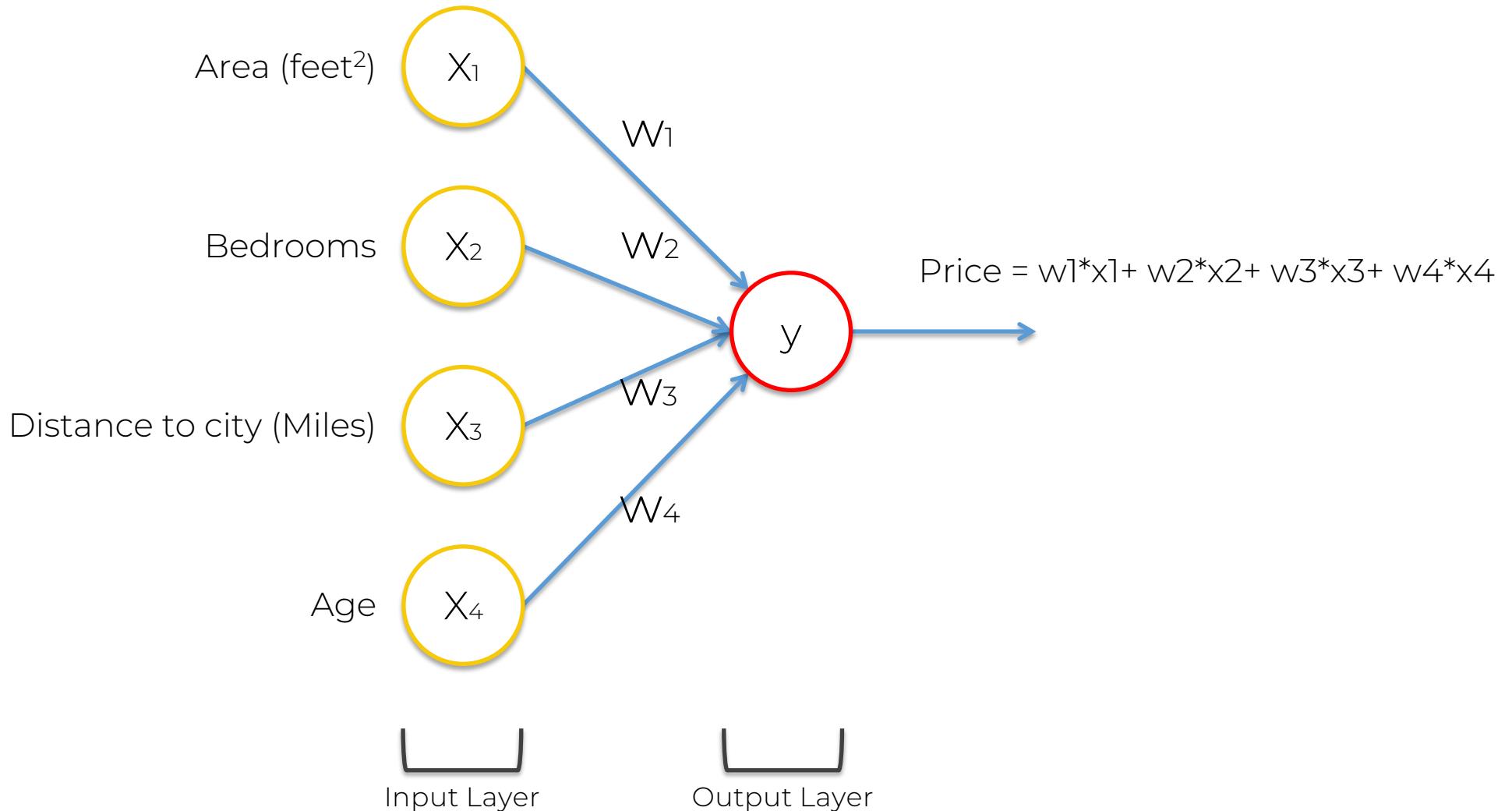


How Do Neural Networks Work?



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?



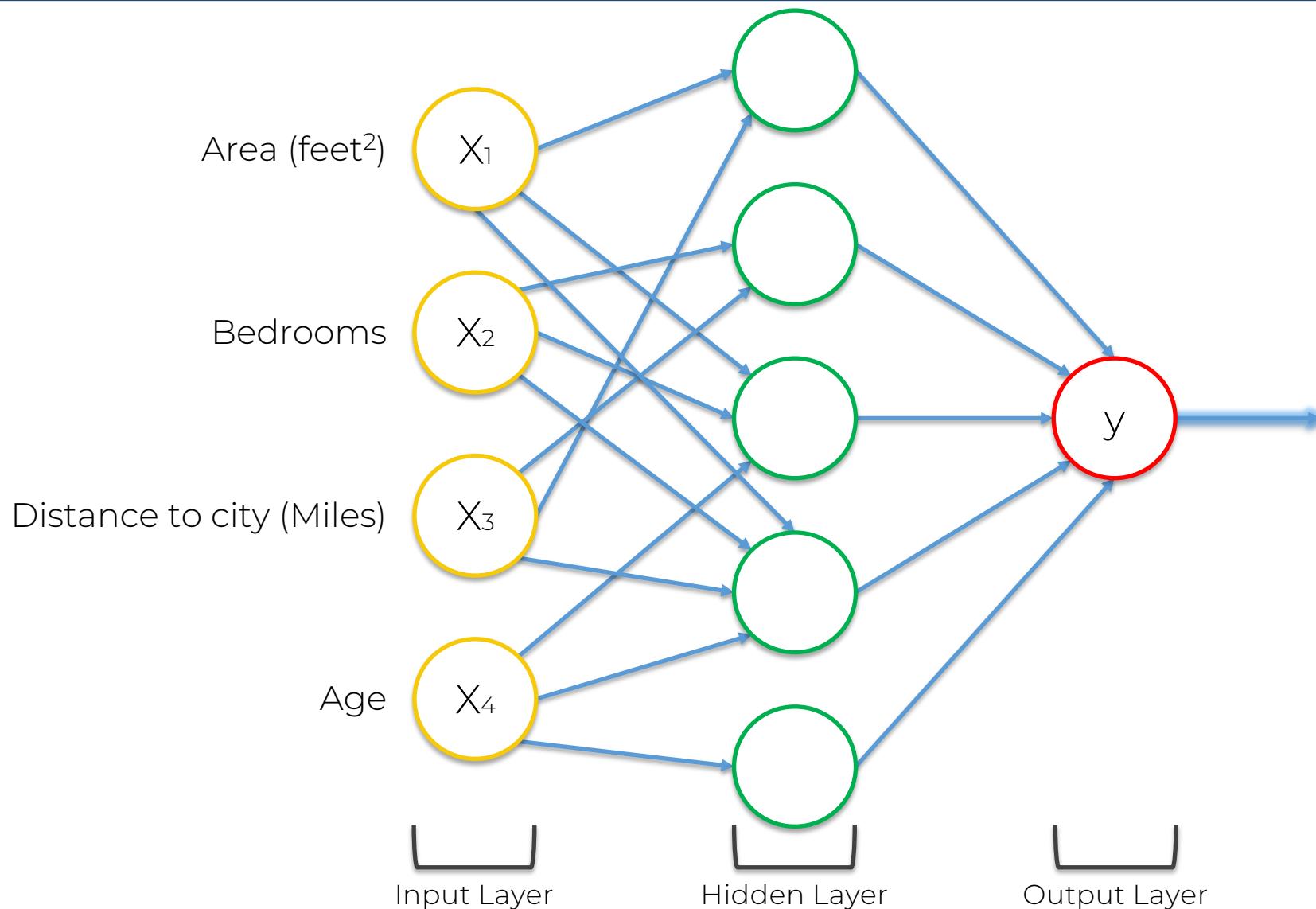
How Do Neural Networks Work?



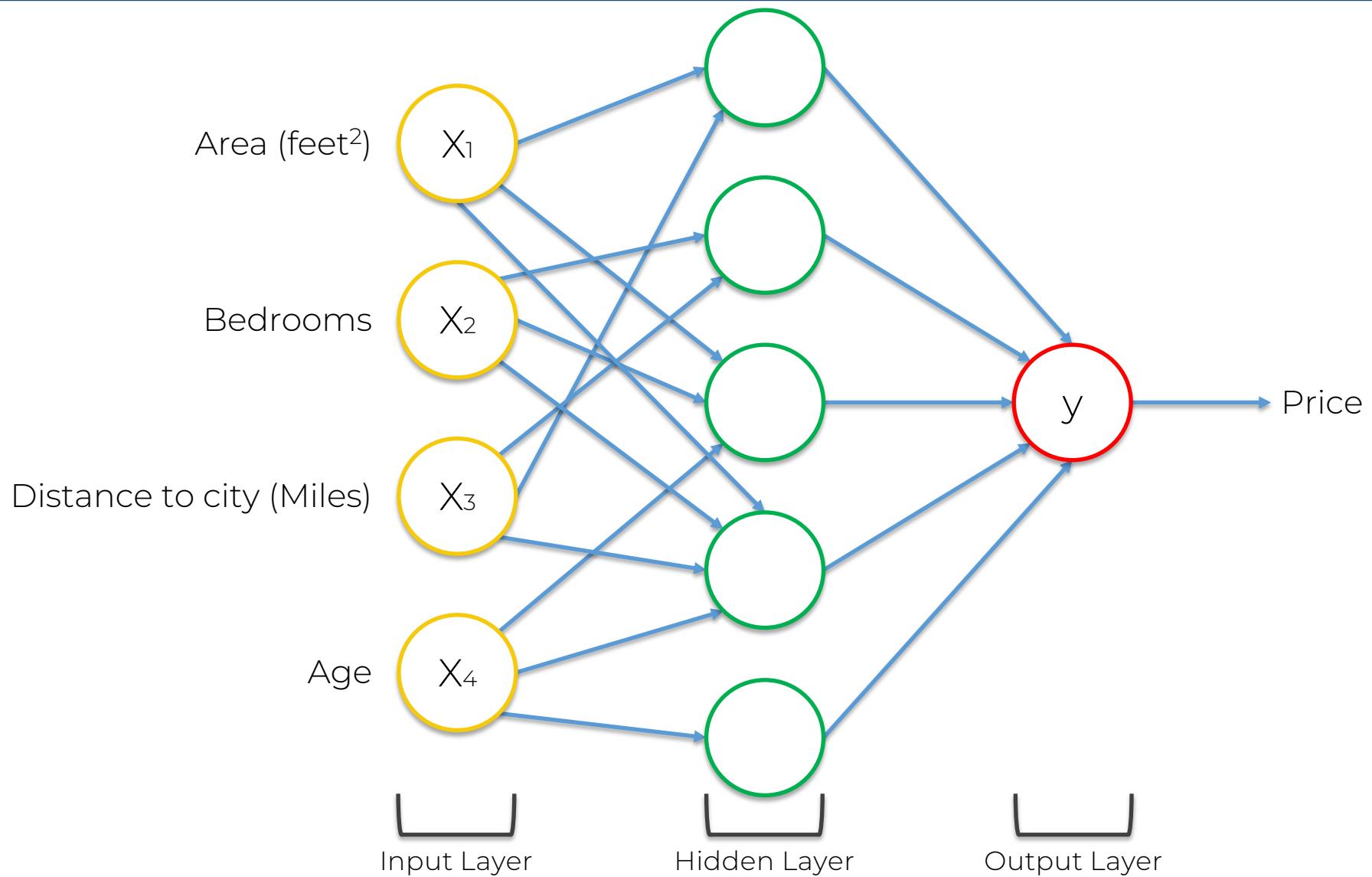
How Do Neural Networks Work?



How Do Neural Networks Work?



How Do Neural Networks Work?

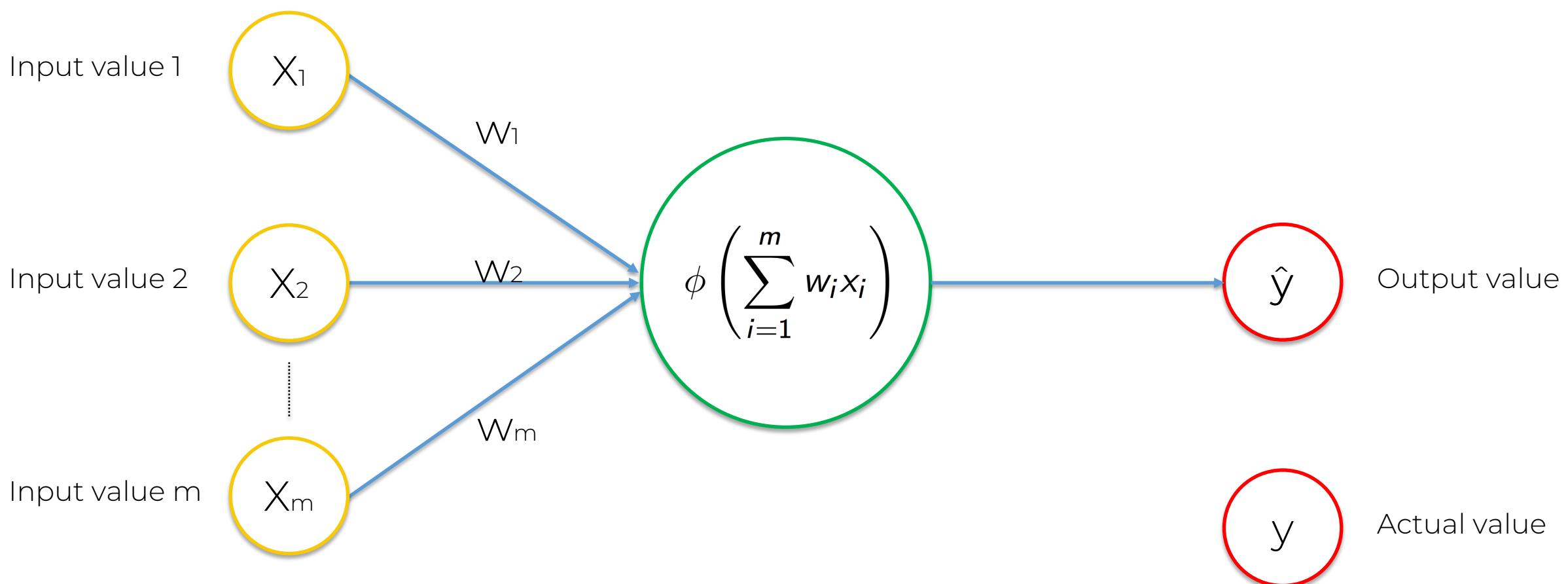


How do NNs Learn?

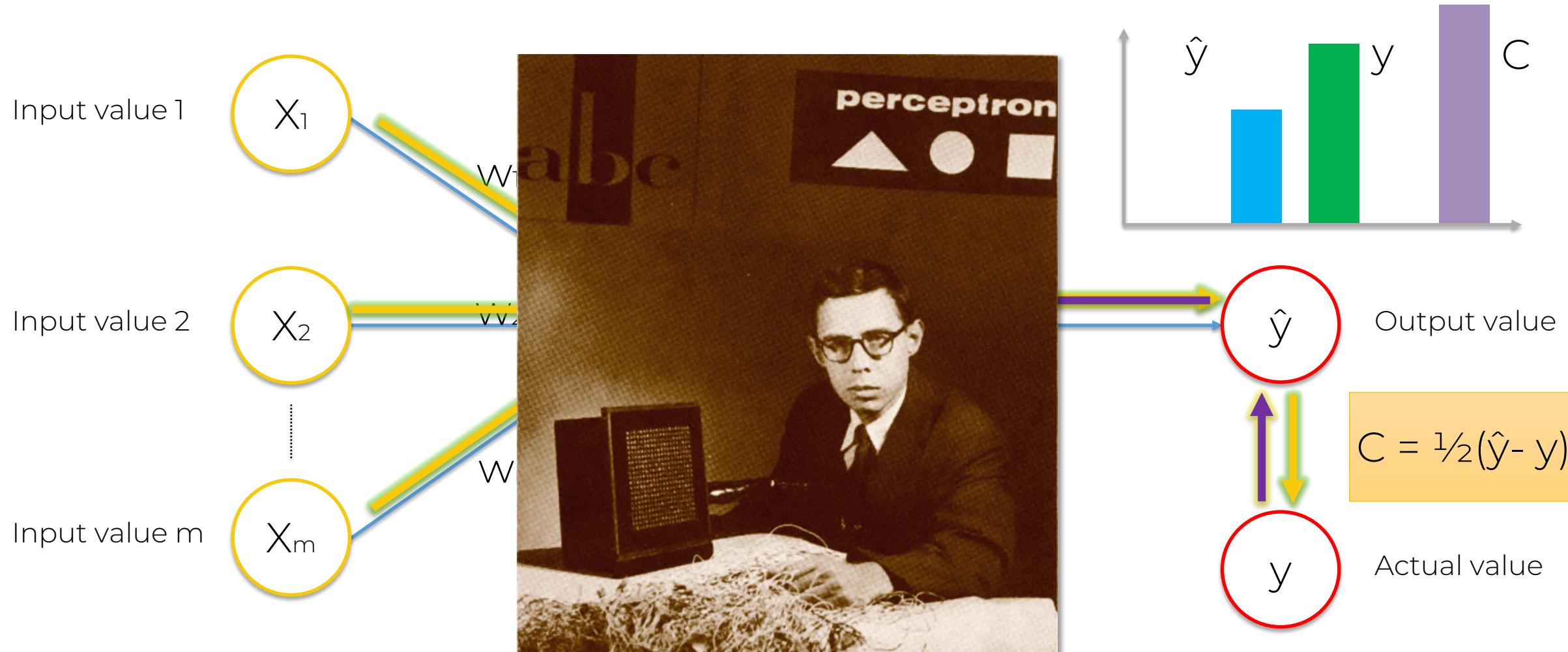
How do Neural Networks learn?



