Artificial Neural Network

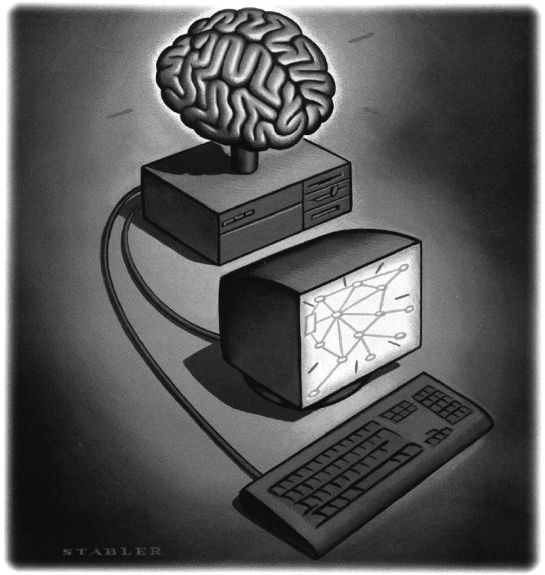
Basics of Neural Network

- What is a Neural Network
- Neural Network Classifier
- Data Normalization
- Neuron and bias of a neuron
- Single Layer Feed Forward
- Limitation
- Multi Layer Feed Forward
- Back propagation

Neural Networks

What is a Neural Network?

•Biologically motivated approach to machine learning



Similarity with biological network

Fundamental processing elements of a neural network is a neuron

1.Receives inputs from other source

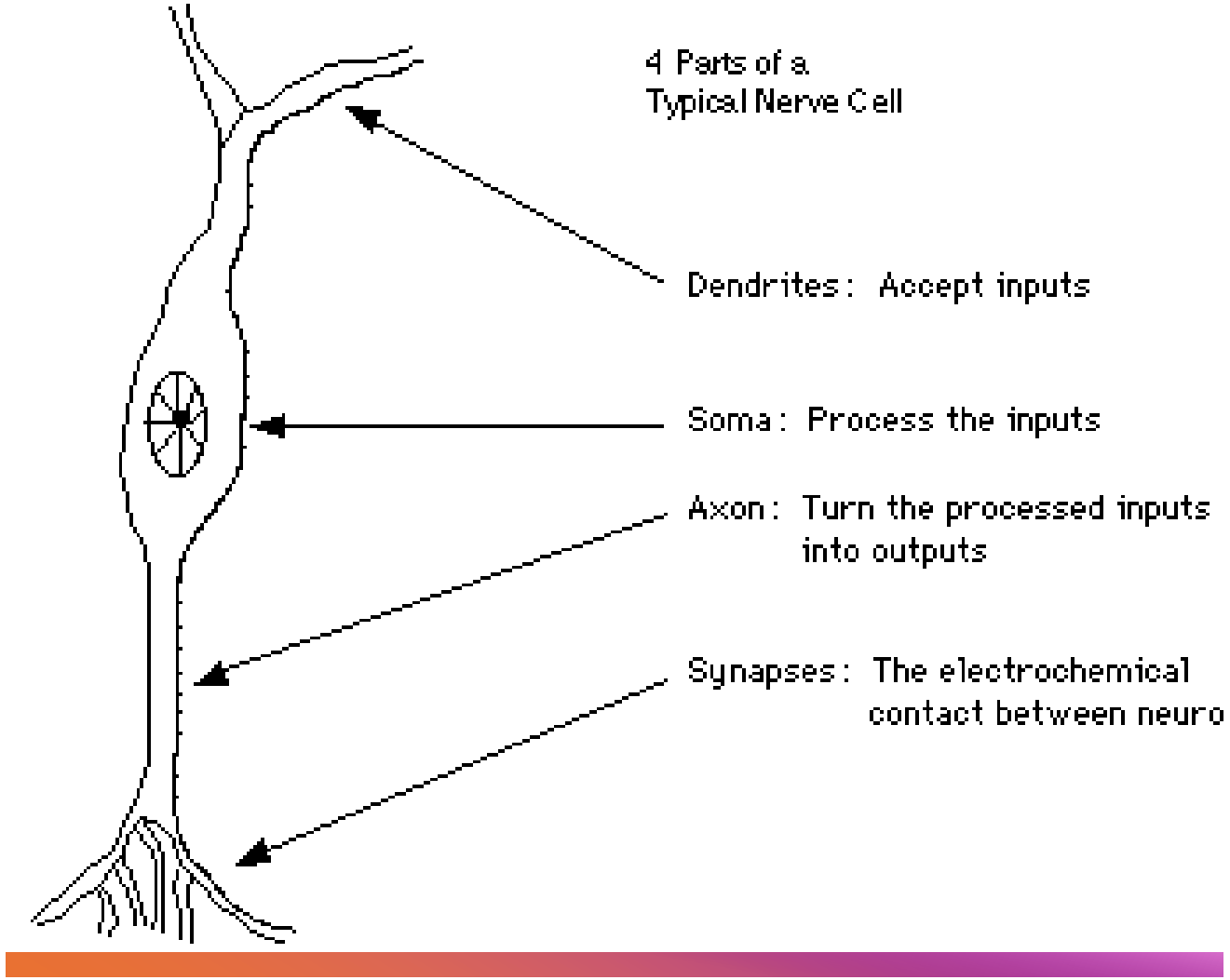
2.Combines them in someway

3.Performs a generally nonlinear operation on the result

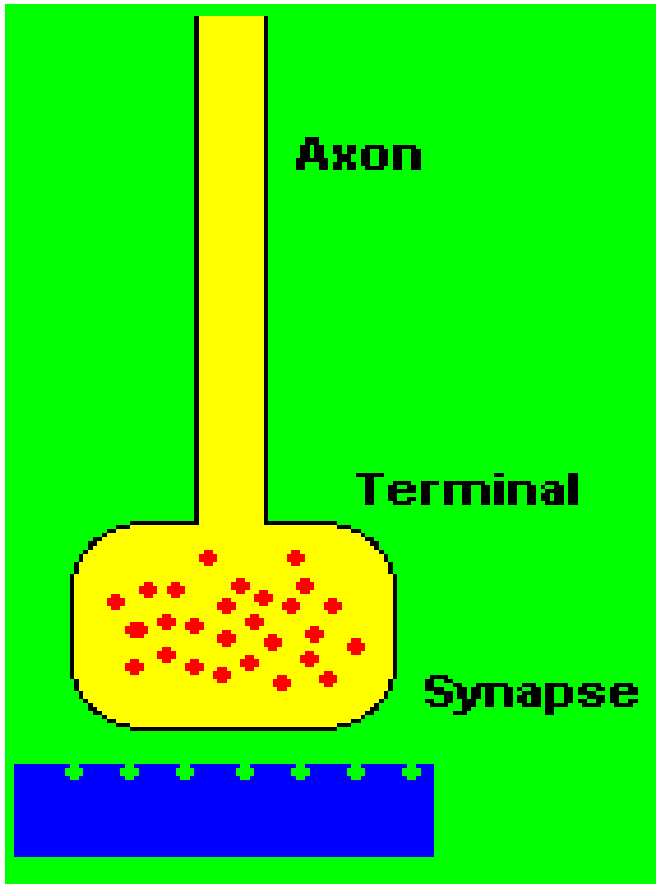
4.Outputs the final result

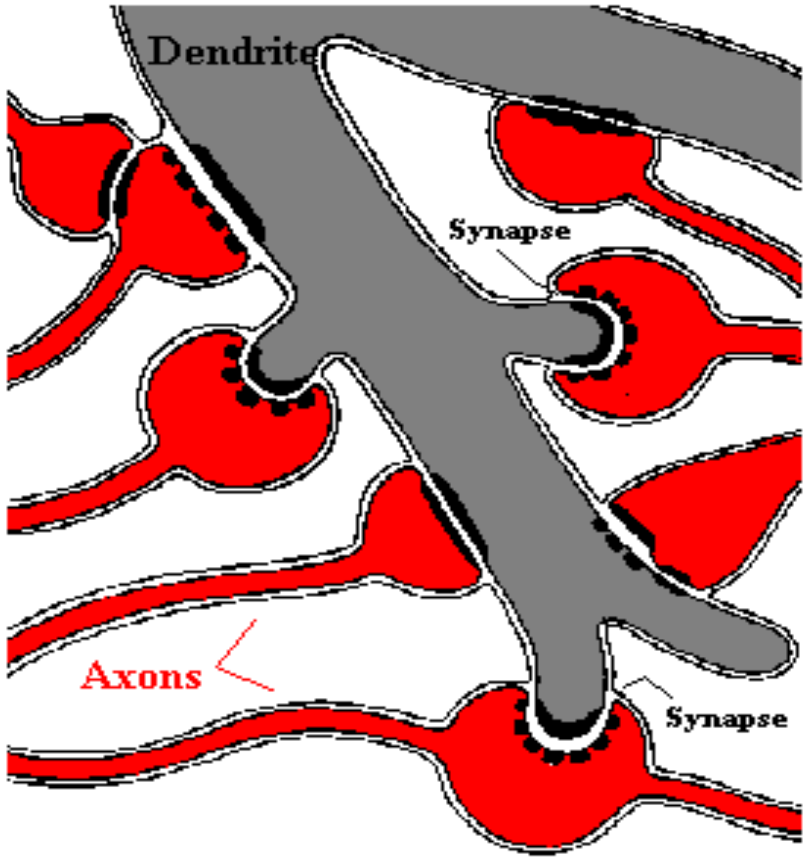
Similarity with Biological Network

- Fundamental processing element of a neural network is a neuron
- A human brain has 100 billion neurons
- An ant brain has 250,000 neurons



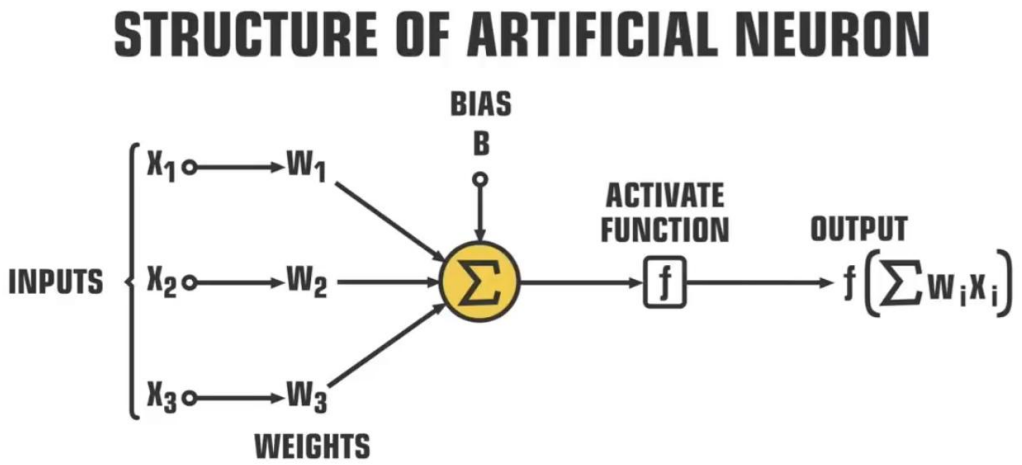
Synapses,
the basis of learning and memory



