Lecture 2

Perceptron

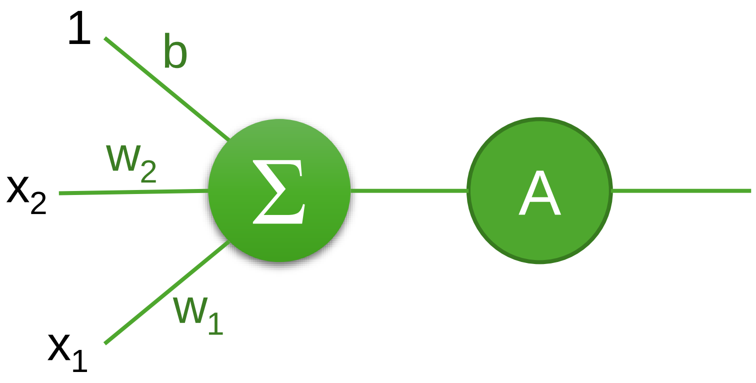
CSE465: Pattern Recognition and Neural Network
Sec: 3
Faculty: Silvia Ahmed (SvA)
Spring 2025

Today’s Topic

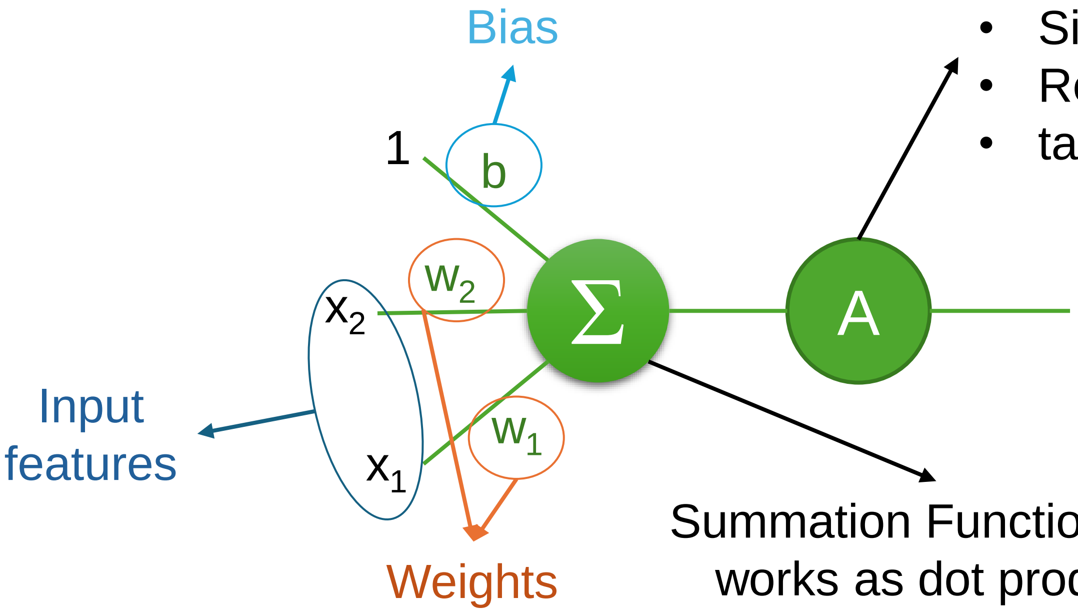
1. Perceptron:
   - What is a Perceptron?
   - Perceptron vs Neuron
   - Geometric Intuition
   - How to train a Perceptron?

What is a Perceptron?

- Fundamental building block of ANN
- It is an algorithm, used for supervised ML.
- A Perceptron is a simple type of artificial neural network algorithm developed by Frank Rosenblatt in 1957.
- It's the basic unit of a neural network, taking multiple binary inputs and producing a single binary output.
- It computes a weighted sum of its input, applies an activation function, and produces an output.



