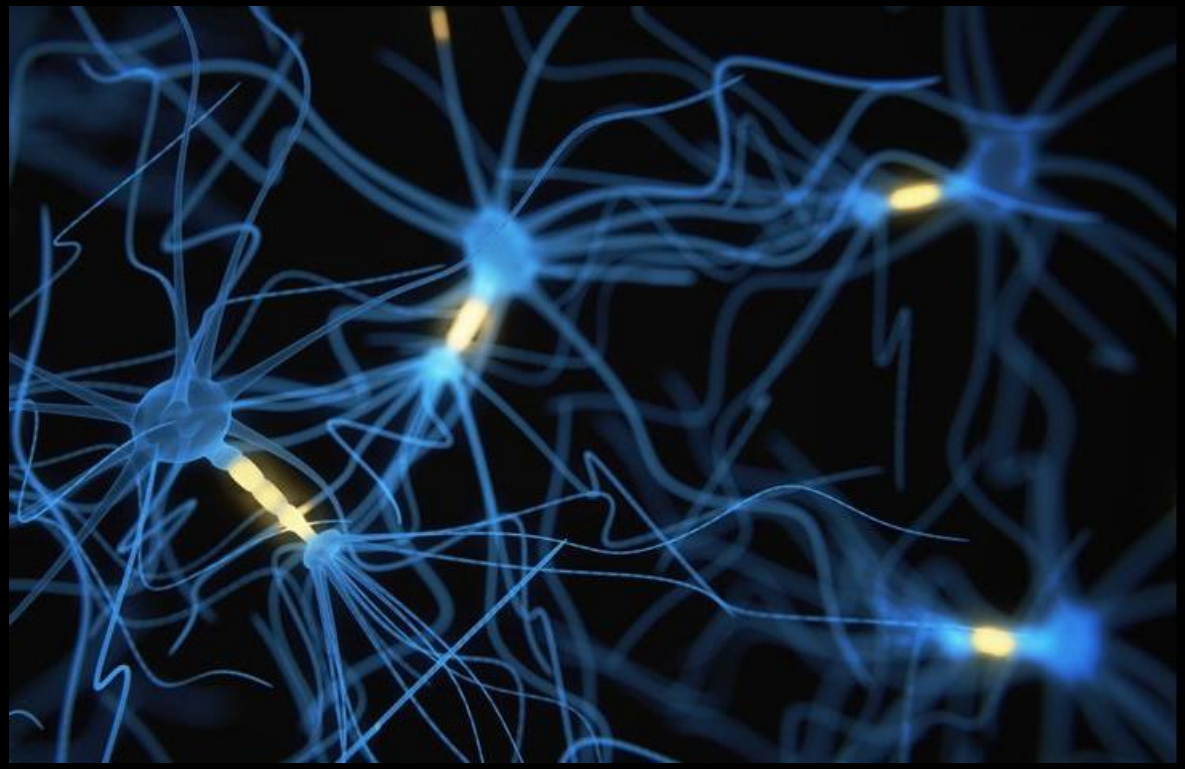
Introduction to Neural Networks and Deep Learning

Topics to be covered

- The Perceptron
- Features, Weights and Activation Functions
- Learning of Neural Network
- Rise of Deep Learning