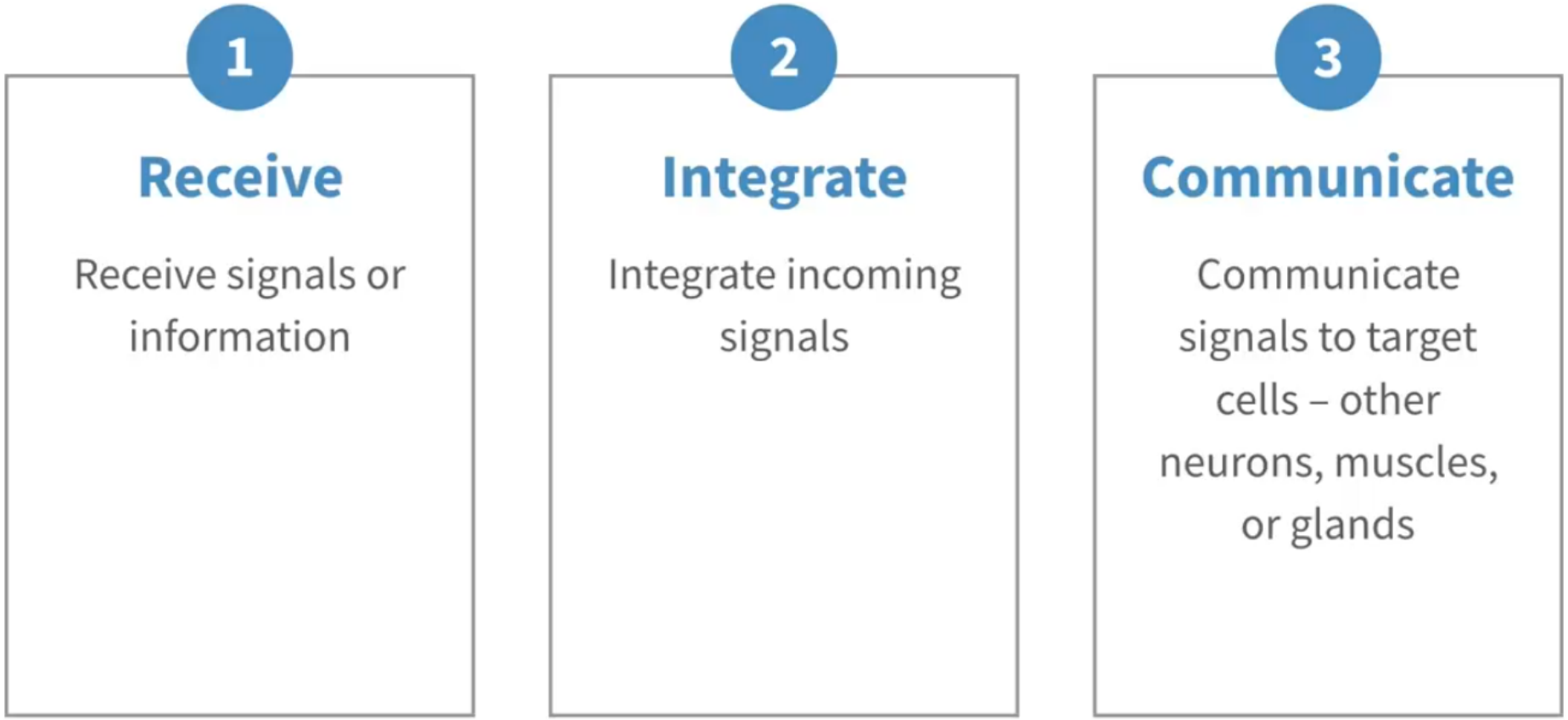


Artificial Neural Network

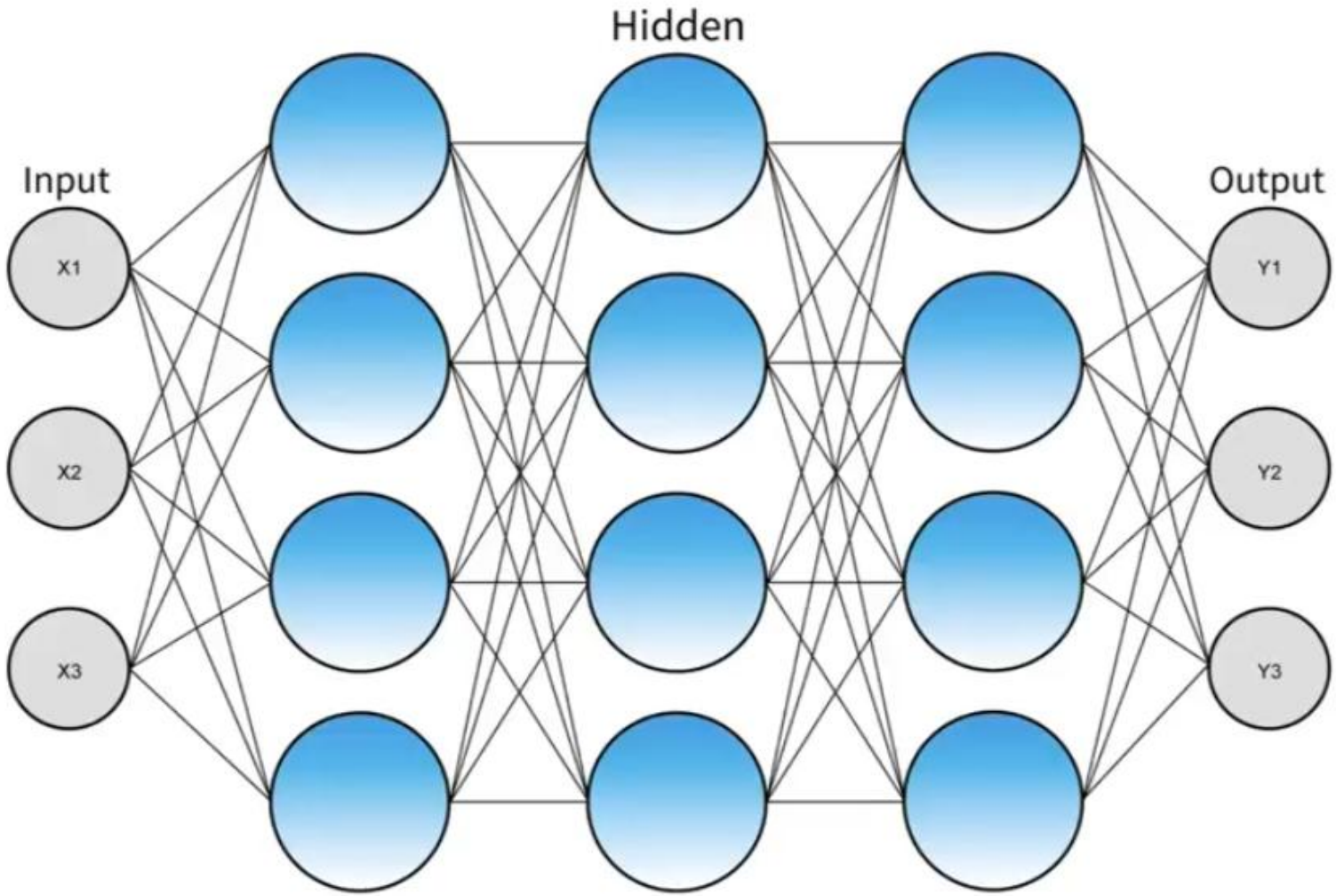
- **Neural Network** is a set of connected
- **INPUT/OUTPUT UNITS**, where each connection has a WEIGHT associated with it.
- Neural Network learning is also called **CONNECTIONIST** learning due to the connections between units.
- It is a case of SUPERVISED, INDUCTIVE or CLASSIFICATION learning.



Artificial Neural Network



Artificial Neural Network



Artificial Neural Network

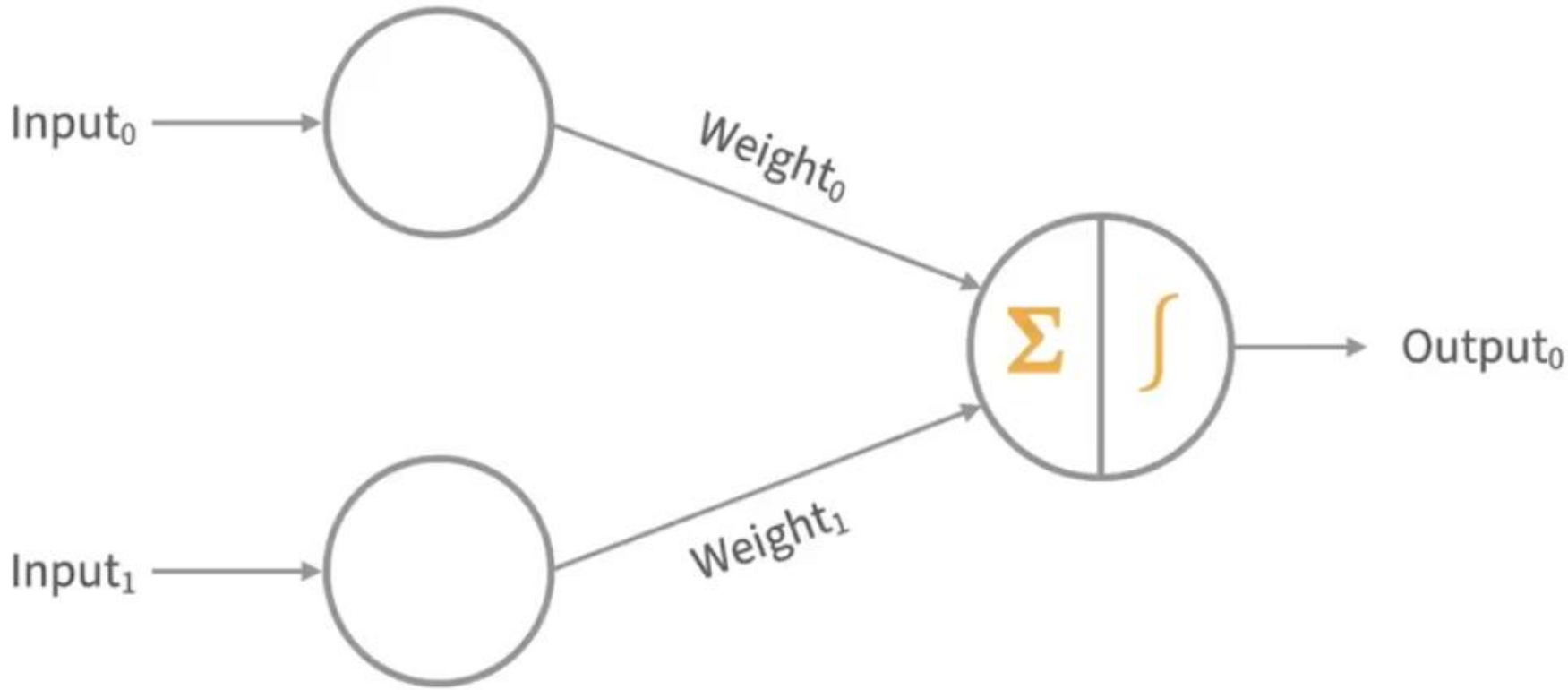
- Artificial Neural Network learns by adjusting the weights so as to be able to correctly classify the training data and hence, after testing phase, to classify unknown data.

- Neural Network needs long time for training.