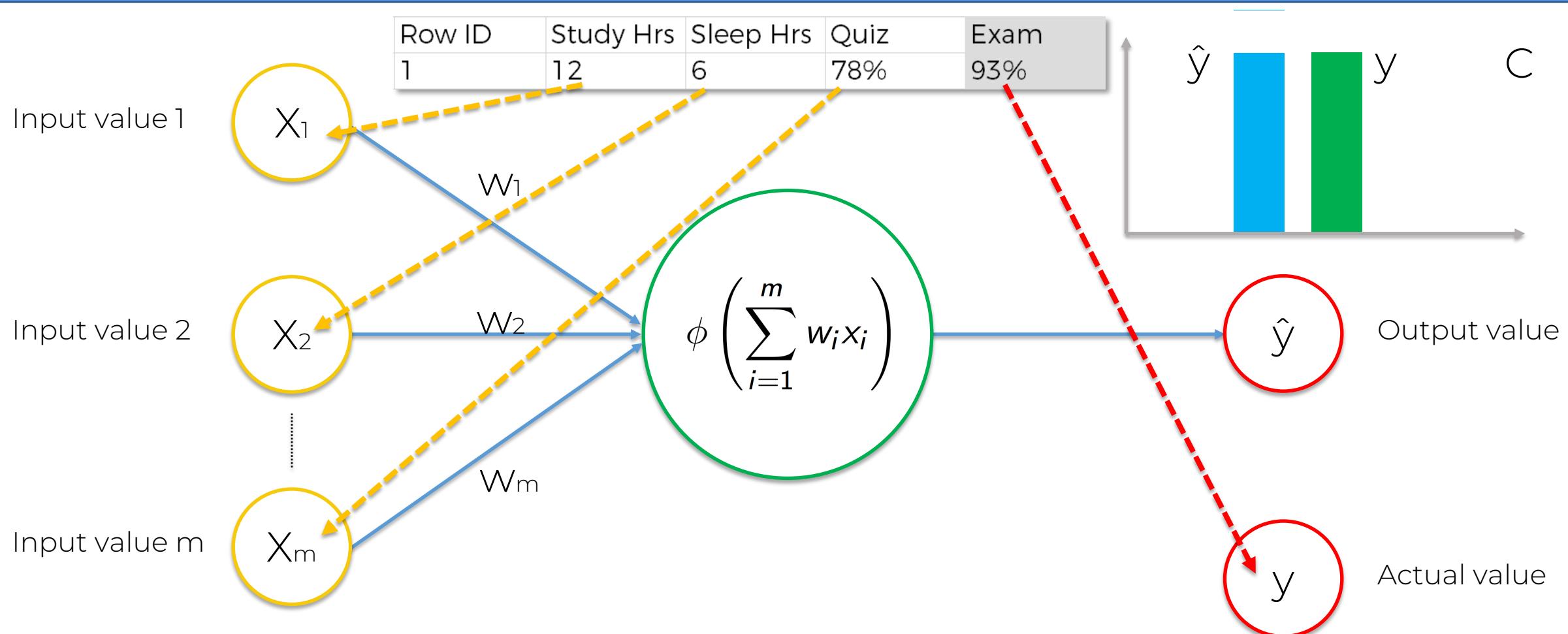
How do Neural Networks learn?



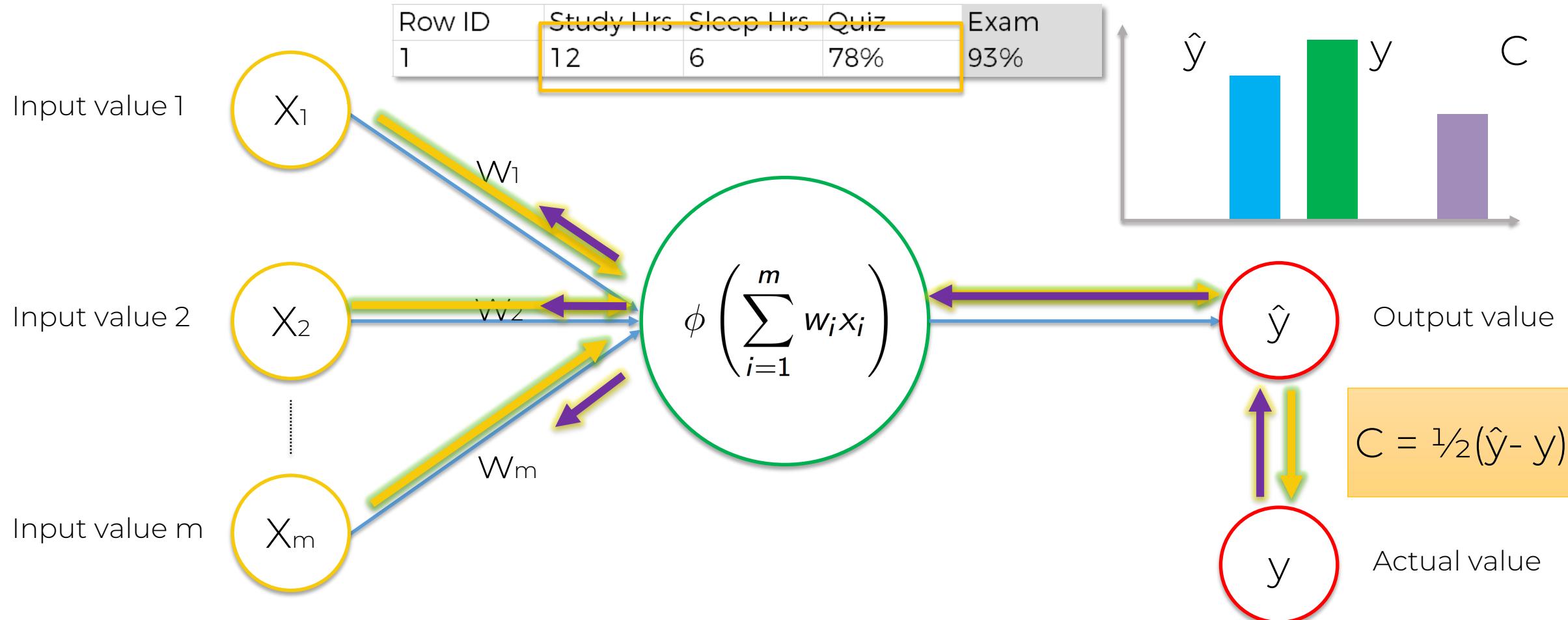
How do Neural Networks learn?



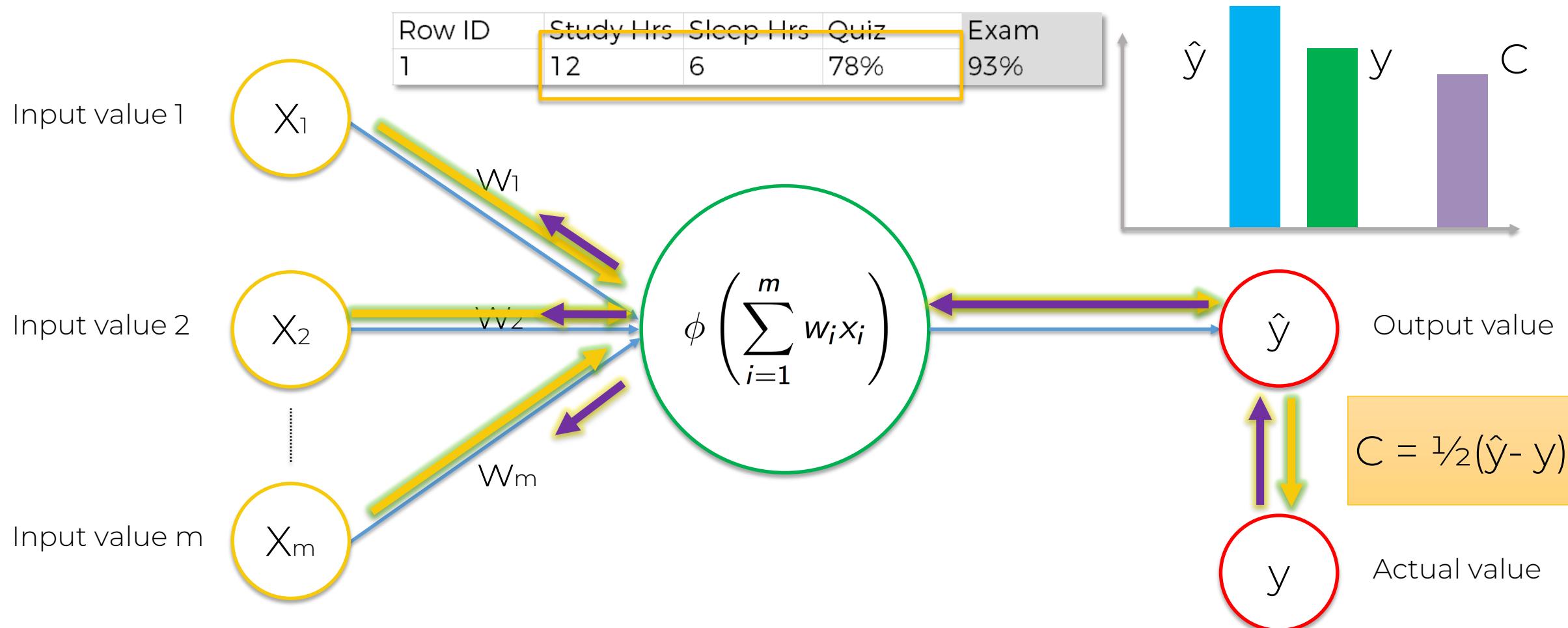
How do Neural Networks learn?



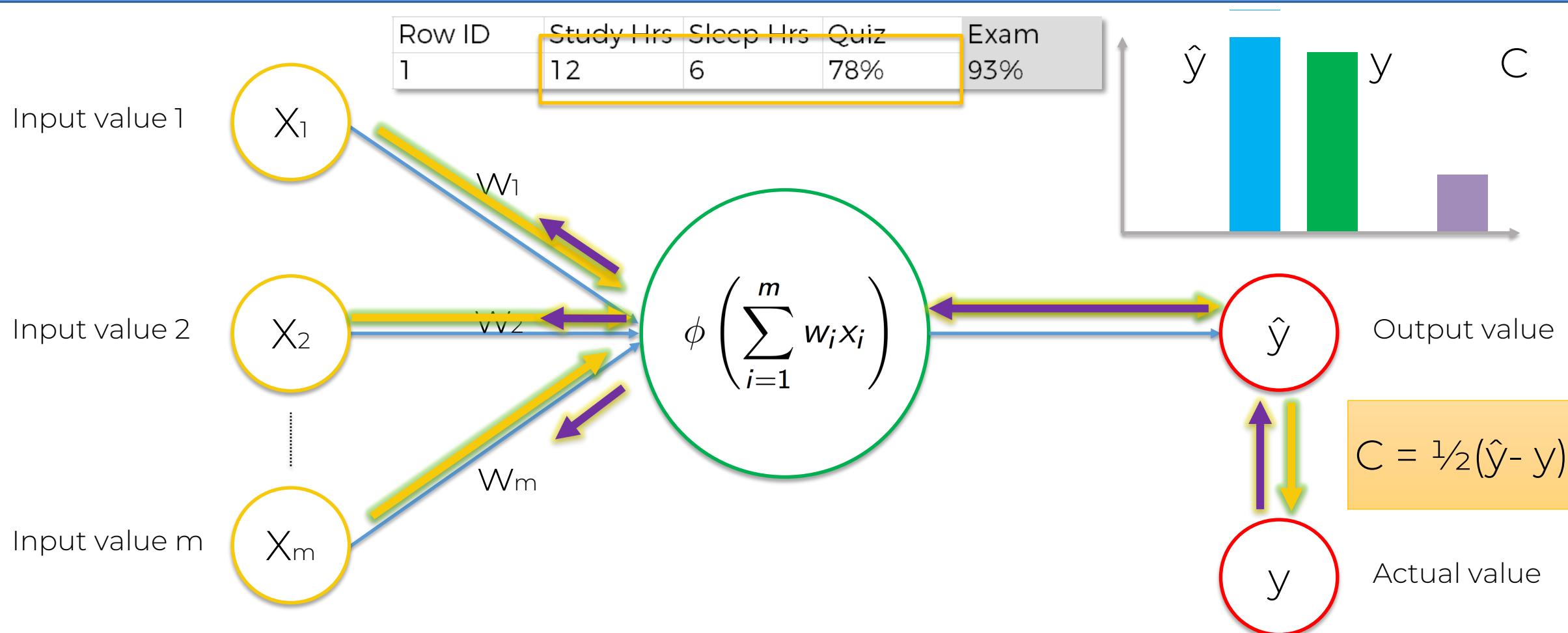
How do Neural Networks learn?



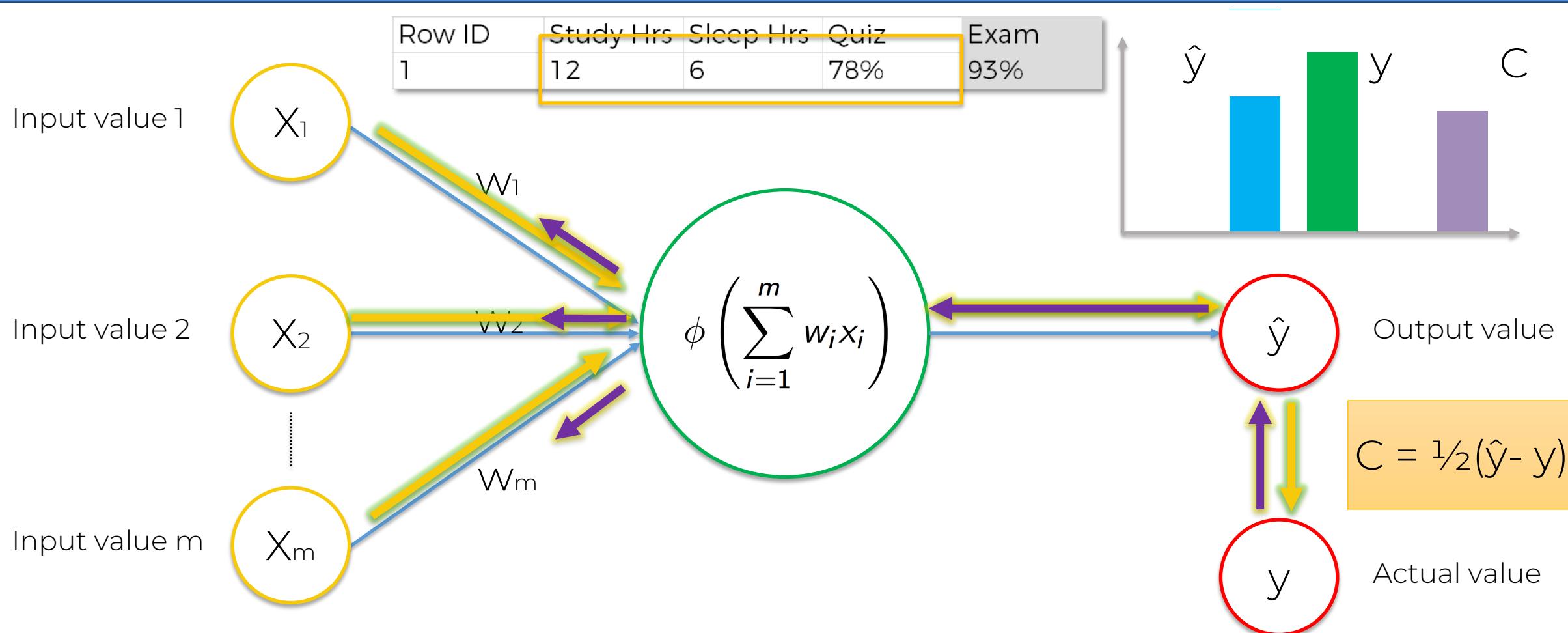
How do Neural Networks learn?



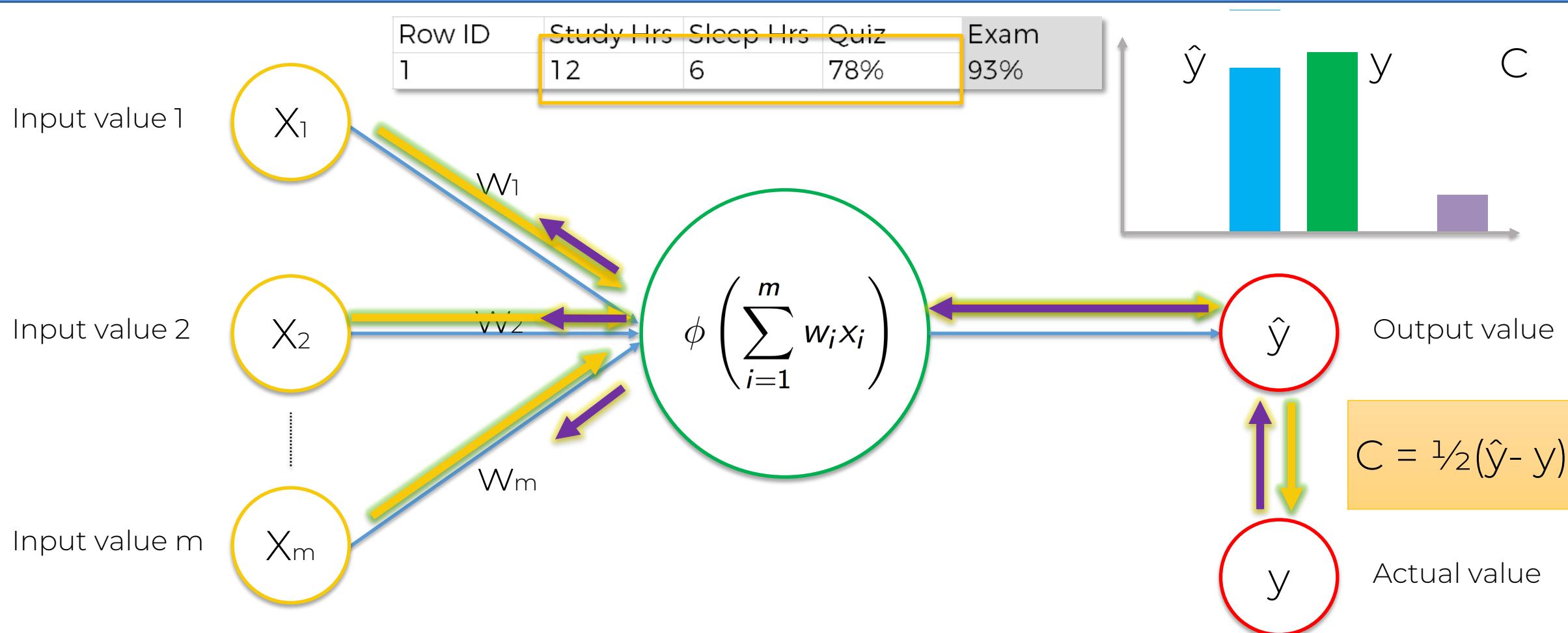
How do Neural Networks learn?



