CYBER 207
Applied Machine Learning for Cybersecurity

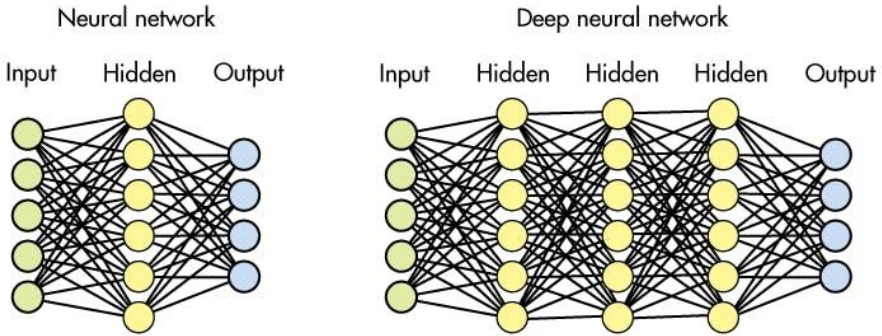
Week 6
Live Session Slides

Neural Networks

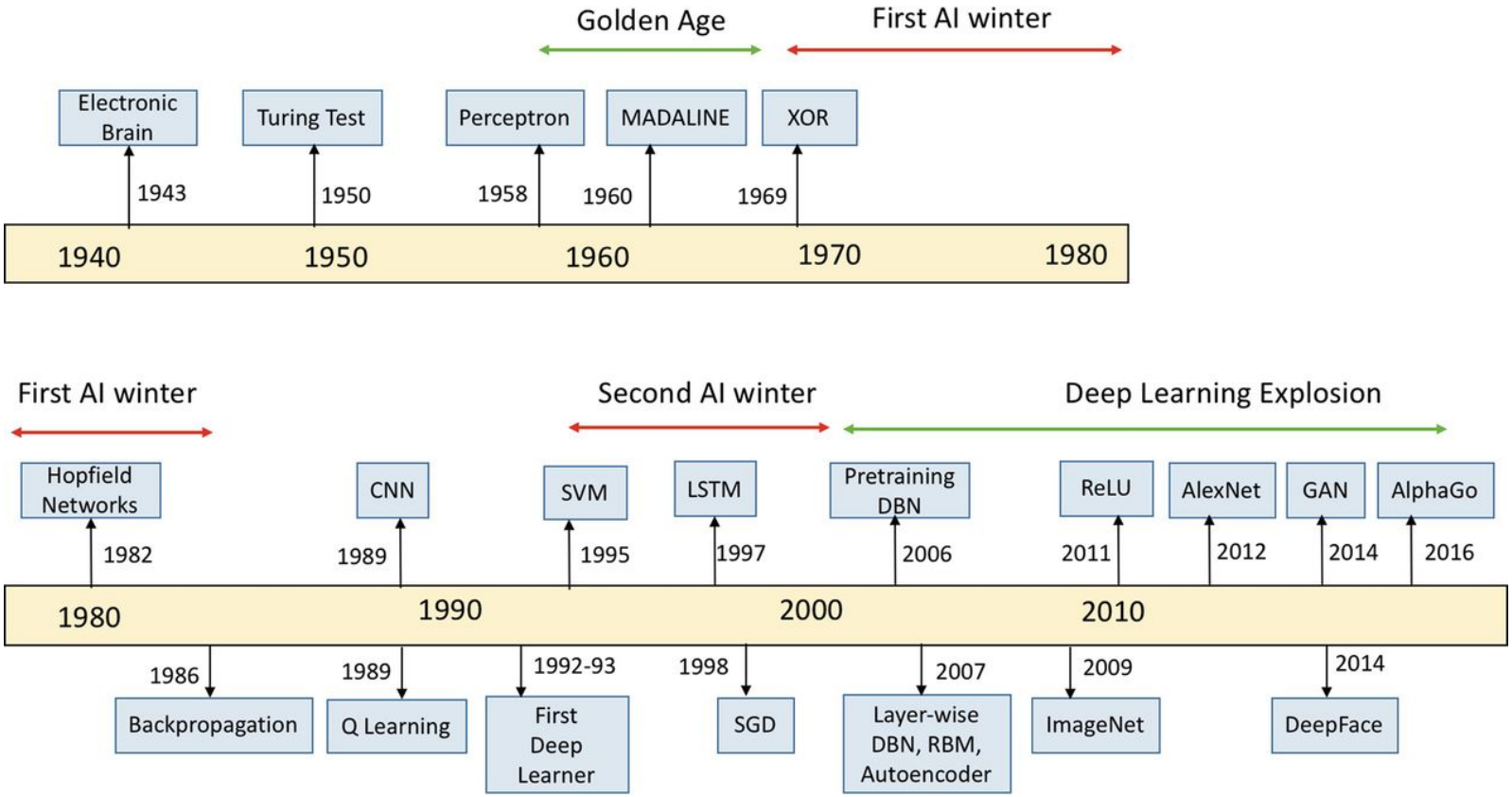
Deep learning algorithm structured similar to the organization of neurons in the brain

A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates.