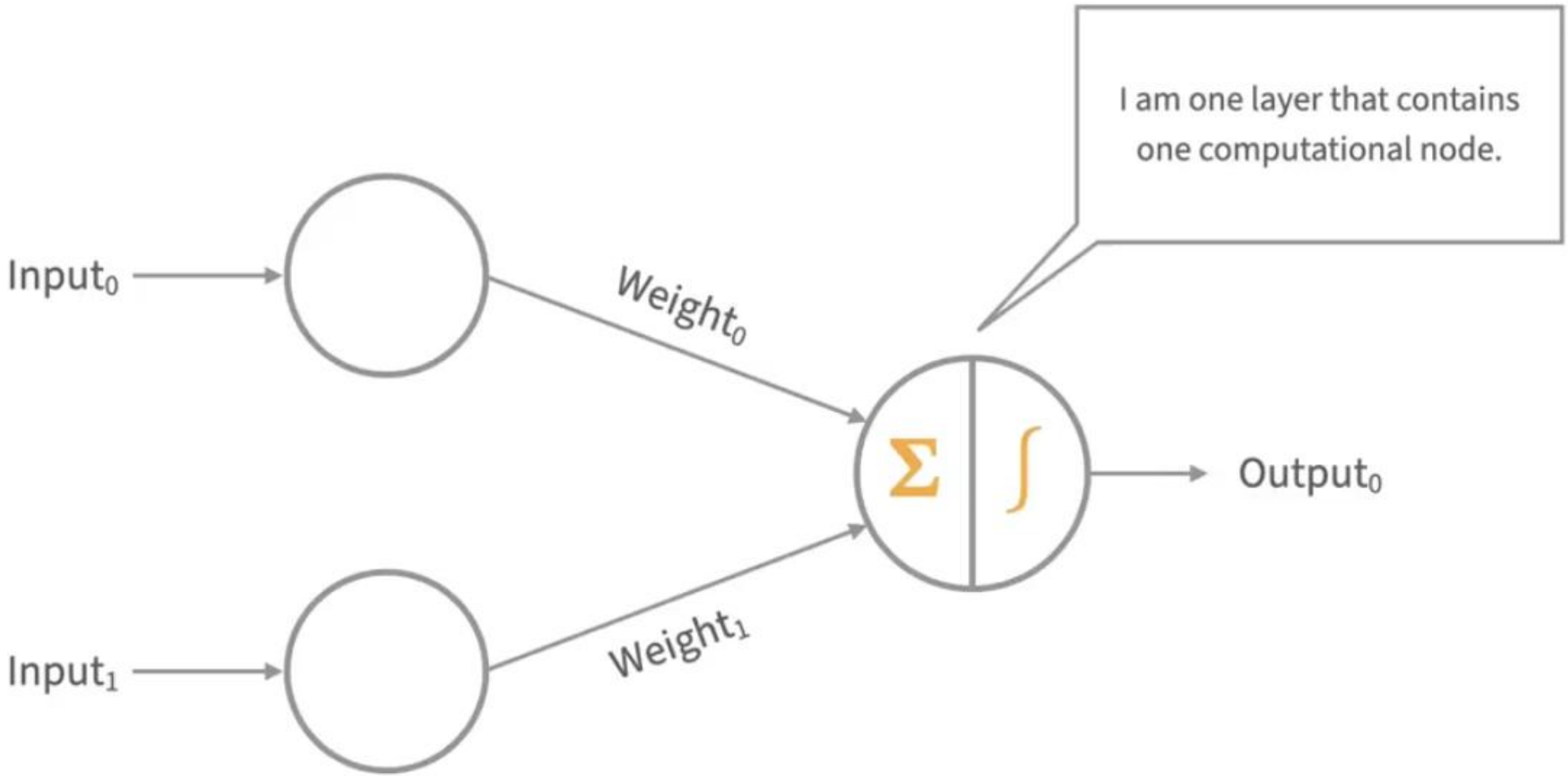
- Neural Network has a high tolerance to noisy and incomplete data

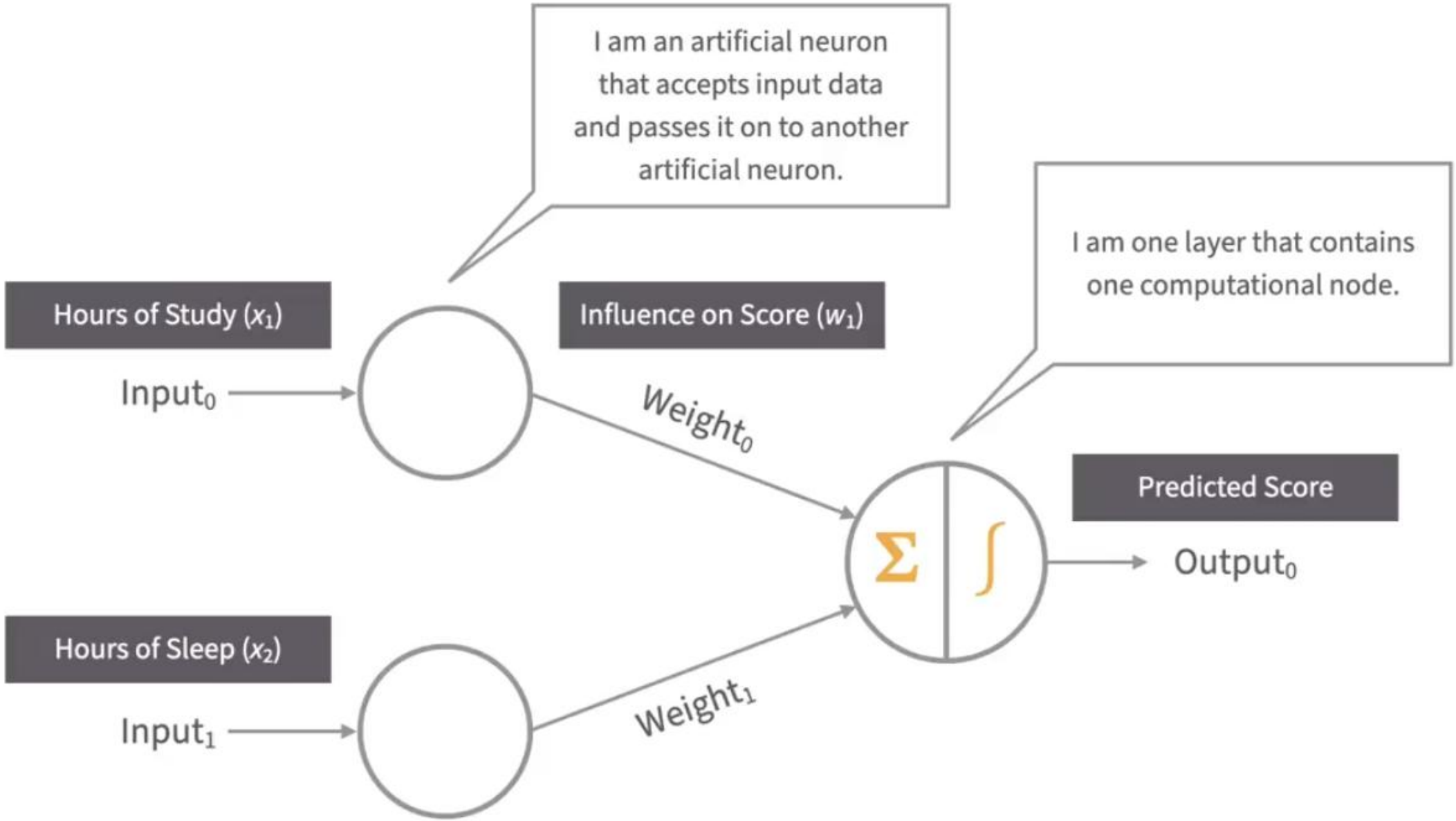
What Is a Single-Layer Perceptron (SLP)?

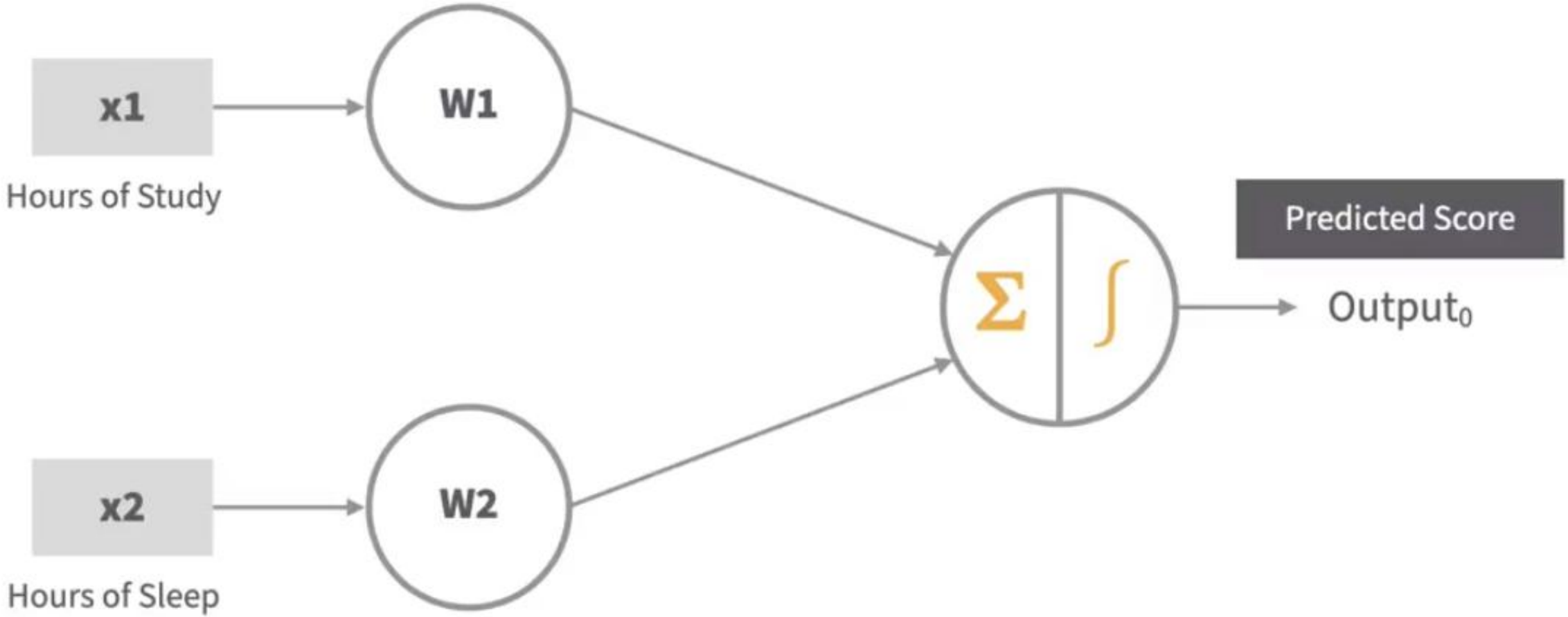
A single-layer perceptron is a type of neuron having multiple inputs and one output.



Why Is It Called "Single Layer"?




I am an artificial neuron that accepts input data and passes it on to another artificial neuron.
I am one layer that contains one computational node.
Hours of Study (x_1)
Influence on Score (w_1)
$Input_0$
$Weight_0$
Predicted Score
$Output_0$
Hours of Sleep (x_2)
$Input_1$
$Weight_1$

