Introduction to deep learning

APSC 8280: Machine learning applied to plant science

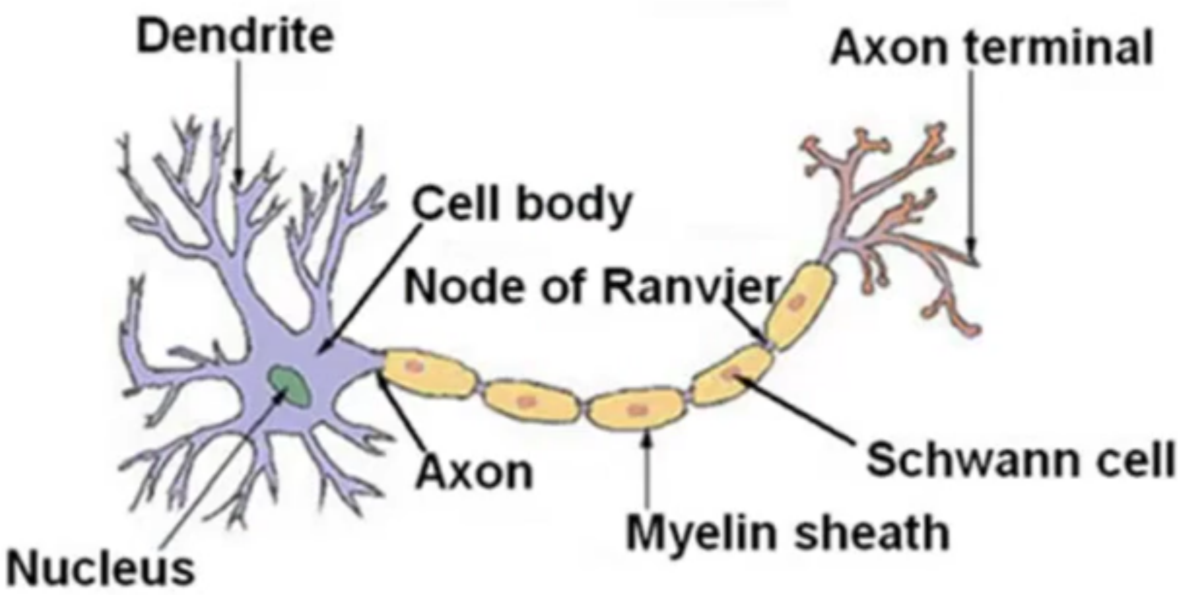
04/16/2020

Outline

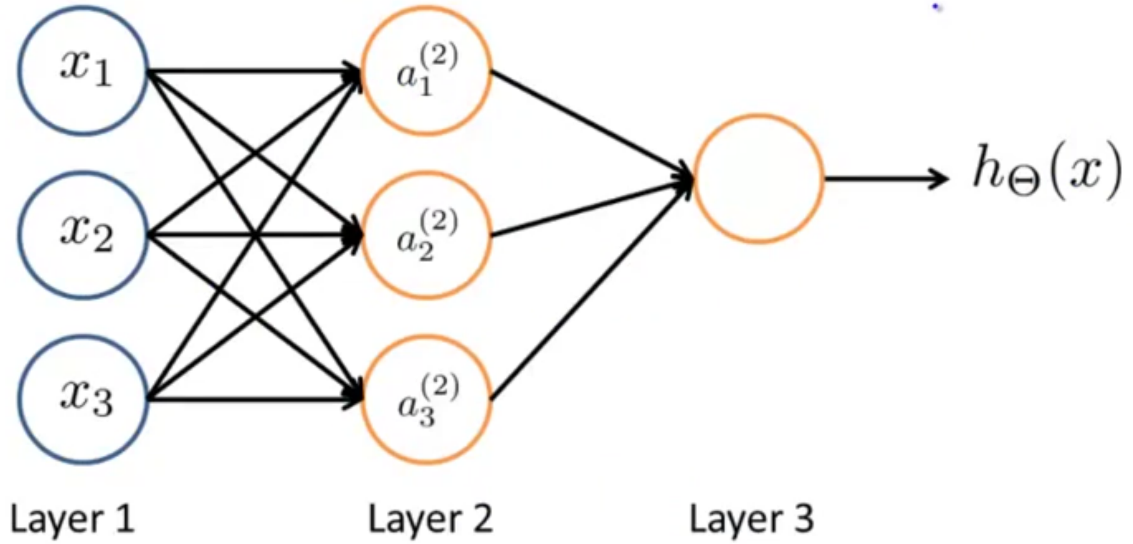
- Introduction to neural networks
- Types of neural network
- Hand-worked example
- Demo