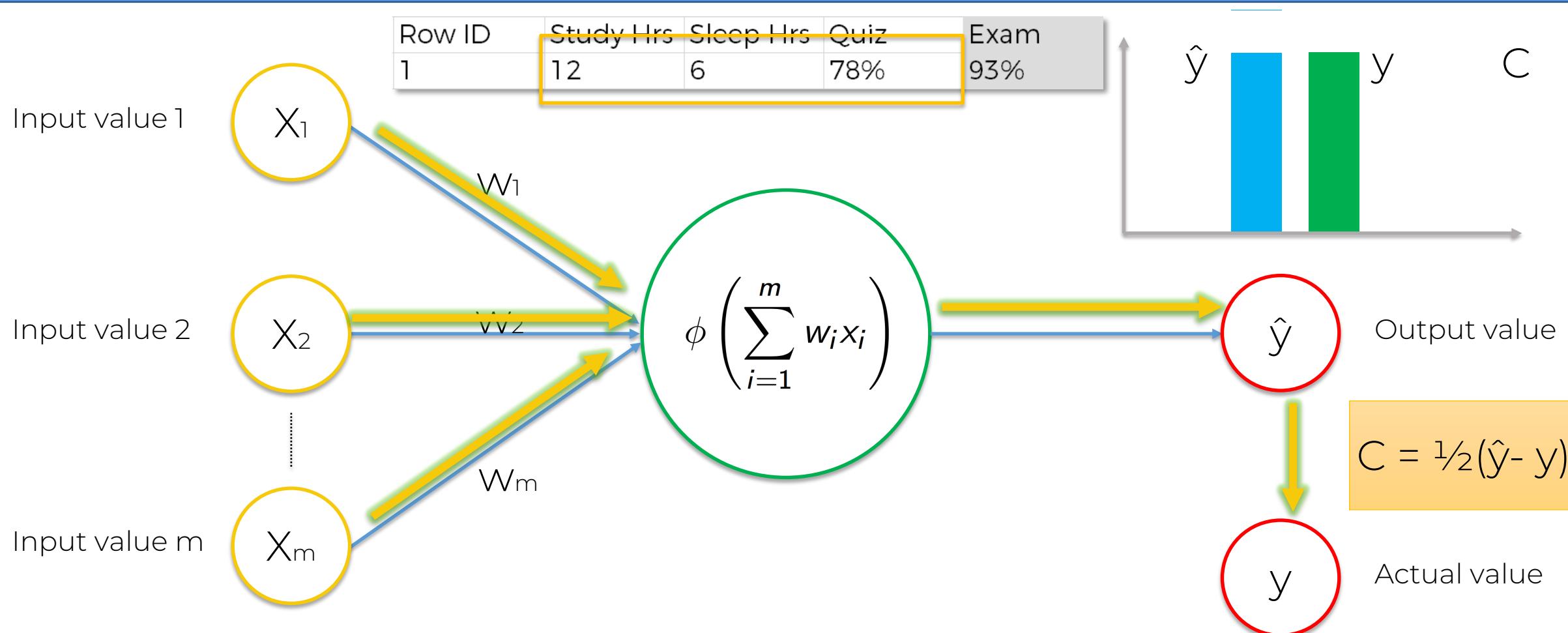
How do Neural Networks learn?



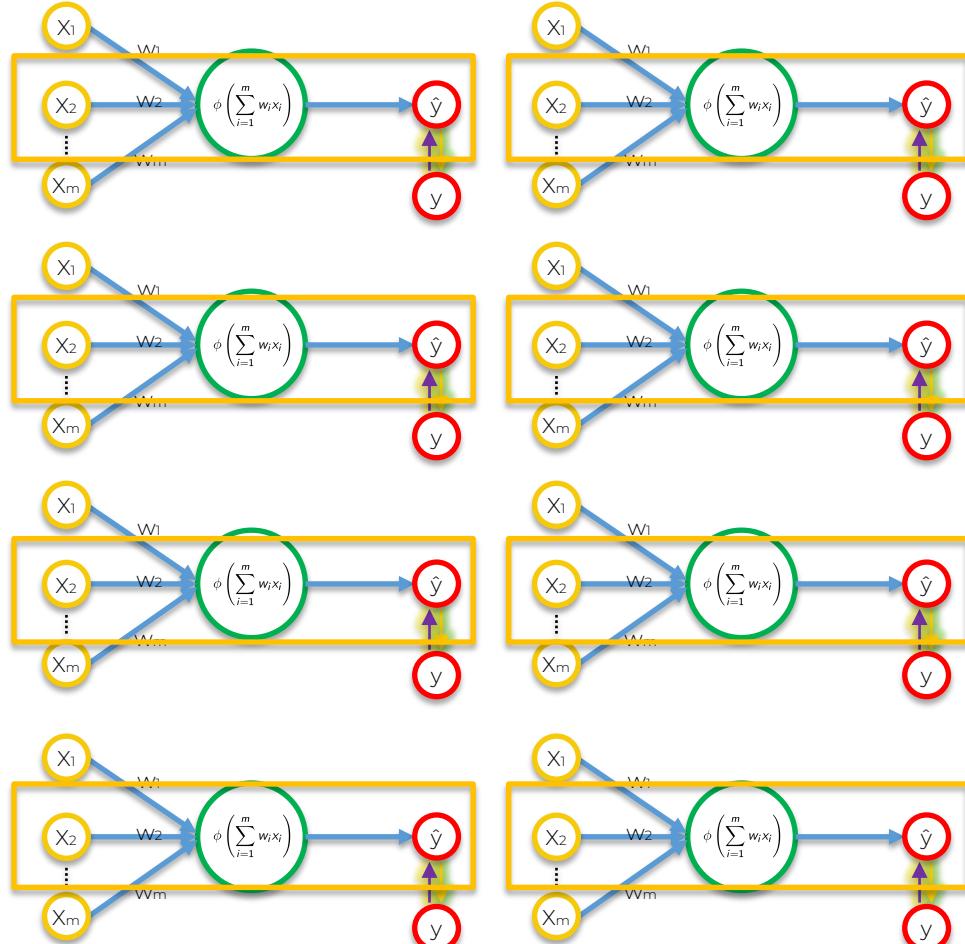
How do Neural Networks learn?



How do Neural Networks learn?



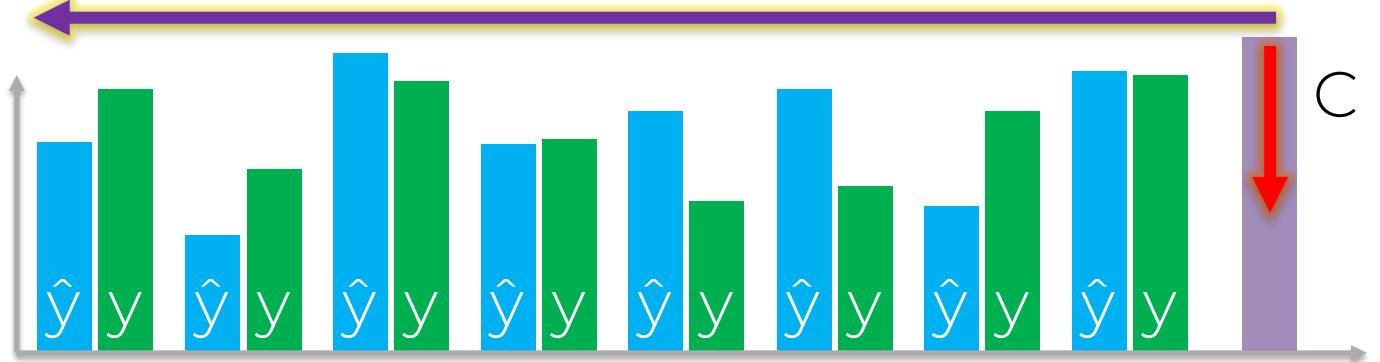
How do Neural Networks learn?



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	51	3	67%	75%
5	0	10	58%	51%
6	5	0	70%	60%
7	92	5	82%	89%
8	57	8	91%	97%

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

Adjust w_1, w_2, w_3



How do Neural Networks learn?

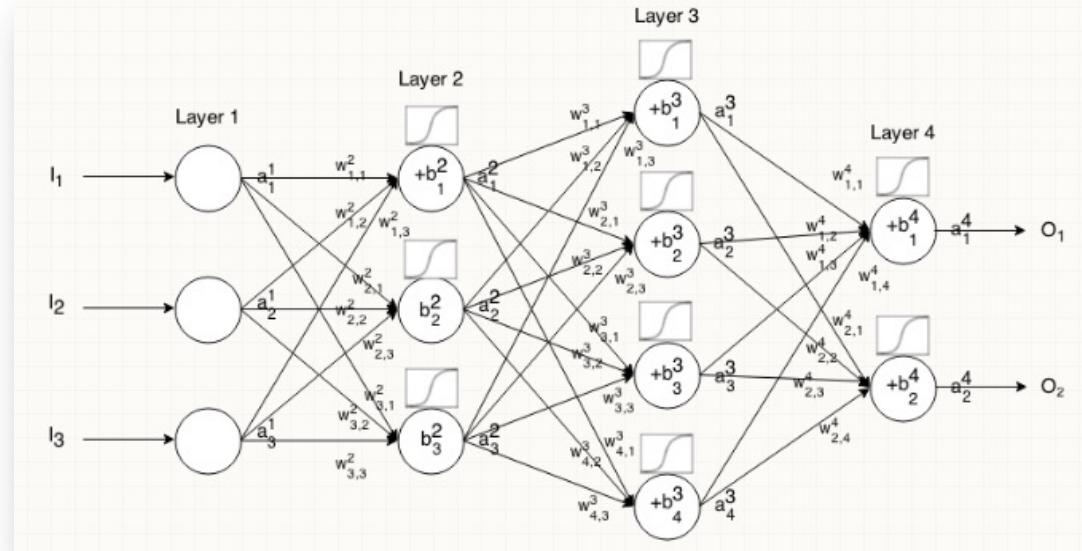
Additional Reading:

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

CrossValidated (2015)

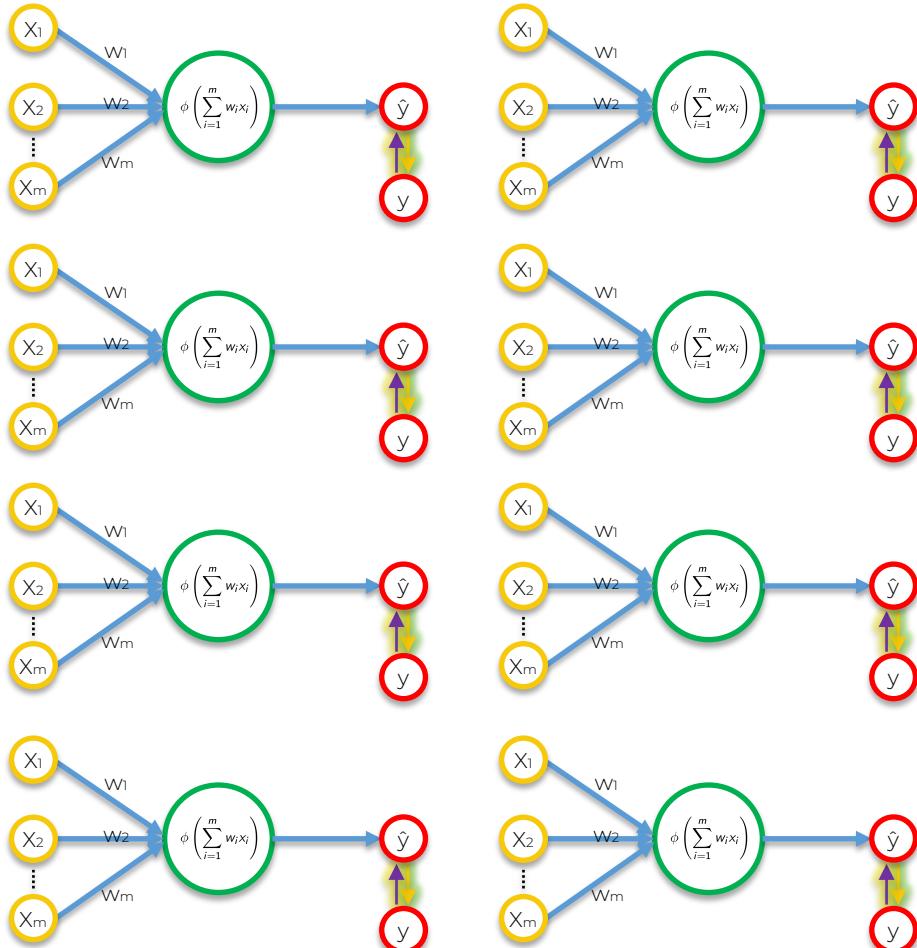
Link:

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



Gradient Descent

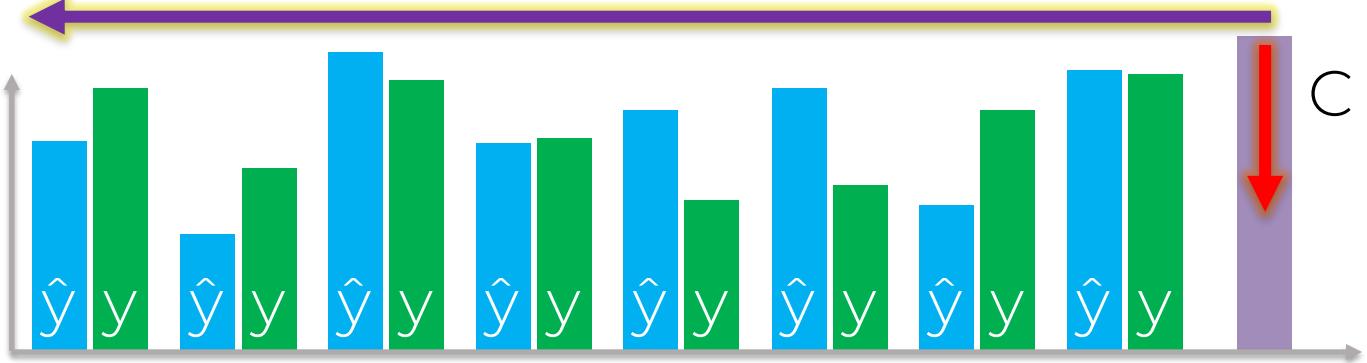
Gradient Descent



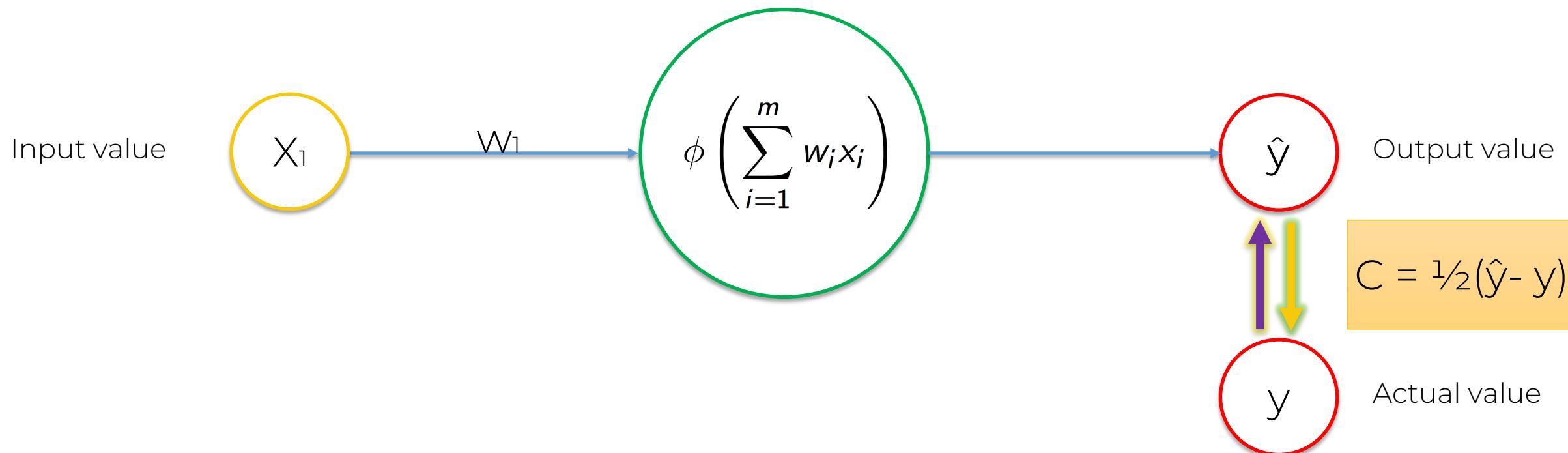
Row ID	Study Hrs	Sleep Hrs	Quiz	Exam
1	12	6	78%	93%
2	22	6.5	24%	68%
3	115	4	100%	95%
4	31	9	67%	75%
5	0	10	58%	51%
6	5	8	78%	60%
7	92	6	82%	89%
8	57	8	91%	97%

