[1]}a^{[0]} + b^{[1]}$
  - $a^{[1]} = \sigma(Z^{[1]})$



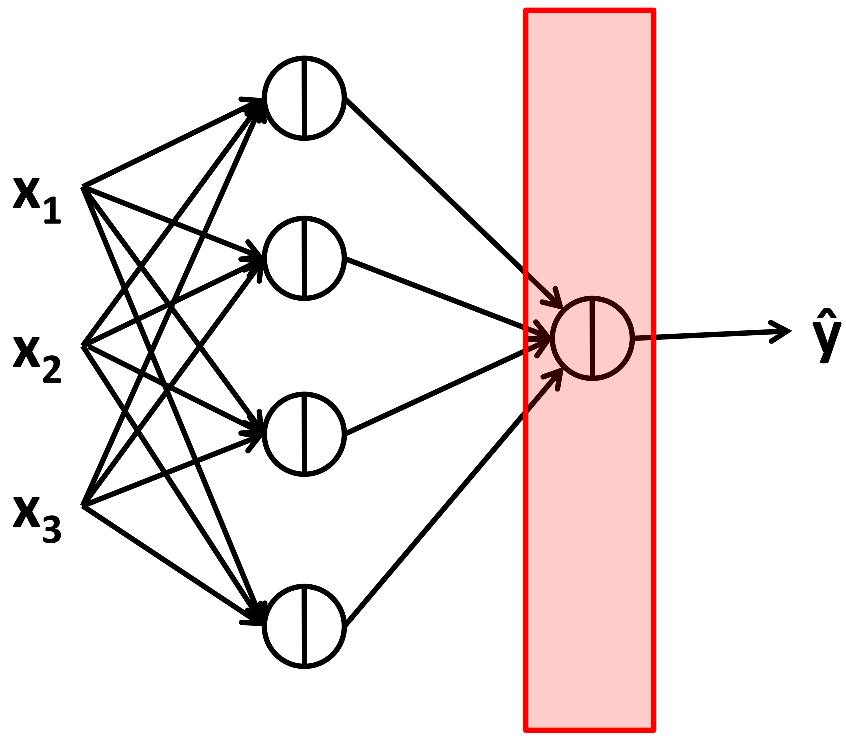
Two Layer Neural Network

- Layer 1
  - $Z^{[1]} = W^{[1]}a^{[0]} + b^{[1]}$
  - $a^{[1]} = \sigma(Z^{[1]})$
- Layer 2
  - $Z^{[2]} = W^{[2]}a^{[1]} + b^{[2]}$



