

Plan of Attack

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

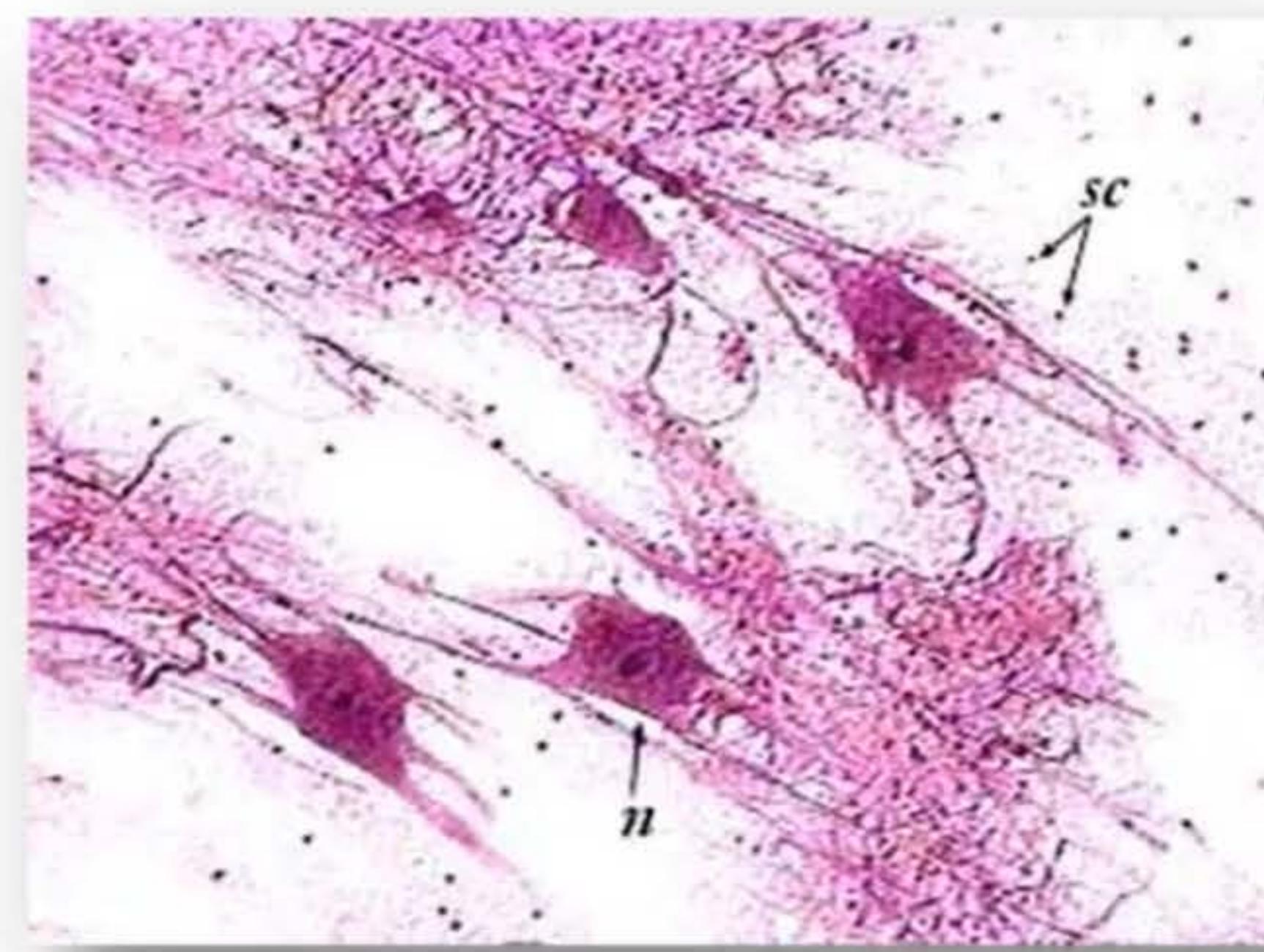
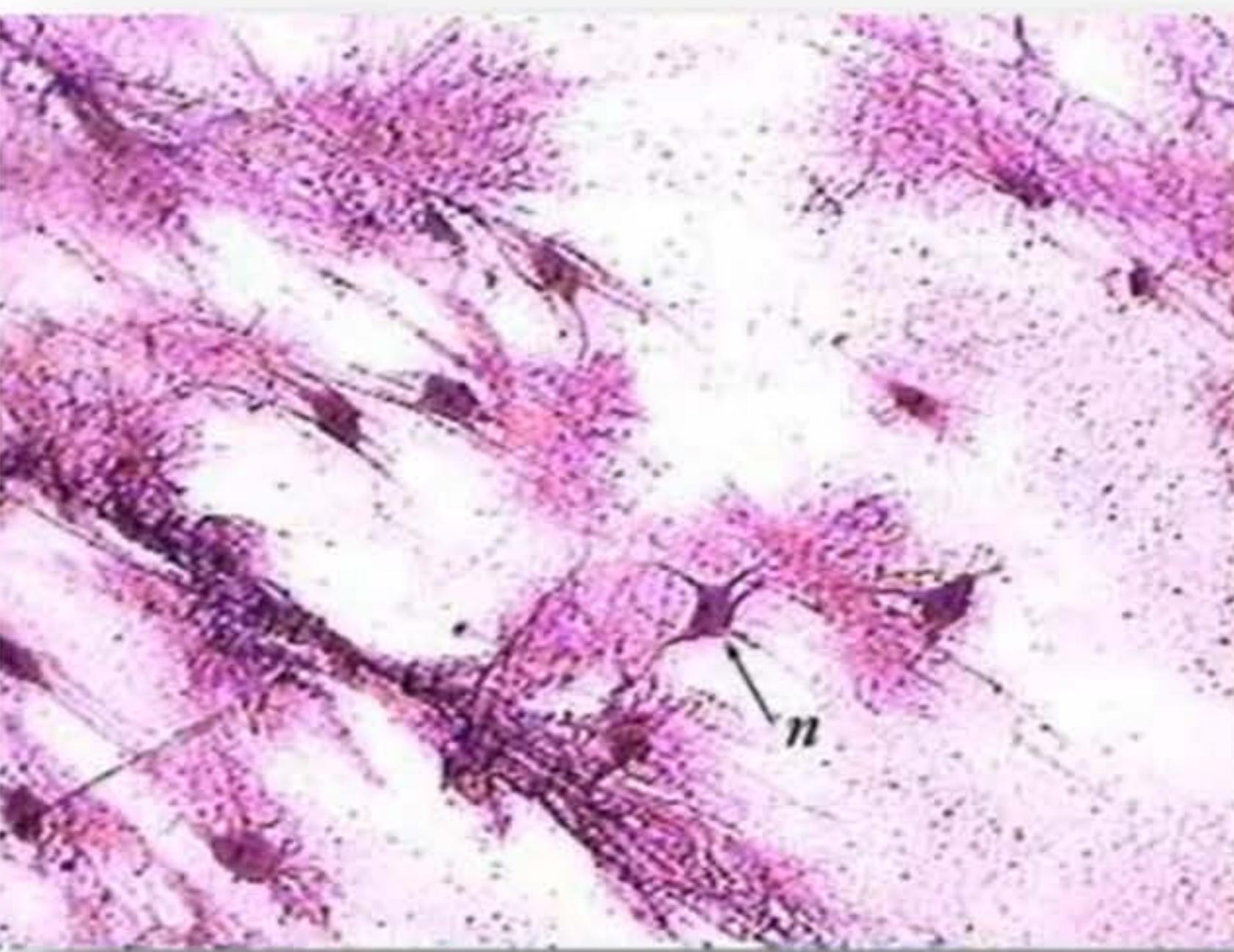


Image Source: www.austincc.edu

The Neuron

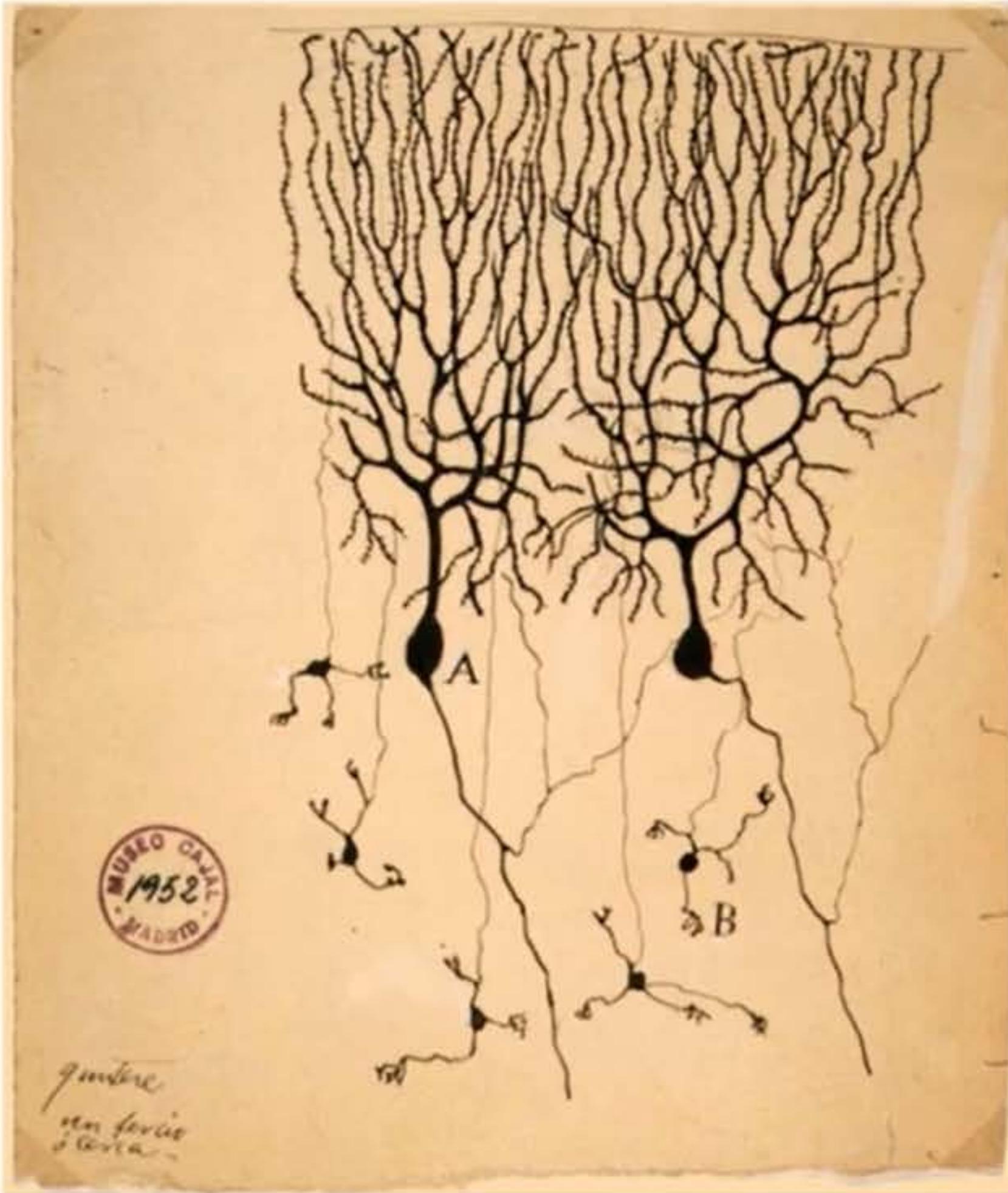


Image Source: Wikipedia

The Neuron

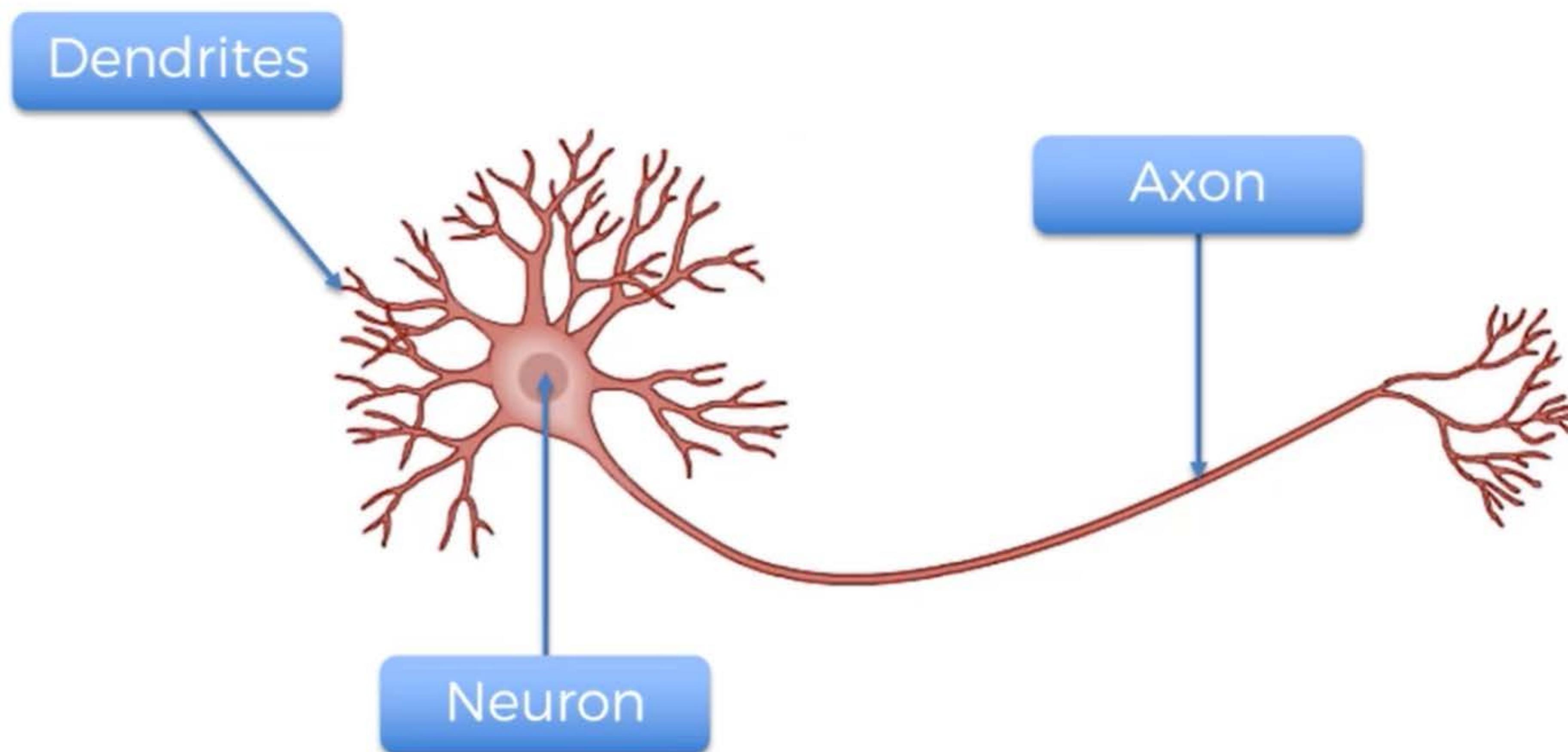


Image Source: Wikipedia

The Neuron

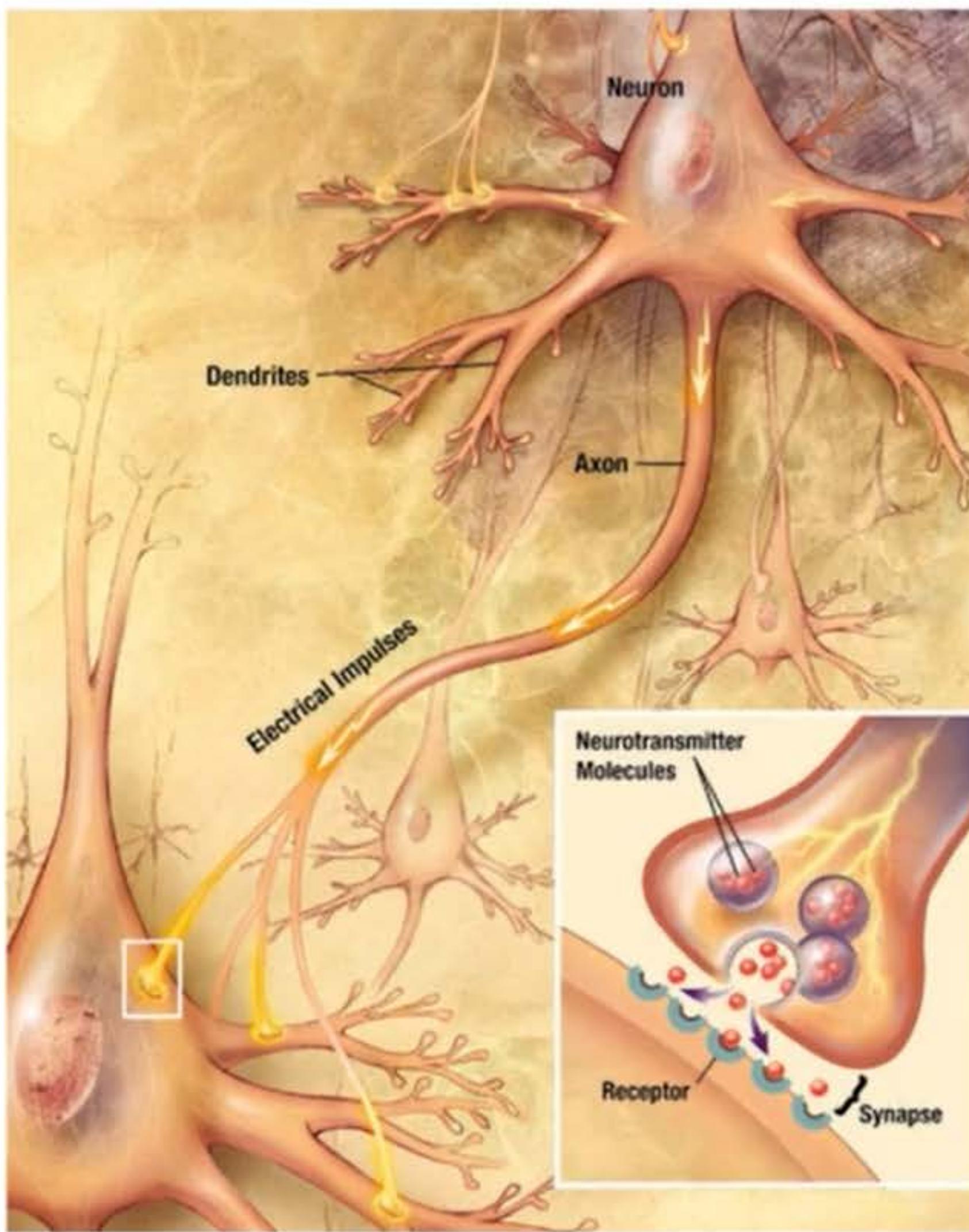
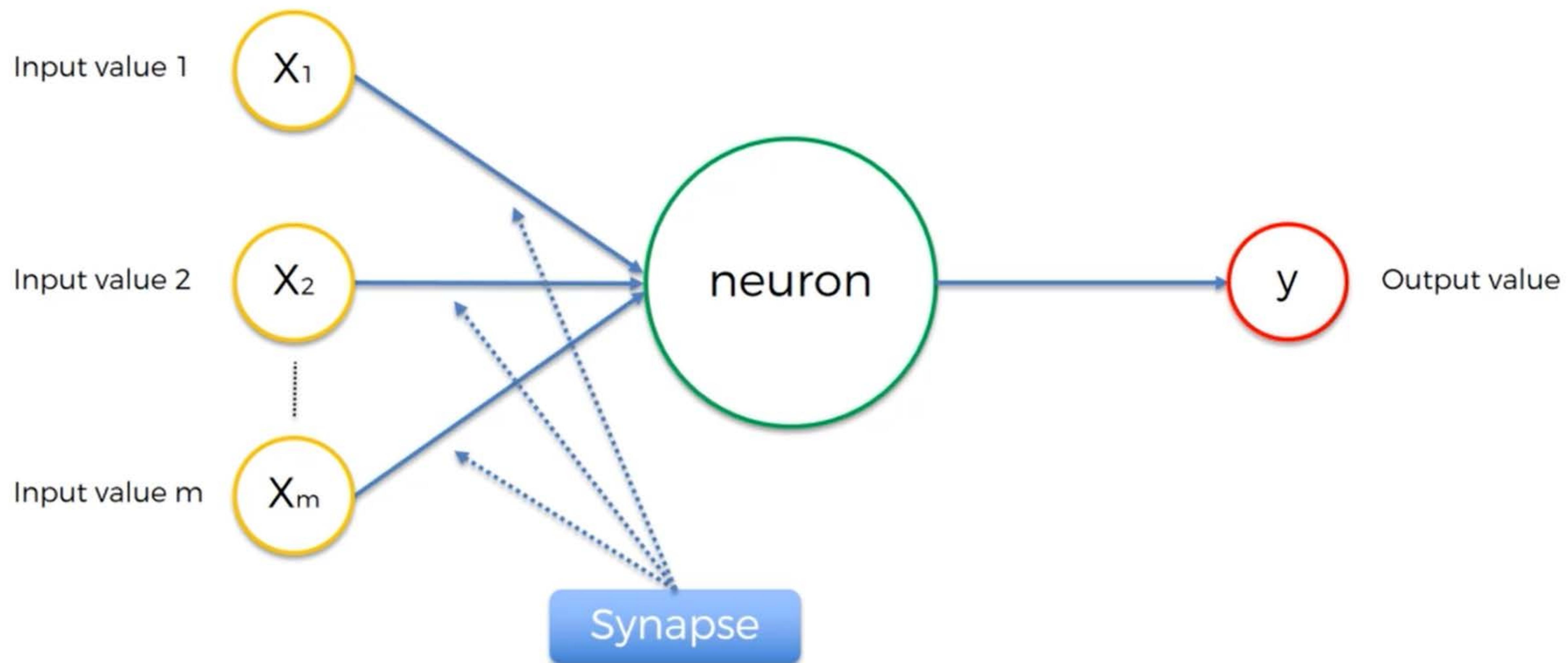
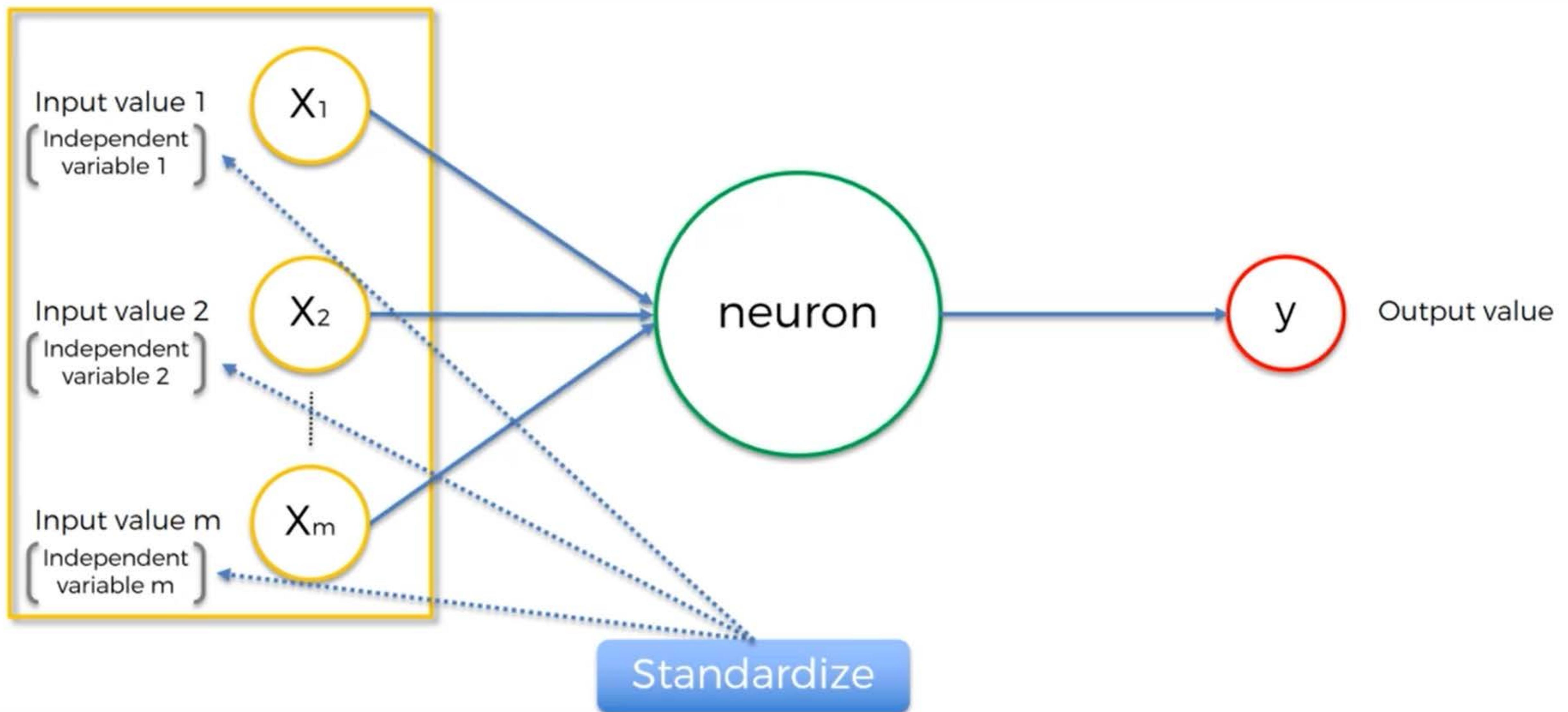


Image Source: Wikipedia

The Neuron



The Neuron



The Neuron

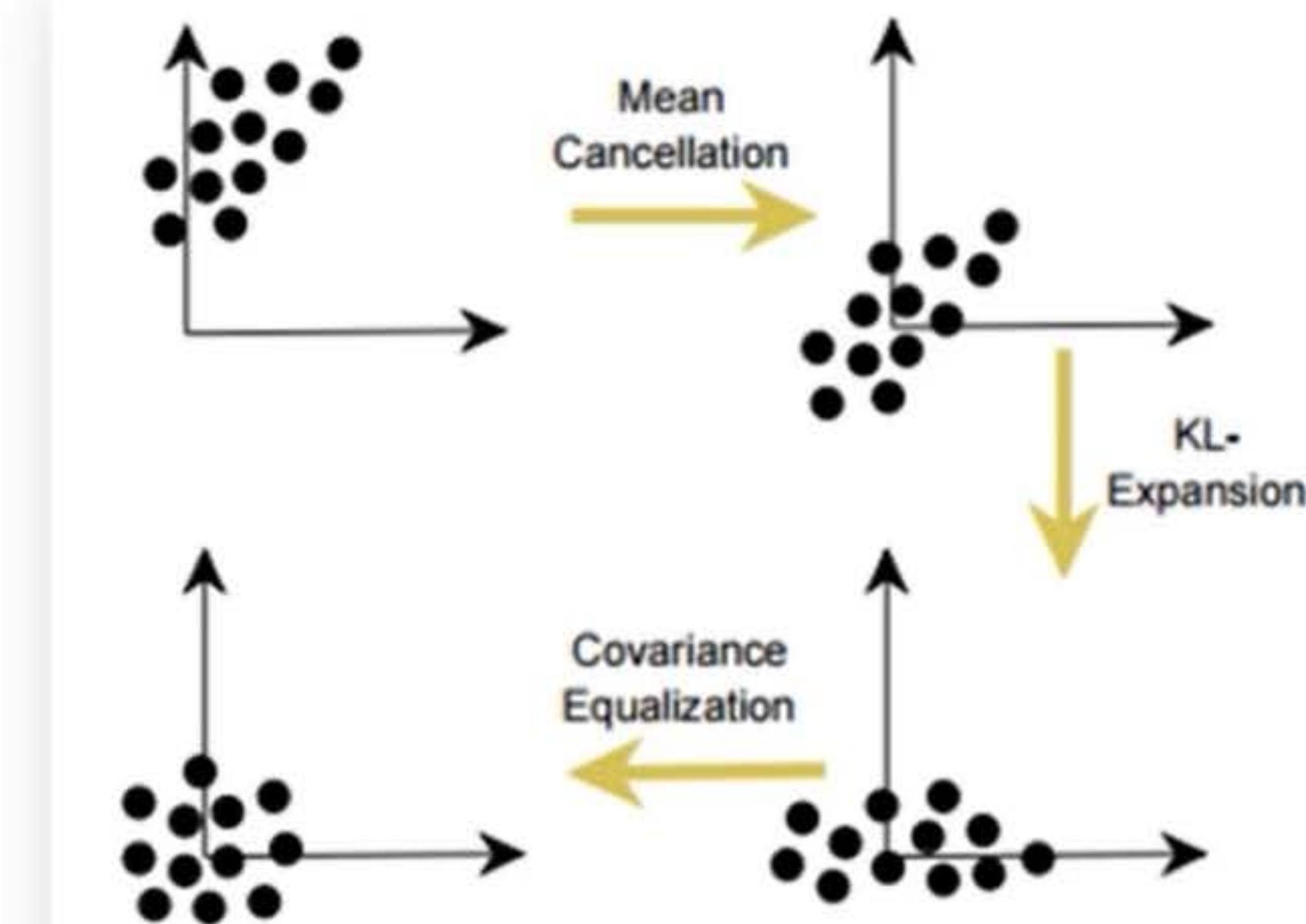
Additional Reading:

