

# What is Deep Learning?

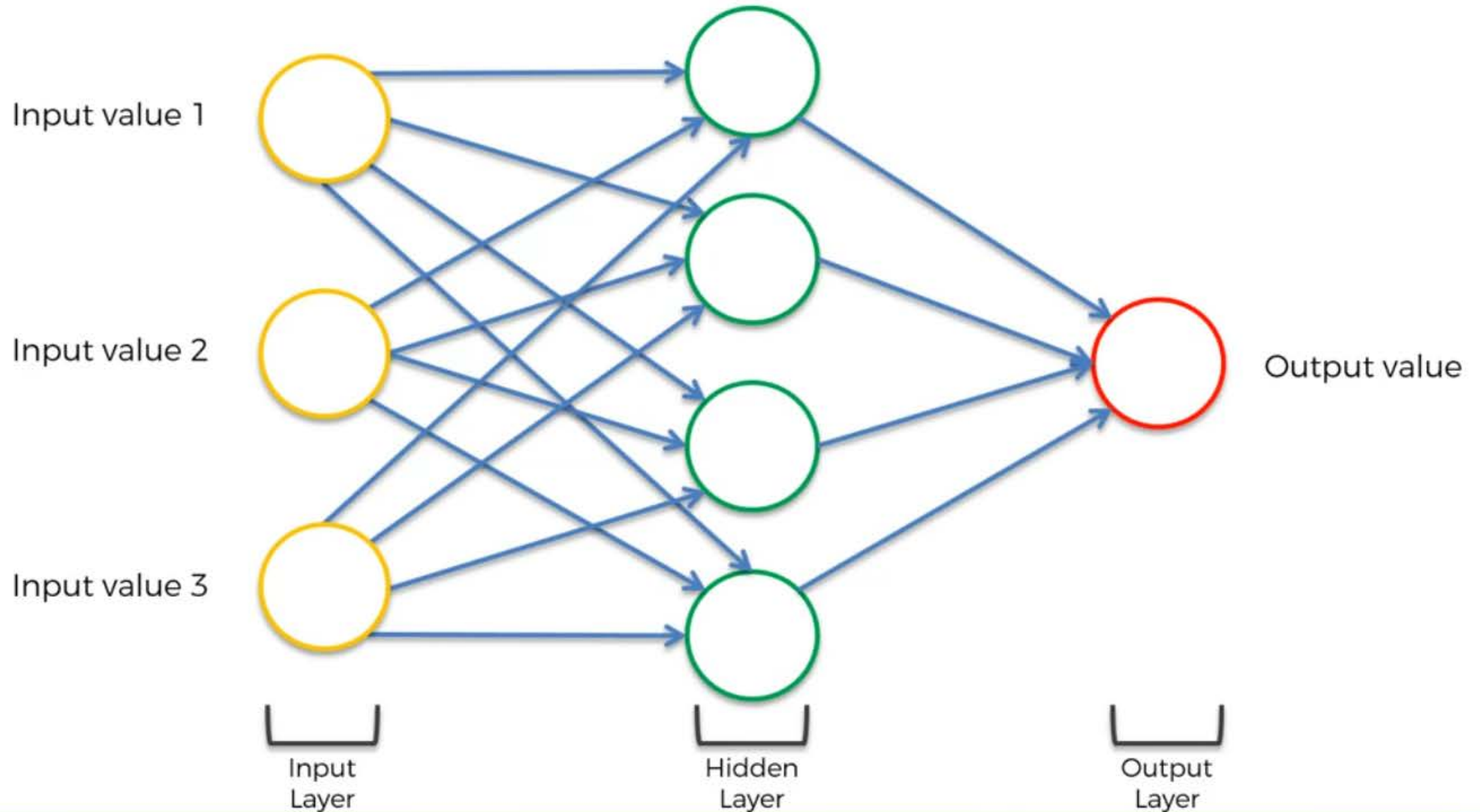
# What is Deep Learning?

---



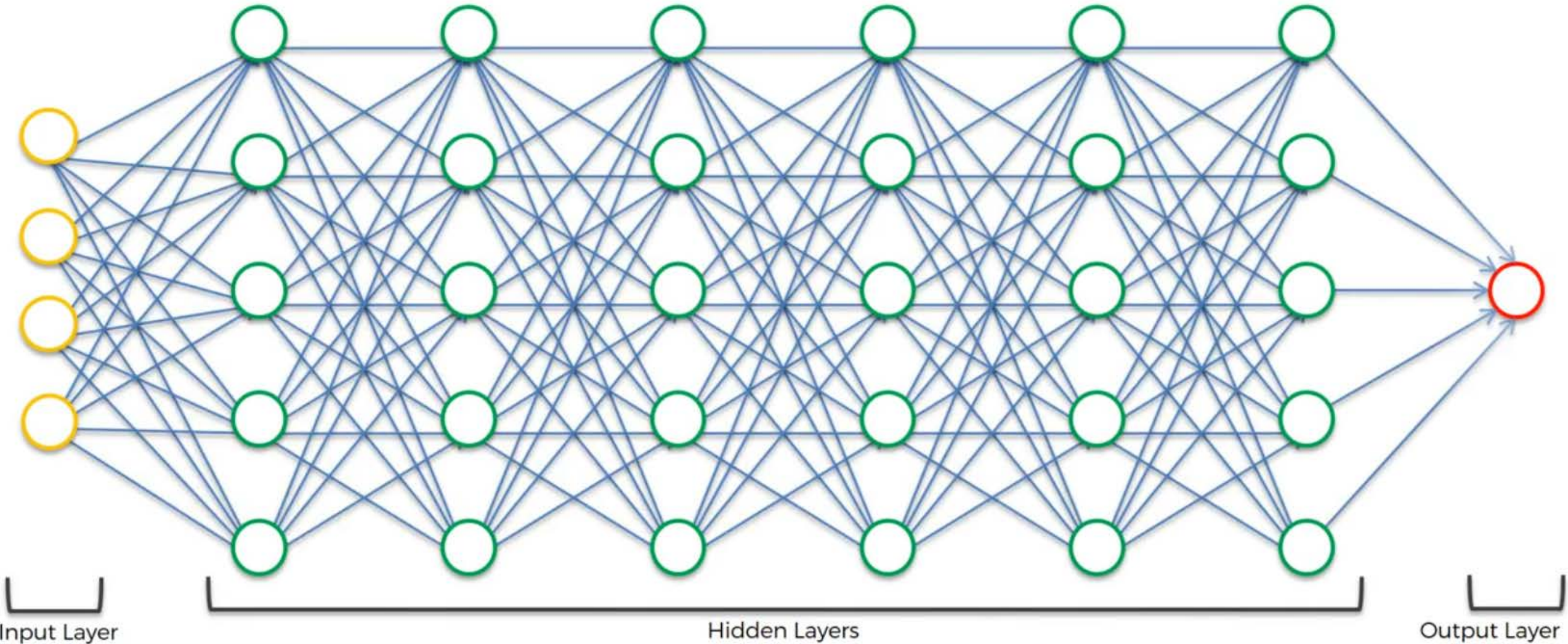
**Geoffrey Hinton**

# What is Deep Learning?





# What is Deep Learning?



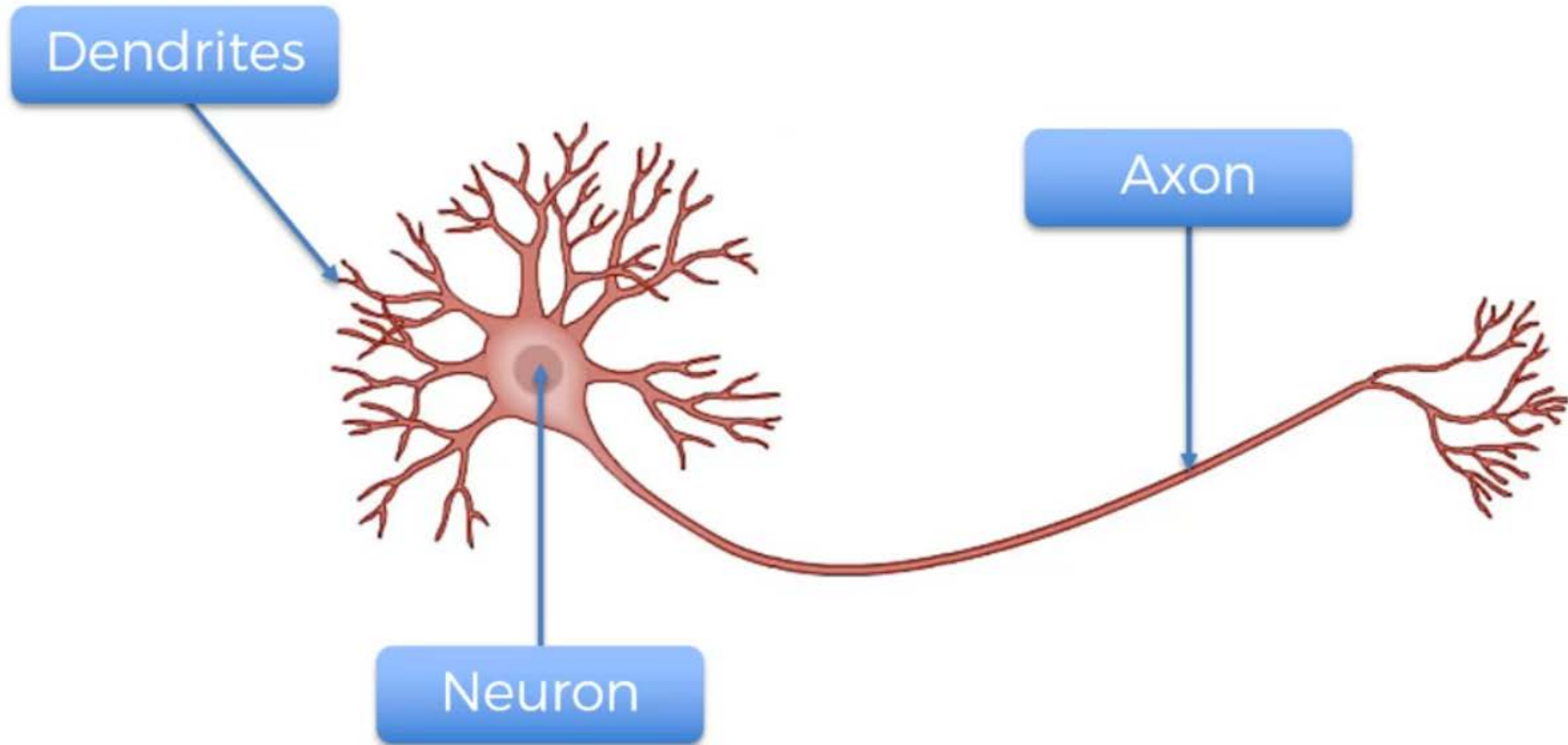
# Plan of Attack

---

What we will learn in this section:

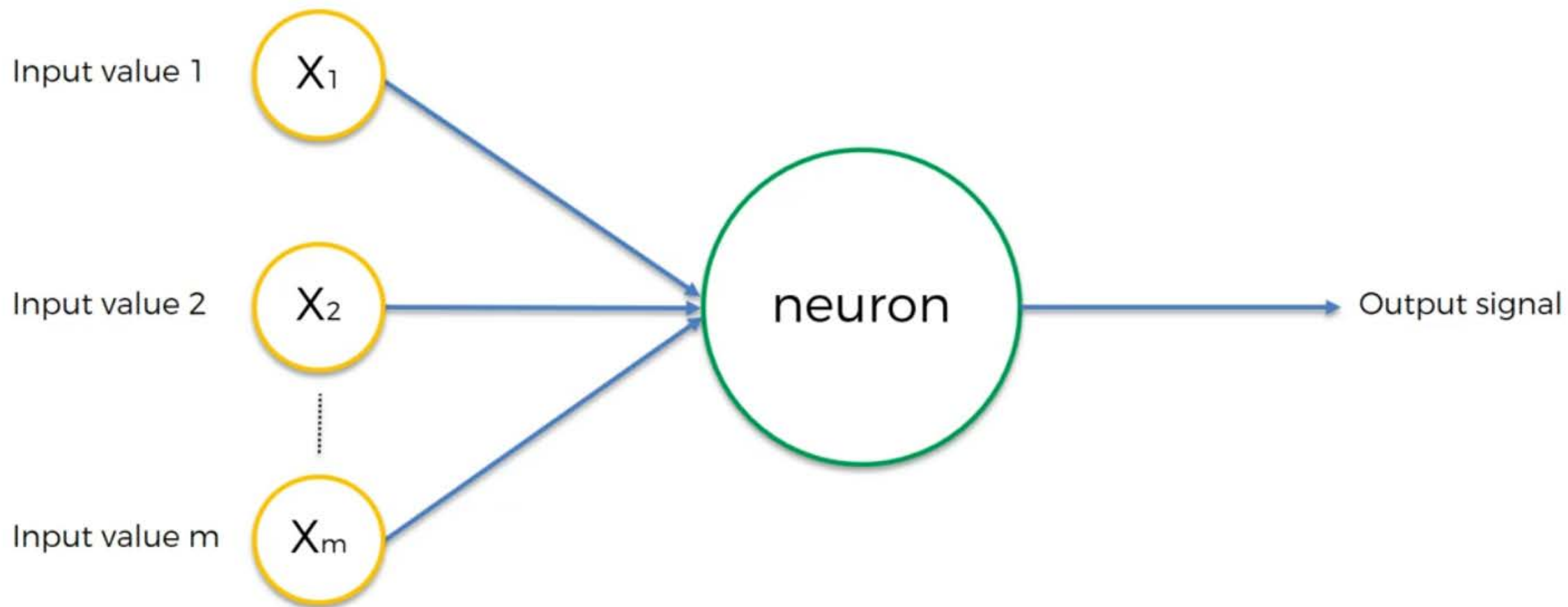
- The Neuron
- The Activation Function
- How do Neural Networks work? (example)
- How do Neural Networks learn?
- Gradient Descent
- Stochastic Gradient Descent
- Backpropagation