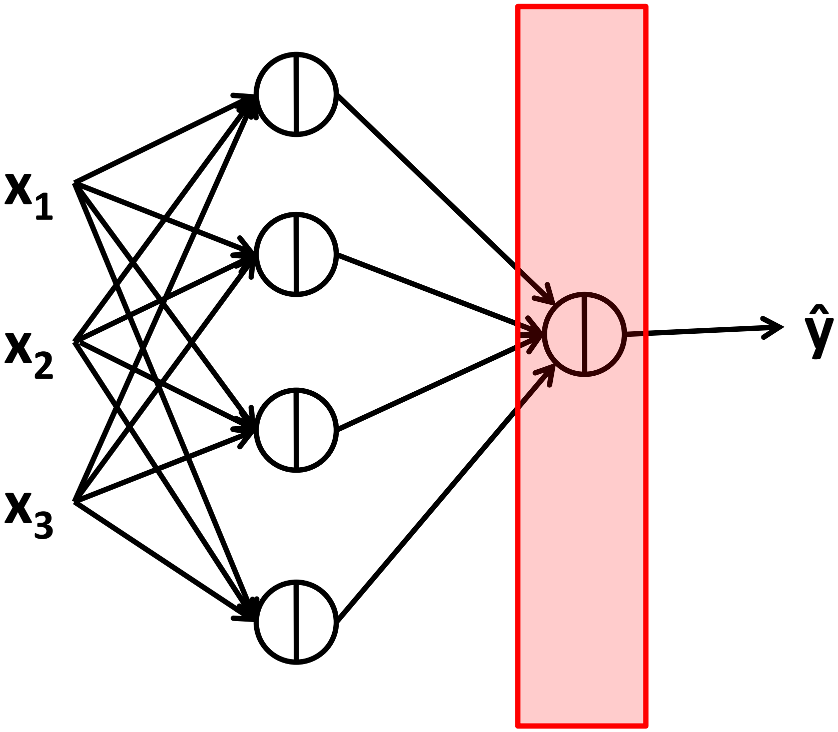
Two Layer Neural Network

- Layer 1
  - $Z^{[1]} = W^{[1]}a^{[0]} + b^{[1]}$
  - $a^{[1]} = \sigma(Z^{[1]})$
- Layer 2
  - $Z^{[2]} = W^{[2]}a^{[1]} + b^{[2]}$
  - $a^{[2]} = \sigma(Z^{[2]})$



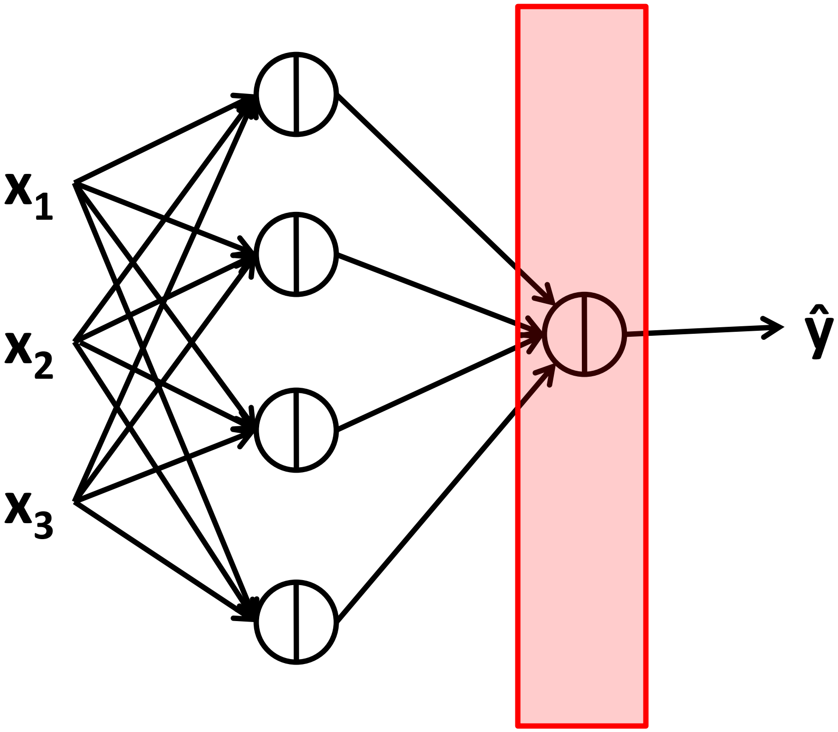
Two Layer Neural Network

- Layer 1
  - $Z^{[1]} = W^{[1]}a^{[0]} + b^{[1]}$
  - $a^{[1]} = \sigma(Z^{[1]})$
- Layer 2
  - $Z^{[2]} = W^{[2]}a^{[1]} + b^{[2]}$
  - $\hat{y} = a^{[2]} = \sigma(Z^{[2]})$
- Loss function
  - $L(\hat{y}, y)$



