Deep Learning

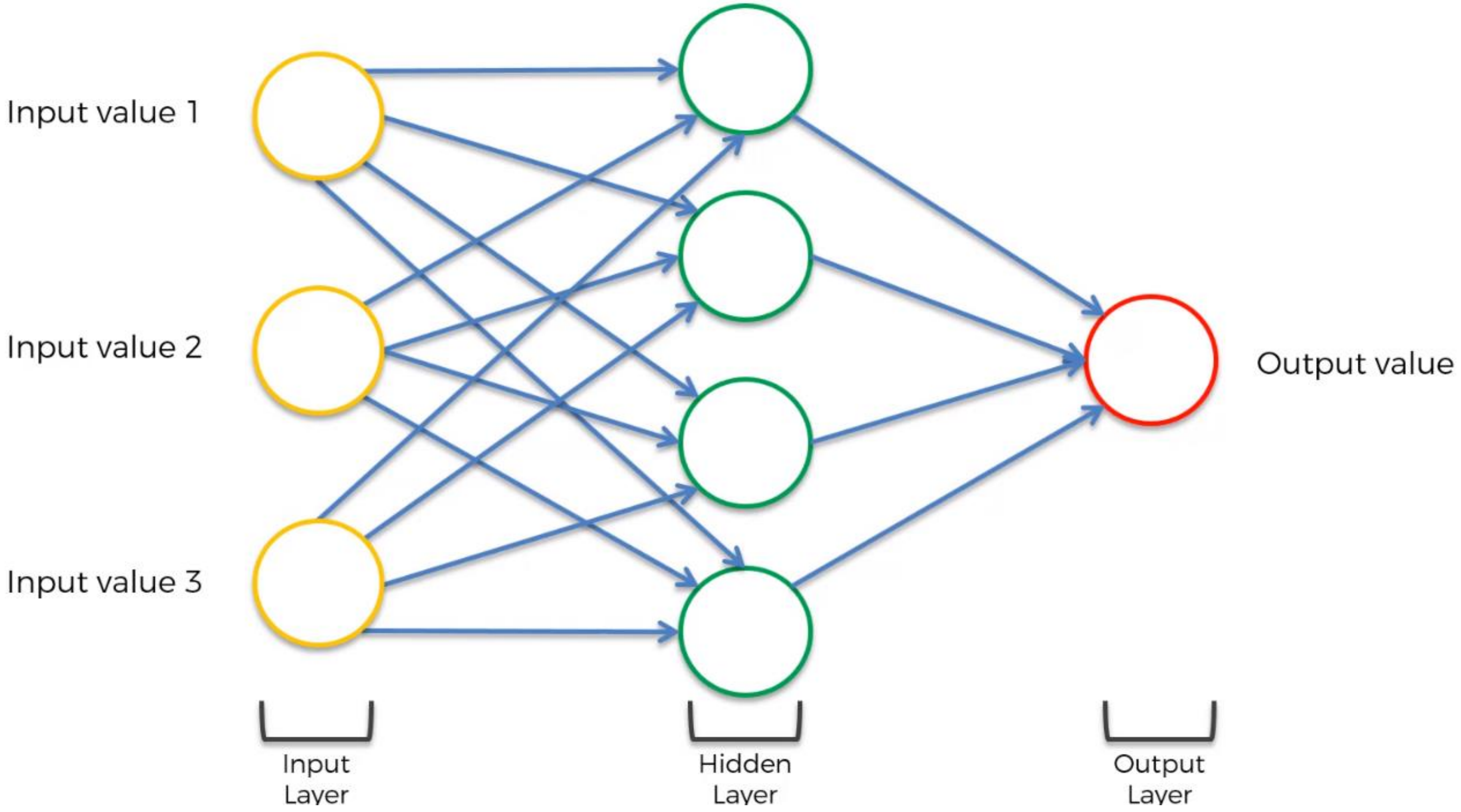
Summary

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

Summary

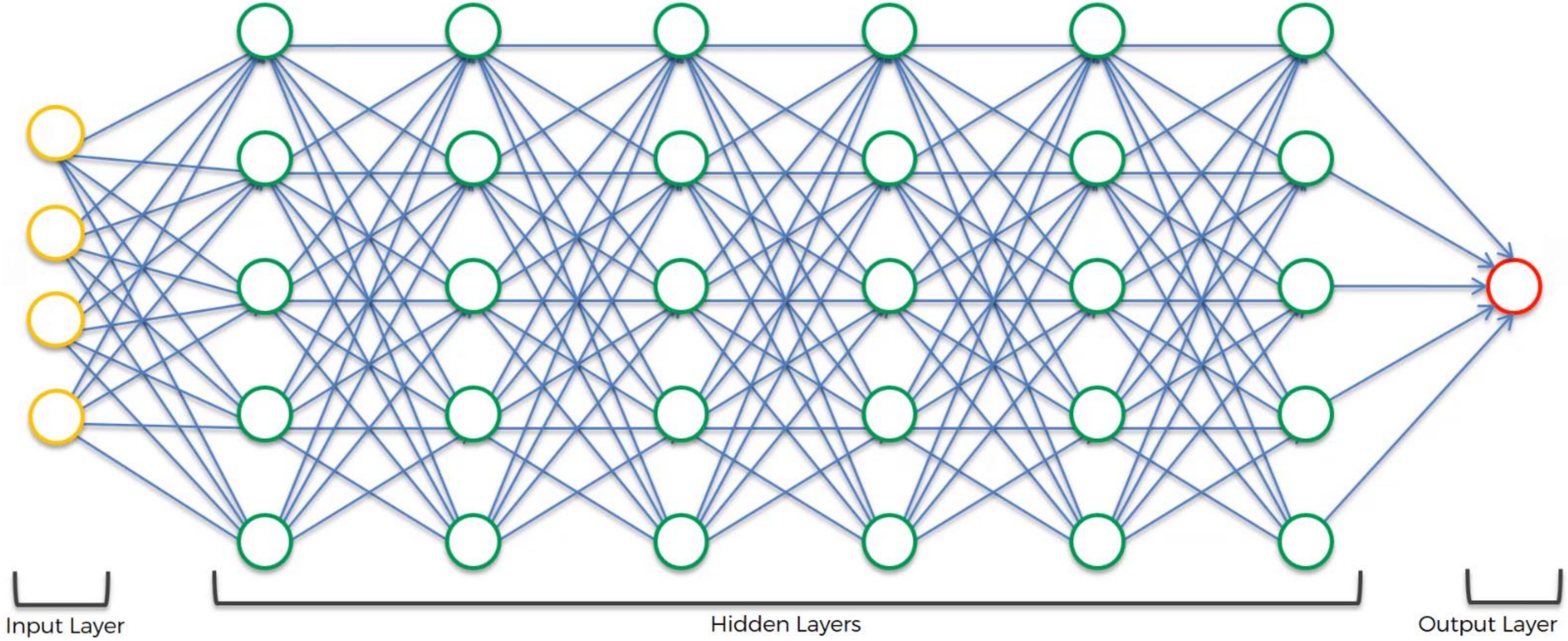
- Overview
- The Neuron
- The Activation Function
- How do Neural Networks work? (example)
- How do Neural Networks learn?
- Gradient Descent
- Stochastic Gradient Descent
- Backpropagation
- Algorithm summary

Overview



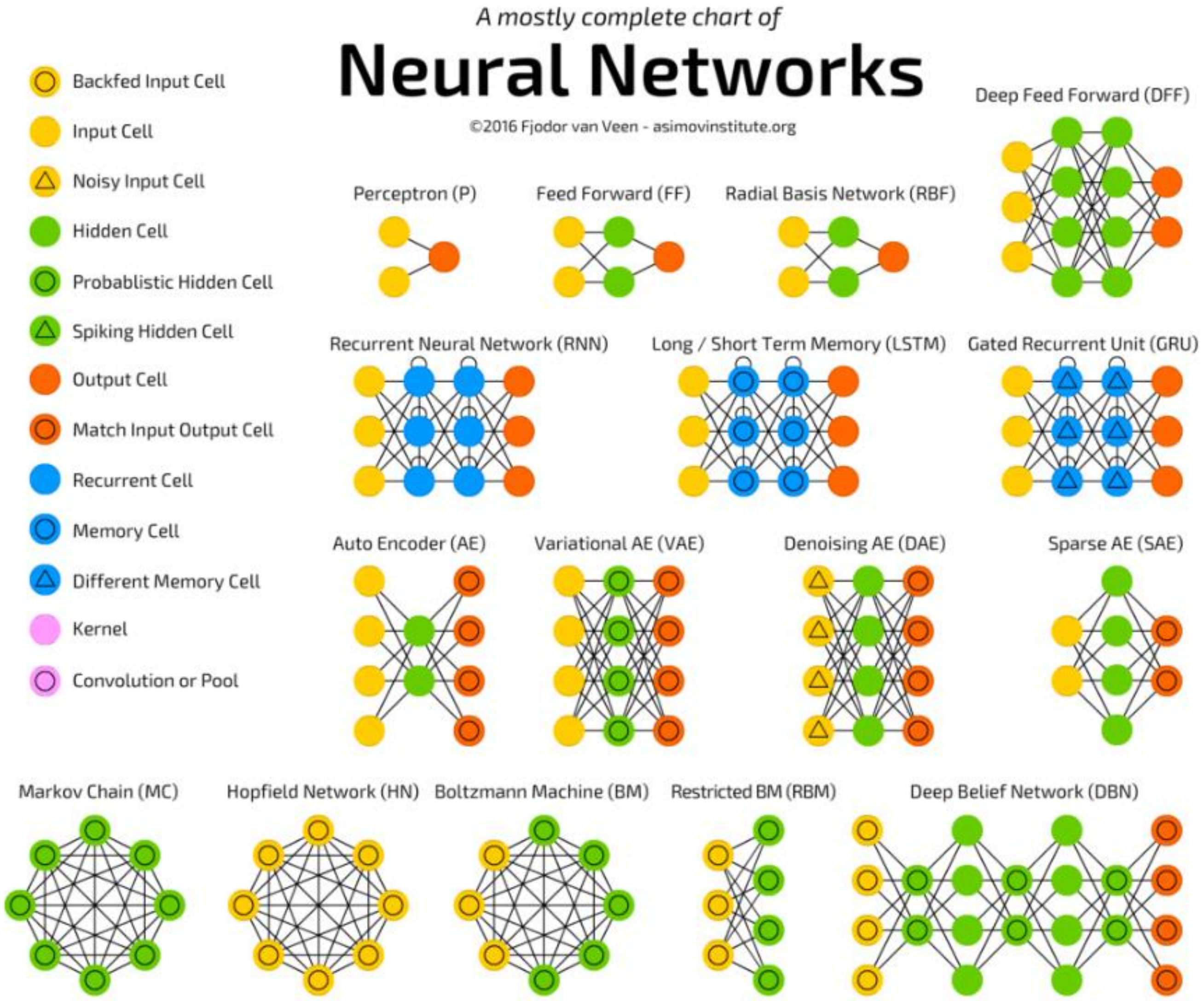
https://www.udemy.com/course/machinelearning/

Overview



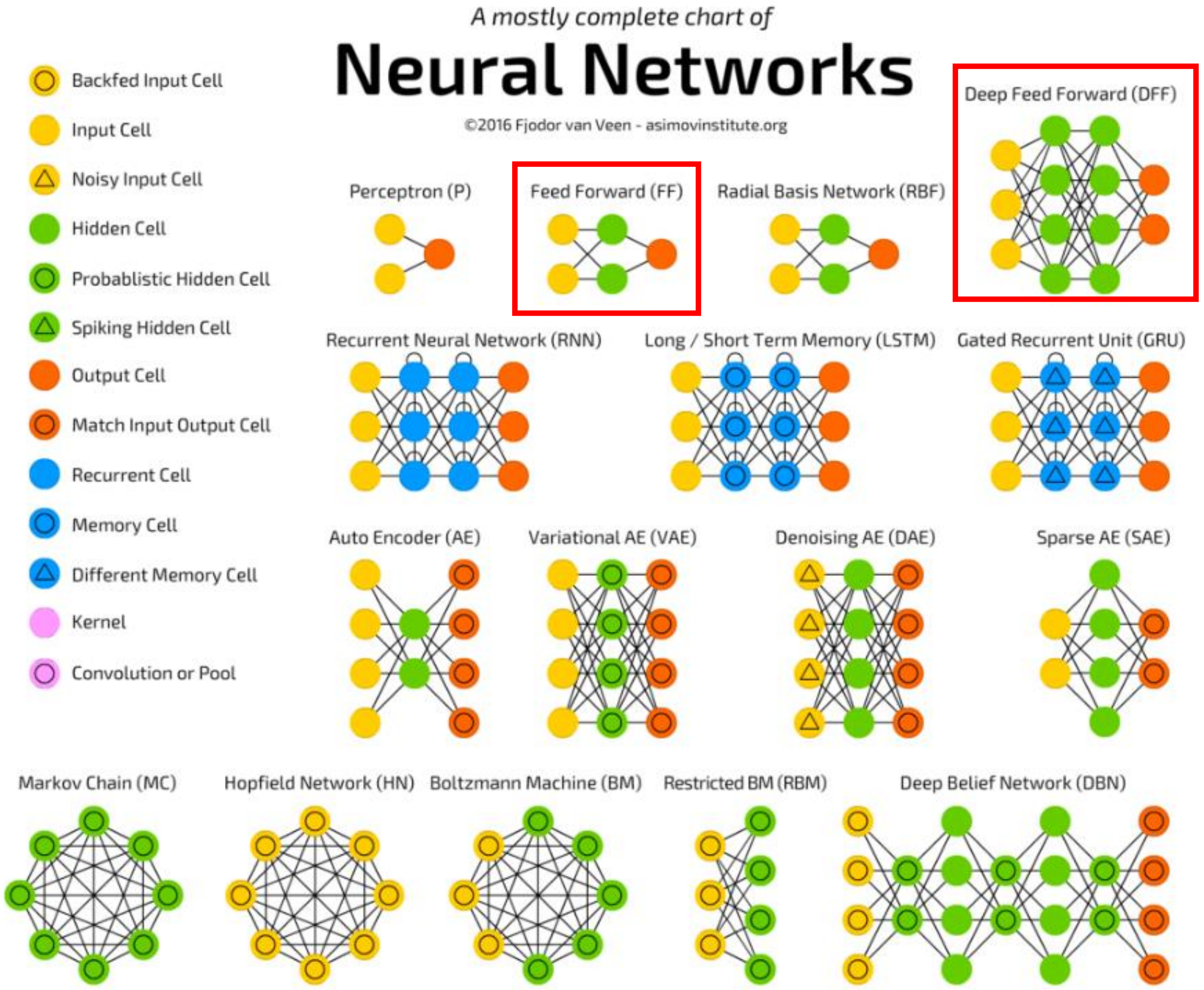
https://www.udemy.com/course/machinelearning/

Overview



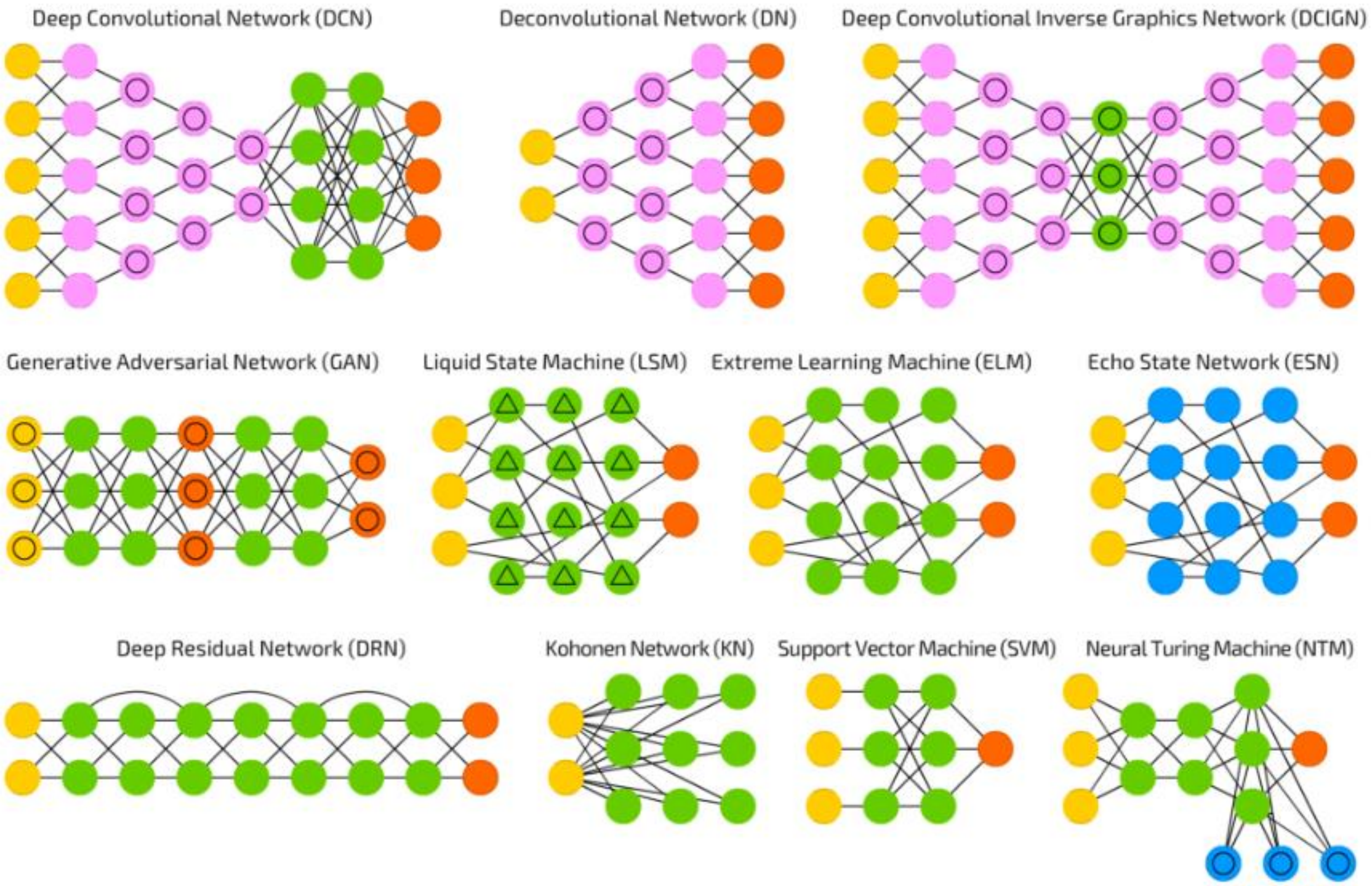
https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464

Overview



https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464

Overview


Deep Convolutional Network (DCN)
Deconvolutional Network (DN)
Deep Convolutional Inverse Graphics Network (DCIGN)
Generative Adversarial Network (GAN)
Liquid State Machine (LSM)
Extreme Learning Machine (ELM)
Echo State Network (ESN)
Deep Residual Network (DRN)
Kohonen Network (KN)
Support Vector Machine (SVM)
Neural Turing Machine (NTM)

https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464

Overview

- **Supervised Learning**: CNNs classifying images in imagenet.
- **Unsupervised Learning**: Boltzmann Machines, AutoEncoders, GANs, DC-GANS, VAE, SOMs, etc.
- **Reinforcement**: Deep Convolutional Q-Learning that plays videogames from pixel input, AlphaGO, etc.