Introduction to neural networks(NN)

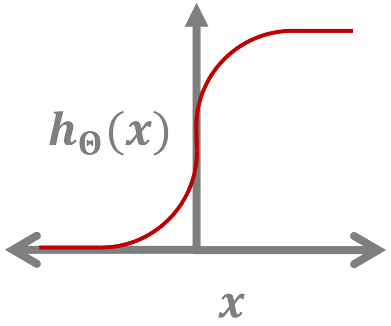
Neurons in the brain



Neural network
Neural model: logistic unit



Activation function



Adapted from Andrew Ng slides

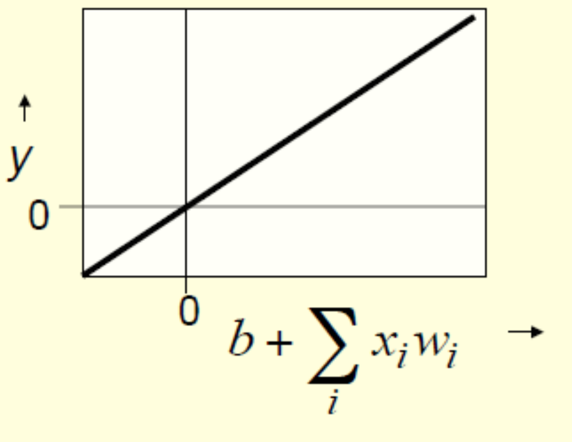
NN types: activation functions

Linear neurons

bias
i^{th} input

$$y = b + \sum_i x_i w_i$$

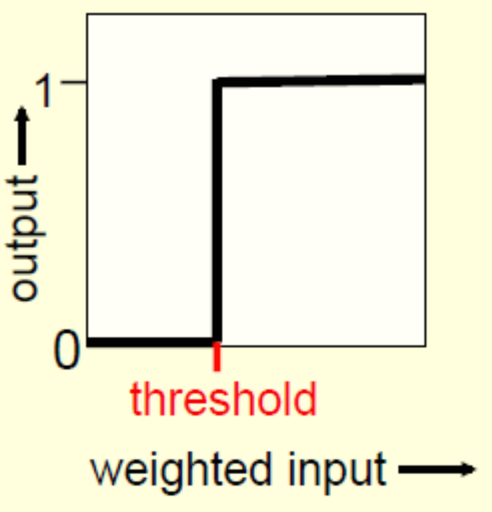
output
index over input connections
weight on i^{th} input