Different parts of Perceptron



Bias

Input features

Weights

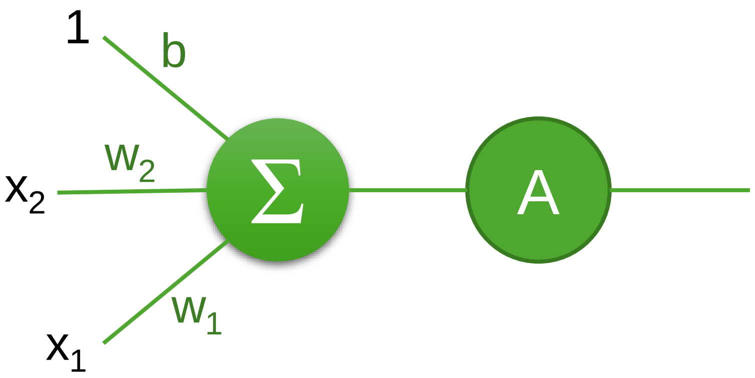
Activation Function:
- Signum Function
- Sigmoid
- ReLU
- tanh

Summation Function that works as dot product

$$z = w_1 \cdot x_1 + w_2 \cdot x_2 + b$$

Example use of a Perceptron

IQ, x_1	CGPA, x_2	Job Placement
78	7.8	1
69	5.1	0
…	…	…



1) Training:

Main job is to learn the values of the weights and the bias from the training samples
Eg. w_1 =1, w_2 = 2, b = 3

2) Prediction:

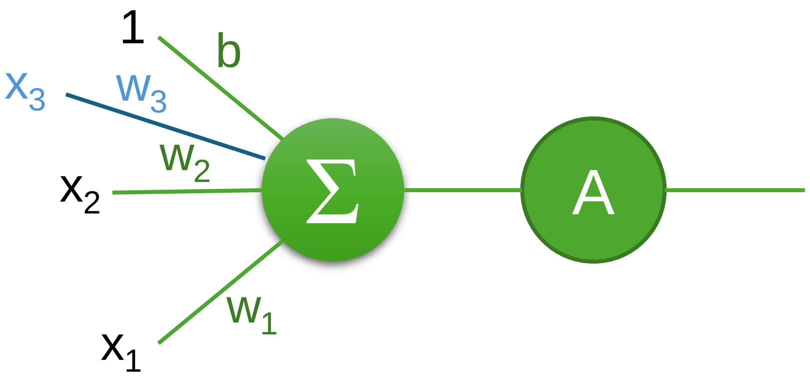
For a new sample where IQ = 100 and CGPA = 5.1:

$$z = 100 \times 1 + 5.1 \times 2 + 3 = 113.2 \geq 0$$

So Job placement = 1

- Question: If there are more than 2 features?

IQ, x_1	CGPA, x_2	State	Job Placement
78	7.8	Dhaka	1
69	5.1	Khulna	0
…	…		…



$$z = w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_3 + b$$

Perceptron vs Neuron

- Deep learning is inspired by nervous system.

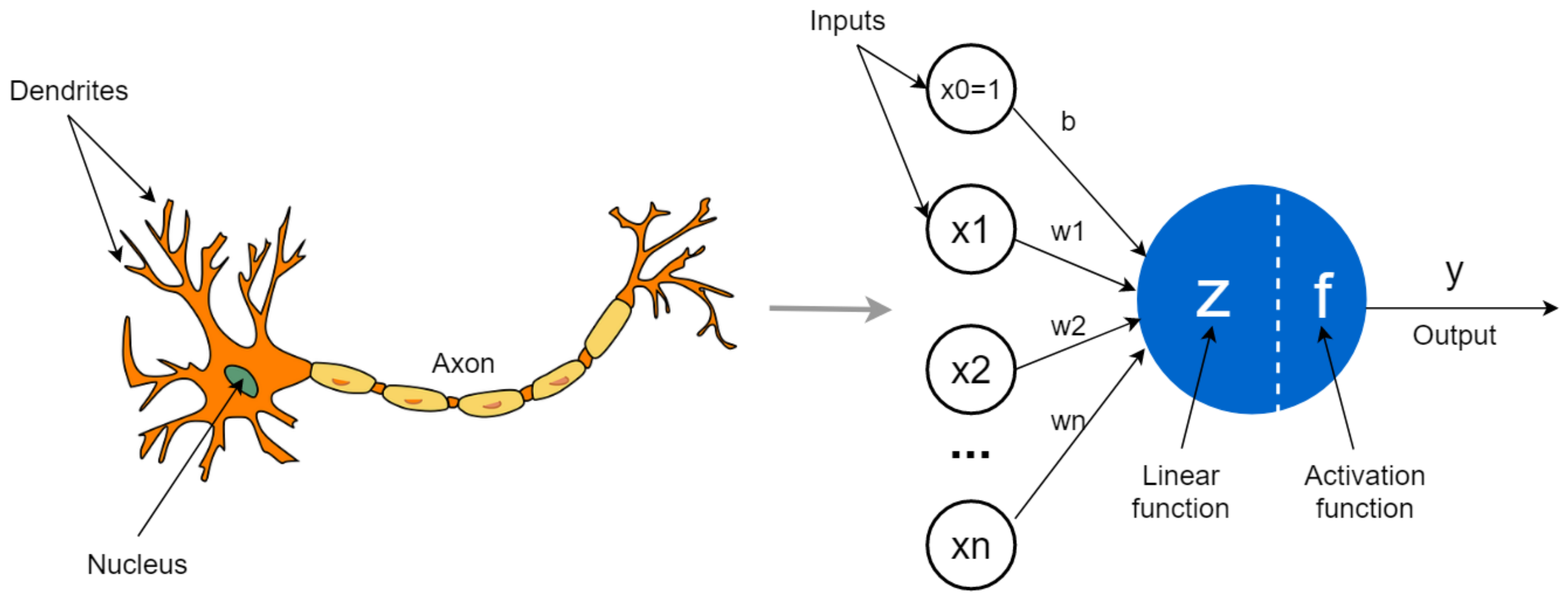
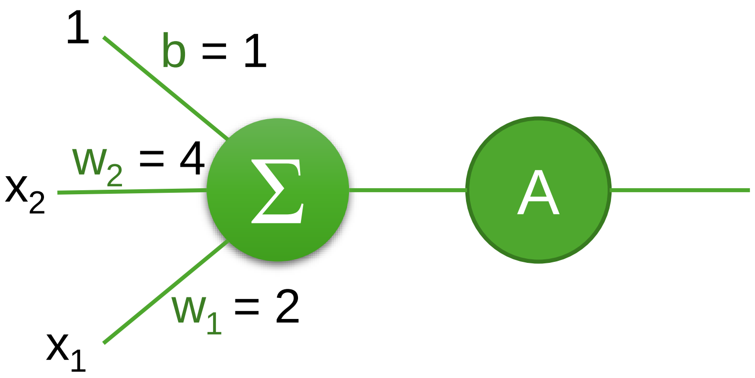