# The Neuron

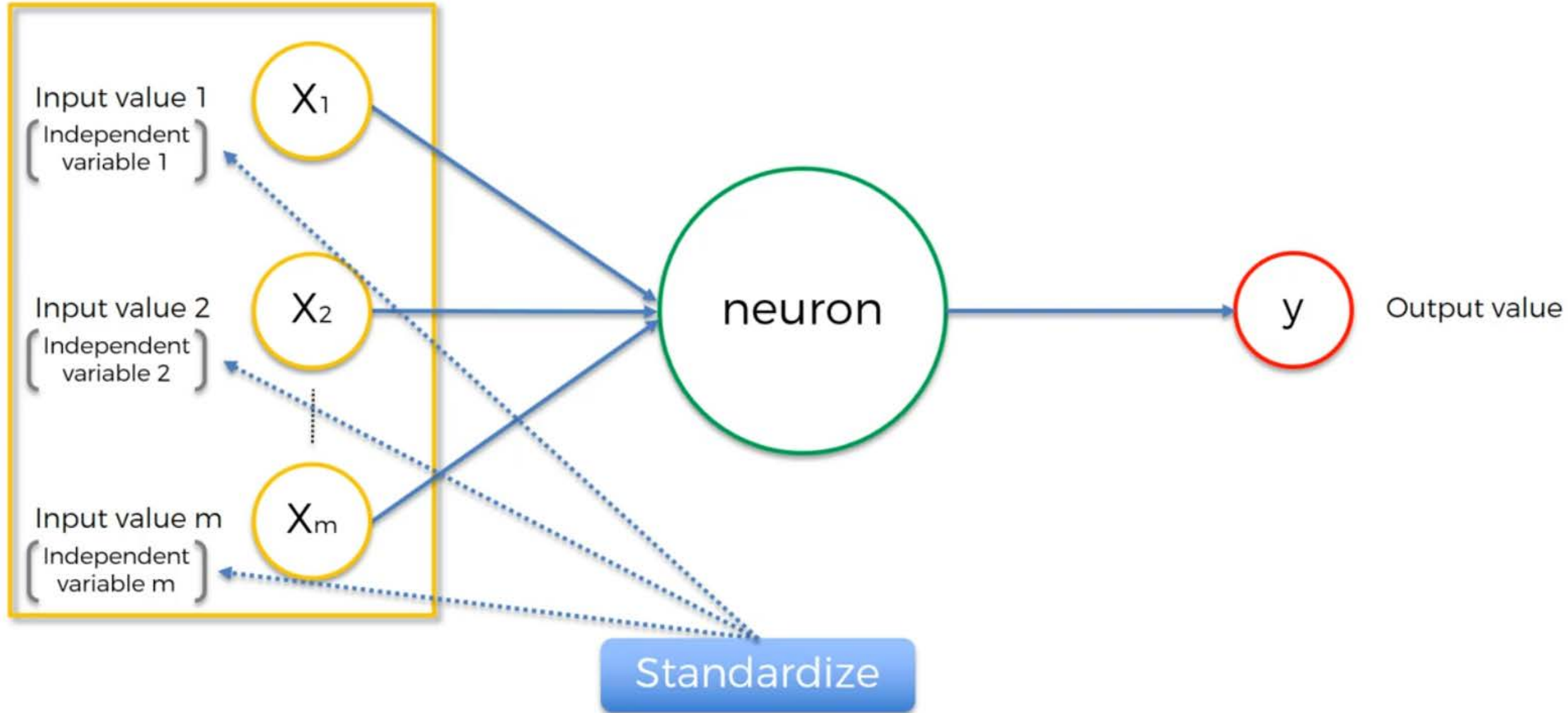




# The Neuron

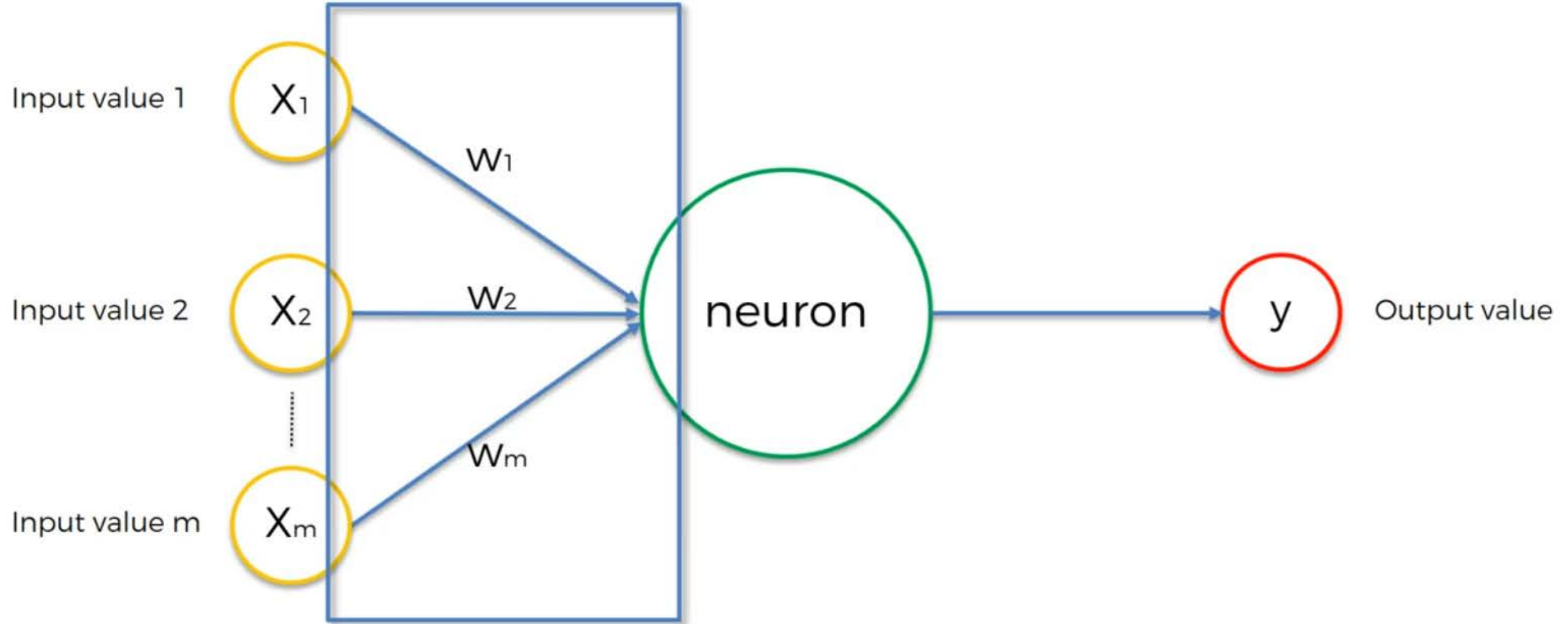


# The Neuron

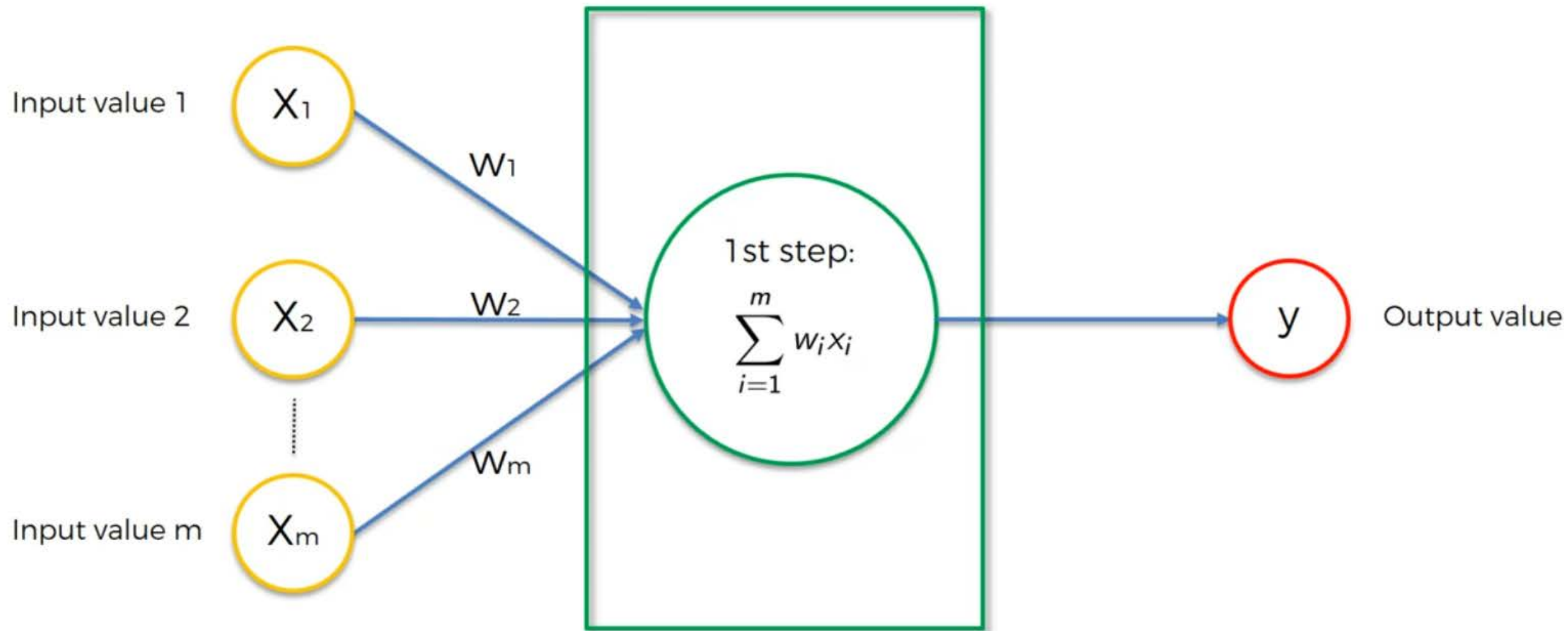




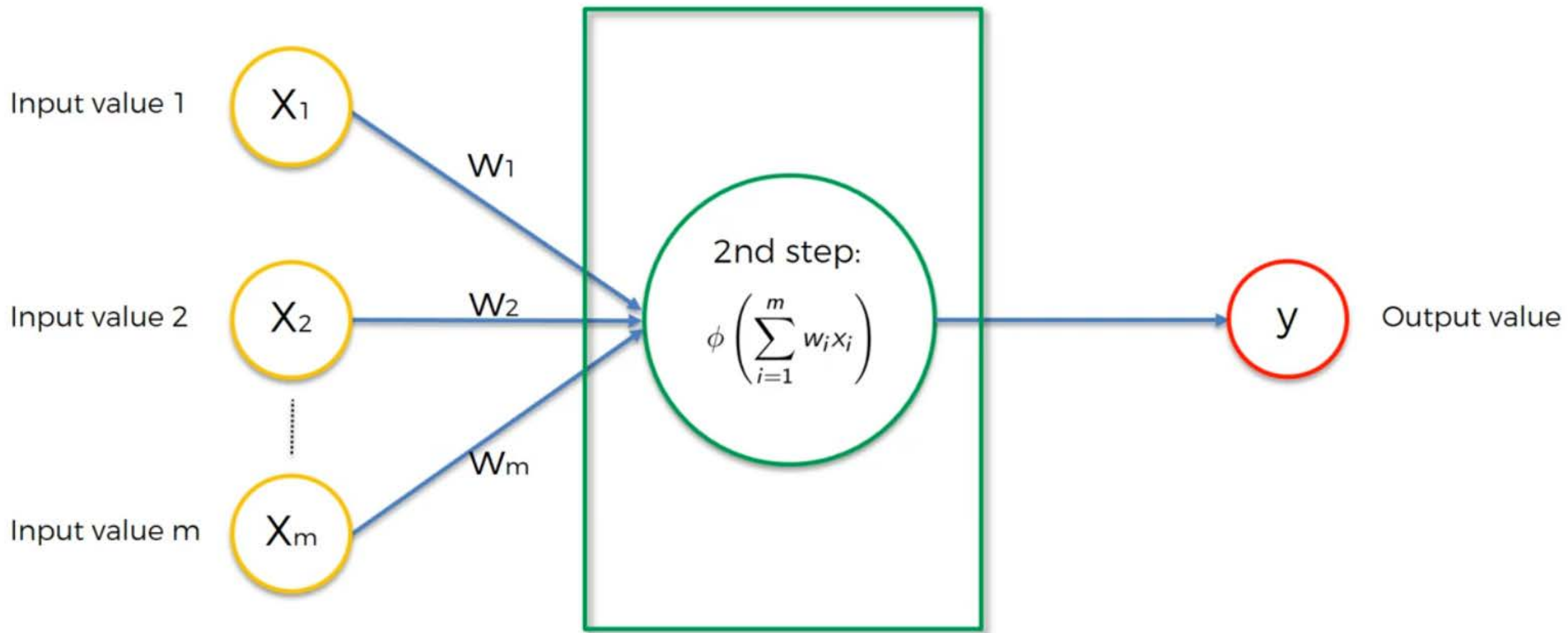
# The Neuron



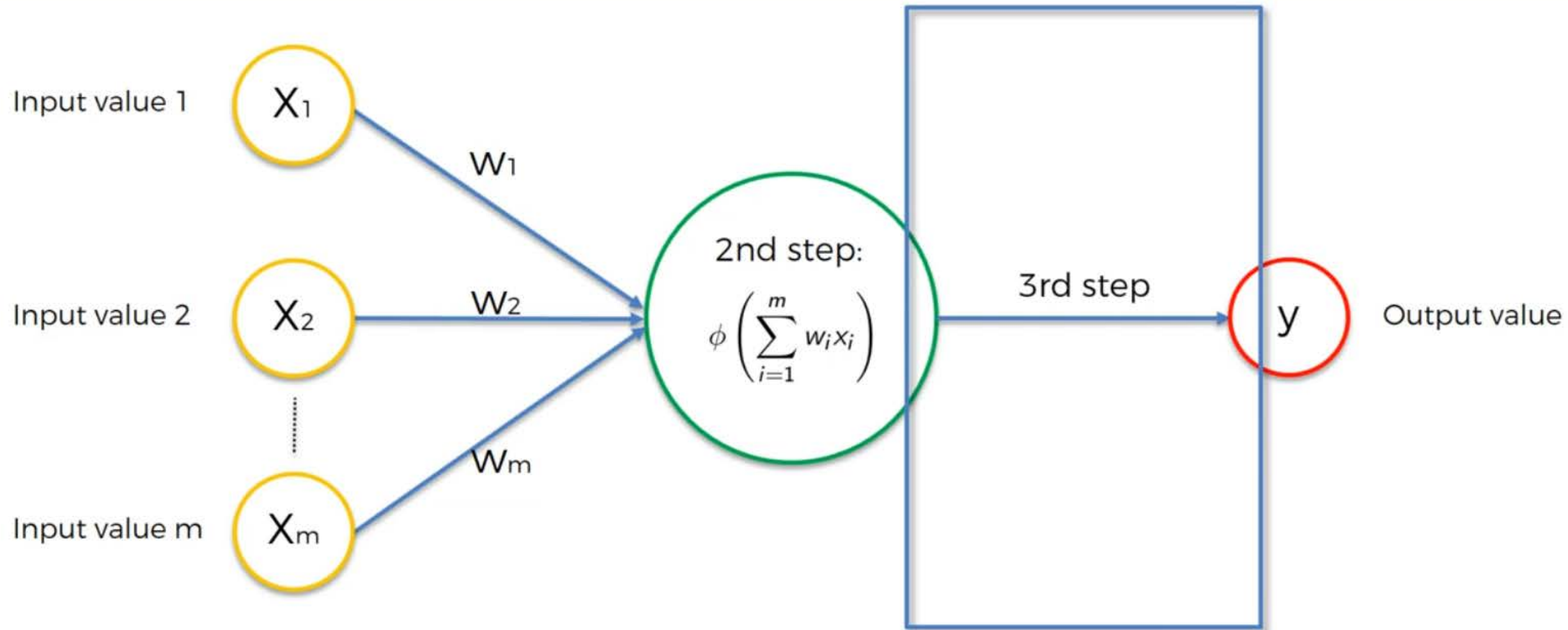