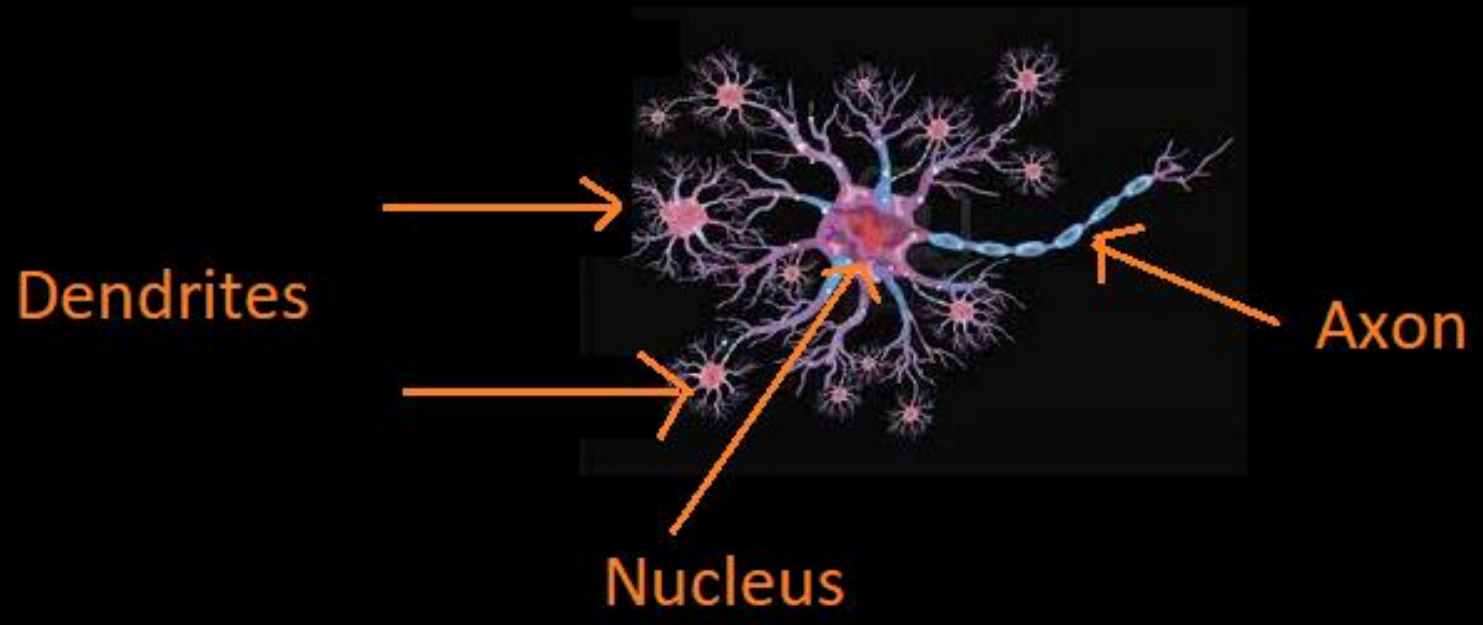
Perceptron

Brain Neuron



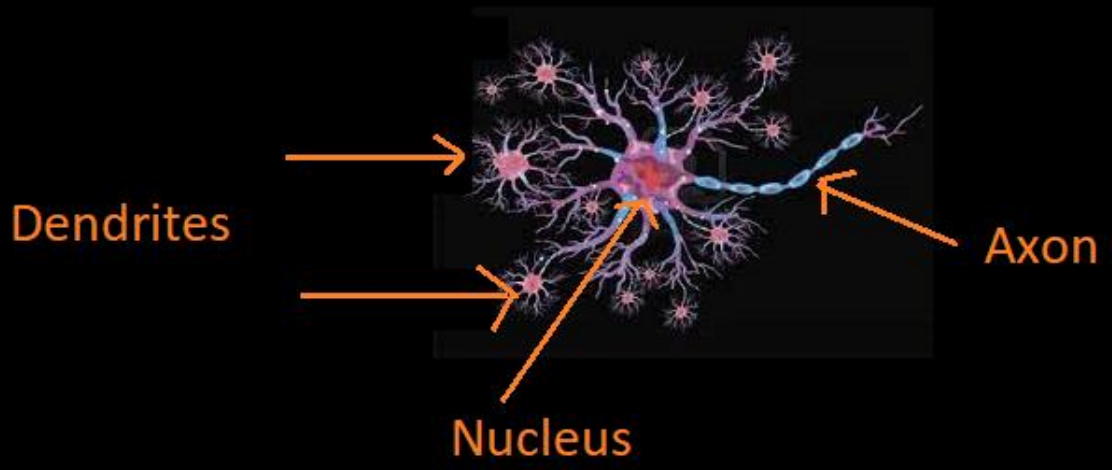
Google Images

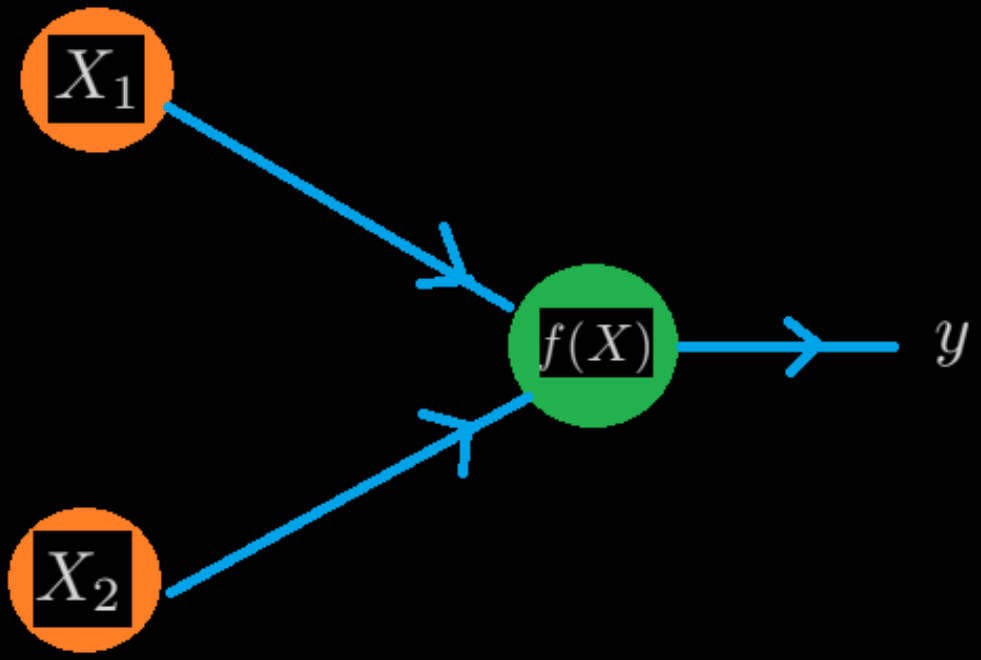
Brain Neuron



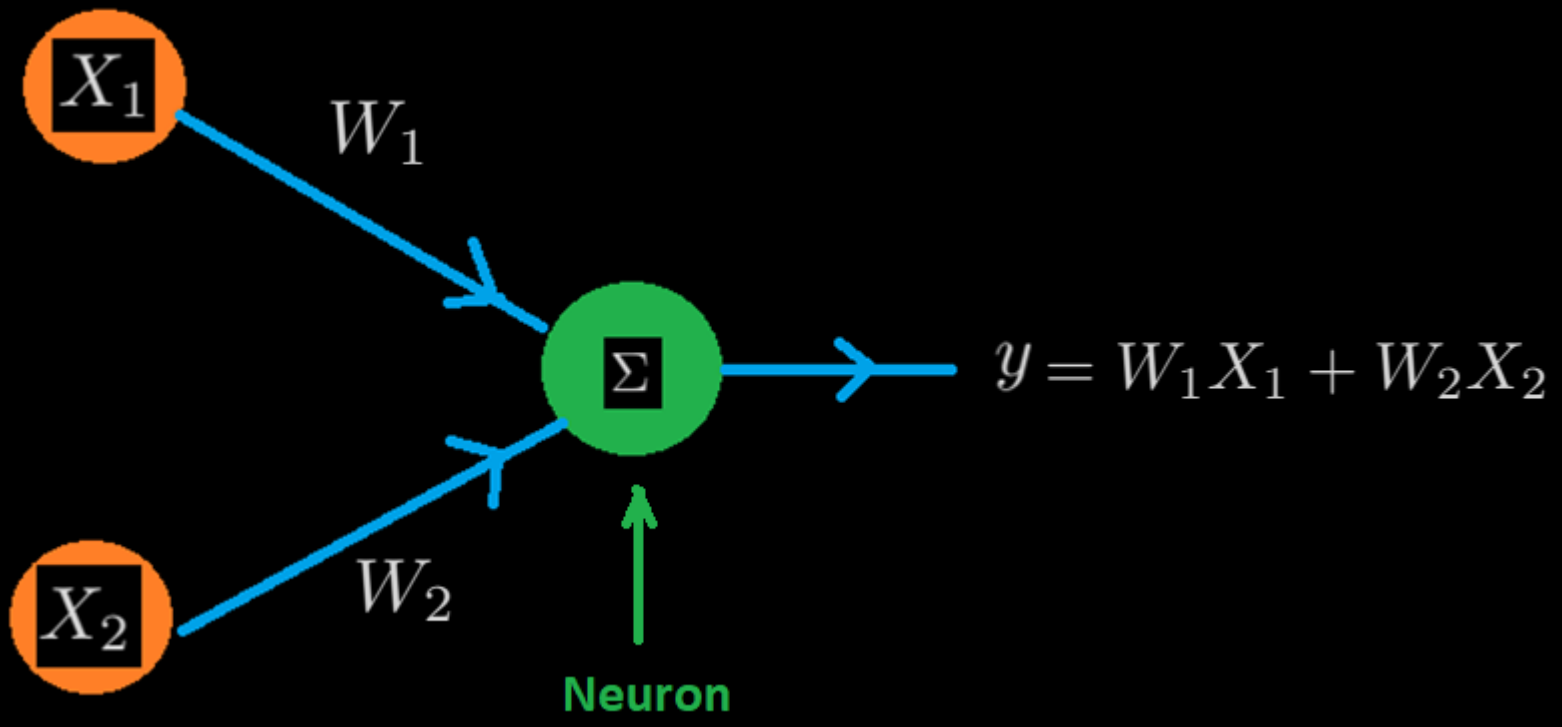