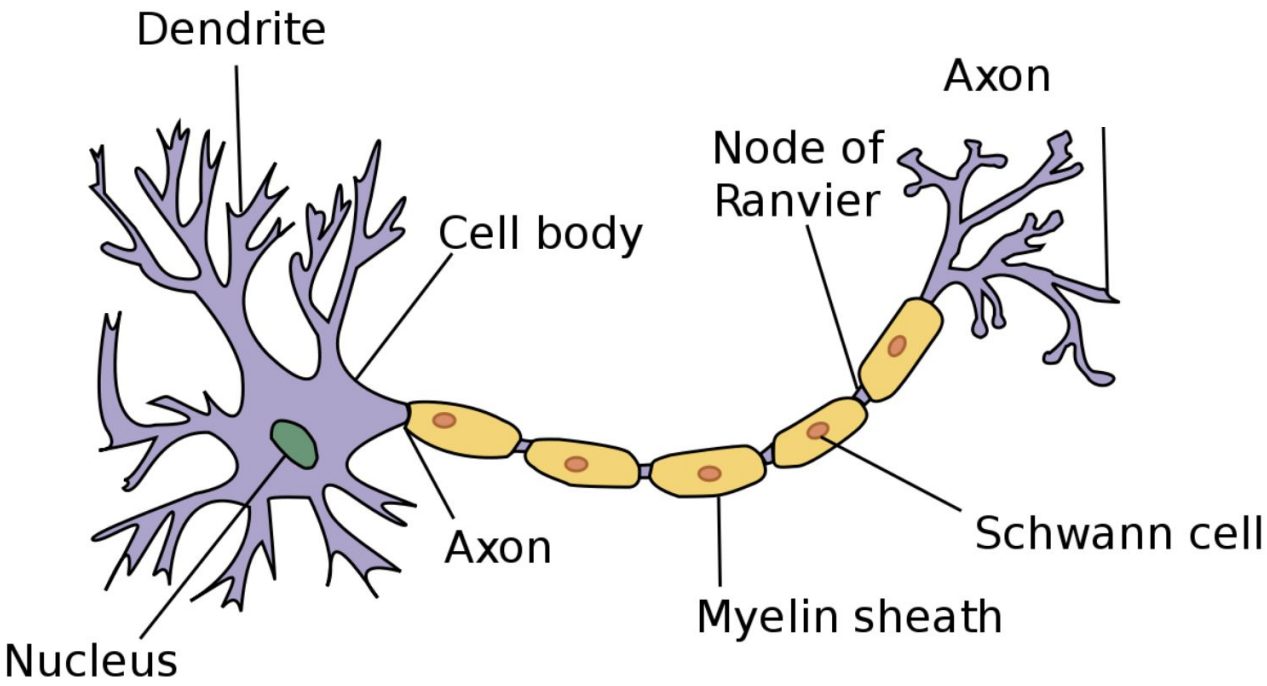
Neural networks can adapt to changing input; so the network generates the best possible result without needing to redesign the output criteria.