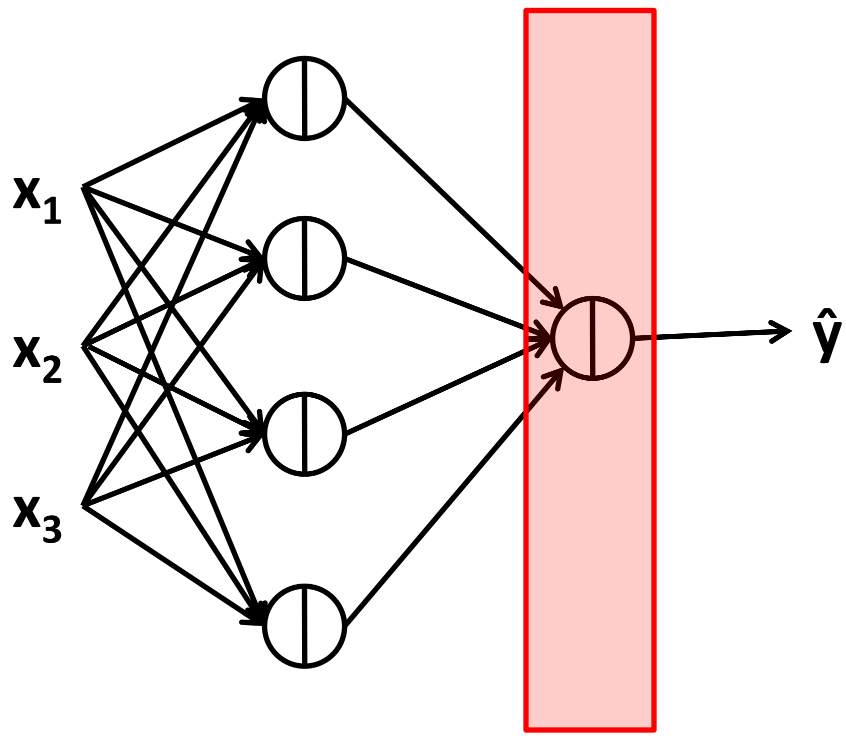
Two Layer Neural Network

- Layer 1
  - $Z^{[1]} = W^{[1]}a^{[0]} + b^{[1]}$
  - $a^{[1]} = \sigma(Z^{[1]})$
- Layer 2
  - $Z^{[2]} = W^{[2]}a^{[1]} + b^{[2]}$
  - $\hat{y} = a^{[2]} = \sigma(Z^{[2]})$
- Loss function
  - $L(\hat{y}^{(i)}, y^{(i)})$