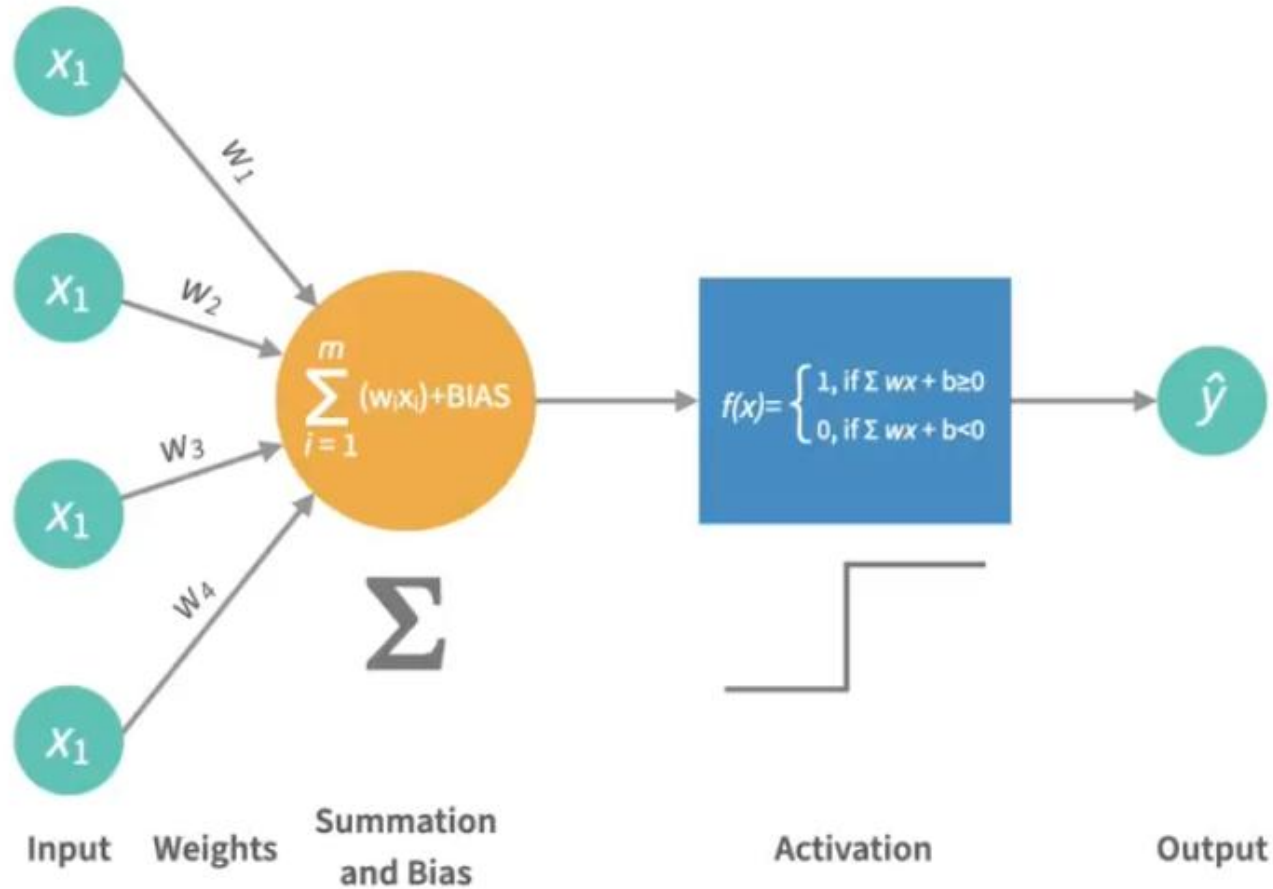

x1
Hours of Study
W1
x2
Hours of Sleep
W2
Σ
∫
Predicted Score
Output0

Perceptron Consists of Four Parts

- **One input layer:** takes the initial data for further processing
- **Weights and bias:**

  Weight: strength of unit connections

  Bias: modifies the output and weighted sum to other neuron
- **Net sum:** calculates the total sum
- **Activation function:** calculates a weighted sum, adds bias



1

Adjustment

Neuron weights are adjusted individually as it “learns.”

2

Summation

Modified inputs are summed to a single value.

3

Activation

Calculation is turned into an output signal.

1

Single-Layer Perceptron

A feed-forward network with an activation function – can learn only linearly separable patterns

2

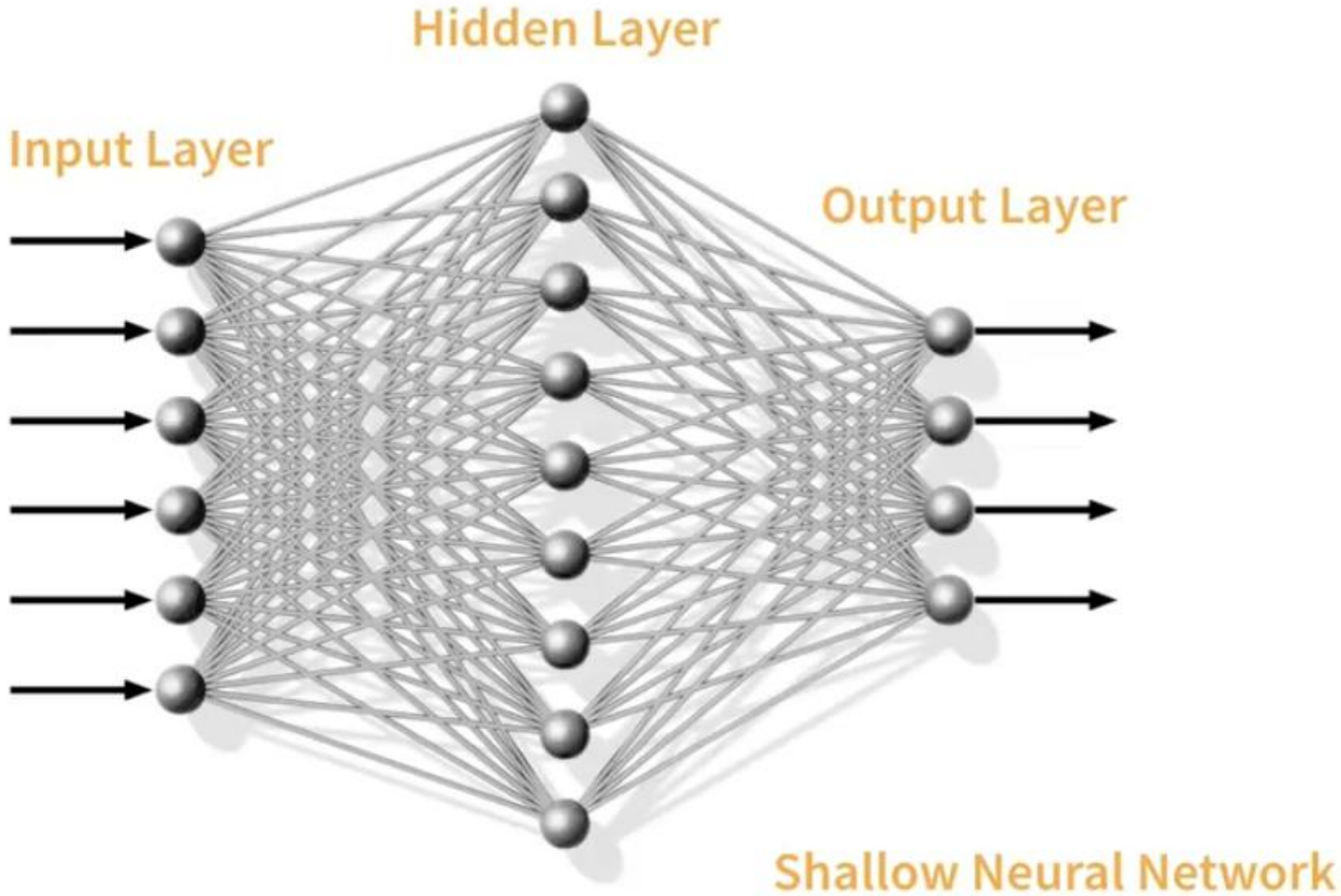
Multilayer Perceptron

Has more hidden layers than SLP; can handle nonlinearity

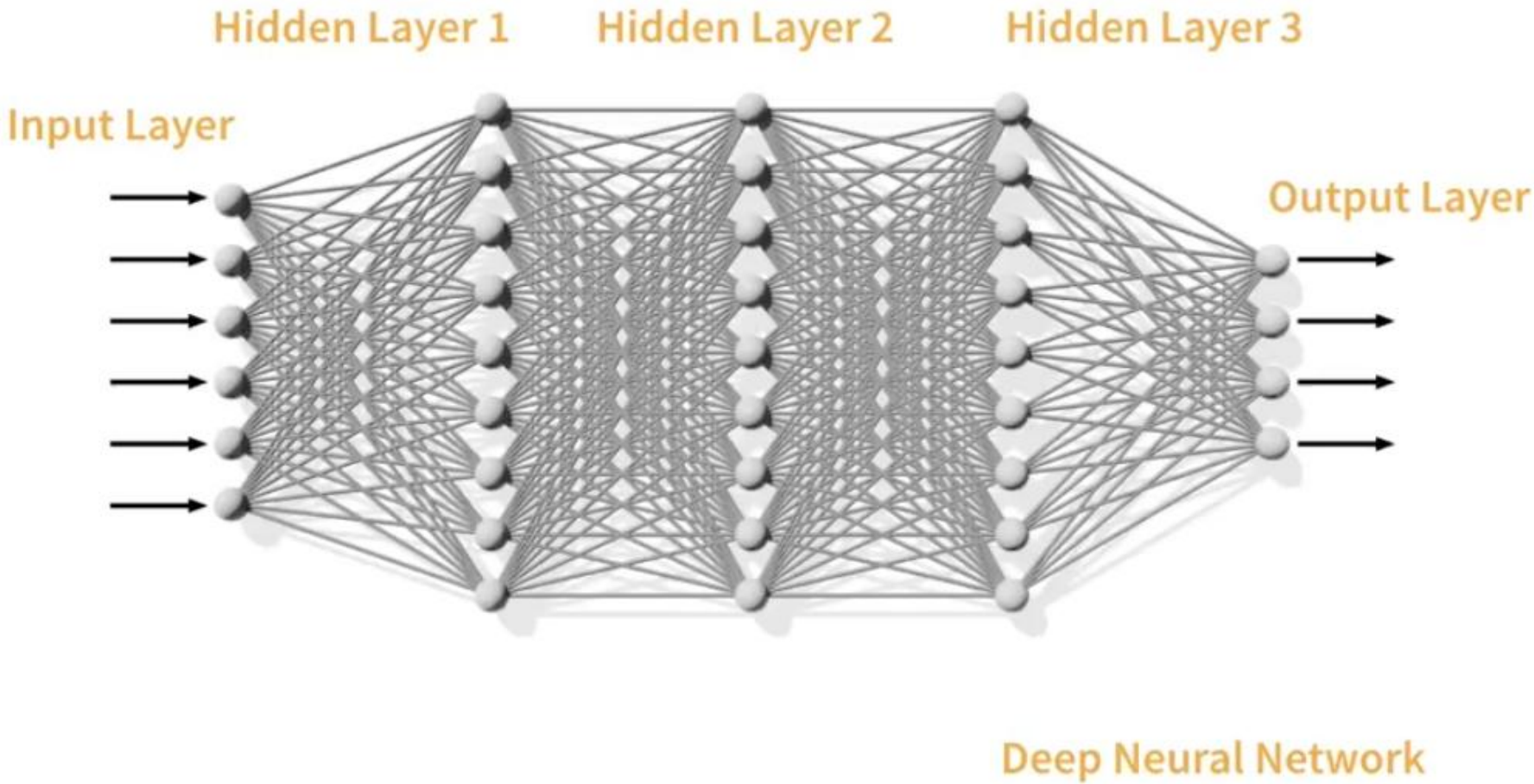
Key Components in Neural Networks

- Multilayer perceptrons
- Layers (input, hidden, output)
- Activation functions
- How neural networks learn

Shallow Neural Network



Deep Neural Network



Predict the selling price of a house in Boston.