Efficient BackProp

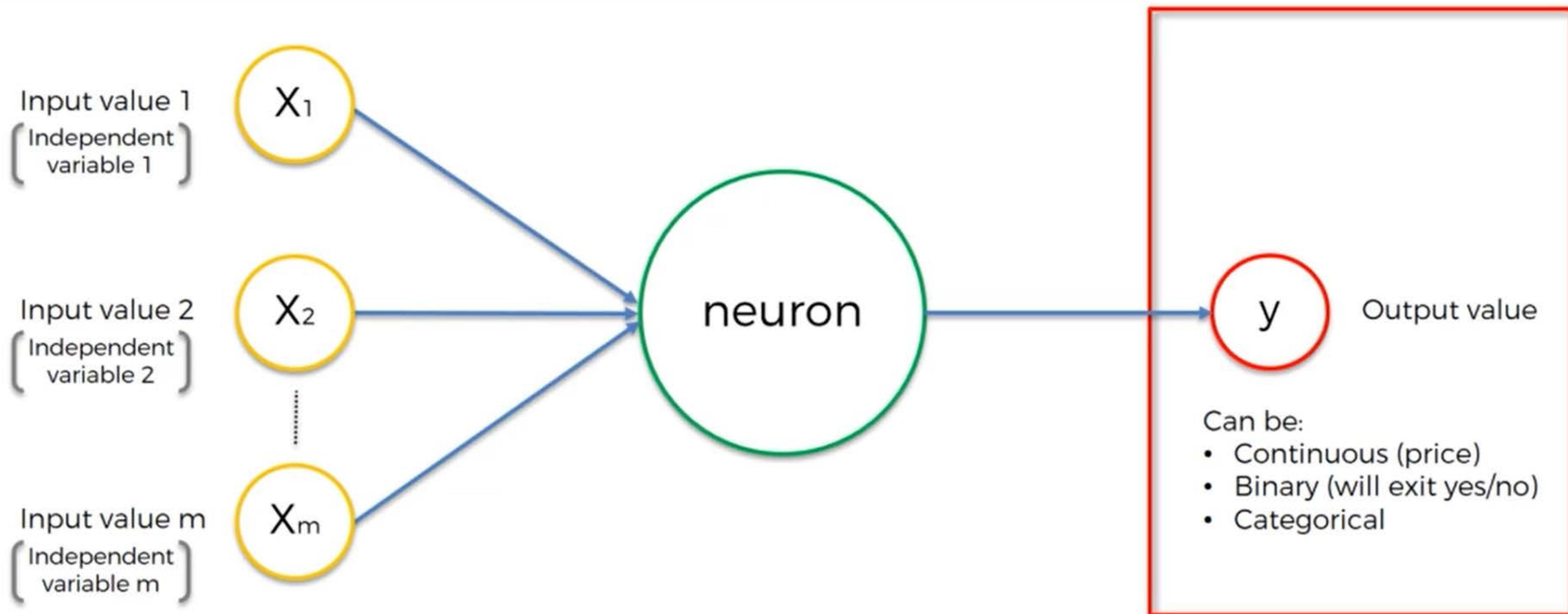
By Yann LeCun et al. (1998)

Link:

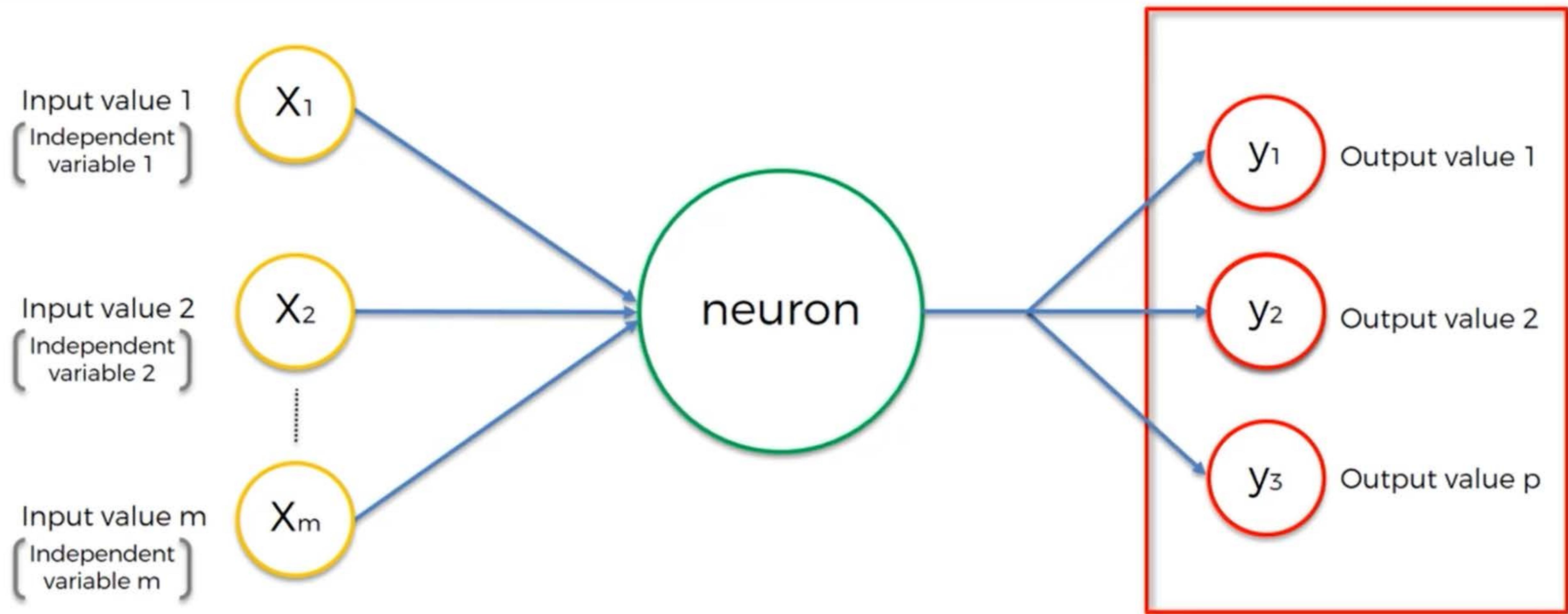
<http://yann.lecun.com/exdb/publis/pdf/lecun-98b.pdf>