Overview

- Artificial Neural Networks for Regression and Classification
- Convolutional Neural Networks for Computer Vision
- Recurrent Neural Networks for Time Series Analysis
- Self Organizing Maps for Feature Extraction
- Deep Boltzmann Machines for Recommendation Systems
- Auto Encoders for Recommendation Systems

https://www.udemy.com/course/machinelearning/

Overview

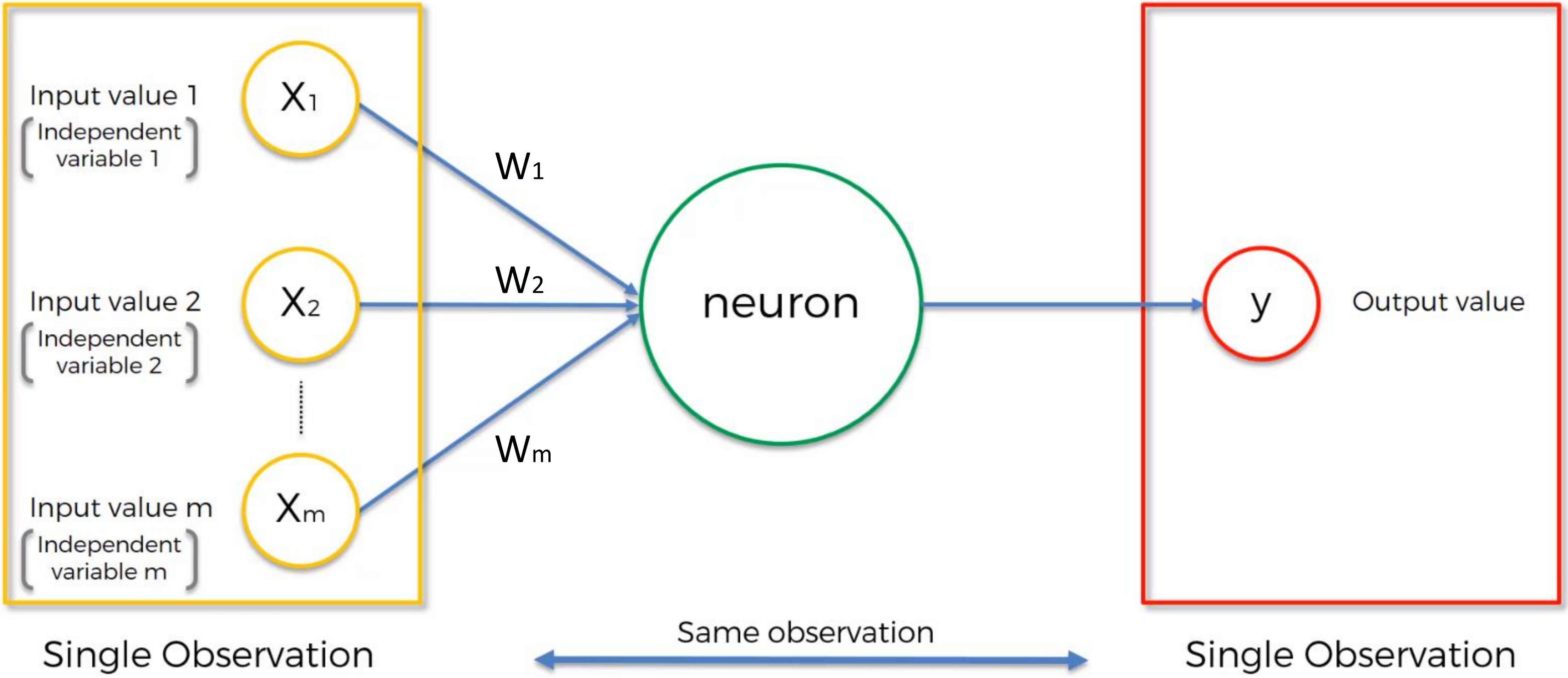
- **Artificial Neural Networks for Regression and Classification**
- **Convolutional Neural Networks for Computer Vision**
- Recurrent Neural Networks for Time Series Analysis
- Self Organizing Maps for Feature Extraction
- Deep Boltzmann Machines for Recommendation Systems
- Auto Encoders for Recommendation Systems

https://www.udemy.com/course/machinelearning/

Summary

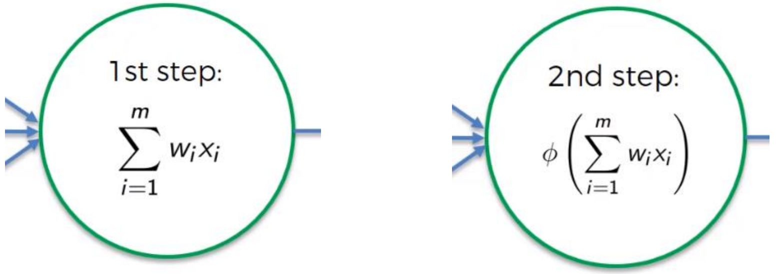
- Overview
- The Neuron
- The Activation Function
- How do Neural Networks work? (example)
- How do Neural Networks learn?
- Gradient Descent
- Stochastic Gradient Descent
- Backpropagation

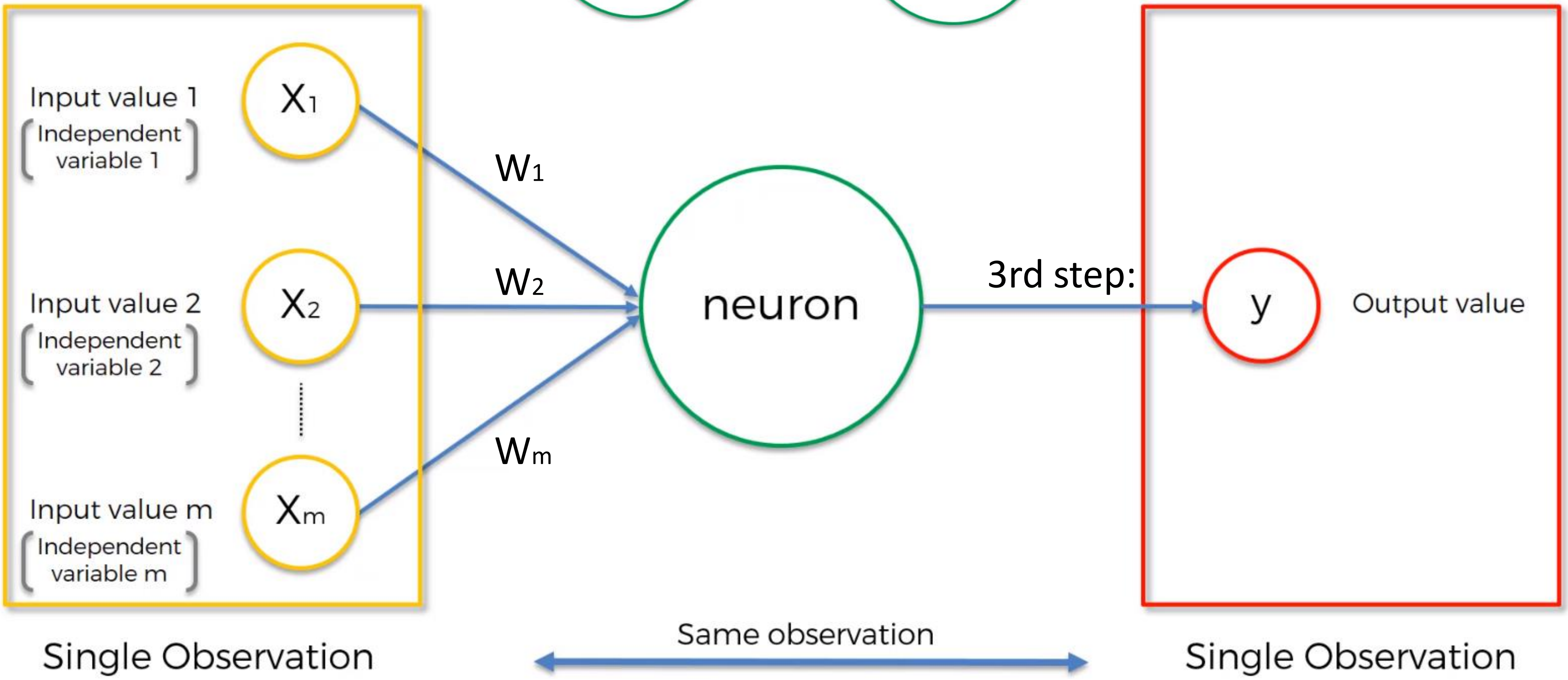
The Neuron



https://www.udemy.com/course/machinelearning/

The Neuron





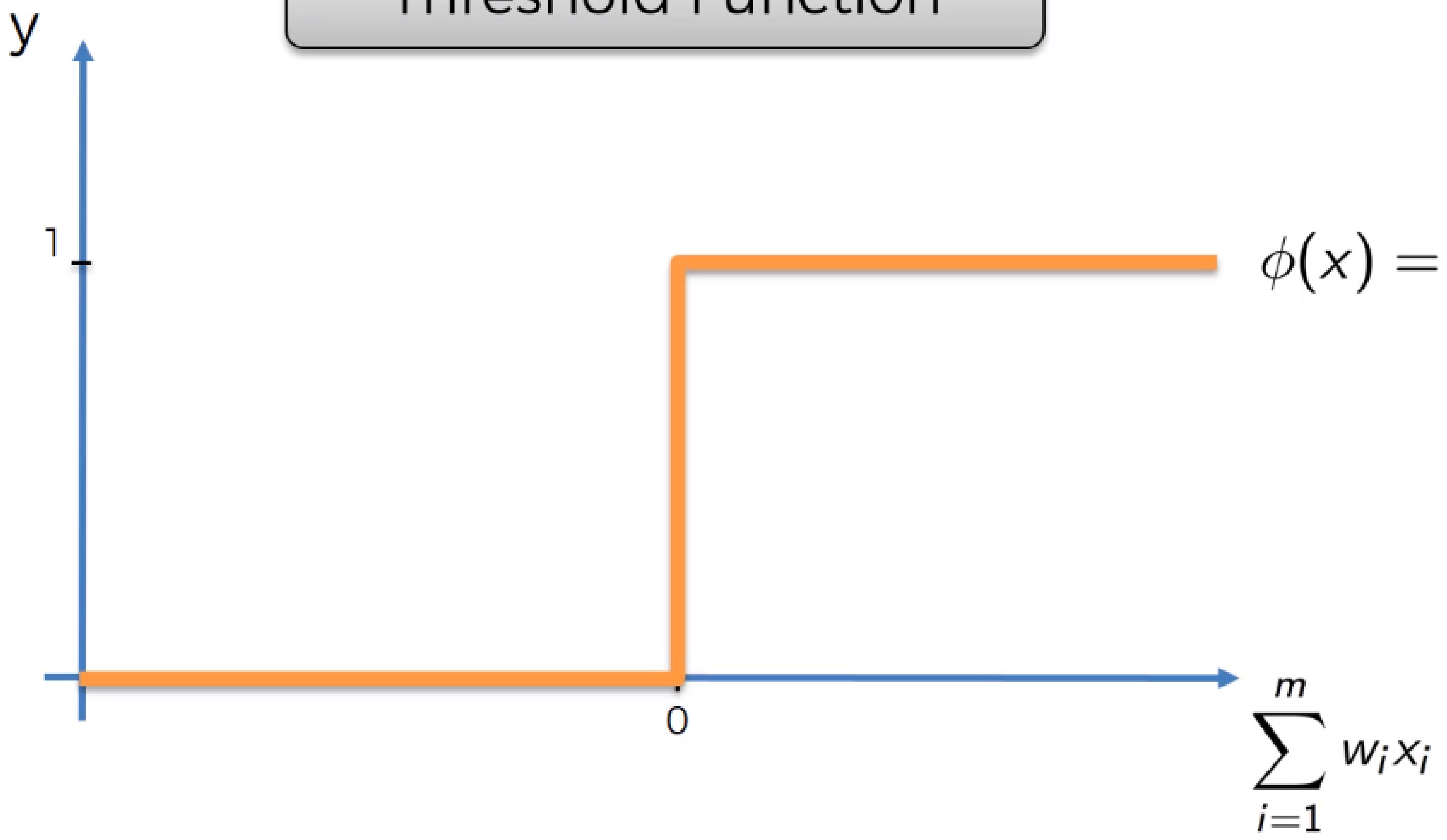
https://www.udemy.com/course/machinelearning/

Summary

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

The Activation Function

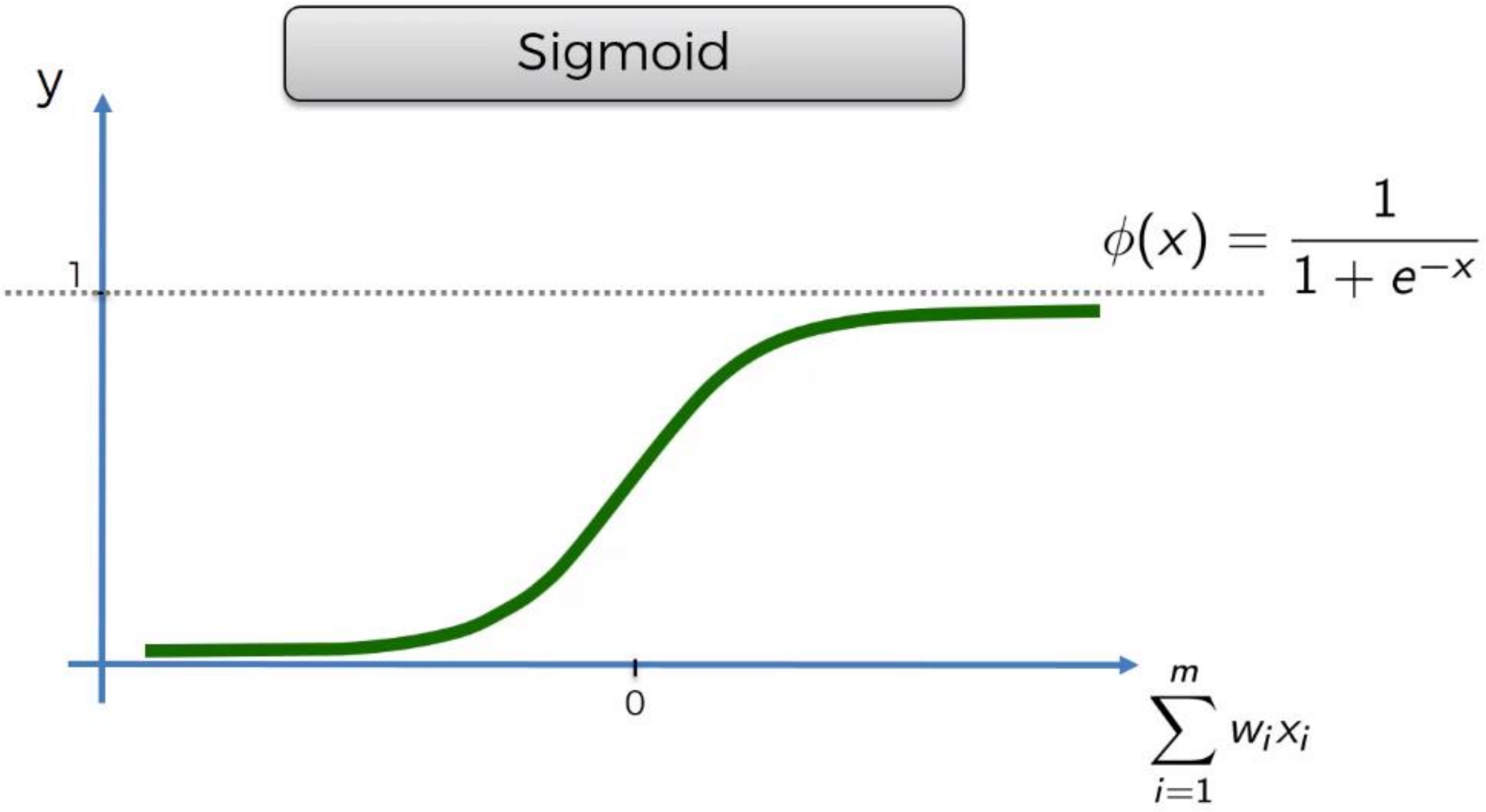
Threshold Function



$$\phi(x) = \begin{cases} 1 \text{ if } x \geq 0 \\ 0 \text{ if } x < 0 \end{cases}$$