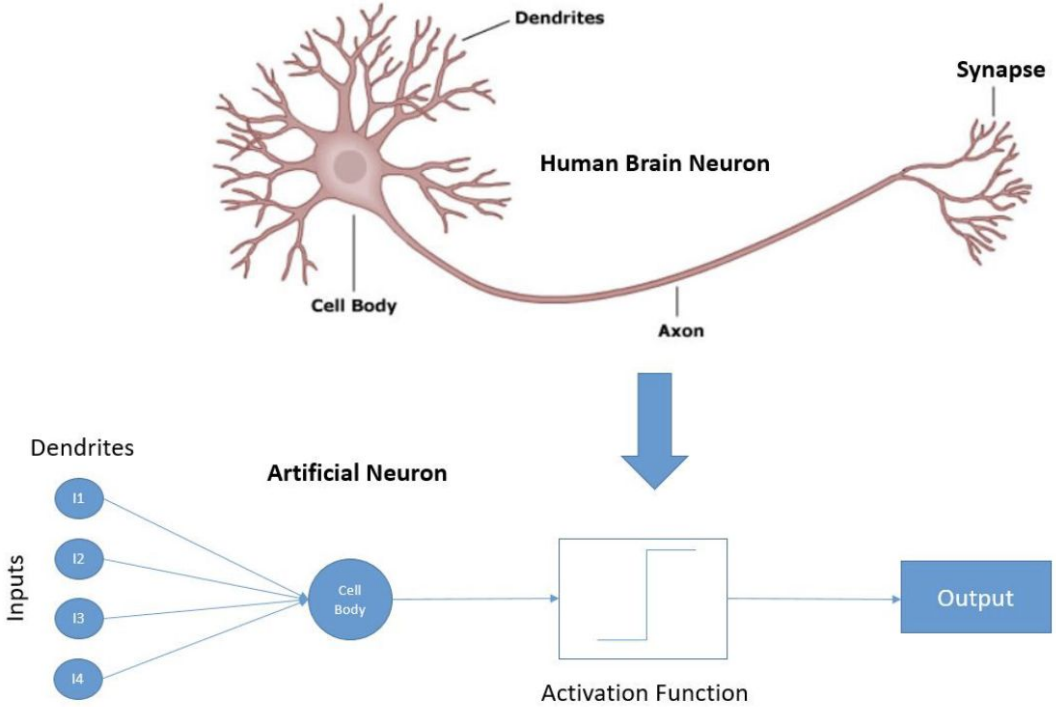
Neural Networks: History and Timeline



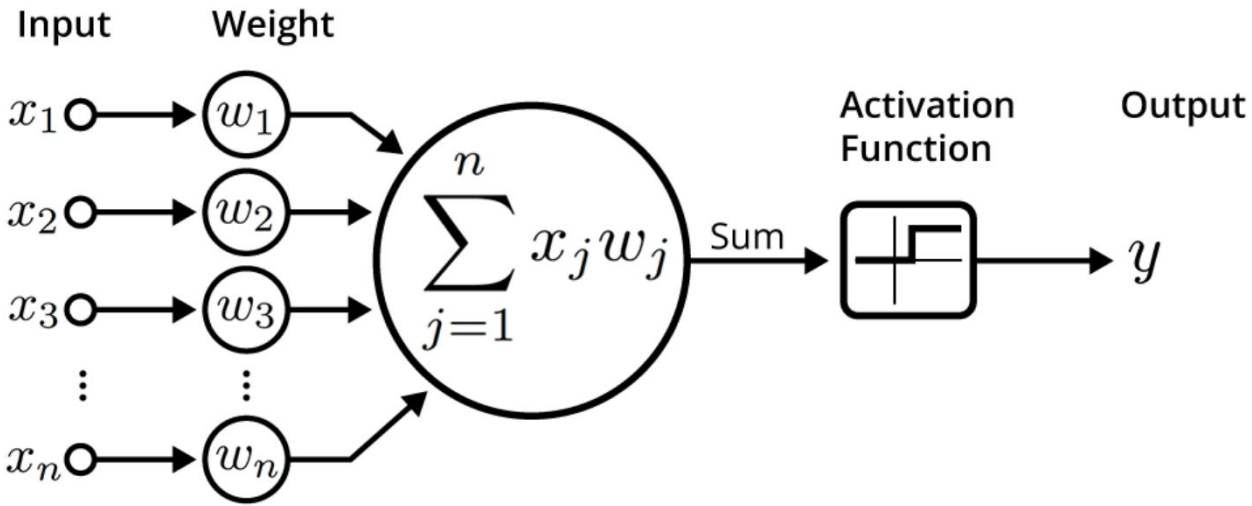
Neural Networks



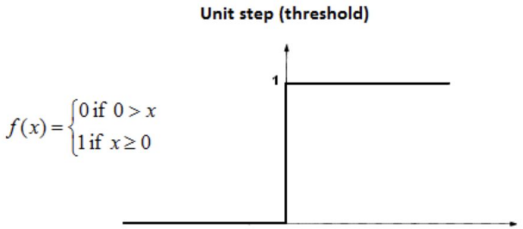
Neural Networks

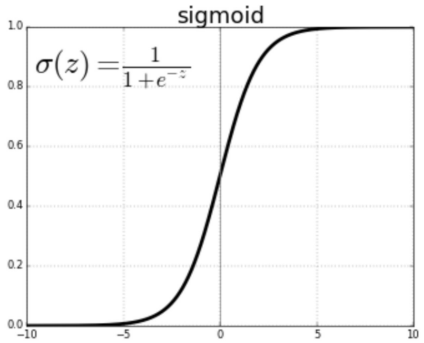


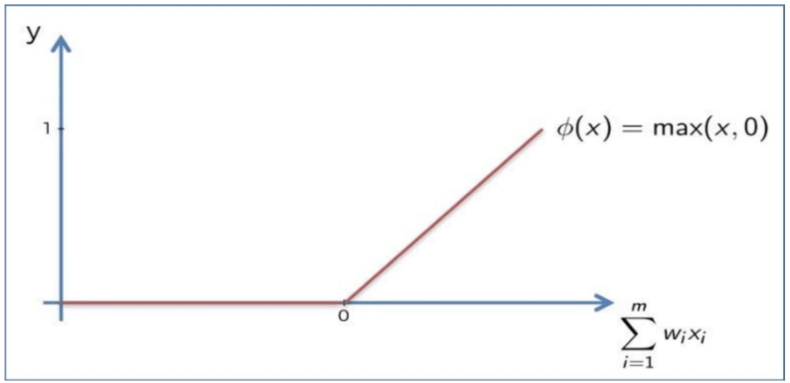