# The Neuron



# The Neuron



# The Neuron

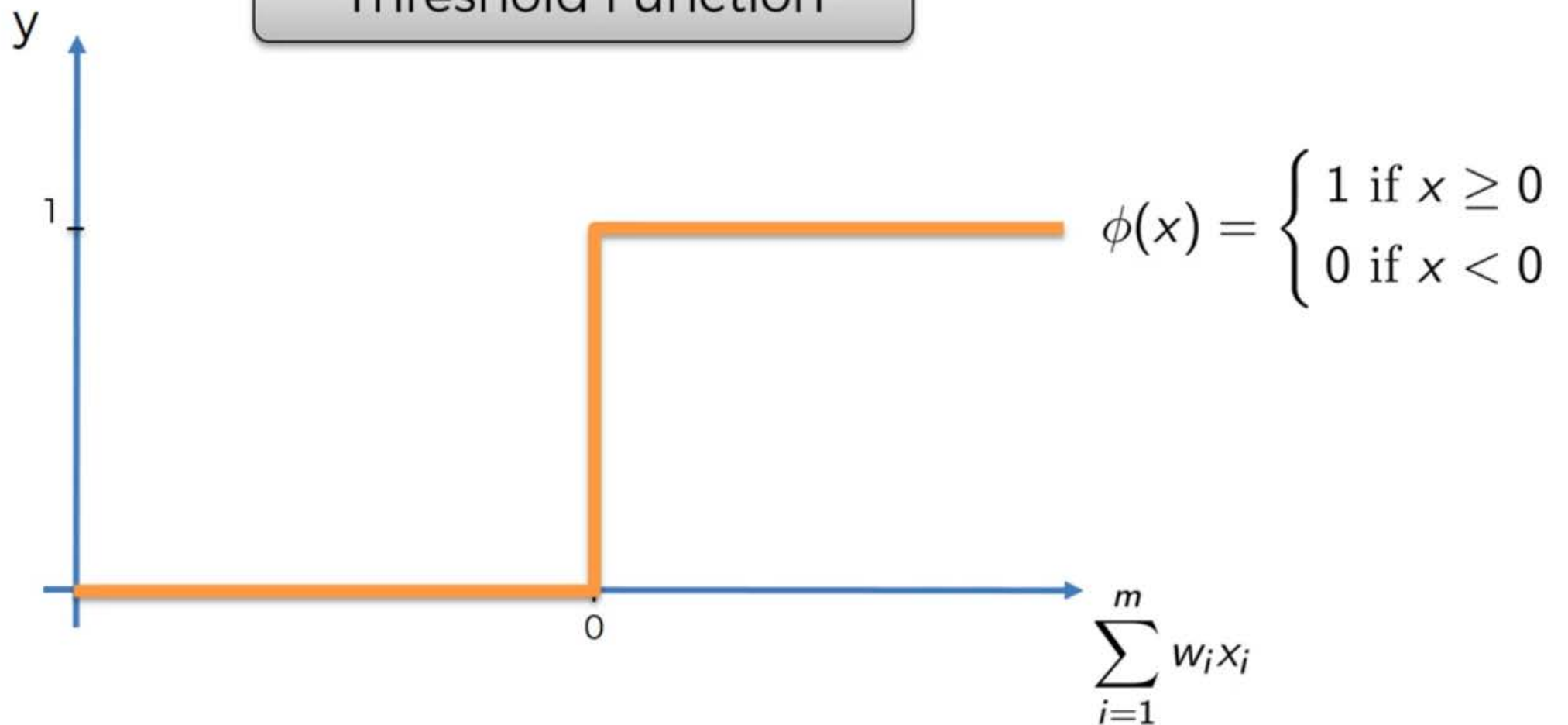




# The Activation Function

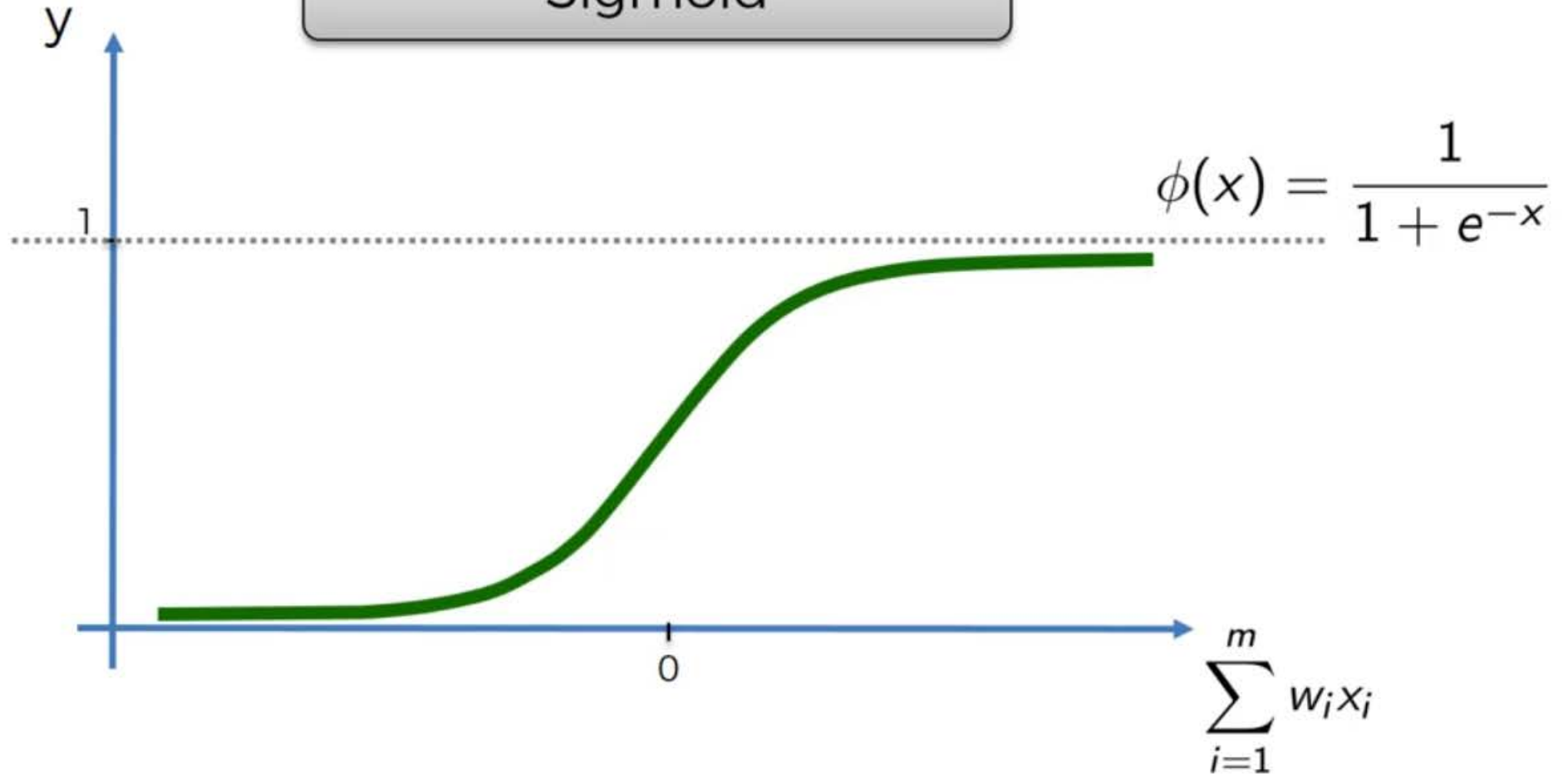
# The Activation Function

Threshold Function

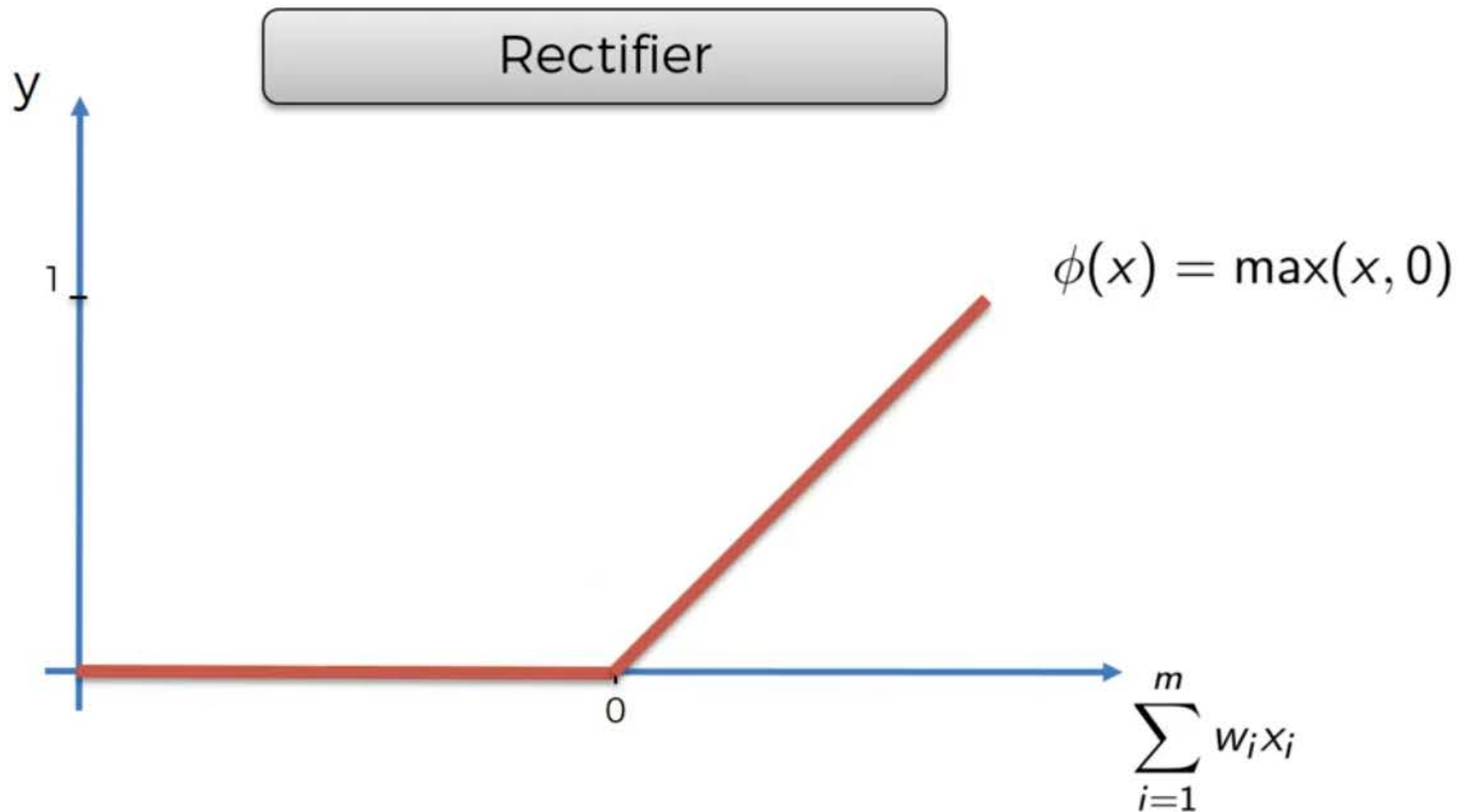