https://www.udemy.com/course/machinelearning/

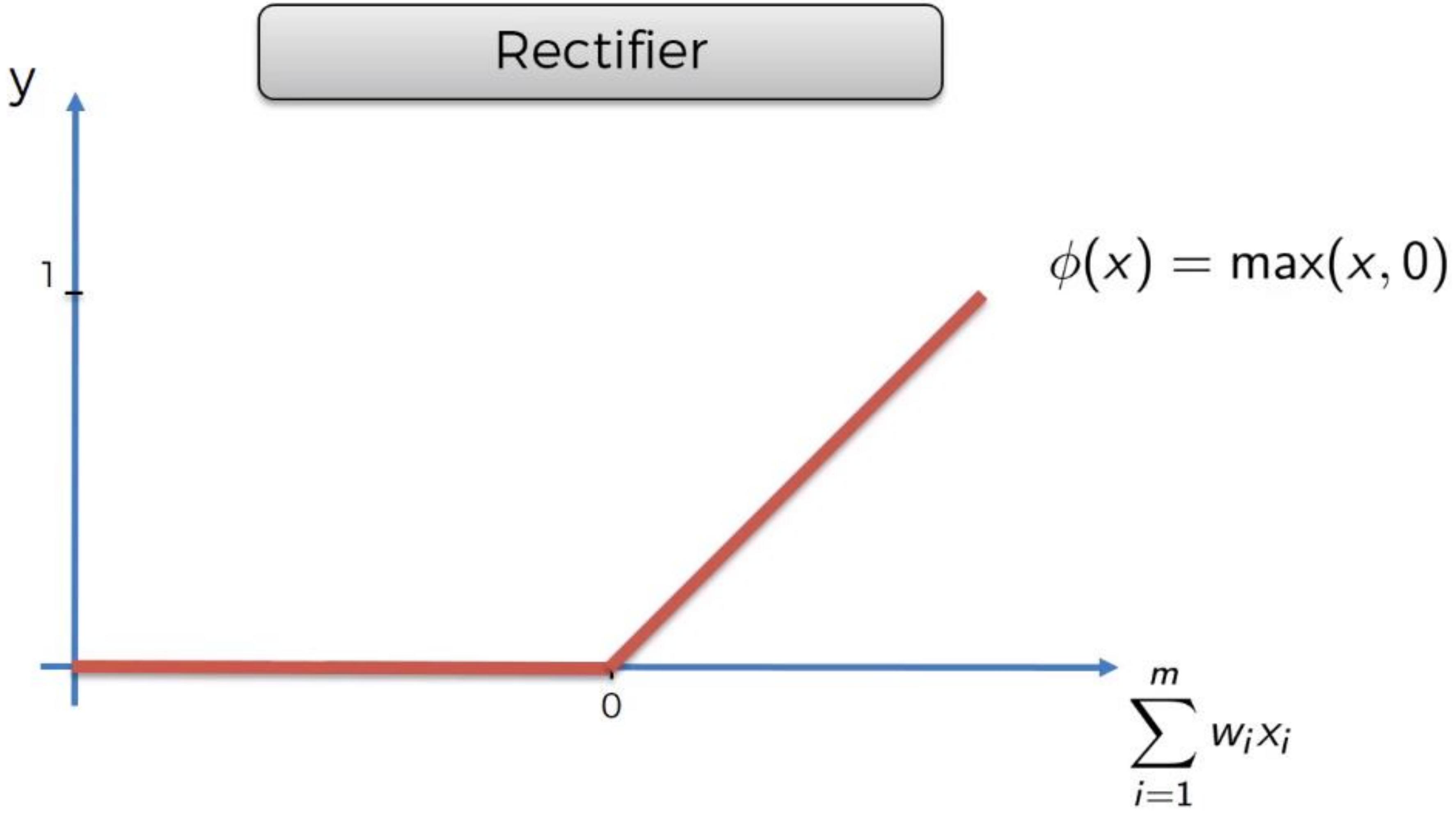
The Activation Function



$$\phi(x) = \frac{1}{1+e^{-x}}$$

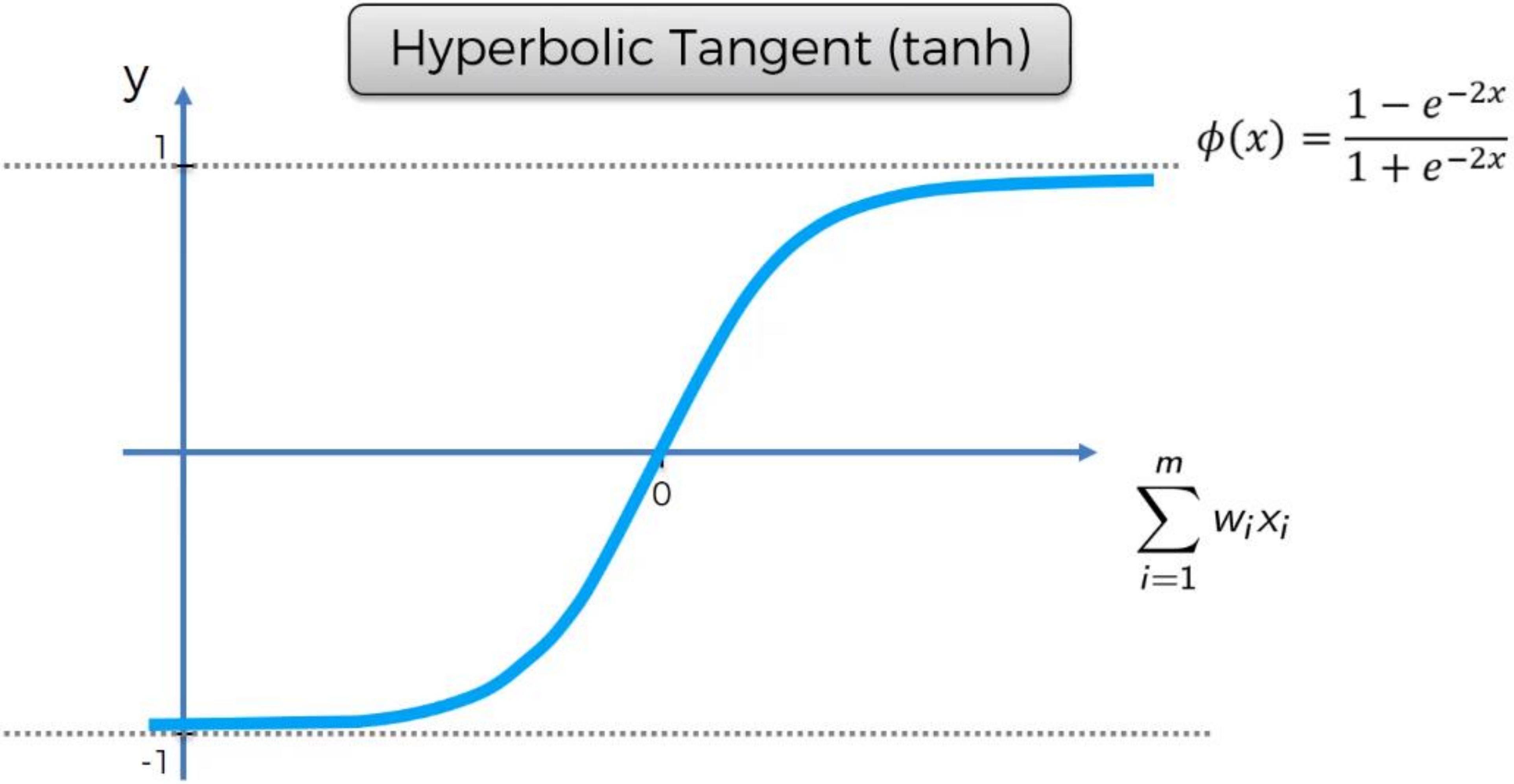
https://www.udemy.com/course/machinelearning/

The Activation Function



https://www.udemy.com/course/machinelearning/

The Activation Function

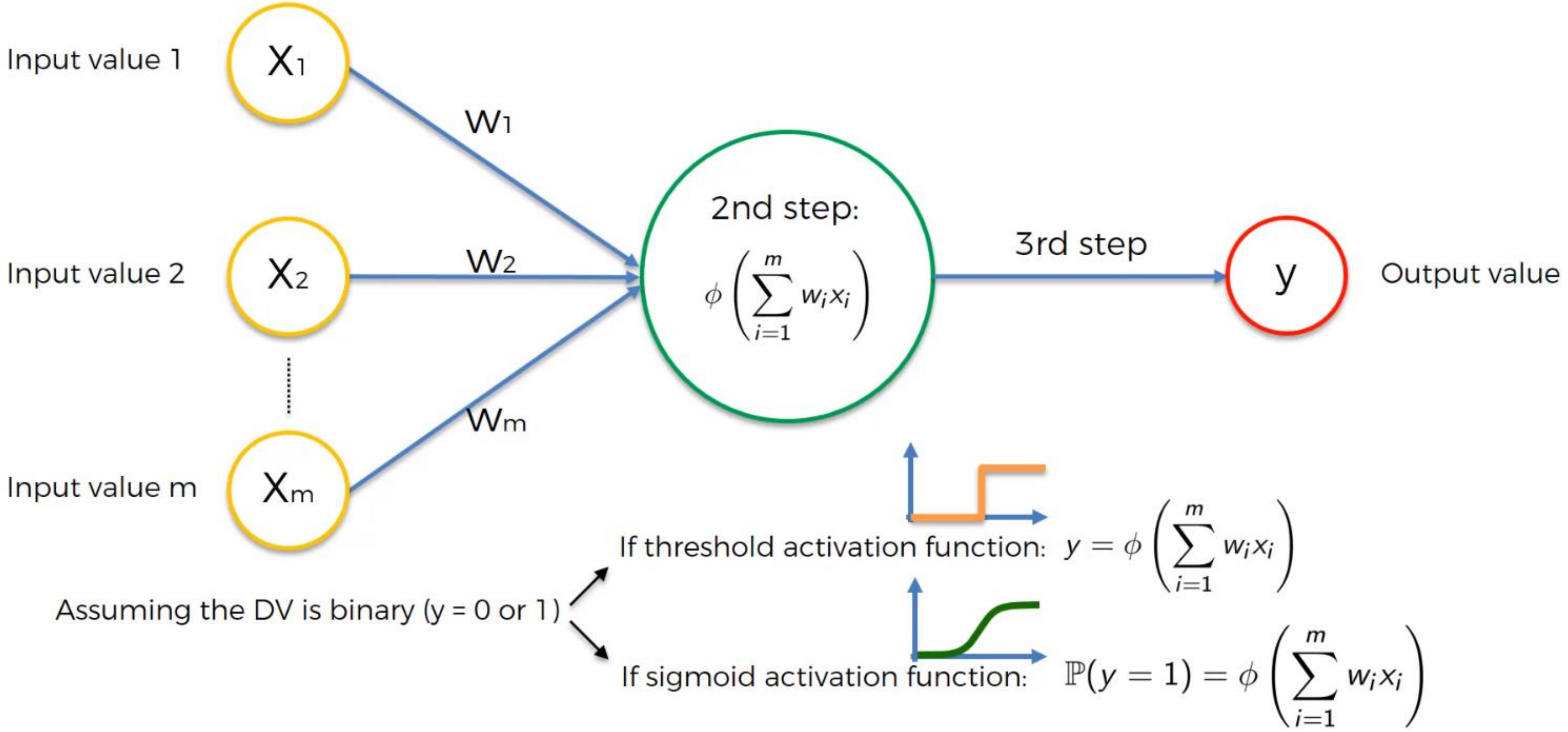


$$\phi(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

$$\sum_{i=1}^{m} w_i x_i$$

https://www.udemy.com/course/machinelearning/

The Activation Function



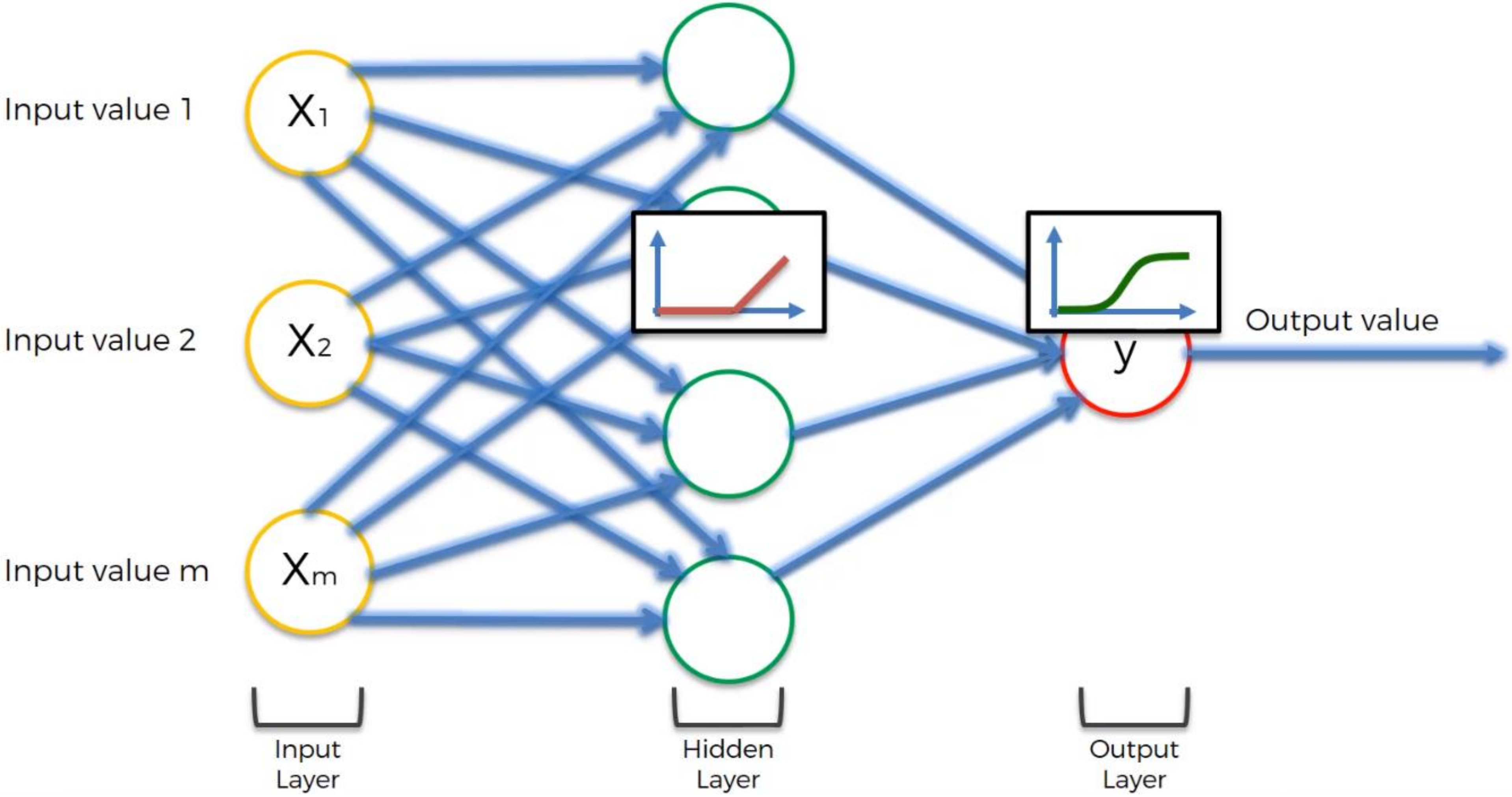
Assuming the DV is binary (y = 0 or 1)