Google Images

Real Neuron to Artificial Neuron (Perceptron)

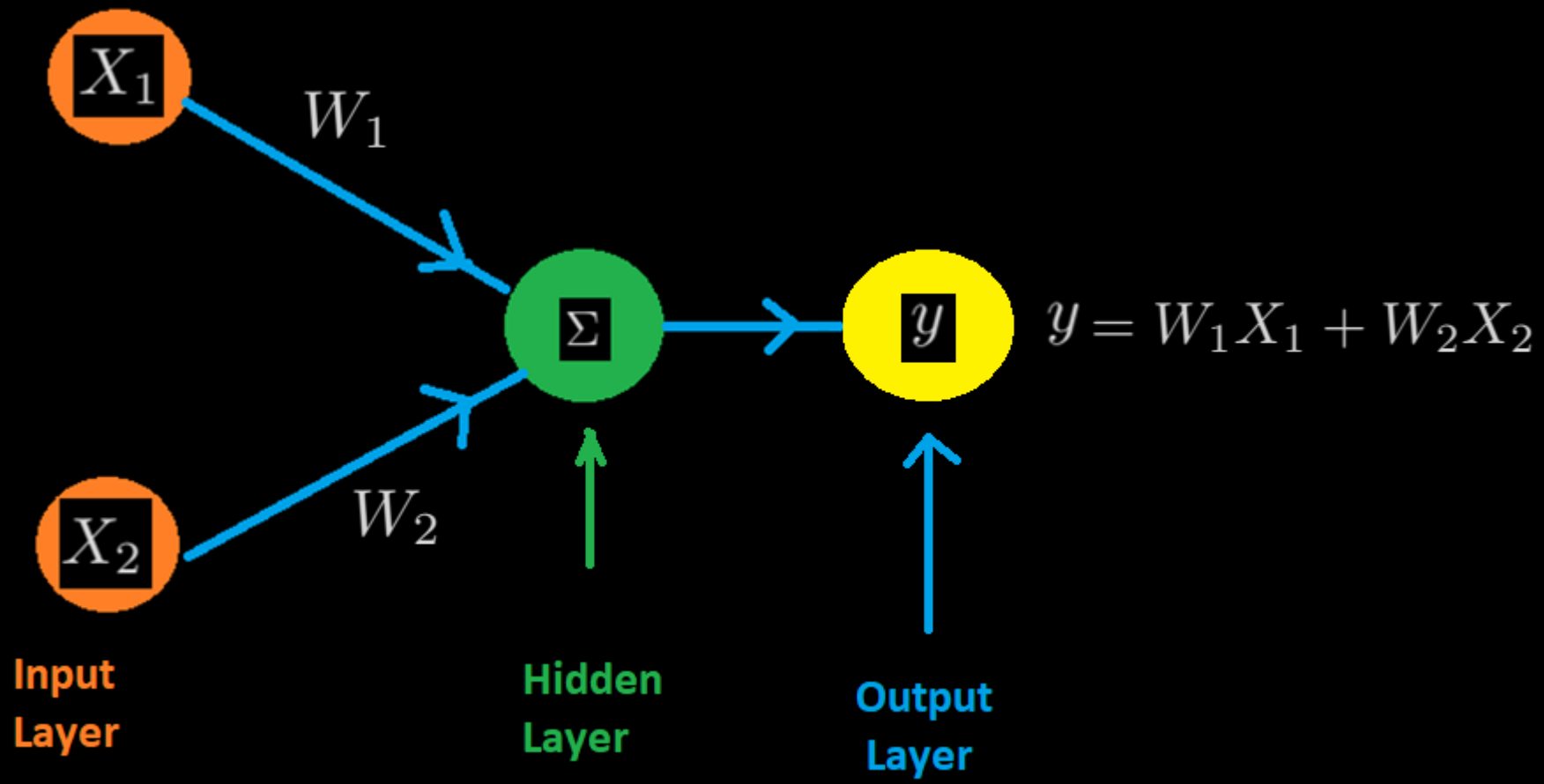




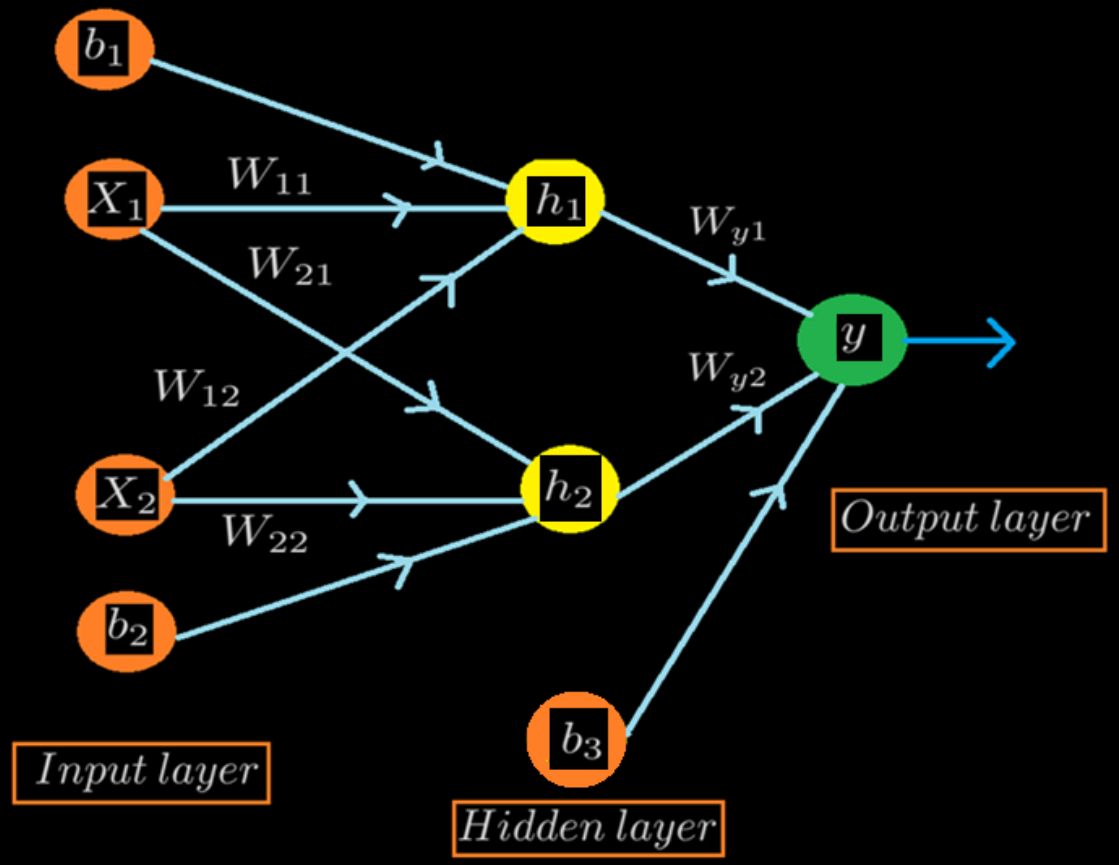
The Perceptron (Modelling the Brain Neuron)