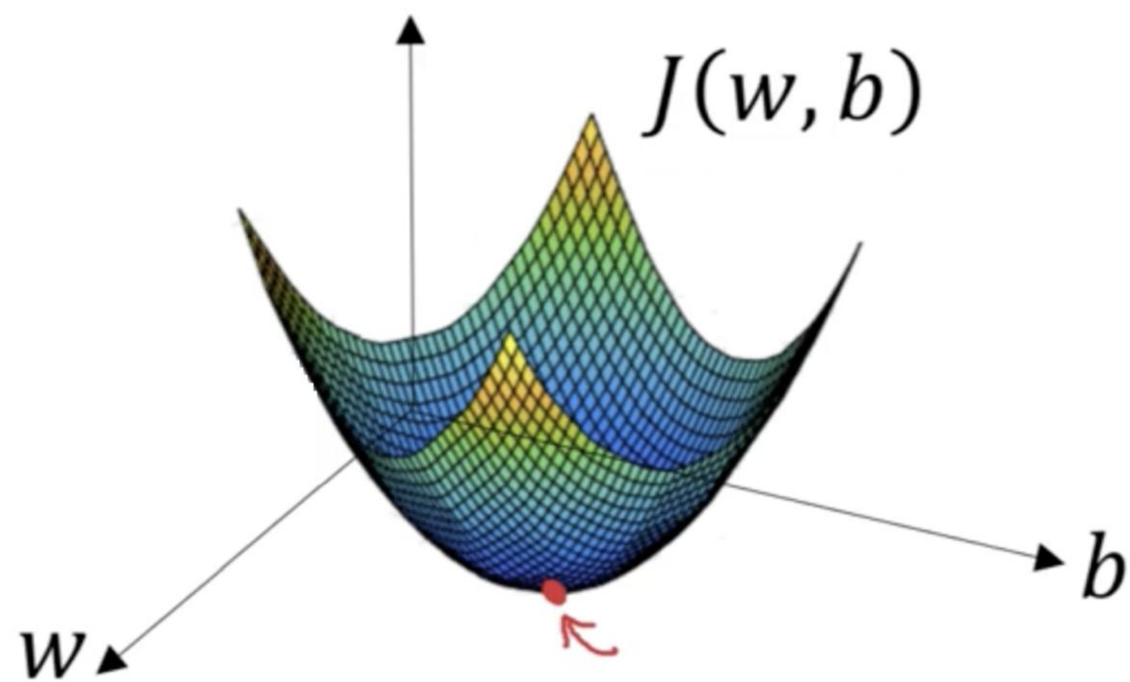
Two Layer Neural Network

- Layer 1
  - $Z^{[1]} = W^{[1]}a^{[0]} + b^{[1]}$
  - $a^{[1]} = \sigma(Z^{[1]})$
- Layer 2
  - $Z^{[2]} = W^{[2]}a^{[1]} + b^{[2]}$
  - $\hat{y} = a^{[2]} = \sigma(Z^{[2]})$
- Cost function
  - $J(W, b)$