If threshold activation function: $y = \phi\left(\sum_{i=1}^{m} w_i x_i\right)$

If sigmoid activation function: $\mathbb{P}(y = 1) = \phi\left(\sum_{i=1}^{m} w_i x_i\right)$

https://www.udemy.com/course/machinelearning/

The Activation Function

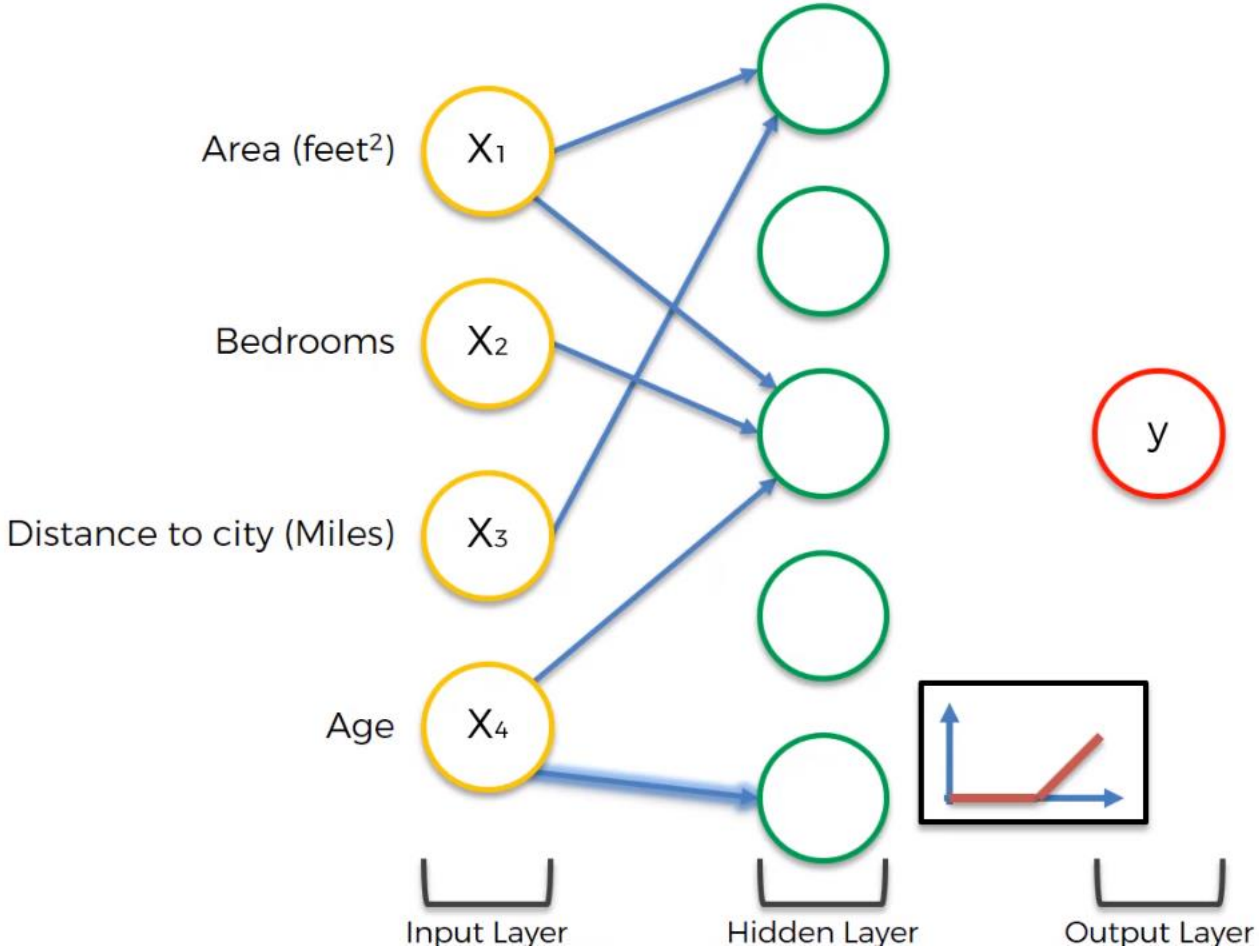


https://www.udemy.com/course/machinelearning/

Summary

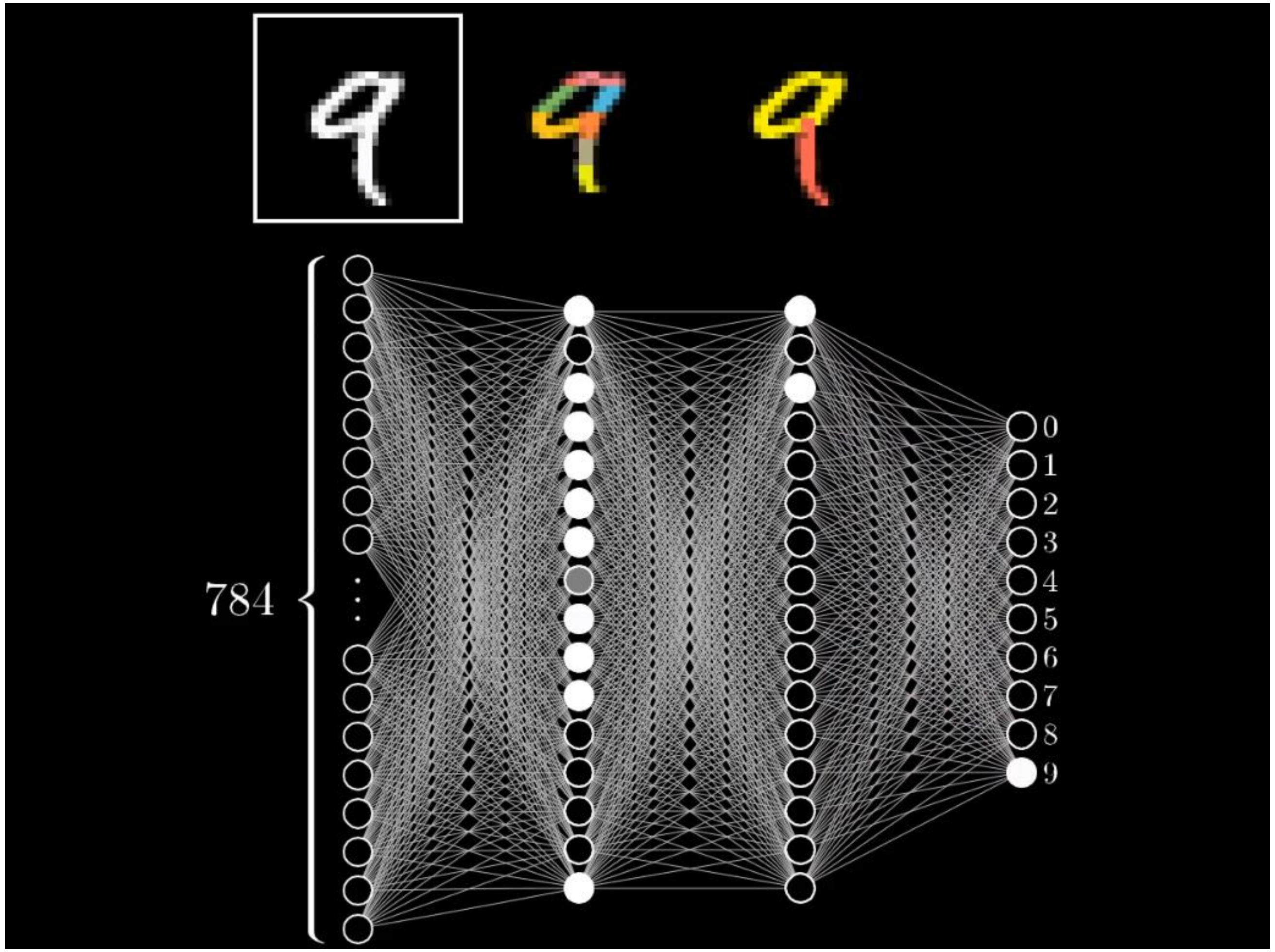
- Overview
- The Neuron
- The Activation Function
- How do Neural Networks work? (example)
- How do Neural Networks learn?
- Gradient Descent
- Stochastic Gradient Descent
- Backpropagation
- Algorithm summary

How do Neural Networks work?



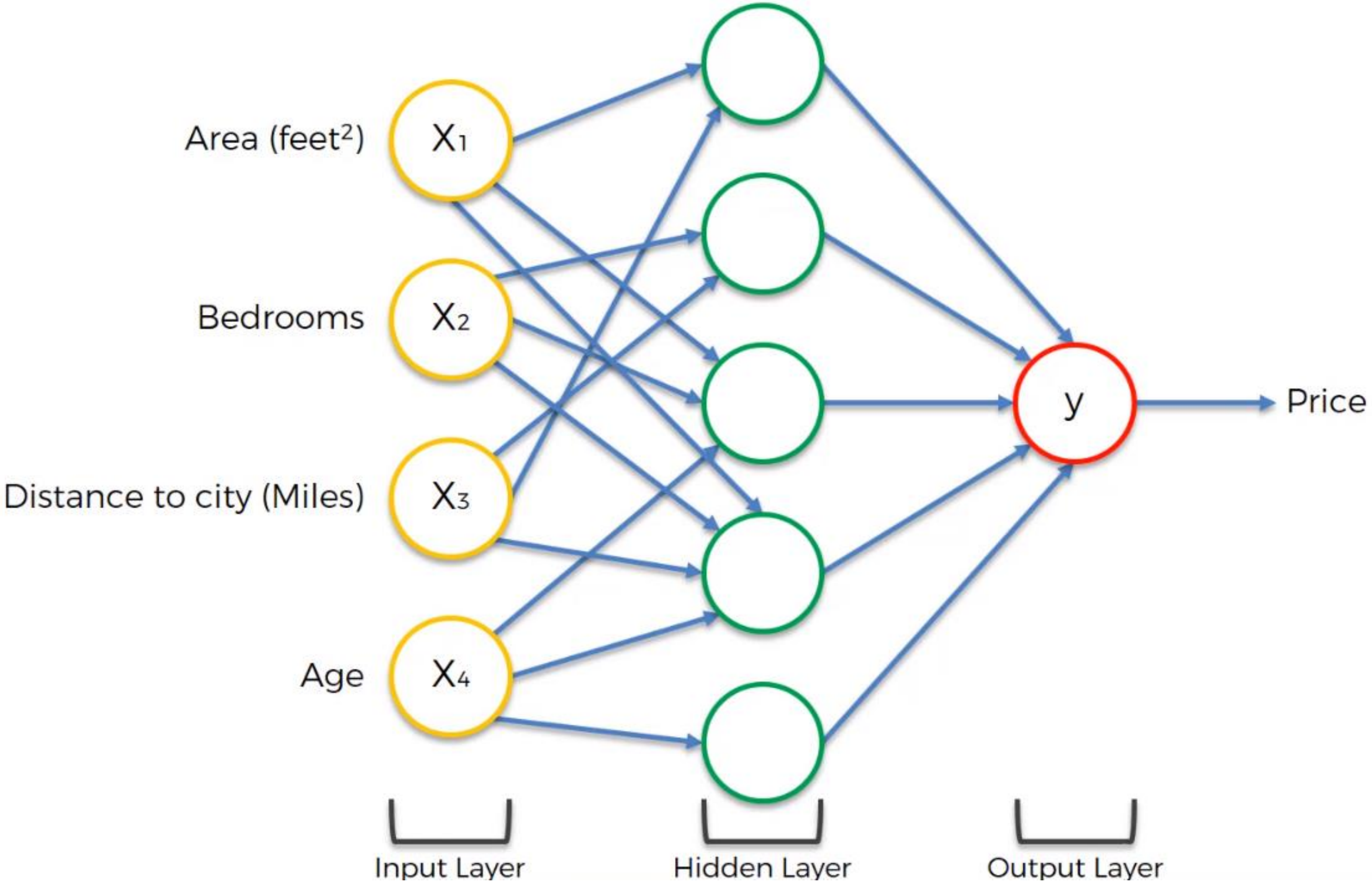
https://www.udemy.com/course/machinelearning/

How do Neural Networks work?



https://youtu.be/aircAruvnKk

How do Neural Networks work?

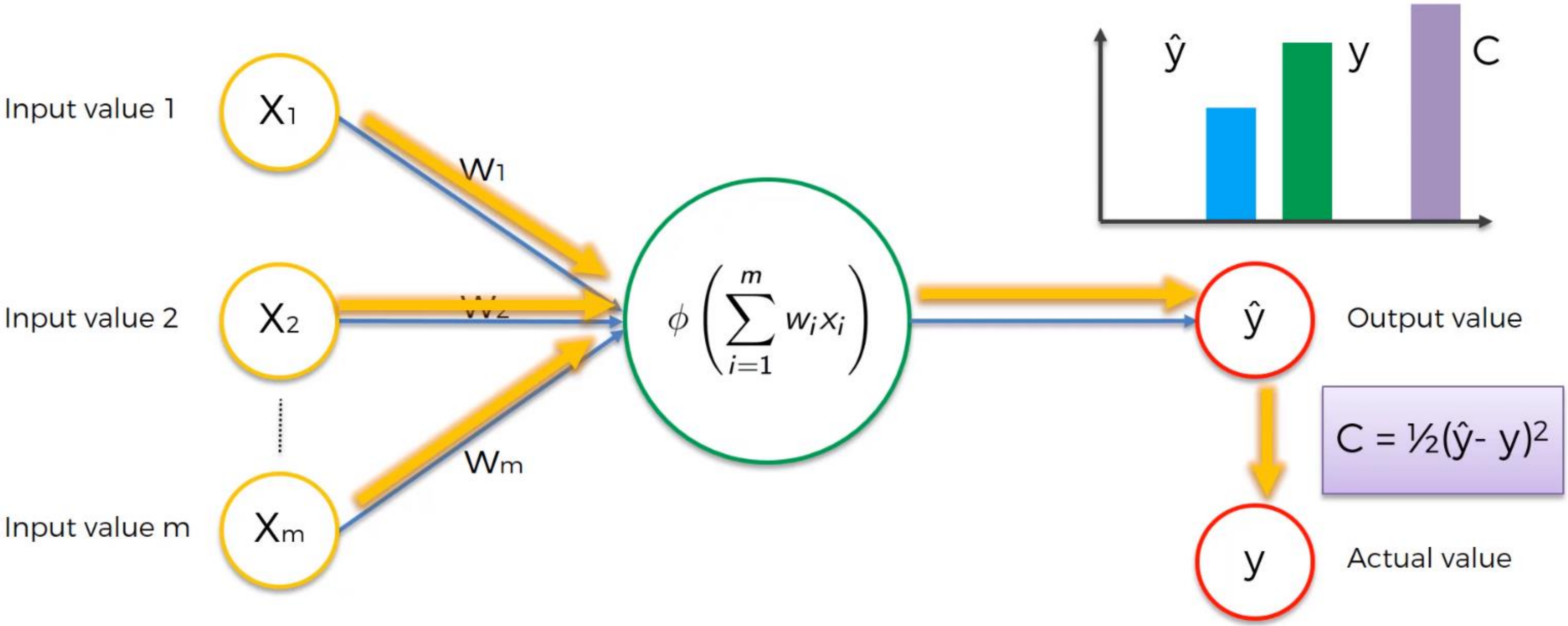


https://www.udemy.com/course/machinelearning/

Summary

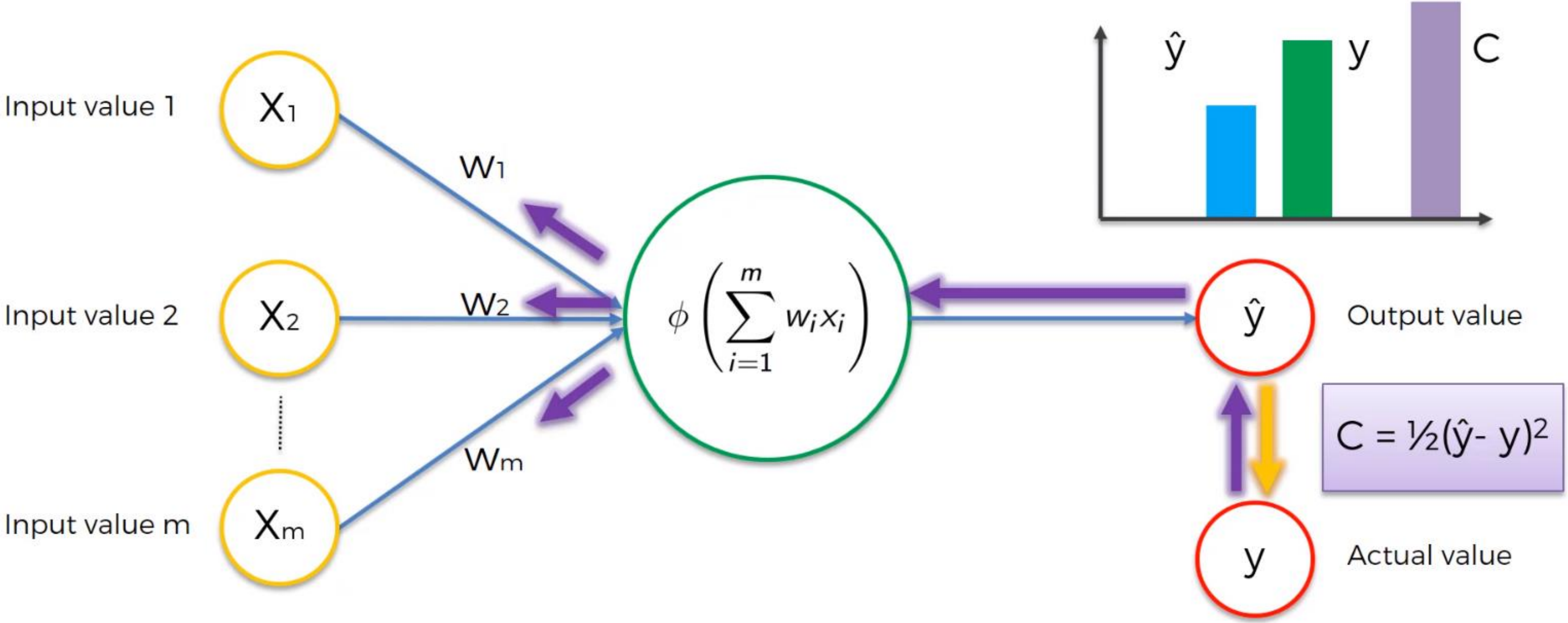
- Overview
- The Neuron
- The Activation Function
- How do Neural Networks work? (example)
- How do Neural Networks learn?
- Gradient Descent
- Stochastic Gradient Descent
- Backpropagation
- Algorithm summary