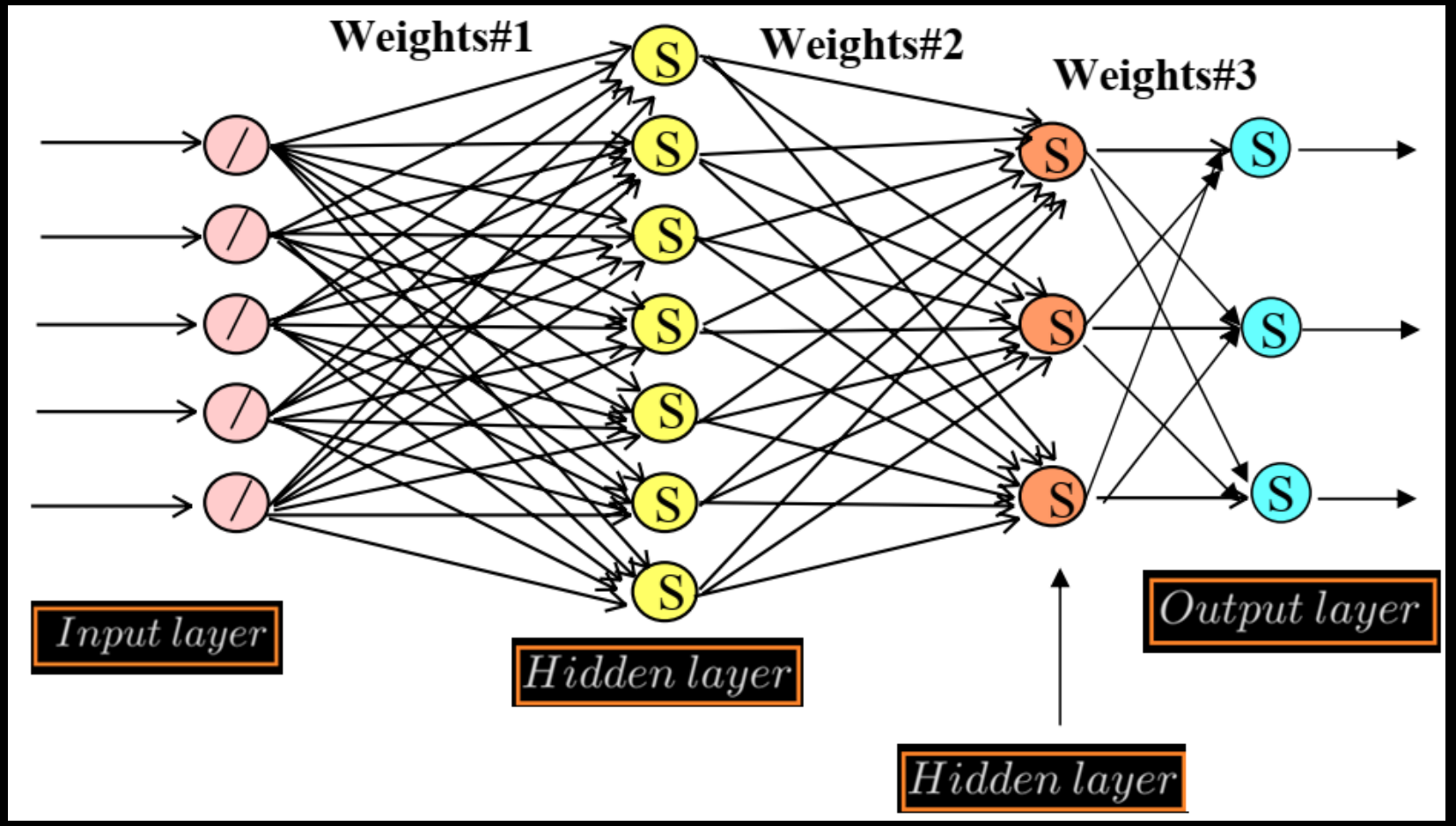
Neural Network



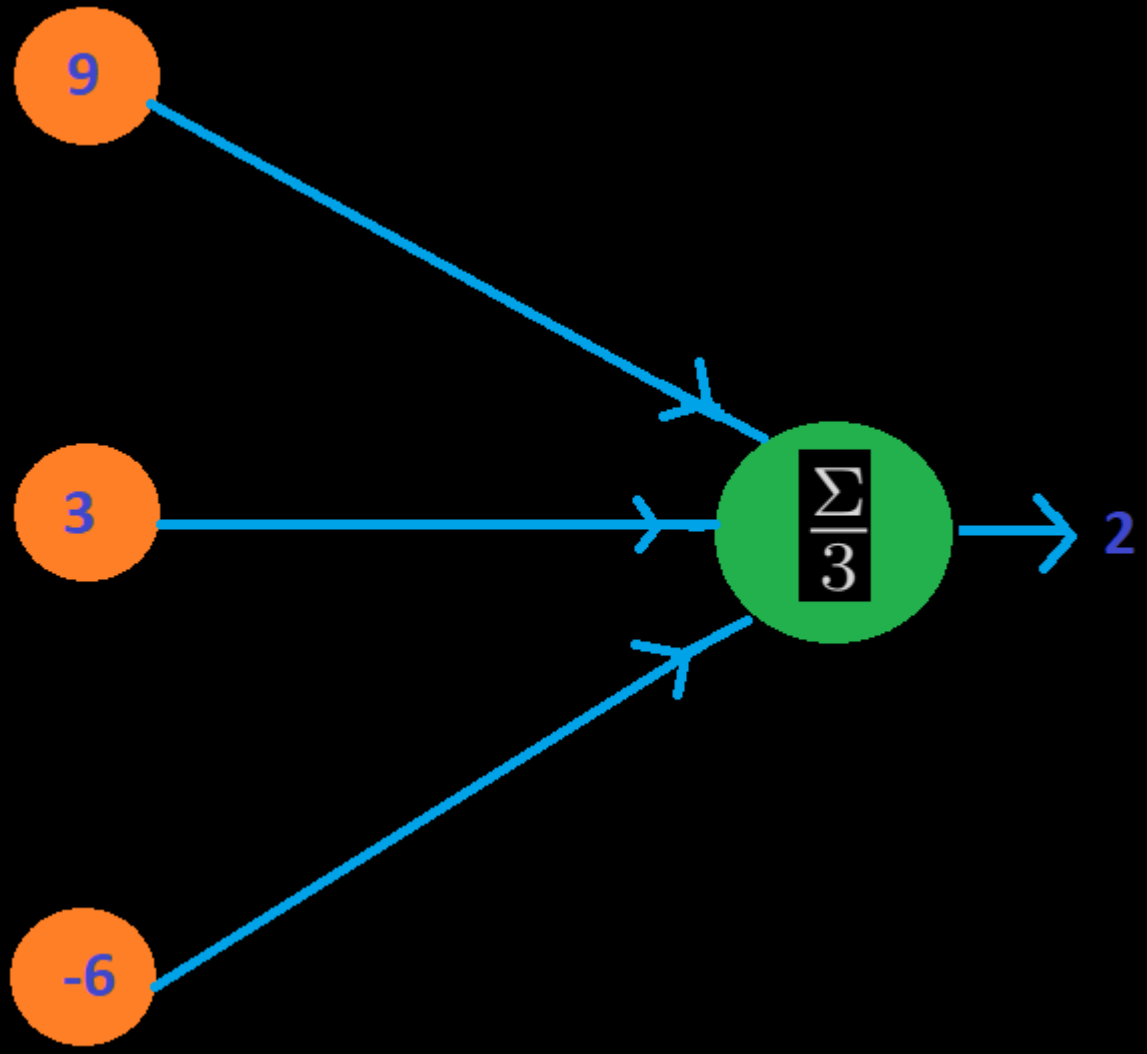
Neural Network



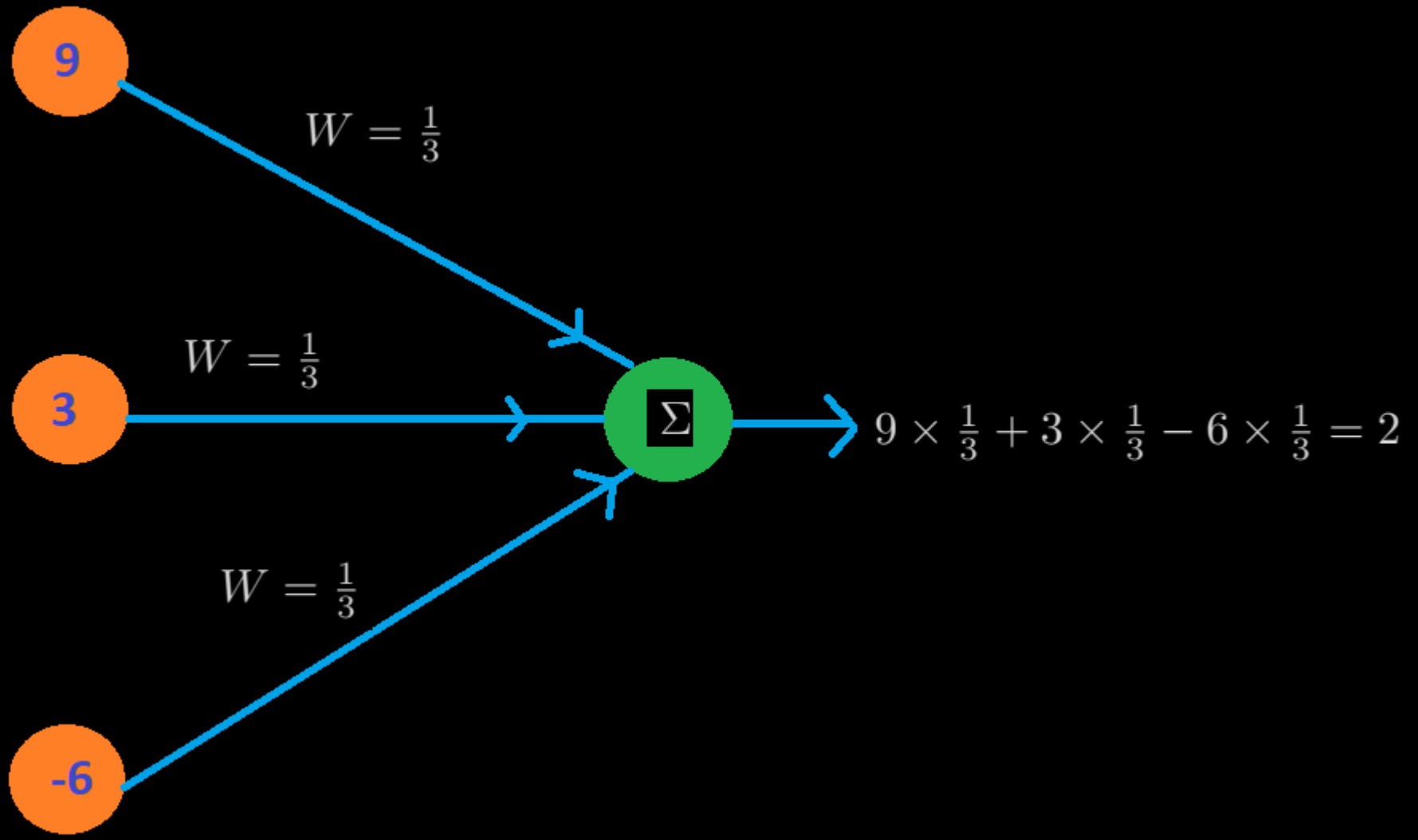
Multilayer Neural Network With Two Layers of Hidden Neurons