Neural Networks

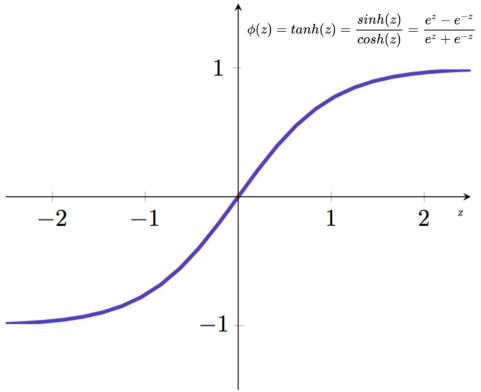


Neural Networks: Activation Functions

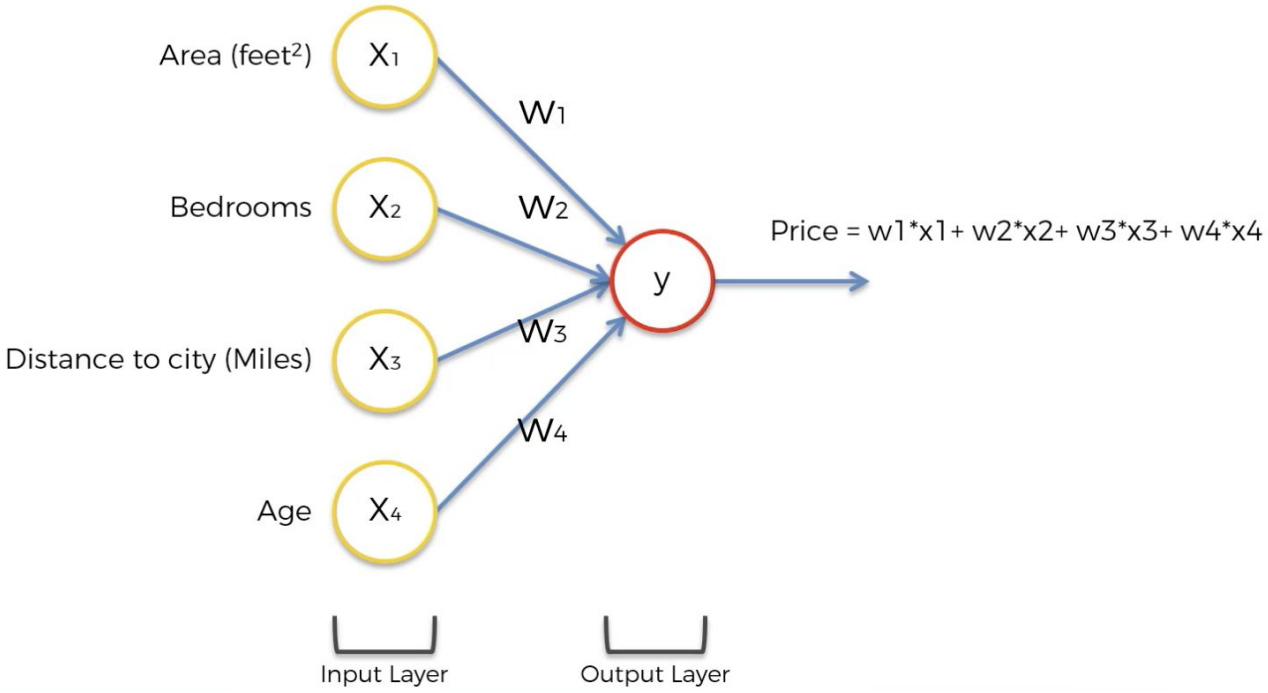




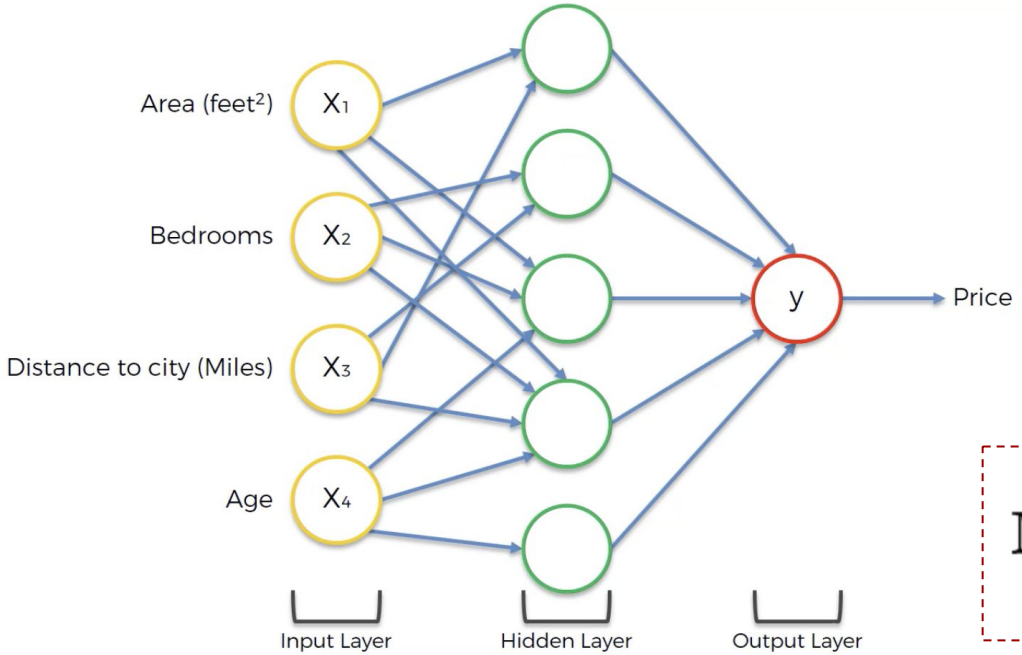




Neural Networks: Example (Property Valuation)



Neural Networks: Example (Property Valuation)