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

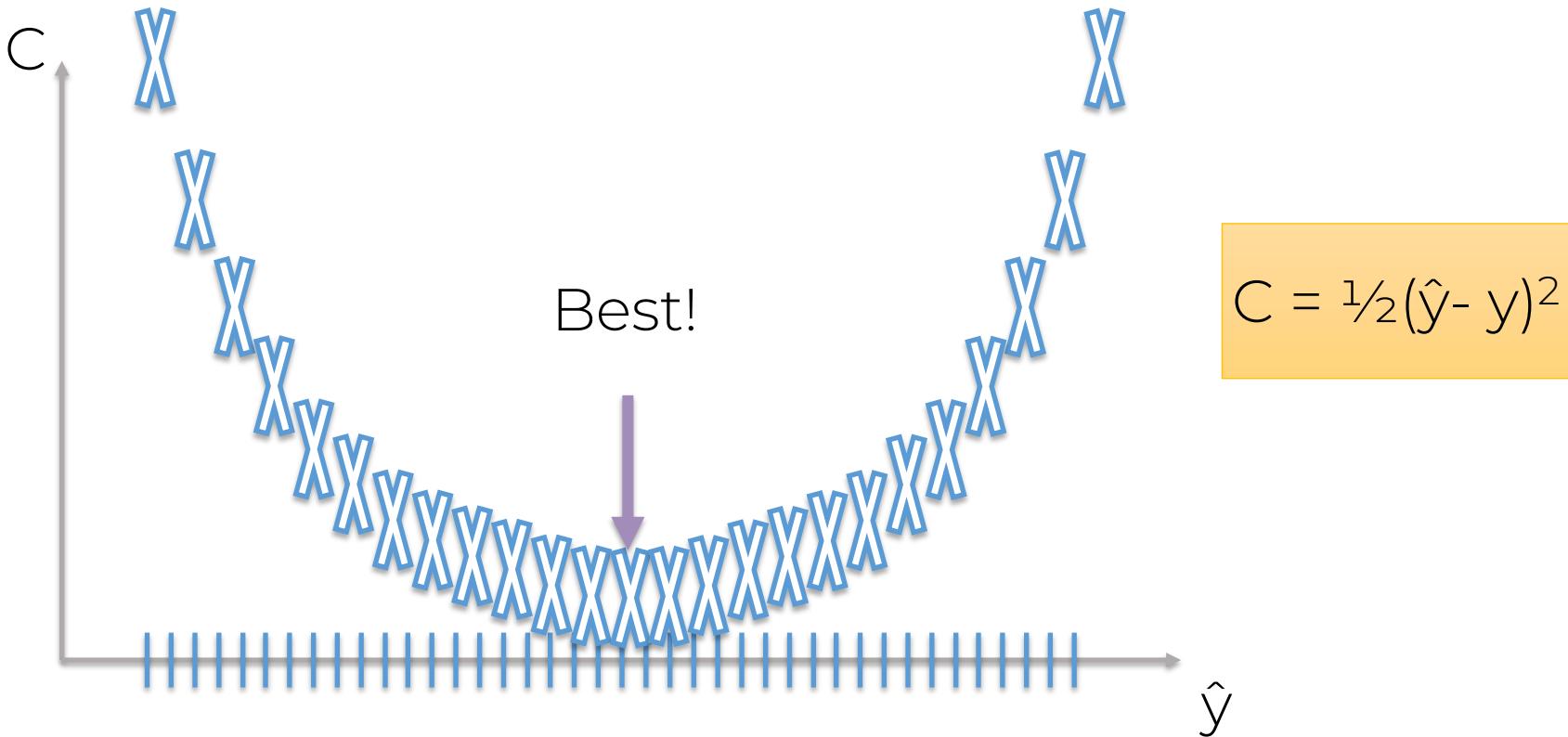
Adjust w_1, w_2, w_3



Gradient Descent



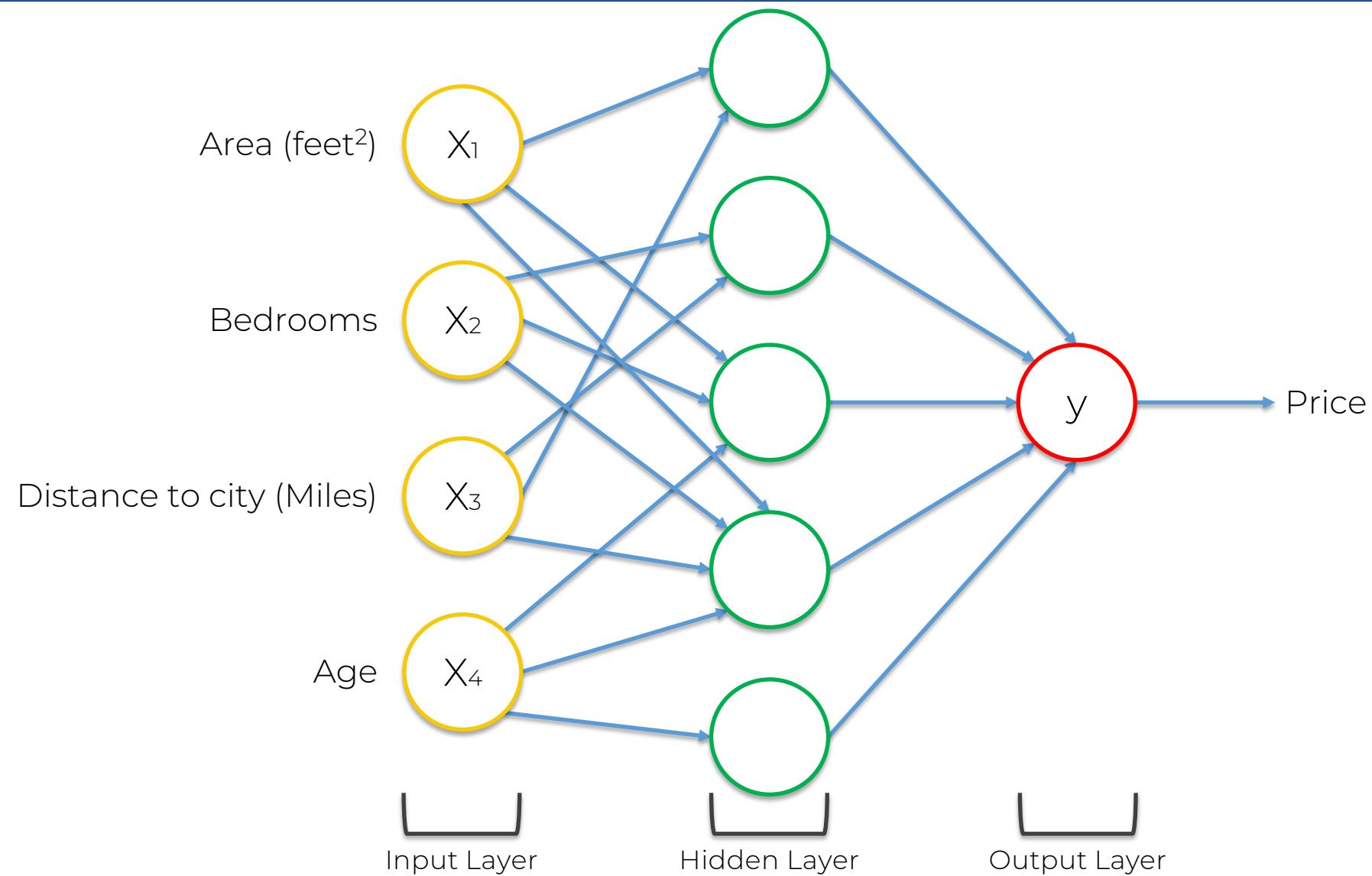
Gradient Descent



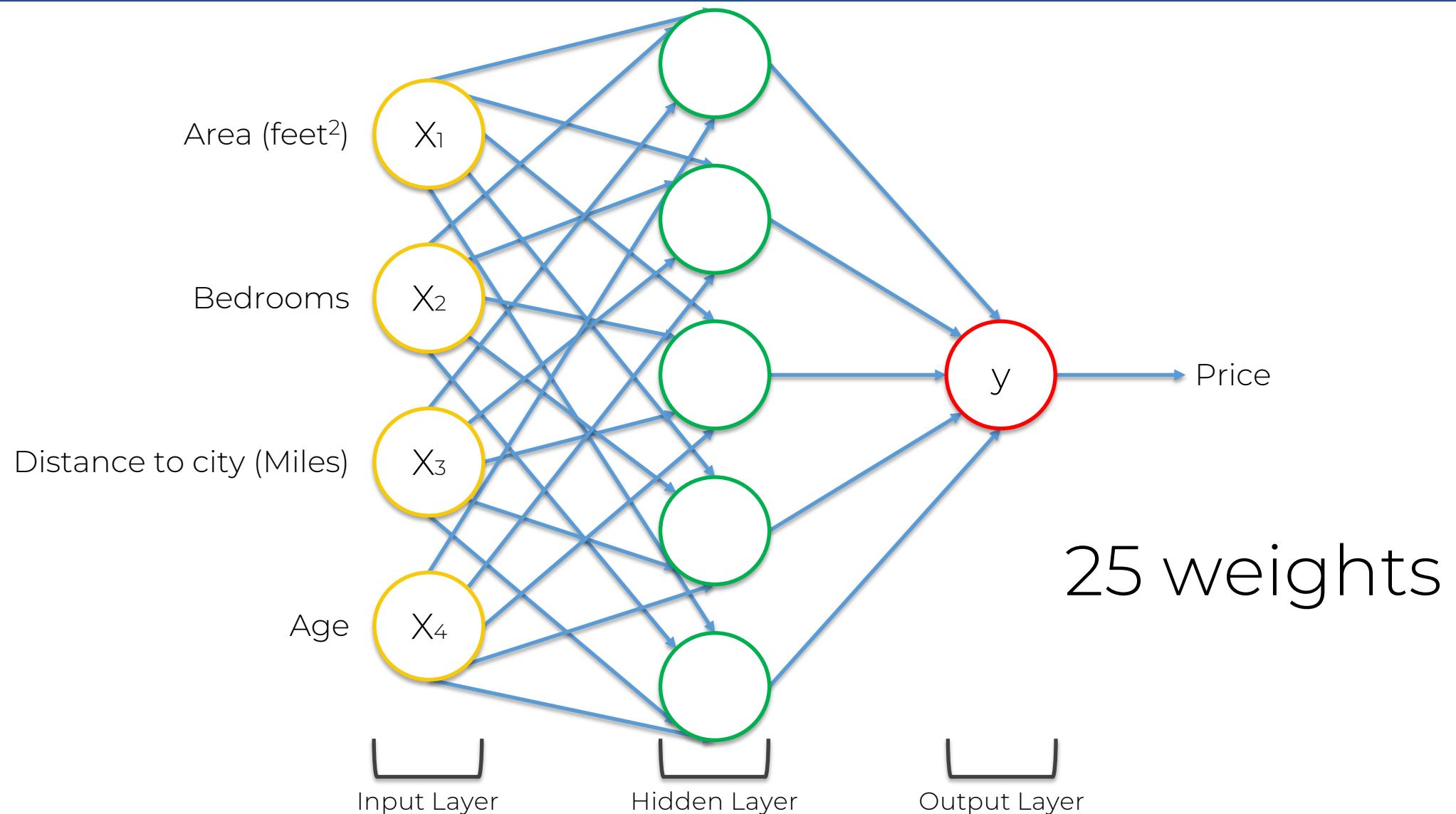
Gradient Descent

Curse of Dimensionality

Gradient Descent



Gradient Descent



Gradient Descent

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

Sunway TaihuLight: World's fastest Super Computer

93 PFLOPS

$$93 \times 10^{15}$$

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

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

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



Gradient Descent

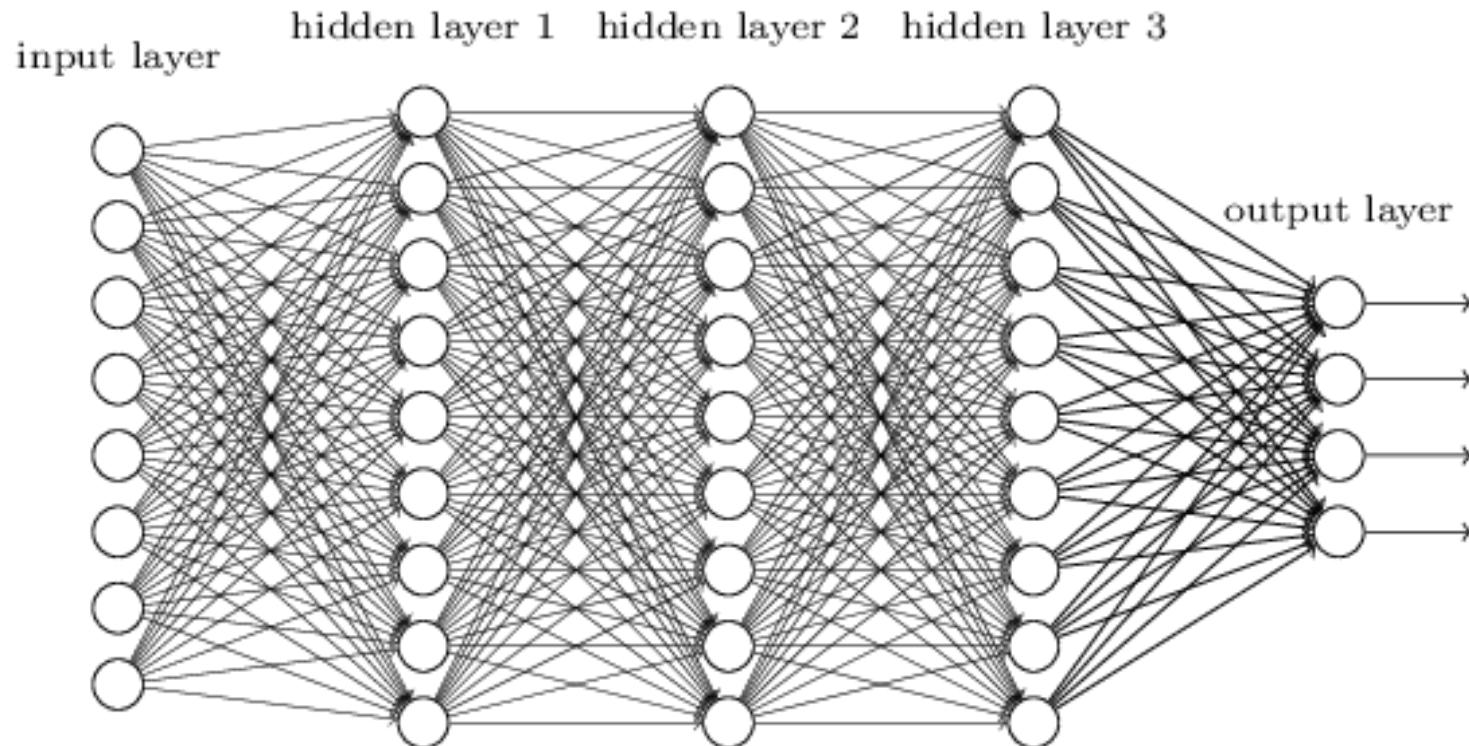
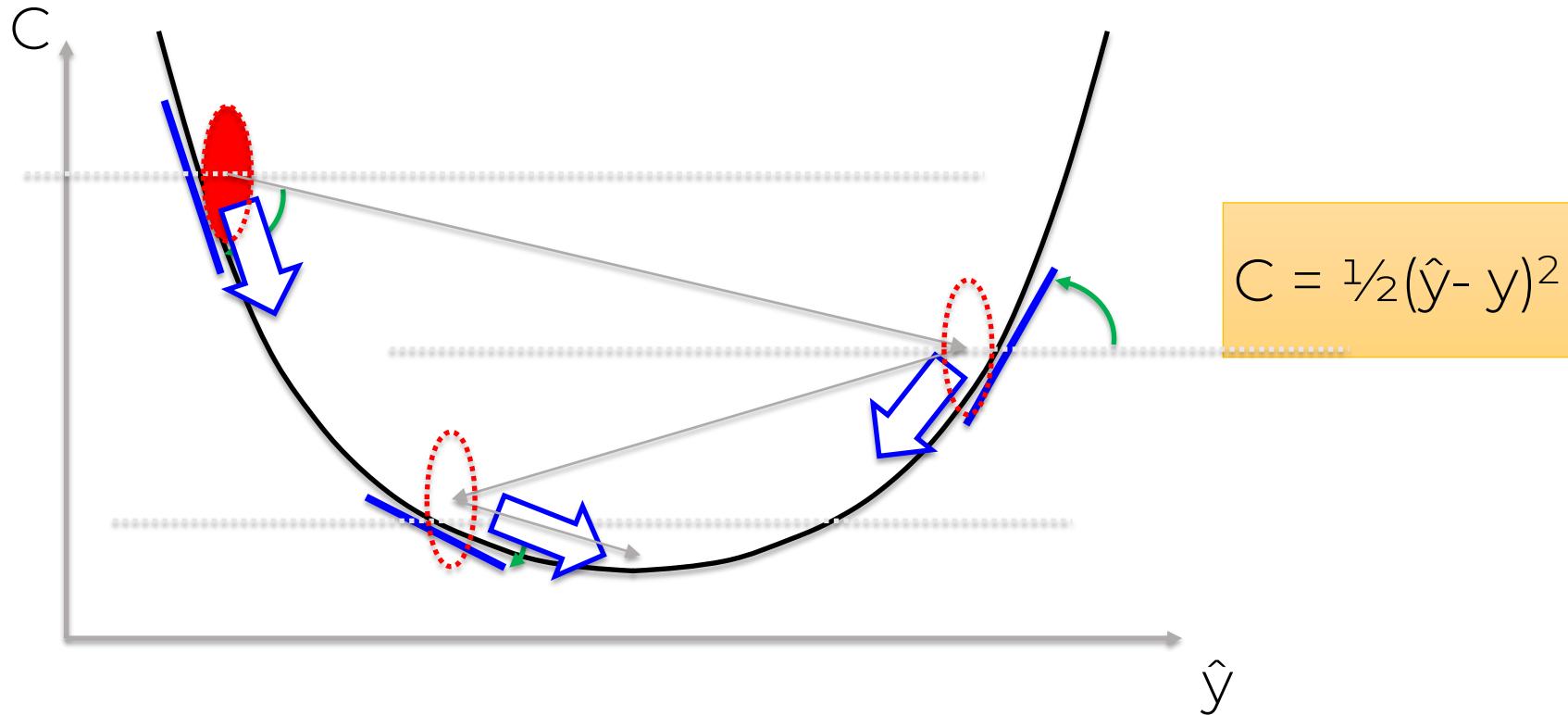