Binary threshold neurons

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

$$y = \begin{cases} 1 \text{ if } z \geq 0 \\ 0 \text{ otherwise} \end{cases}$$

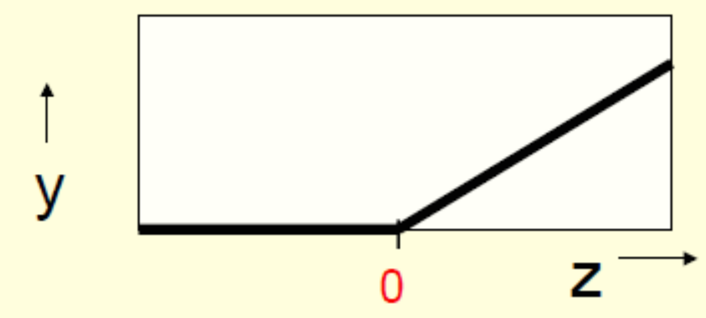


NN types: activation functions

Rectified linear neurons

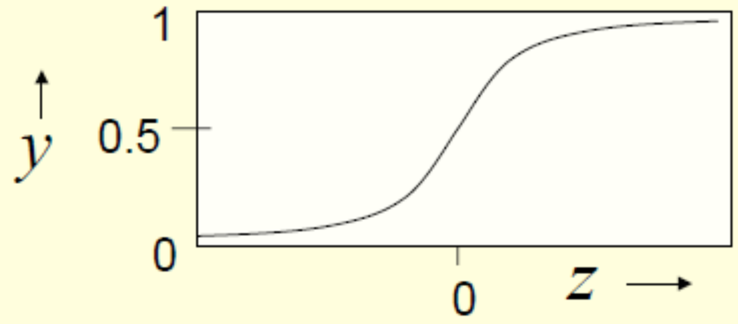
$$z = b + \sum_i x_i w_i$$

$$y = \begin{cases} z & \text{if } z > 0 \\ 0 & \text{otherwise} \end{cases}$$