Large Complex Data a Better Fit for a Neural Network

id	size (x1)	price (y)	bed(x2)	bath(x3)	floors(x4)	base(x5)	yr(x6)	zipcode(x7)	lat(x8)	long(x9)	sqft_living15(x10)	More
7129300520	1180	221900	3	1	1	Y	1955	98178	47.5112	-122.257	1340	▪▪▪
6414100192	2570	538000	3	2.25	2	N	1951	98125	47.721	-122.319	1690	▪▪▪
5631500400	770	180000	2	1	1	Y	1933	98028	47.7379	-122.233	2720	▪▪▪
2487200875	1960	604000	4	3	1	Y	1965	98136	47.5208	-122.393	1360	▪▪▪
1954400510	1680	510000	3	2	1	Y	1987	98074	47.6168	-122.045	1800	▪▪▪
7237550310	5420	1225000	4	4.5	1	Y	2001	98053	47.6561	-122.005	4760	▪▪▪
1321400060	1715	257500	3	2.25	2	N	1995	98003	47.3097	-122.327	2238	▪▪▪
2008000270	1060	291850	3	1.5	1	N	1963	98198	47.4095	-122.315	1650	▪▪▪
2414600126	1780	229500	3	1	1	Y	1960	98146	47.5123	-122.337	1780	▪▪▪
3793500160	1890	323000	3	2.5	2	Y	2003	98038	47.3684	-122.031	2390	▪▪▪
1736800520	3560	662500	3	2.5	1	N	1965	98007	47.6007	-122.145	2210	▪▪▪
114101516	▪▪▪▪▪▪	▪▪▪▪▪▪	3	1	1.5	N	1927	98028	47.7558	-122.229	1780	▪▪▪

Multilayer Perceptrons (MLPs)

- Learn linear functions
- Learn nonlinear functions
- Regression and classification problems

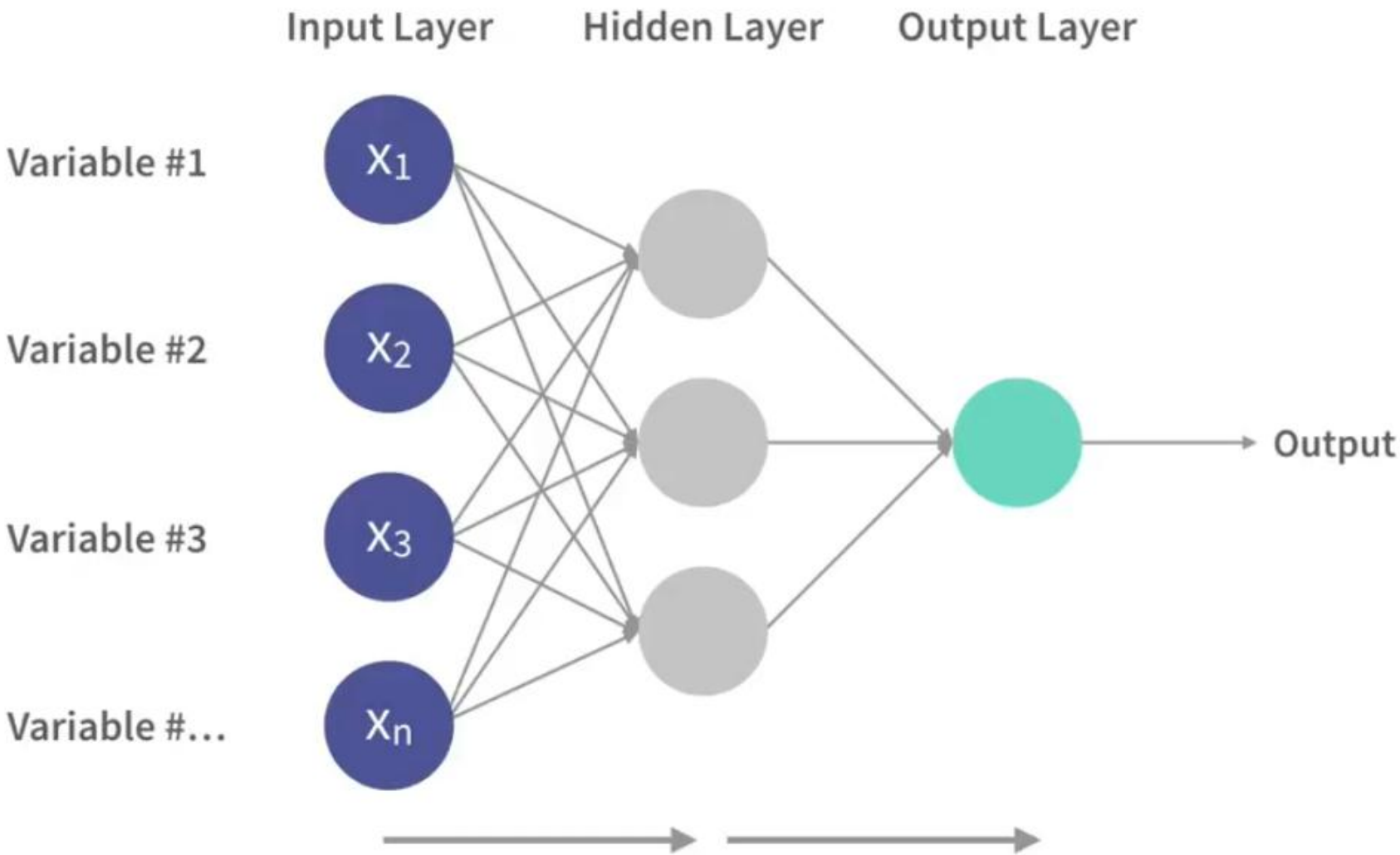
Regression problem – predict selling price

Classification problem – classify whether it has a garage

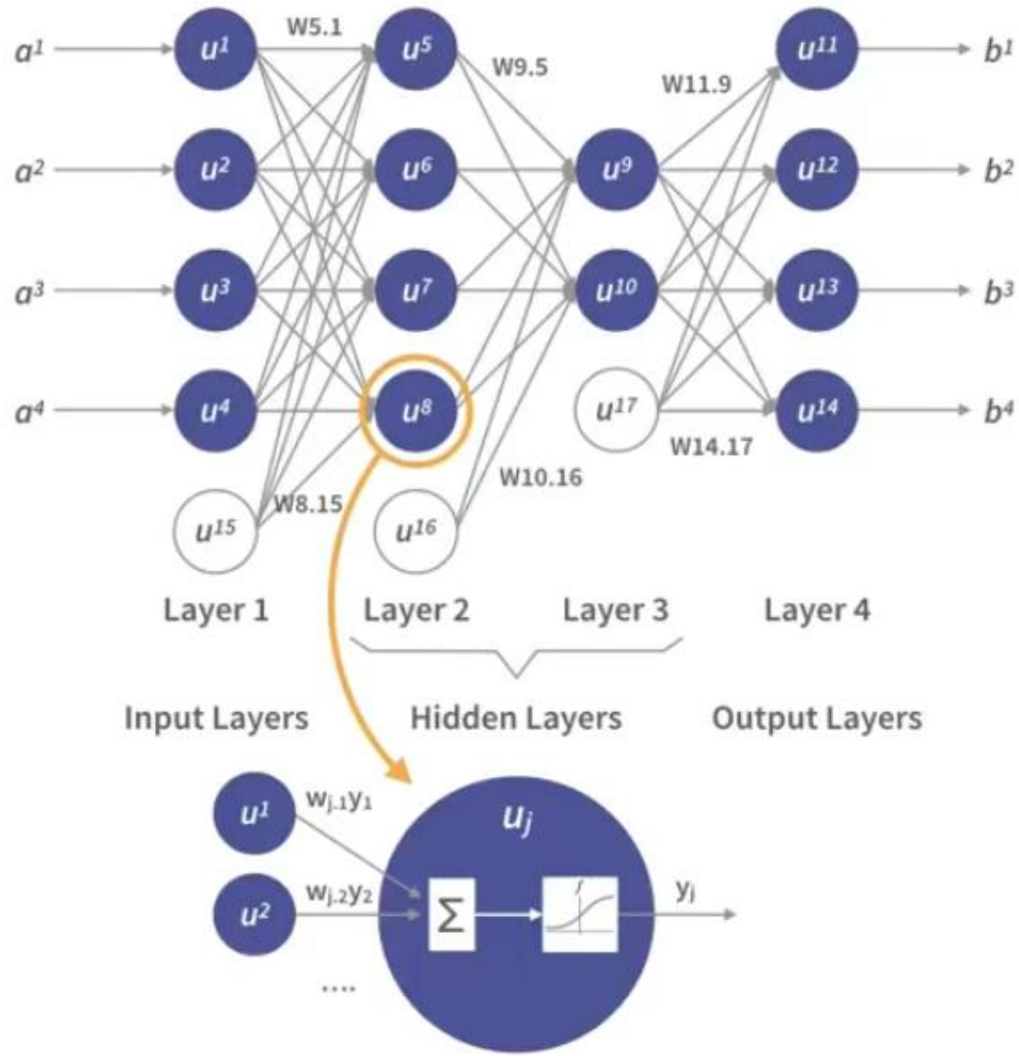
MLP Characteristics

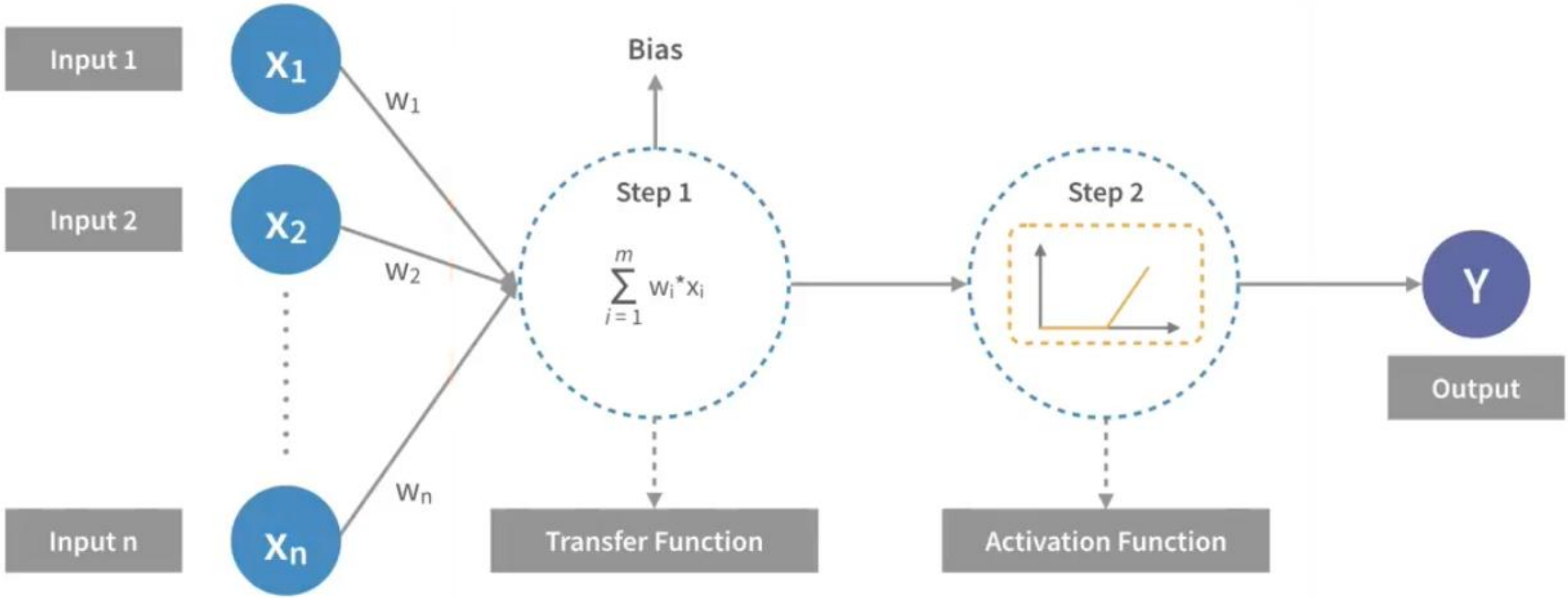
- Feed-forward network
- Multiple layers of perceptrons
- Perceptron in each layer connected to perceptrons
- Output of each perceptron used as input to next perceptron