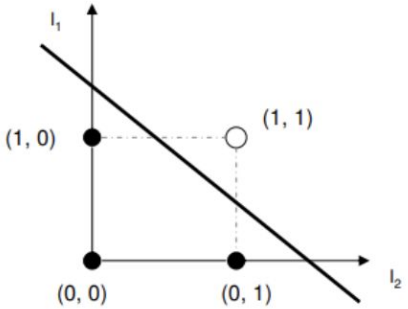


Cost Function

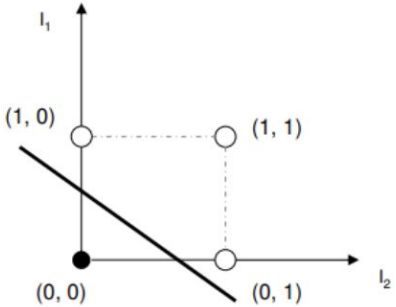
$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2.$$

Neural Networks: Perceptron

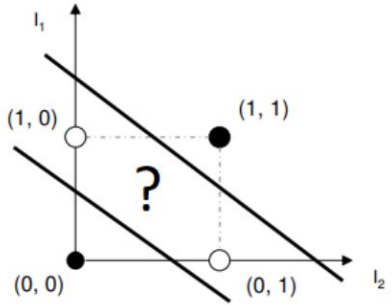
AND		
I_1	I_2	out
0	0	0
0	1	0
1	0	0
1	1	1