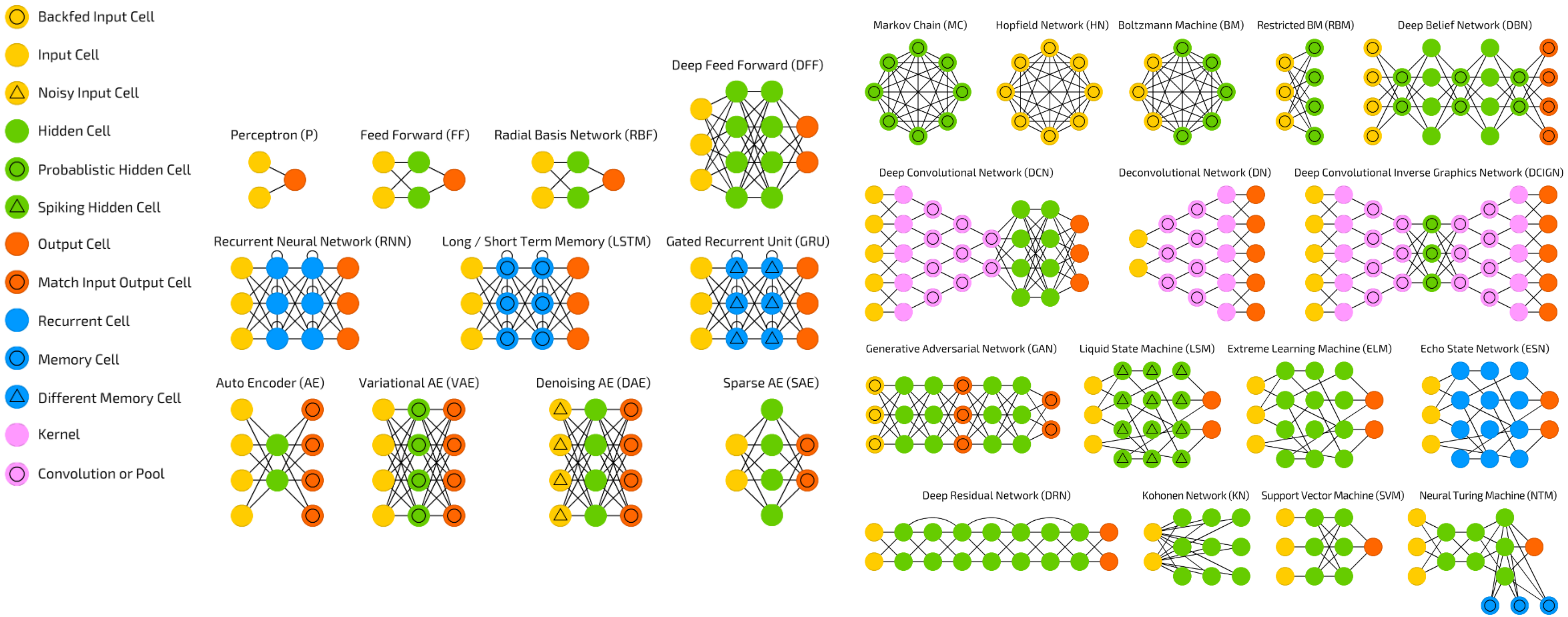
Sigmoid neurons

$$z = b + \sum_i x_i w_i \qquad y = \frac{1}{1+e^{-z}}$$



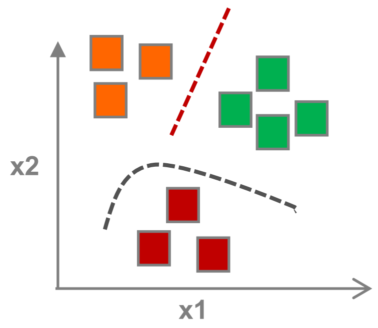
Adapted from Geoffrey Hinton slides

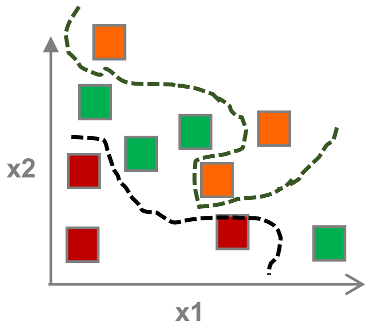
Neural network architectures

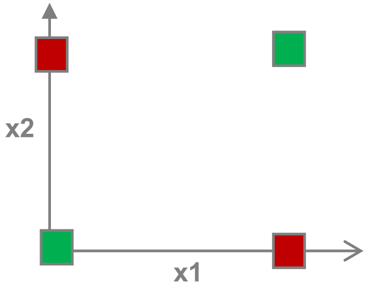


NN : simple demonstrations

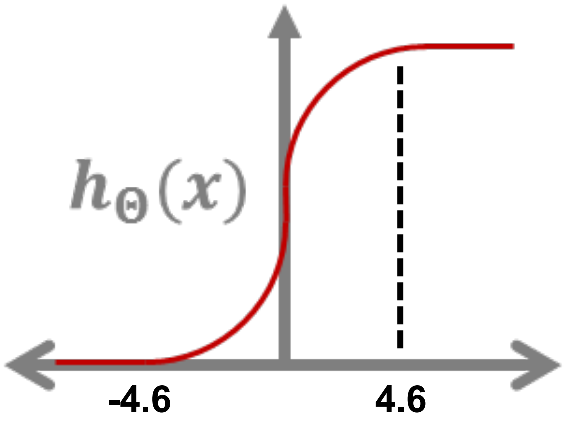
Classification: drawing decision boundaries