How do Neural Networks learn?



https://www.udemy.com/course/machinelearning/

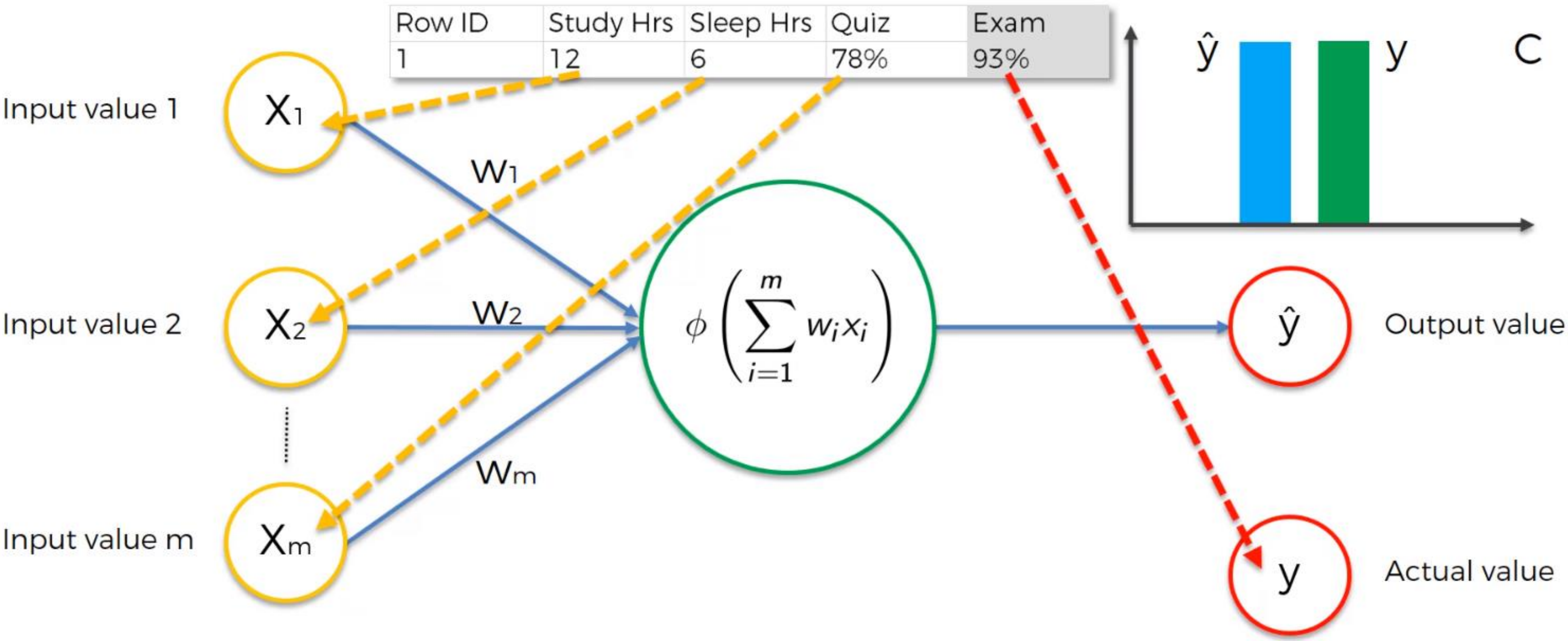
How do Neural Networks learn?



$C = ½(ŷ - y)^2$

https://www.udemy.com/course/machinelearning/

How do Neural Networks learn?

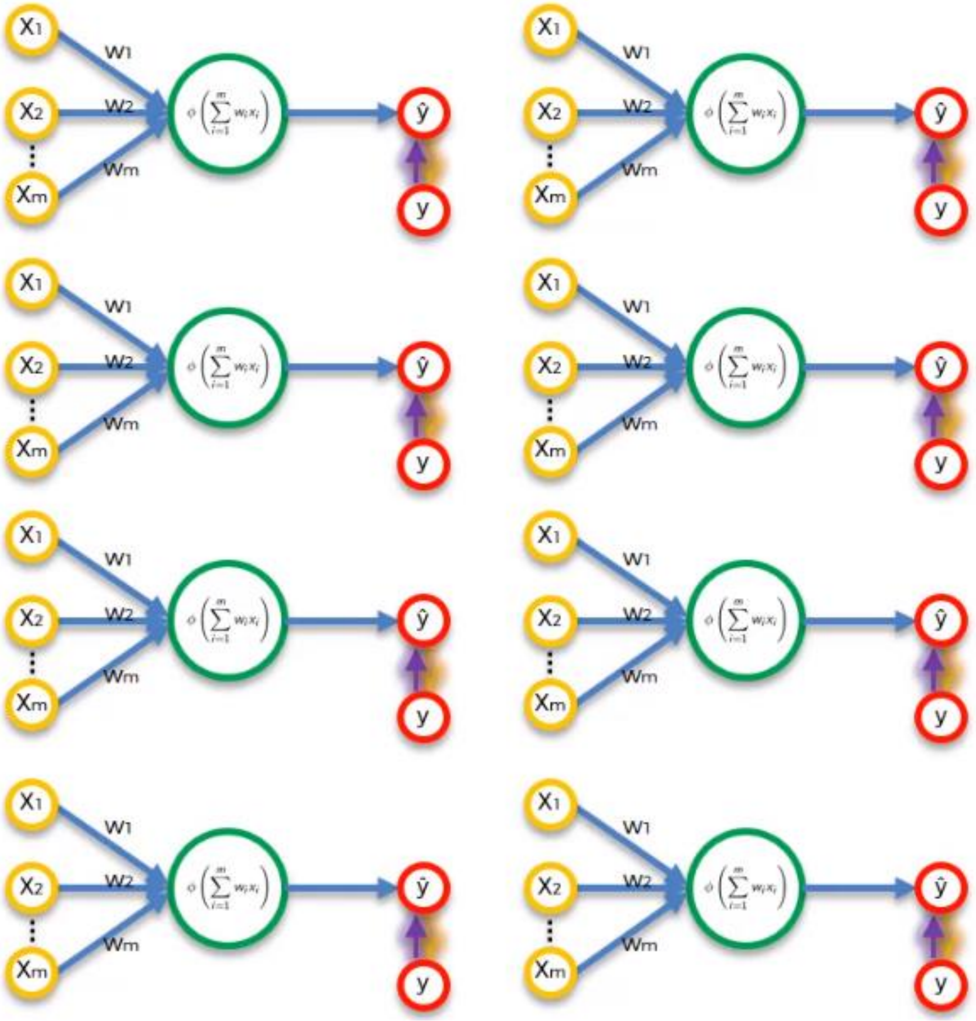


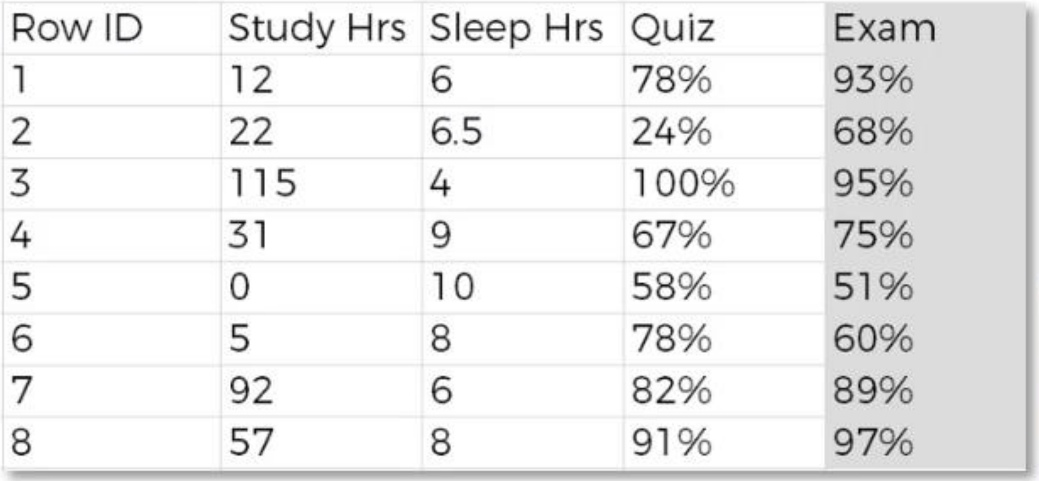
Row ID	Study Hrs	Sleep Hrs	Quiz	Exam
1	12	6	78%	93%

$$\phi\left(\sum_{i=1}^{m} w_i x_i\right)$$

https://www.udemy.com/course/machinelearning/

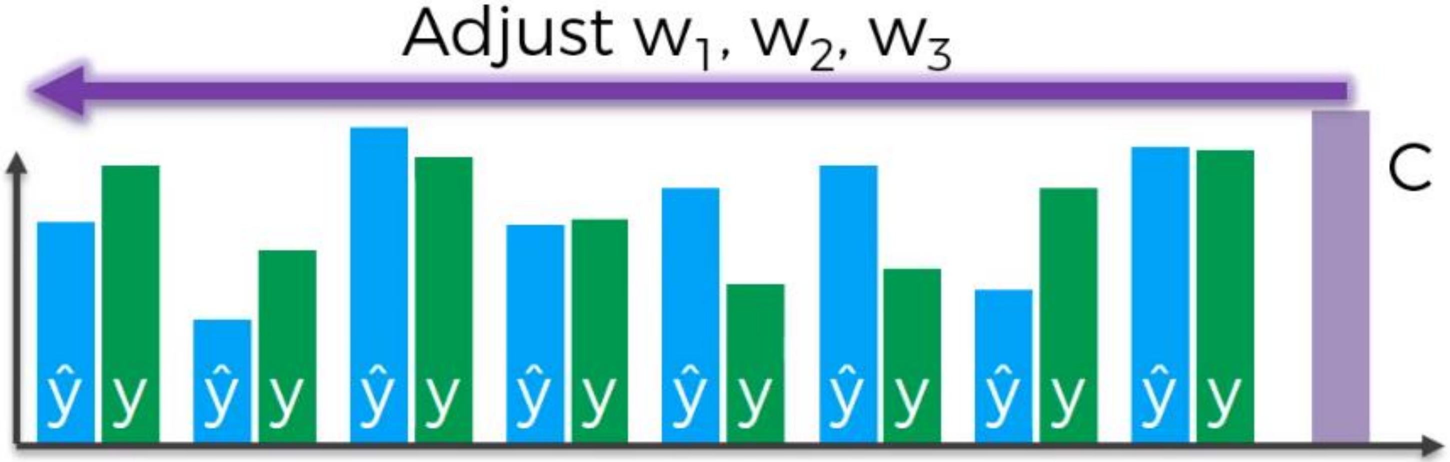
How do Neural Networks learn?