OR		
I_1	I_2	out
0	0	0
0	1	1
1	0	1
1	1	1



XOR		
I_1	I_2	out
0	0	0
0	1	1
1	0	1
1	1	0

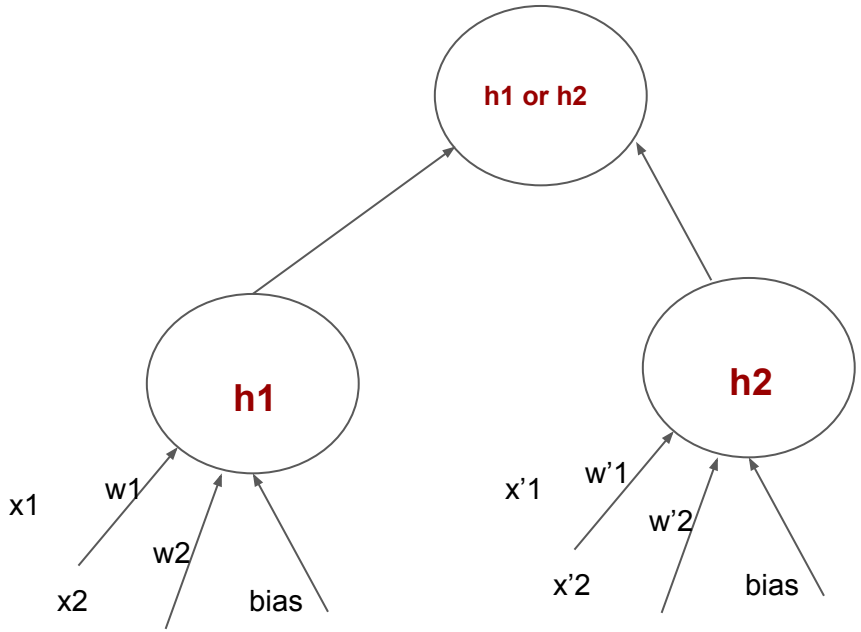


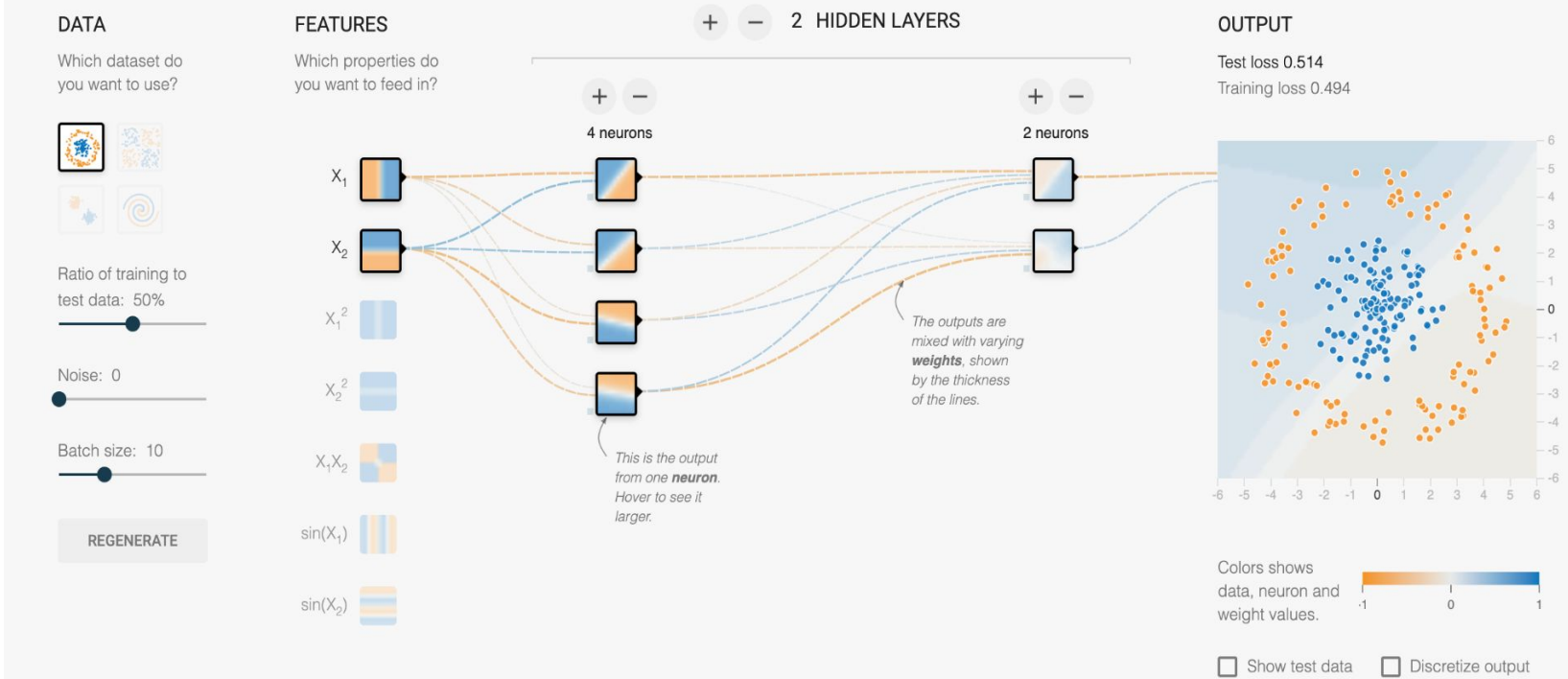
Neural Networks: Perceptron (XOR)

			h1	h2	
X1	X2	Y	X1 AND ⌐X2	⌐X1 AND X2	h1 OR h2
0	0	0	0	0	0
0	1	1	0	1	1
1	0	1	1	0	1
1	1	0	0	0	0

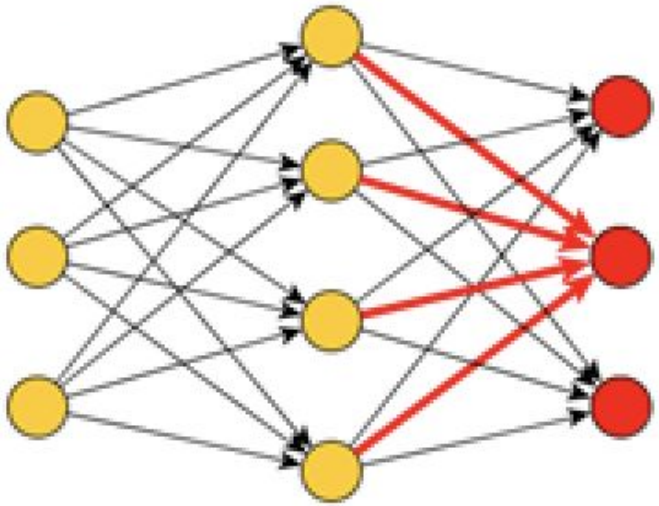
$$y = \begin{cases} 0, & \text{if } w \cdot x + b \leq 0 \\ 1, & \text{if } w \cdot x + b > 0 \end{cases}$$