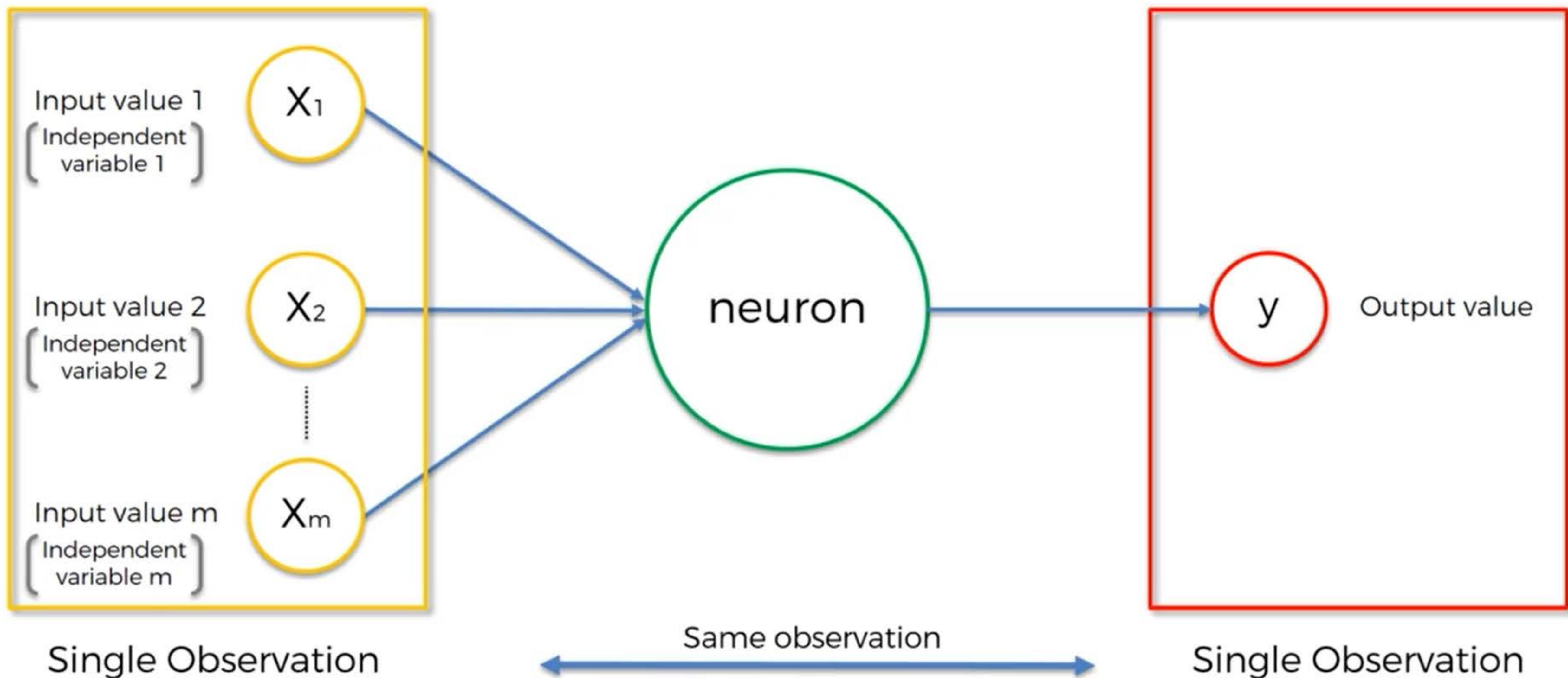
The Neuron



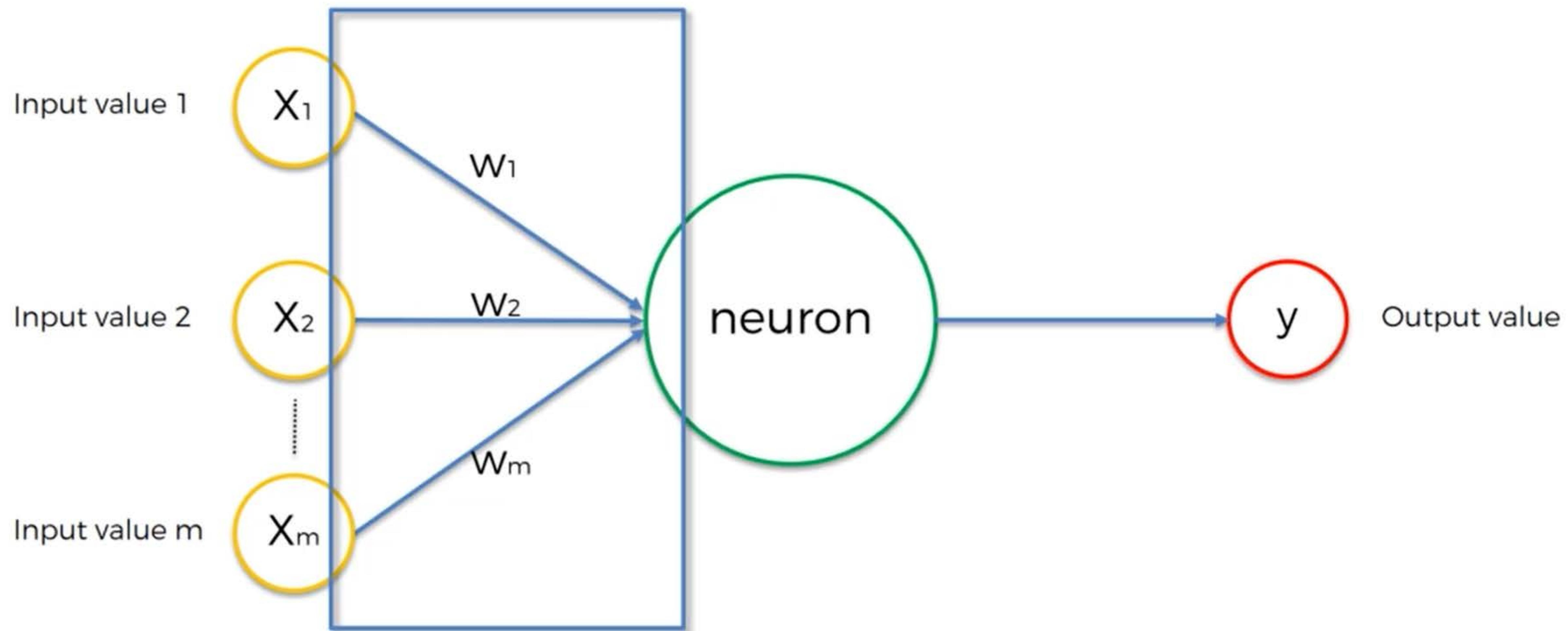
The Neuron



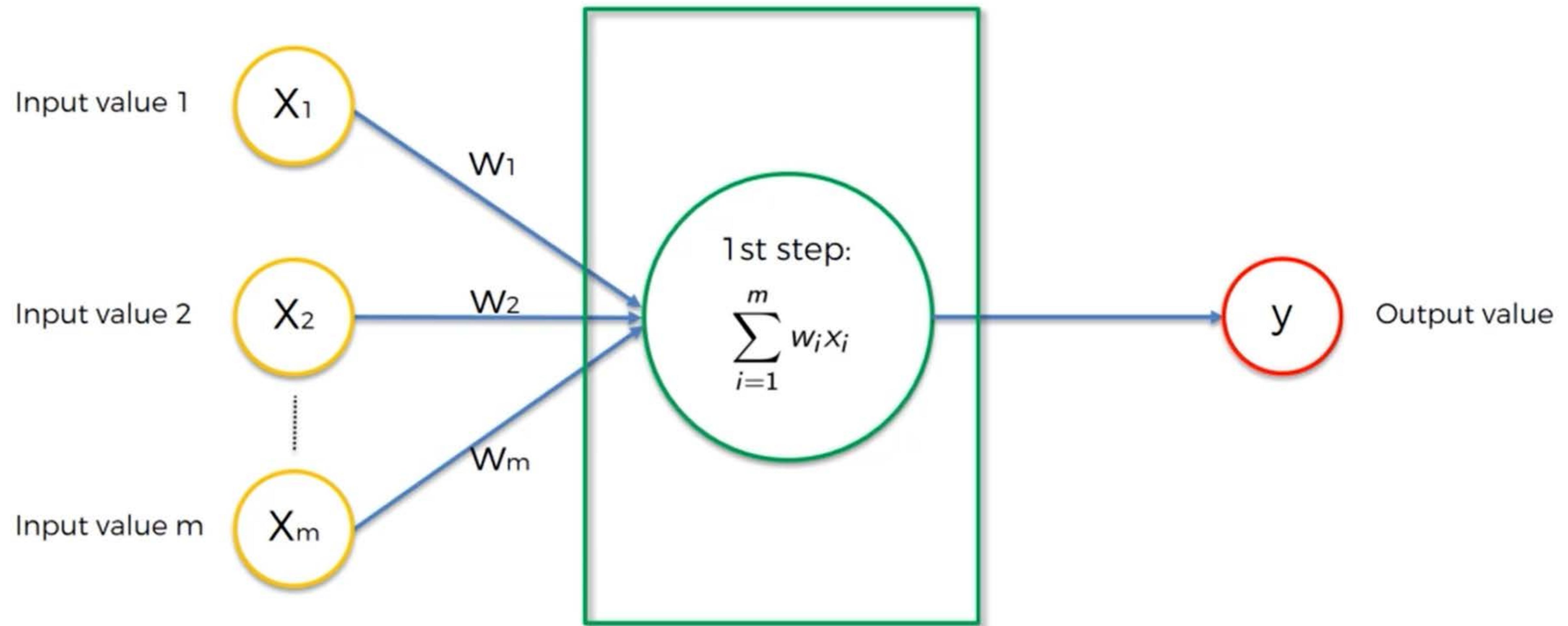
The Neuron



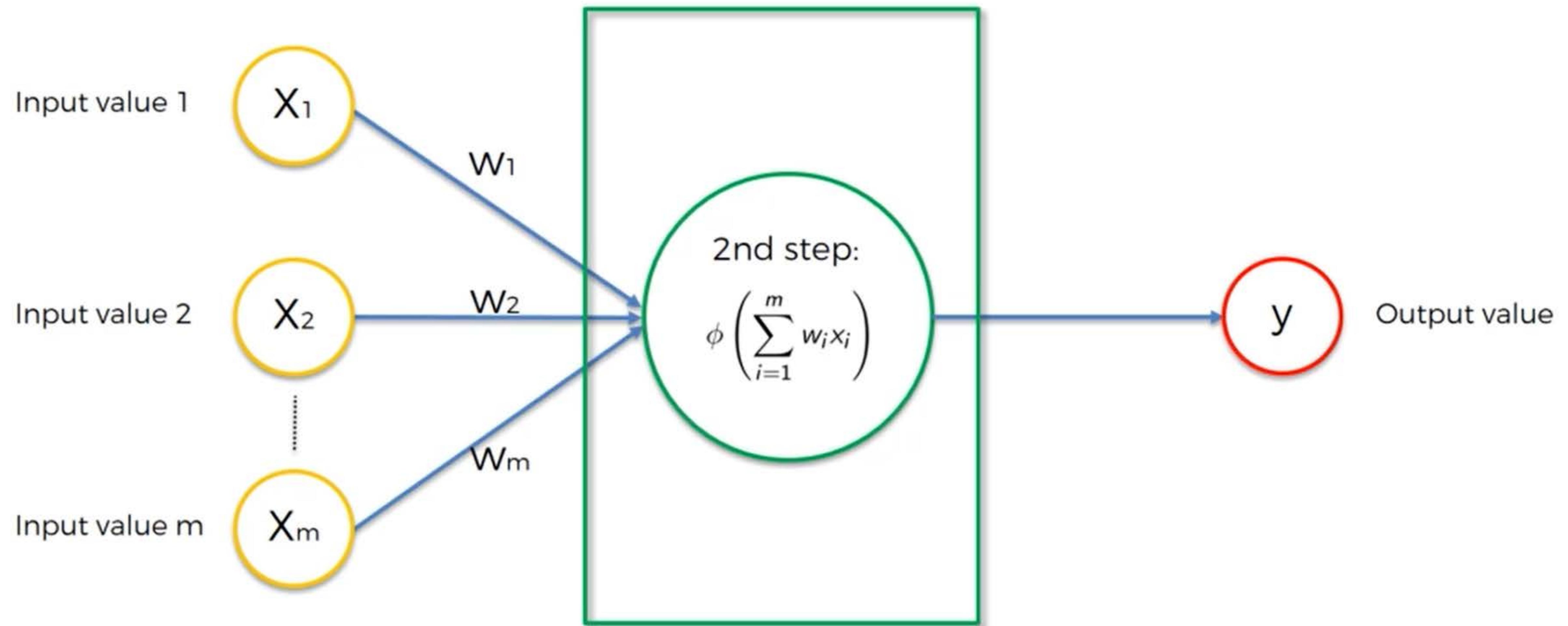
The Neuron



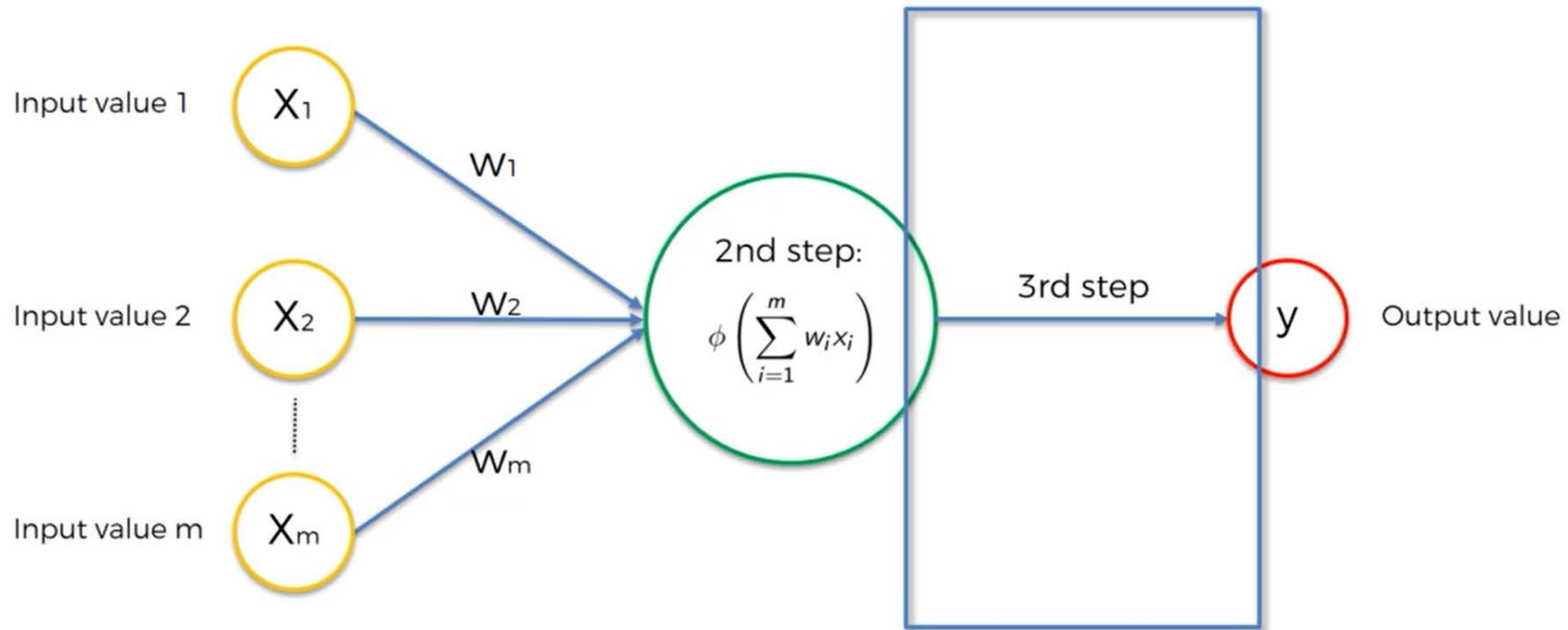
The Neuron



The Neuron

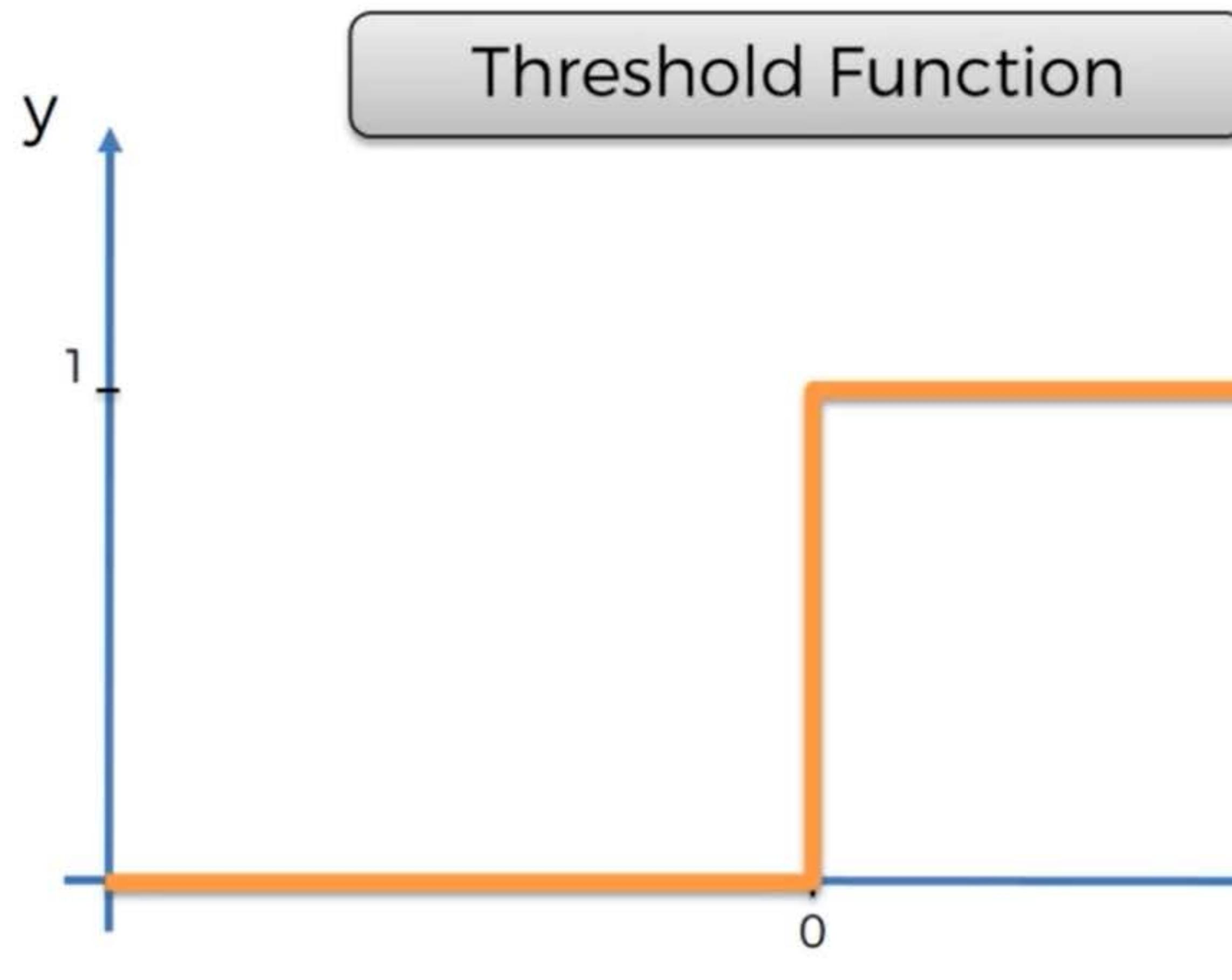


The Neuron



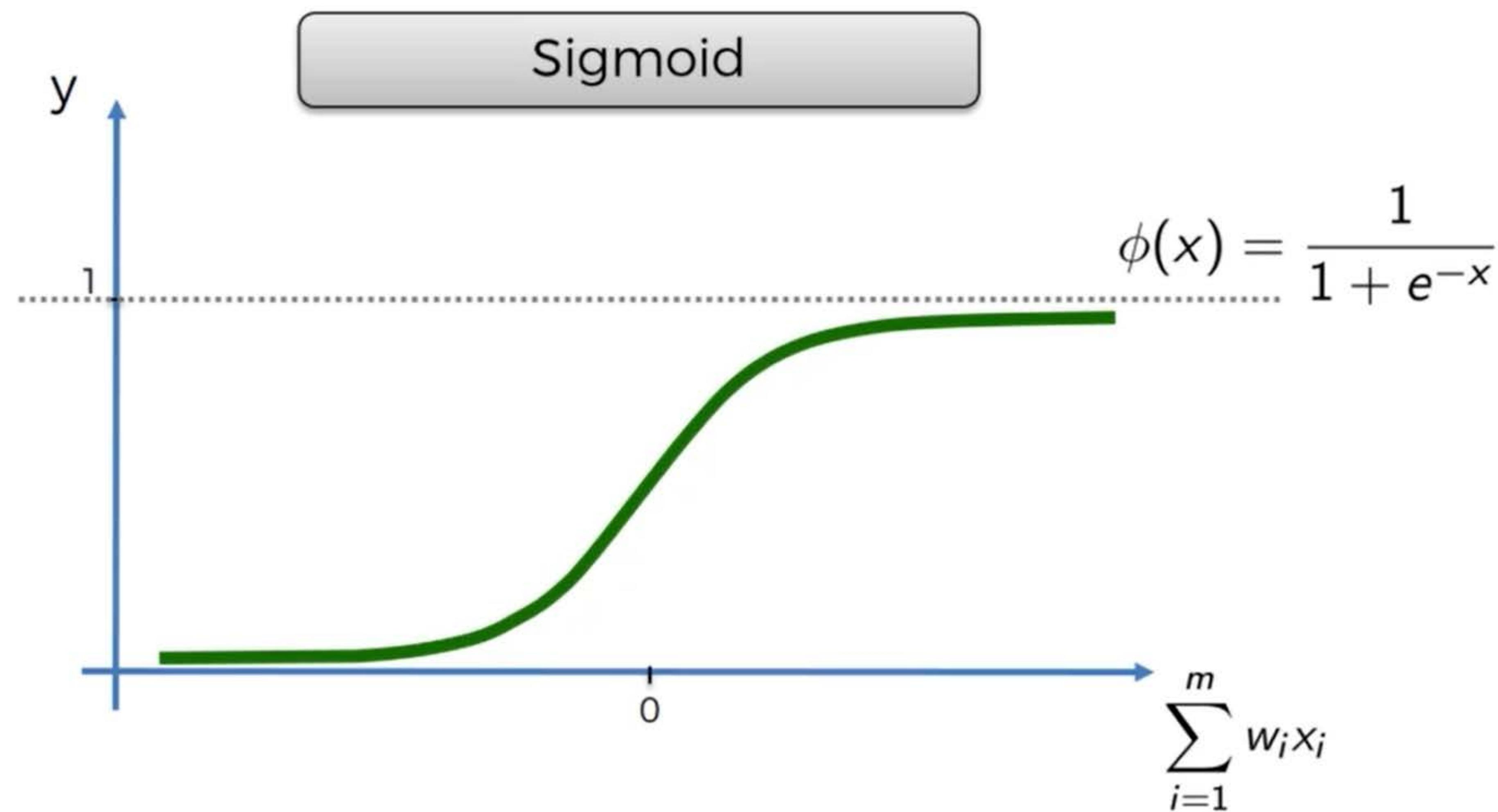
The Activation Function

The Activation Function

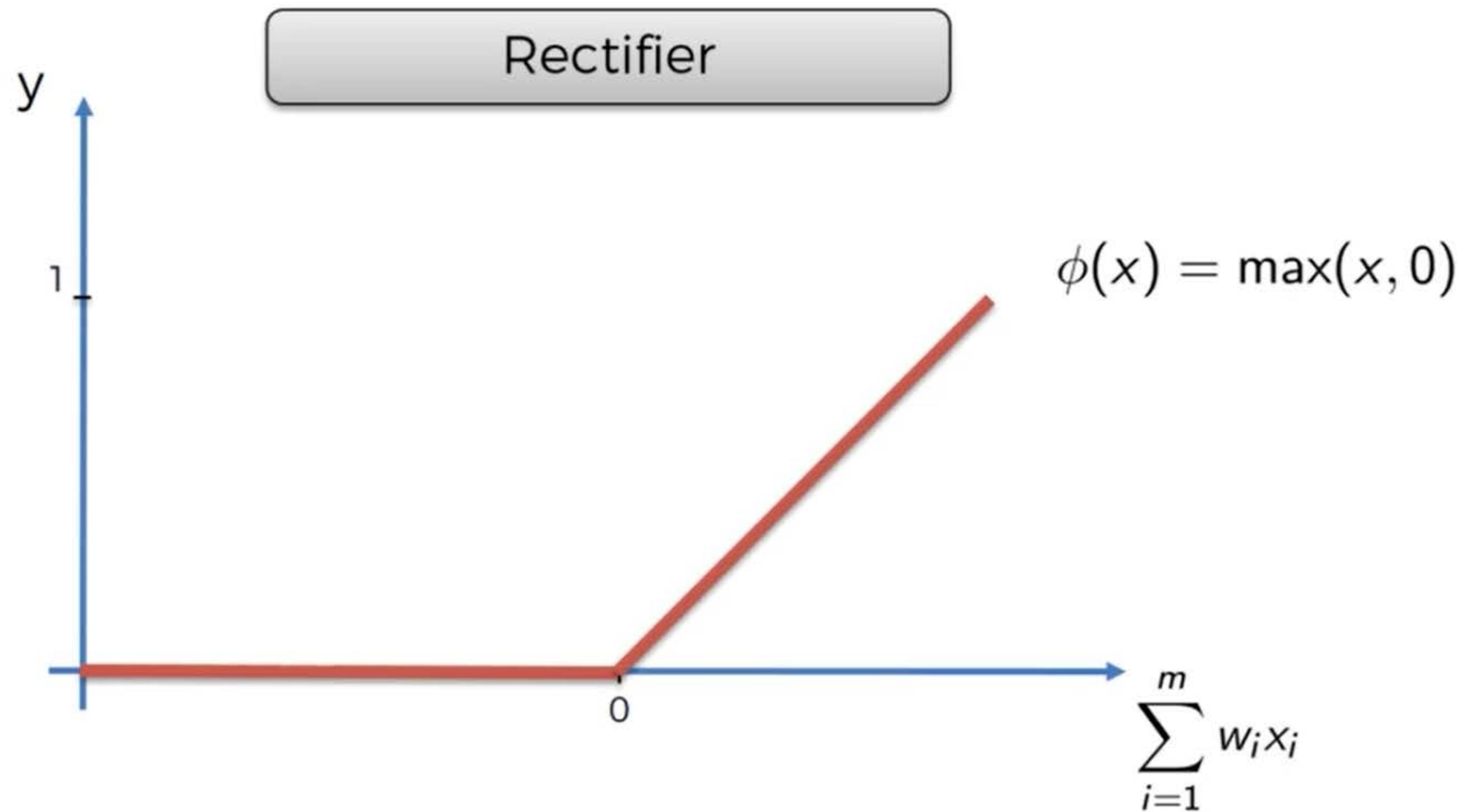


$$\phi(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$
$$\sum_{i=1}^m w_i x_i$$

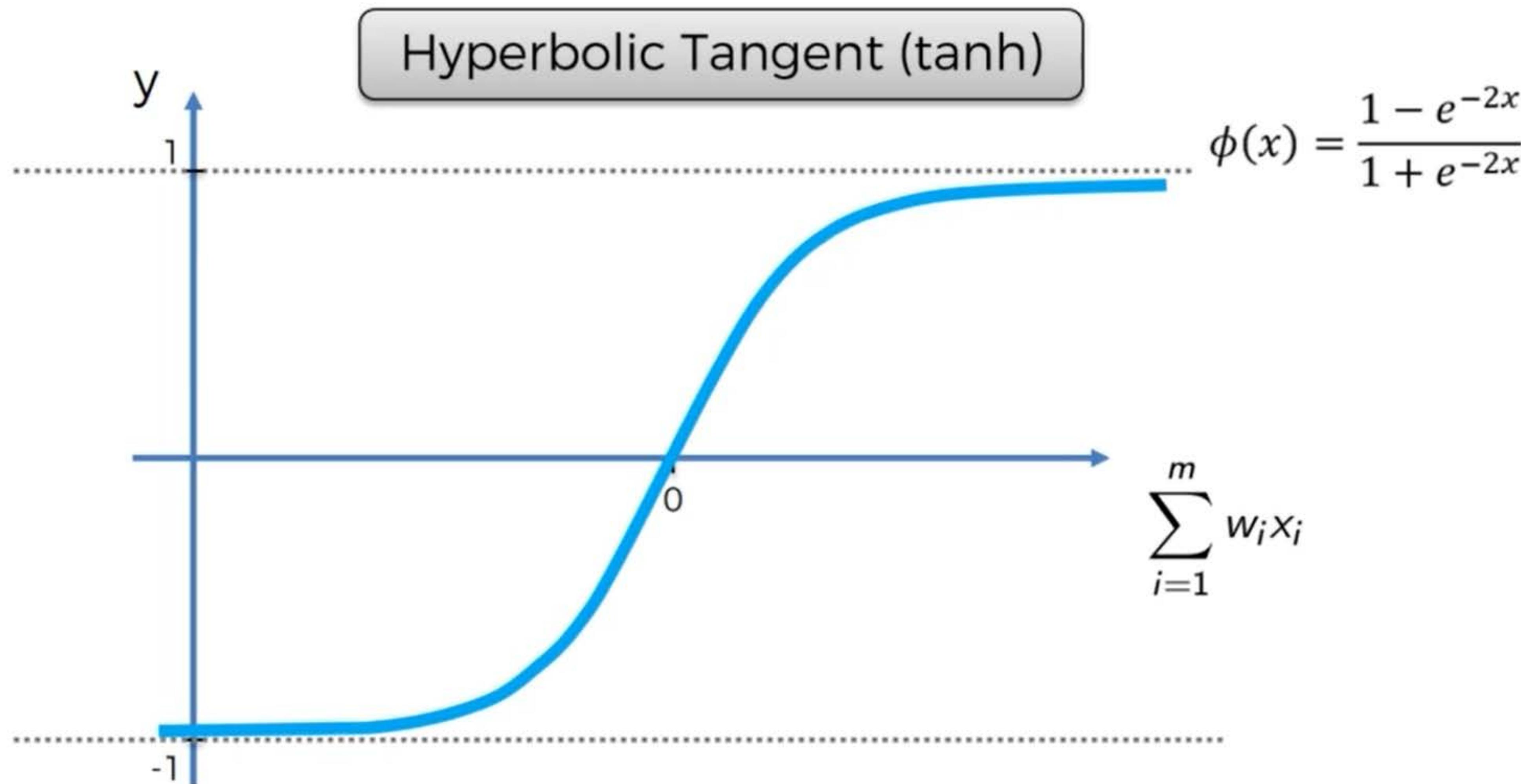
The Activation Function



The Activation Function



The Activation Function

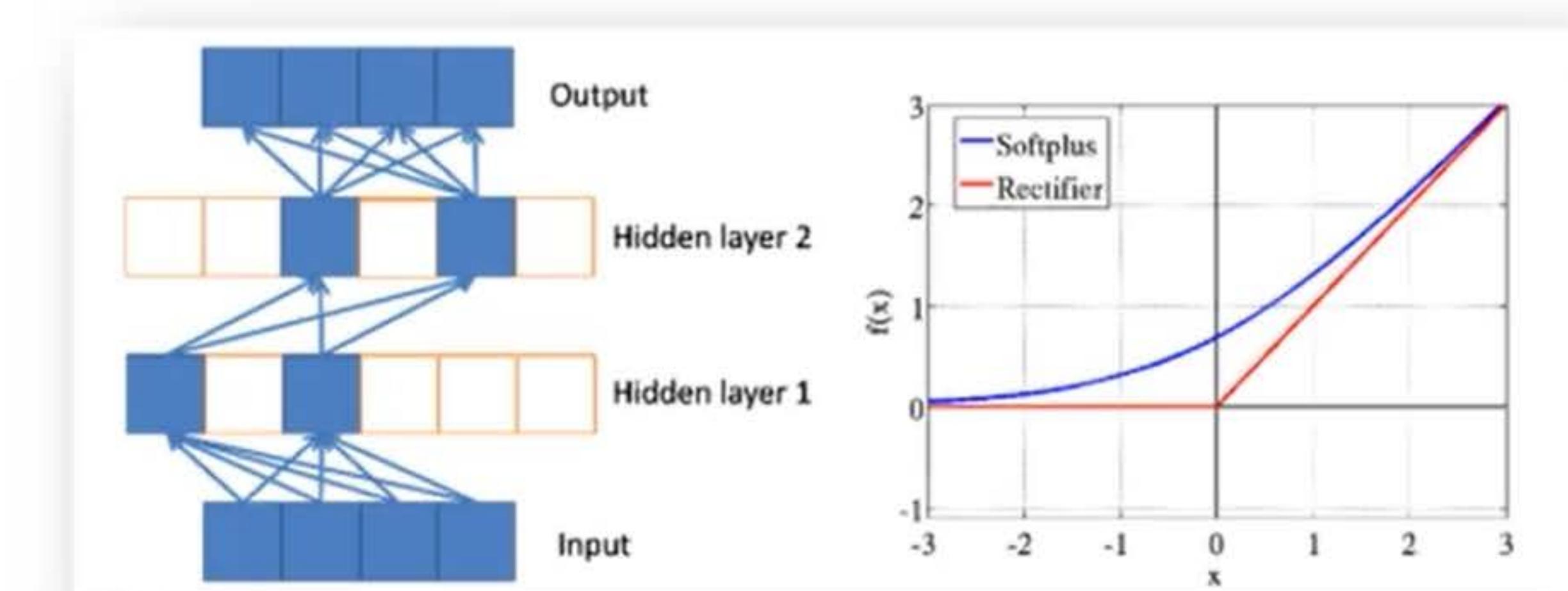


The Activation Function

Additional Reading:

Deep sparse rectifier neural networks

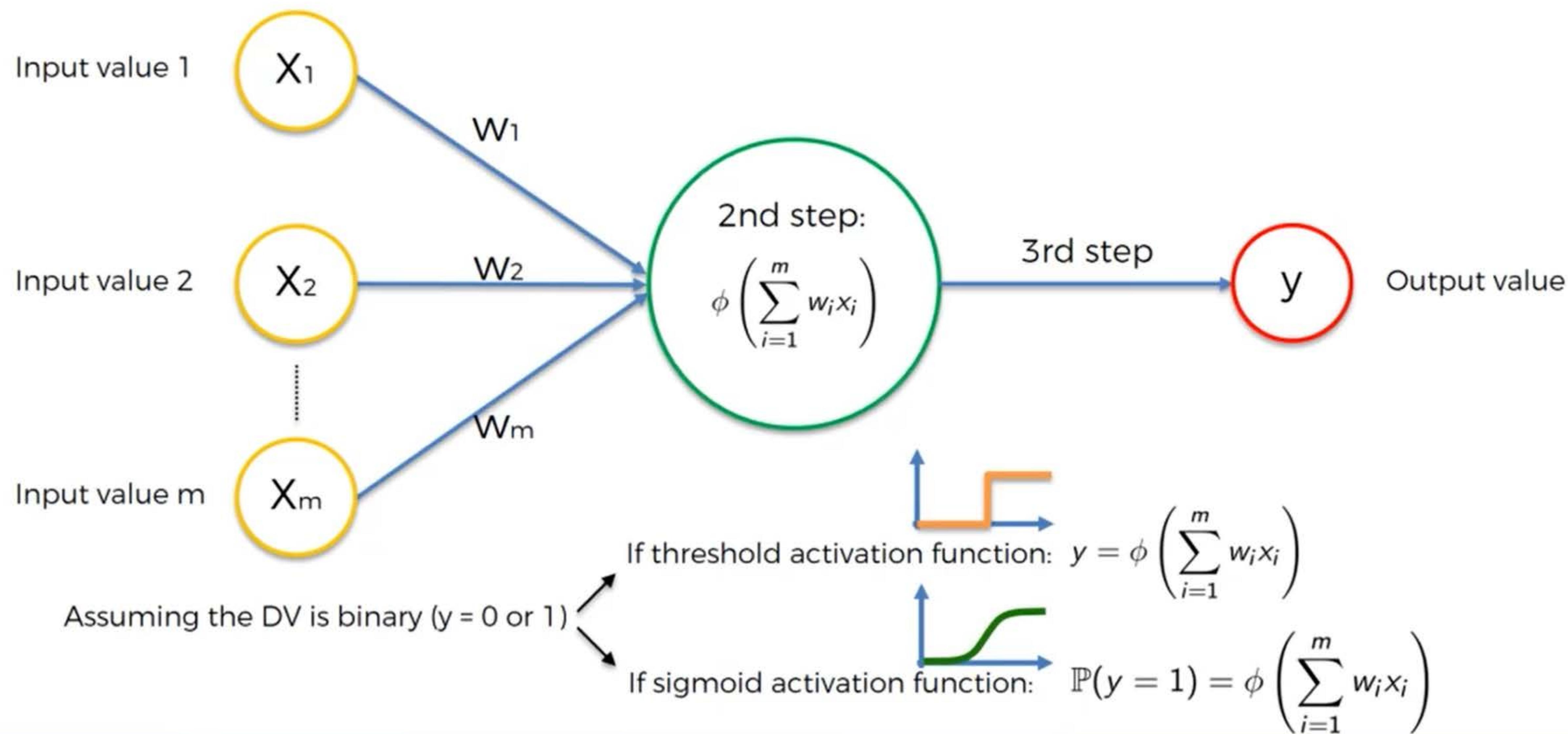
By Xavier Glorot et al. (2011)



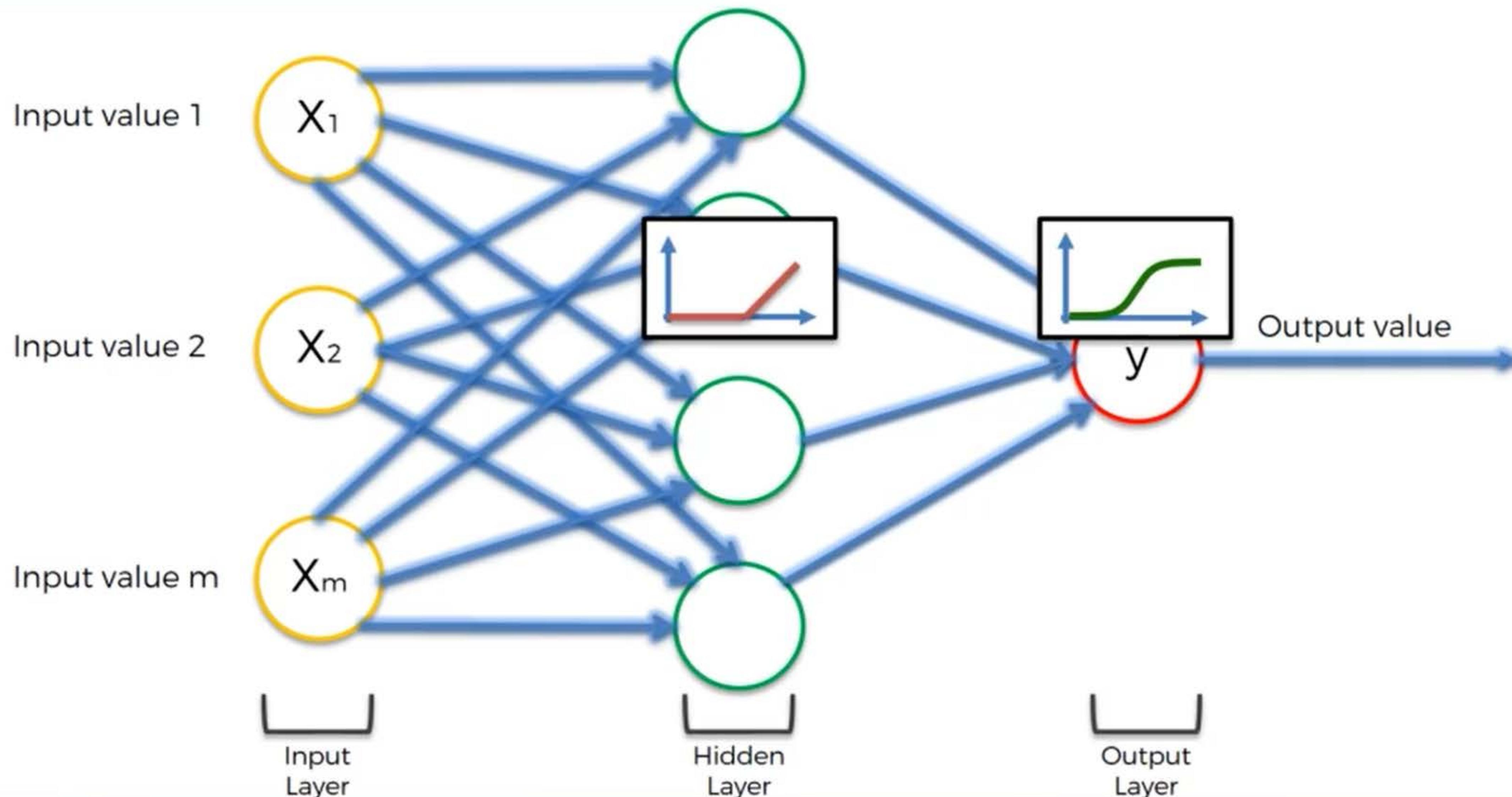
Link:

<http://jmlr.org/proceedings/papers/v15/glorot11a/glorot11a.pdf>

The Activation Function



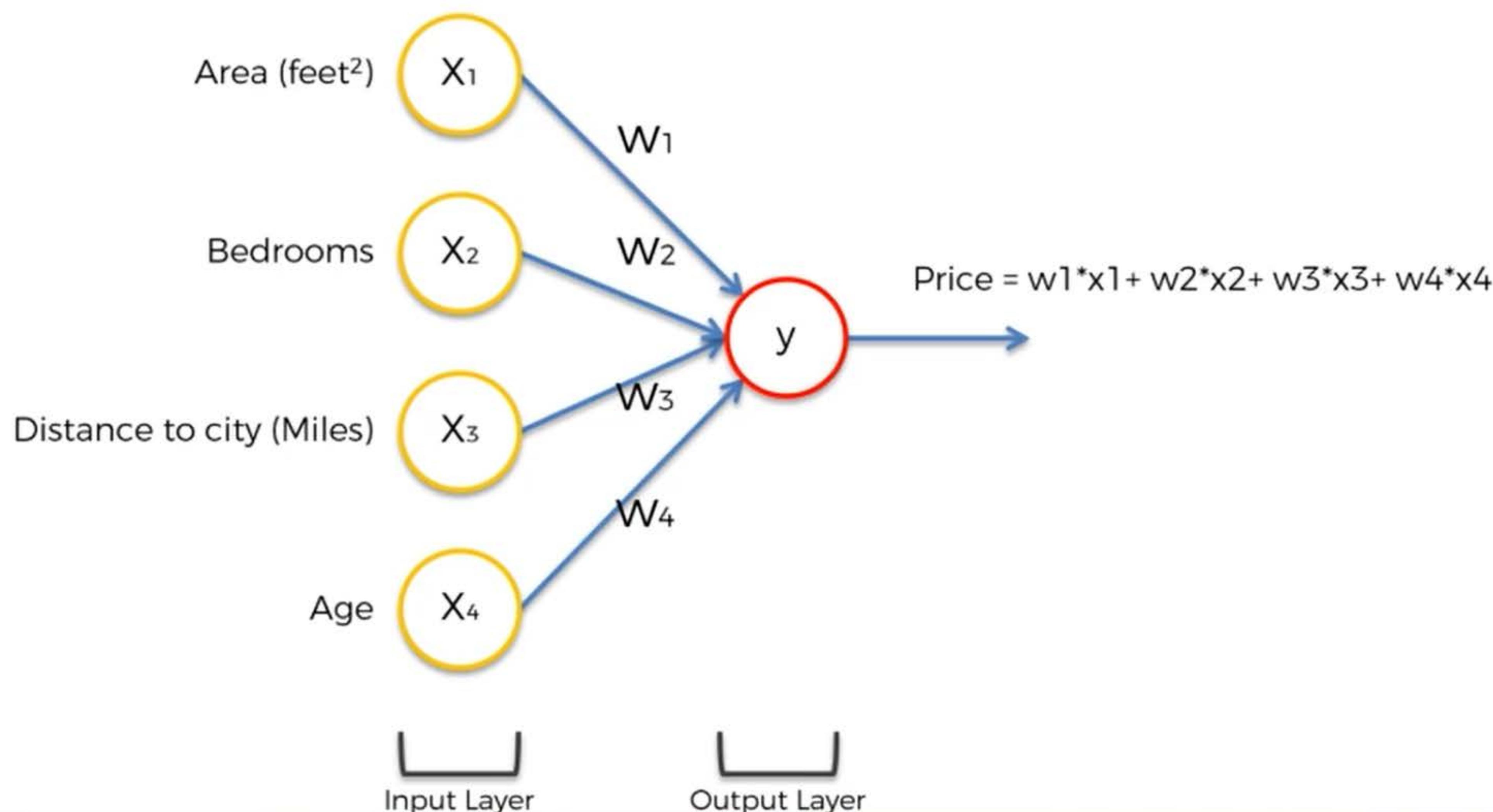
The Activation Function



How do NNs Work?



How Do Neural Networks Work?



How Do Neural Networks Work?



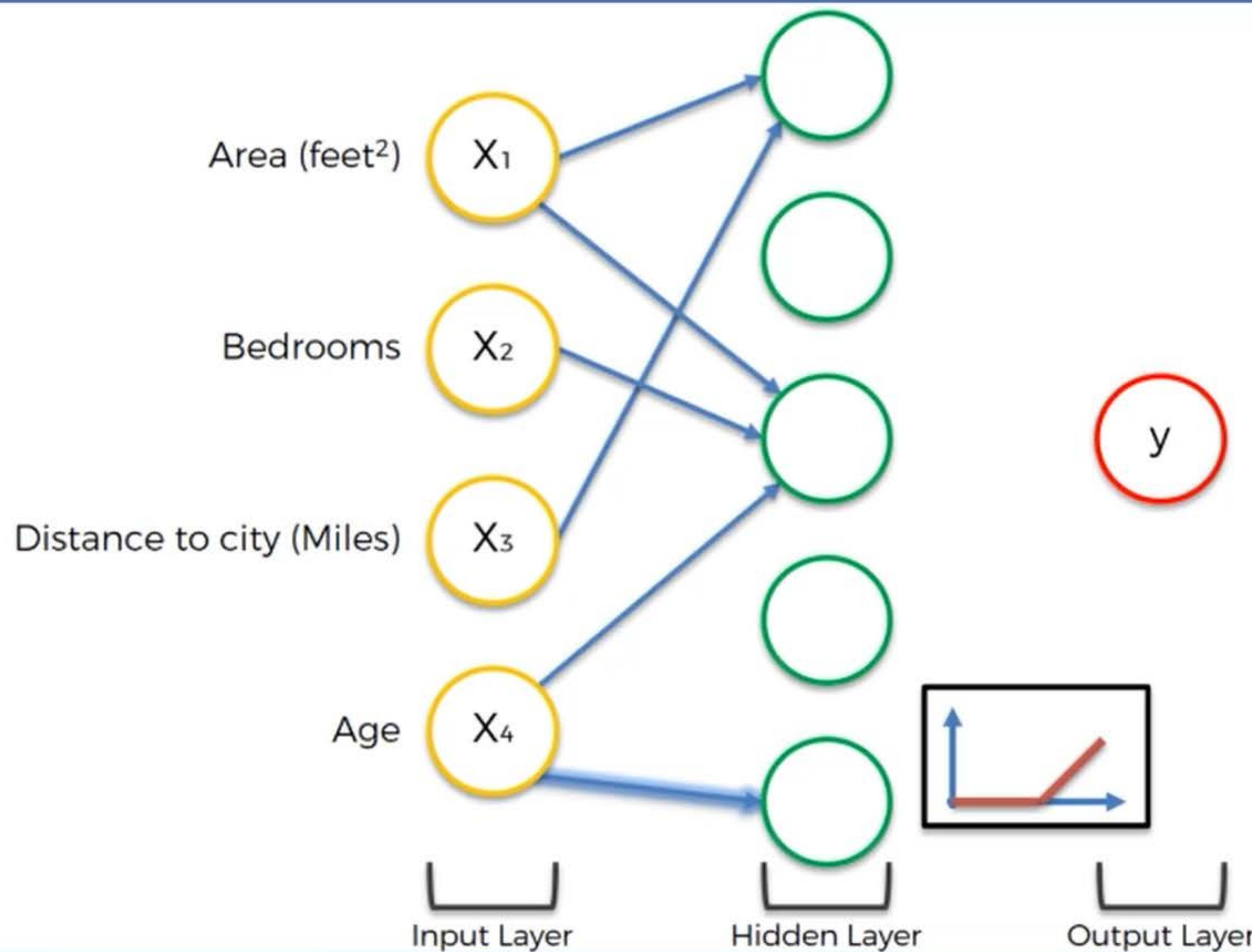
How Do Neural Networks Work?