Figure: Perceptron vs Neuron [2]

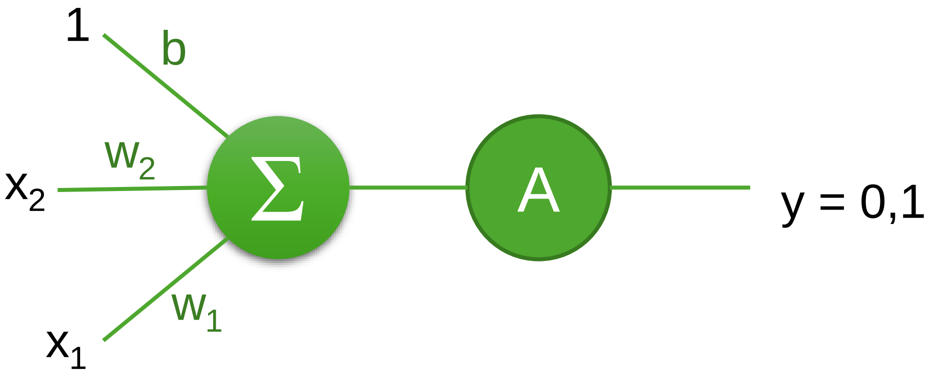
Interpretation

IQ, x_1	CGPA, x_2	Job Placement
78	7.8	1
69	5.1	0
…	…	…



- Weights actually depicts the strength of each (input) connections.
- Weights are mostly the feature importance.