Image Source: neuralnetworksanddeeplearning.com

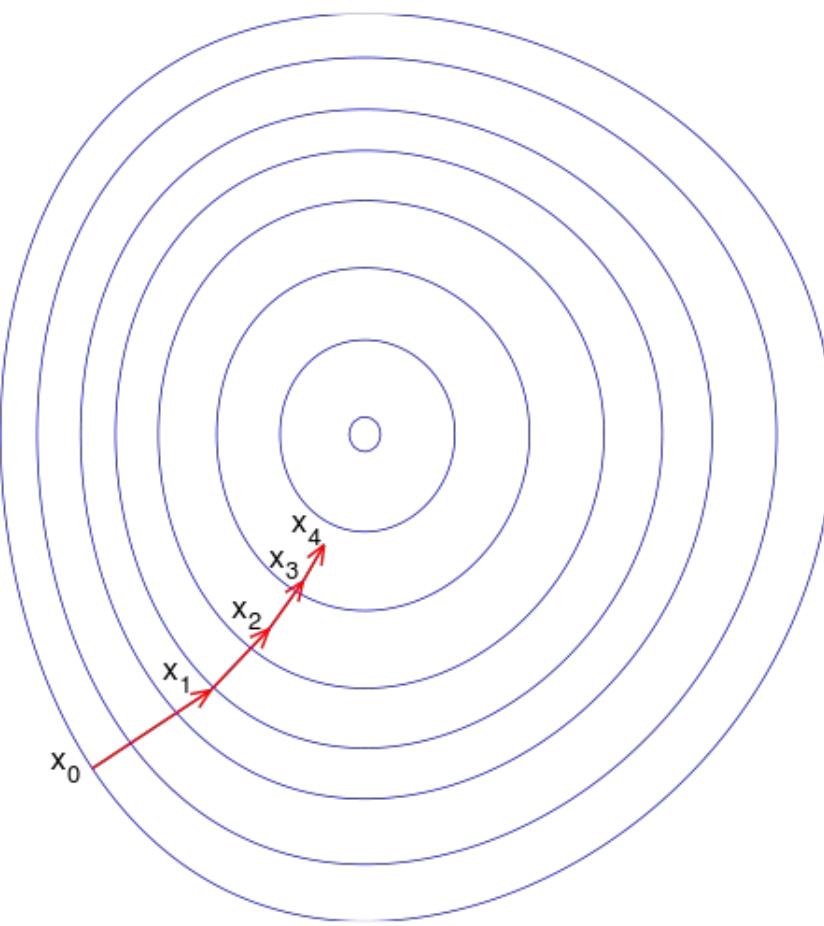
Gradient Descent

Gradient Descent

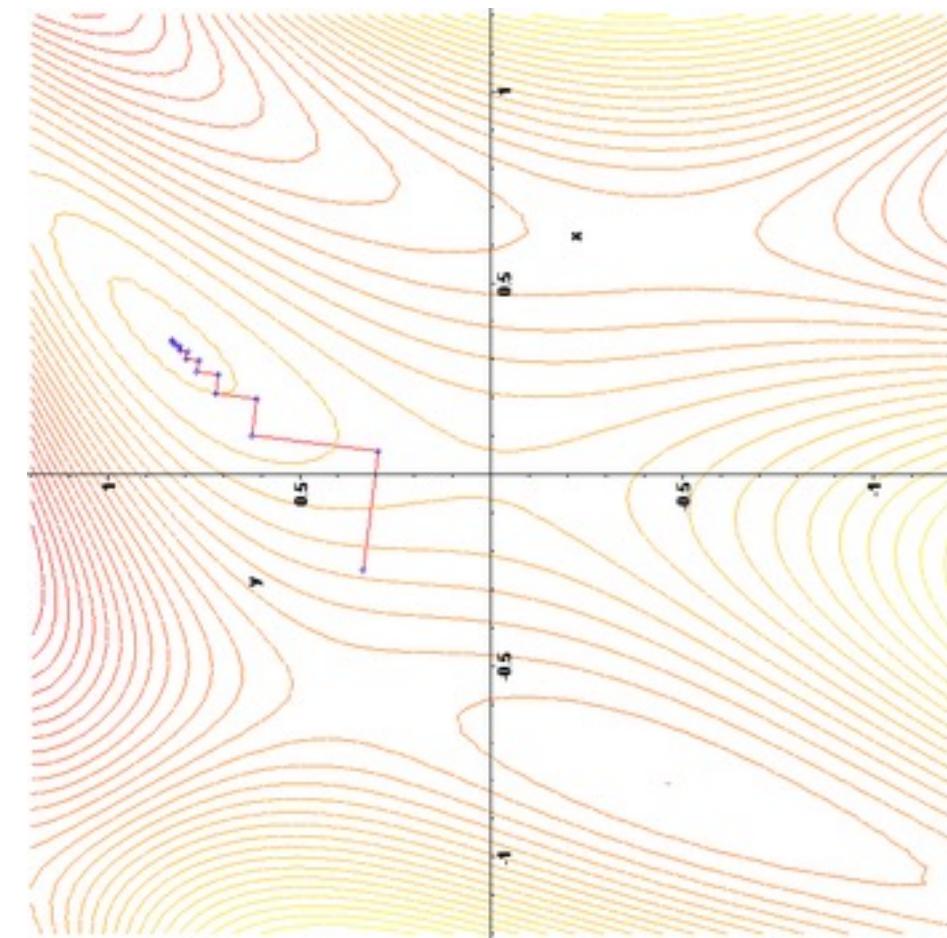
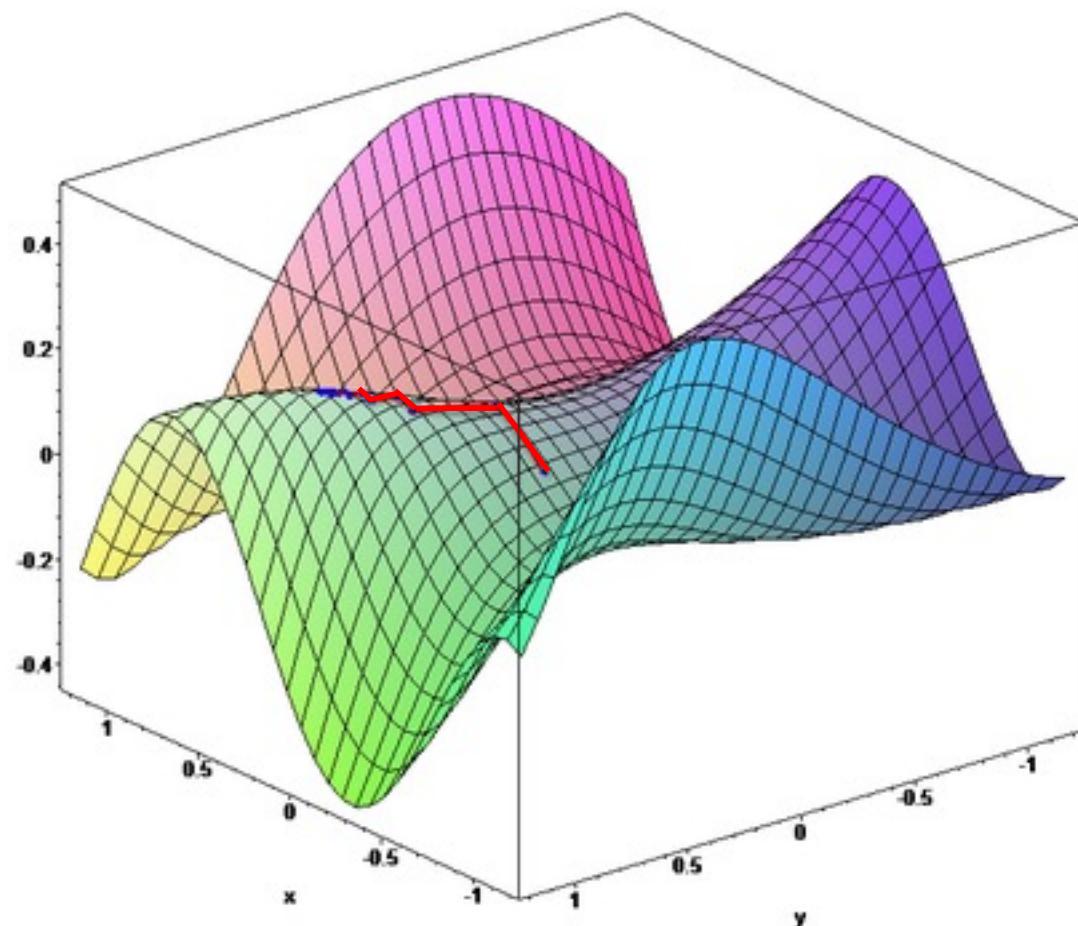
Gradient Descent



Gradient Descent

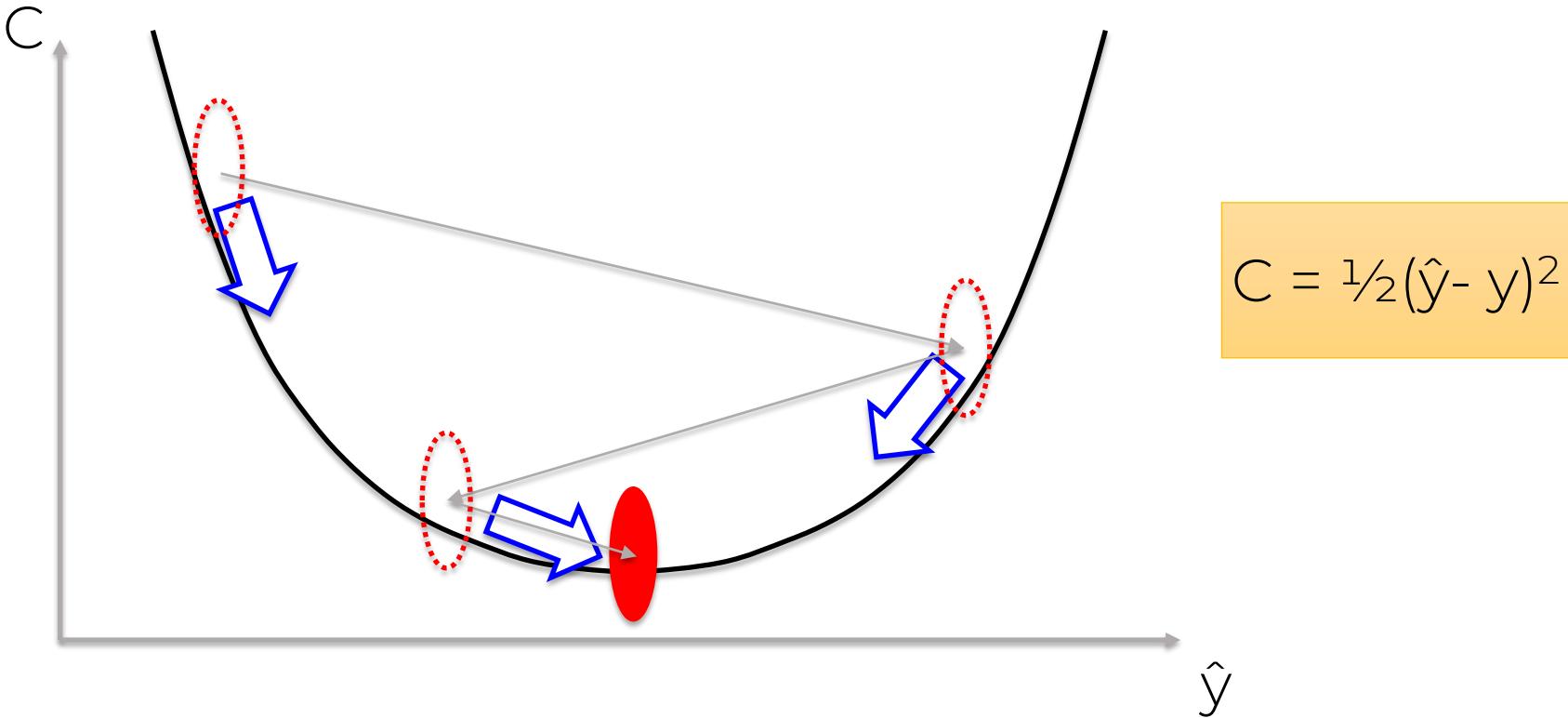


Gradient Descent

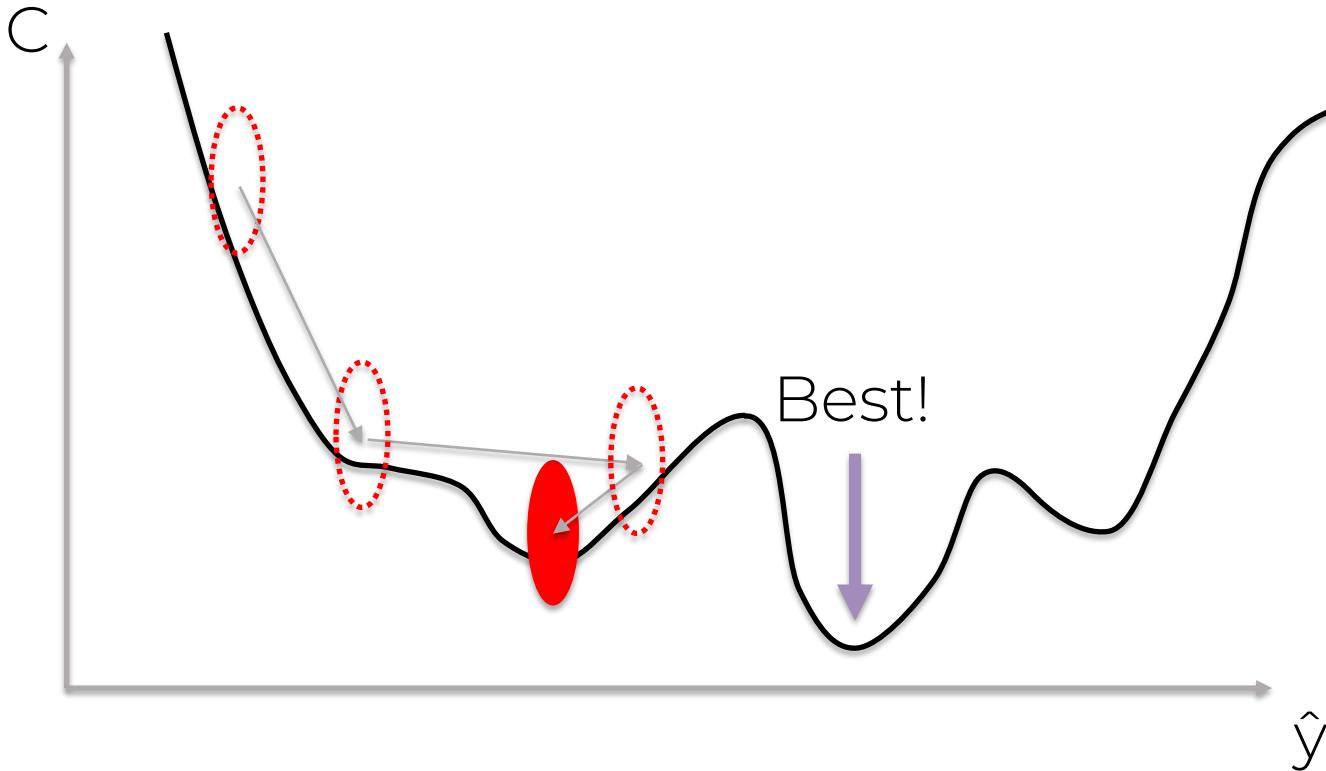


Stochastic Gradient Descent

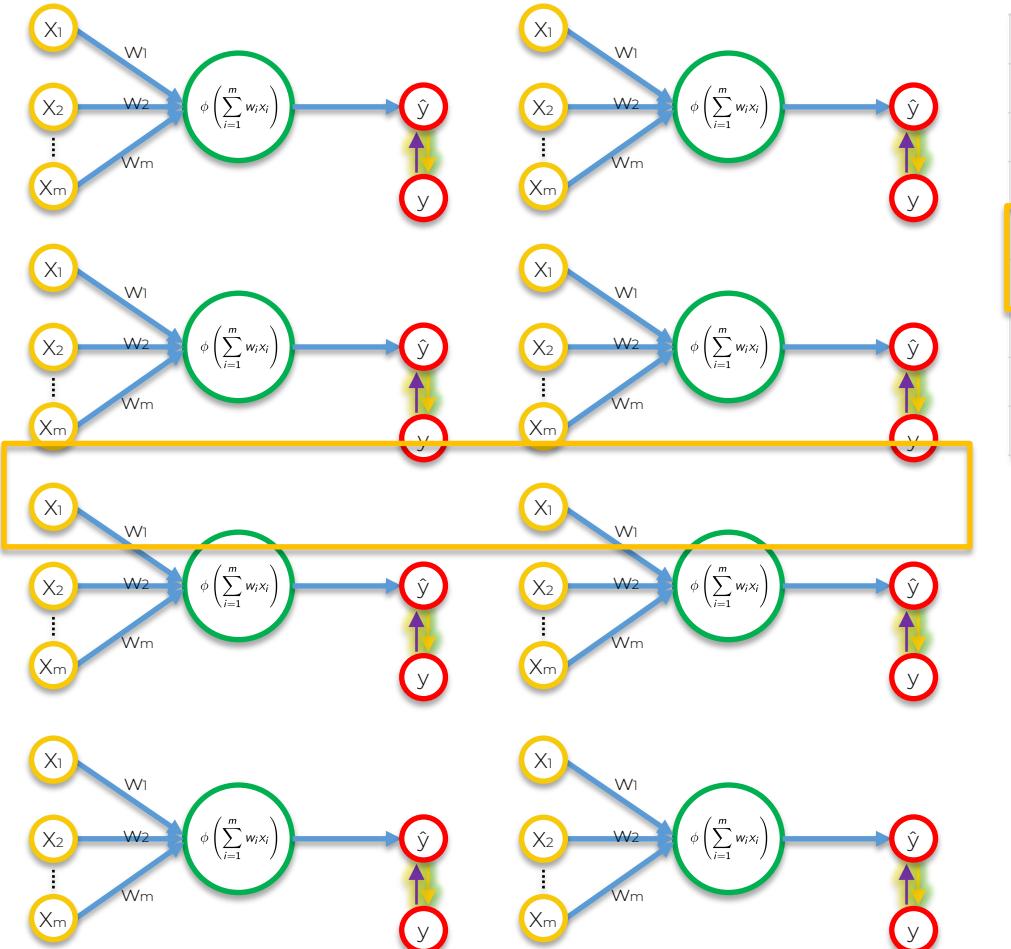
Stochastic Gradient Descent



Stochastic Gradient Descent



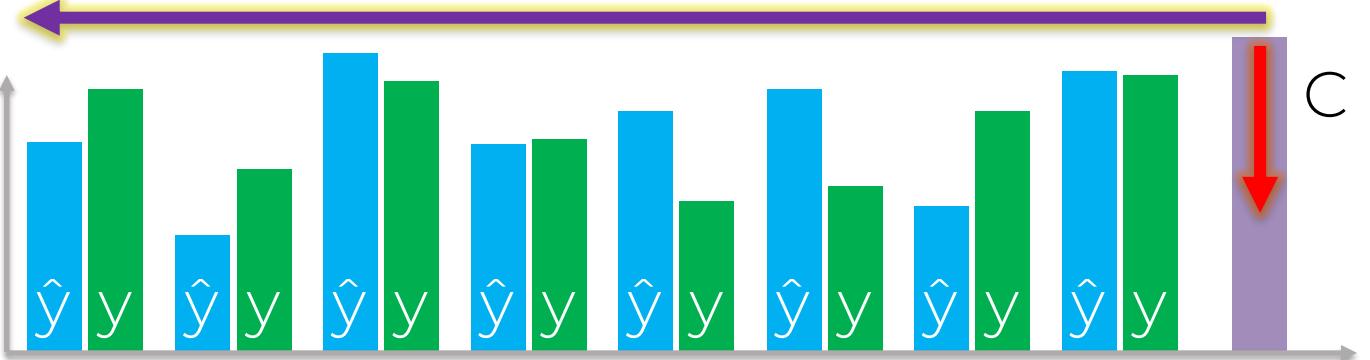
Stochastic Gradient Descent



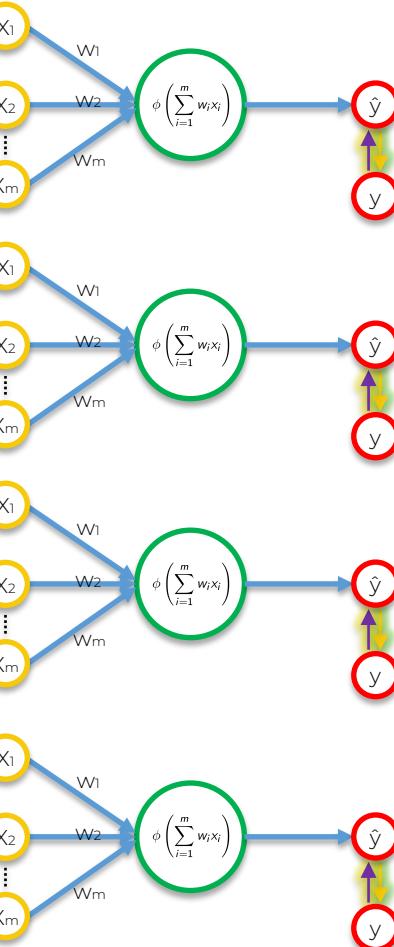
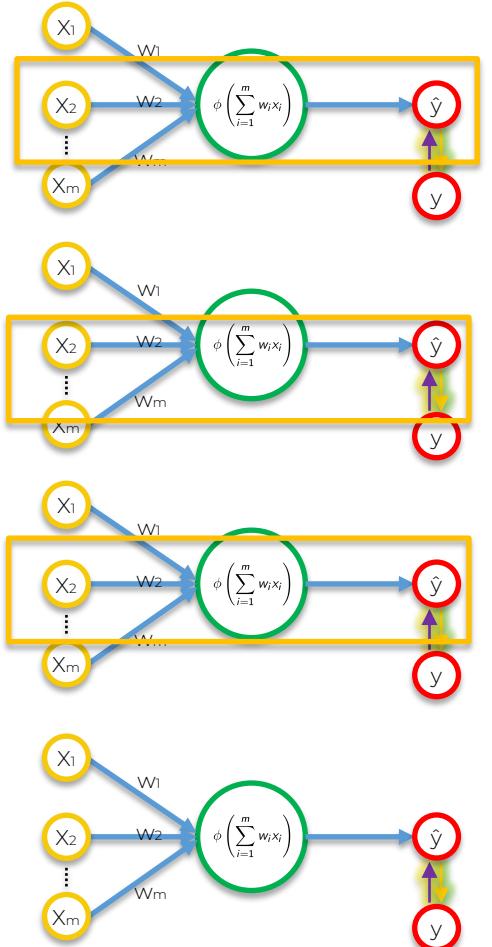
Row ID	Study Hrs	Sleep Hrs	Quiz	Exam
1	12	6	78%	93%
2	22	6.5	24%	68%
3	115	4	100%	95%
4	31	9	67%	75%
5	0	10	58%	51%
6	5	8	78%	60%
7	92	6	82%	89%
8	57	8	91%	97%