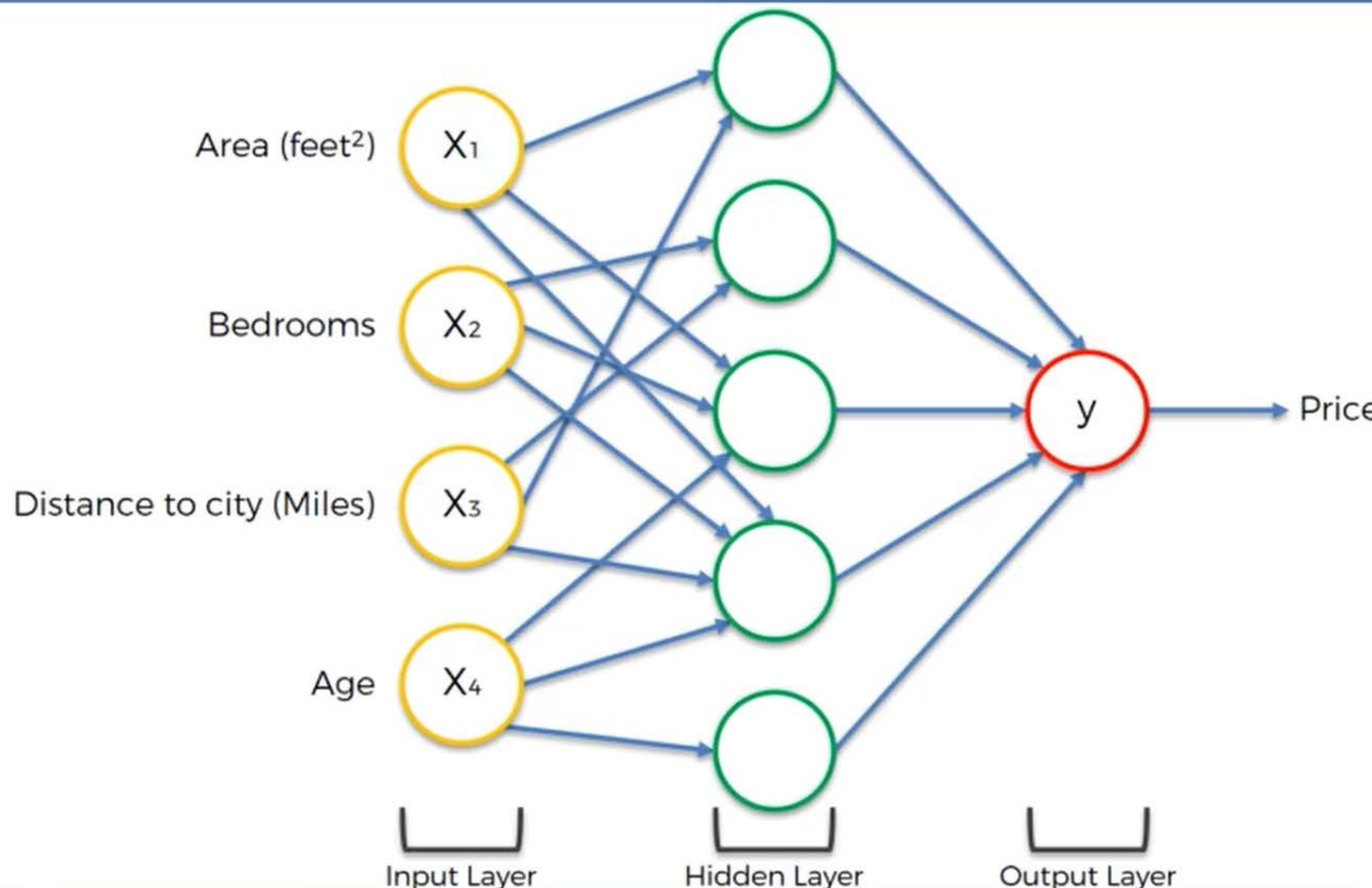
How Do Neural Networks Work?



How Do Neural Networks Work?



How Do Neural Networks Work?



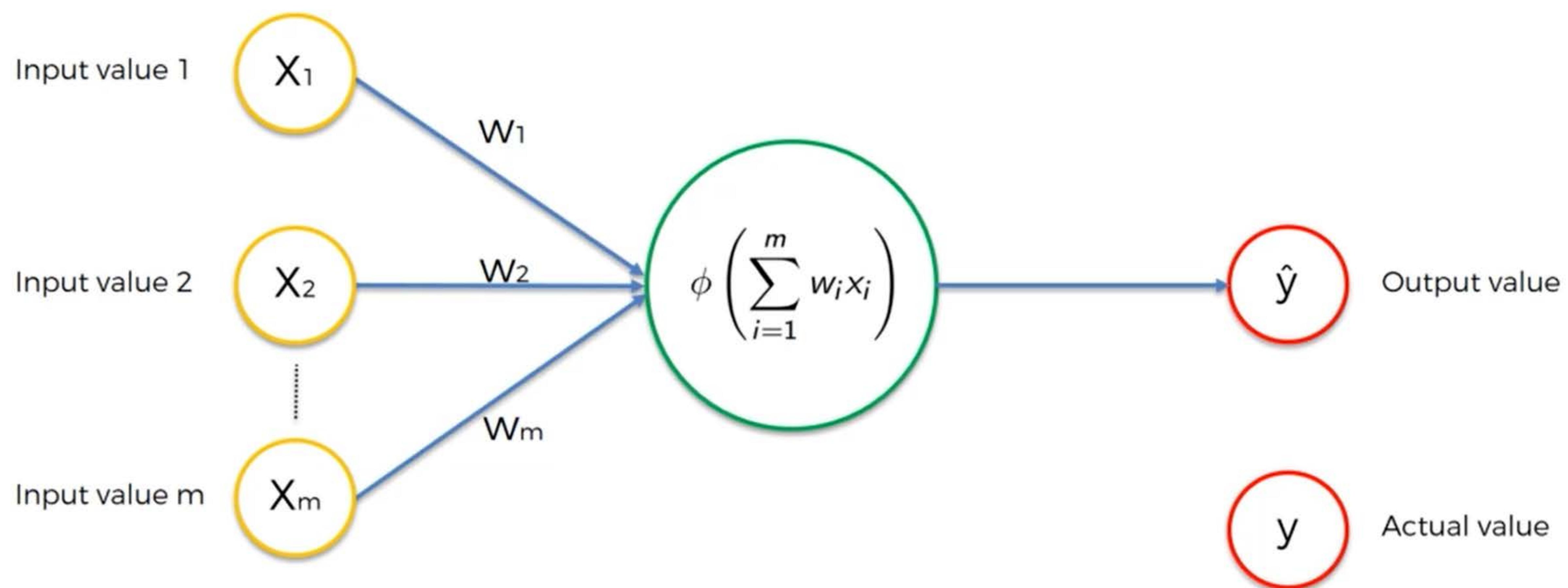
How do NNs Learn?

How do Neural Networks learn?

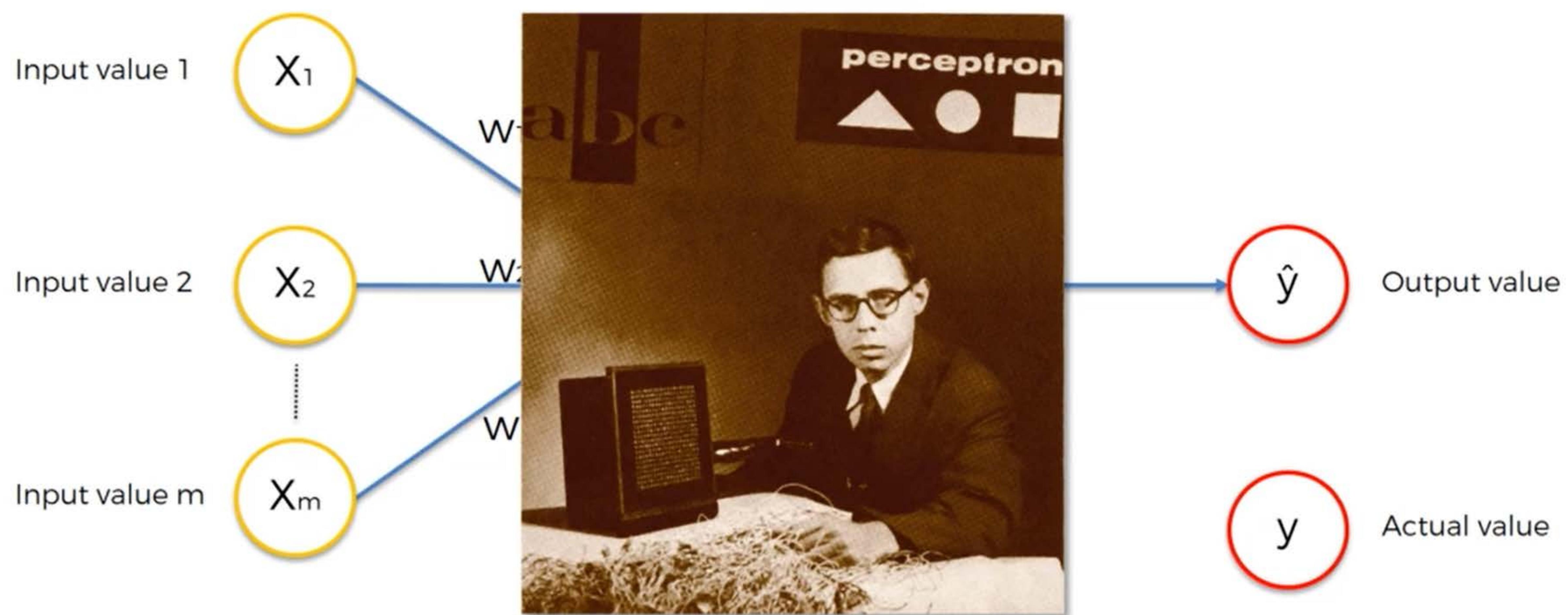
```
function use_array(a, b) {  
    for (let i = -1; i < a.length; i++) {  
        if (i < 0) {  
            let c = a[i];  
            if (c === undefined) {  
                c = 0;  
            }  
            if (b[i] === undefined) {  
                b[i] = c;  
            } else {  
                b[i] += c;  
            }  
        } else {  
            let d = a[i];  
            if (d === undefined) {  
                d = 0;  
            }  
            if (b[i] === undefined) {  
                b[i] = d;  
            } else {  
                b[i] += d;  
            }  
        }  
    }  
    return b;  
}  
  
function use_array(a, b) {  
    for (let i = -1; i < a.length; i++) {  
        if (i < 0) {  
            let c = a[i];  
            if (c === undefined) {  
                c = 0;  
            }  
            if (b[i] === undefined) {  
                b[i] = c;  
            } else {  
                b[i] += c;  
            }  
        } else {  
            let d = a[i];  
            if (d === undefined) {  
                d = 0;  
            }  
            if (b[i] === undefined) {  
                b[i] = d;  
            } else {  
                b[i] += d;  
            }  
        }  
    }  
    return b;  
}
```



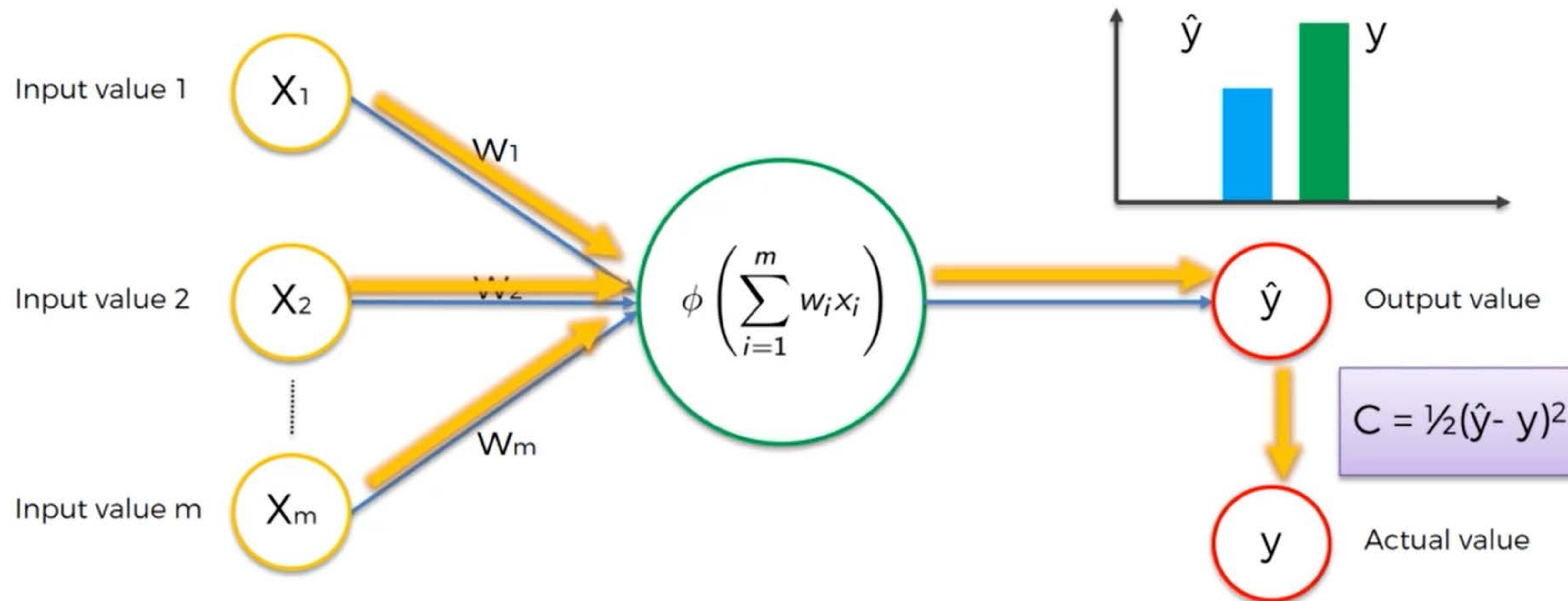
How do Neural Networks learn?



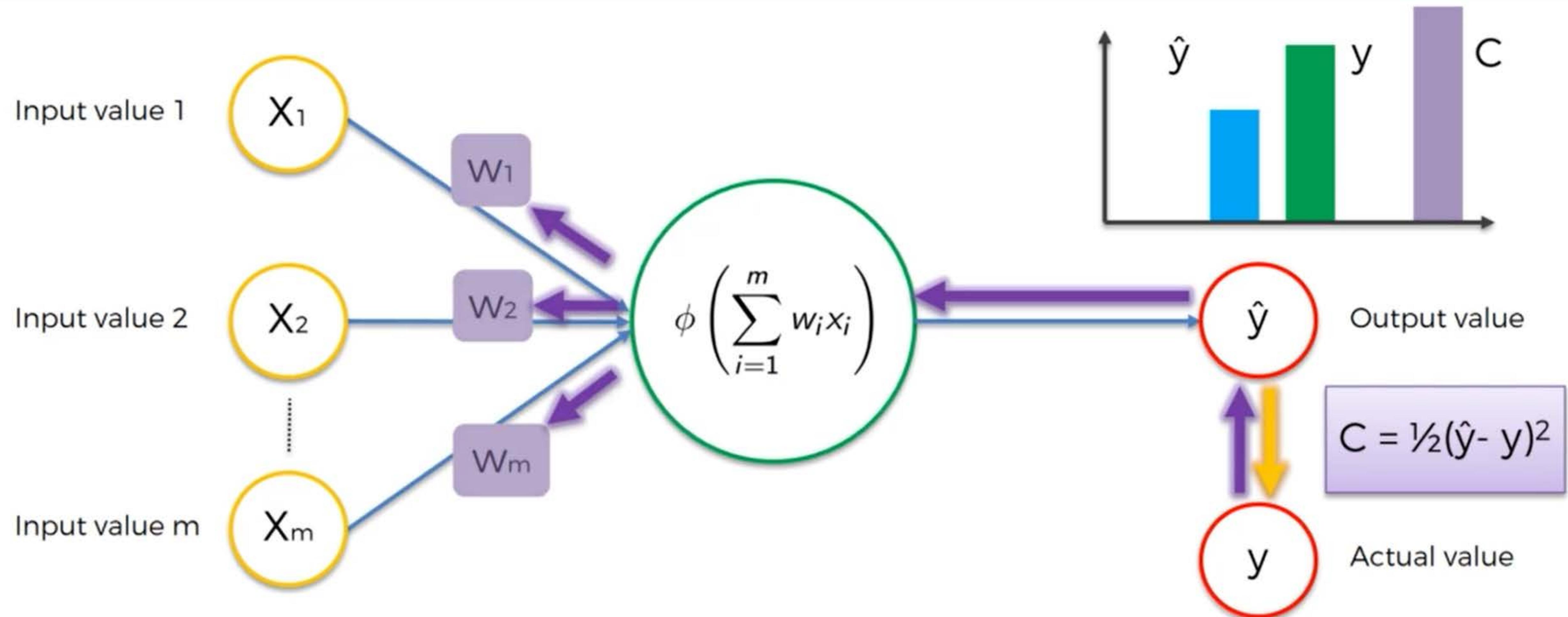
How do Neural Networks learn?



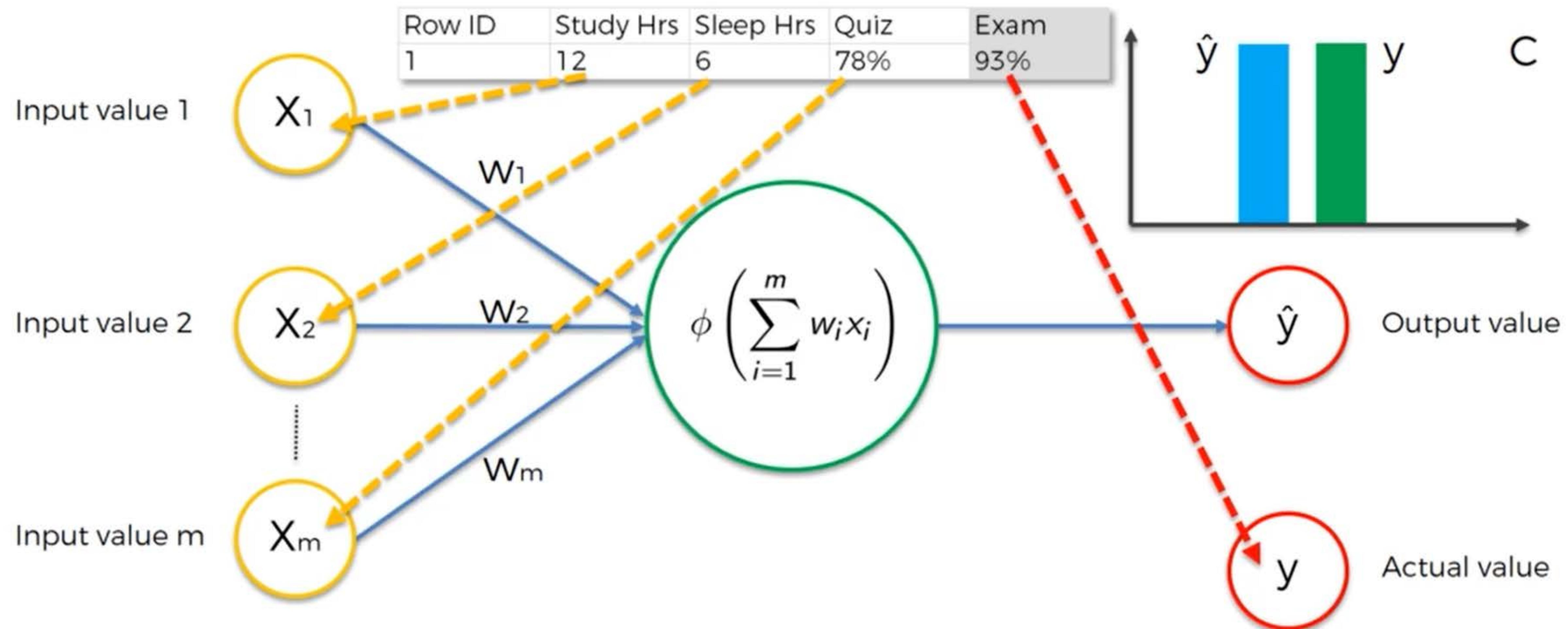
How do Neural Networks learn?



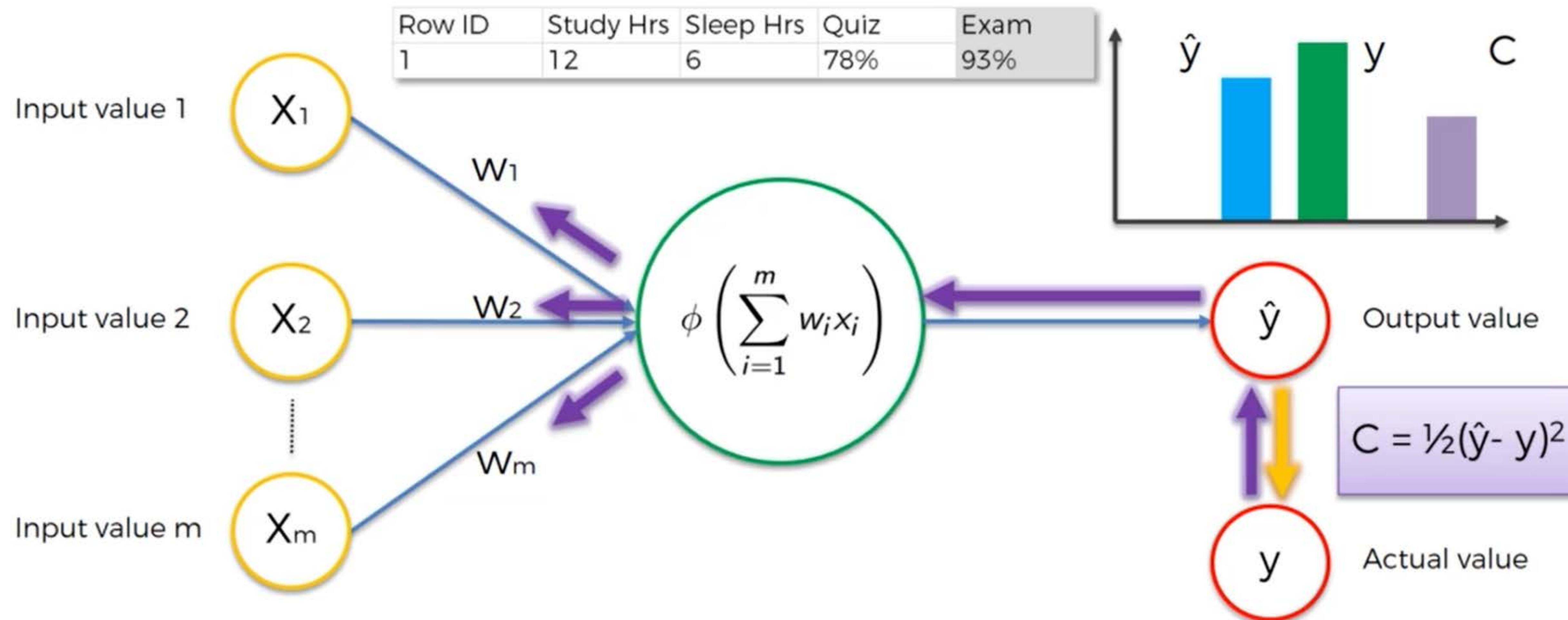
How do Neural Networks learn?



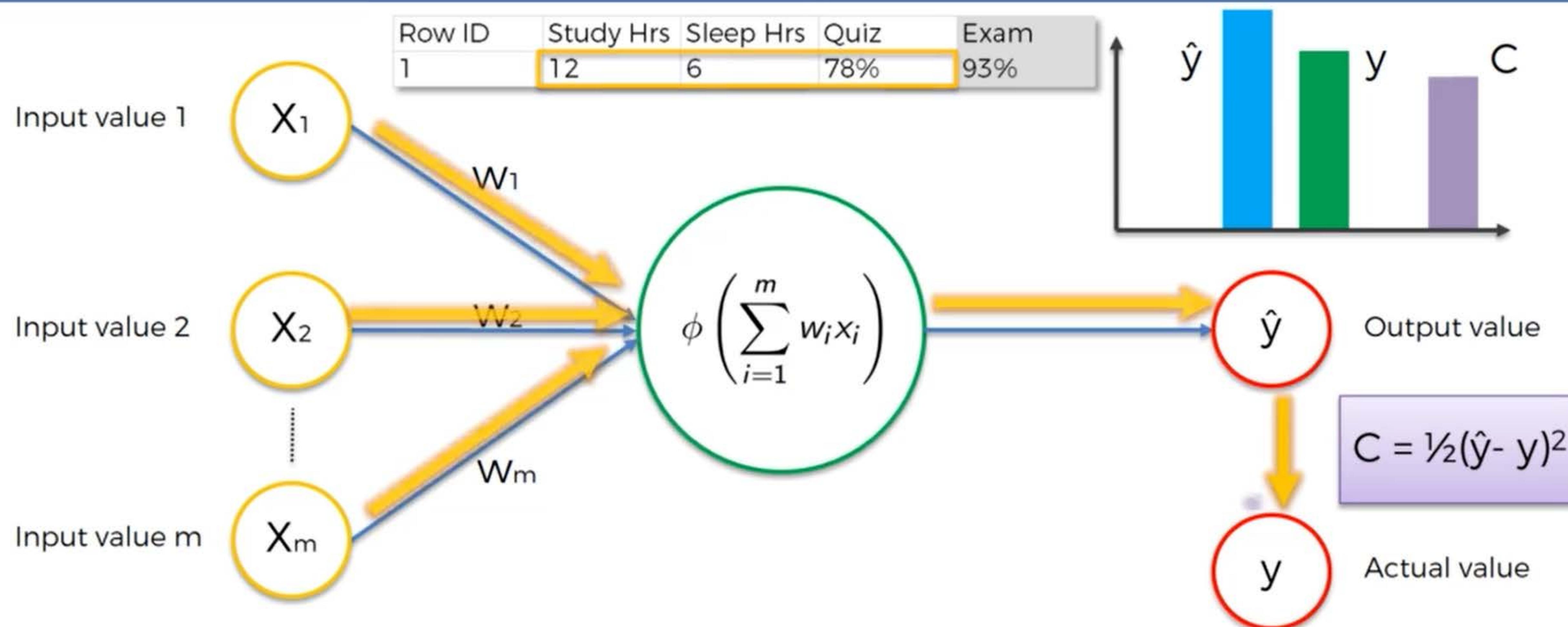
How do Neural Networks learn?