Geometric Intuition



$$z = w_1 \cdot x_1 + w_2 \cdot x_2 + b$$

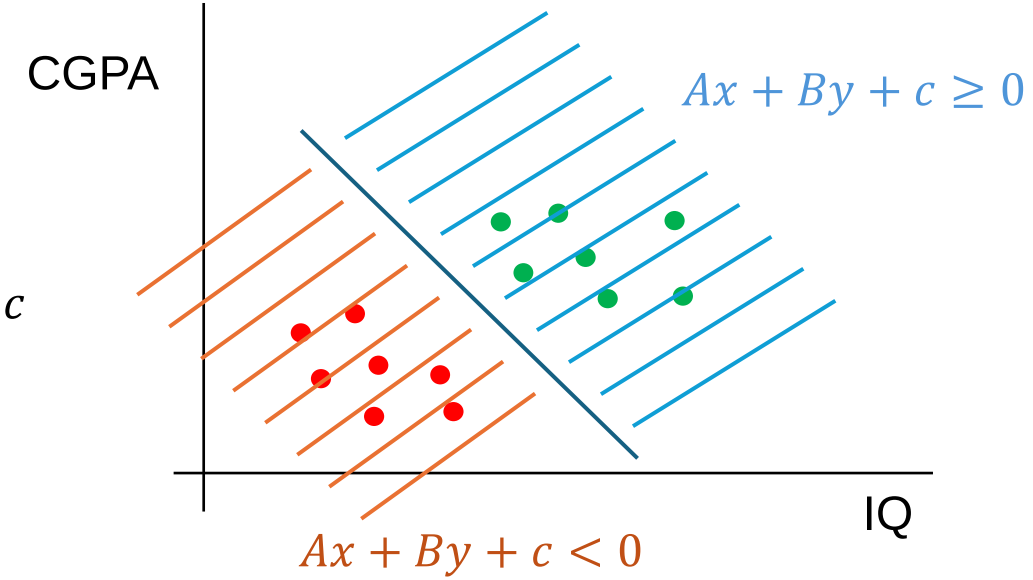
$$y = f(z) = \begin{cases} 1 & z \geq 0 \\ 0 & z < 0 \end{cases}$$