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

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

Upd w's ←

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%

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

Stochastic Gradient Descent

Stochastic Gradient Descent

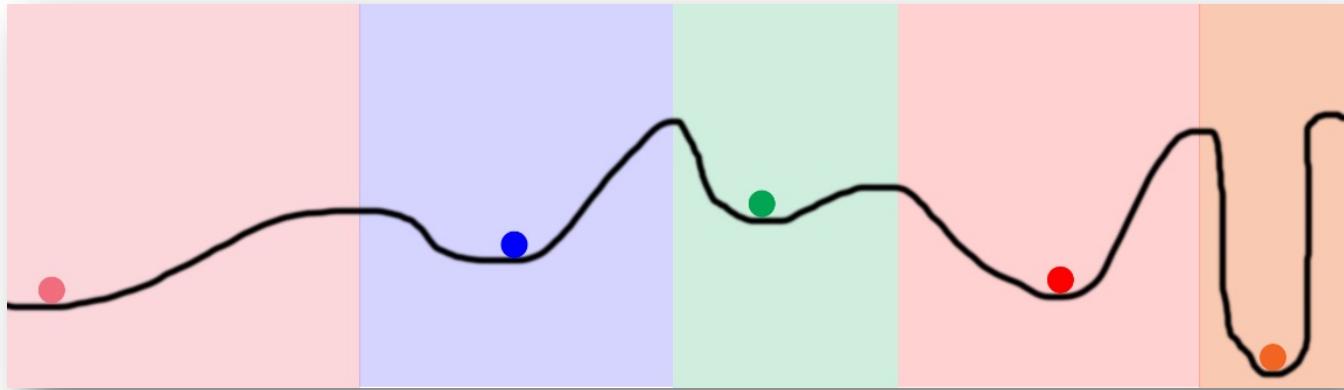
Additional Reading:

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

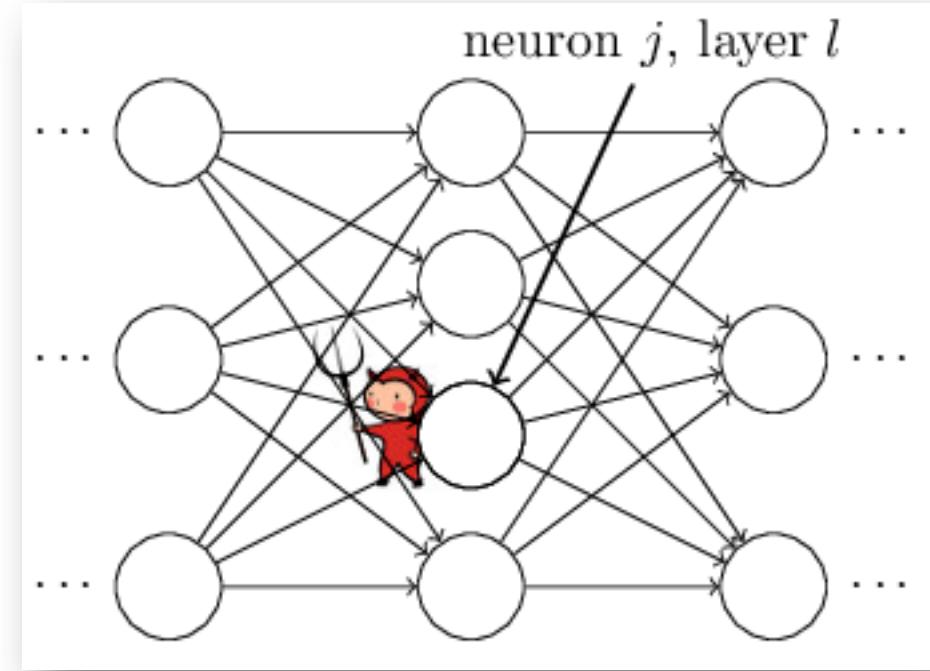
Additional Reading:

Neural Networks and Deep Learning

Michael Nielsen (2015)

Link:

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



Backpropagation

Gradient Descent

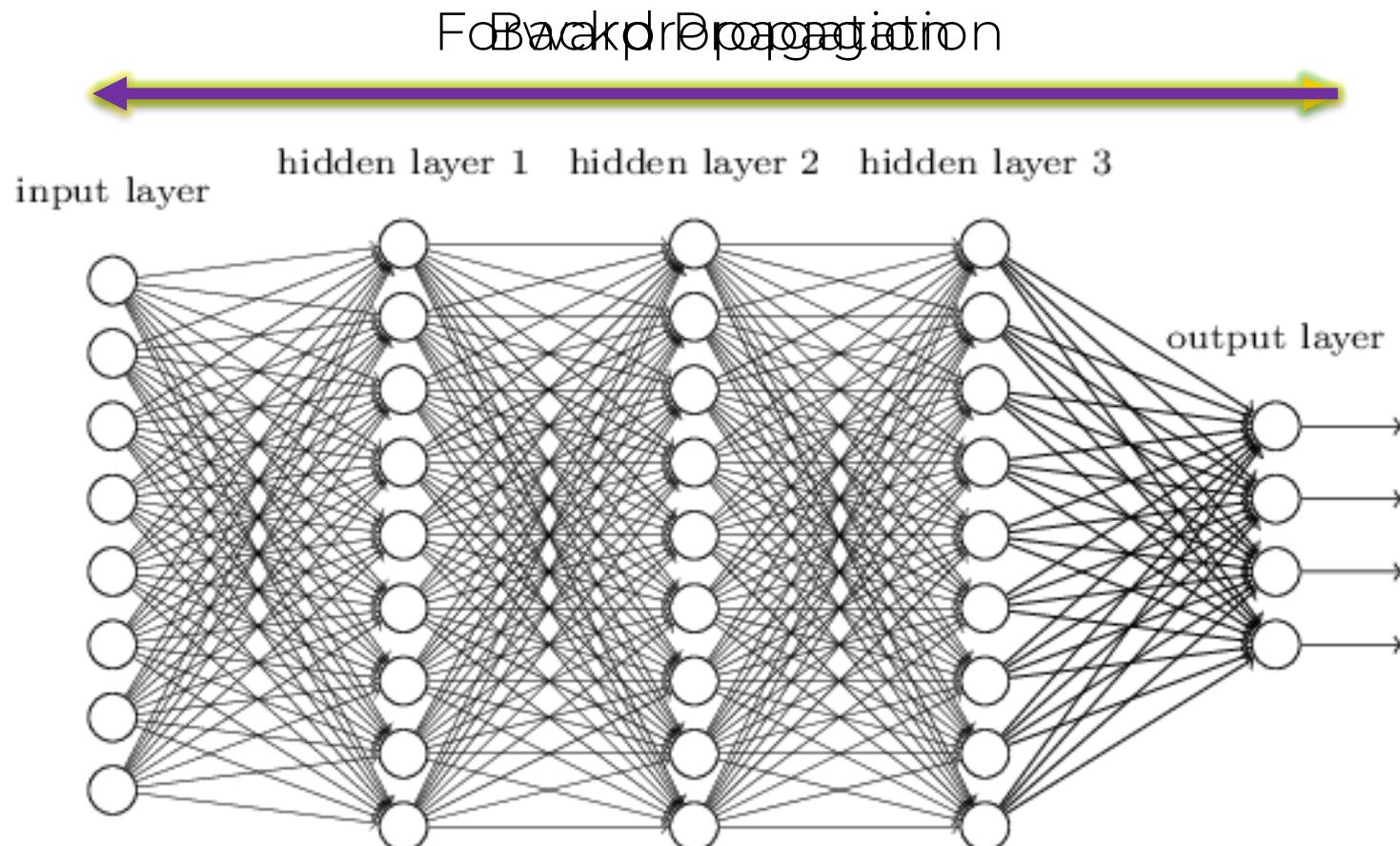


Image Source: neuralnetworksanddeeplearning.com

Stochastic Gradient Descent

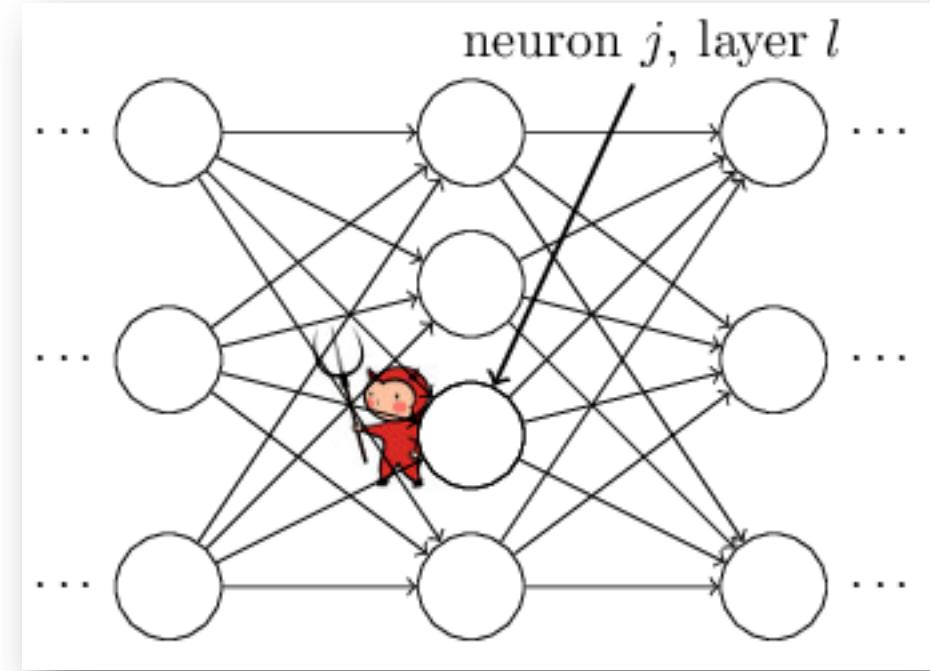
Additional Reading:

Neural Networks and Deep Learning

Michael Nielsen (2015)

Link:

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



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.

CONVOLUTIONAL NEURAL NETWORKS (CNNs)

Plan of Attack

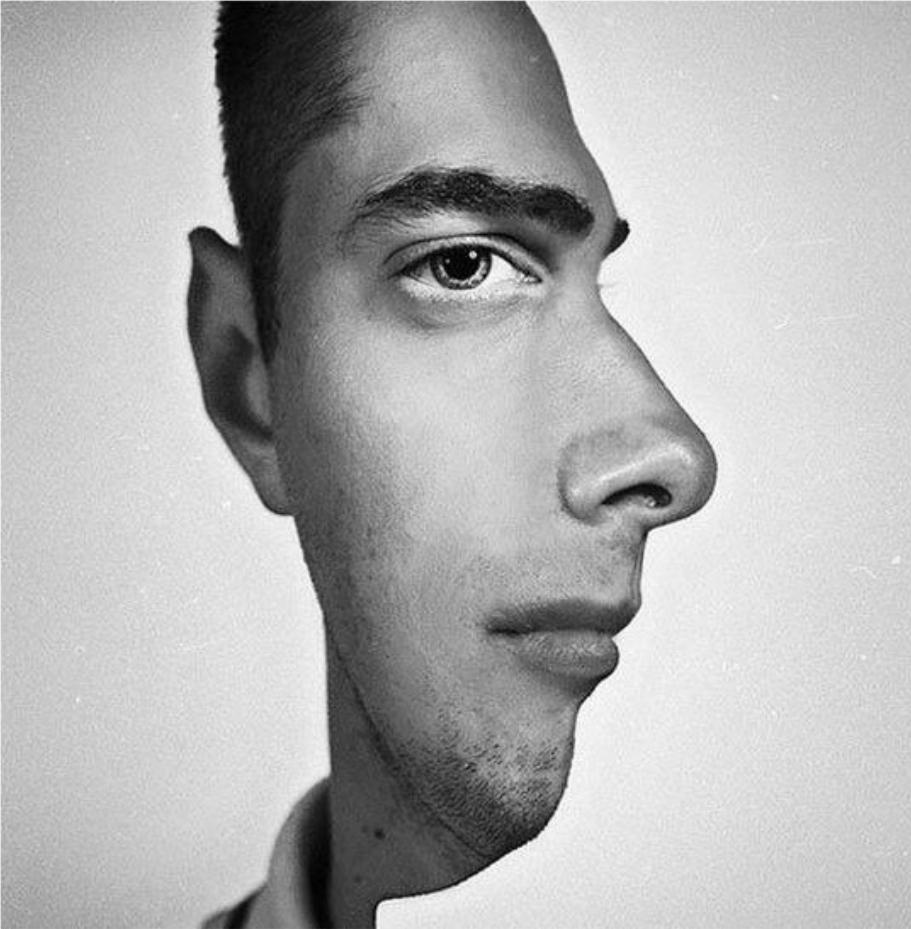
Plan of Attack

What we will learn in this section:

- What are Convolutional Neural Networks?
- Step 1 - Convolution Operation
- Step 1(b) - ReLU Layer
- Step 2 - Pooling
- Step 3 - Flattening
- Step 4 - Full Connection
- Summary
- EXTRA: Softmax & Cross-Entropy