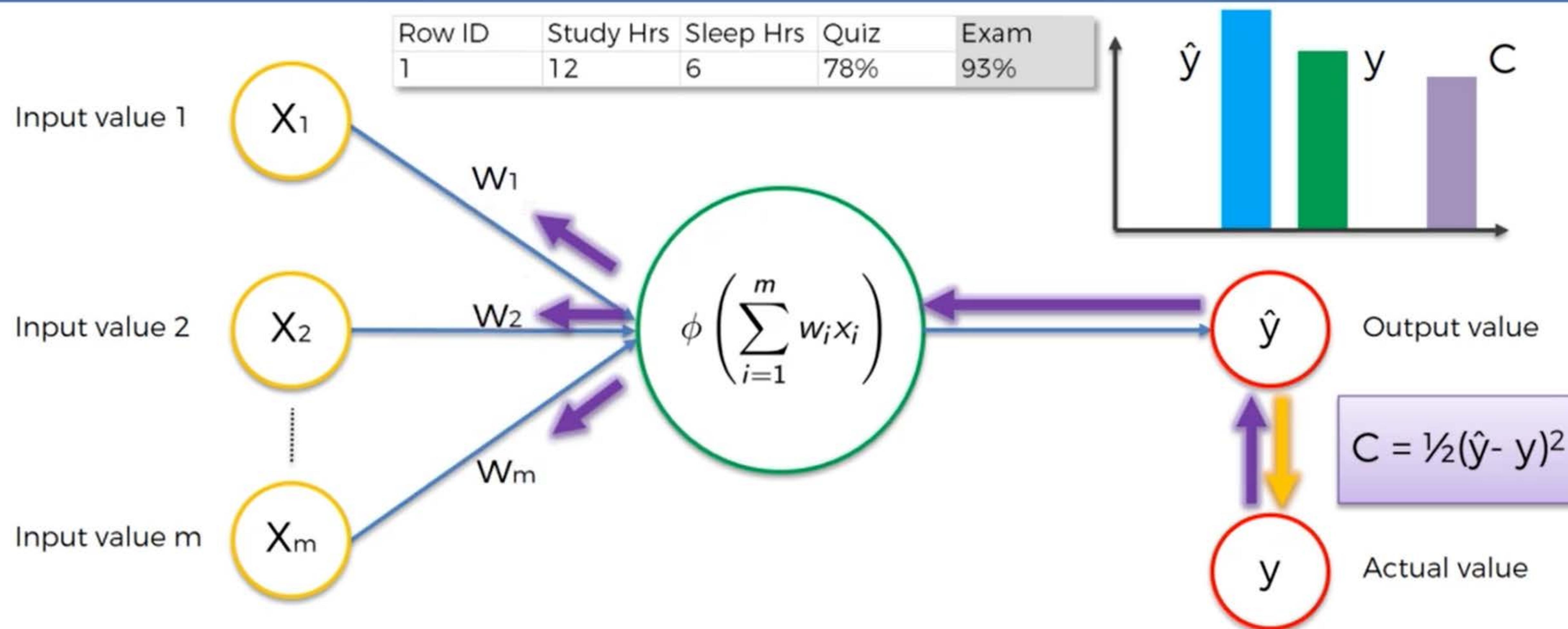
How do Neural Networks learn?



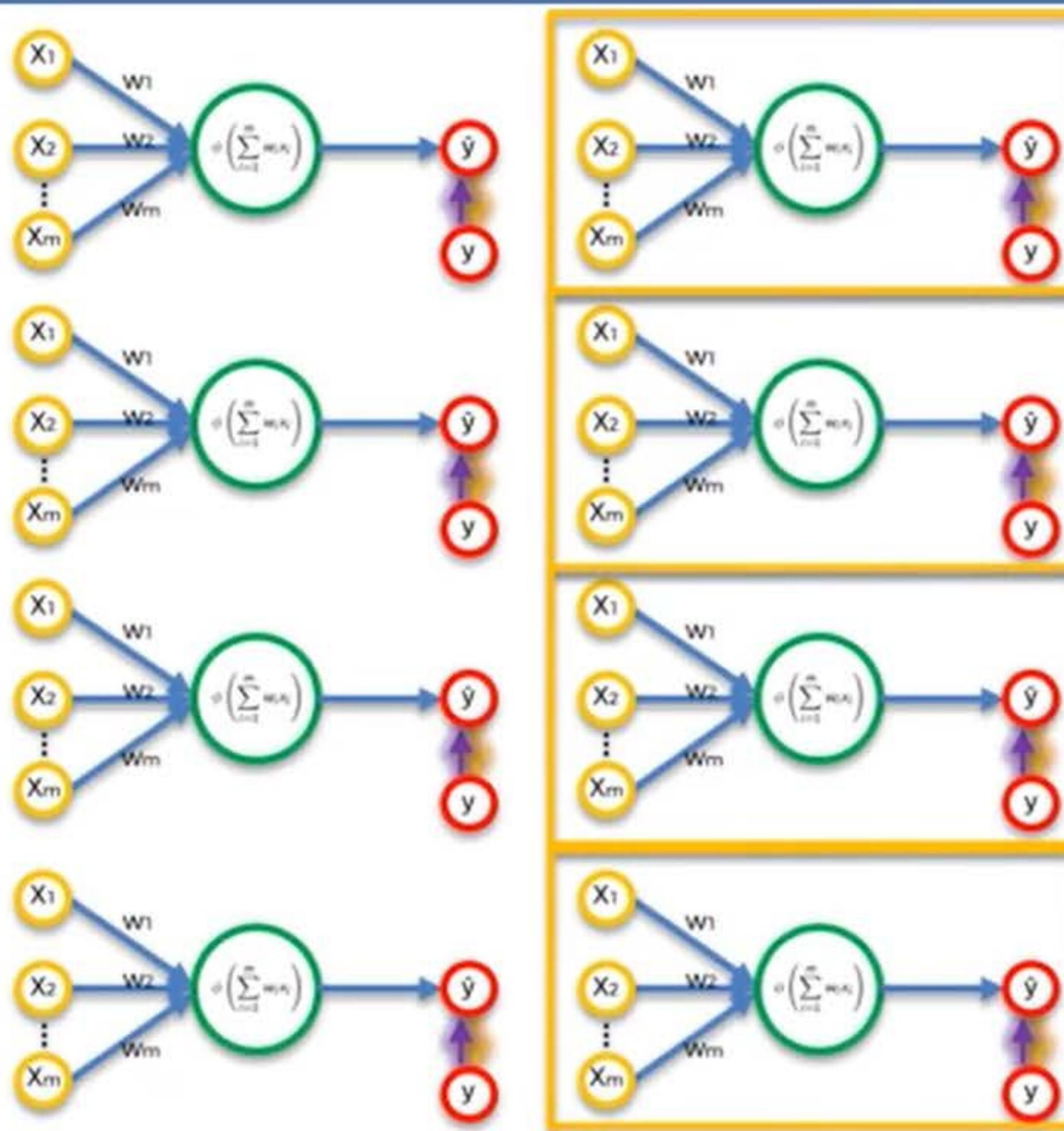
How do Neural Networks learn?



How do Neural Networks learn?



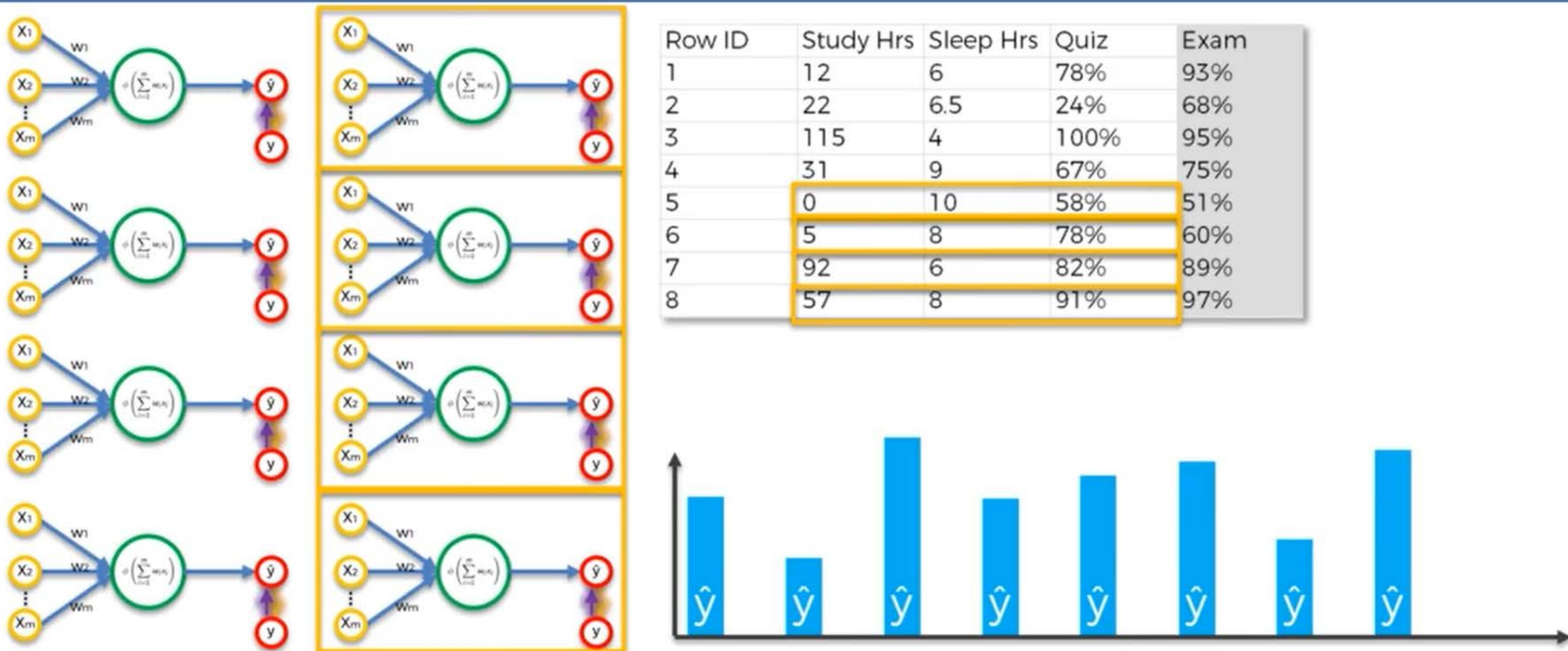
How do Neural Networks learn?



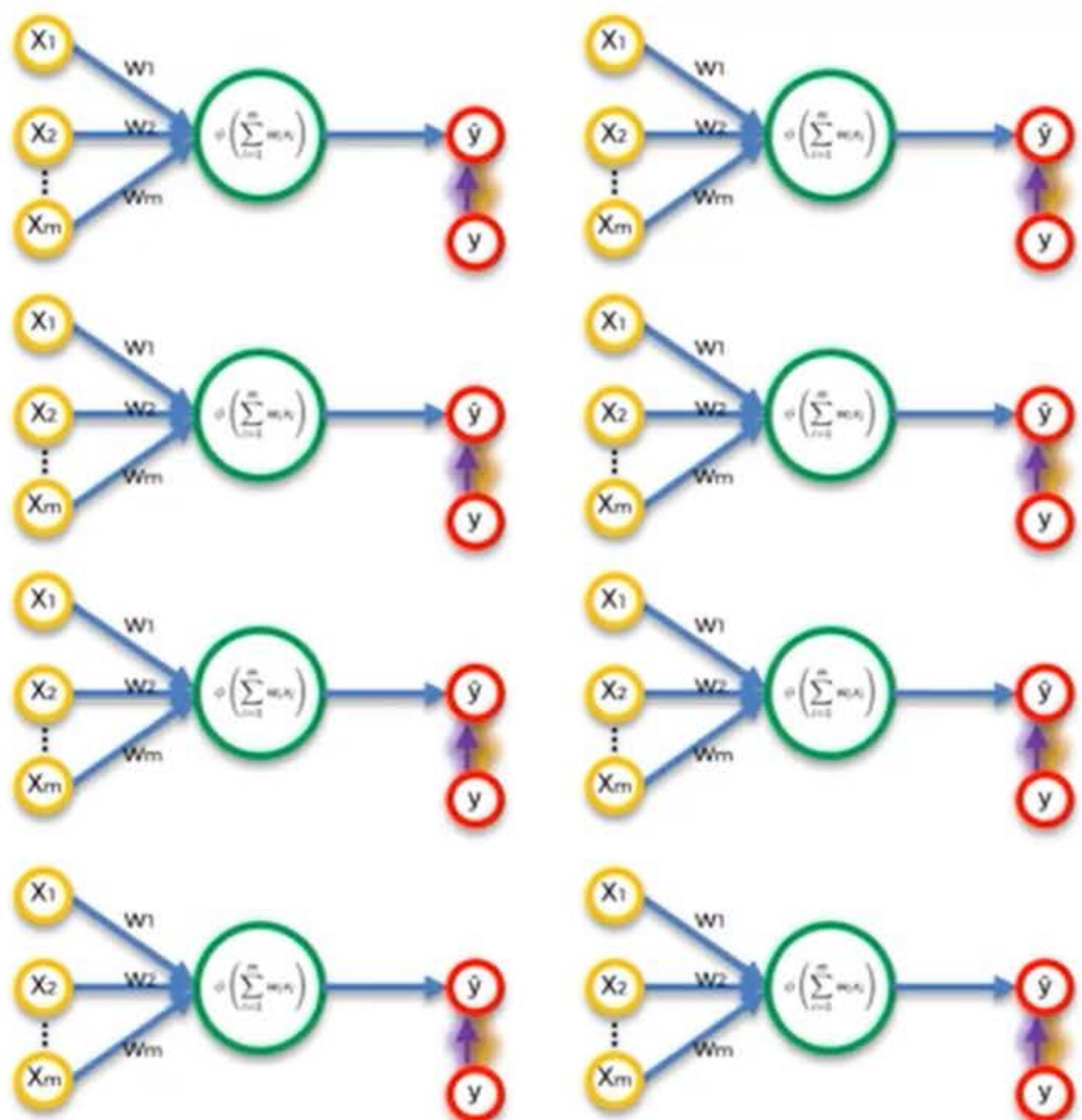
Row ID	Study Hrs	Sleep Hrs	Quiz	Exam
1	12	6	78%	93%
2	22	6.5	24%	68%
3	115	4	100%	95%
4	31	9	67%	75%
5	0	10	58%	51%
6	5	8	78%	60%
7	92	6	82%	89%
8	57	8	91%	97%



How do Neural Networks learn?



How do Neural Networks learn?



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%

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



How do Neural Networks learn?

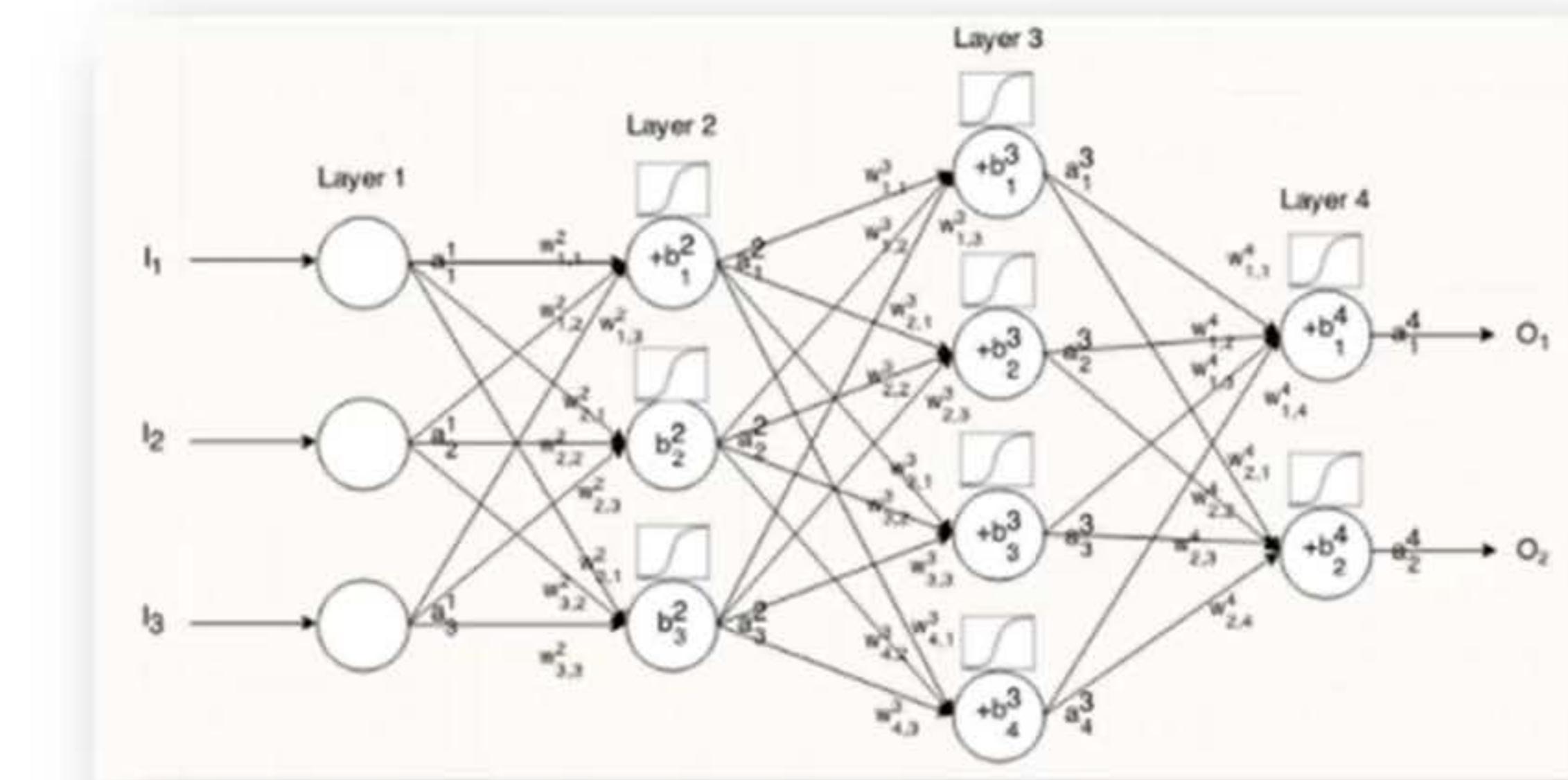
Additional Reading:

A list of cost functions used in neural networks, alongside applications

CrossValidated (2015)

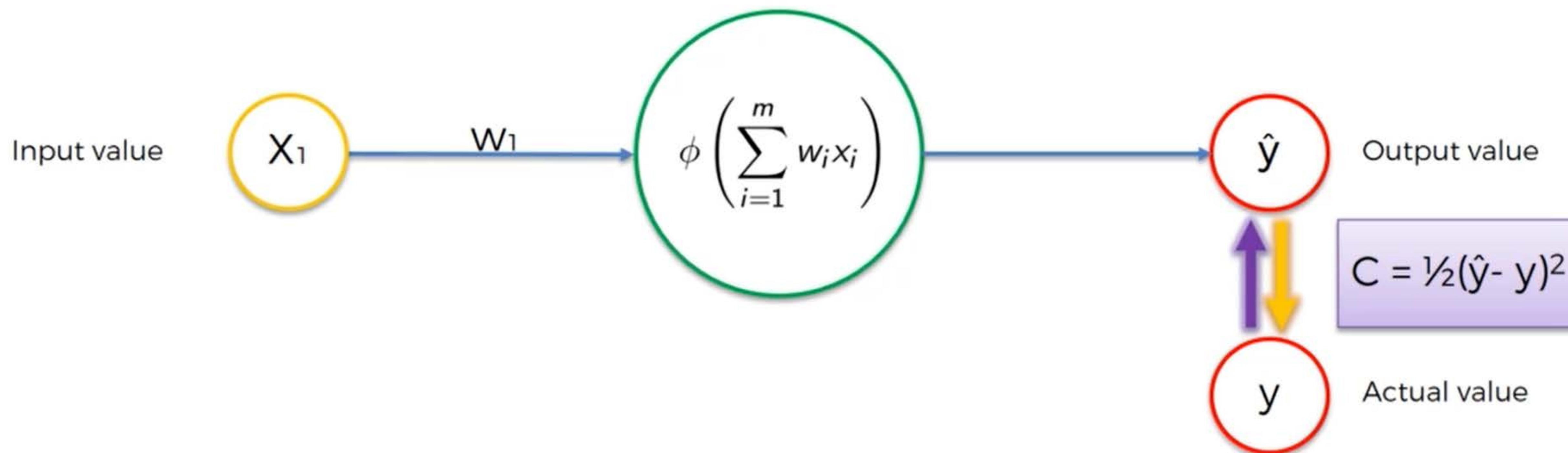
Link:

<http://stats.stackexchange.com/questions/154879/a-list-of-cost-functions-used-in-neural-networks-alongside-applications>

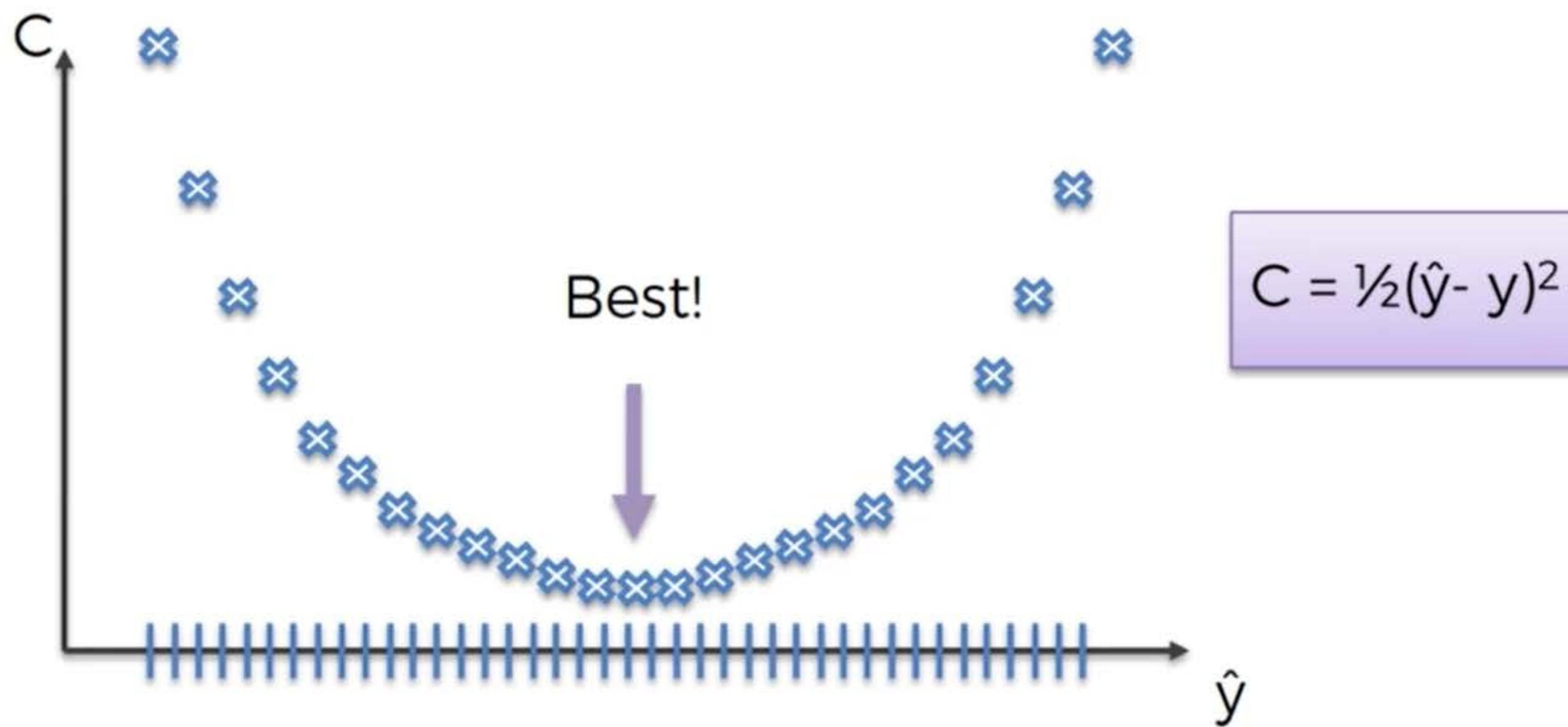


Gradient Descent

Gradient Descent

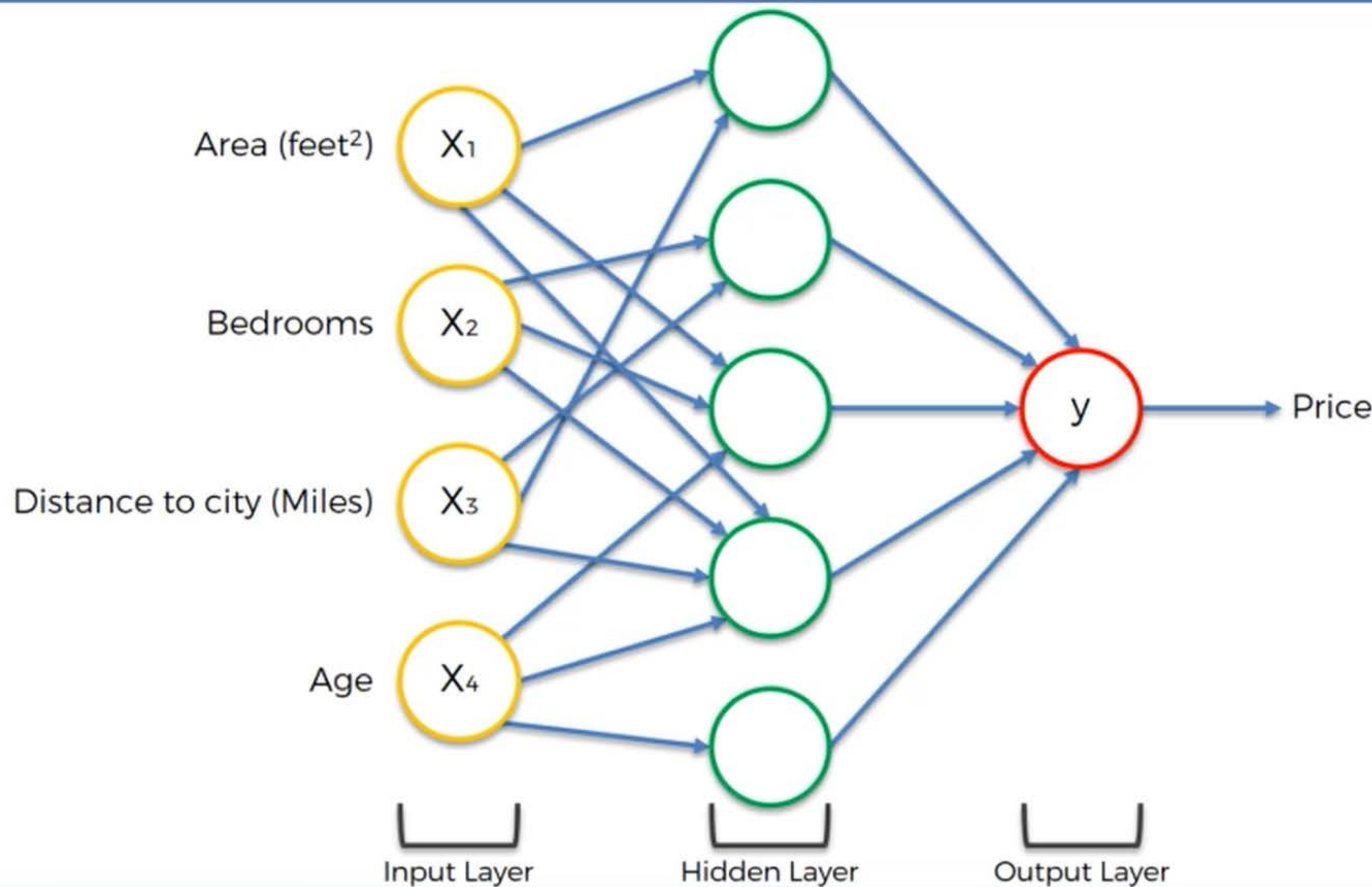


Gradient Descent

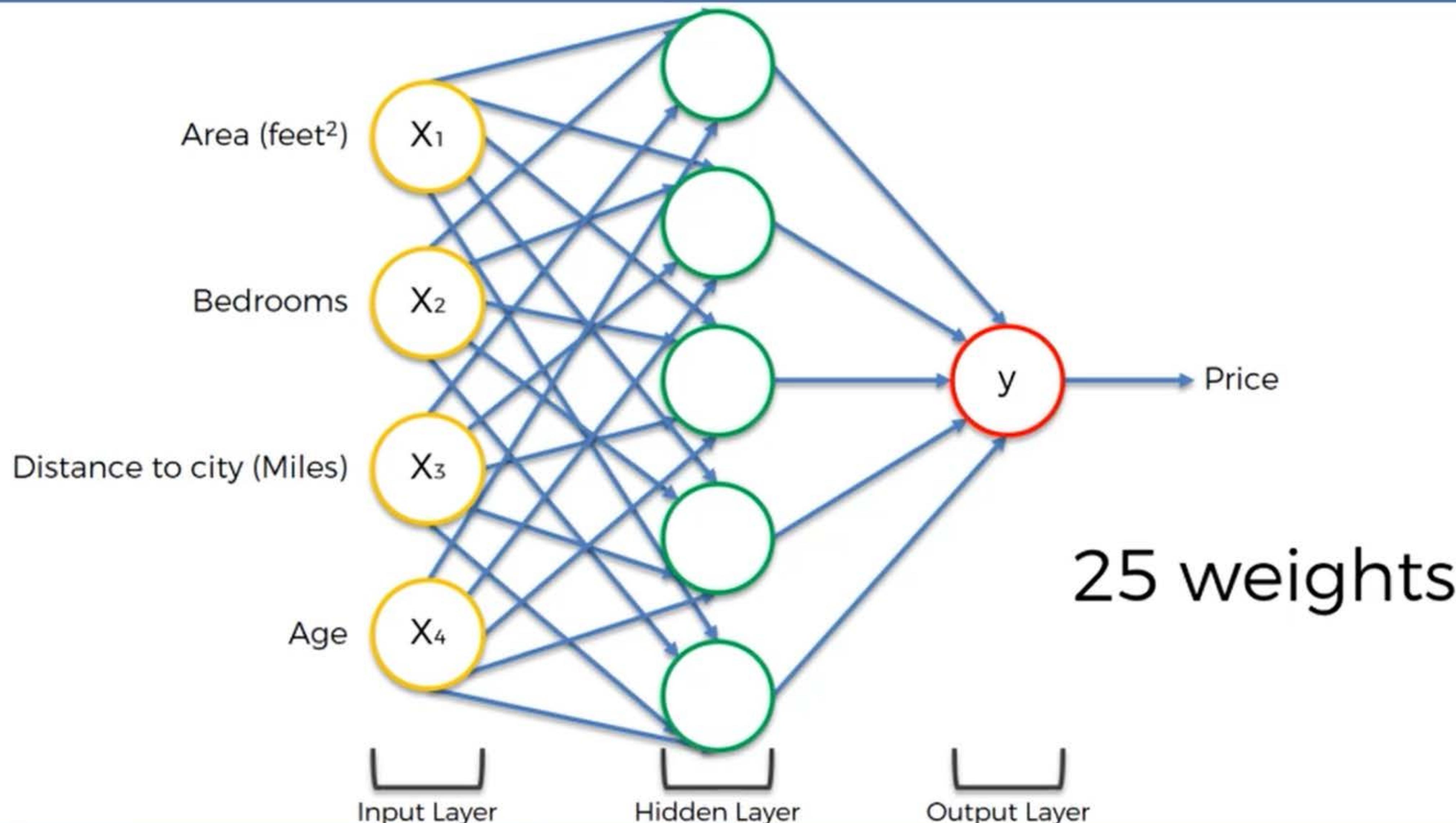


Curse of Dimensionality

Gradient Descent



Gradient Descent



Gradient Descent

$1,000 \times 1,000 \times \dots \times 1,000 = 1,000^{25} = 10^{75}$ combinations