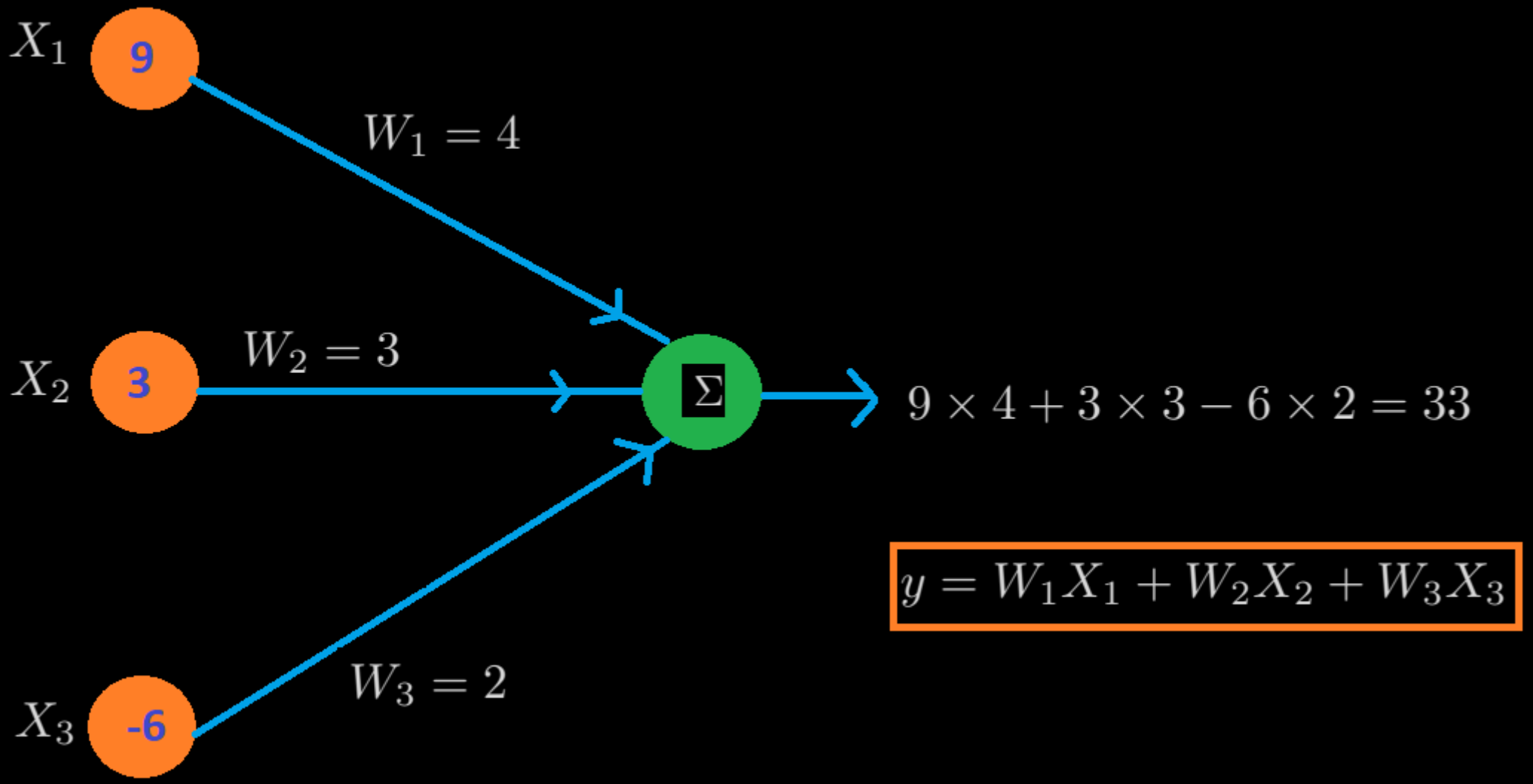
Averaging Machine



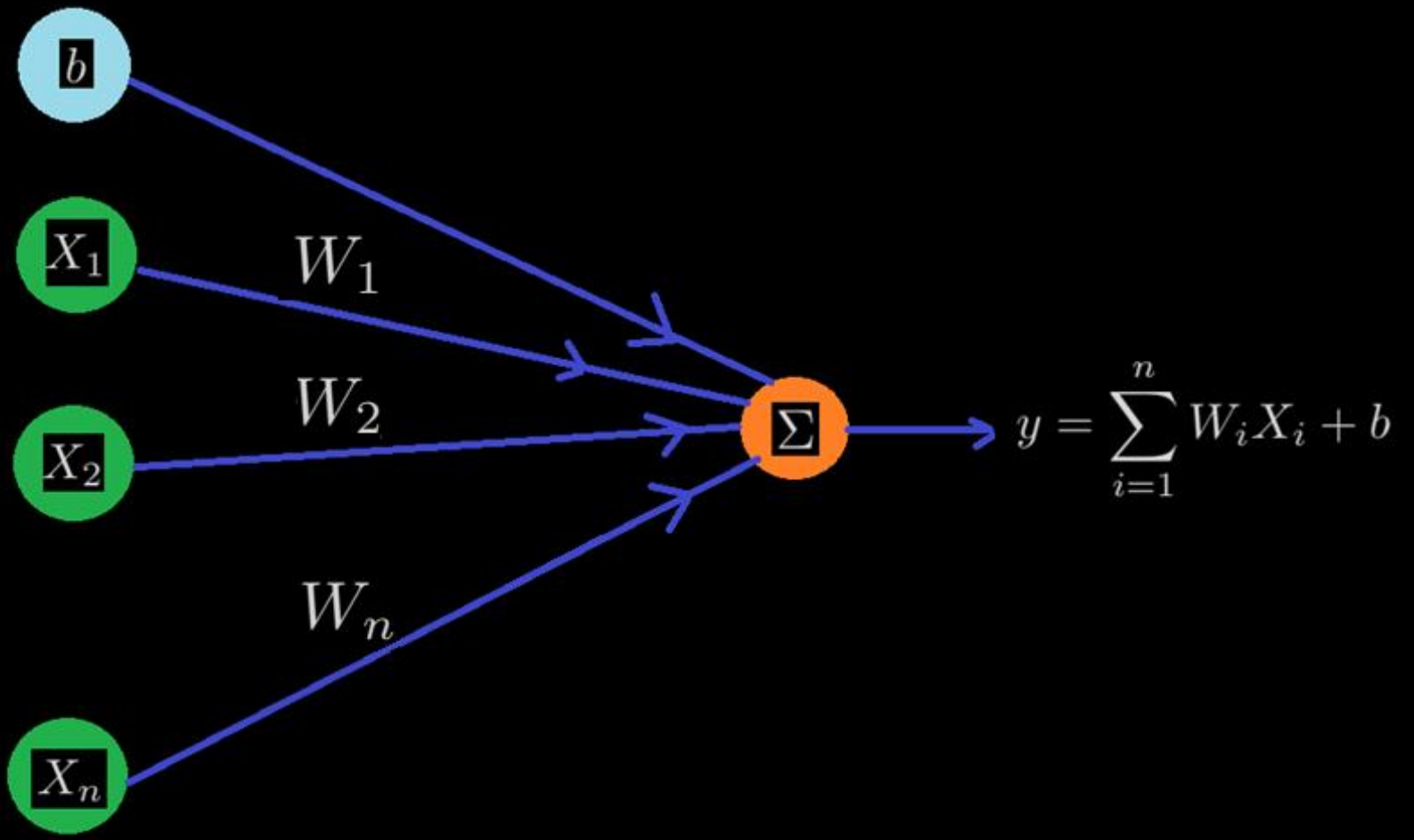
Another Averaging Machine



Weighted Averaging Machine



Basic Perceptron With Inputs and Bias

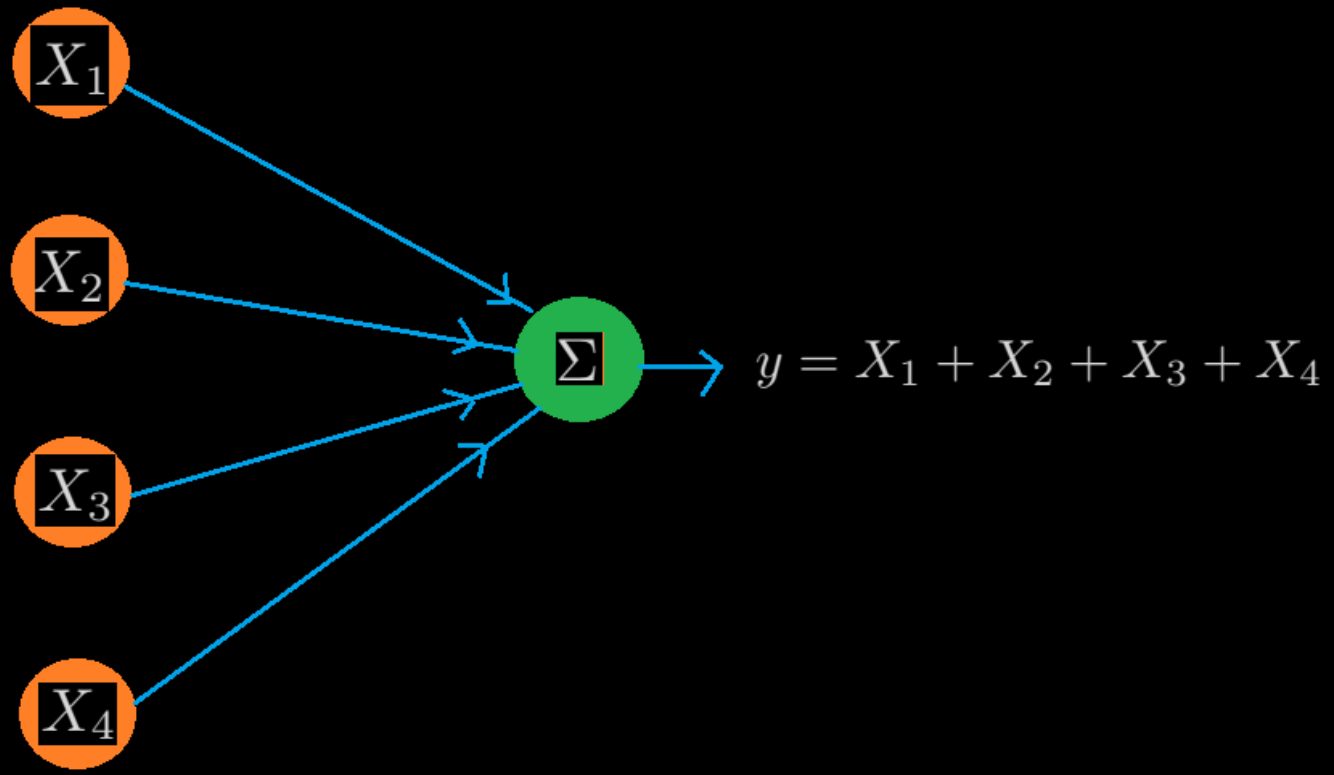


Features, Weights And Activation Functions

Suppose few candidates appear for interview in a company and the company's output after the interview will be either selected (1) OR not selected (0). The company will classify (selected OR not selected)the candidates based on some qualities (features) of the candidates.

- Qualification relevant to the job
- Experience
- Communication skill
- Address of the candidate

Lets call these features $X_1, X_2, X_3,$ *and* X_4.