Convolutional Neural Networks

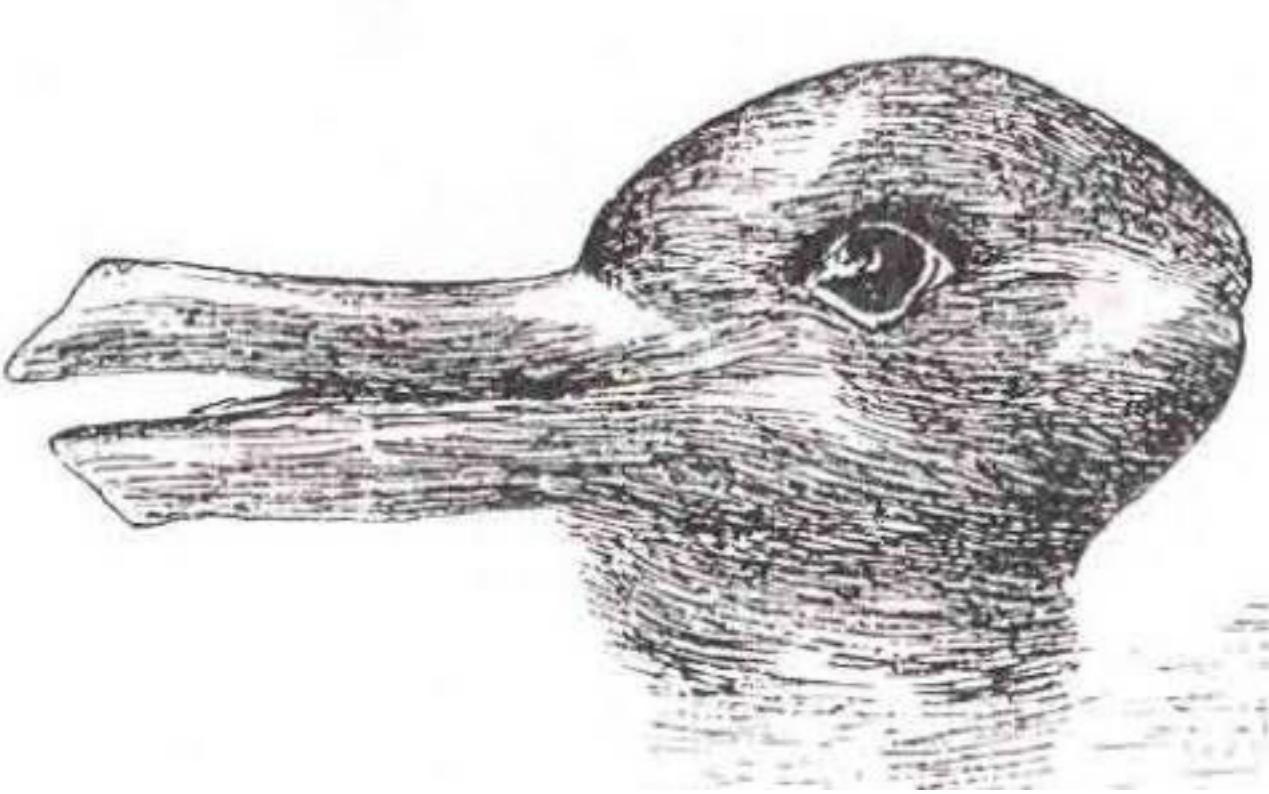
Convolutional Neural Networks



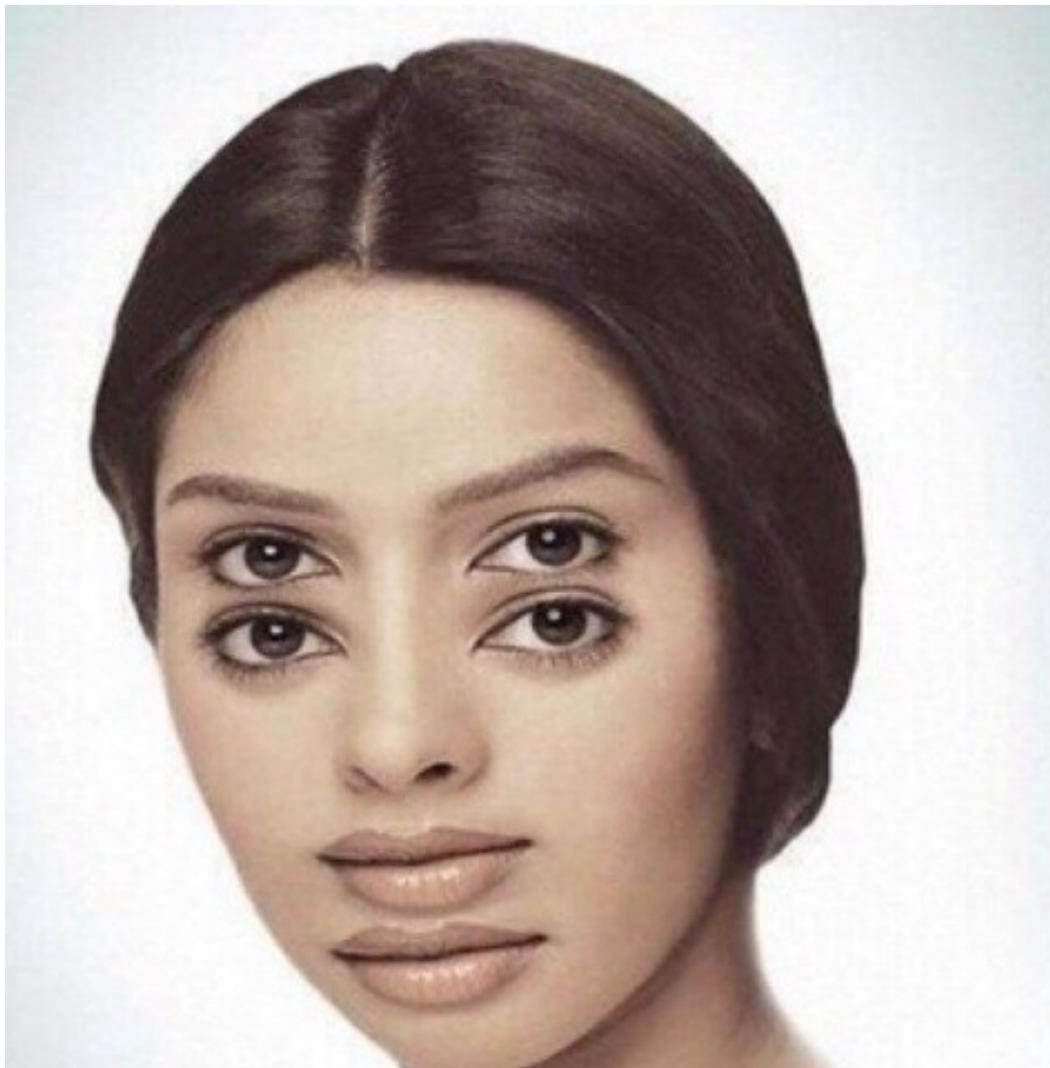
Convolutional Neural Networks



Convolutional Neural Networks



Convolutional Neural Networks



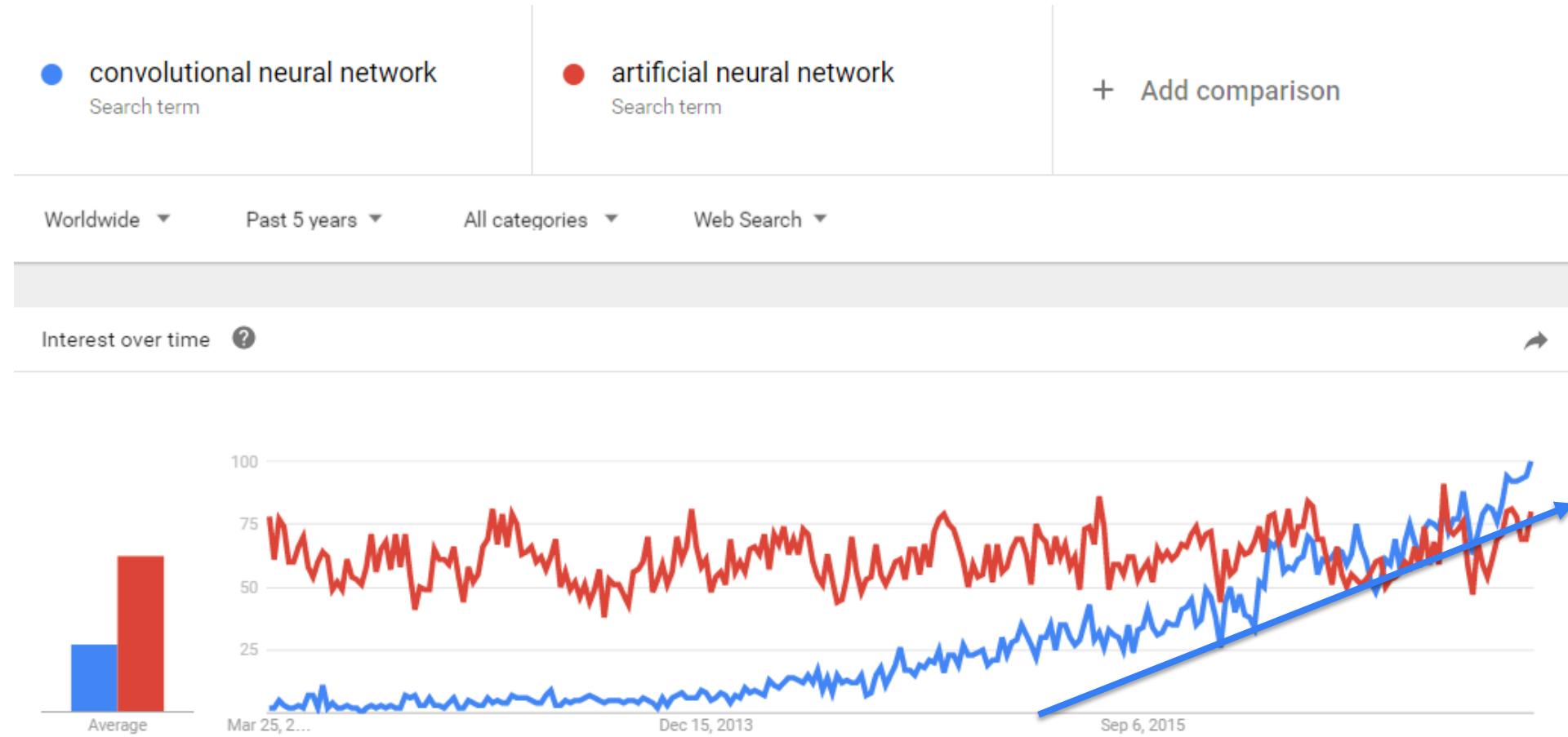
Convolutional Neural Networks

Examples from the test set
(with the network's guesses)



Image Source: a talk by Geoffrey Hinton

Convolutional Neural Networks



Source: google trends

Convolutional Neural Networks



Yann Lecun

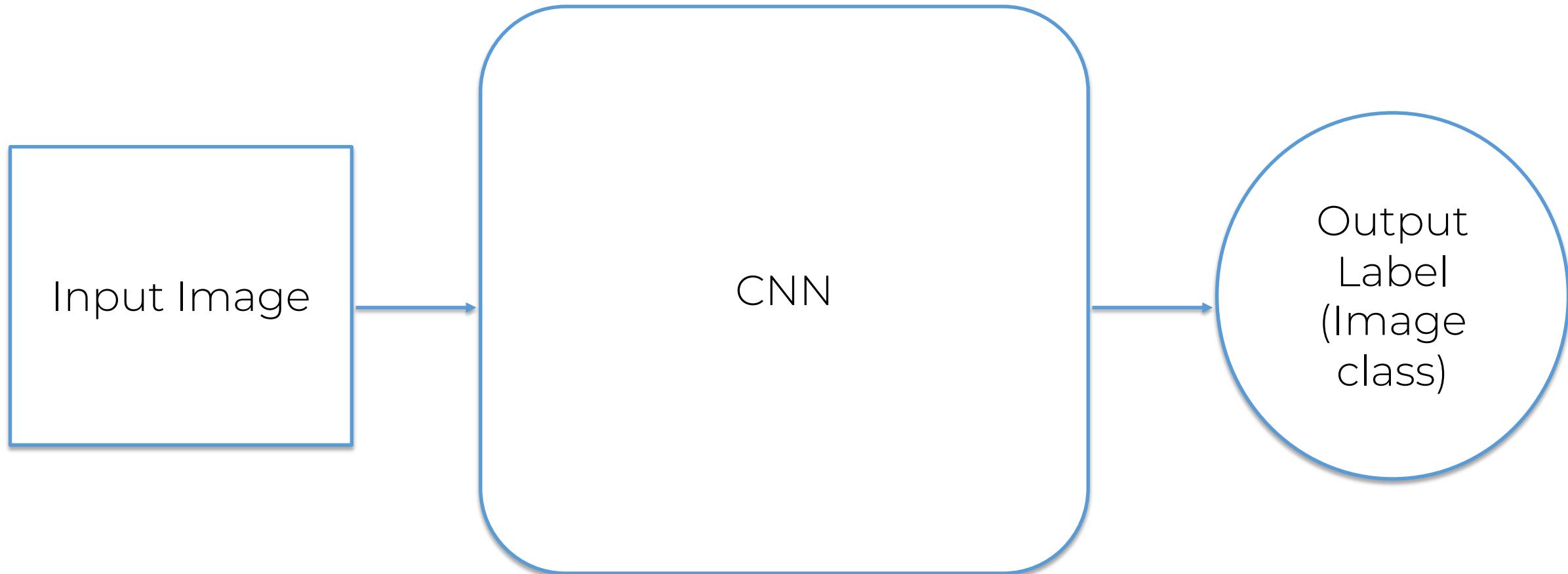
Convolutional Neural Networks

Google

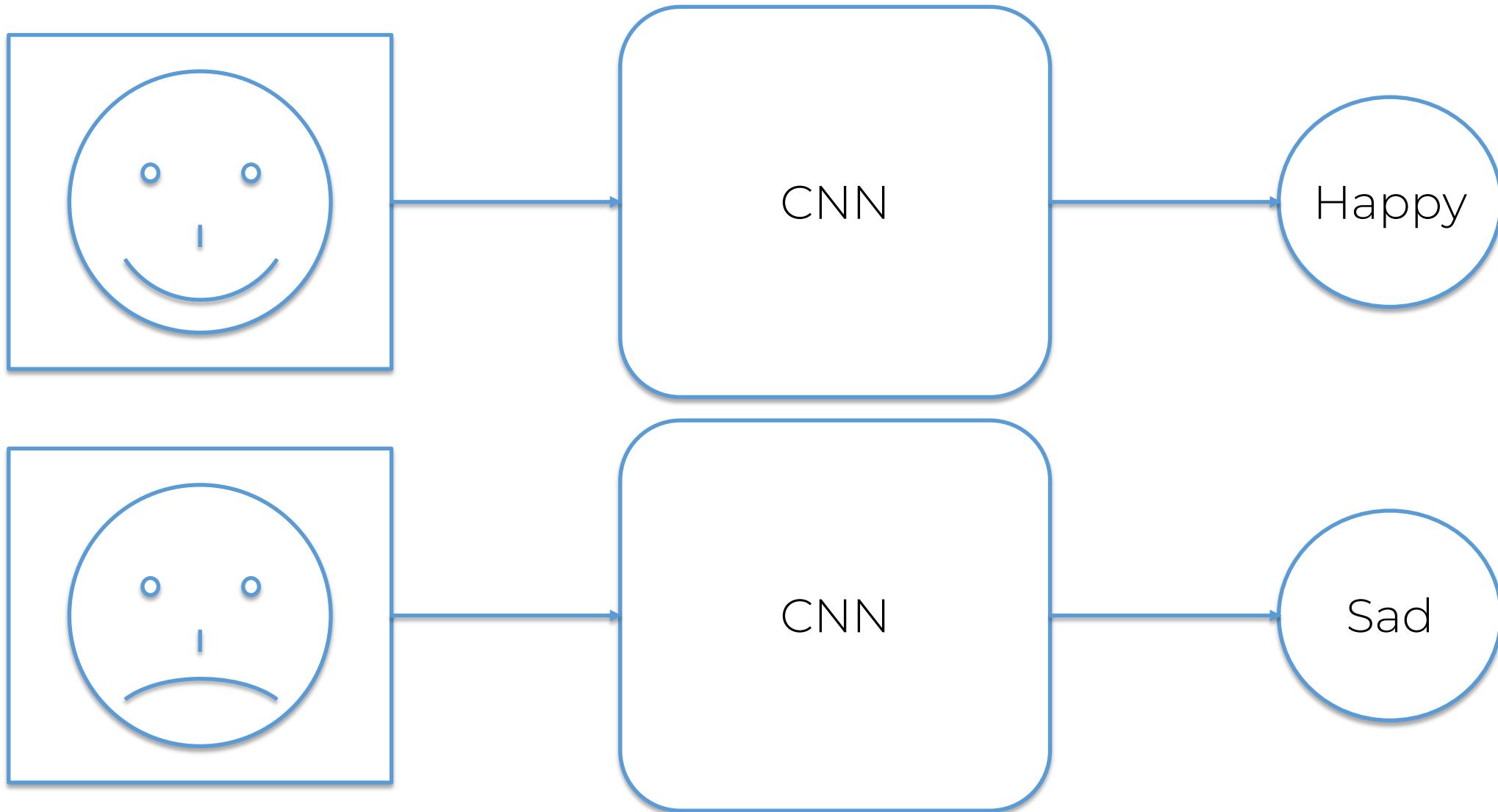
Facebook



Convolutional Neural Networks

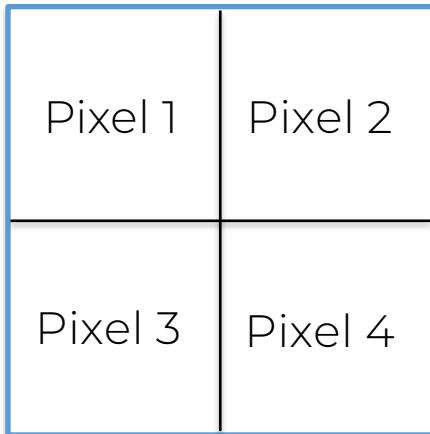


Convolutional Neural Networks

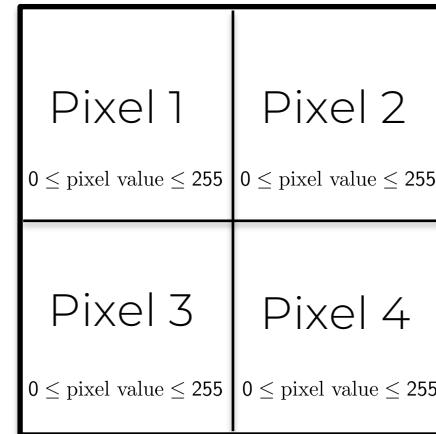


Convolutional Neural Networks

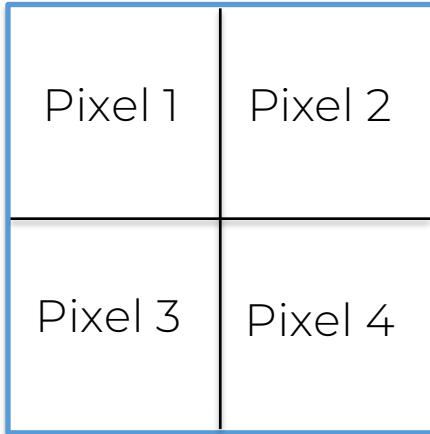
B / W Image 2x2px



2d array



Colored Image 2x2px

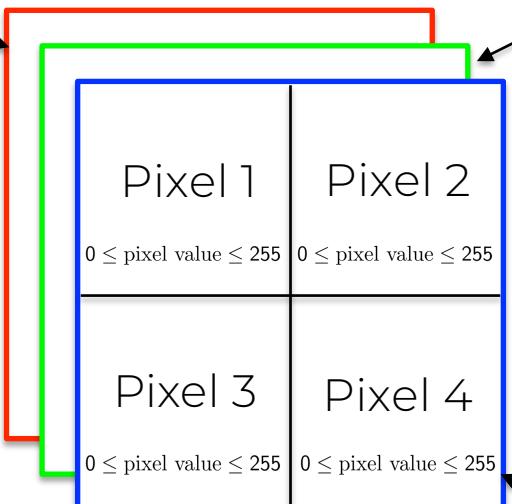


3d array

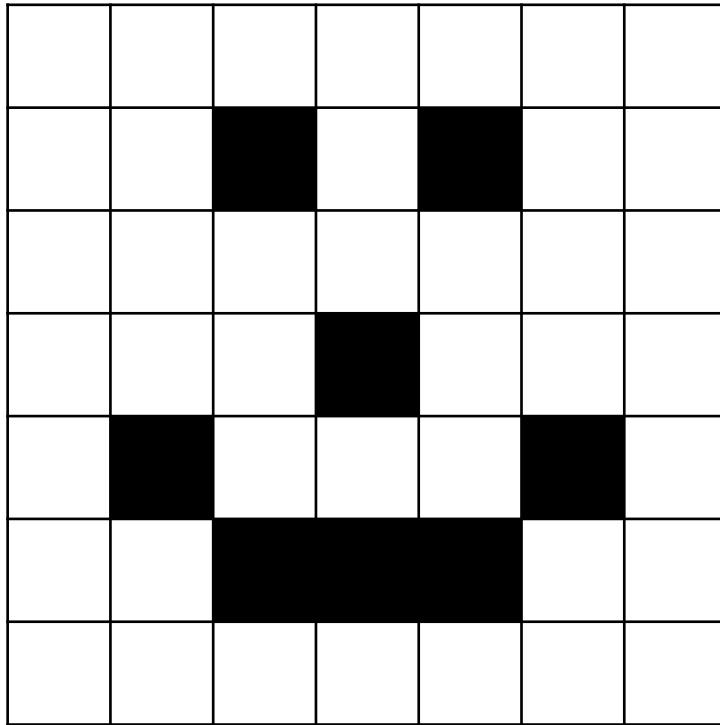
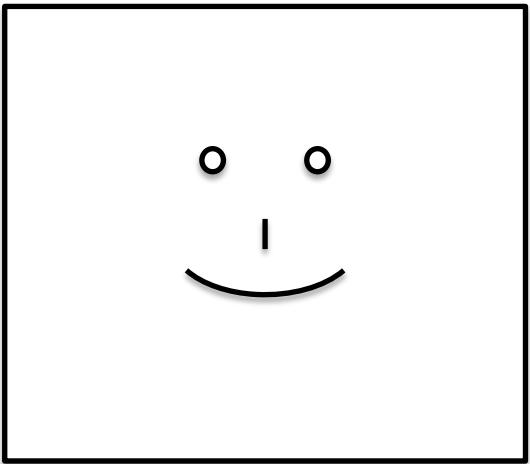
Red channel