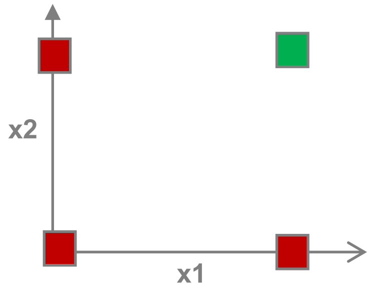




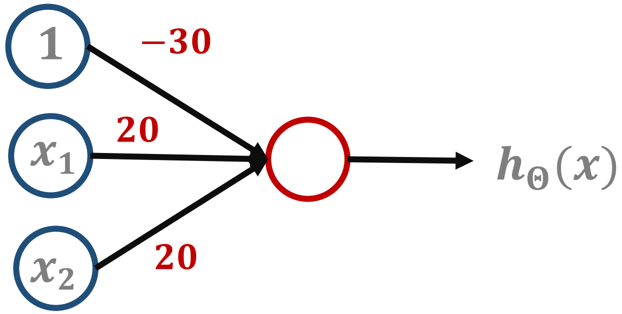


Activation function





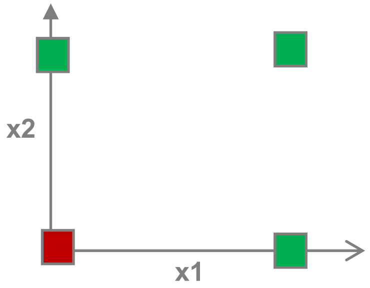
x_1 AND x_2

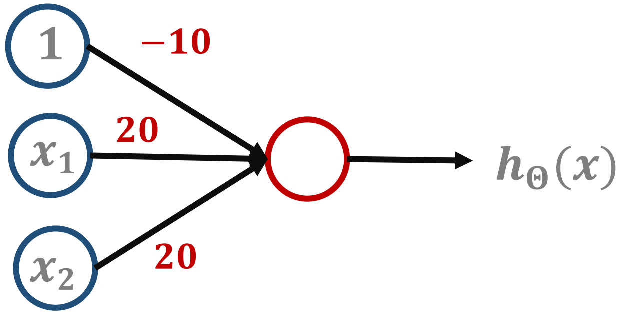


x_1	x_2	$h_\Theta(x)$
0	**0**	$h_\Theta\left(-30 + 20*(0) + 20*(0)\right) = h_\Theta(-30) \approx \mathbf{0}$
0	**1**	$h_\Theta\left(-30 + 20*(0) + 20*(1)\right) = h_\Theta(-10) \approx \mathbf{0}$
1	**0**	$h_\Theta\left(-30 + 20*(1) + 20*(0)\right) = h_\Theta(-10) \approx \mathbf{0}$
1	**1**	$h_\Theta\left(-30 + 20*(1) + 20*(1)\right) = h_\Theta(10) \approx \mathbf{1}$

Try it yourself!

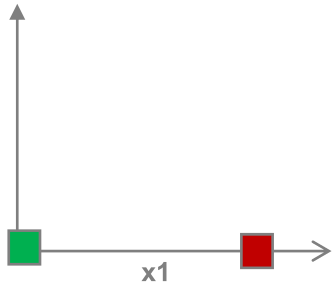
x_1 OR x_2

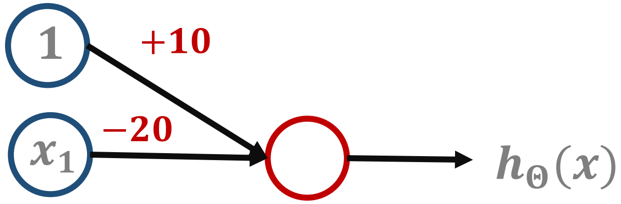




x_1	x_2	$h_\Theta(x)$
0	0	$h_\Theta\big(-10 + 20*(0) + 20*(0)\big) = h_\Theta(-10) \approx \mathbf{0}$
0	1	$h_\Theta\big(-10 + 20*(0) + 20*(1)\big) = h_\Theta(10) \approx \mathbf{1}$
1	0	$h_\Theta\big(-10 + 20*(1) + 20*(0)\big) = h_\Theta(10) \approx \mathbf{1}$
1	1	$h_\Theta\big(-10 + 20*(1) + 20*(1)\big) = h_\Theta(30) \approx \mathbf{1}$

NOT x_1

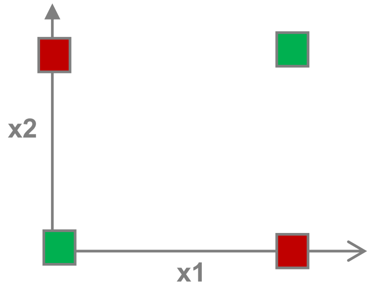




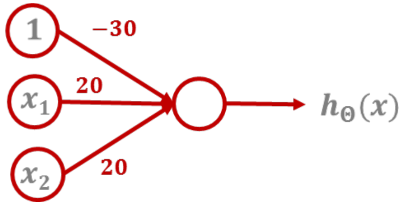
x_1	$h_\Theta(x)$
0	$h_\Theta\big(10 - 20*(0)\big) = h_\Theta(10) \approx \mathbf{1}$
1	$h_\Theta\big(10 - 20*(1)\big) = h_\Theta(-10) \approx \mathbf{0}$