Lets give importance (Weights) to each feature based on the job requirements

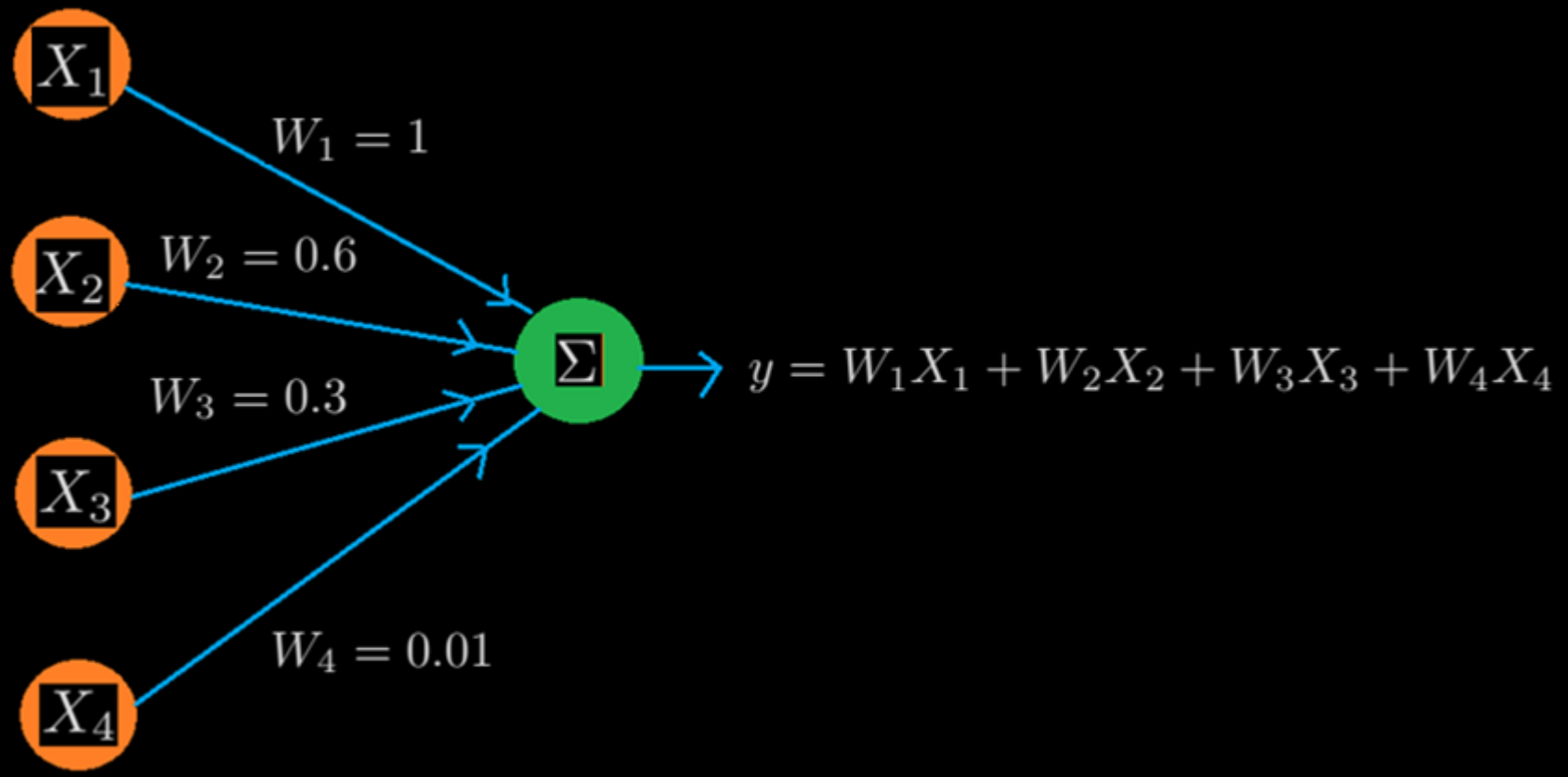
$W_1 = 1$

$W_2 = 0.6$

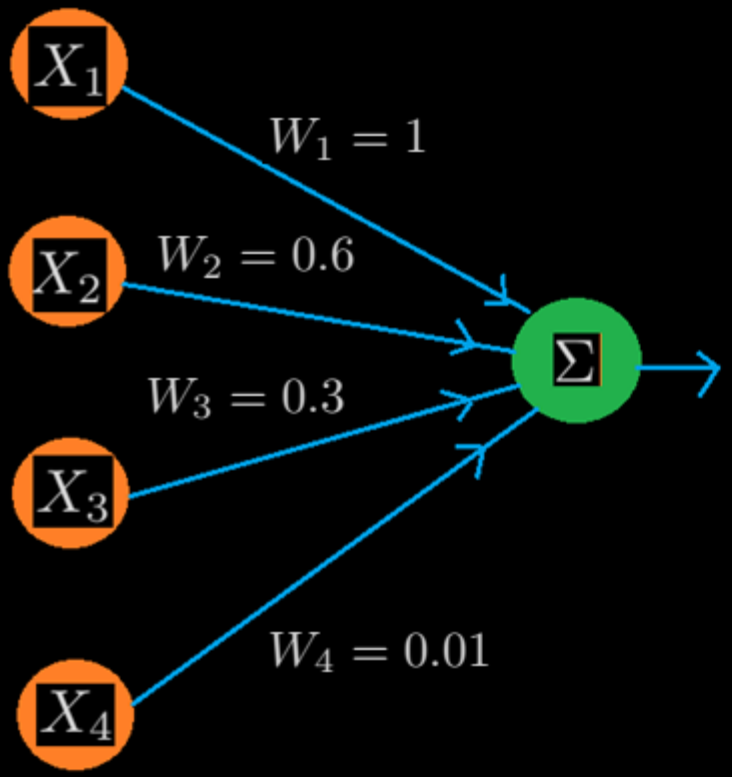
$W_3 = 0.3$

$W_4 = 0.01$

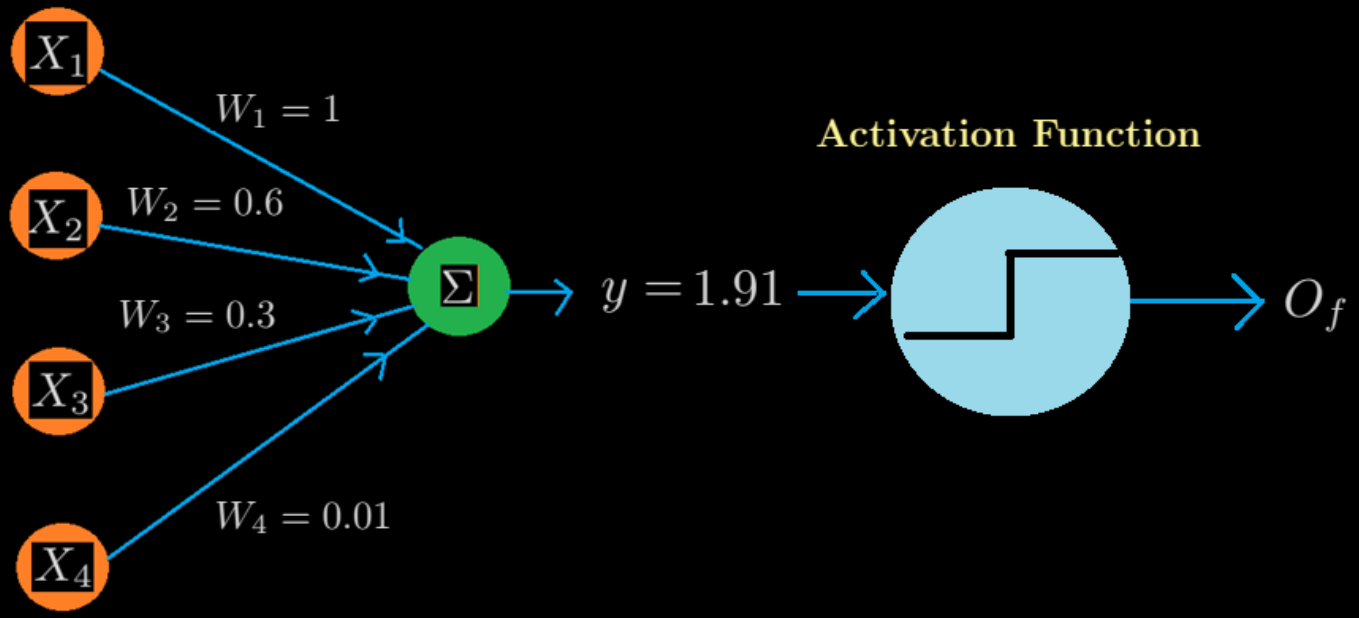
- Qualification relevant to the job
- Experience
- Communication skill
- Address of the candidate


X_1
$W_1 = 1$
X_2
$W_2 = 0.6$
Σ
$y = W_1X_1 + W_2X_2 + W_3X_3 + W_4X_4$
$W_3 = 0.3$
X_3
$W_4 = 0.01$
X_4

If $X_1 = X_2 = X_3 = X_4 = 1$



$$y = 1 * 1 + 0.6 * 1 + 0.3 * 1 + 0.01 * 1$$
$$= 1.91$$



Threshold Activation Function

$$O_f = \begin{cases} 1 & \text{if } y>1 \\ 0 & \text{if } y<1 \end{cases}$$

How Neural Networks Learn ?

Neural Network learns in two stages / propagations

- Forward Propagation
- Backward Propagation

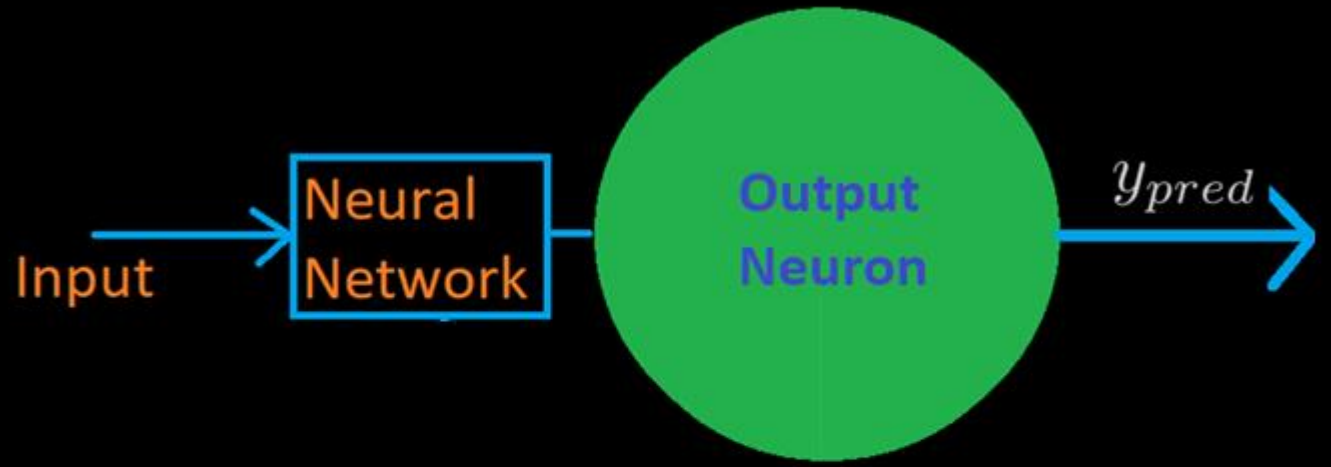
Forward Propagation

In forward propagation, we calculate error of the neural network.

Backward Propagation

In backward propagation, we propagate the error back to the Layers of neural network and we update the weights and bias iteratively.