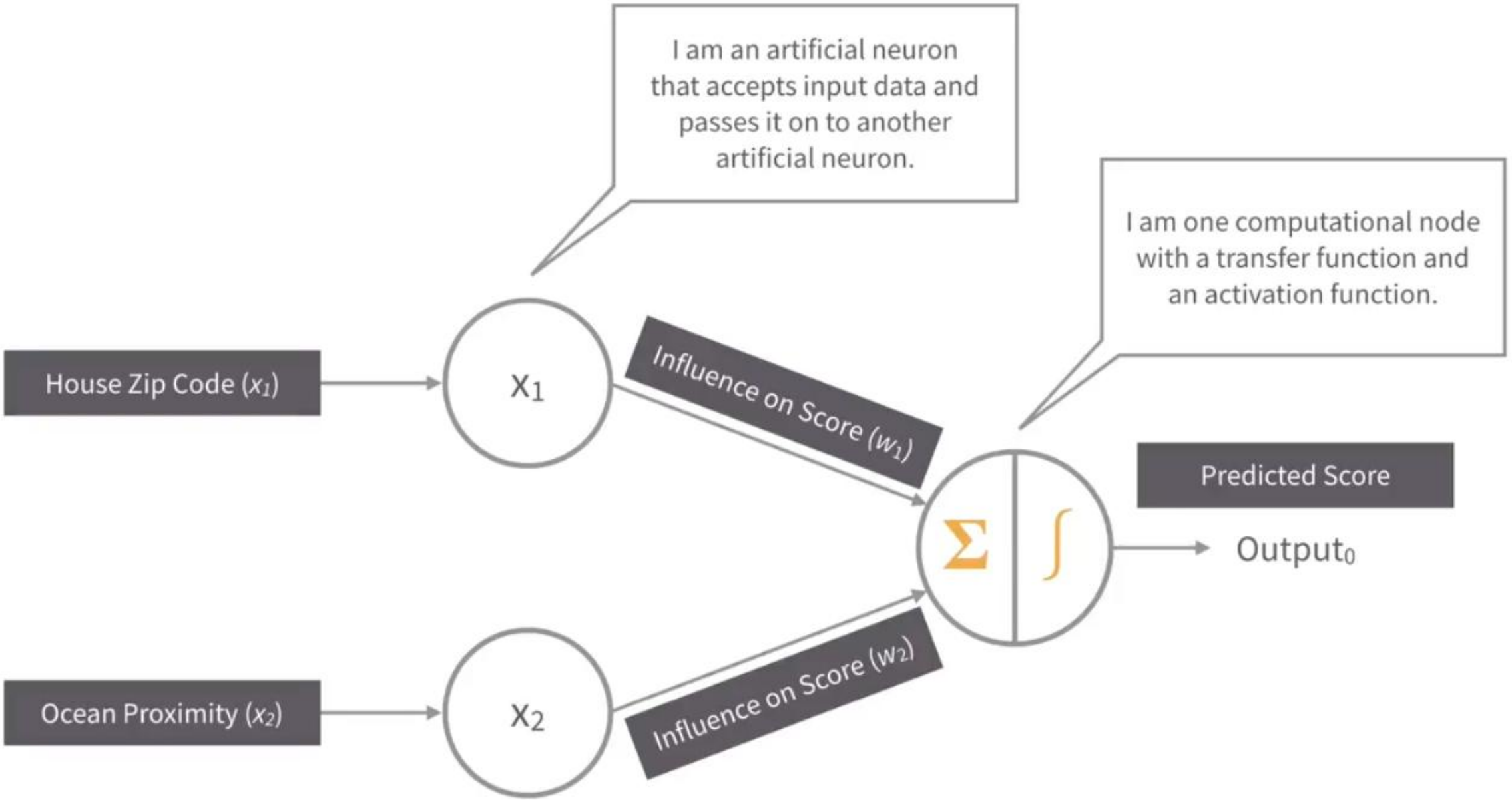
Fed Forward and Fully Connected

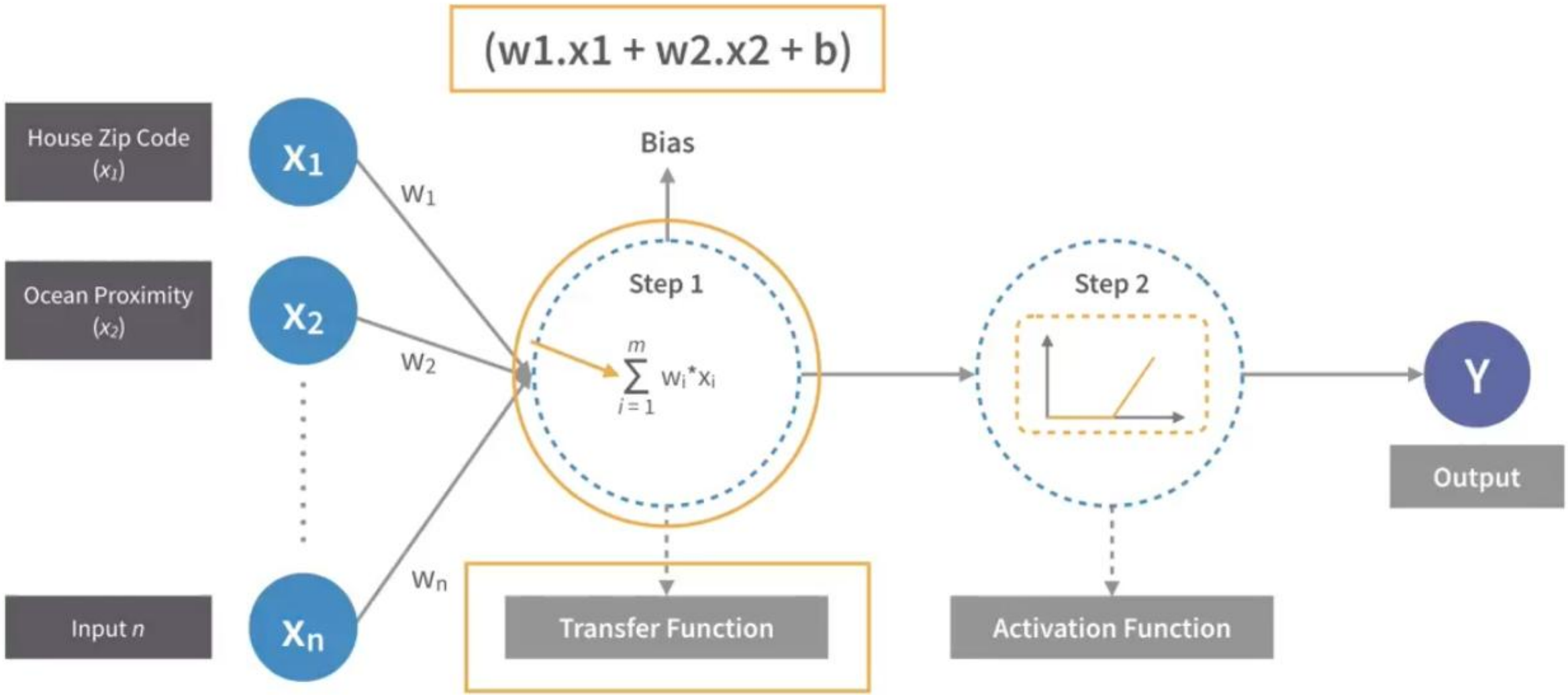


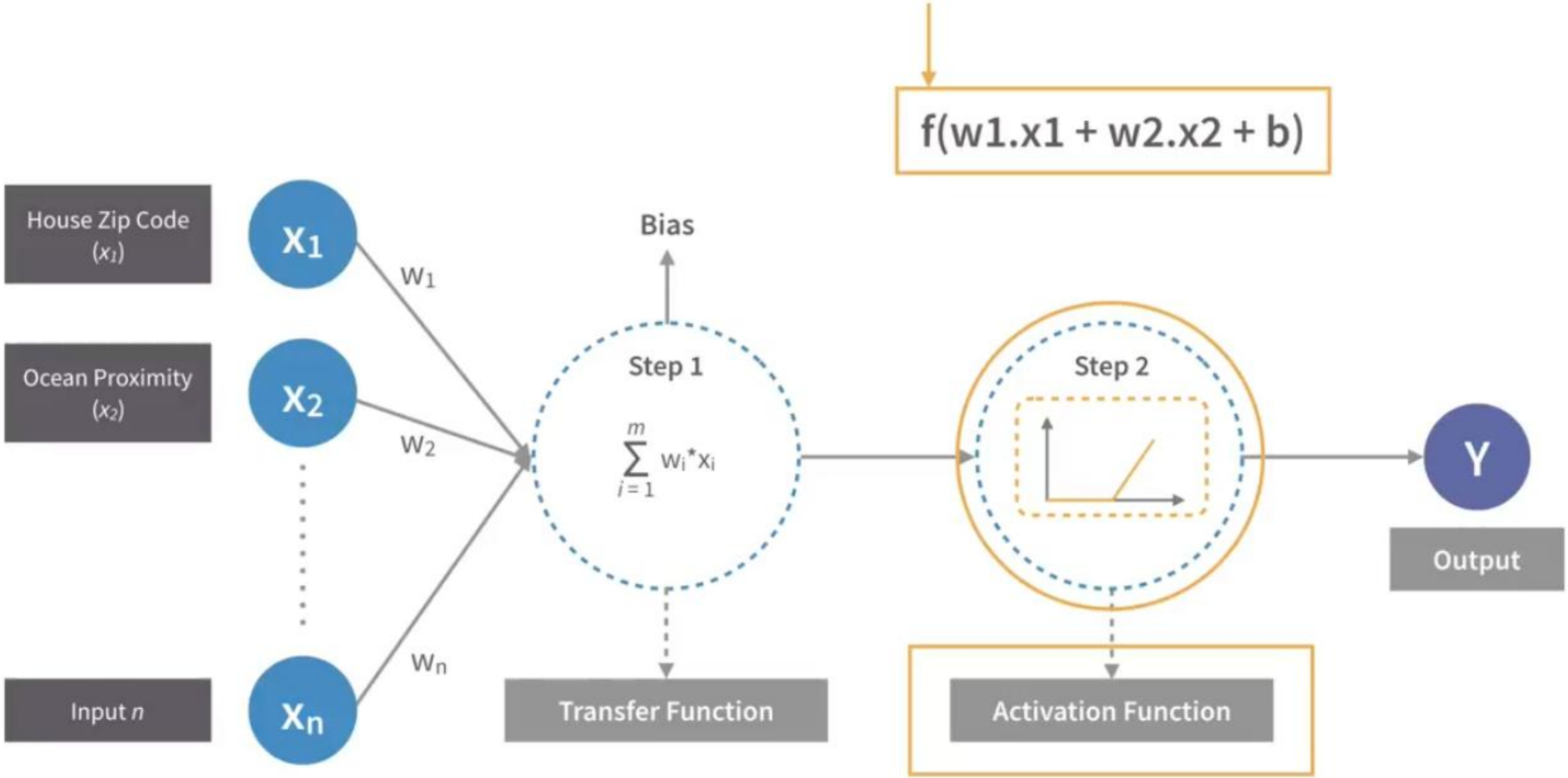
Inside a Hidden Layer Node




Input 1
Input 2
Input n
X1
X2
Xn
W1
W2
Wn
Bias
Step 1
$\sum_{i=1}^{m} w_i * x_i$
Step 2
Y
Output
Transfer Function
Activation Function


I am an artificial neuron that accepts input data and passes it on to another artificial neuron.
I am one computational node with a transfer function and an activation function.
House Zip Code (x_1)
Ocean Proximity (x_2)
x_1
x_2
Influence on Score (w_1)
Influence on Score (w_2)
Σ
∫
Predicted Score
Output$_0$

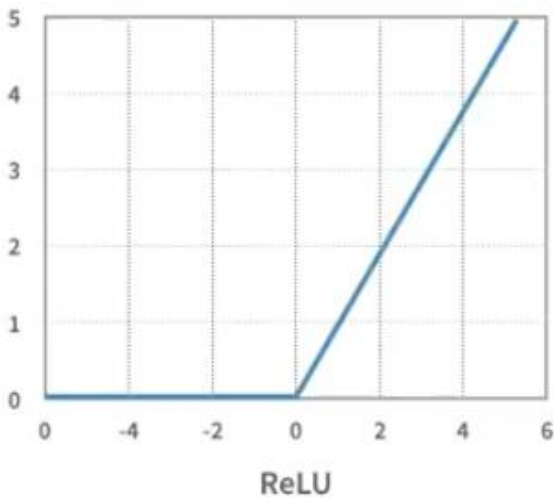



f(w1.x1 + w2.x2 + b)
House Zip Code
(x1)
Ocean Proximity
(x2)
Input n
X1
X2
Xn
W1
W2
Wn
Bias
Step 1
$\sum_{i=1}^{m} w_i * x_i$
Step 2
Y
Output
Transfer Function
Activation Function