Sunway TaihuLight: World's fastest Super Computer

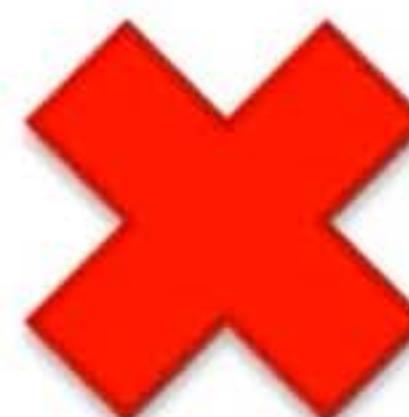
93 PFLOPS

93×10^{15}

$10^{75} / (93 \times 10^{15})$

$= 1.08 \times 10^{58}$ seconds

$= 3.42 \times 10^{50}$ years



Gradient Descent

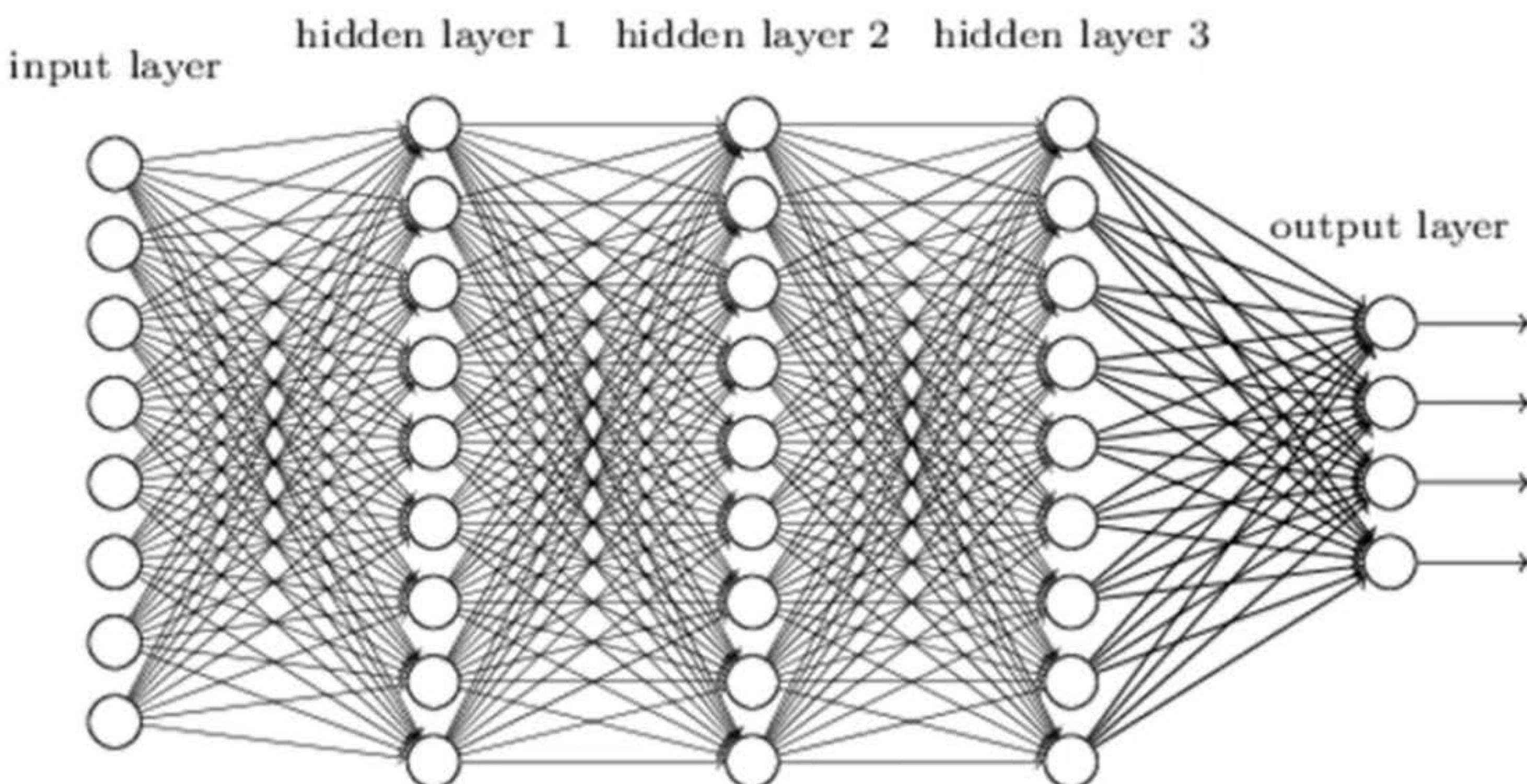
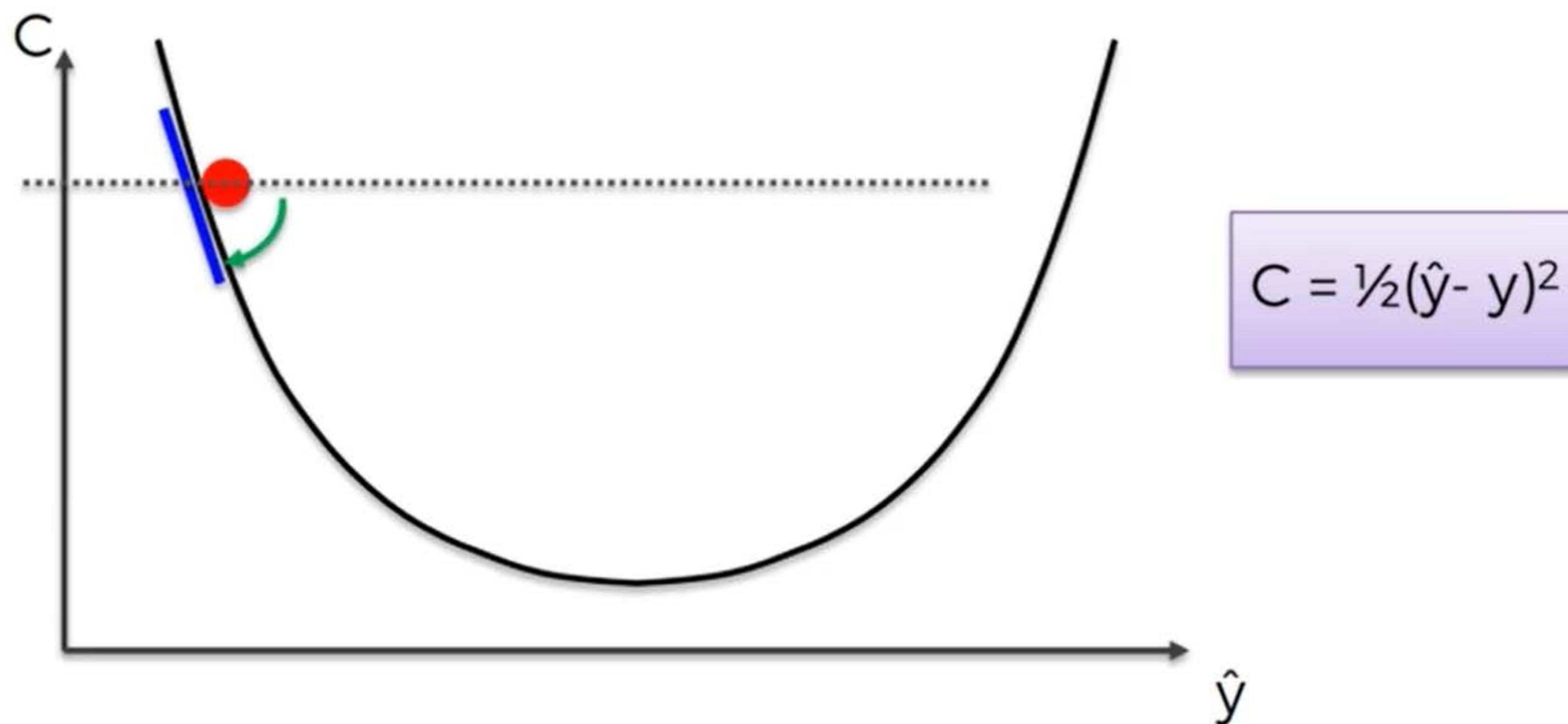