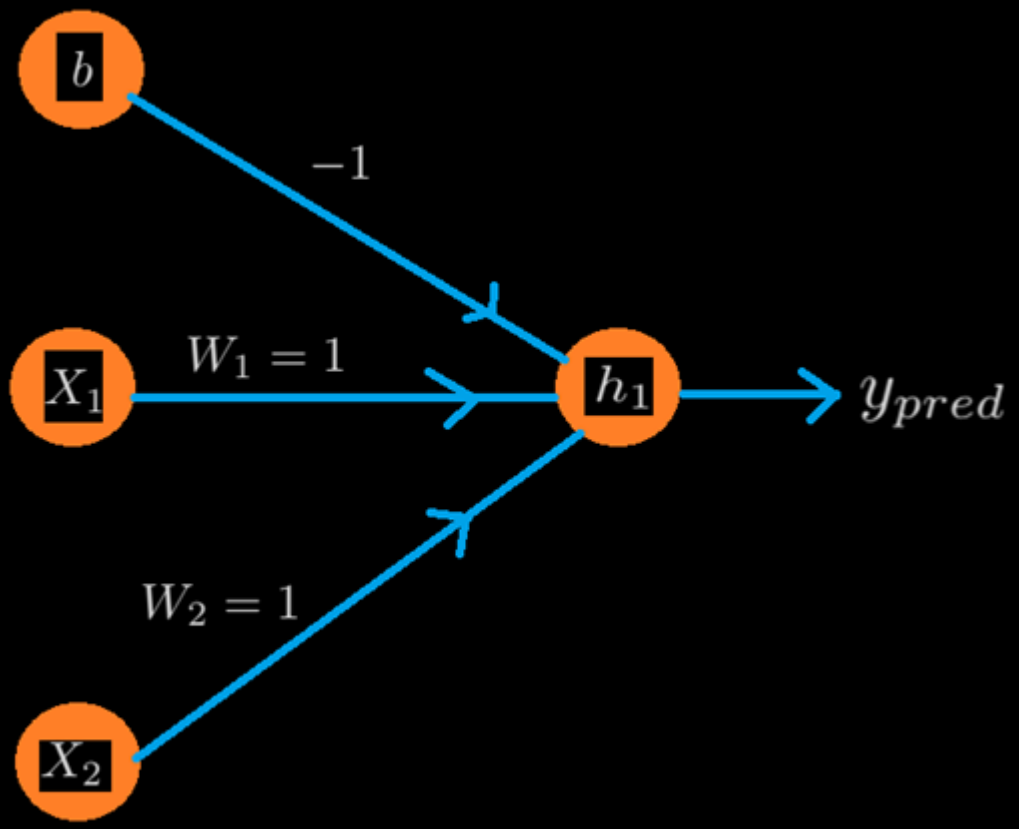
Forward Propagation



X_1	X_2	y
0	0	0
0	1	1
1	0	1
1	1	1

$$E = y - y_{pred}$$

Forward Propagation



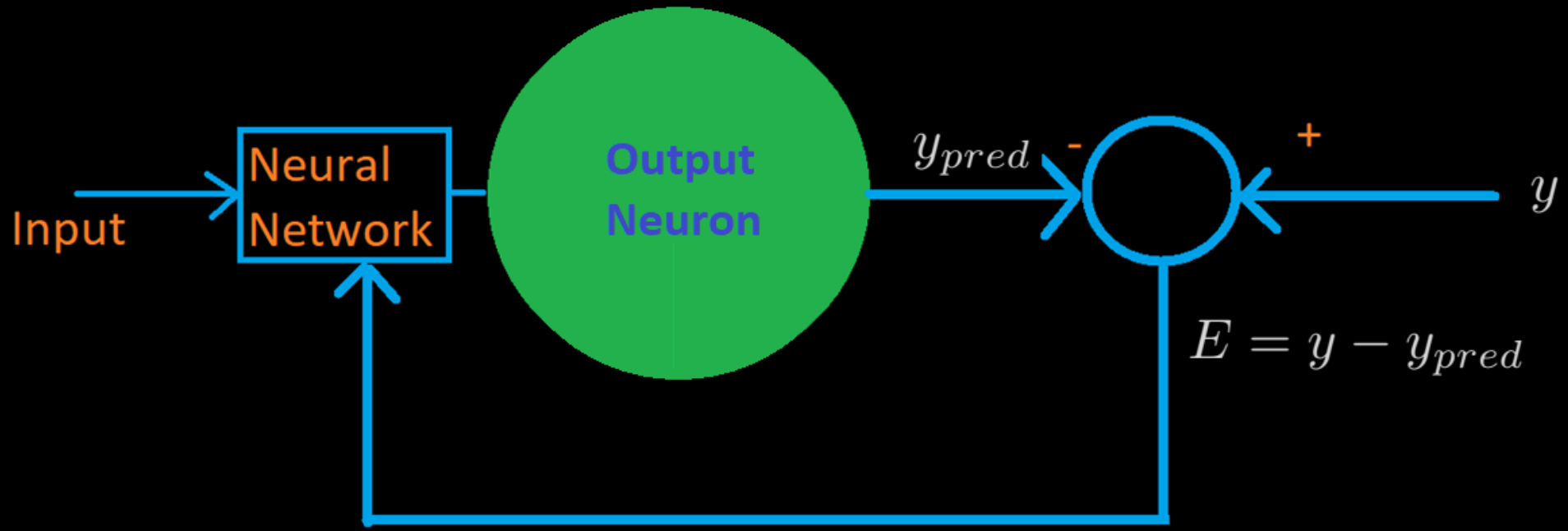
X_1	X_2	y	y_{pred}
0	0	0	-1
0	1	1	
1	0	1	
1	1	1	

Calculation of Error

$$E = y - y_{pred}$$

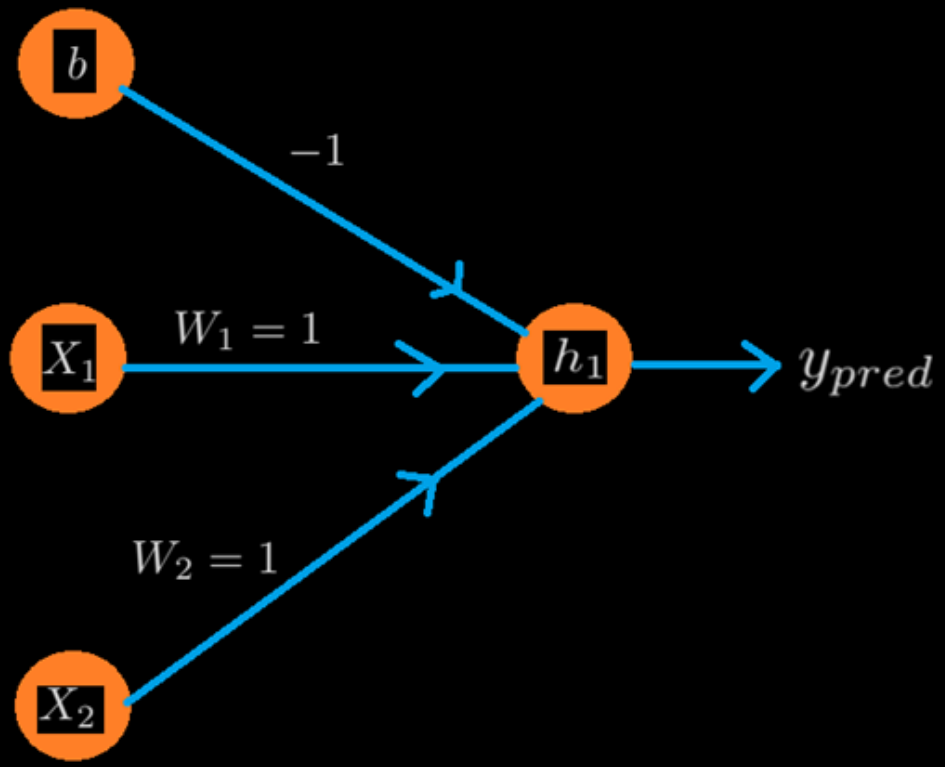
$$Loss\ function = \frac{1}{2}(y - y_{pred})^2$$

Backward Propagation



Error is propagated back through the layers of the Neural Network and the weights of the Neural Network are adapted iteratively.

Backward Propagation



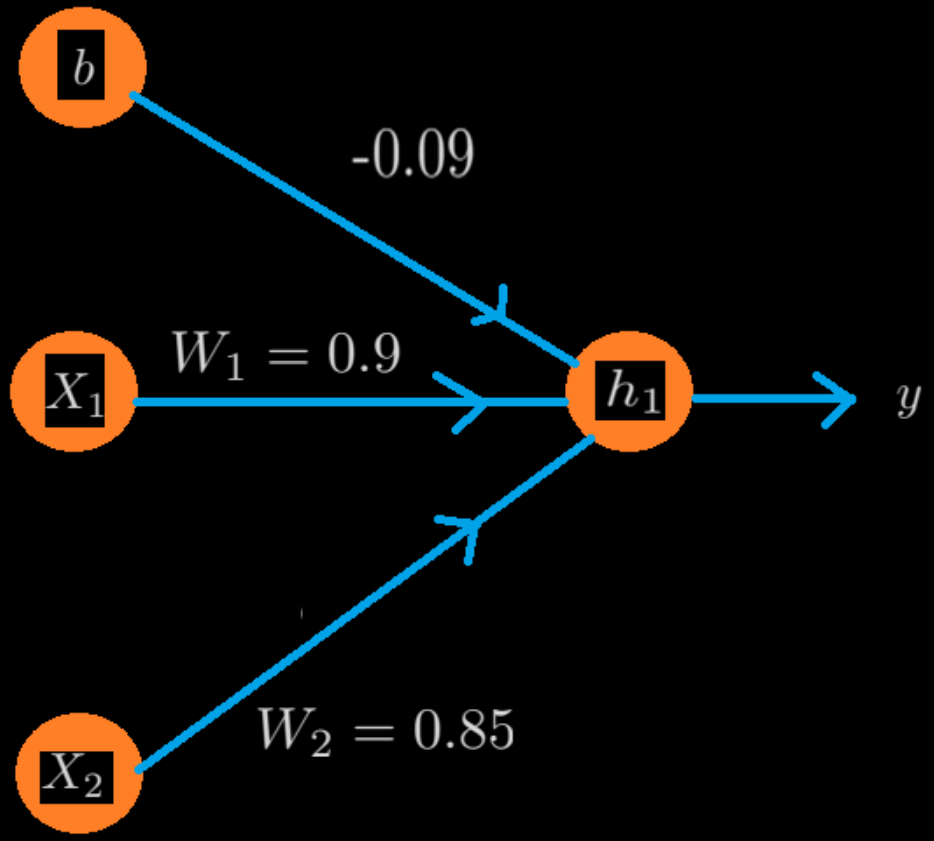
Updating W_1

$$\frac{\partial E}{\partial W_1} = \frac{\partial E}{\partial y}\frac{\partial y}{\partial h_1}\frac{\partial h_1}{\partial W_1}$$

$$W_1(updated) = W_1(old) + \eta\frac{\partial E}{\partial W_1}$$

where η = Learning rate

Forward Propagation (Iter 02)



X_1	X_2	y	y_{pred}
0	0	0	-0.09
0	1	1	
1	0	1	
1	1	1	

New Predicted Value

The Rise of Deep Neural Networks
(Deep Learning)

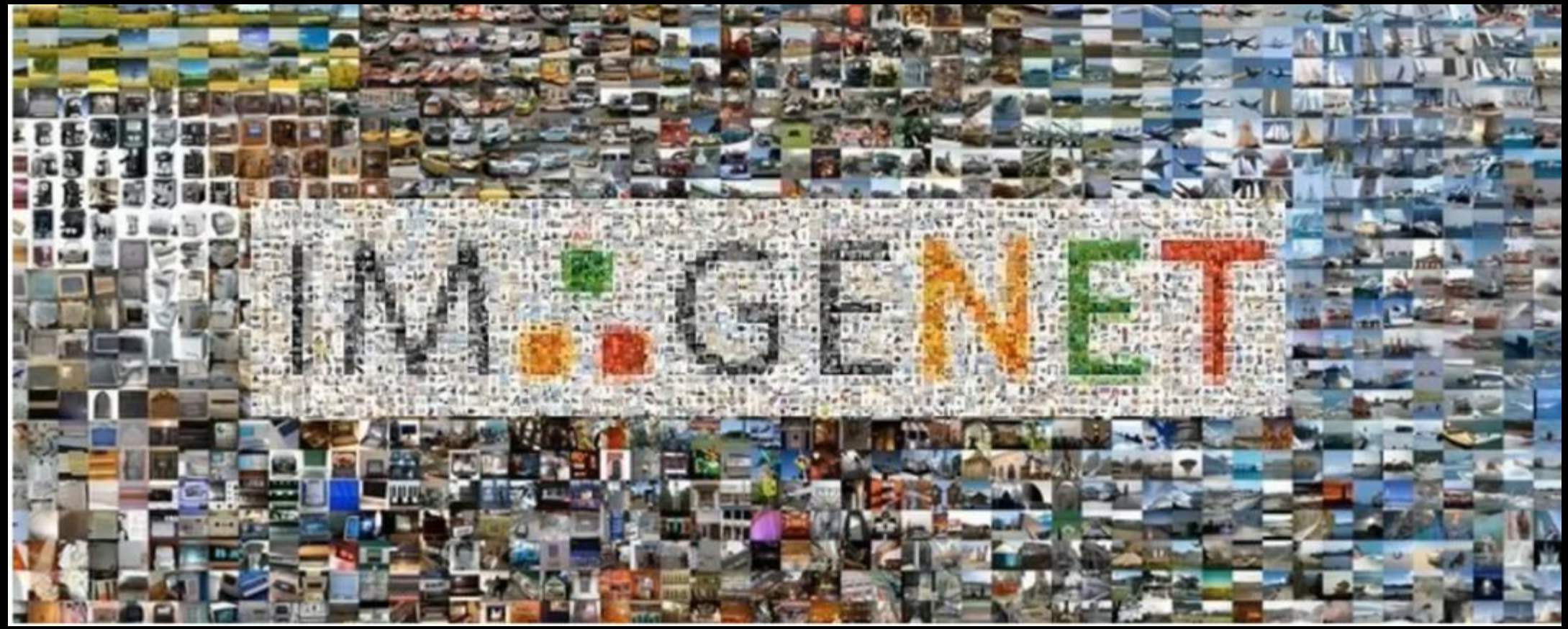
ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