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%

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

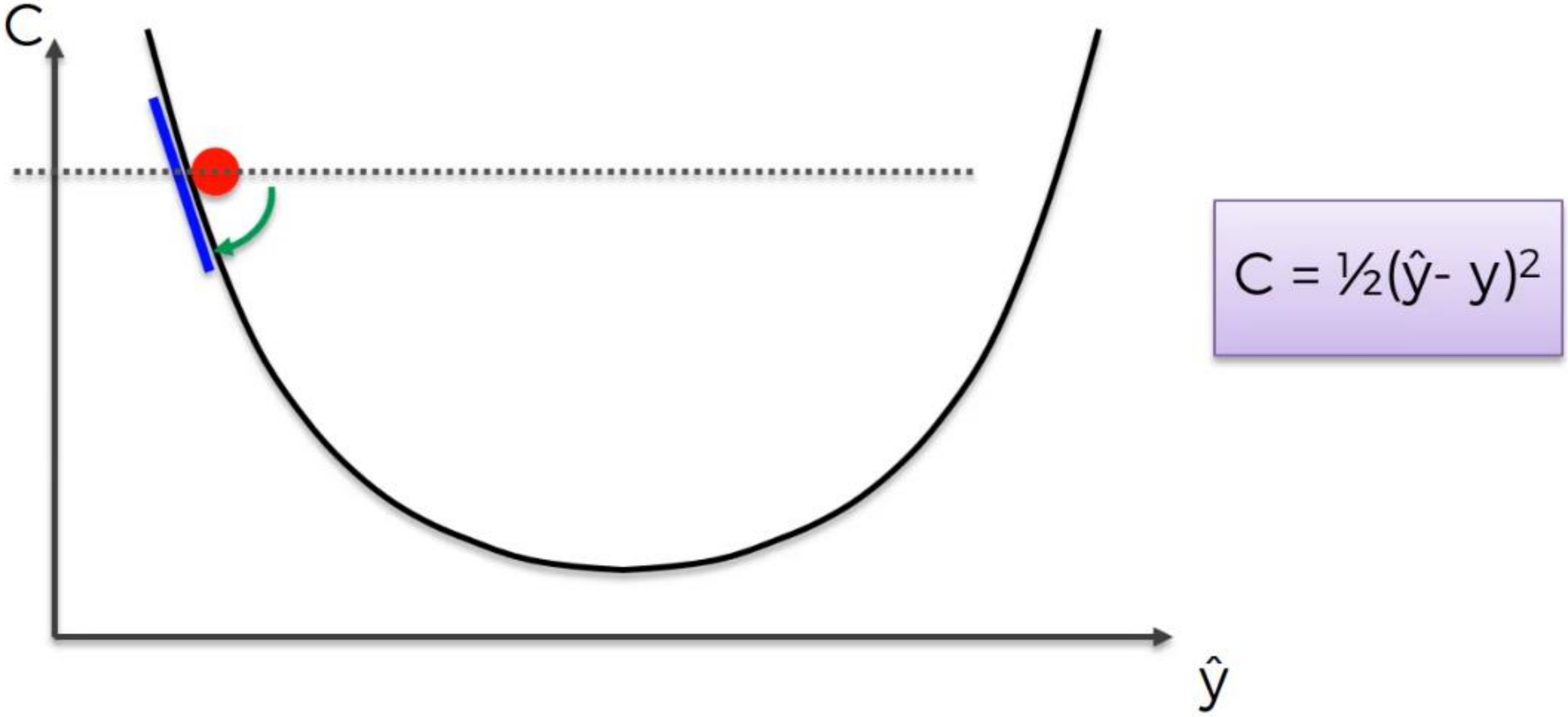


https://www.udemy.com/course/machinelearning/

Summary

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

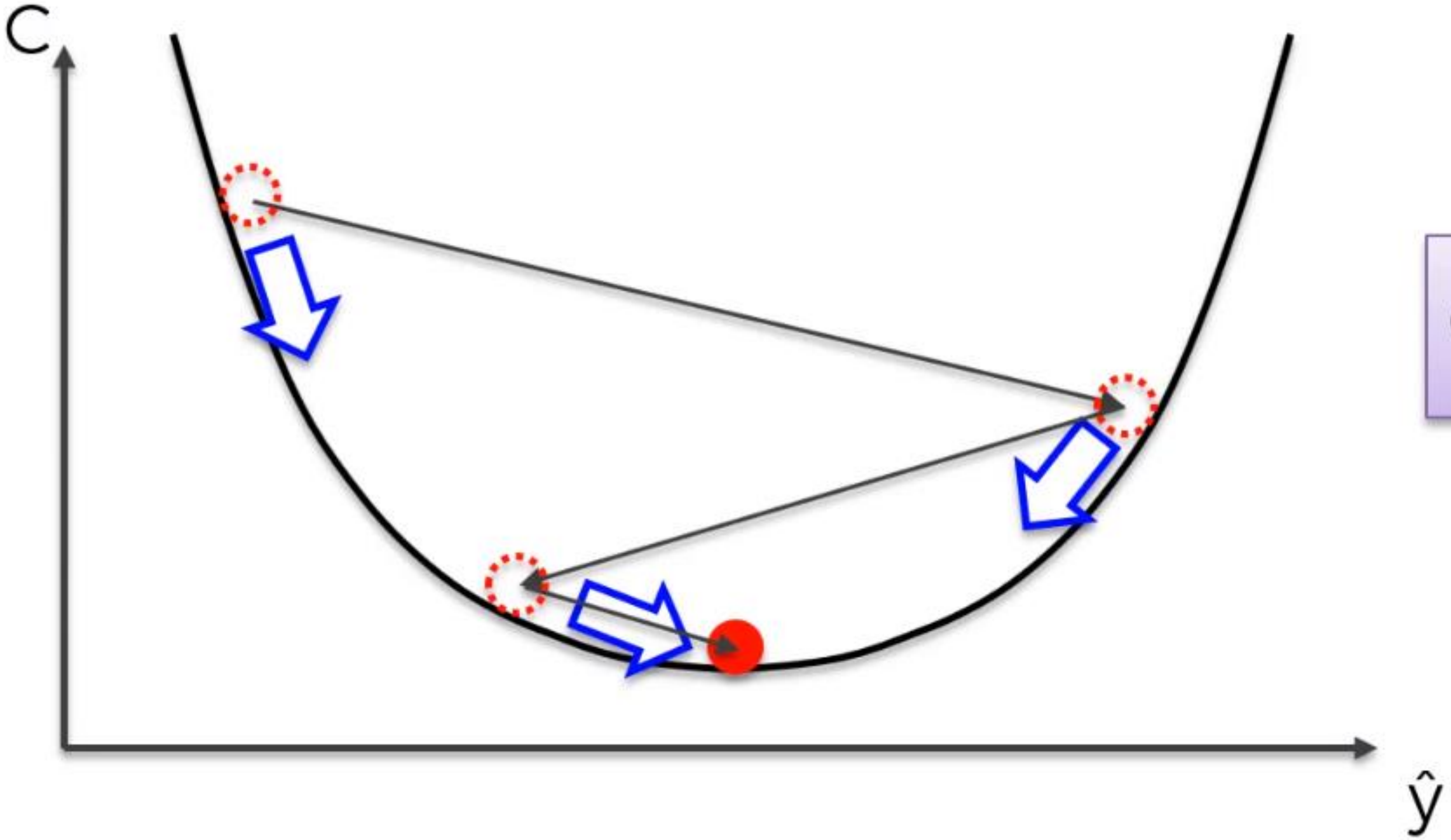
Gradient Descent



$C = ½(\hat{y} - y)^2$

https://www.udemy.com/course/machinelearning/

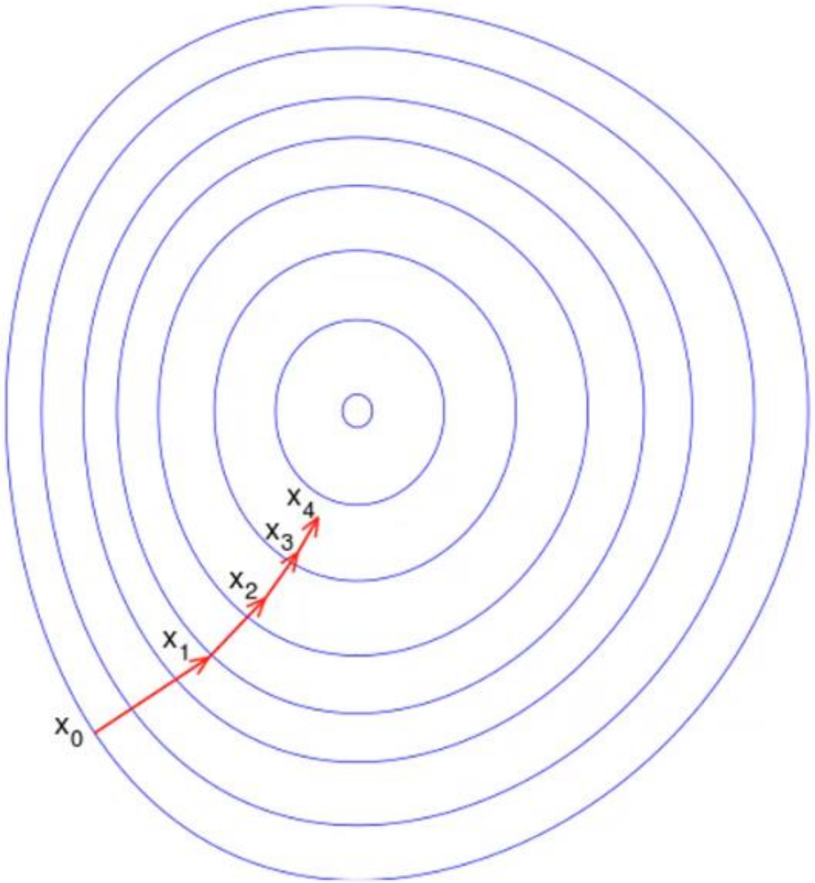
Gradient Descent



$C = ½(\hat{y}- y)^2$

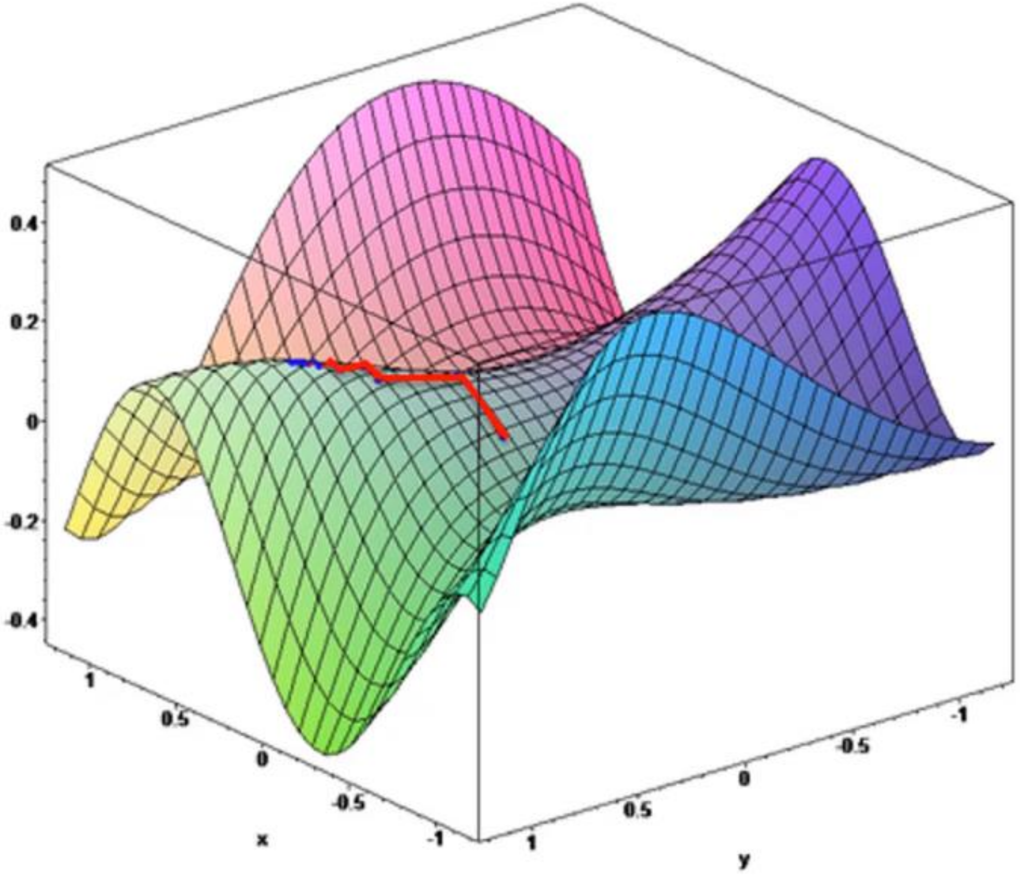
https://www.udemy.com/course/machinelearning/

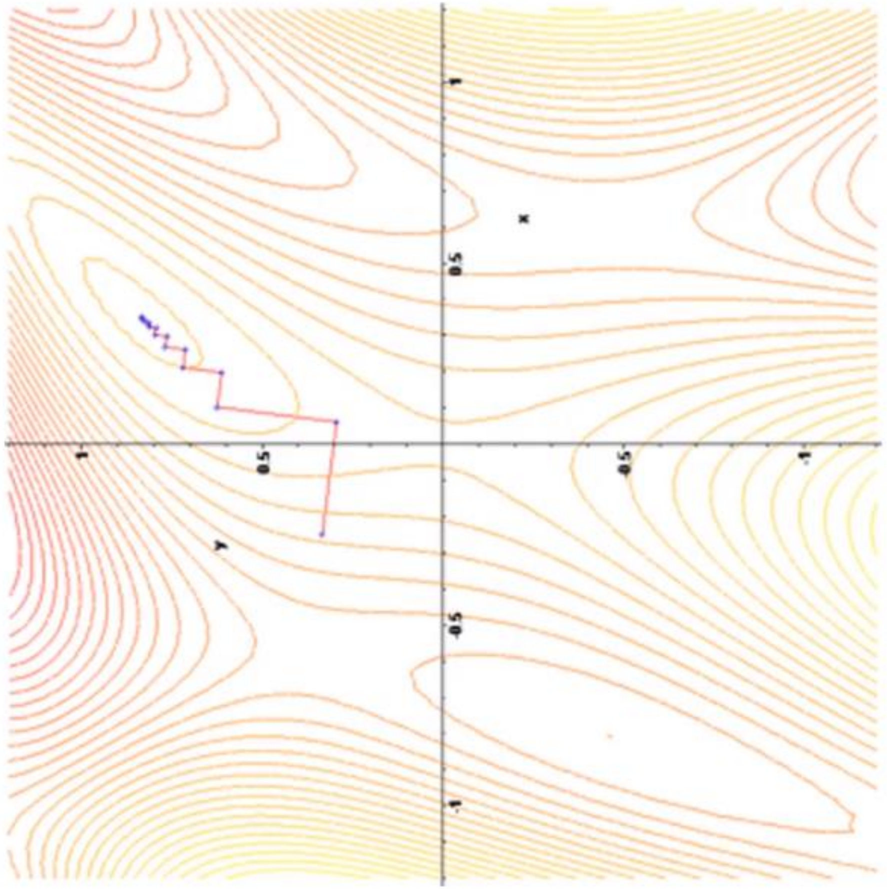
Gradient Descent



https://www.udemy.com/course/machinelearning/

Gradient Descent





https://www.udemy.com/course/machinelearning/

Gradient Descent

Step 1: Take the derivative of the **Cost Function** for each parameter in it. In fancy Machine Learning lingo, take the **Gradient** of the **Loss Function**.

Step 2: Pick random values for the parameters.

Step 3: Plug the parameter values into the derivatives (**Gradient**).

Step 4: Calculate the Step Sizes: Step Size = **Slope * Learning Rate**

Step 5: Calculate the New Parameters:

New Parameter = Old Parameter – Step Size

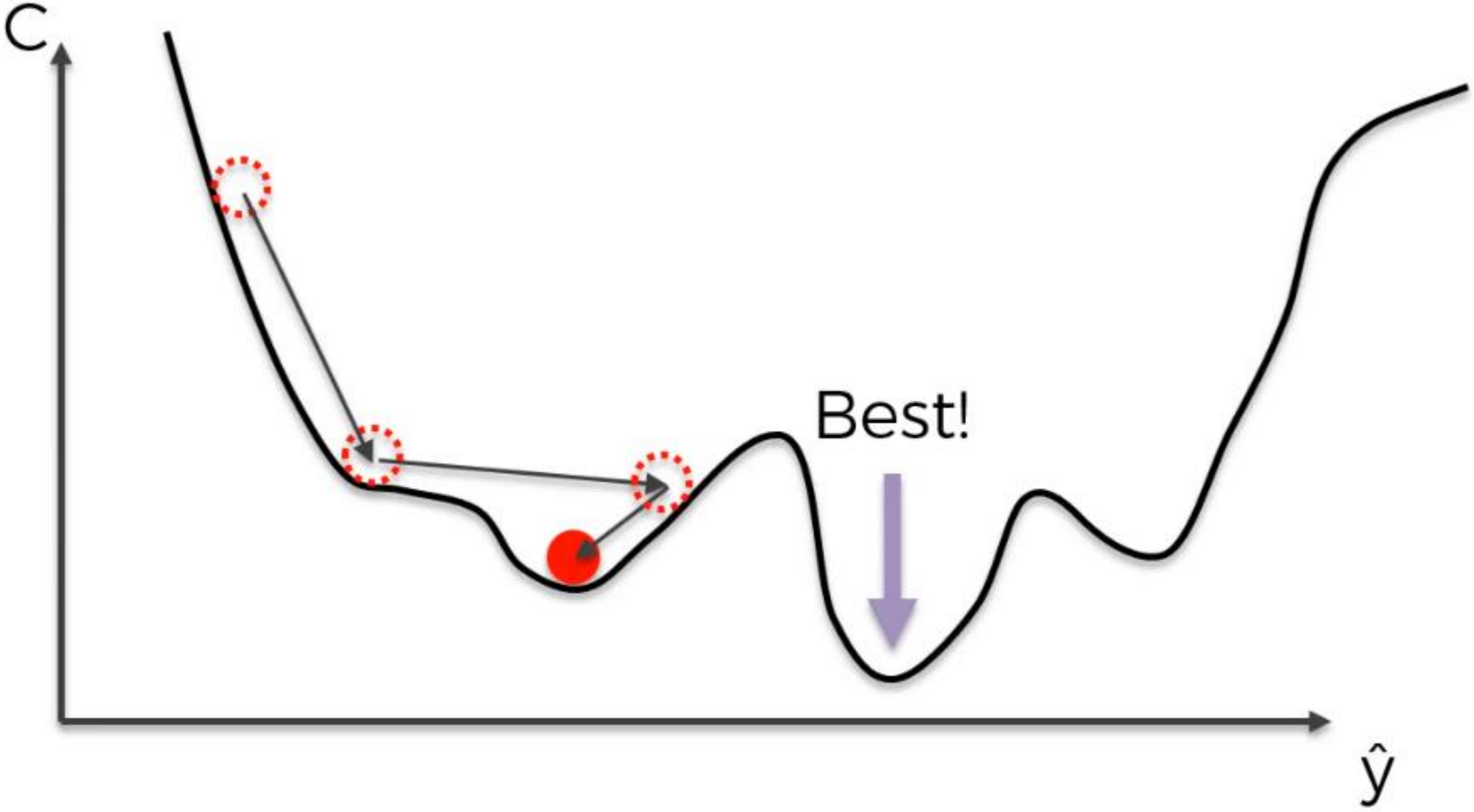
Step 6: Go back to **Step 3** and repeat until **Step Size** is very small, or you reach the **Maximum Number of Steps**

https://youtu.be/sDv4f4s2SB8

Summary

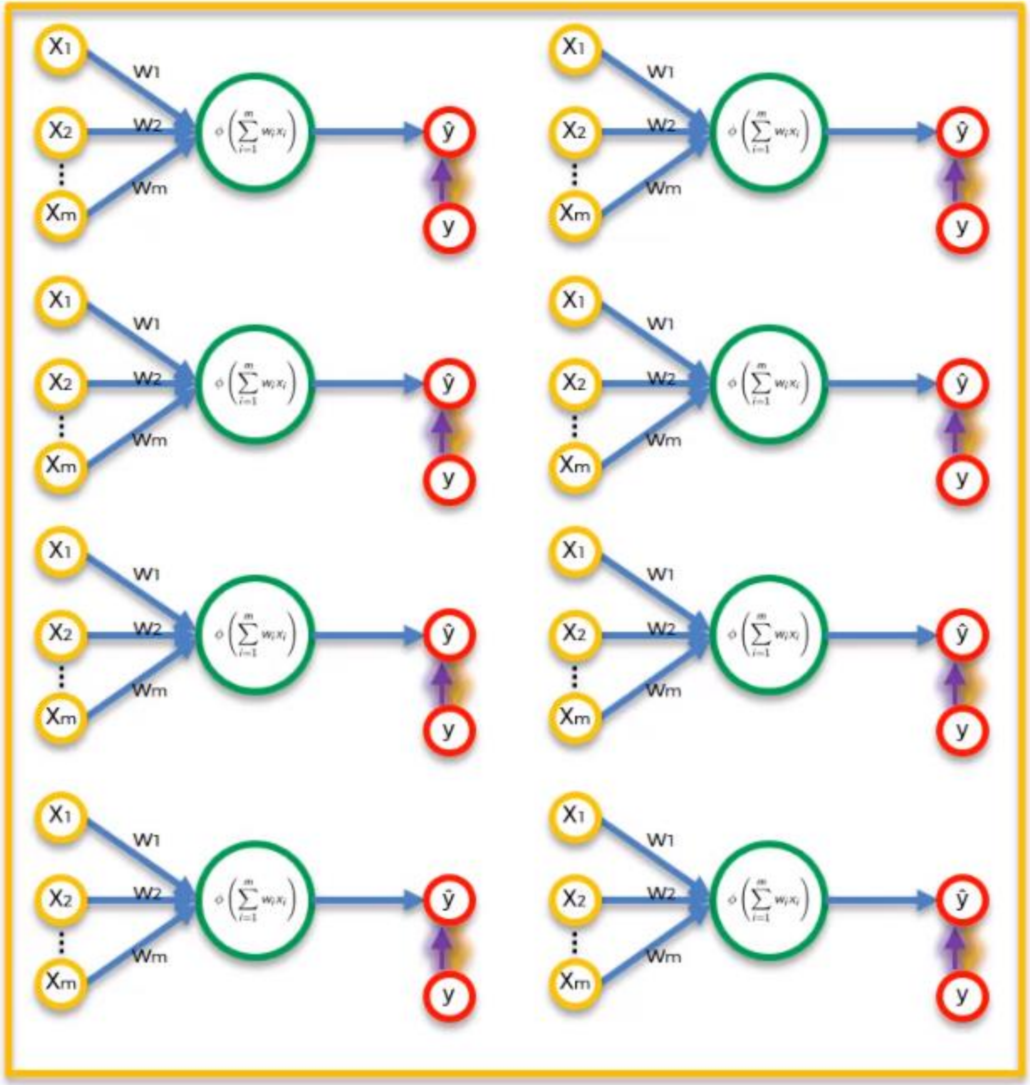
- Overview
- The Neuron
- The Activation Function
- How do Neural Networks work? (example)
- How do Neural Networks learn?
- Gradient Descent
- Stochastic Gradient Descent
- Backpropagation
- Algorithm summary