$$w_1 => A, w_2 => B, b => c$$
$$x_1 => x, x_2 => y$$

$$Ax + By + c$$

Equation of a line

IQ, x_1	CGPA, x_2	Job Placement

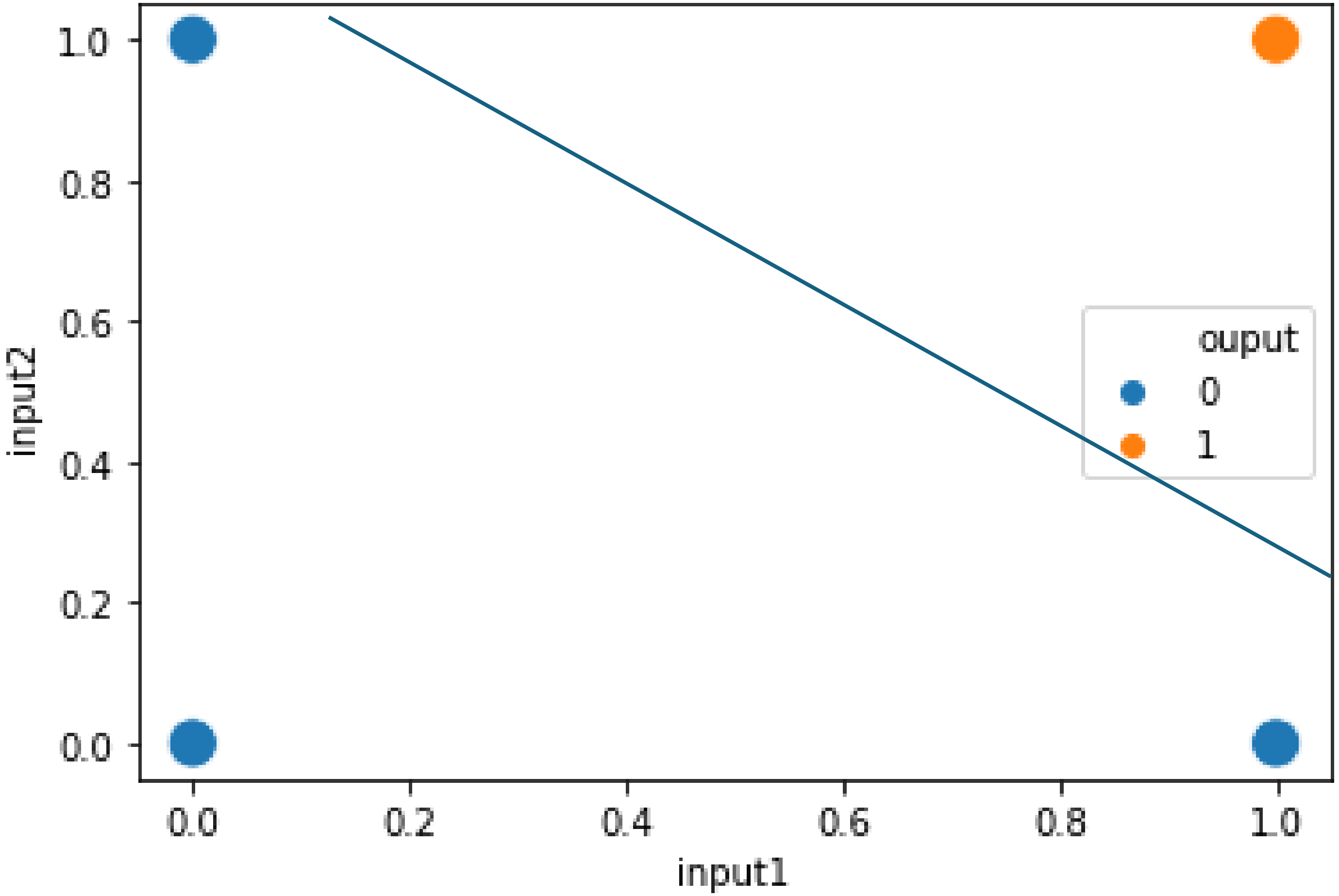


- Perceptron is a “line” and its main functionality is to create “regions”
- Perceptron is a **binary classifier**.