Gradient Descent

- Convex function
- Global optimum



Back Propagation

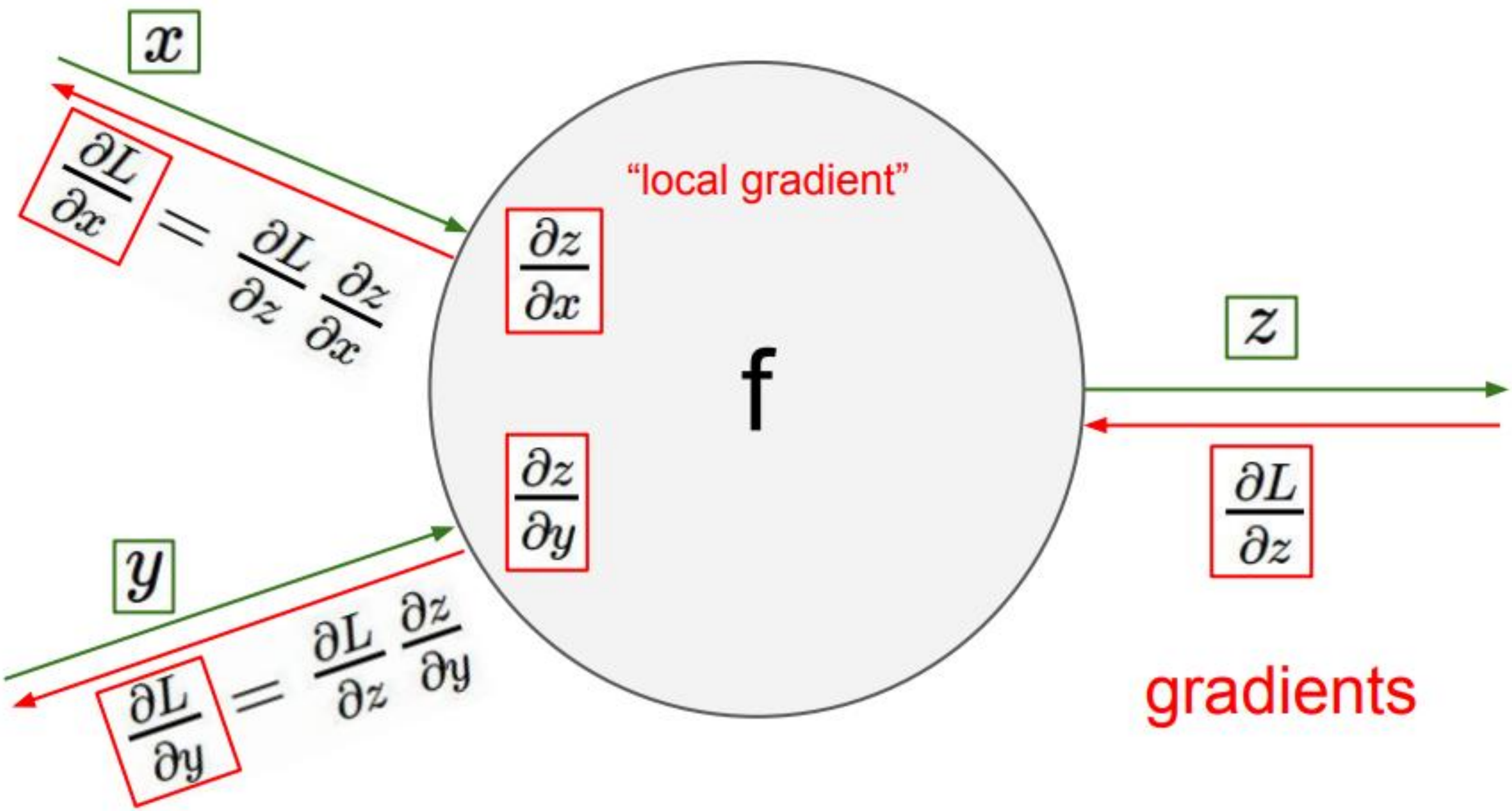


Image ack – Google

Back Propagation

- Parameters
  - $W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]}$
- Cost function
  - $J(W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]})$
- Forward pass
  - $Z^{[1]} = W^{[1]}a^{[0]} + b^{[1]}$
  - $a^{[1]} = \sigma(Z^{[1]})$

Back Propagation

- Parameters
  - $W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]}$
- Cost function
  - $J(W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]})$
- Forward pass
  - $Z^{[1]} = W^{[1]} a^{[0]} + b^{[1]}$
  - $a^{[1]} = \sigma(Z^{[1]})$
  - $Z^{[2]} = W^{[2]} a^{[1]} + b^{[2]}$
  - $\hat{y} = a^{[2]} = \sigma(Z^{[2]})$

Back Propagation

- Parameters
  - $W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]}$
- Cost function
  - $J(W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]})$
- Forward pass
  - $Z^{[1]} = W^{[1]}a^{[0]} + b^{[1]}$
  - $a^{[1]} = \sigma(Z^{[1]})$
  - $Z^{[2]} = W^{[2]}a^{[1]} + b^{[2]}$
  - $\hat{y} = a^{[2]} = \sigma(Z^{[2]})$
  - $L(\hat{y}^{(i)}, y^{(i)})$
  - $J(W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]})$

Back Propagation

- Parameters
  - $W^{[1]}$, $b^{[1]}$, $W^{[2]}$, $b^{[2]}$
- Cost function
  - $J(W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]})$
- Backward pass

$$dW^{[2]} = \frac{\partial J}{\partial W^{[2]}}, db^{[2]} = \frac{\partial J}{\partial b^{[2]}}$$

Back Propagation

- Parameters
  - $W^{[1]}$, $b^{[1]}$, $W^{[2]}$, $b^{[2]}$
- Cost function
  - $J(W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]})$
- Backward pass

$$dW^{[2]} = \frac{\partial J}{\partial W^{[2]}}, db^{[2]} = \frac{\partial J}{\partial b^{[2]}}$$

$$dW^{[1]} = \frac{\partial J}{\partial W^{[1]}}, db^{[1]} = \frac{\partial J}{\partial b^{[1]}}$$

Back Propagation

- Parameters
  - $W^{[1]}$, $b^{[1]}$, $W^{[2]}$, $b^{[2]}$
- Cost function
  - $J(W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]})$

- Backward pass

$$dW^{[2]} = \frac{\partial J}{\partial W^{[2]}}, db^{[2]} = \frac{\partial J}{\partial b^{[2]}}$$

$$dW^{[1]} = \frac{\partial J}{\partial W^{[1]}}, db^{[1]} = \frac{\partial J}{\partial b^{[1]}}$$

$$W^{[2]} = W^{[2]} - \alpha * dW^{[2]}$$

$$b^{[2]} = b^{[2]} - \alpha * db^{[2]}$$

Back Propagation

- Parameters
  - $W^{[1]}$, $b^{[1]}$, $W^{[2]}$, $b^{[2]}$
- Cost function
  - $J(W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]})$