Stochastic Gradient Descent



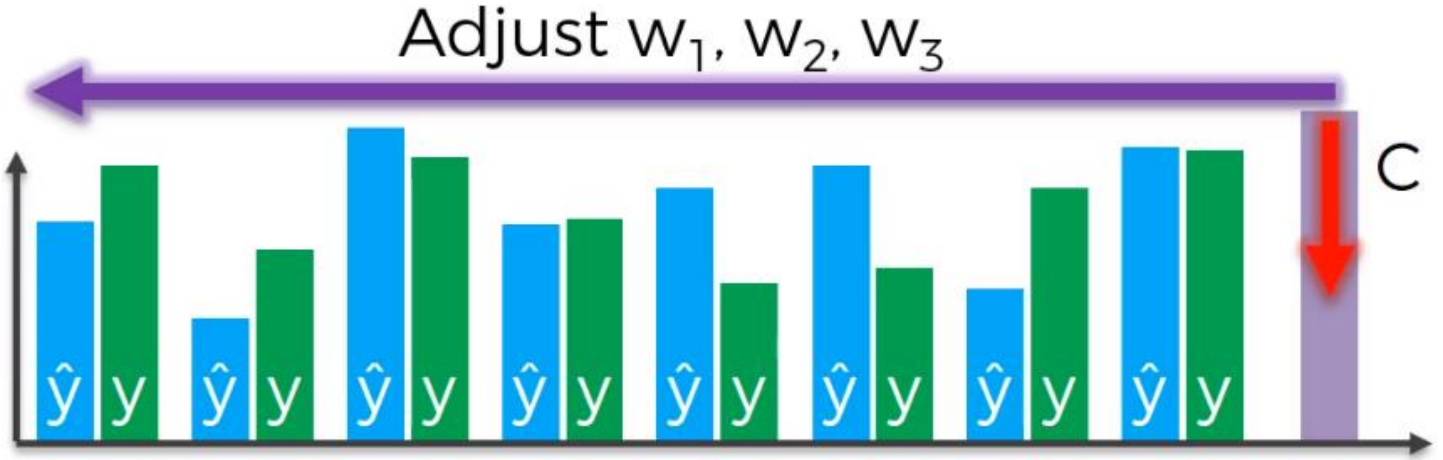
https://www.udemy.com/course/machinelearning/

Stochastic Gradient Descent



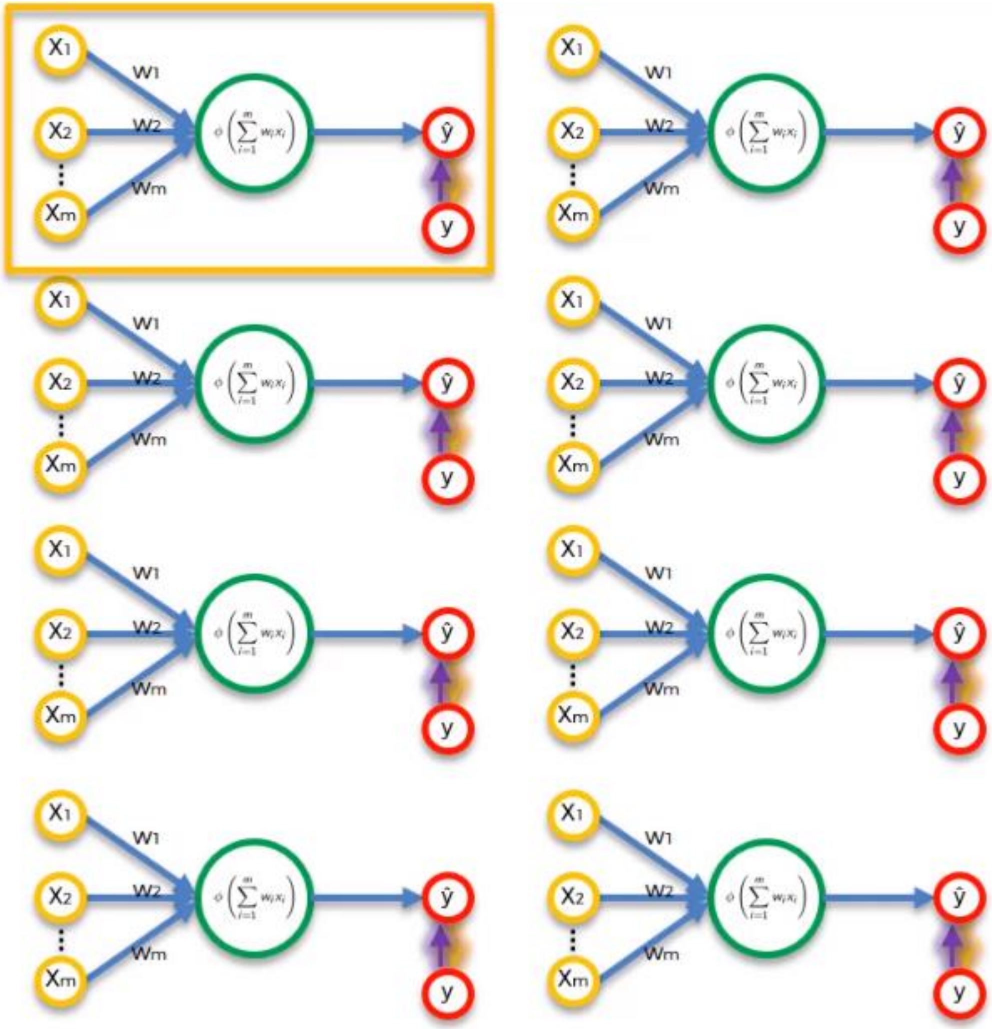
Row ID	Study Hrs	Sleep Hrs	Quiz	Exam
1	12	6	78%	93%
2	22	6.5	24%	68%
3	115	4	100%	95%
4	31	9	67%	75%
5	0	10	58%	51%
6	5	8	78%	60%
7	92	6	82%	89%
8	57	8	91%	97%

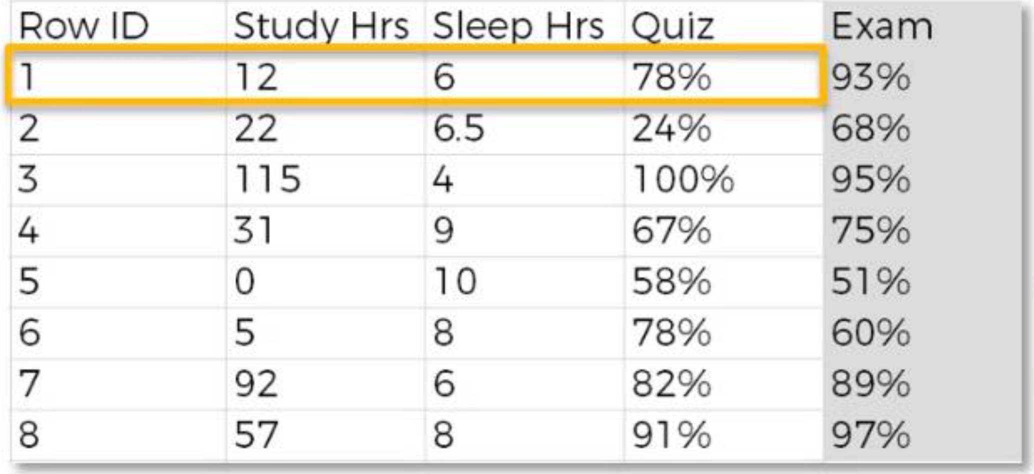
$$C = \sum \tfrac{1}{2}(\hat{y} - y)^2$$


Adjust w_1, w_2, w_3

https://www.udemy.com/course/machinelearning/

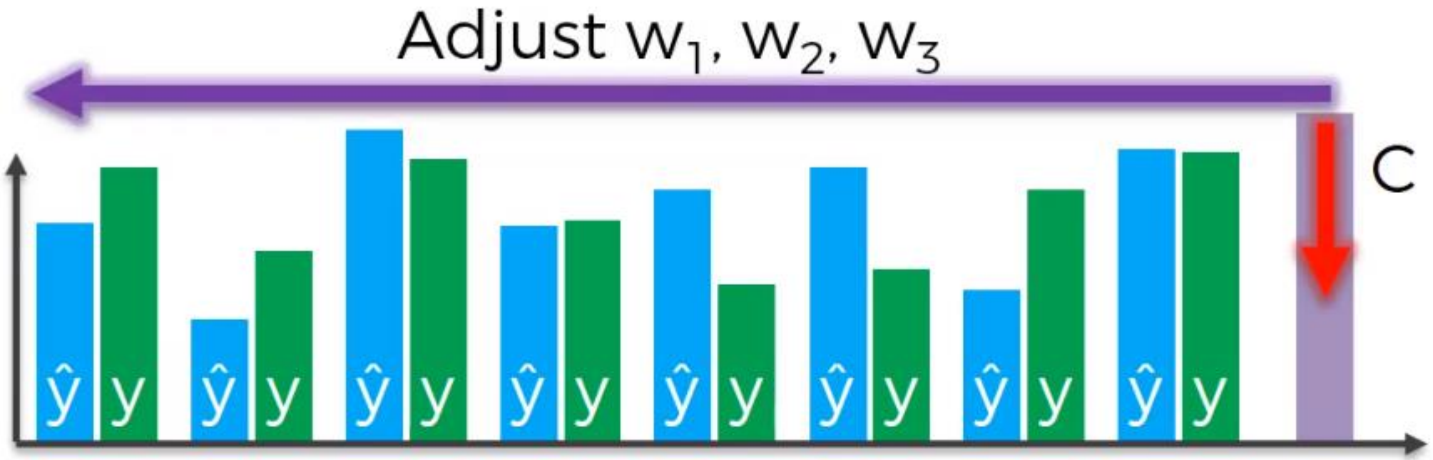
Stochastic Gradient Descent





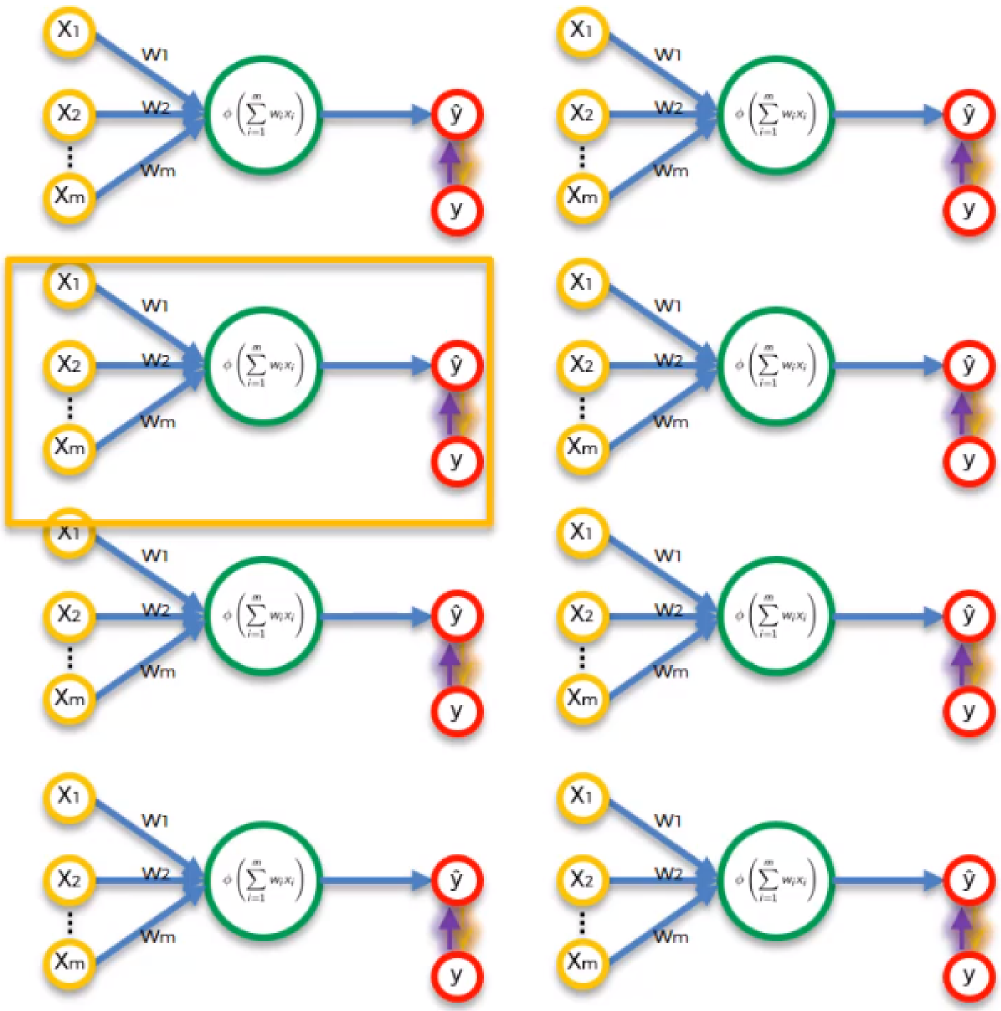
Row ID	Study Hrs	Sleep Hrs	Quiz	Exam
1	12	6	78%	93%
2	22	6.5	24%	68%
3	115	4	100%	95%
4	31	9	67%	75%
5	0	10	58%	51%
6	5	8	78%	60%
7	92	6	82%	89%
8	57	8	91%	97%

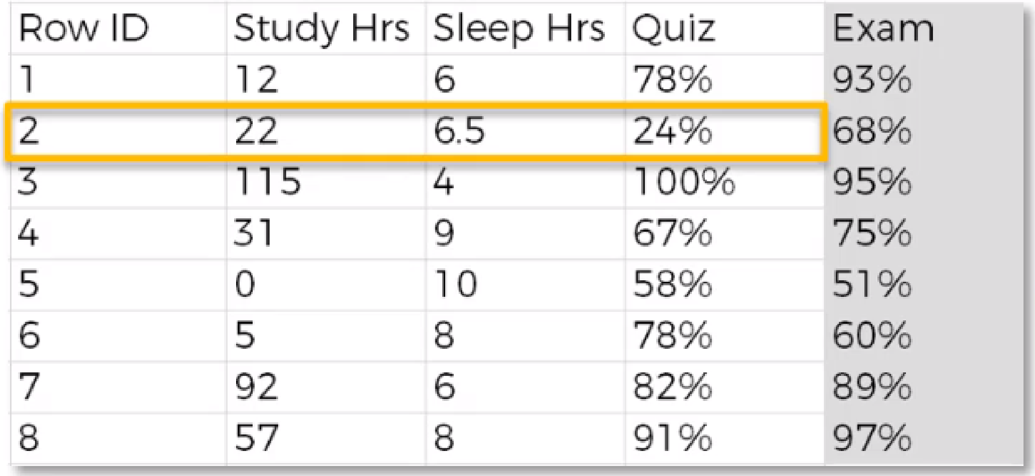
$$C = \sum \tfrac{1}{2}(\hat{y} - y)^2$$



https://www.udemy.com/course/machinelearning/

Stochastic Gradient Descent

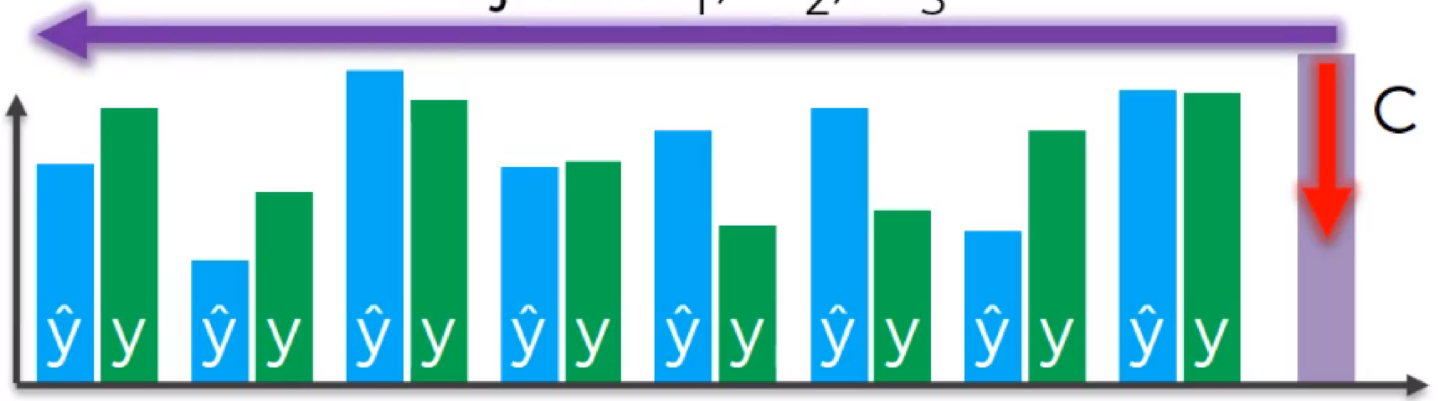




Row ID	Study Hrs	Sleep Hrs	Quiz	Exam
1	12	6	78%	93%
2	22	6.5	24%	68%
3	115	4	100%	95%
4	31	9	67%	75%
5	0	10	58%	51%
6	5	8	78%	60%
7	92	6	82%	89%
8	57	8	91%	97%