Image Source: neuralnetworksanddeeplearning.com

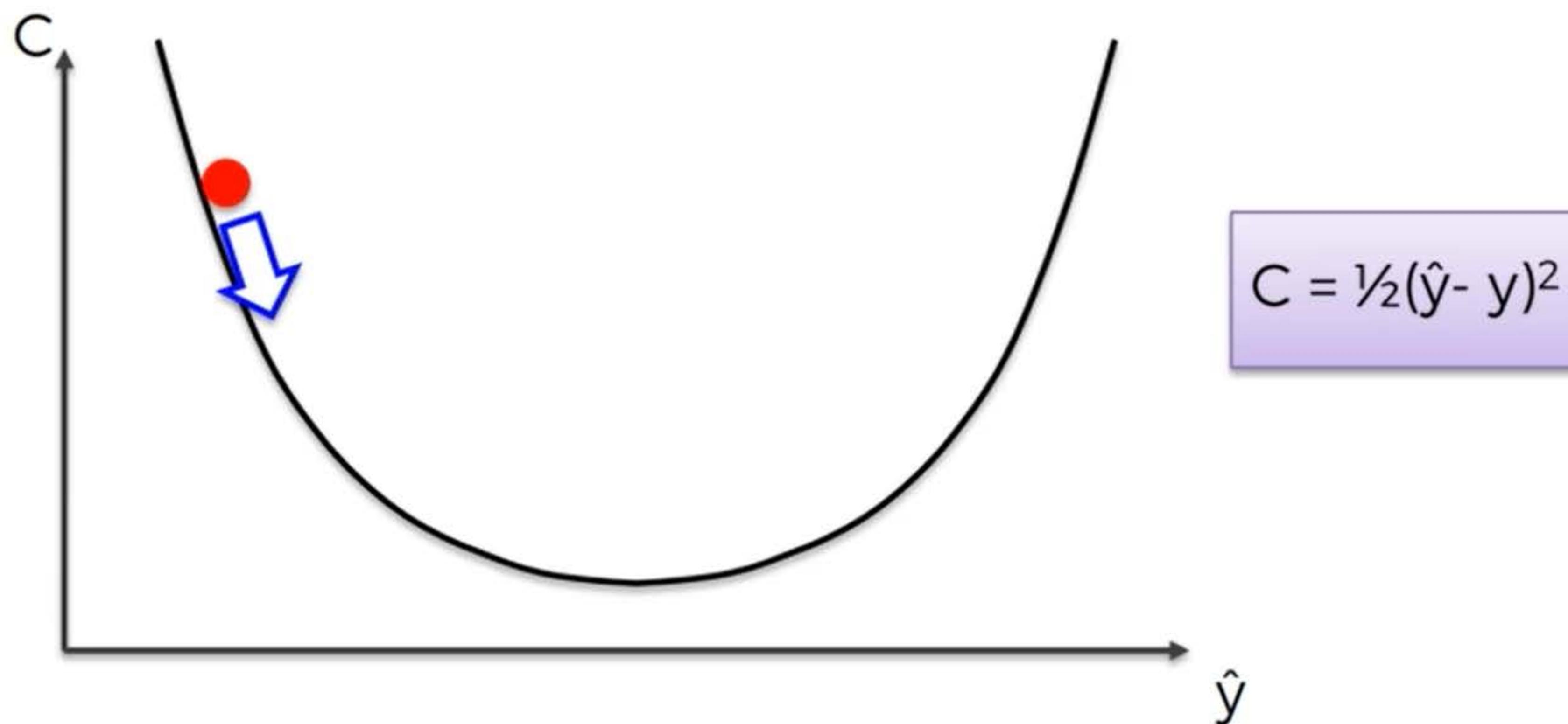
Gradient Descent

Gradient Descent

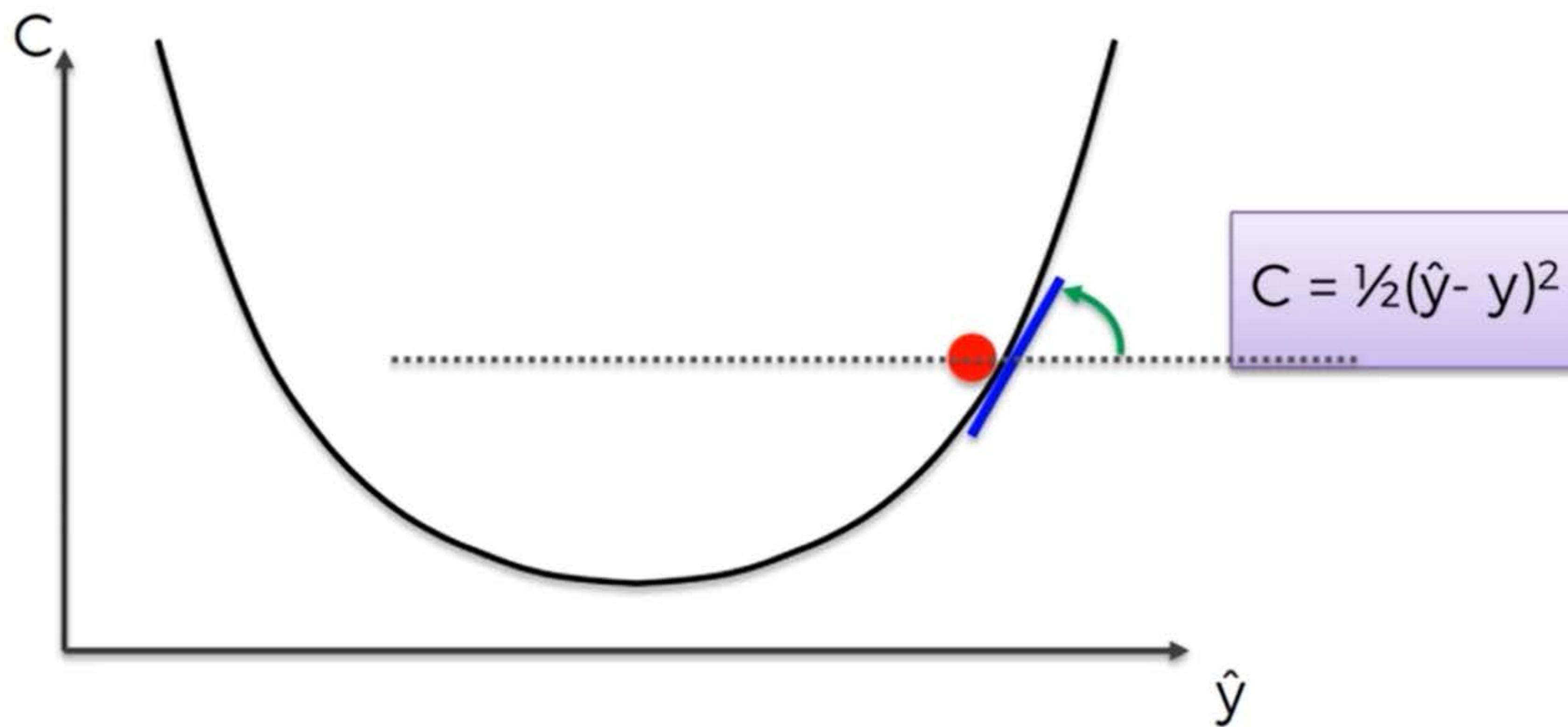
Gradient Descent



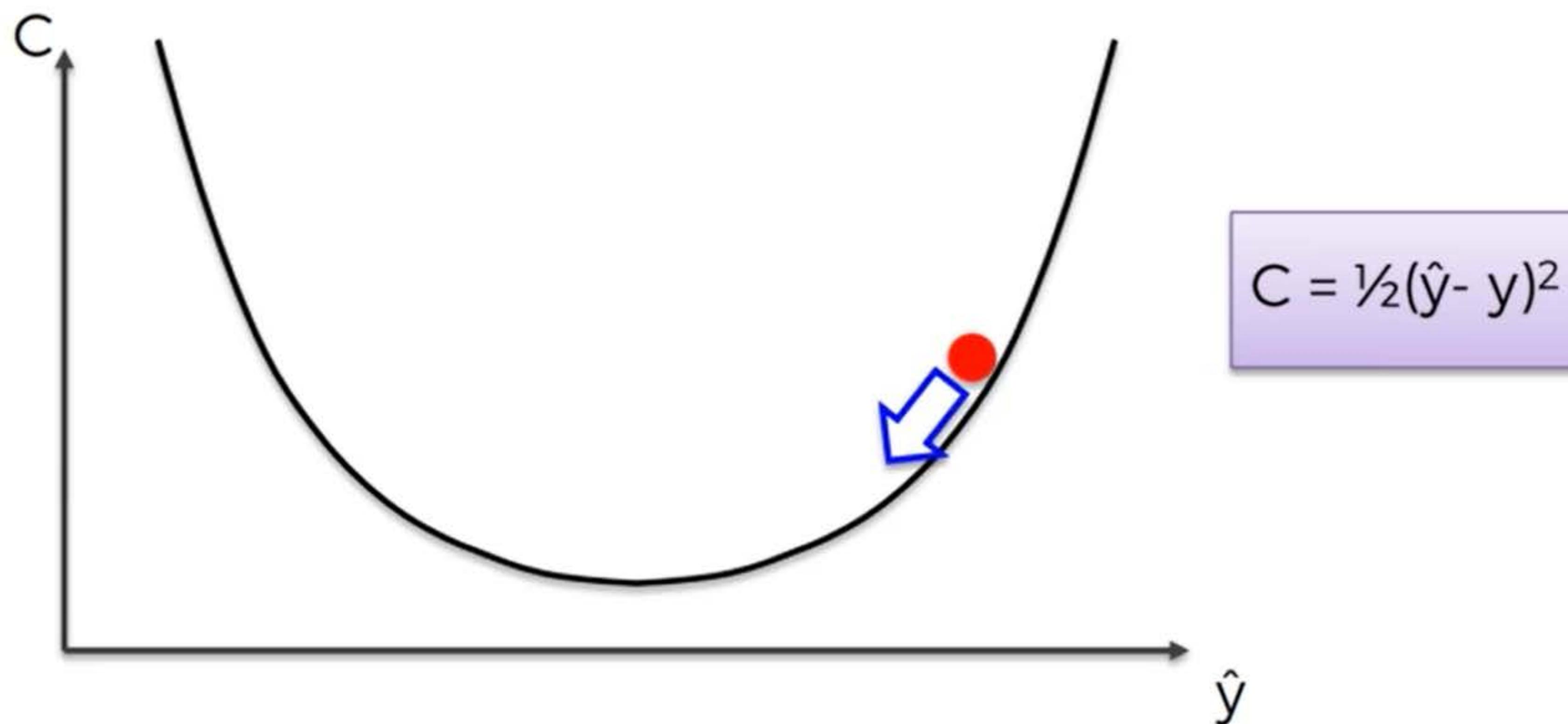
Gradient Descent



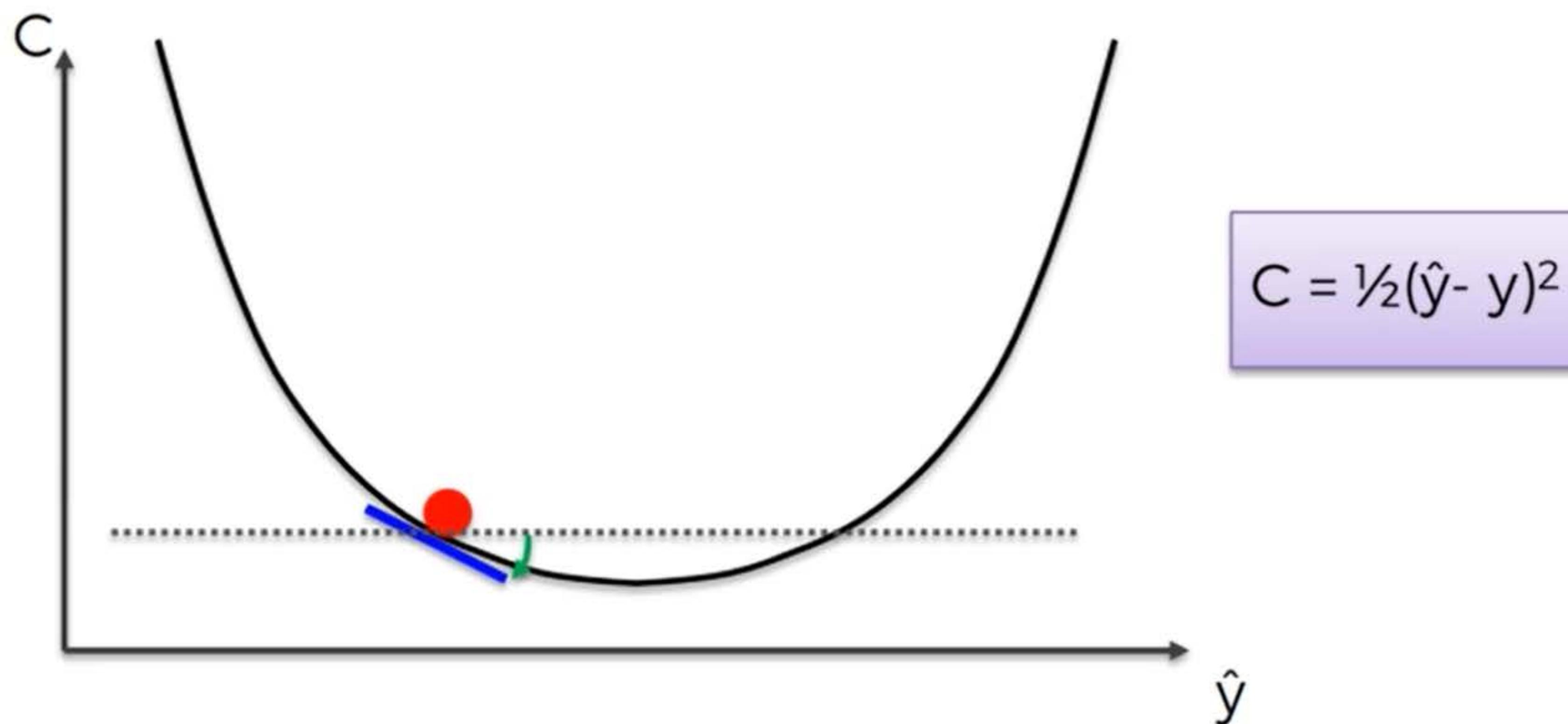
Gradient Descent



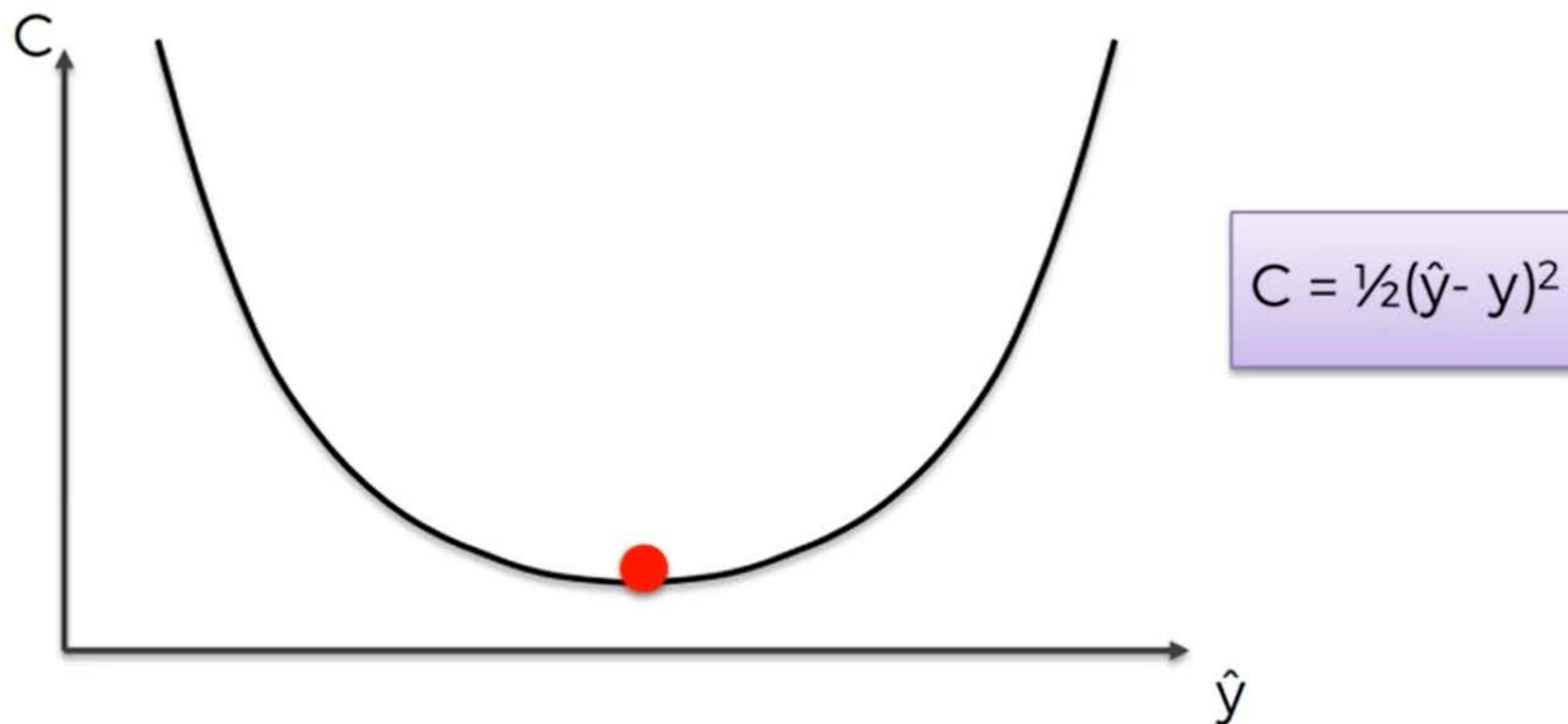
Gradient Descent



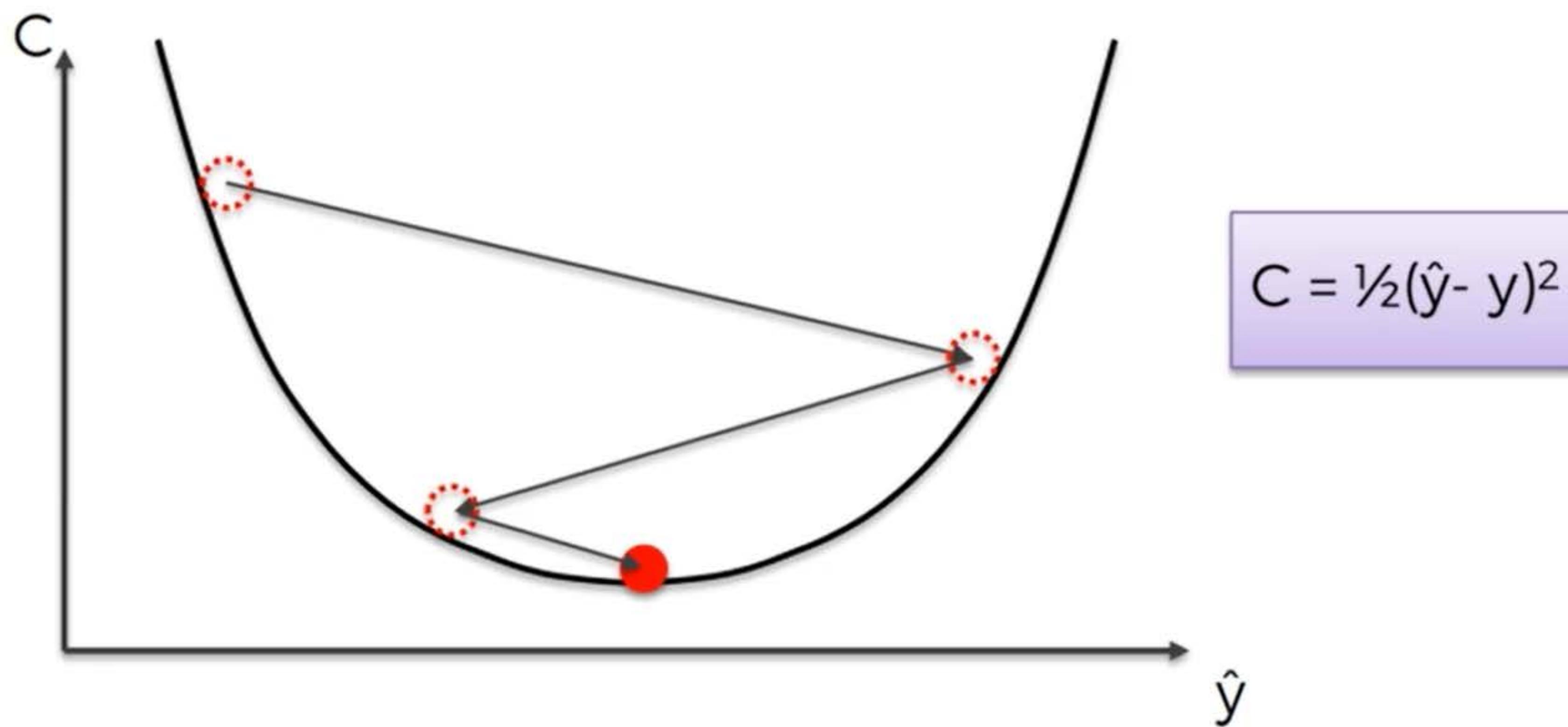
Gradient Descent



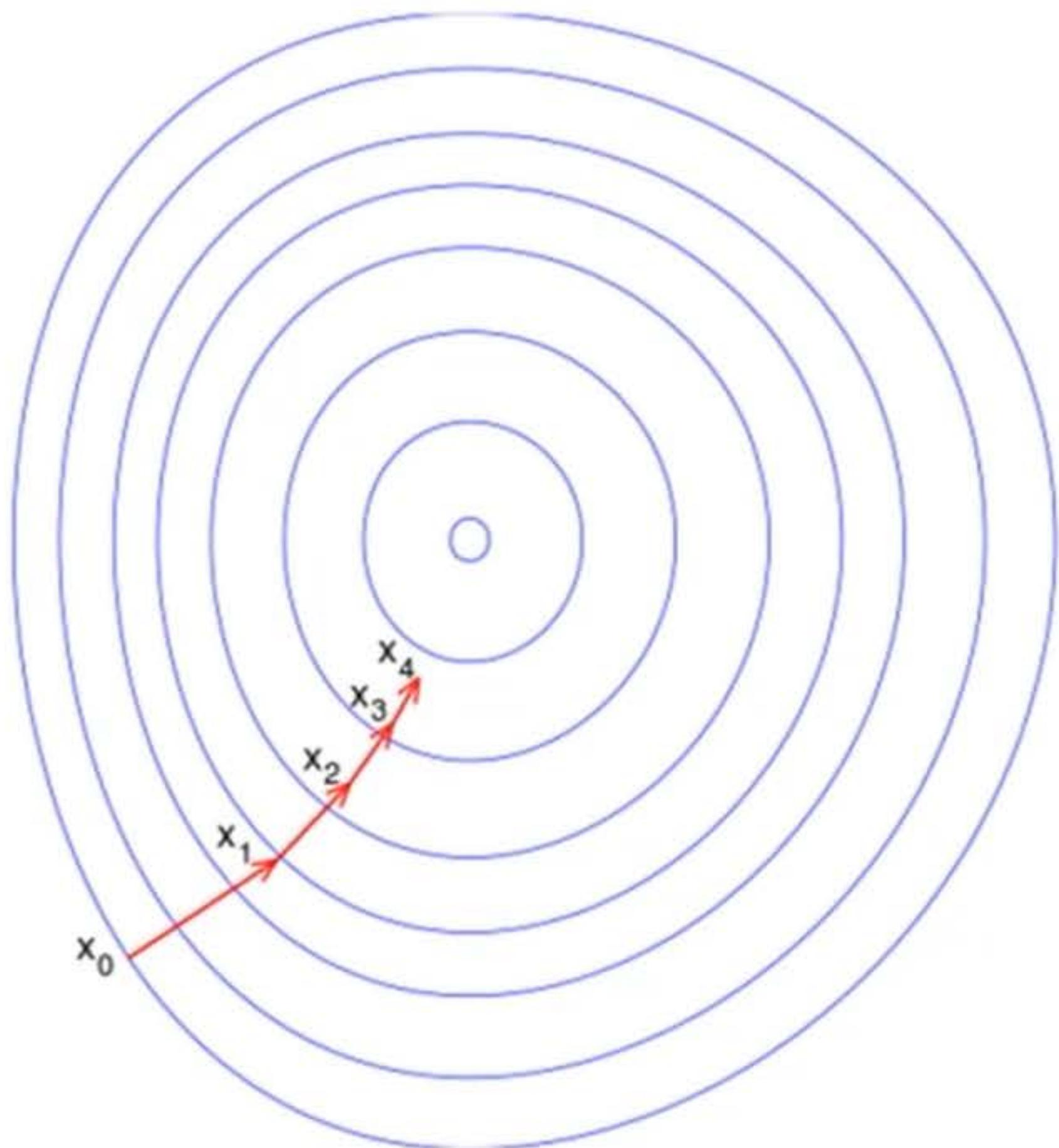
Gradient Descent



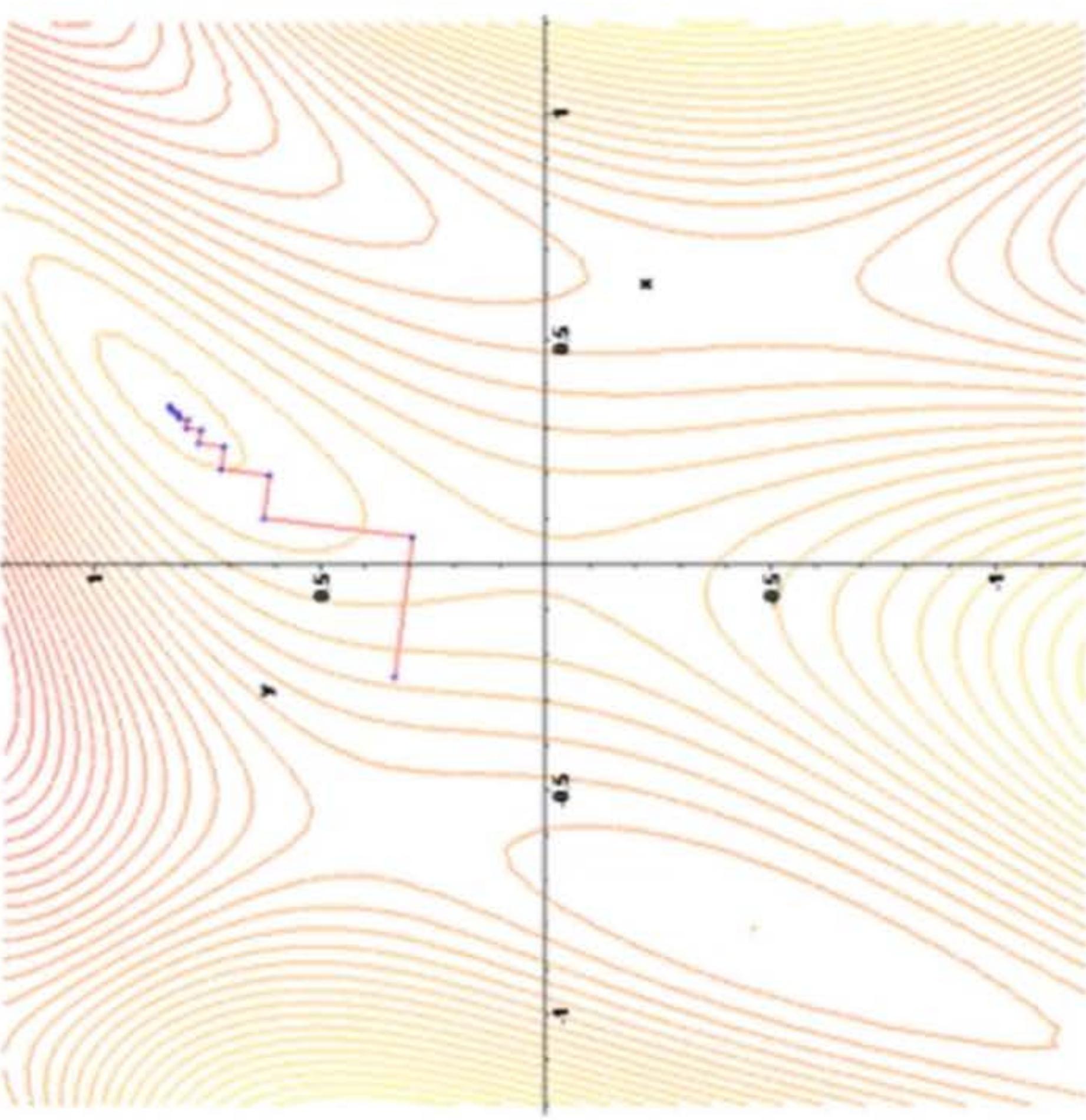
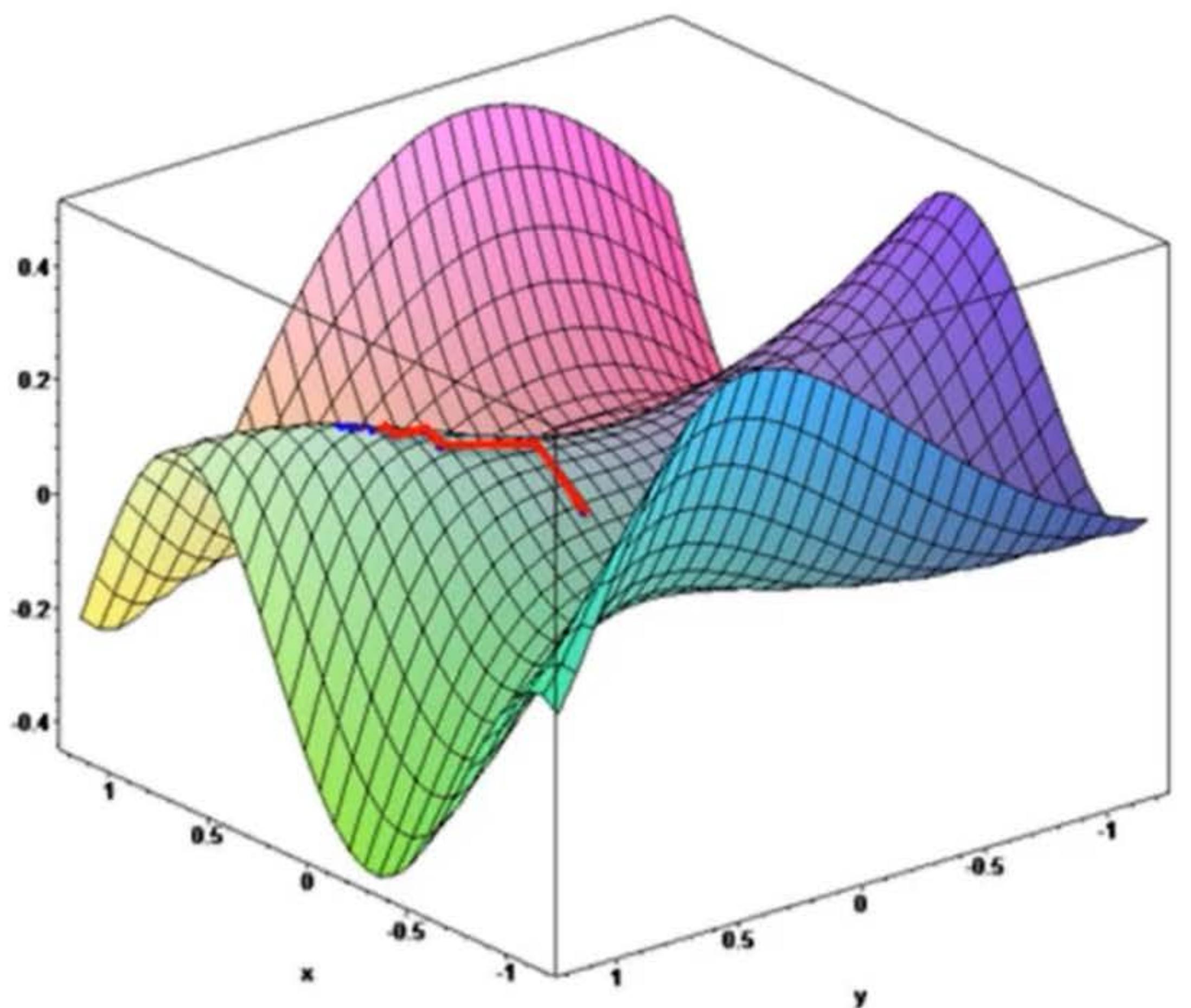
Gradient Descent