- Backward pass

$$dW^{[2]} = \frac{\partial J}{\partial W^{[2]}}, db^{[2]} = \frac{\partial J}{\partial b^{[2]}}$$

$$dW^{[1]} = \frac{\partial J}{\partial W^{[1]}}, db^{[1]} = \frac{\partial J}{\partial b^{[1]}}$$

$$W^{[2]} = W^{[2]} - \alpha * dW^{[2]}$$

$$b^{[2]} = b^{[2]} - \alpha * db^{[2]}$$

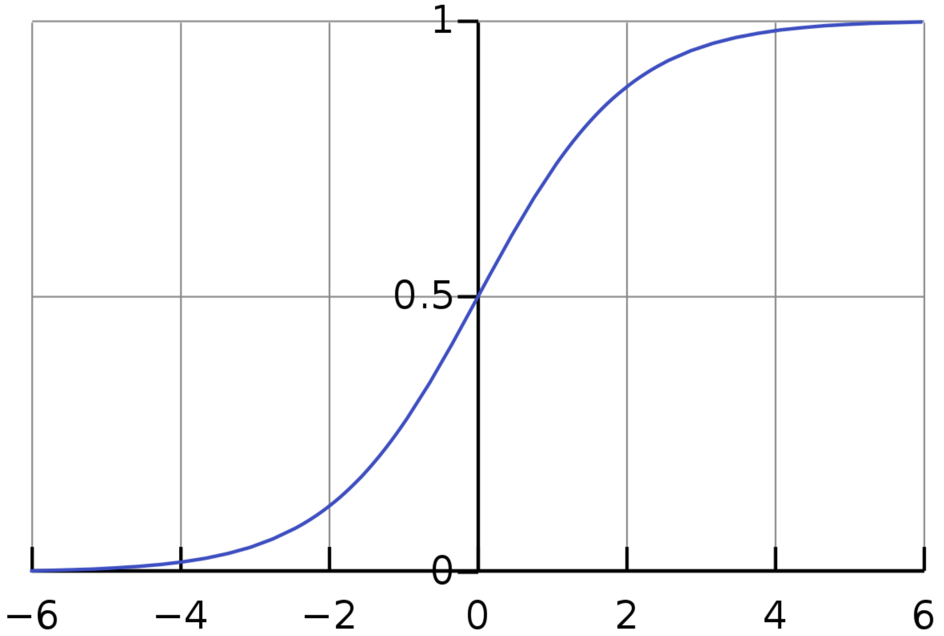
$$W^{[1]} = W^{[1]} - \alpha * dW^{[1]}$$

$$b^{[1]} = b^{[1]} - \alpha * db^{[1]}$$

Activation Functions

- Sigmoid activation
- tanh activation
- ReLU – Rectified Linear Units
- Leaky ReLU

Sigmoid Activation



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

Sigmoid Activation

- $0 <= \hat{y} <= 1.0$
- Binary classification

Sigmoid Activation

$$s(z) = \frac{1}{1+e^{-z}} = (1+e^{-z})^{-1}$$

$$\frac{ds}{dz} = -1(1+e^{-z})^{-2}\frac{d}{dz}(1+e^{-z})$$

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

$$= \frac{e^{-z}}{(1+e^{-z})^2}$$

$$s(z)\cdot(1-s(z))$$

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

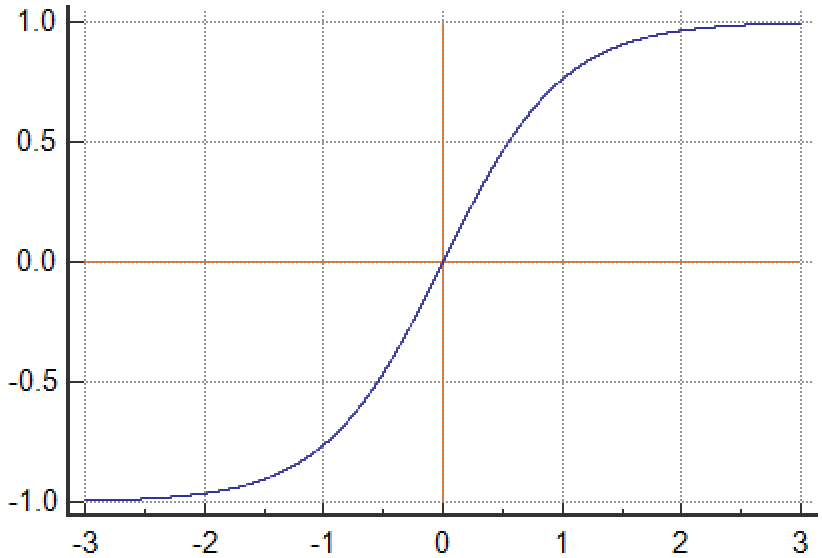
$$= \left(\frac{1}{1+e^{-z}}\right) - \left(\frac{1}{1+e^{-z}}\right)^2$$