# The Activation Function

Sigmoid



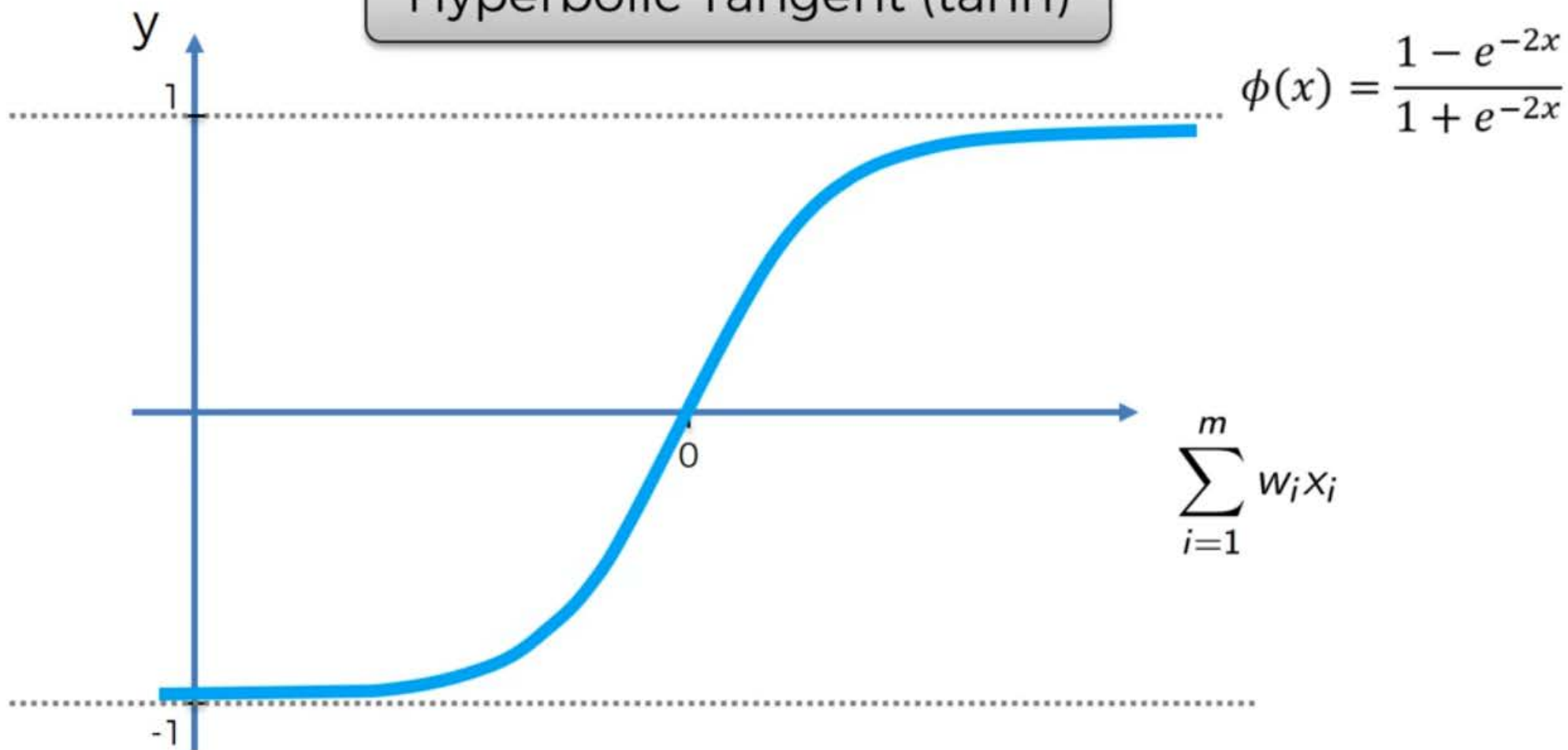
# The Activation Function



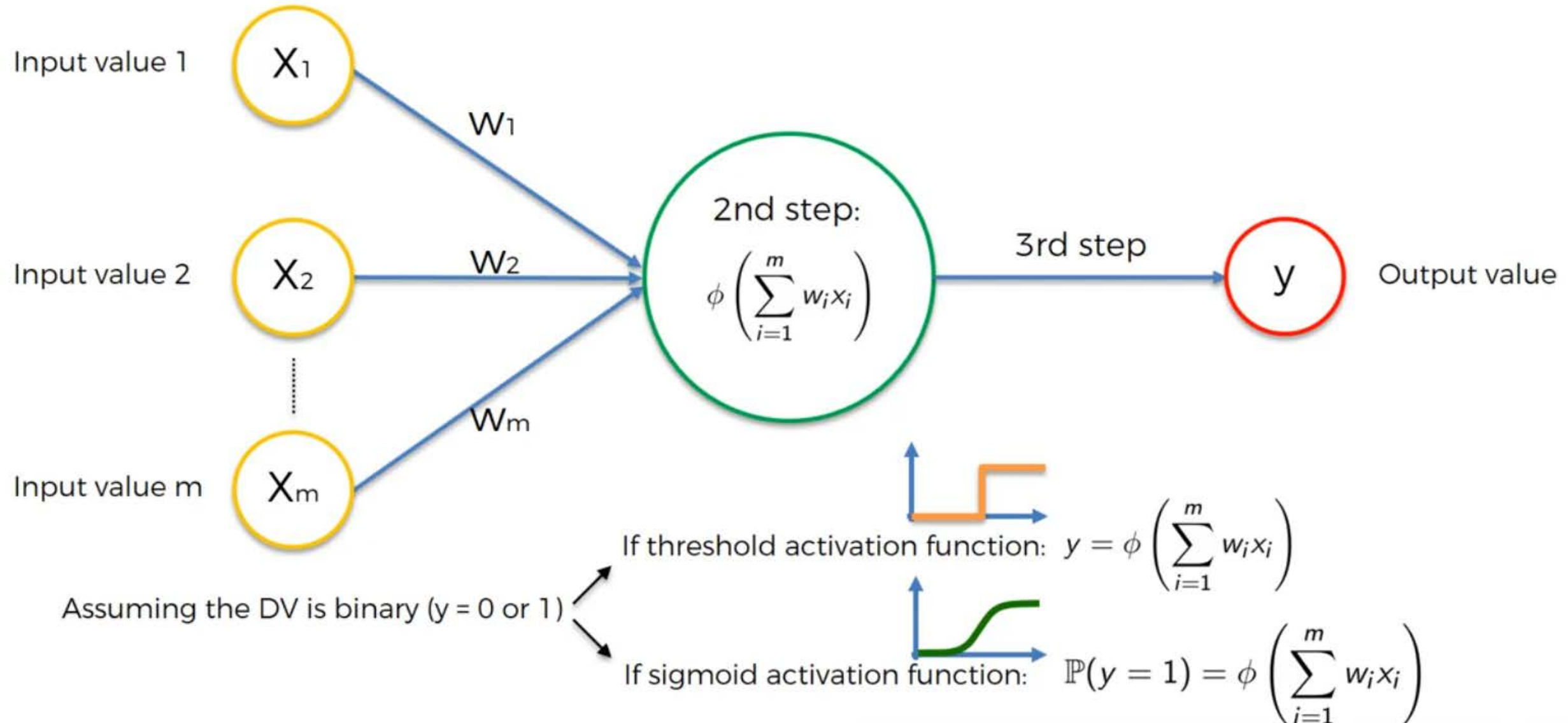


# The Activation Function

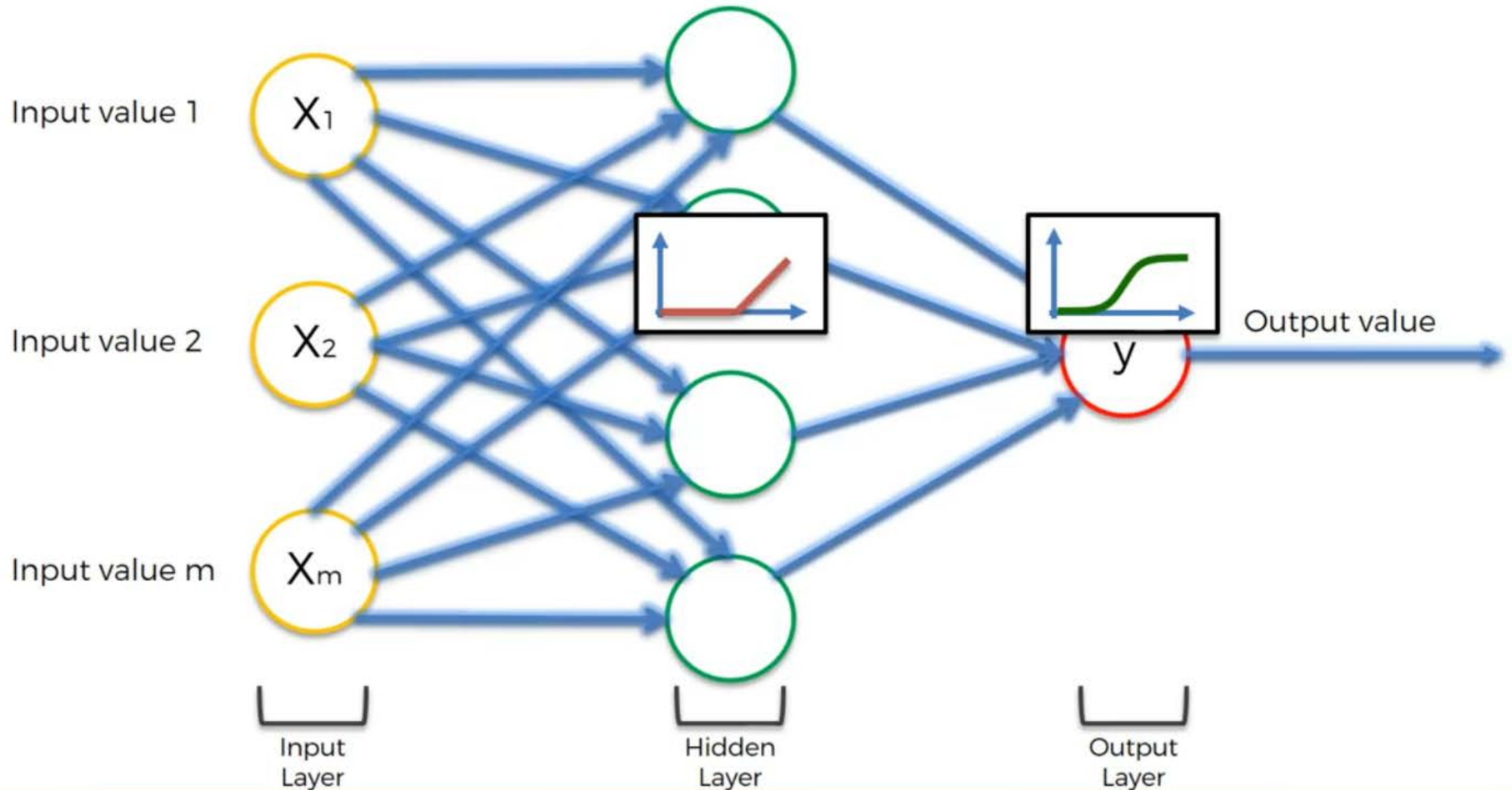
Hyperbolic Tangent (tanh)