Gradient Descent

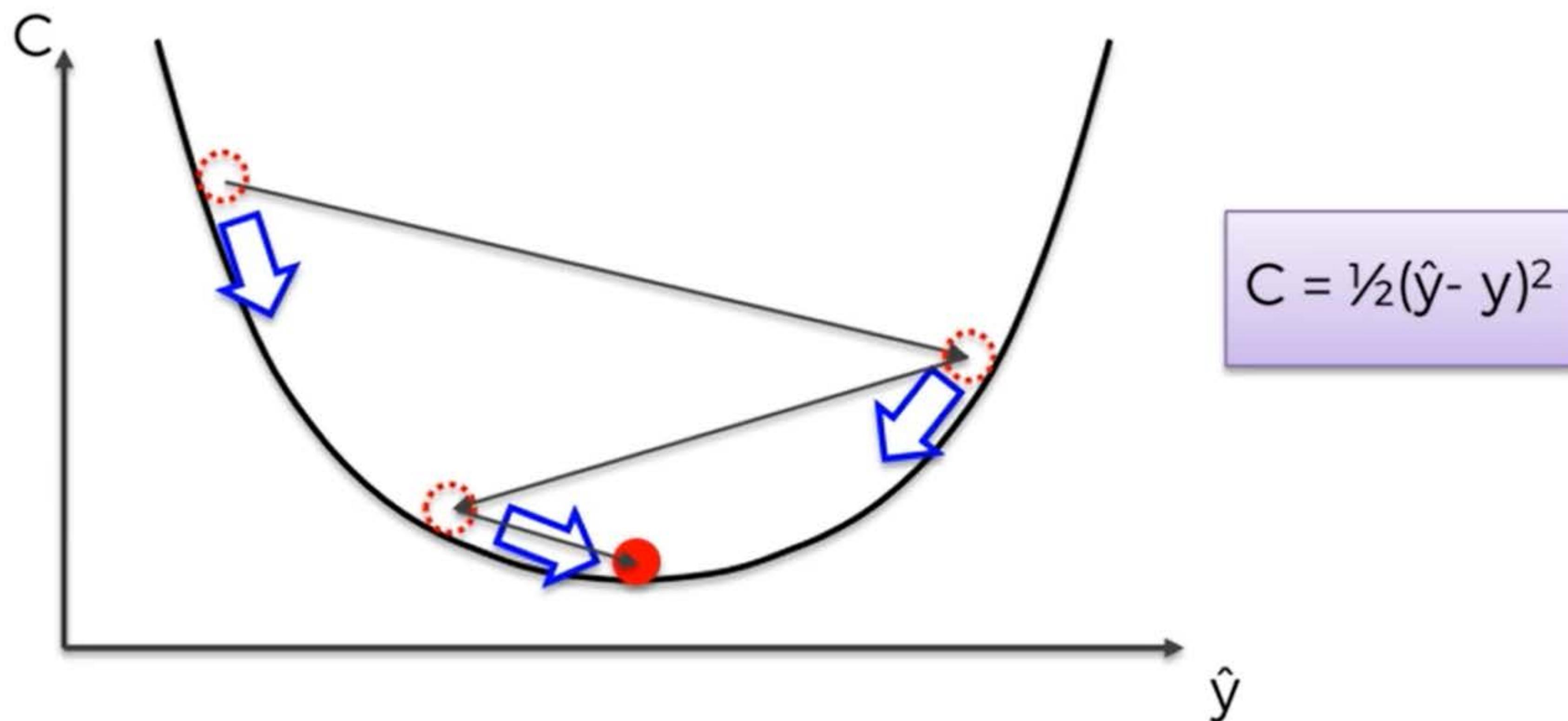


Gradient Descent

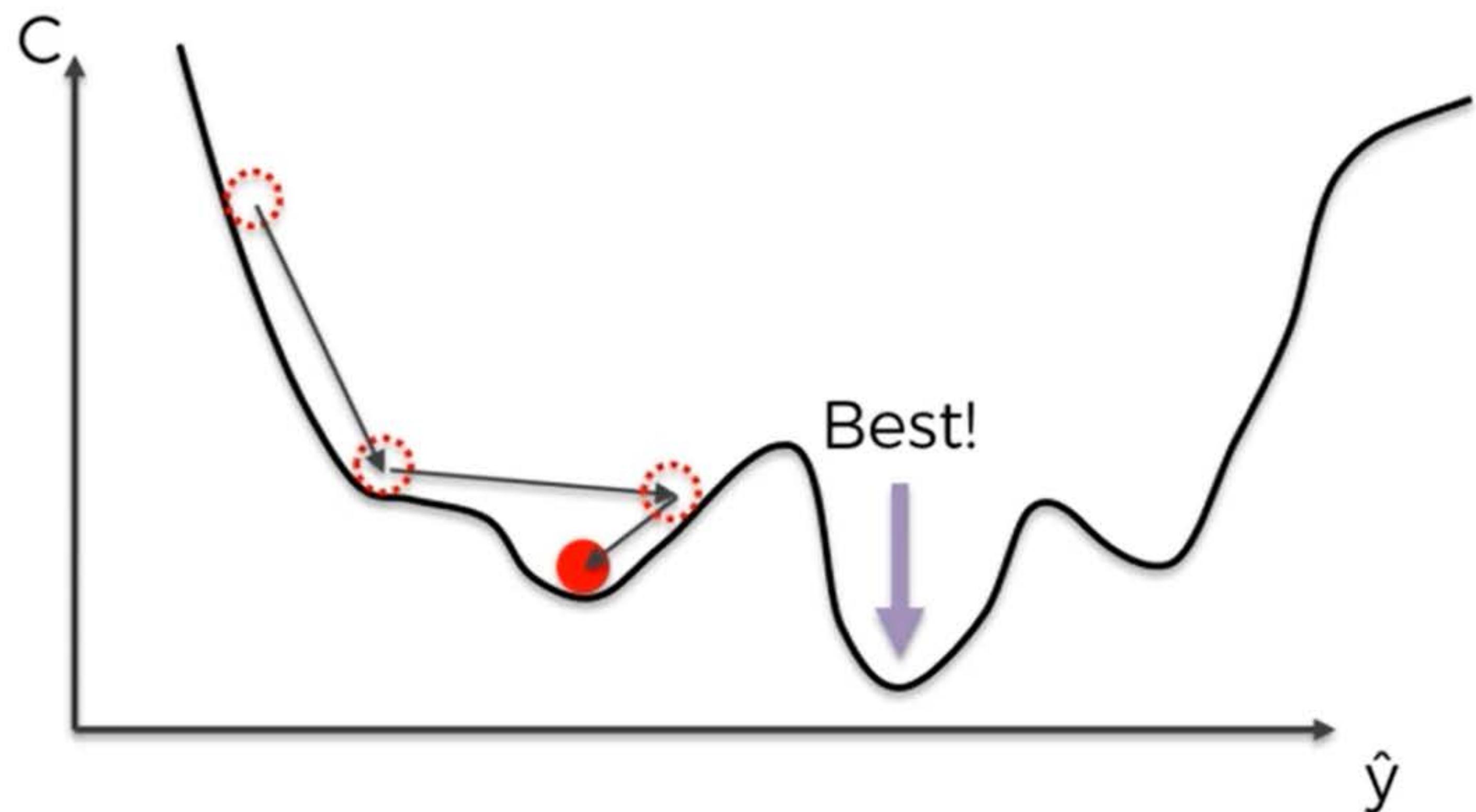


Stochastic Gradient Descent

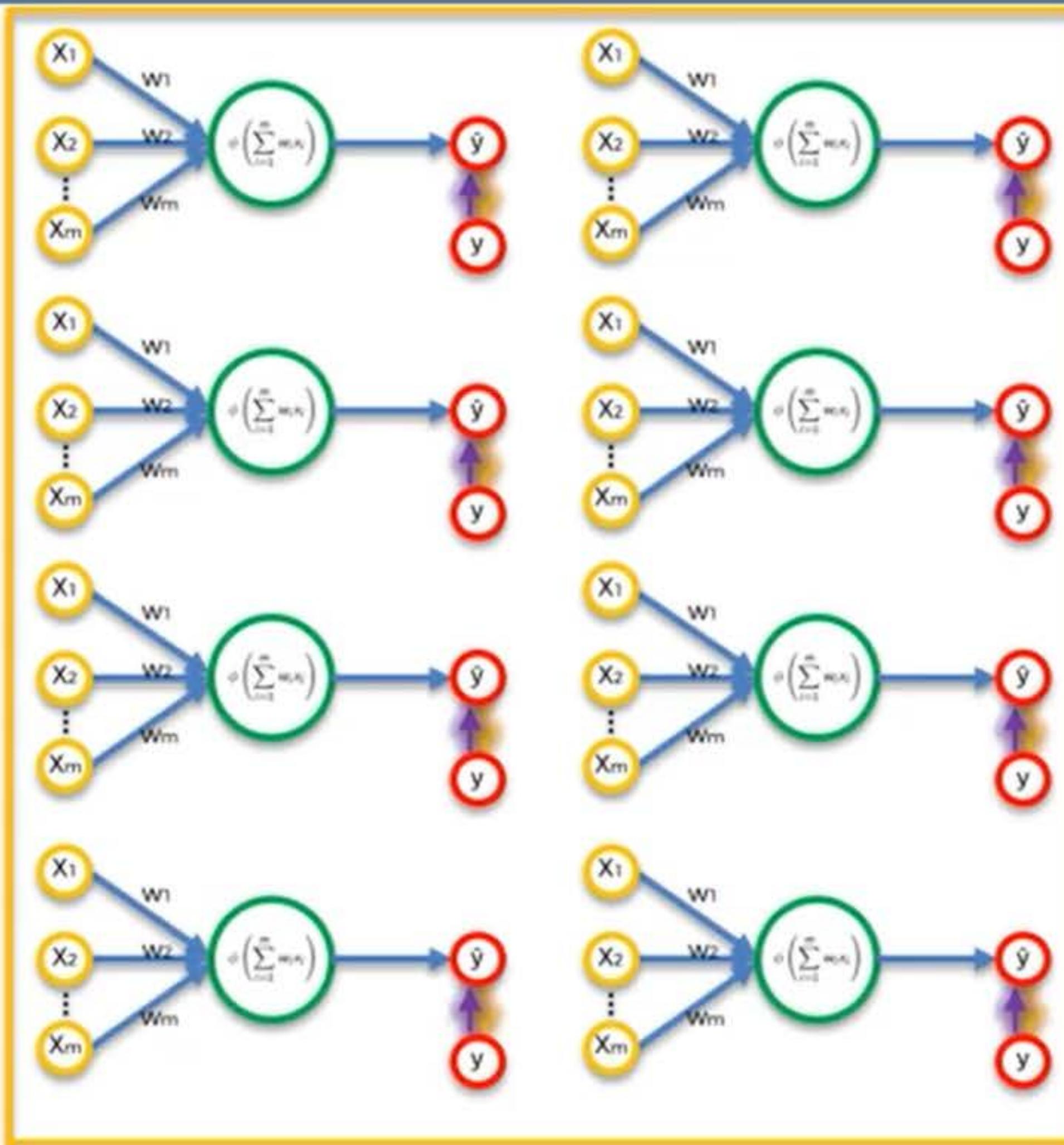
Stochastic Gradient Descent



Stochastic Gradient Descent



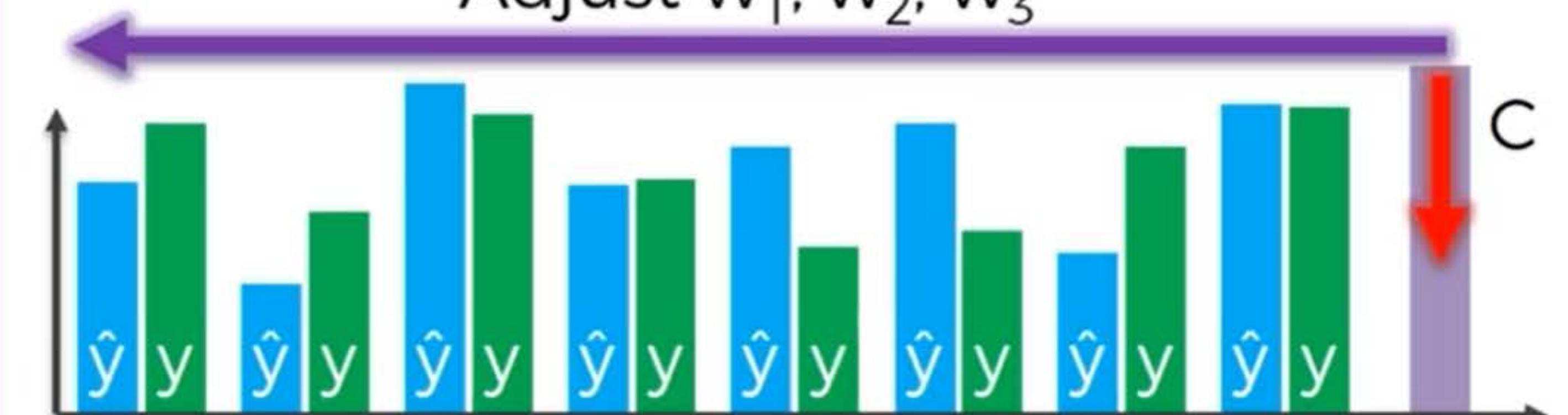
Stochastic Gradient Descent



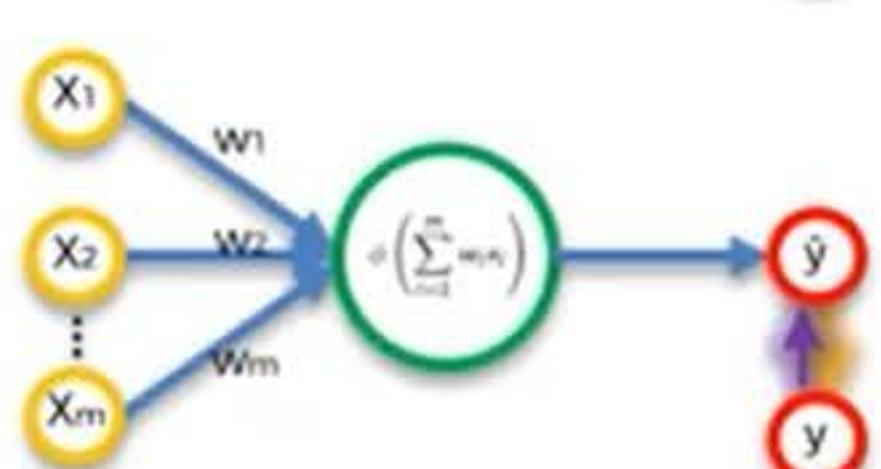
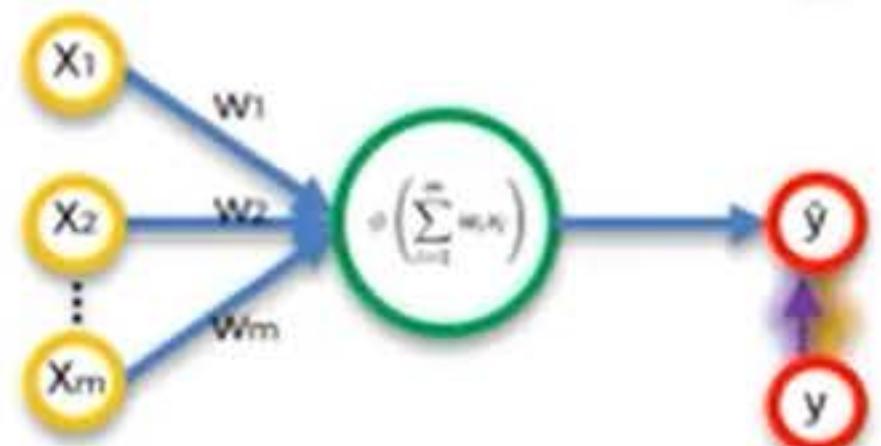
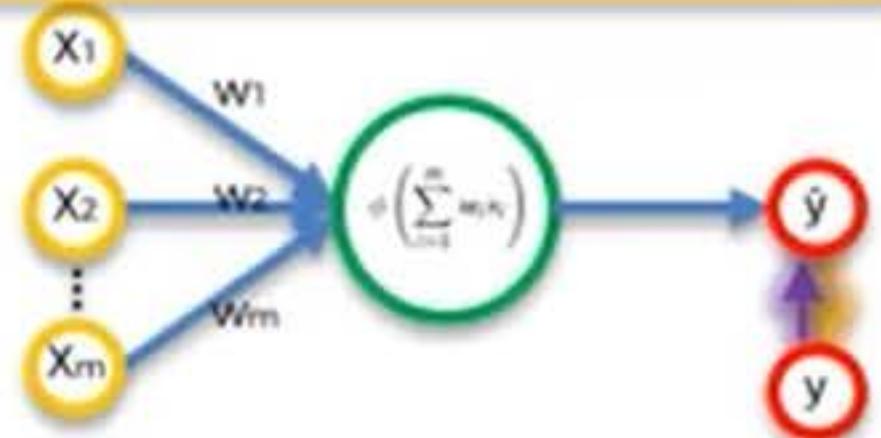
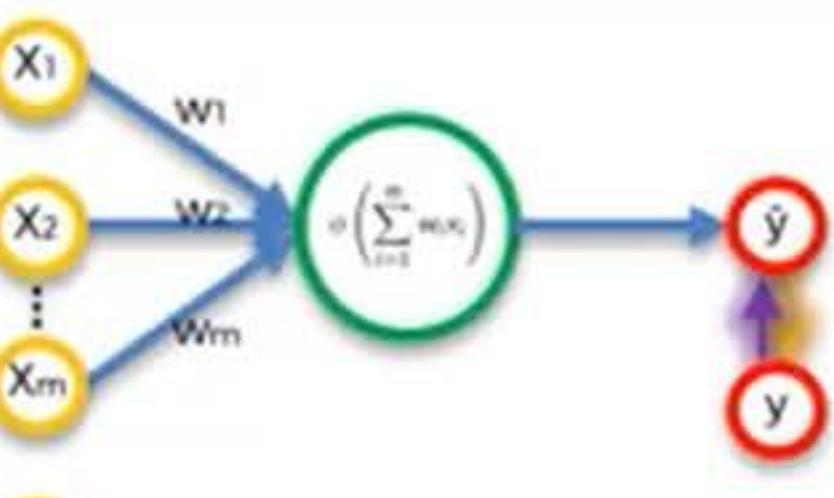
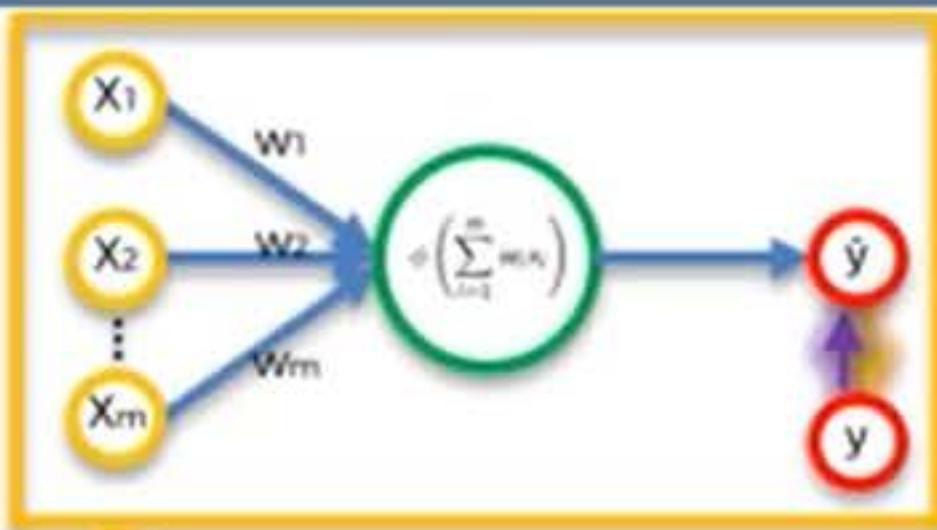
Row ID	Study Hrs	Sleep Hrs	Quiz	Exam
1	12	6	78%	93%
2	22	6.5	24%	68%
3	115	4	100%	95%
4	31	9	67%	75%
5	0	10	58%	51%
6	5	8	78%	60%
7	92	6	82%	89%
8	57	8	91%	97%

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

Adjust w_1, w_2, w_3



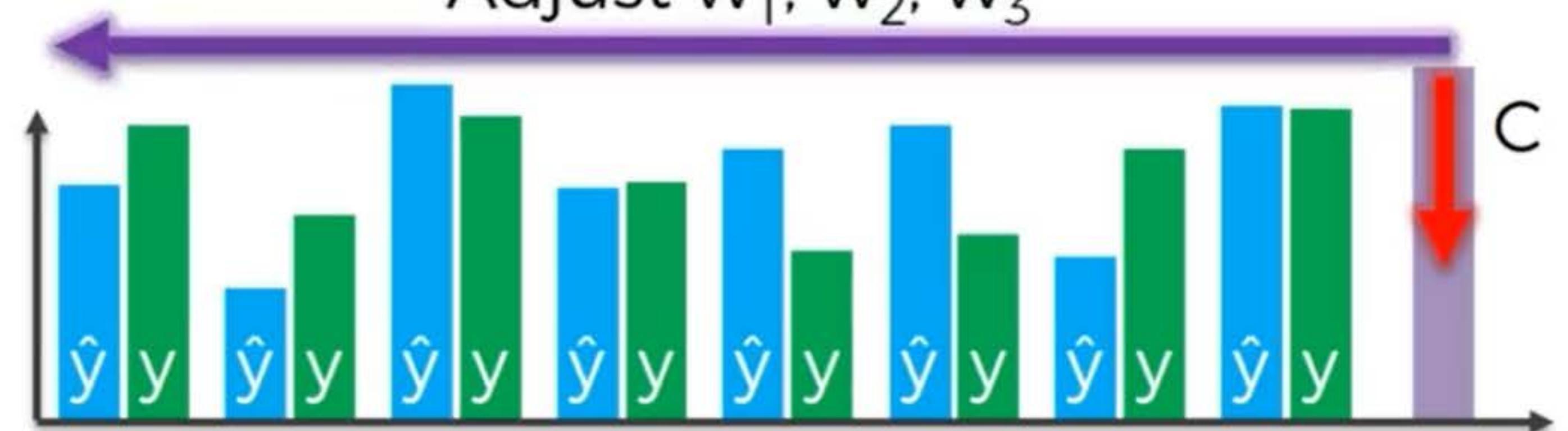
Stochastic Gradient Descent



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%

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

Adjust w_1, w_2, w_3



Stochastic Gradient Descent

Row ID	Study Hrs	Sleep Hrs	Ouiz	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%

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%

Upd w's ↑

Batch
Gradient
Descent

Stochastic
Gradient
Descent

Stochastic Gradient Descent

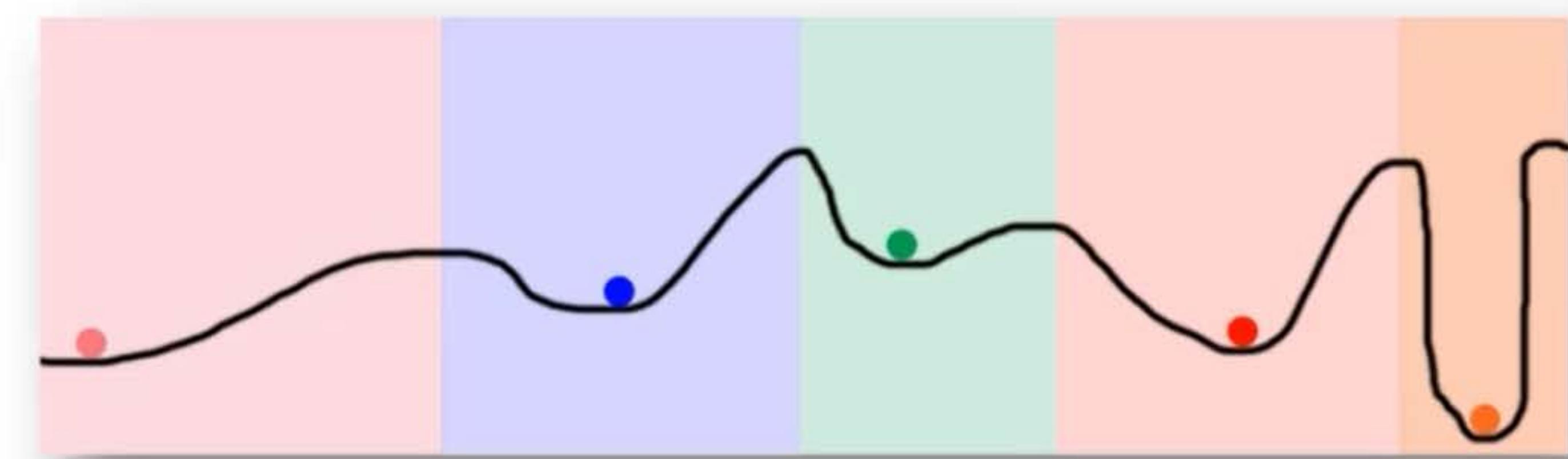
Additional Reading:

*A Neural Network in 13 lines
of Python (Part 2 - Gradient
Descent)*

Andrew Trask (2015)

Link:

<https://iamtrask.github.io/2015/07/27/python-network-part2/>



Stochastic Gradient Descent

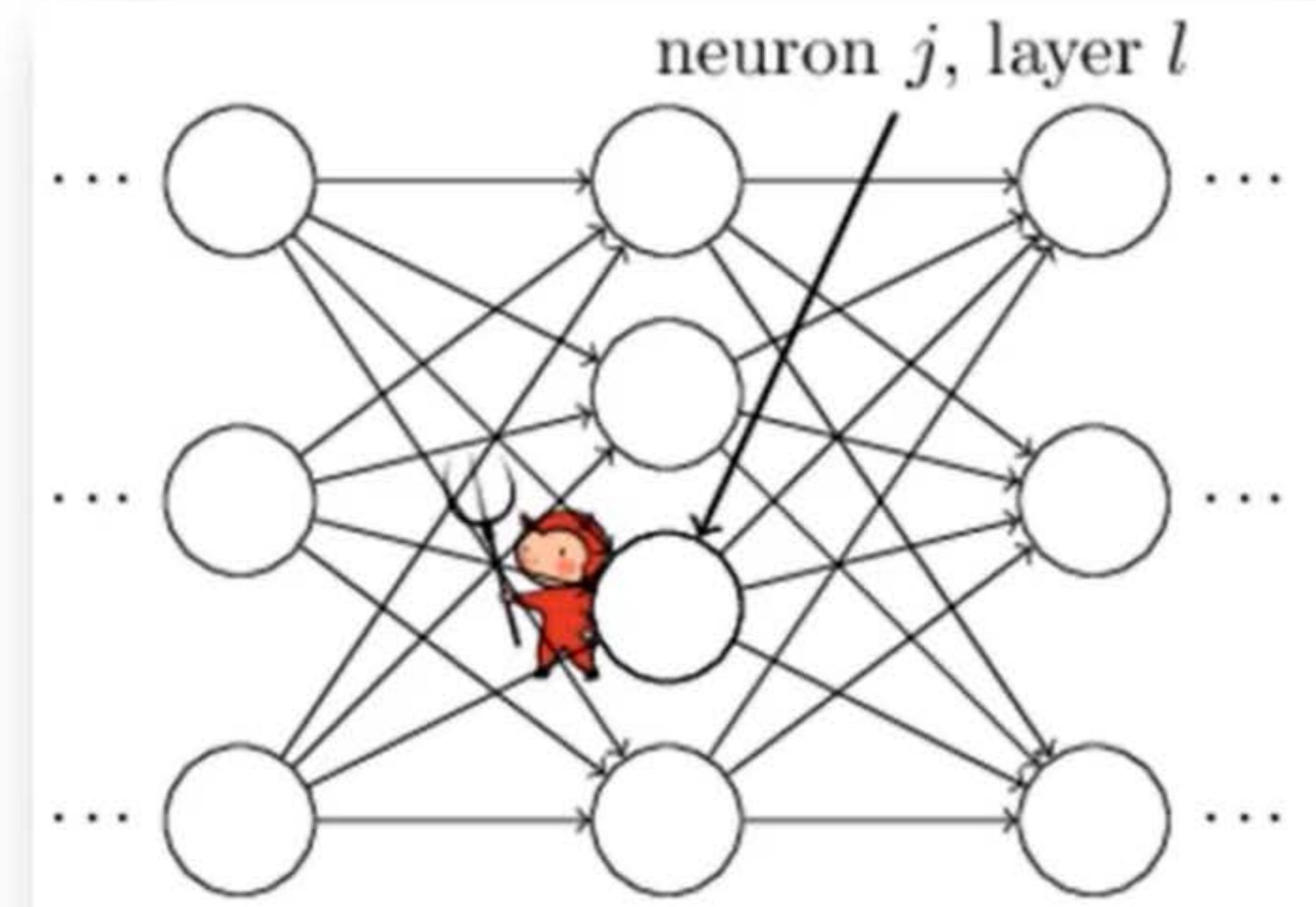
Additional Reading:

Neural Networks and Deep Learning

Michael Nielsen (2015)

Link:

<http://neuralnetworksanddeeplearning.com/chap2.html>



Backpropagation

Gradient Descent

Forward Propagation

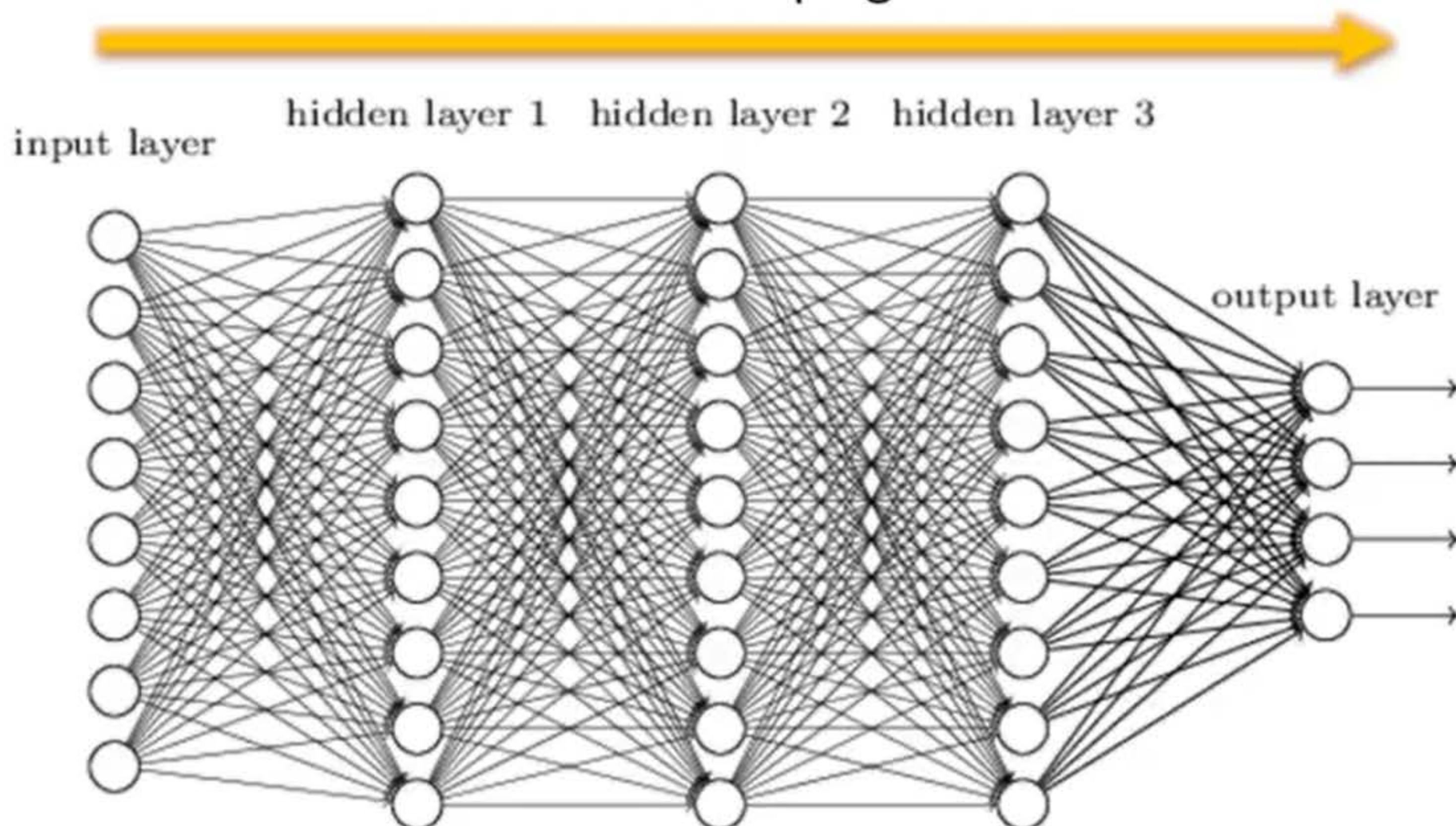


Image Source: neuralnetworksanddeeplearning.com

Gradient Descent

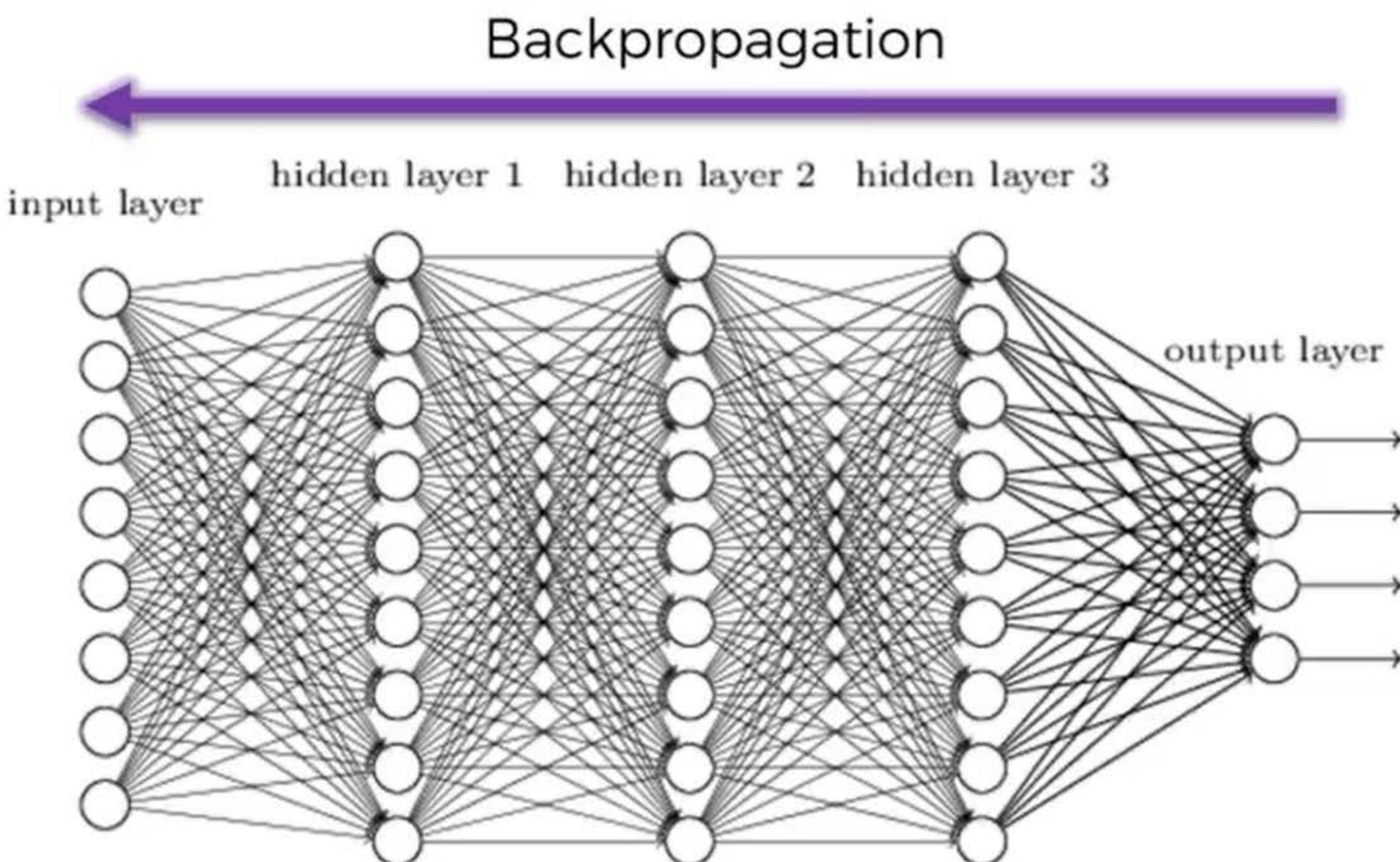


Image Source: neuralnetworksanddeeplearning.com

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.

Churn Modelling.csv - Microsoft Excel

Home Insert Page Layout Formulas Data Review View Developer Add-Ins Team

Cut Copy Format Painter Paste

Font Alignment Number Styles Cells Editing

A2 f1

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProduct	HasCrCard	IsActiveMember	EstimatedSalary	Exited	
2	1	15634602	Hargrave	619	France	Female	42	2	0	1	1	1	101348.88	1	
3	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0	
4	3	15619304	Onio	502	France	Female	42	8	159660.8	3	1	0	113931.57	1	
5	4	15701354	Boni	699	France	Female	39	1	0	2	0	0	93826.63	0	
6	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.1	0	
7	6	15574012	Chu	645	Spain	Male	44	8	113755.78	2	1	0	149756.71	1	
8	7	15592531	Bartlett	822	France	Male	50	7	0	2	1	1	10062.8	0	
9	8	15656148	Obinna	376	Germany	Female	29	4	115046.74	4	1	0	119346.88	1	
10	9	15792365	He	501	France	Male	44	4	142051.07	2	0	1	74940.5	0	
11	10	15592389	H?	684	France	Male	27	2	134603.88	1	1	1	71725.73	0	
12	11	15767821	Bearce	528	France	Male	31	6	102016.72	2	0	0	80181.12	0	
13	12	15737173	Andrews	497	Spain	Male	24	3	0	2	1	0	76390.01	0	
14	13	15632264	Kay	476	France	Female	34	10	0	2	1	0	26260.98	0	
15	14	15691483	Chin	549	France	Female	25	5	0	2	0	0	190857.79	0	
16	15	15600882	Scott	635	Spain	Female	35	7	0	2	1	1	65951.65	0	
17	16	15643966	Goforth	616	Germany	Male	45	3	143129.41	2	0	1	64327.26	0	
18	17	15737452	Romeo	653	Germany	Male	58	1	132602.88	1	1	0	5097.67	1	
19	18	15788218	Henderson	549	Spain	Female	24	9	0	2	1	1	14406.41	0	
20	19	15661507	Muldrow	587	Spain	Male	45	6	0	1	0	0	158684.81	0	
21	20	15568982	Hao	726	France	Female	24	6	0	2	1	1	54724.03	0	
22	21	15577657	McDonald	732	France	Male	41	8	0	2	1	1	170886.17	0	
23	22	15597945	Dellucci	636	Spain	Female	32	8	0	2	1	0	138555.46	0	
24	23	15699309	Gerasimov	510	Spain	Female	38	4	0	1	1	0	118913.53	1	
25	24	15725737	Mosman	669	France	Male	46	3	0	2	0	1	8487.75	0	
26	25	15625047	Yen	846	France	Female	38	5	0	1	1	1	187616.16	0	

File Edit View Insert Cell Kernel Help

Not Trusted

Python 3



Part 2 - Building the ANN

Initializing the ANN

```
In [14]: ann = tf.keras.models.Sequential()
```

Adding the input layer and the first hidden layer

```
In [16]: #relu : rectified activation function  
#units=6 is just random  
#Dense class is to connect densely with all the neurons  
ann.add(tf.keras.layers.Dense(units=6, activation='relu'))
```

Adding the second hidden layer

```
In [17]: ann.add(tf.keras.layers.Dense(units=6, activation='relu'))
```

Adding the output layer

```
In [18]: #for output we use sigmoid(for binary output)  
# use softmax activation function(for categorical output)  
ann.add(tf.keras.layers.Dense(units=1, activation='sigmoid'))
```

Part 3 - Training the ANN

Compiling the ANN

```
In [19]: #adam is for stochastic gradient decent  
#loss : binary_crossentropy (for binary output) calculate cost  
#loss: category_crossentropy(for categorical output)  
ann.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])
```

Training the ANN on the Training set

```
In [17]: ann.fit(X_train, y_train, batch_size = 32, epochs = 100)
```