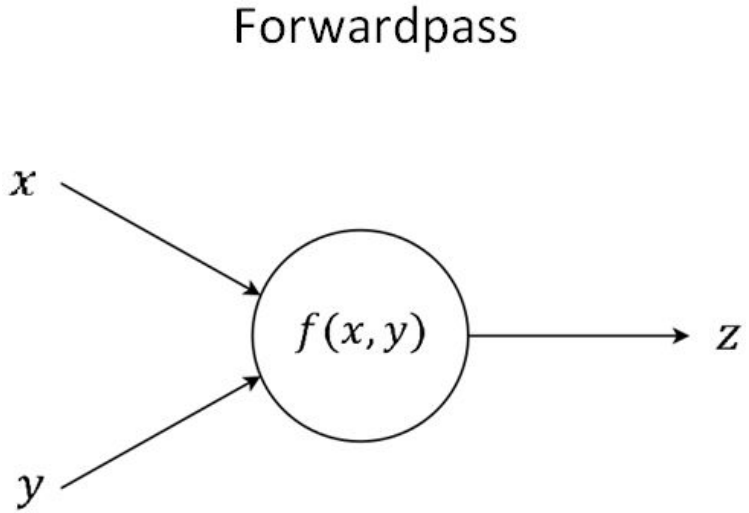
Neural Networks: Perceptron (XOR)




DATA
Which dataset do you want to use?
Ratio of training to test data: 50%
Noise: 0
Batch size: 10
REGENERATE
FEATURES
Which properties do you want to feed in?
X1
X2
X1²
X2²
X1X2
sin(X1)
sin(X2)
2 HIDDEN LAYERS
4 neurons
2 neurons
This is the output from one neuron. Hover to see it larger.
The outputs are mixed with varying weights, shown by the thickness of the lines.
OUTPUT
Test loss 0.514
Training loss 0.494
Colors shows data, neuron and weight values.
-1
0
1
Show test data
Discretize output

Neural Networks: Forward Propagation





Chain Rule

Suppose you have a composite function:

$$h(x) = f(g(x))$$

and both *f* and *g* are differentiable functions.

Then the chain rule says that the derivative of *h* is the following product:

$$h'(x) = f'(g(x))g'(x)$$

The chain rule expressed with partial derivatives:

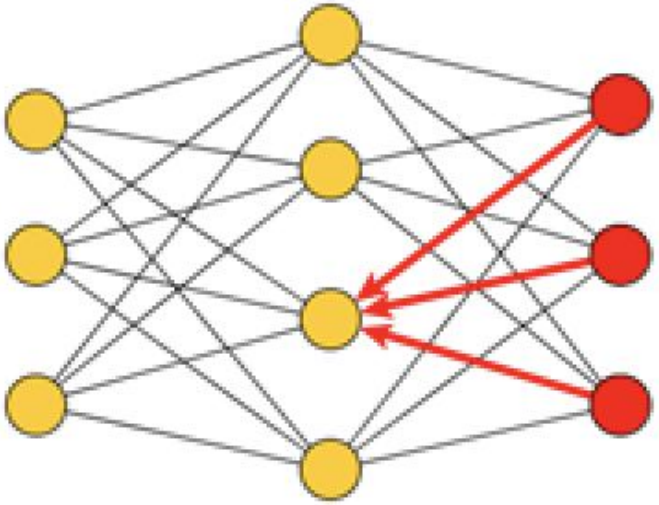
$$\frac{\partial}{\partial x} f(g(x)) = \frac{\partial f}{\partial g} \frac{\partial g}{\partial x}$$

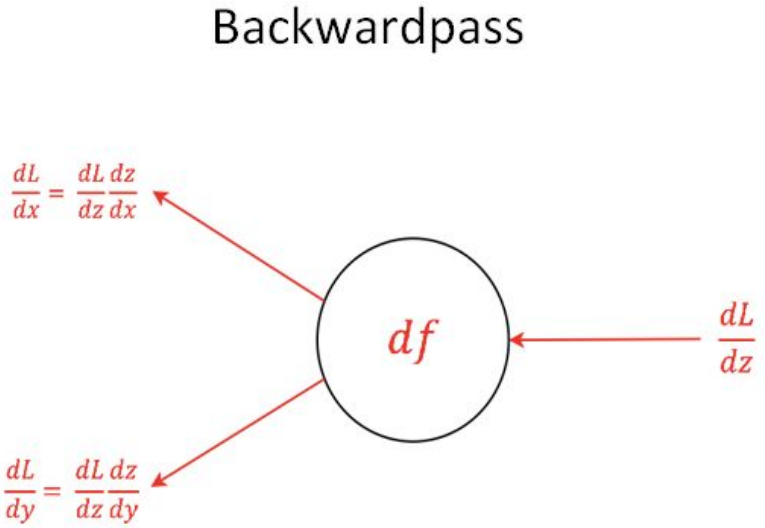
Instantaneous rate of change of *f* to *x*