1

ReLU

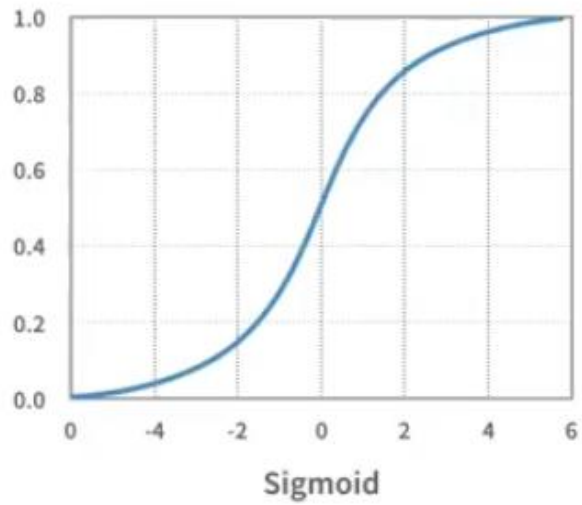
Only positive values



2

Sigmoid

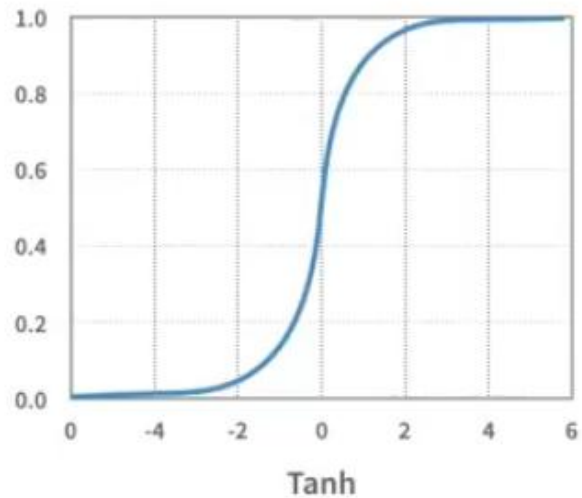
Between 0 and 1



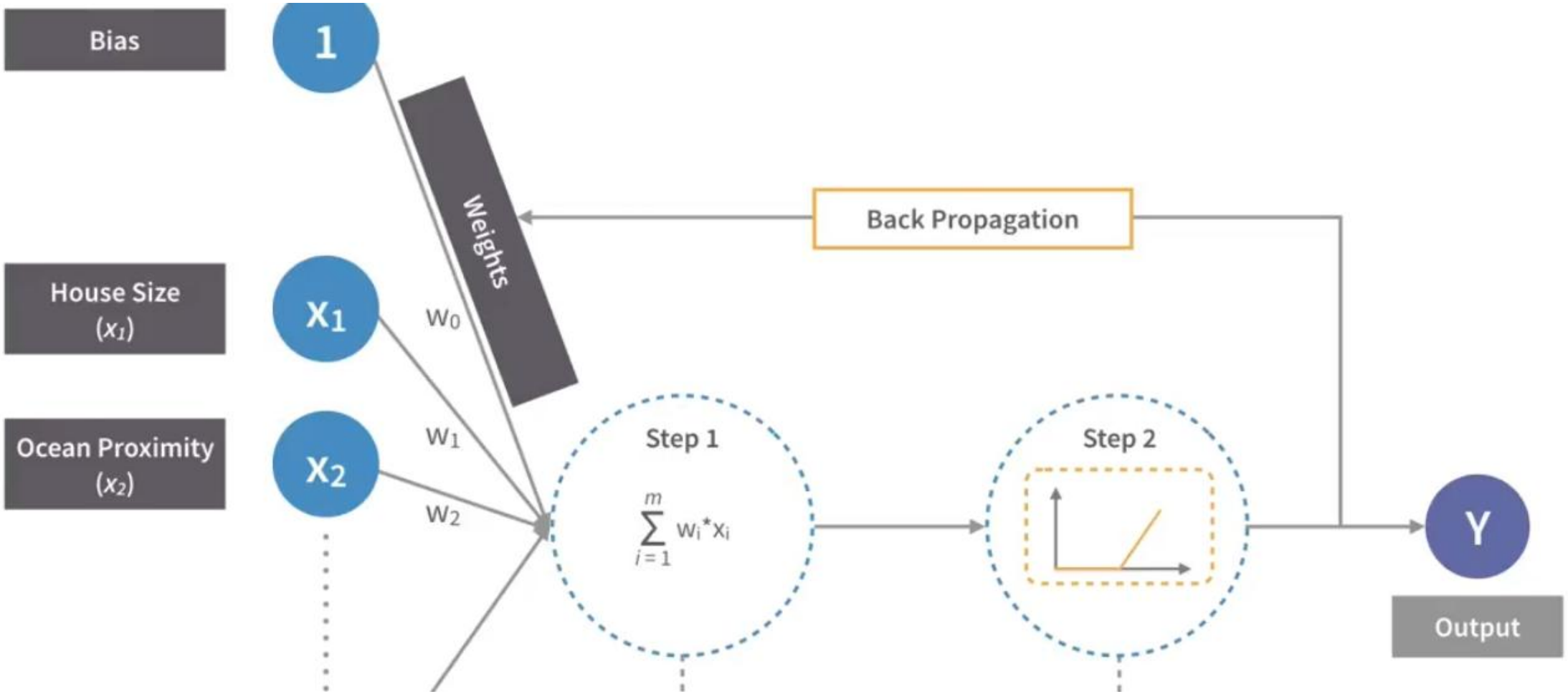
3

Tanh

Between −1 and 1



How Neural network learns

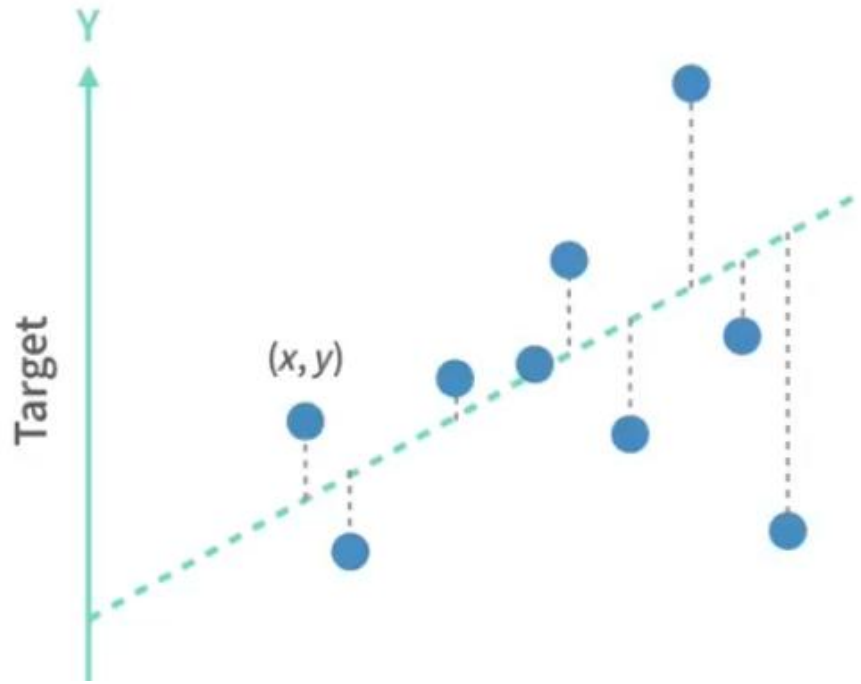


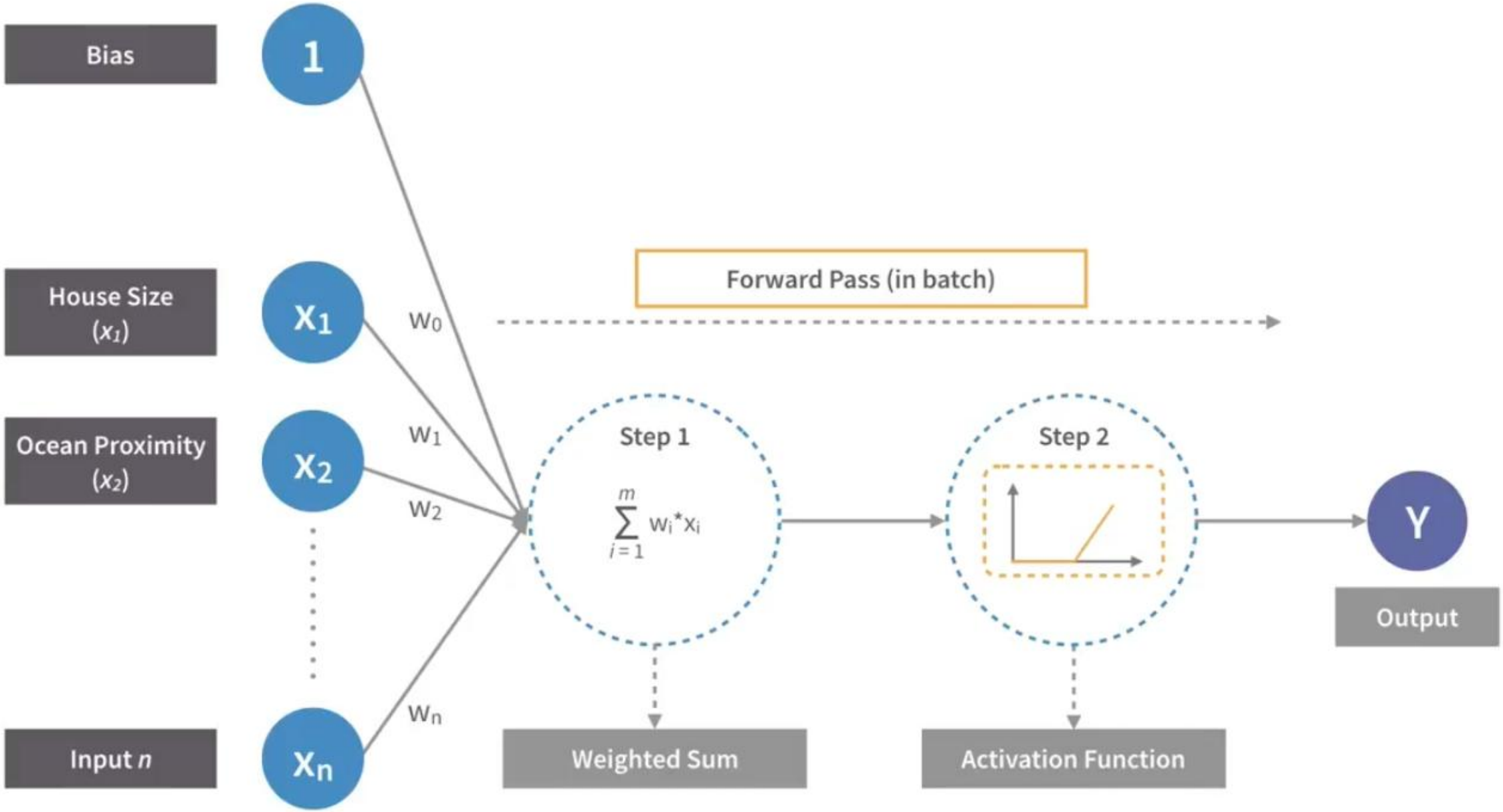
How Neural network learns

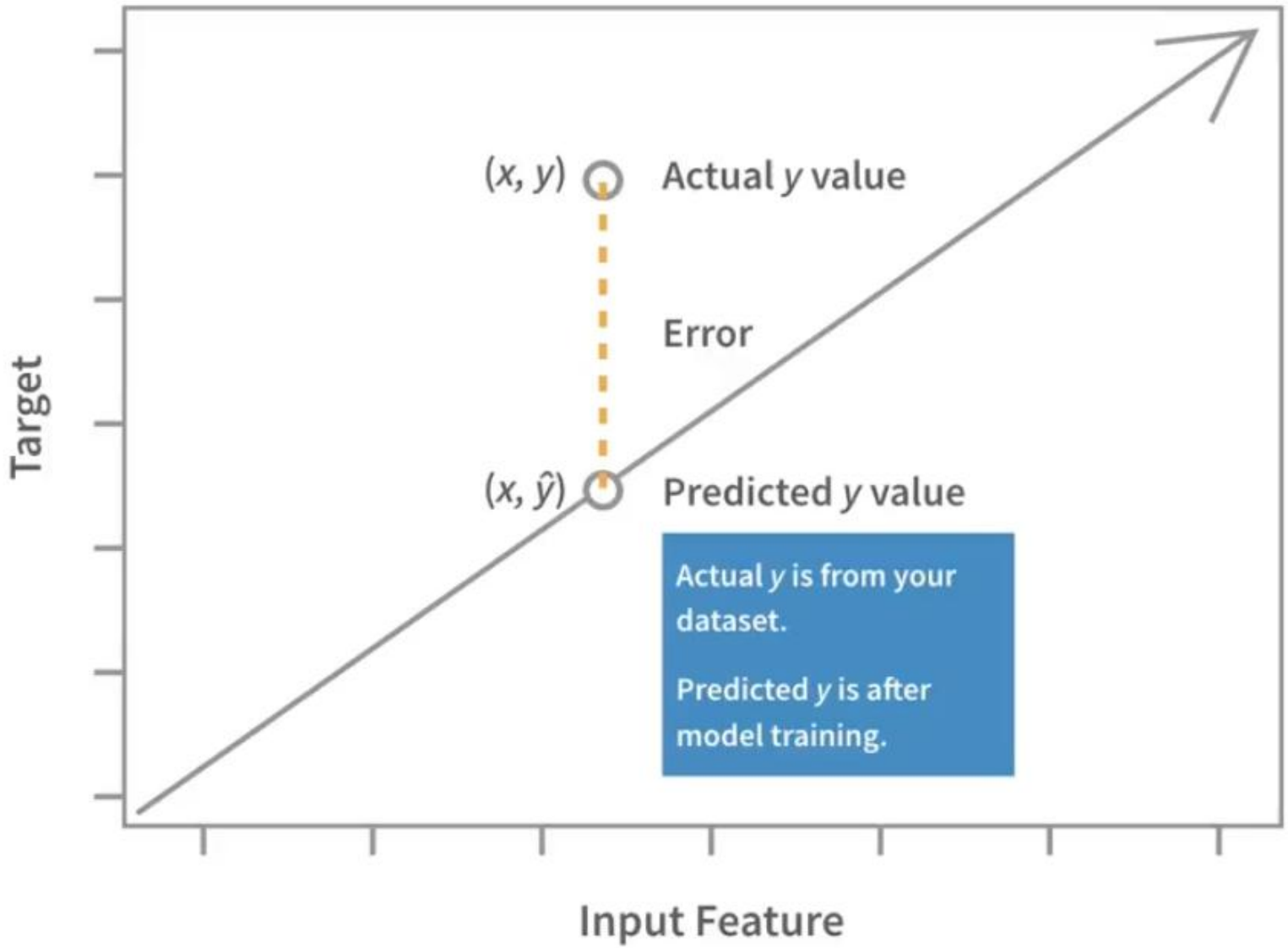
Regression Problem

Example: House Price Prediction

	Size x		Price y
x_1	45	y_1	800
	60		1200
	61		1400
	70		1600
	74		1750
x_n	80	y_n	2100




Bias
1
House Size
(x_1)
X_1
Ocean Proximity
(x_2)
X_2
Input *n*
X_n
W_0
W_1
W_2
W_n
Forward Pass (in batch)
Step 1
$\sum_{i=1}^{m} W_i * X_i$
Step 2
Y
Output
Weighted Sum
Activation Function



Goal: minimize error between actual and predicted value

Root Mean Squared Error (RMSE): Housing Dataset Sample

Input	Actual	Predicted	
x	y	$\hat{y}$	
45	800	848.83	$= (800 - 848.83)^2$
60	1200	1295.03	$+ (1200 - 1297.03)^2$