# The Activation Function



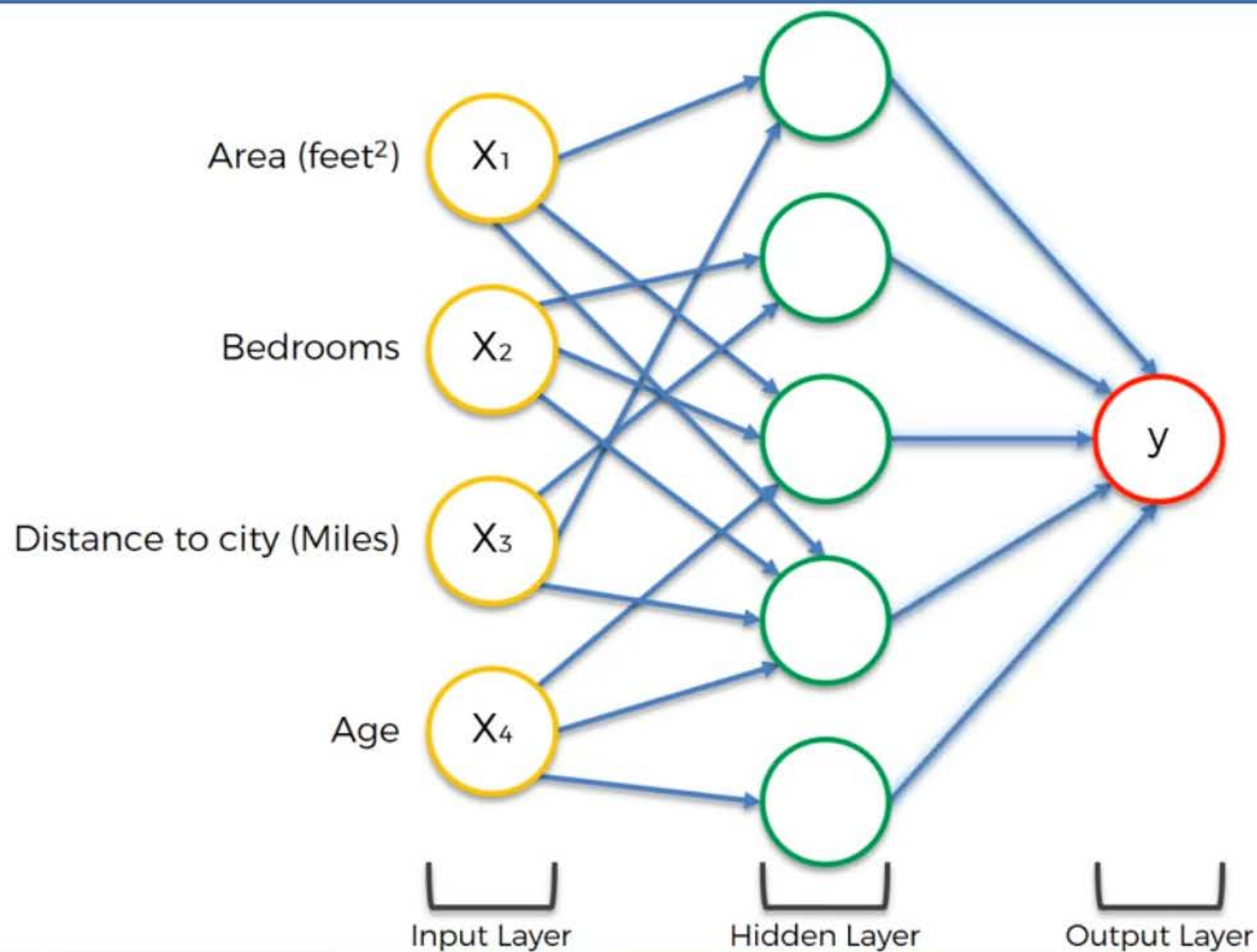
# The Activation Function



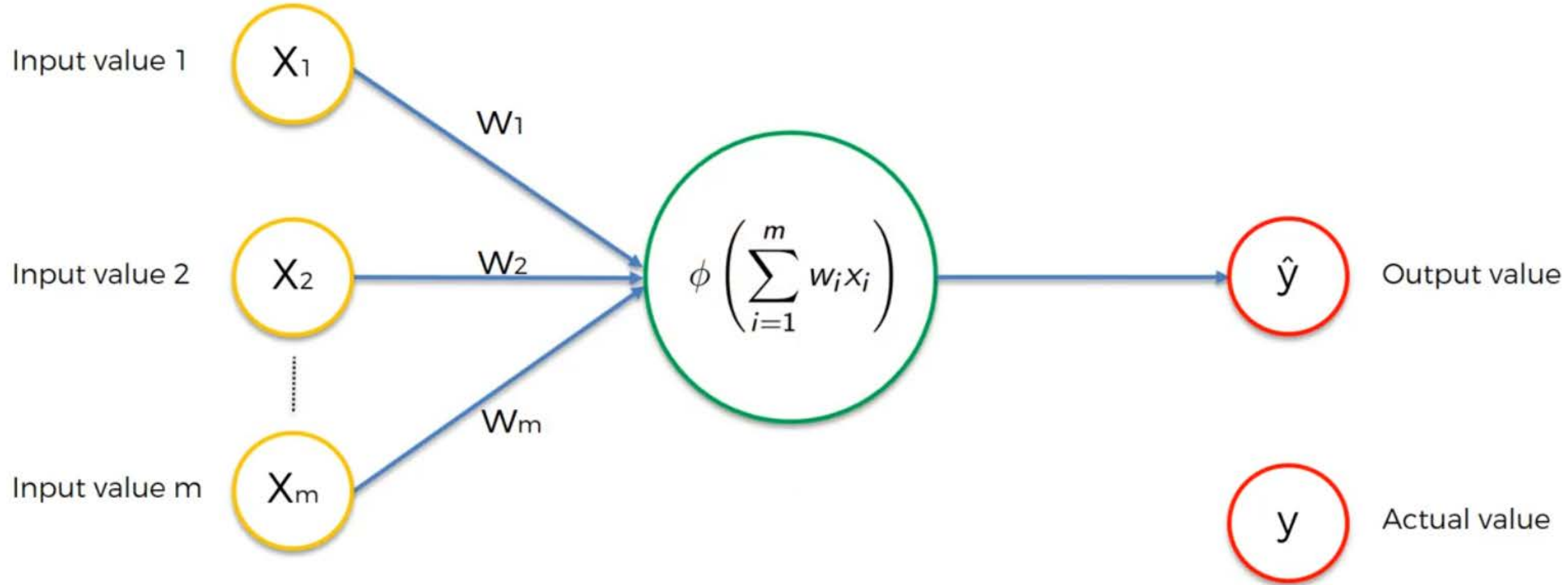
# How do NNs Work?



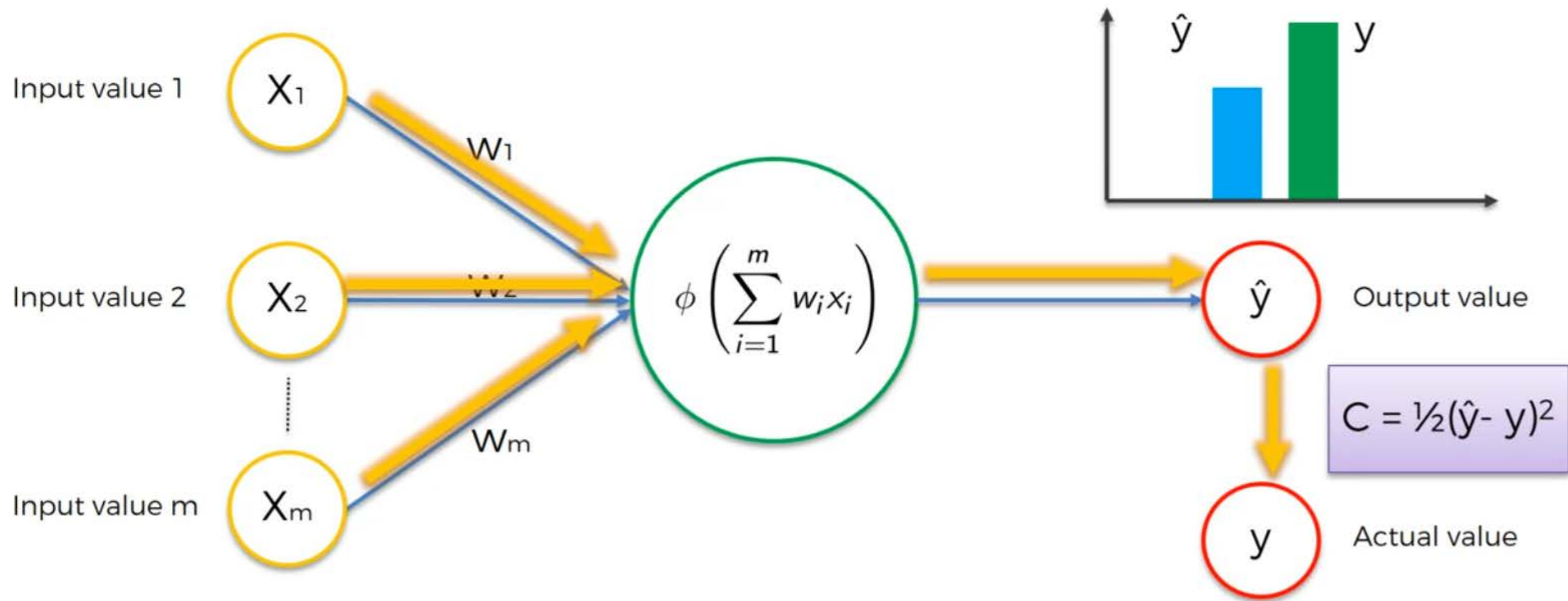
# How Do Neural Networks Work?