Putting it all together

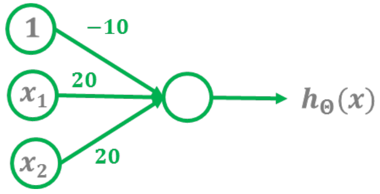
x_1 XNOR x_2



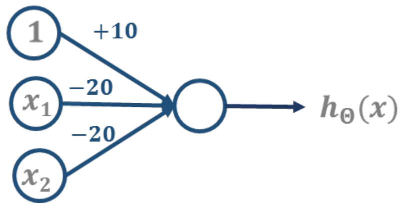
x_1 AND x_2

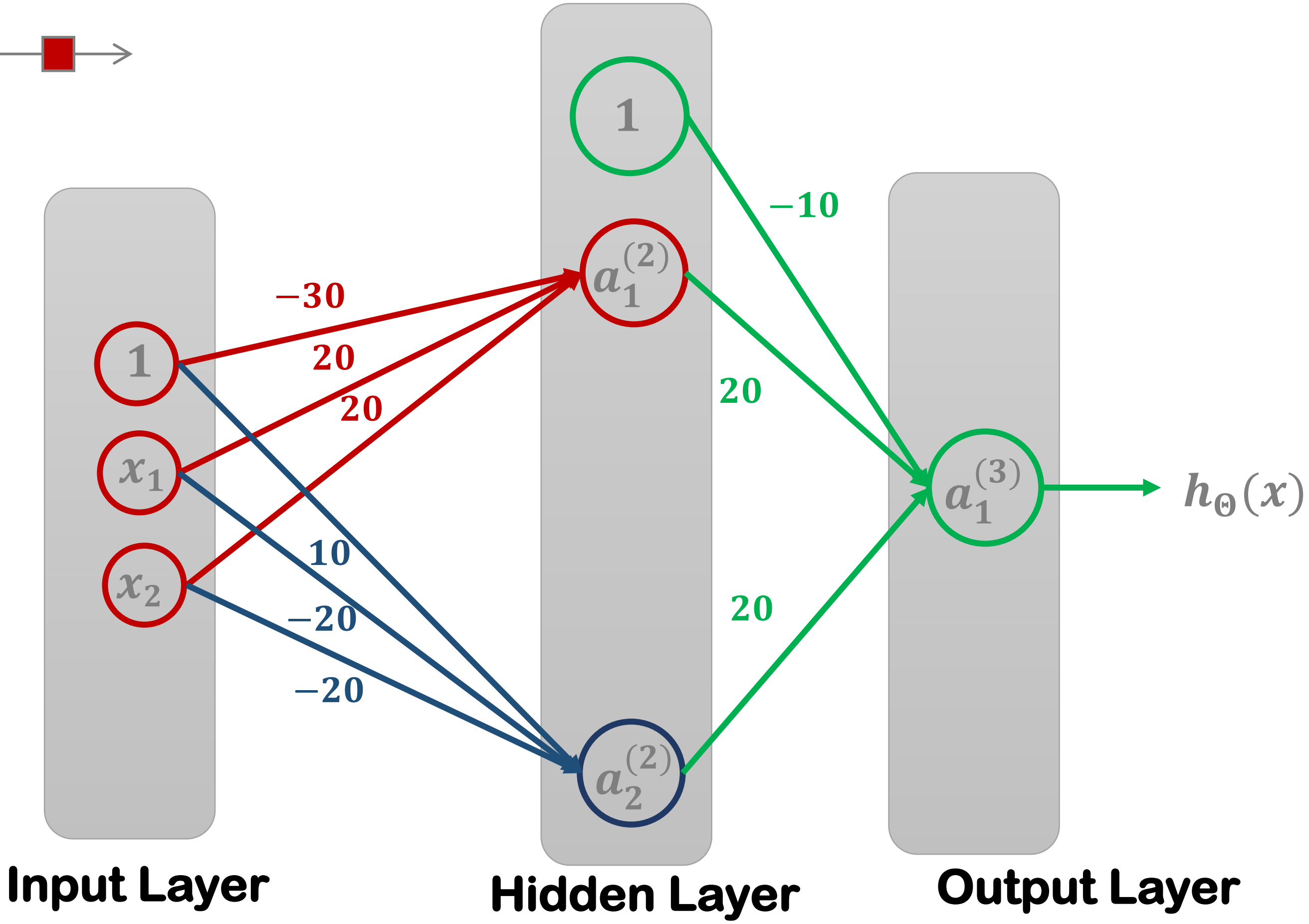


x_1 OR x_2



NOT (x_1) AND NOT(x2)





x_1	x_2	$a_1^{(2)}$	$a_2^{(2)}$	$h_\Theta(x)$
0	0	0	1	1
0	1	0	0	0
1	0	0	0	0
1	1	1	0	1

Training a neural network

- Pick a network architecture
- Randomly initialize weights
- Forward propagation to get the activations
- Compute the cost function
- Backpropagation to compute partial derivatives

Neural networks

Strengths	Weaknesses
✓ Perhaps the most effective method for modeling complex patterns	X Can be extremely computationally intensive to train
✓ Makes few assumptions about the data	X Prone to overfitting
	X Black box model with little interpretability