61	1400	1324.78	$+ (1400 - 1324.78)^2$
70	1600	1592.5	$+ (1600 - 1592.5)^2$
74	1750	1711.48	$+ (1750 - 1711.48)^2$
80	2100	1889.96	$+ (2100 - 1889.96)^2$
90	2000	2187.43	$+ (2000 - 2187.43)^2 = 97858.86$

$$\sqrt{\frac{1}{n} \times \sum_{i=1}^{n} (\hat{y}_i - y_i)^2}$$

Cost Function (Loss): Root Mean Square Error

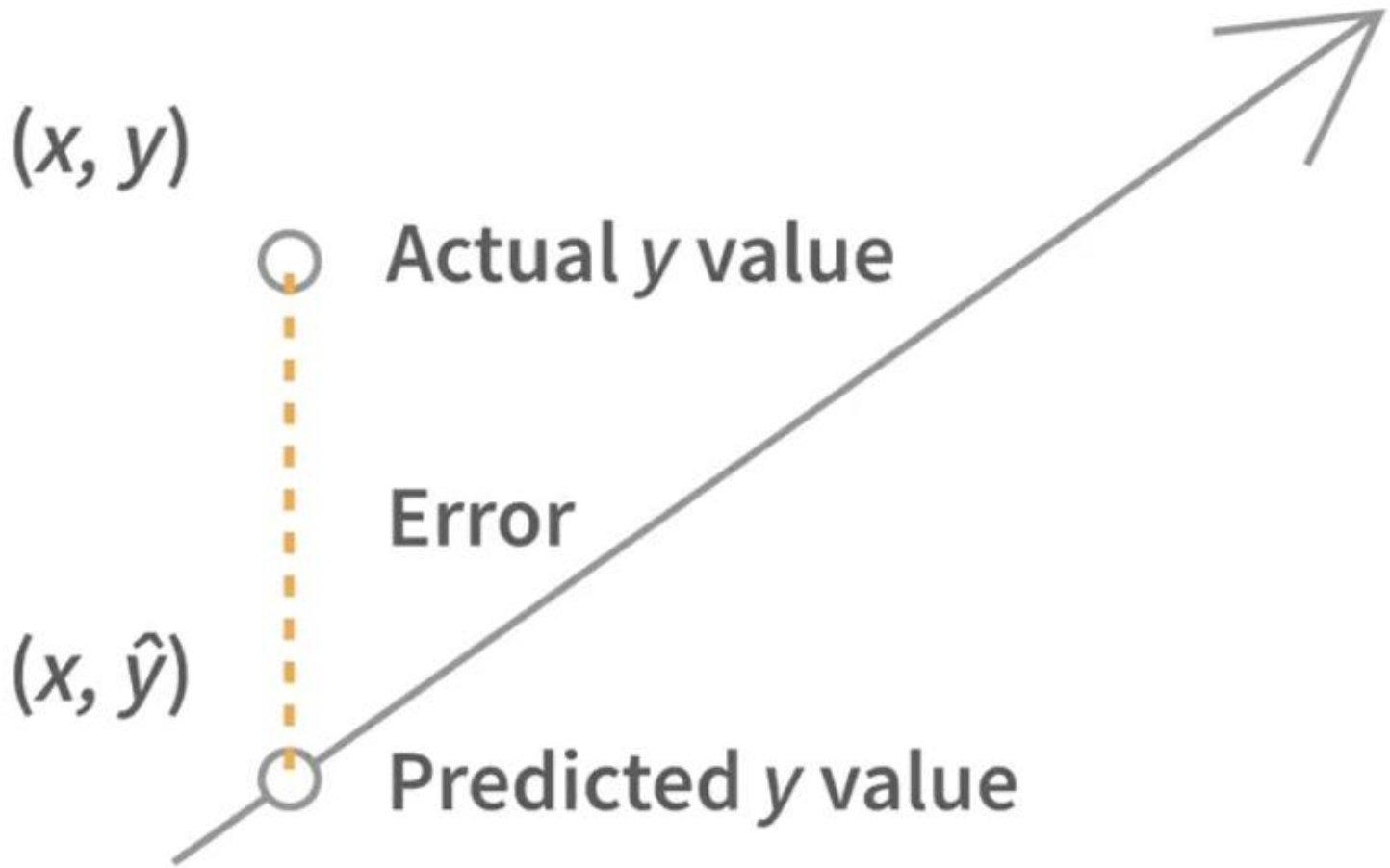
1. Get the errors for the training examples.
2. Compute the squares of the error values.
3. Compute the mean of the squared error values.
4. Take the square root of the mean.

+0.70	0.49
+1.10	1.21
+0.65	0.42
-1.20	1.44
-1.15	1.32
+1.10	1.21
+3.09	9.55
-2.10	4.41

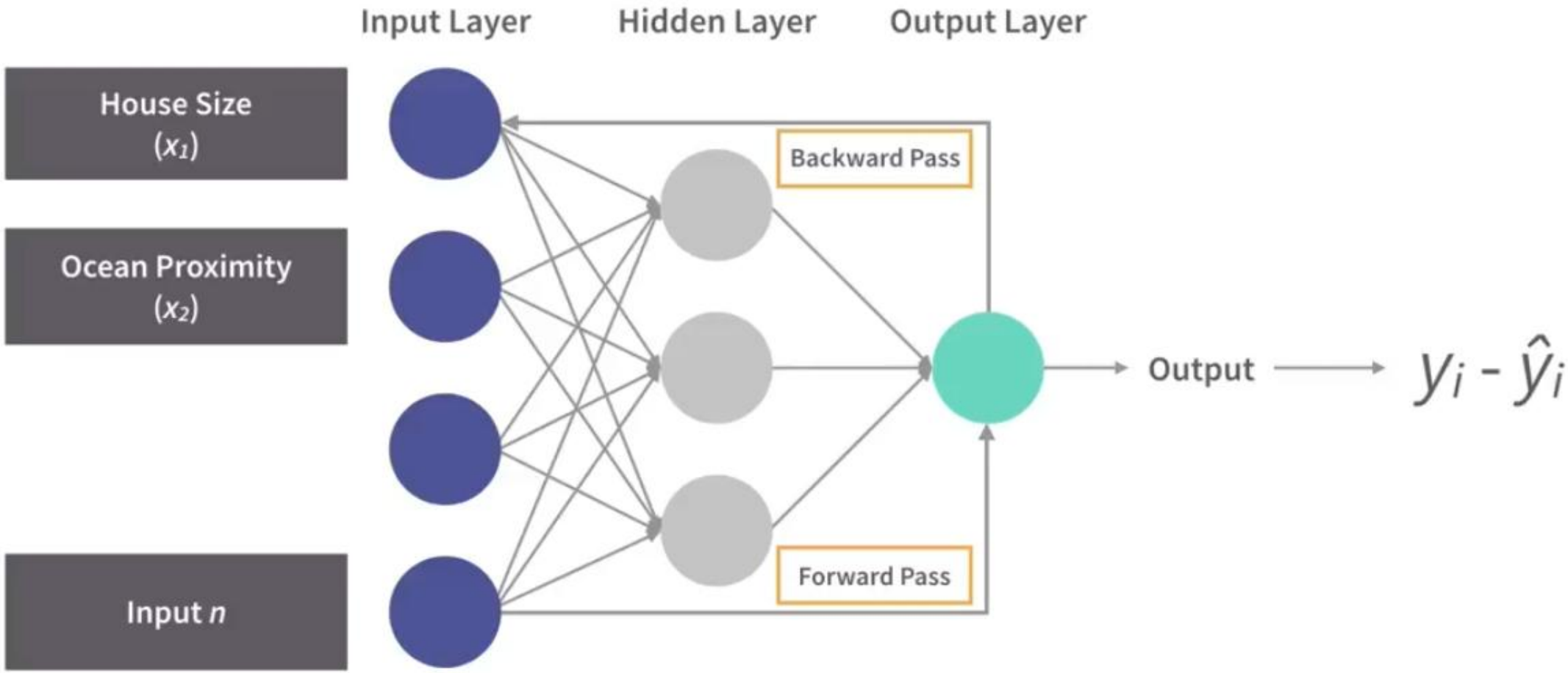
$$2.51$$

$$\sqrt{\frac{1}{n} \times \sum_{i=1}^{n} (\hat{y}_i - y_i)^2}$$

Goal: decrease distance to the line



Back Propagation



Types of Neural Networks

Types of Neural Networks

- Convolutional neural networks (CNNs)
- Recurrent neural networks (RNNs)
- Transformer neural networks