# How do Neural Networks learn?

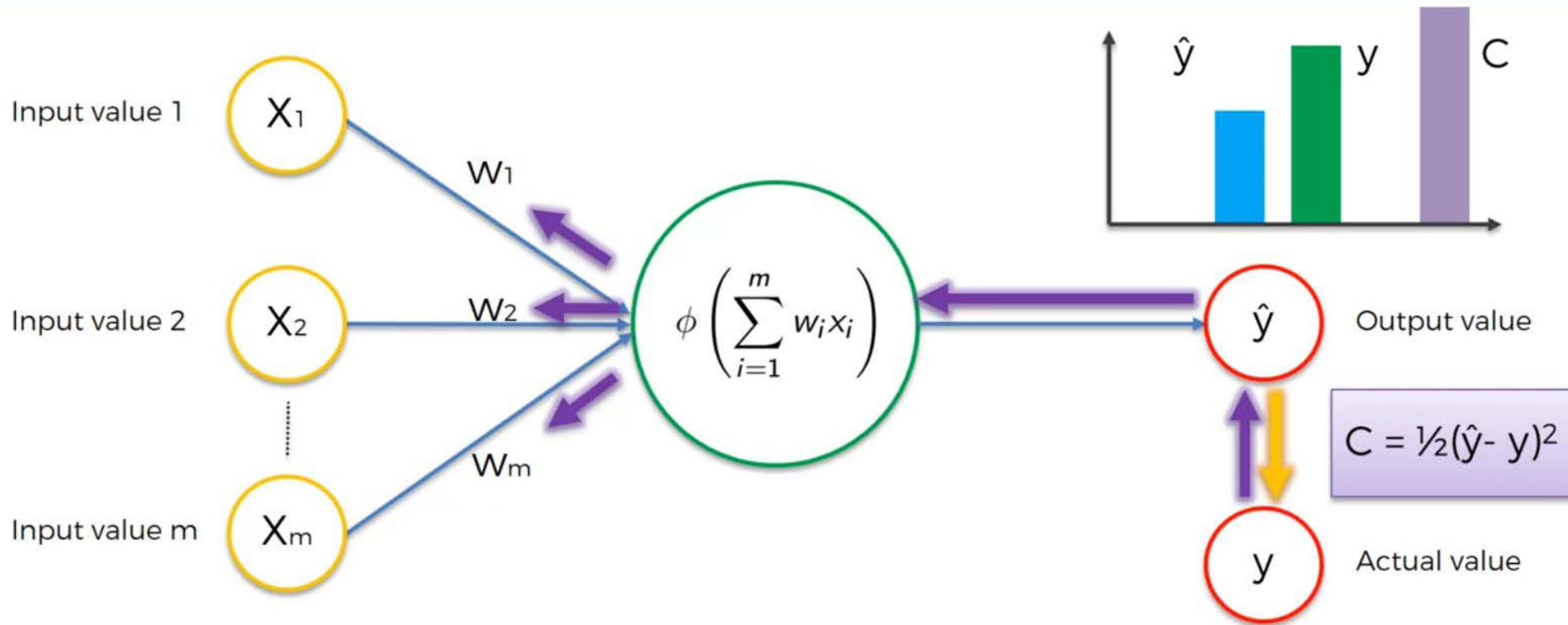


# How do Neural Networks learn?

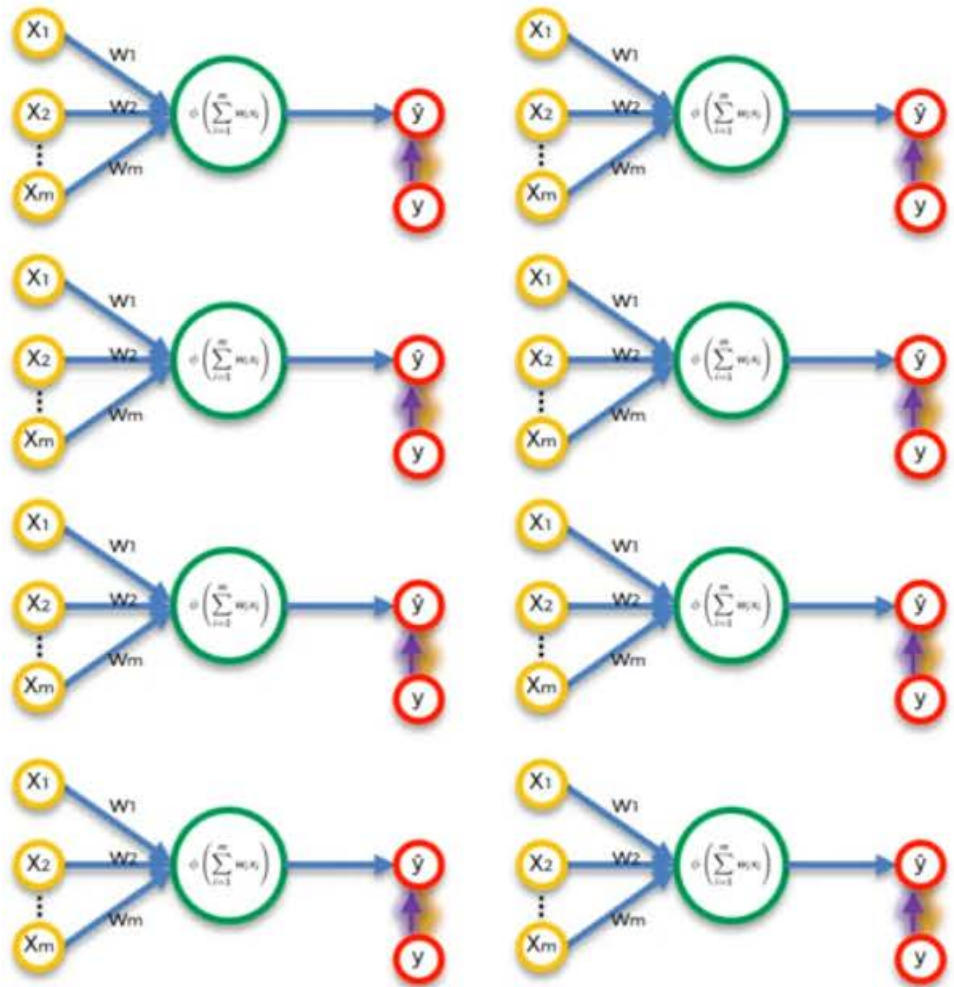


# How do Neural Networks learn?

05:40 / 12:59





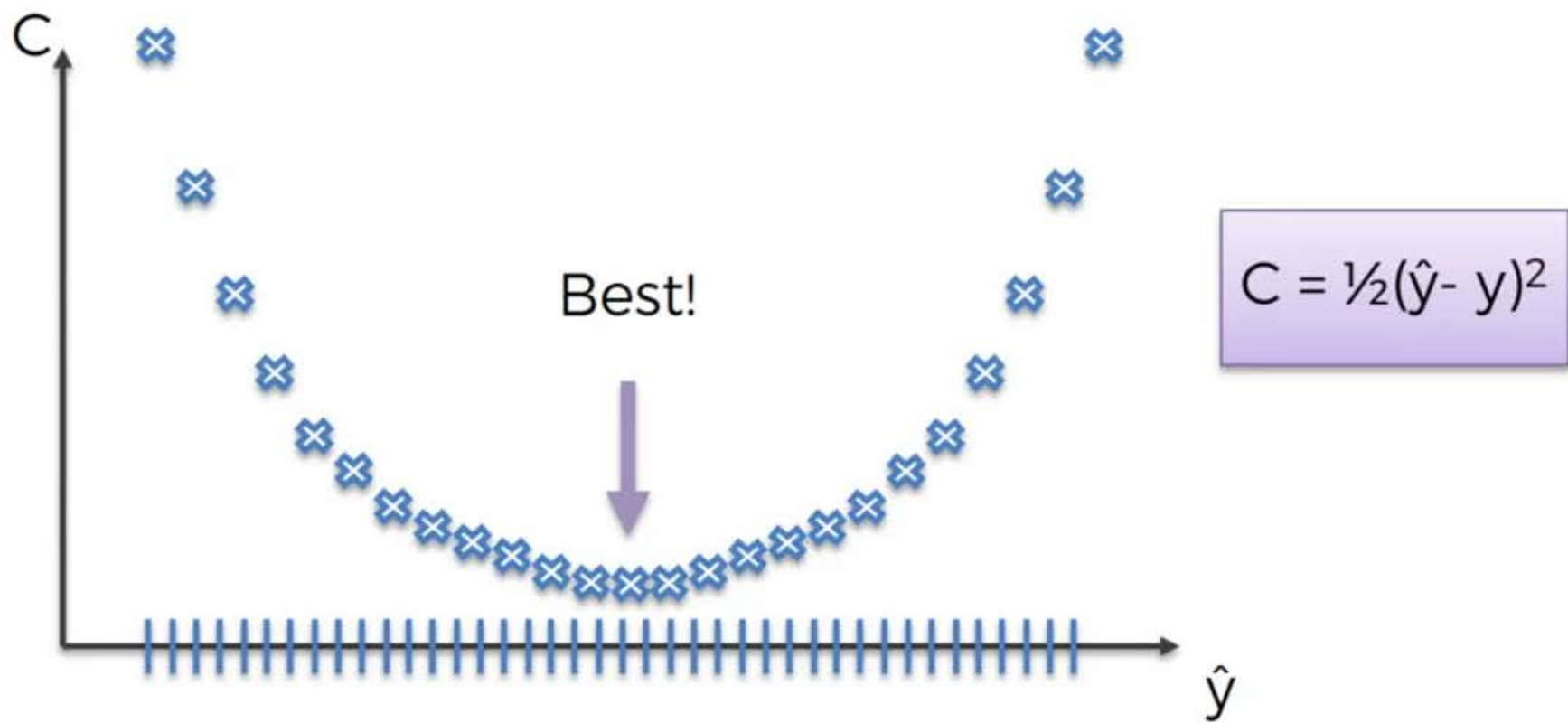


Row ID	Study Hrs	Sleep Hrs	Quiz	Exam
1	12	6	78%	93%
2	22	6.5	24%	68%
3	115	4	100%	95%
4	31	9	67%	75%
5	0	10	58%	51%
6	5	8	78%	60%
7	92	6	82%	89%
8	57	8	91%	97%