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

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

$$= \frac{e^{-z}}{(1+e^{-z})^2}$$

tanh Activation



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

tanh Activation

- tanh(Z) ~ 1 – For Z >> 0
- tanh(Z) ~ -1 – For Z << 0
- tanh(Z) = 0 – For Z = 0

tanh Activation

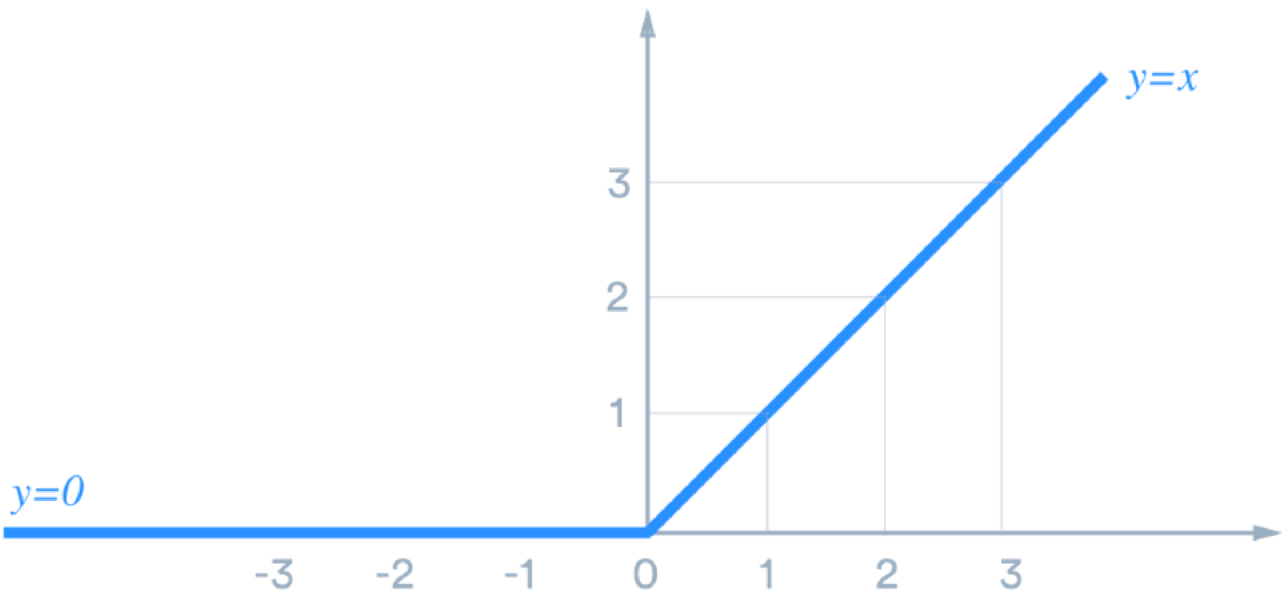
- Zero mean
- Range – -1.0 to +1.0
- Scaled and zero mean Sigmoid function
- Better than Sigmoid activation function
- Neural network
- Recurrent neural network