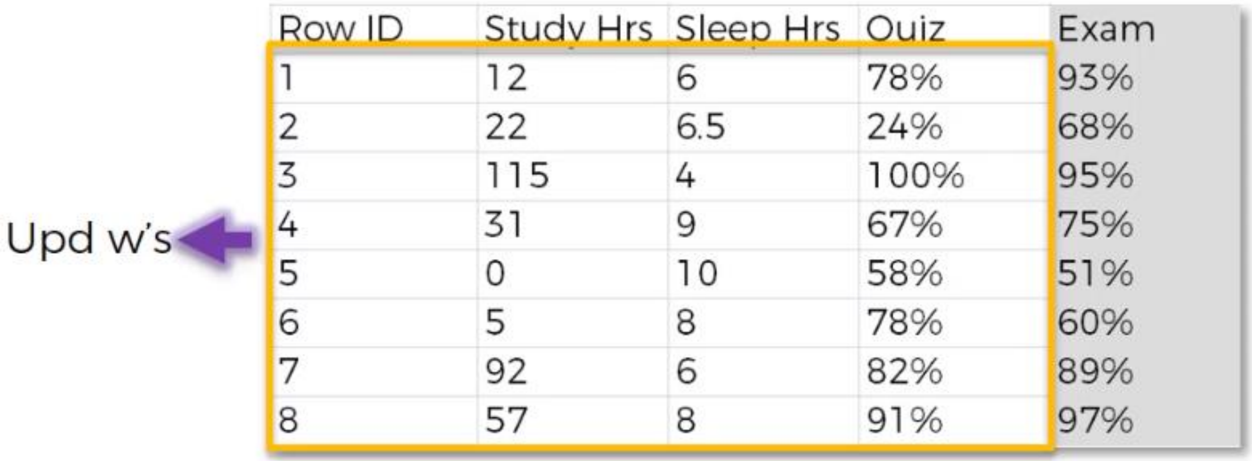
$$C = \sum \tfrac{1}{2}(\hat{y} - y)^2$$

Adjust w_1, w_2, w_3



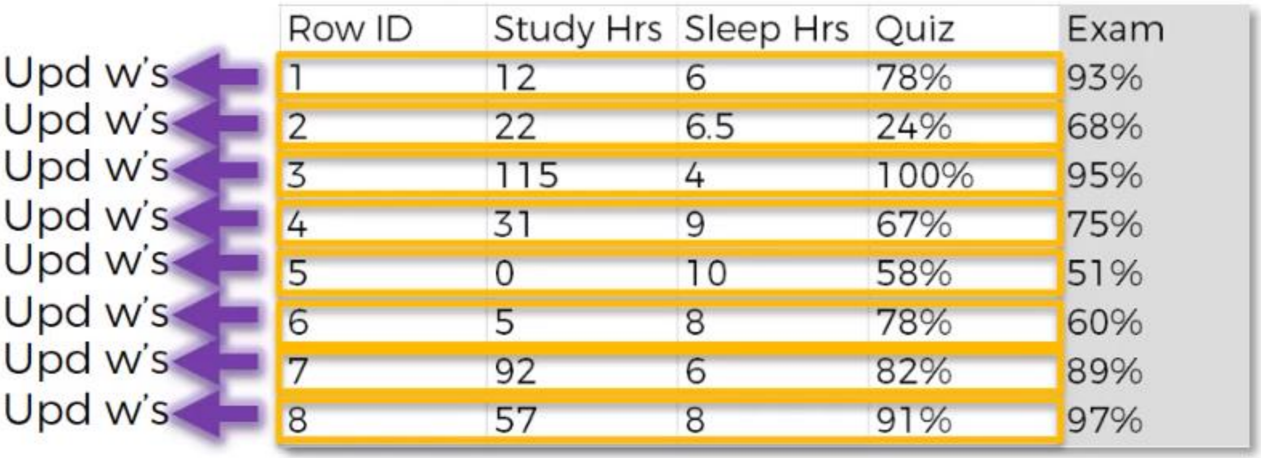
https://www.udemy.com/course/machinelearning/

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%

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%

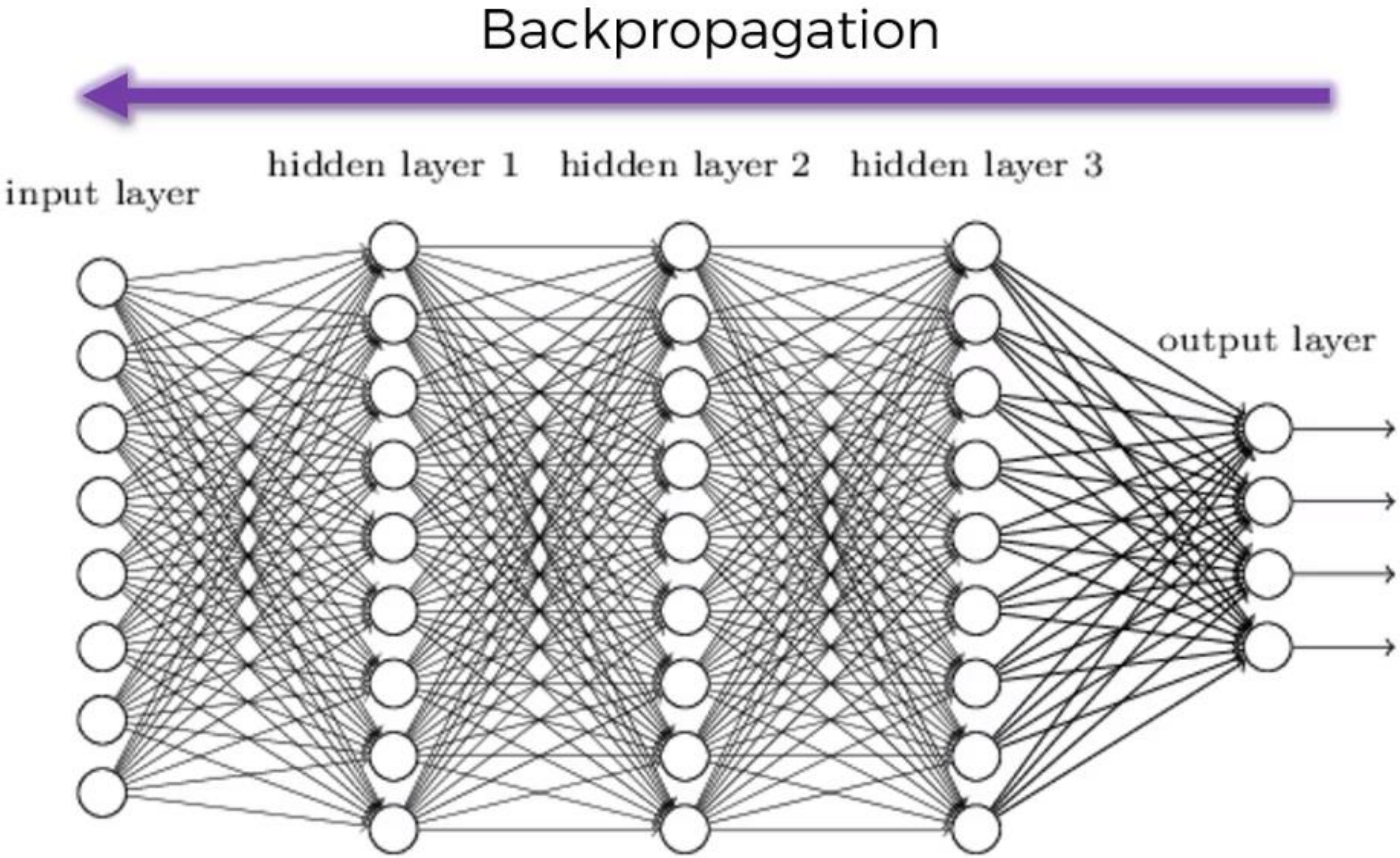
Stochastic Gradient Descent

https://www.udemy.com/course/machinelearning/

Summary

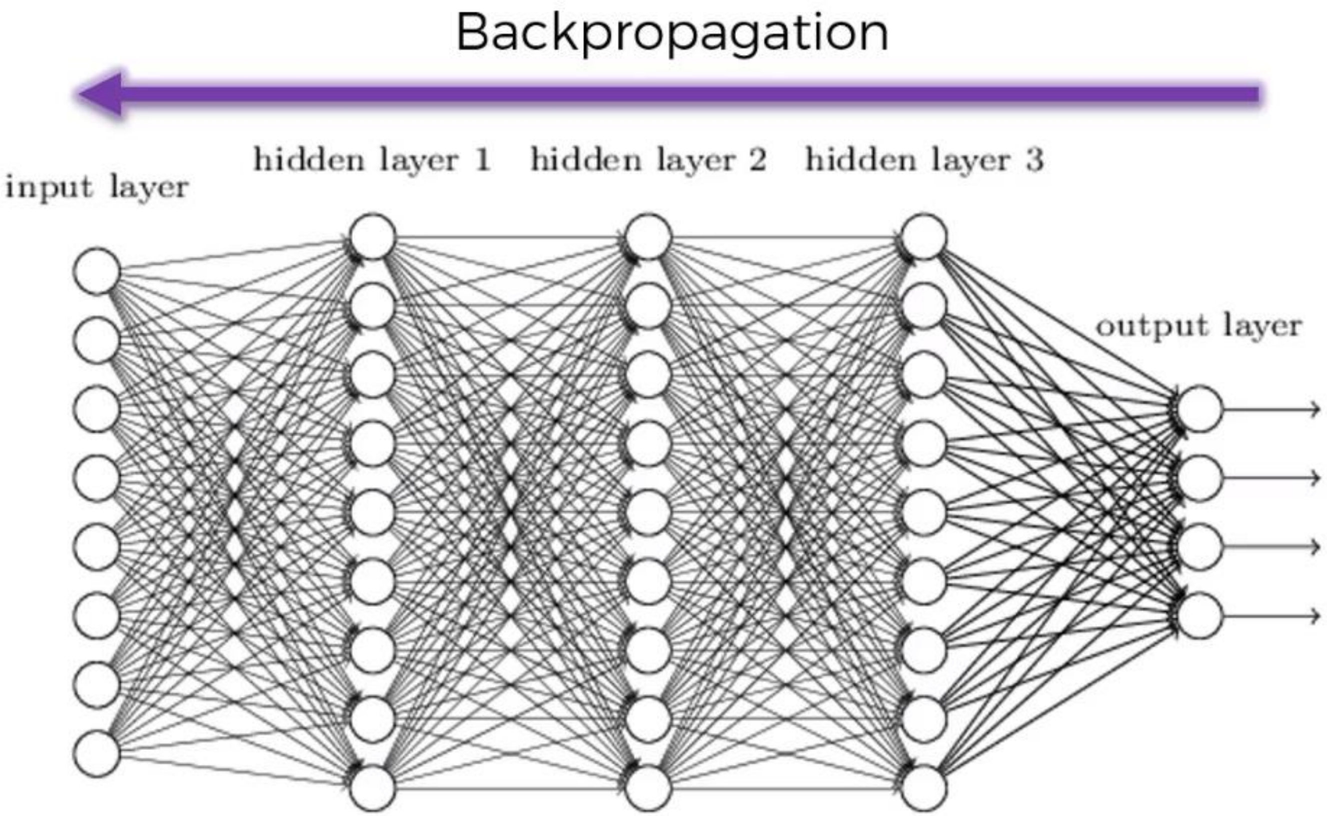
- Overview
- The Neuron
- The Activation Function
- How do Neural Networks work? (example)
- How do Neural Networks learn?
- Gradient Descent
- Stochastic Gradient Descent
- Backpropagation
- Algorithm summary

Backpropagation



https://www.udemy.com/course/machinelearning/

Backpropagation



Gradient of the cost function

Input: 270 weights/biases
Output: 270 weights/biases nudges
Parameters: 1 number (the cost)

Cost function

Input: 270 weights/biases
Output: 1 number (the cost)
Parameters: Many, many, many training example

Neural Network function

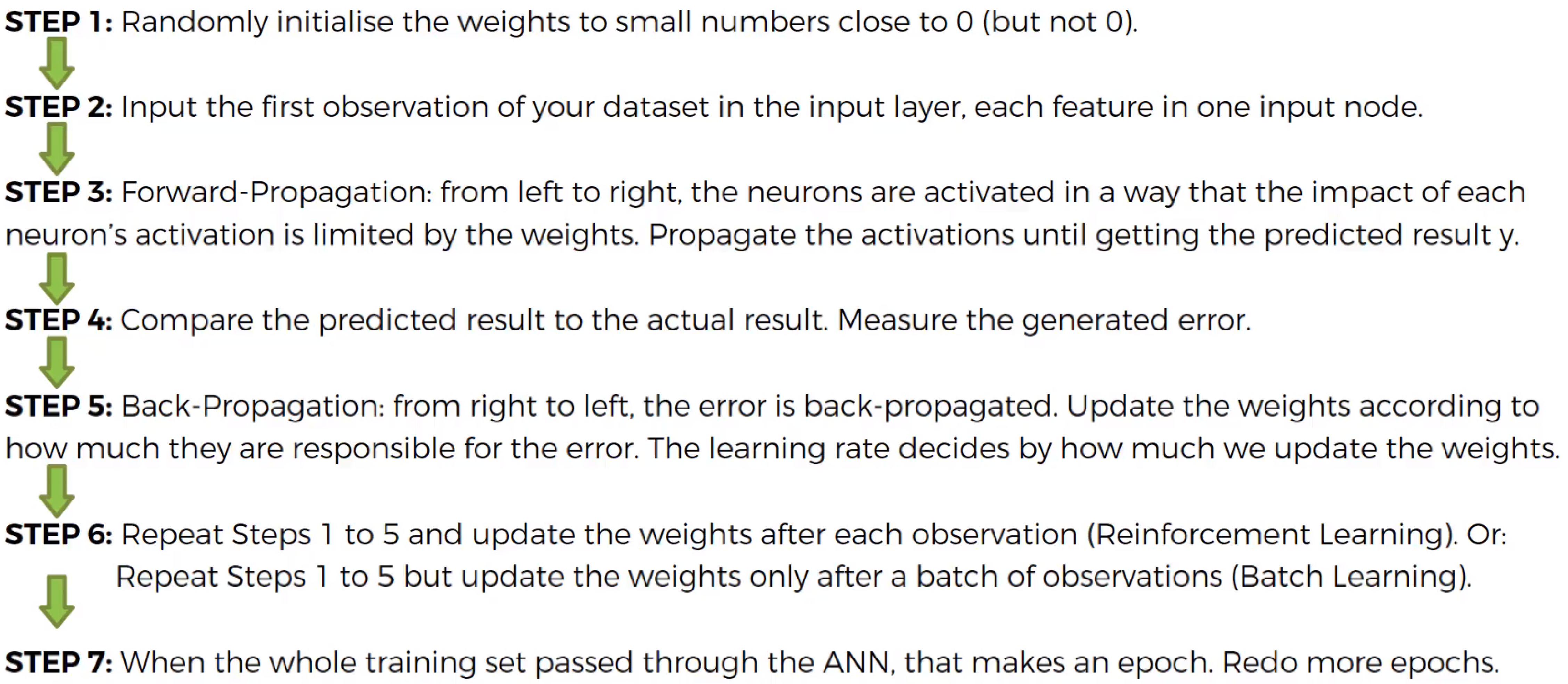
Input: 8 numbers (features)
Output: 4 numbers
Parameters: 270 weights/biases

https://www.udemy.com/course/machinelearning/

Summary

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

Algorithm summary



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.

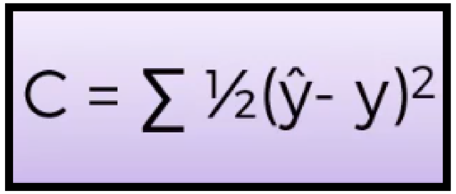
STEP 6: Repeat Steps 1 to 5 and update the weights after each observation (Reinforcement Learning). Or:
Repeat Steps 1 to 5 but update the weights only after a batch of observations (Batch Learning).

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

https://www.udemy.com/course/machinelearning/

Algorithms used

- Neural Network architecture/topology: **Deep Feed Forward (DFF)**
- Activation function: **Rectifier** and **sigmoid**
- Training method/Optimizer:
  - **Backpropagation** to calculate the derivatives and
  - **Gradient Descent** which descends through the gradient, i.e. adjust the parameters
- Cost/loss function:



$$C = \sum ½(\hat{y} - y)^2$$