Instantaneous rate of change of *f* to *g*

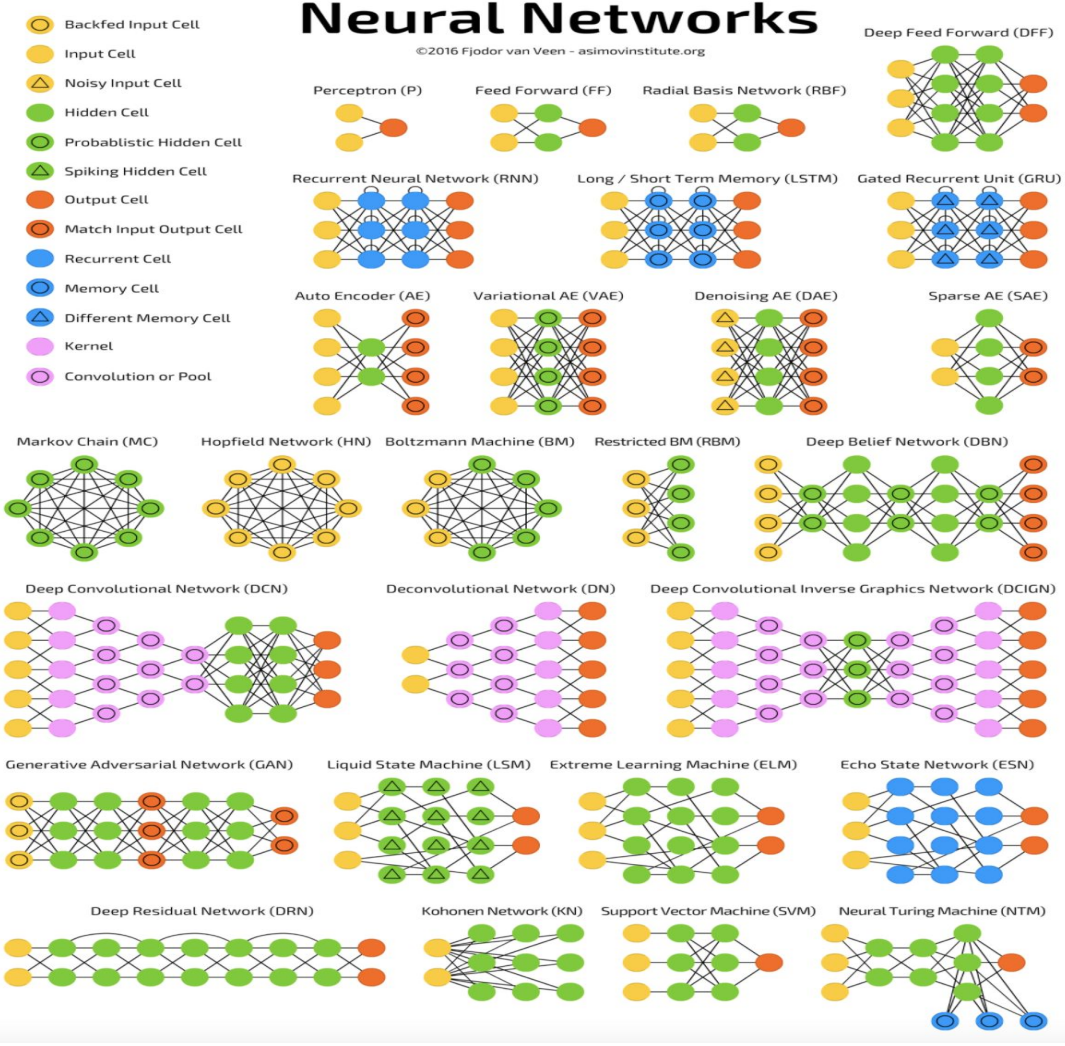
Instantaneous rate of change of *g* to *x*

Neural Networks: Backpropagation





Neural Networks



Neural Networks: Perceptron (XOR)

X1	X2	Y
0	0	0
0	1	1
1	0	1
1	1	0

→ $b \leq 0$

→ $b + w2 > 0$

→ $b + w1 > 0$

→ $b + w1 + w2 \leq 0$

$$y = \begin{cases} 0, & \text{if } w \cdot x + b \leq 0 \\ 1, & \text{if } w \cdot x + b > 0 \end{cases}$$