tanh Activation

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

$$\frac{d}{dZ}\tanh(Z) = 1 - \tanh^{2}(\mathrm{Z})$$

ReLU Activation



$$\mathrm{Re}\,LU(Z) = \max(0,\ Z)$$

ReLU Activation

- ReLU(Z) ~ Z – For Z > 0
- ReLU(Z) ~ 0 – For Z < 0
- ReLU(Z) = ? – For Z = 0

ReLU Activation

- Neural network
- Convolutional neural network

ReLU Activation

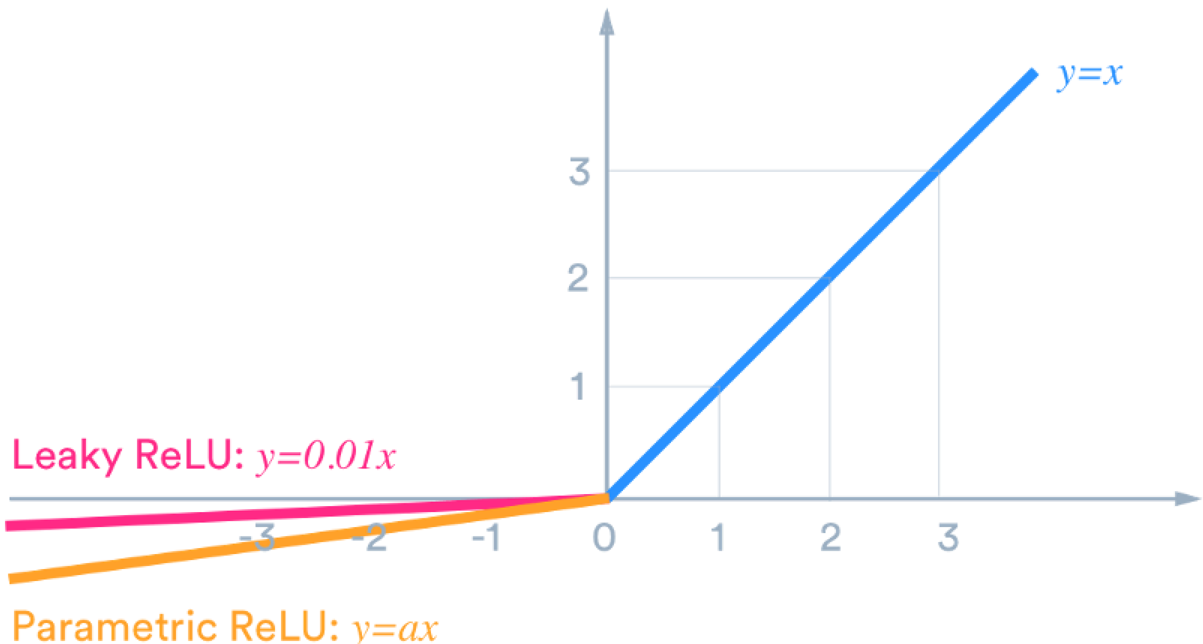
$$\mathrm{ReLU}(Z) = \max(0,\ Z)$$

$$\frac{d}{dZ}\mathrm{ReLU}(Z) = 1 \quad Z > 0$$

$$\frac{d}{dZ}\mathrm{ReLU}(Z) = 0 \quad Z < 0$$

$$\frac{d}{dZ}\mathrm{ReLU}(Z) = ? \quad Z = 0$$

Leaky ReLU Activation



$$\text{Leaky ReLU}(Z) = \max(0,\ 0.01 * Z)$$

$$\text{Parametric ReLU}(Z) = \max(0,\ a * Z)$$

Leaky ReLU Activation

- Leaky ReLU(Z) ~ Z – For Z > 0
- Leaky ReLU(Z) ~ 0.01 * Z – For Z < 0
- Leaky ReLU(Z) = ? – For Z = 0

Leaky ReLU Activation

- Neural network
- Convolutional neural network

Leaky ReLU Activation

$$LeakyReLU(Z) = \max(0,\ 0.01 * Z) = g(Z)$$

$$\frac{d}{dZ} g(Z) = 1 \quad Z > 0$$

$$\frac{d}{dZ} g(Z) = 0.01 * Z \quad Z < 0$$

$$\frac{d}{dZ} g(Z) = ? \quad Z = 0$$

Questions?

Thank you