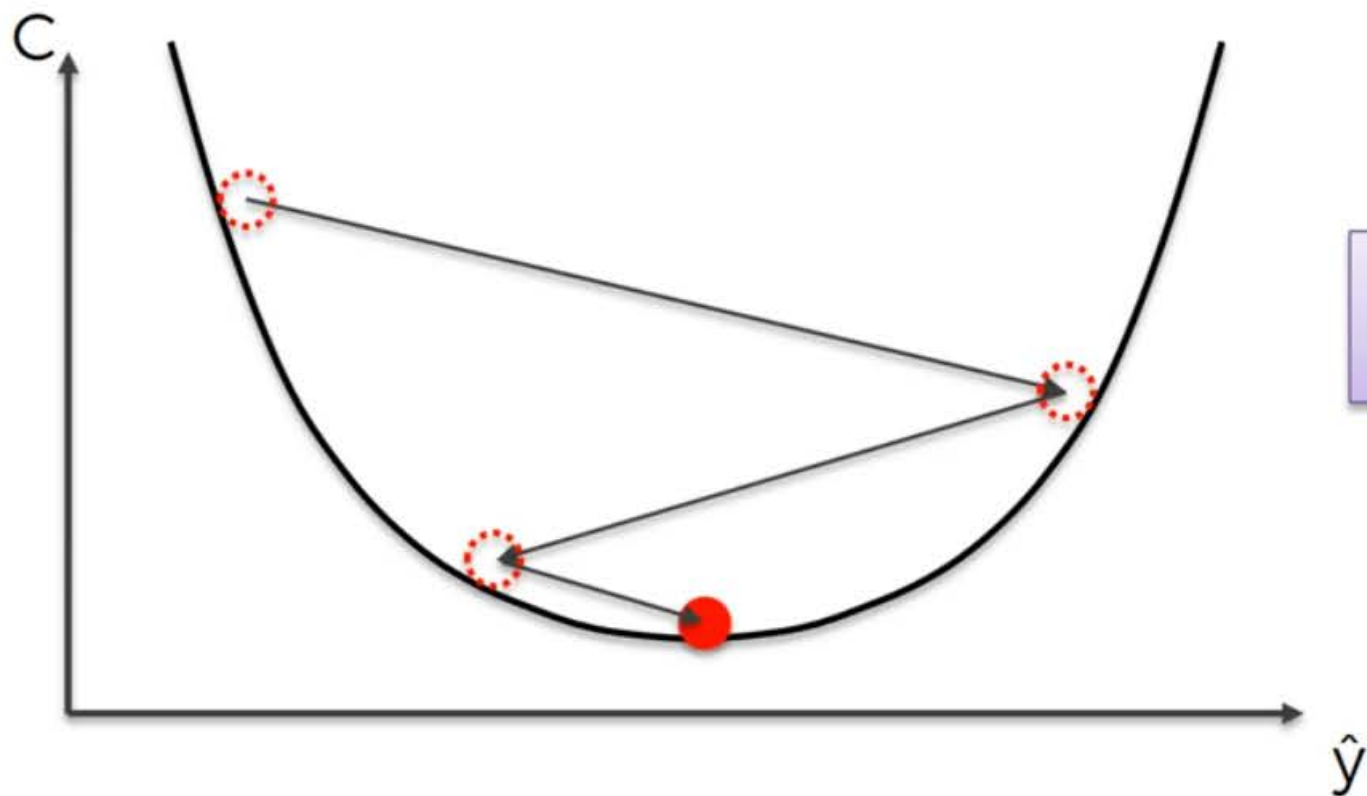
$$C = \sum \frac{1}{2}(\hat{y} - y)^2$$



# Gradient Descent



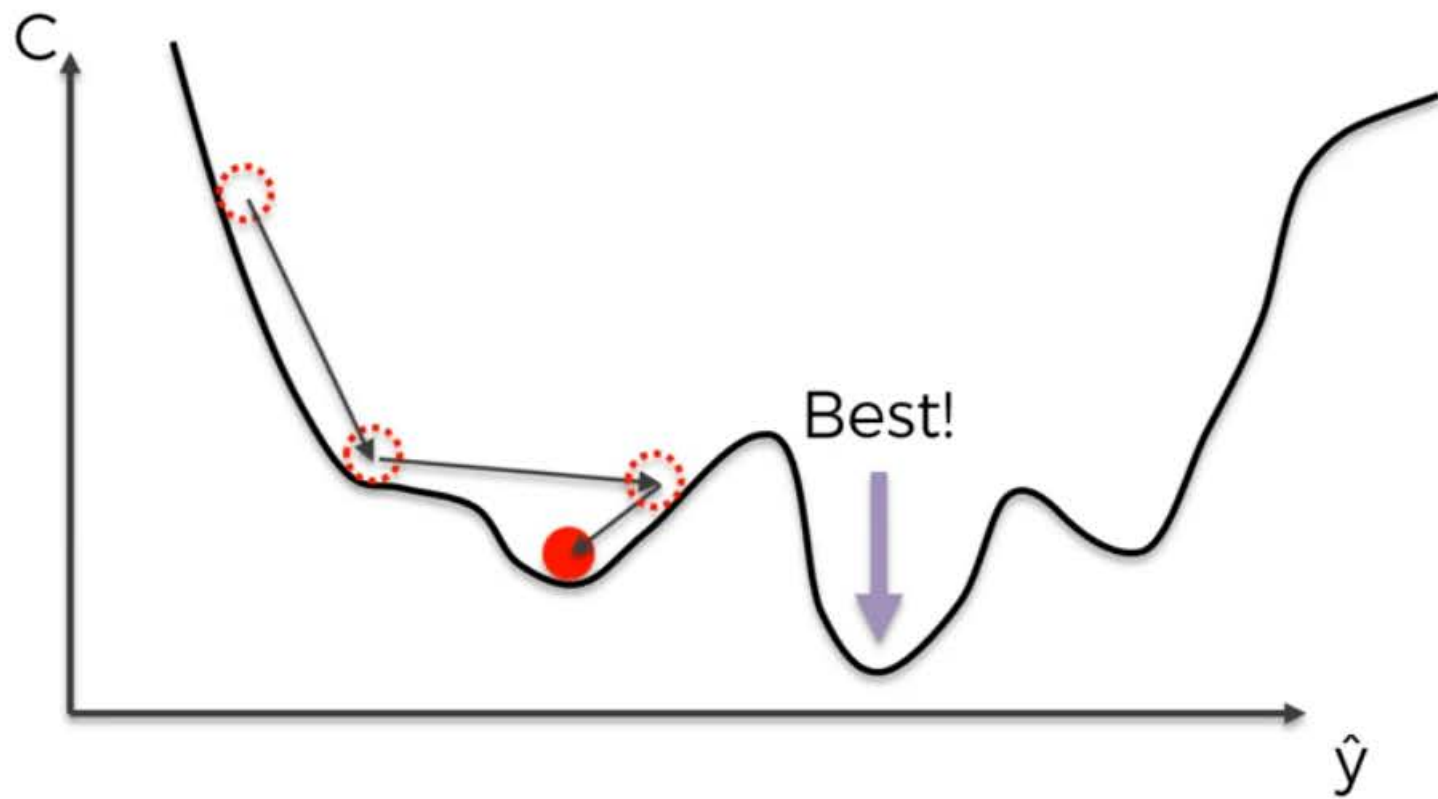
# Gradient Descent



$$C = \frac{1}{2}(\hat{y} - y)^2$$


# Stochastic Gradient Descent

# Stochastic Gradient Descent













# Stochastic Gradient Descent

Upd w's 

Row ID	Study Hrs	Sleep Hrs	Quiz	Exam
1	12	6	78%	93%
2	22	6.5	24%	68%
3	115	4	100%	95%
4	31	9	67%	75%
5	0	10	58%	51%
6	5	8	78%	60%
7	92	6	82%	89%
8	57	8	91%	97%

Batch  
Gradient  
Descent

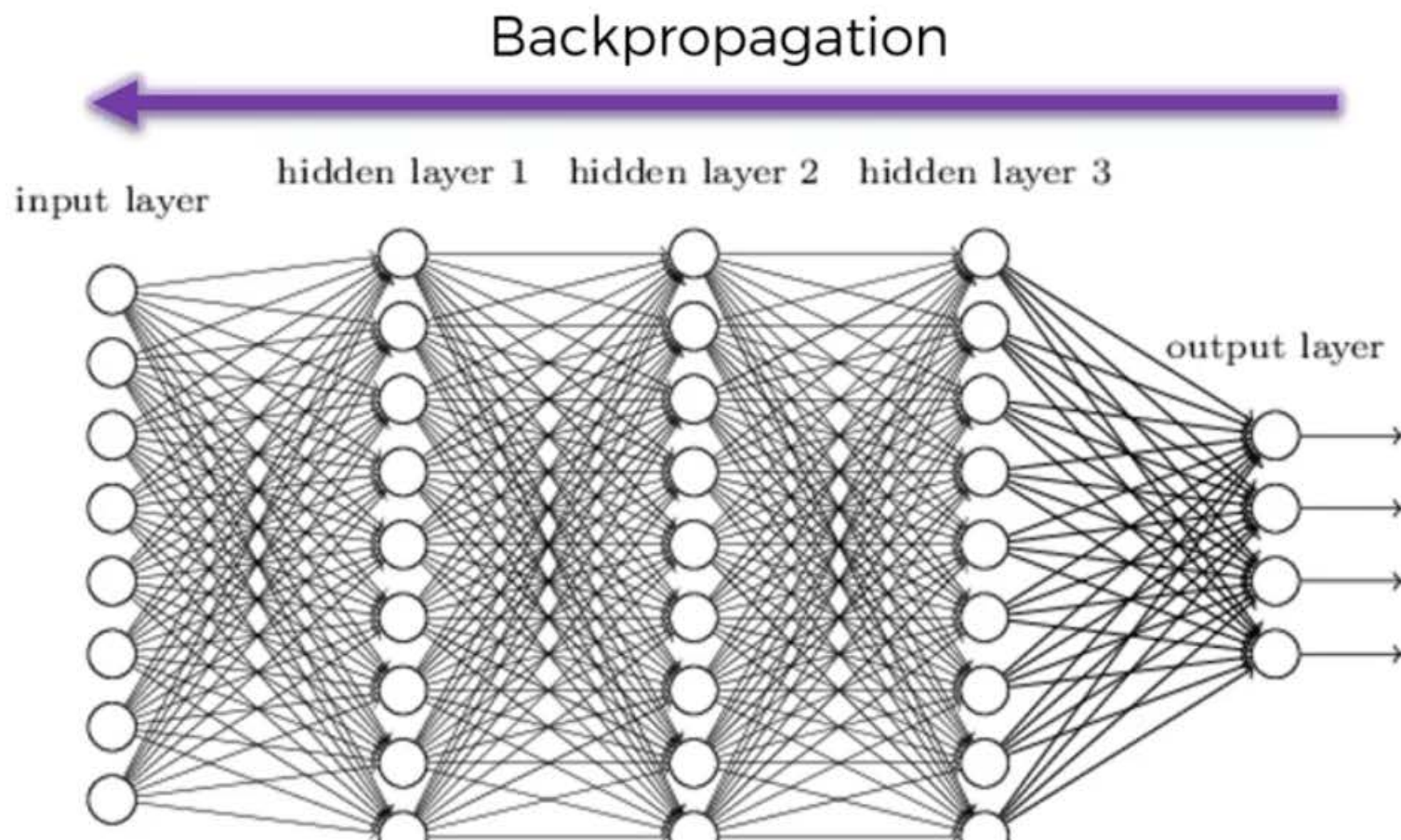
Upd w's   
Upd w's   
Upd w's   
Upd w's   
Upd w's   
Upd w's   
Upd w's   
Upd w's 

Row ID	Study Hrs	Sleep Hrs	Quiz	Exam
1	12	6	78%	93%
2	22	6.5	24%	68%
3	115	4	100%	95%
4	31	9	67%	75%
5	0	10	58%	51%
6	5	8	78%	60%
7	92	6	82%	89%
8	57	8	91%	97%

Stochastic  
Gradient  
Descent

# Backpropagation

# Gradient Descent





# Training the ANN with Stochastic Gradient Descent

**STEP 1:** Randomly initialise the weights to small numbers close to 0 (but not 0).



**STEP 2:** Input the first observation of your dataset in the input layer, each feature in one input node.



**STEP 3:** Forward-Propagation: from left to right, the neurons are activated in a way that the impact of each neuron's activation is limited by the weights. Propagate the activations until getting the predicted result  $y$ .



**STEP 4:** Compare the predicted result to the actual result. Measure the generated error.



**STEP 5:** Back-Propagation: from right to left, the error is back-propagated. Update the weights according to how much they are responsible for the error. The learning rate decides by how much we update the weights.



**STEP 6:** Repeat Steps 1 to 5 and update the weights after each observation (Reinforcement Learning). Or:  
Repeat Steps 1 to 5 but update the weights only after a batch of observations (Batch Learning).



**STEP 7:** When the whole training set passed through the ANN, that makes an epoch. Redo more epochs.