Availability of Large Datasets



Google Images

Computational Resources

Google Images

State-of-the-art Deep Learning Models

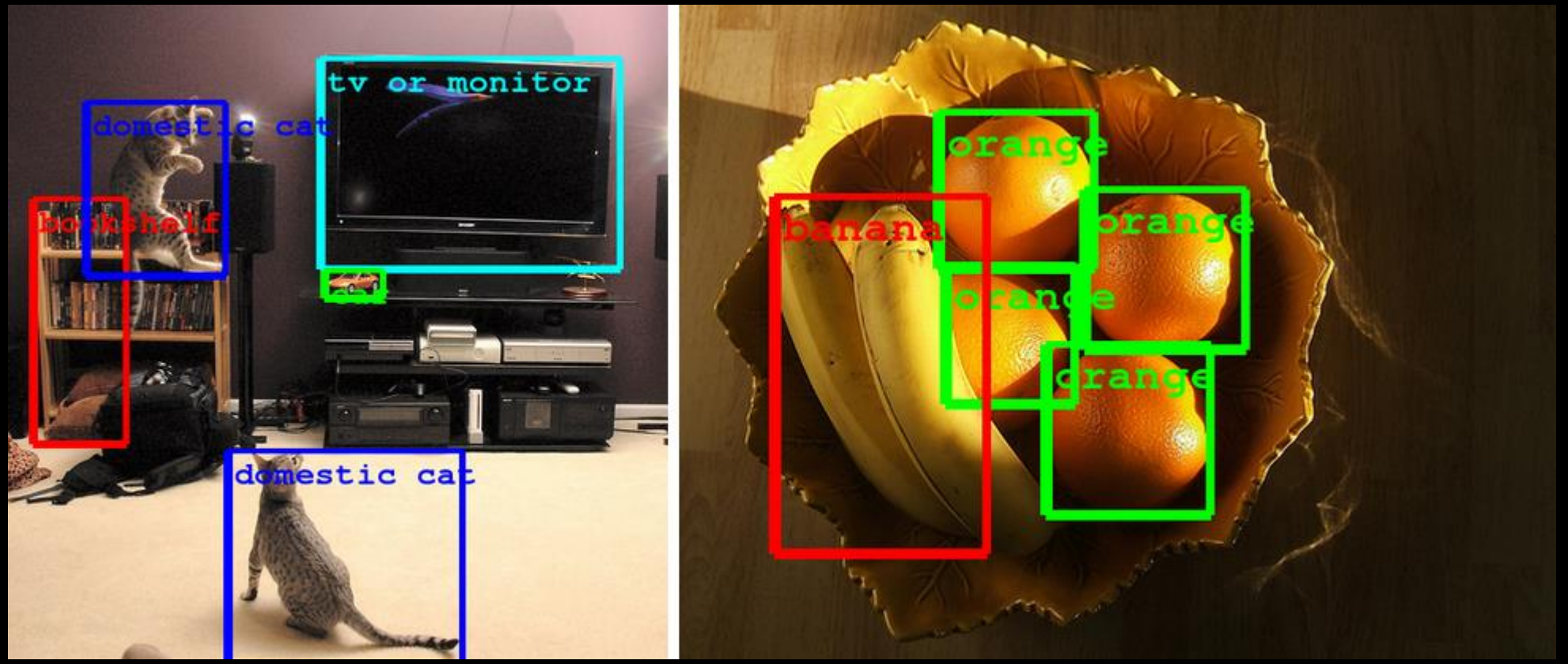
- VGG-16, ResNet-18, ResNet-50.
- Long-Short-Term-Memory (LSTM) and its variants.
- Autoencoders.
- Generative Adversarial Network (GAN).
- Transformers.

Applications of Deep Learning Models

Face Detection

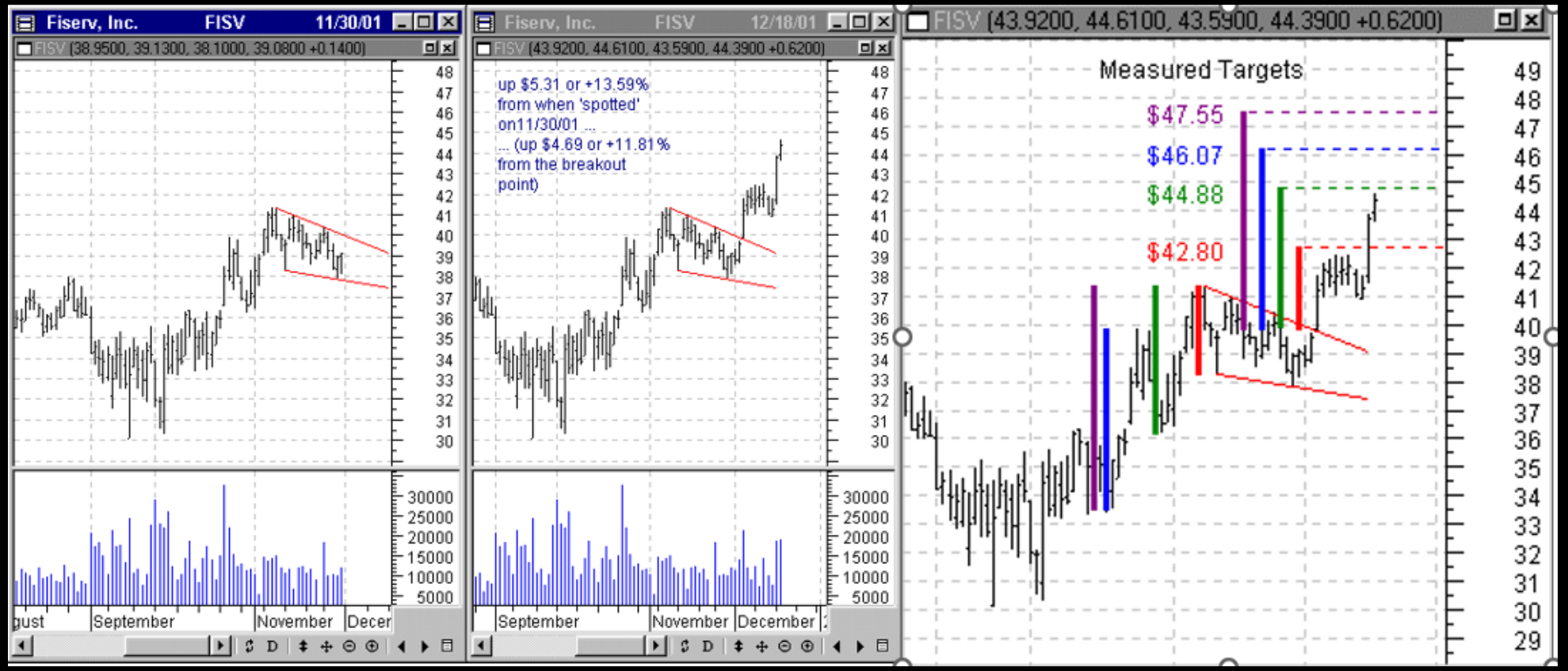
. Yu, J. Yang, A direct LDA algorithm for high-dimensional data with application to face recognition, Pattern Recognition 34 (2001)

Object Detection



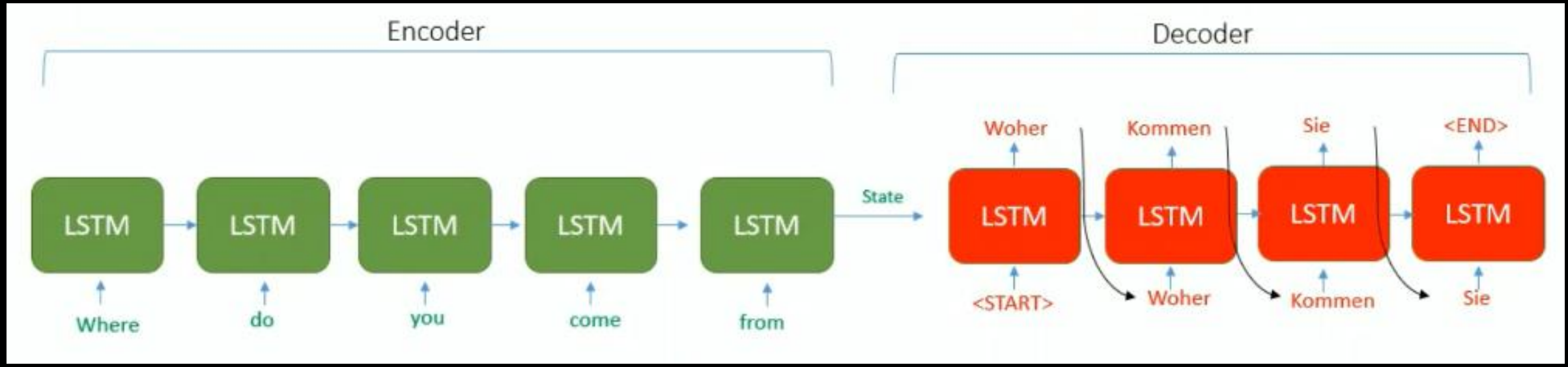
. Szegedy Et Al, Going Deeper with Convolutions , CVPR 2015.

Financial Forecasting



Google Images

Machine Translation



Google Images

Thank you!

Thank you!