Green channel

Blue channel

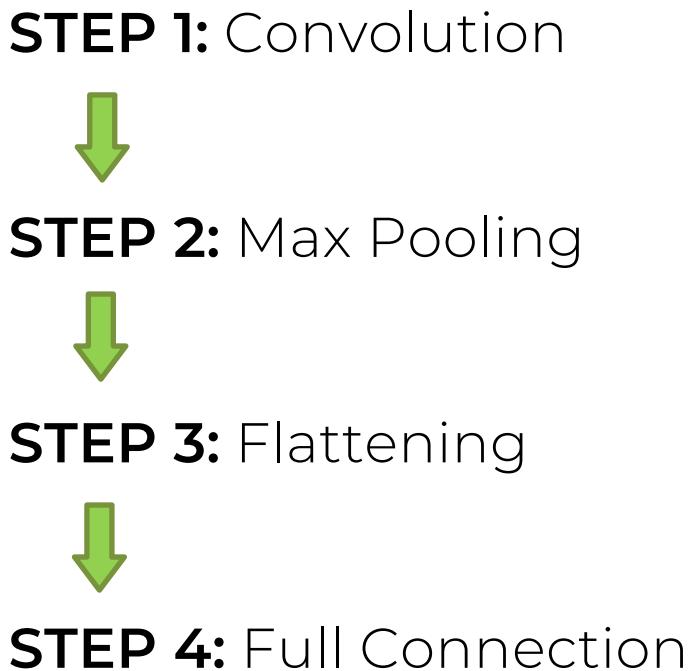


Convolutional Neural Networks



0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	0	0	1	0	0
0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0

Convolutional Neural Networks



Convolutional Neural Networks

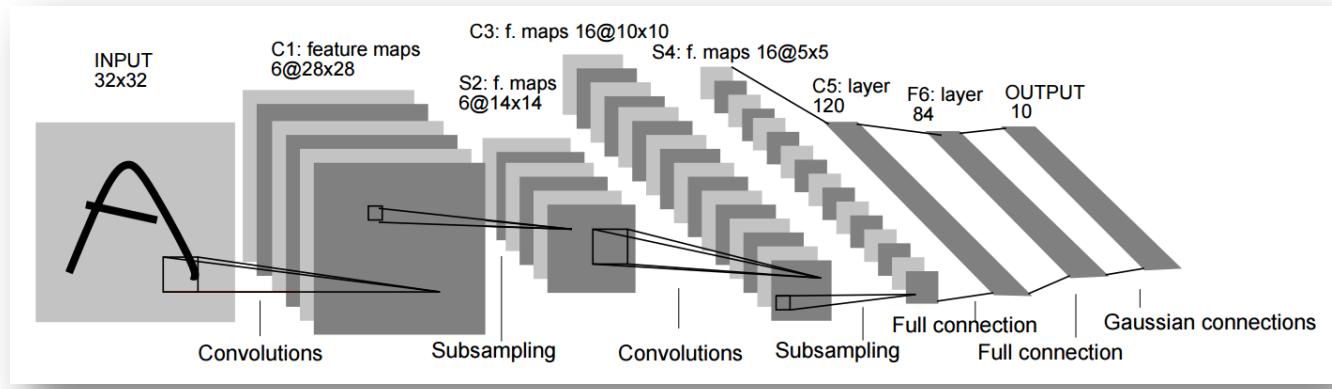
Additional Reading:

*Gradient-Based Learning
Applied to Document
Recognition*

By Yann LeCun et al. (1998)

Link:

<http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf>



Step 1 - Convolution

Step 1 - Convolution

$$(f * g)(t) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau) g(t - \tau) d\tau$$

Step 1 - Convolution

Additional Reading:

*Introduction to
Convolutional Neural
Networks*

By Jianxin Wu (2017)

$$\begin{aligned}\frac{\partial z}{\partial (\text{vec}(\mathbf{y})^T)}(F^T \otimes I) &= \left((F \otimes I) \frac{\partial z}{\partial \text{vec}(\mathbf{y})} \right)^T \\ &= \left((F \otimes I) \text{vec} \left(\frac{\partial z}{\partial Y} \right) \right)^T \\ &= \text{vec} \left(I \frac{\partial z}{\partial Y} F^T \right)^T \\ &= \text{vec} \left(\frac{\partial z}{\partial Y} F^T \right)^T,\end{aligned}$$

Link:

<http://cs.nju.edu.cn/wujx/paper/CNN.pdf>

Step 1 - Convolution

0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

Input
Image

0	0	1
1	0	0
0	1	1

Feature
Detector

Step 1- Convolution

0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	0	0	1	0	0
0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0



Input
Image

0	0	1
1	0	0
0	1	1

Feature
Detector



0				

Feature Map

Step 1 - Convolution

0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	0	0	1	0	0
0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0



Input
Image

0	0	1
1	0	0
0	1	1

Feature
Detector



0	1						

Feature Map

Step 1 - Convolution

0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	0	0	1	0	0
0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0



Input
Image

0	0	1
1	0	0
0	1	1

Feature
Detector



0	1	0					

Feature Map

Step 1 - Convolution

0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0



Input
Image

0	0	1
1	0	0
0	1	1

Feature
Detector



0	1	0	0

Feature Map

Step 1 - Convolution

0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0



Input
Image

0	0	1
1	0	0
0	1	1

Feature
Detector



0	1	0	0	0

Feature Map

Step 1 - Convolution

0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	0	0	1	0	0
0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0



Input
Image

0	0	1
1	0	0
0	1	1

Feature
Detector



0	1	0	0	0
0				

Feature Map

Step 1 - Convolution

0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	0	0	1	0	0
0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0



Input
Image

0	0	1
1	0	0
0	1	1

Feature
Detector



0	1	0	0	0
0	1			

Feature Map

Step 1 - Convolution

0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	0	0	1	0	0
0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0



Input
Image

0	0	1
1	0	0
0	1	1

Feature
Detector



0	1	0	0	0
0	1	1		

Feature Map

Step 1 - Convolution

0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0



Input
Image

0	0	1
1	0	0
0	1	1

Feature
Detector



0	1	0	0	0
0	1	1	1	

Feature Map

Step 1 - Convolution

0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	0	0	1	0	0
0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0



Input
Image

0	0	1
1	0	0
0	1	1

Feature
Detector

=

0	1	0	0	0
0	1	1	1	0

Feature Map

Step 1 - Convolution

0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	0	0	1	0	0
0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0



Input
Image

0	0	1
1	0	0
0	1	1

Feature
Detector



0	1	0	0	0
0	1	1	1	0
1				

Feature Map

Step 1 - Convolution

0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	0	0	1	0	0
0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0



Input
Image

0	0	1
1	0	0
0	1	1

Feature
Detector



0	1	0	0	0
0	1	1	1	0
1	0			

Feature Map

Step 1 - Convolution

0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	0	0	1	0	0
0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0



Input
Image

0	0	1
1	0	0
0	1	1

Feature
Detector



0	1	0	0	0
0	1	1	1	0
1	0	1		

Feature Map

Step 1 - Convolution

0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0



Input
Image

0	0	1
1	0	0
0	1	1

Feature
Detector



0	1	0	0	0
0	1	1	1	0
1	0	1	2	

Feature Map

Step 1 - Convolution

0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	0	0	1	0	0
0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0



Input
Image

0	0	1
1	0	0
0	1	1

Feature
Detector



0	1	0	0	0
0	1	1	1	0
1	0	1	2	1

Feature Map

Step 1 - Convolution

0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	0	0	1	0	0
0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0



Input
Image

0	0	1
1	0	0
0	1	1

Feature
Detector



0	1	0	0	0
0	1	1	1	0
1	0	1	2	1
1				

Feature Map

Step 1 - Convolution

0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	0	0	1	0	0
0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0



Input
Image

0	0	1
1	0	0
0	1	1

Feature
Detector



0	1	0	0	0
0	1	1	1	0
1	0	1	2	1
1	4			

Feature Map

Step 1 - Convolution

0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	0	0	1	0	0
0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0



Input
Image

0	0	1
1	0	0
0	1	1

Feature
Detector



0	1	0	0	0
0	1	1	1	0
1	0	1	2	1
1	4	2		

Feature Map

Step 1 - Convolution

0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	0	0	1	0	0
0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0



Input
Image

0	0	1
1	0	0
0	1	1

Feature
Detector



0	1	0	0	0
0	1	1	1	0
1	0	1	2	1
1	4	2	1	

Feature Map

Step 1 - Convolution

0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	0	0	1	0	0
0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0



Input
Image

0	0	1
1	0	0
0	1	1

Feature
Detector



0	1	0	0	0
0	1	1	1	0
1	0	1	2	1
1	4	2	1	0

Feature Map

Step 1 - Convolution

0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	0	0	1	0	0
0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0



Input
Image

0	0	1
1	0	0
0	1	1

Feature
Detector



0	1	0	0	0
0	1	1	1	0
1	0	1	2	1
1	4	2	1	0
0				

Feature Map

Step 1 - Convolution

0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	0	0	1	0	0
0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0



Input
Image

0	0	1
1	0	0
0	1	1

Feature
Detector



0	1	0	0	0
0	1	1	1	0
1	0	1	2	1
1	4	2	1	0
0	0			

Feature Map

Step 1 - Convolution

0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	0	0	1	0	0
0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0



Input
Image

0	0	1
1	0	0
0	1	1

Feature
Detector



0	1	0	0	0
0	1	1	1	0
1	0	1	2	1
1	4	2	1	0
0	0	1		

Feature Map

Step 1 - Convolution

0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	0	0	1	0	0
0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0



Input
Image

0	0	1
1	0	0
0	1	1

Feature
Detector



0	1	0	0	0
0	1	1	1	0
1	0	1	2	1
1	4	2	1	0
0	0	1	2	

Feature Map

Step 1 - Convolution

0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	0	0	1	0	0
0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0



Input
Image

0	0	1
1	0	0
0	1	1

Feature
Detector



0	1	0	0	0
0	1	1	1	0
1	0	1	2	1
1	4	2	1	0
0	0	1	2	1

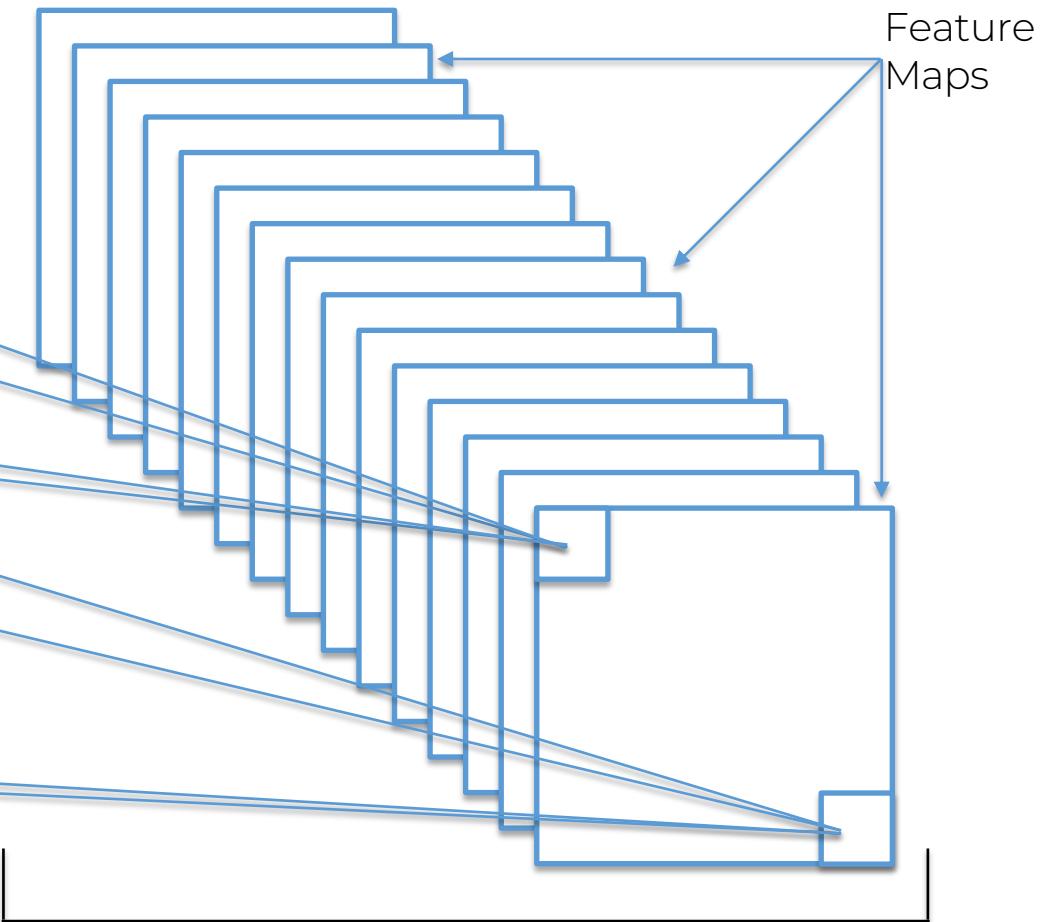
Feature Map

Step 1 - Convolution

0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	0	0	1	0	0
0	0	1	1	1	1	0	0
0	0	0	0	0	0	0	0

Input Image

We create many feature maps to obtain our first convolution layer



Convolutional Layer

Step 1 - Convolution

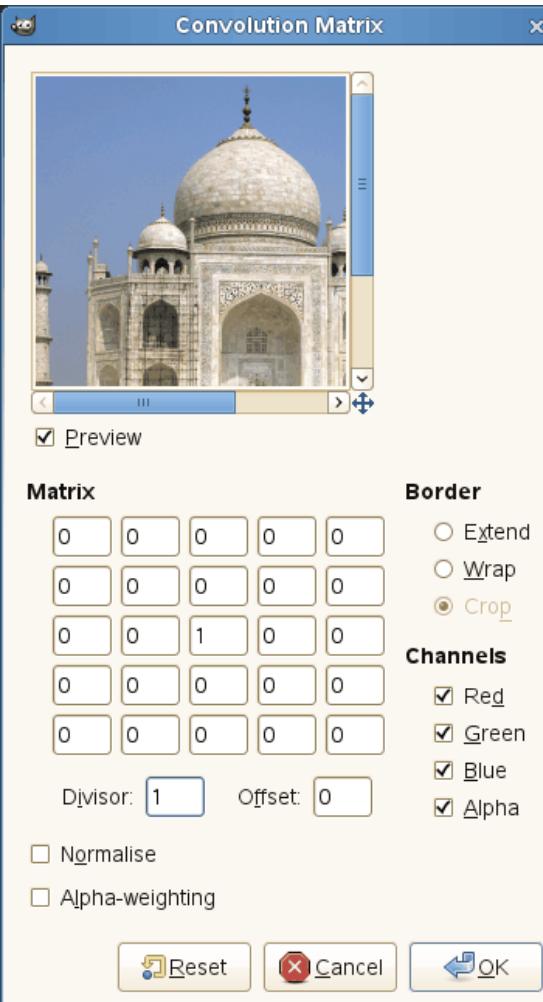


Image Source: docs.gimp.org/en/plug-in-convmatrix.html

Step 1 - Convolution

Sharpen:

0	0	0	0	0
0	0	-1	0	0
0	-1	5	-1	0
0	0	-1	0	0
0	0	0	0	0



Image Source: docs.gimp.org/en/plug-in-convmatrix.html

Step 1 - Convolution

Blur:

0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0



Image Source: docs.gimp.org/en/plug-in-convmatrix.html

Step 1 - Convolution

Edge Enhance:

$$\begin{matrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{matrix}$$

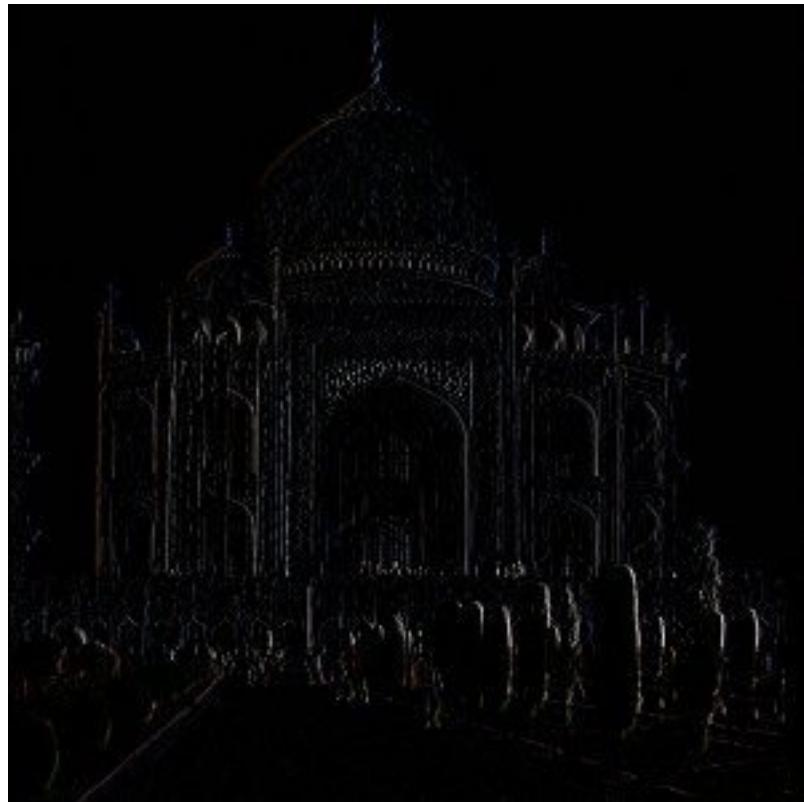


Image Source: docs.gimp.org/en/plug-in-convmatrix.html