2D -> line
3D -> plane
≥4D -> hyperplane

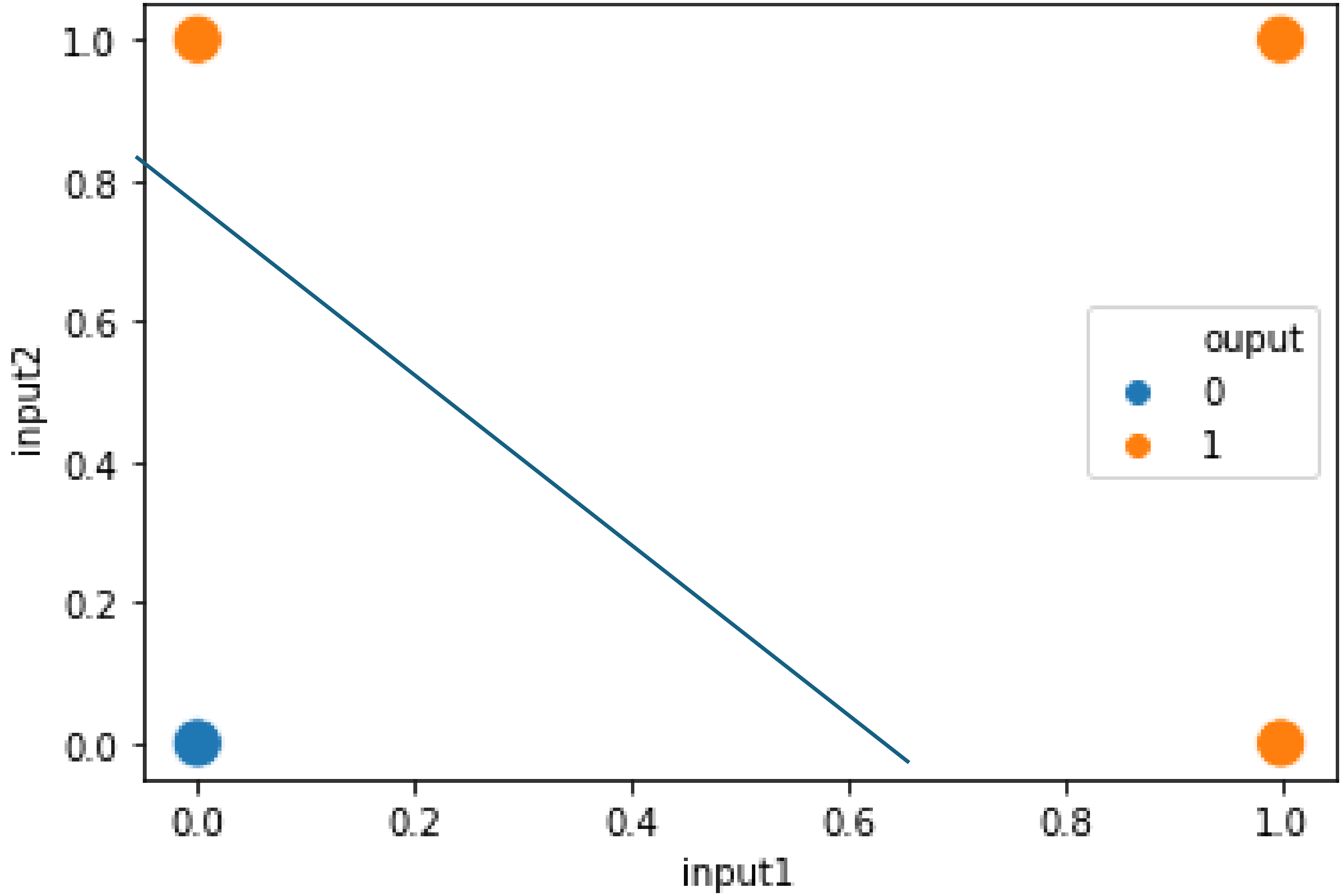
Logic AND

input 1	input 2	output
1	1	1
1	0	0
0	1	0
0	0	0



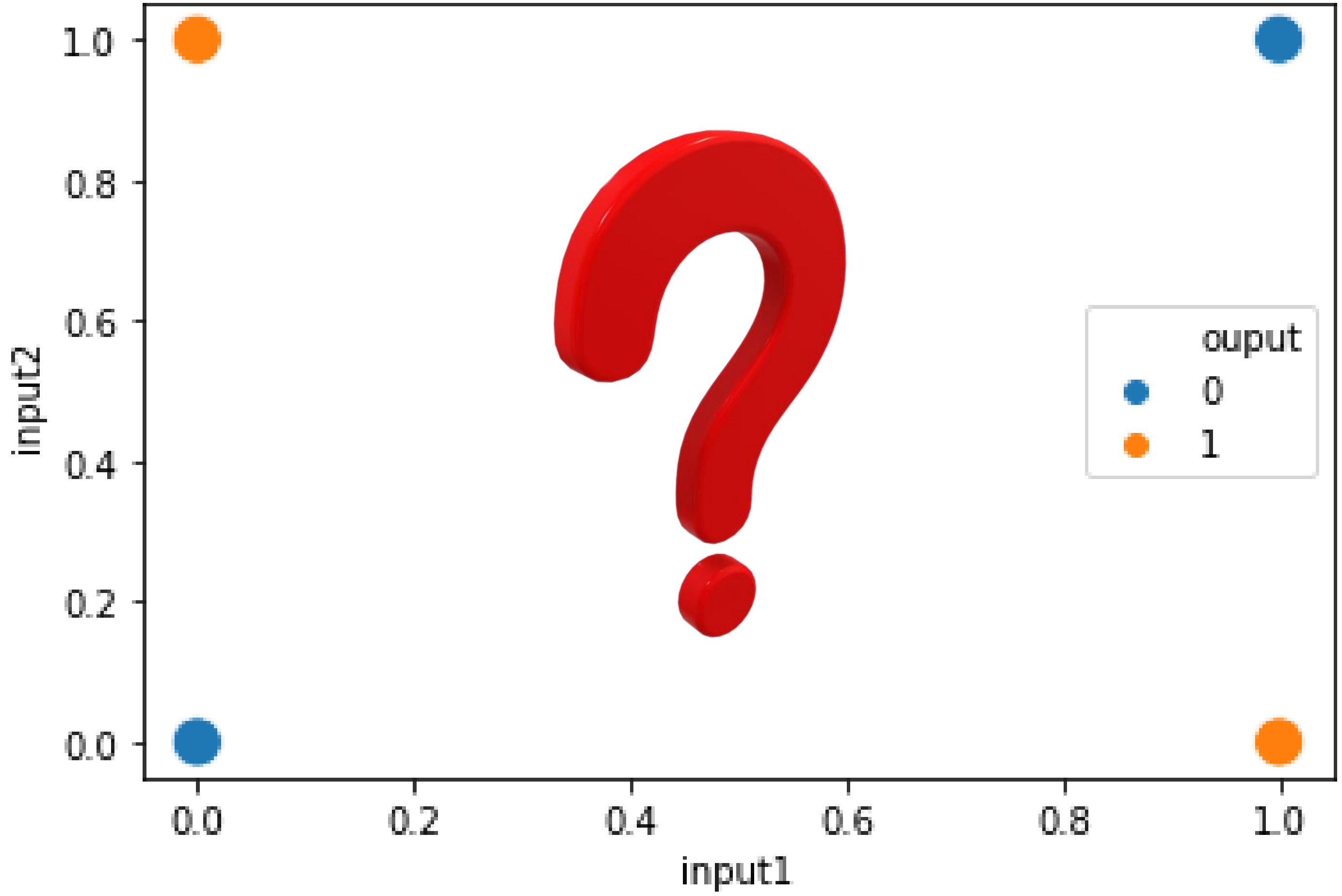
Logic OR

input 1	input 2	output
1	1	1
1	0	1
0	1	1
0	0	0



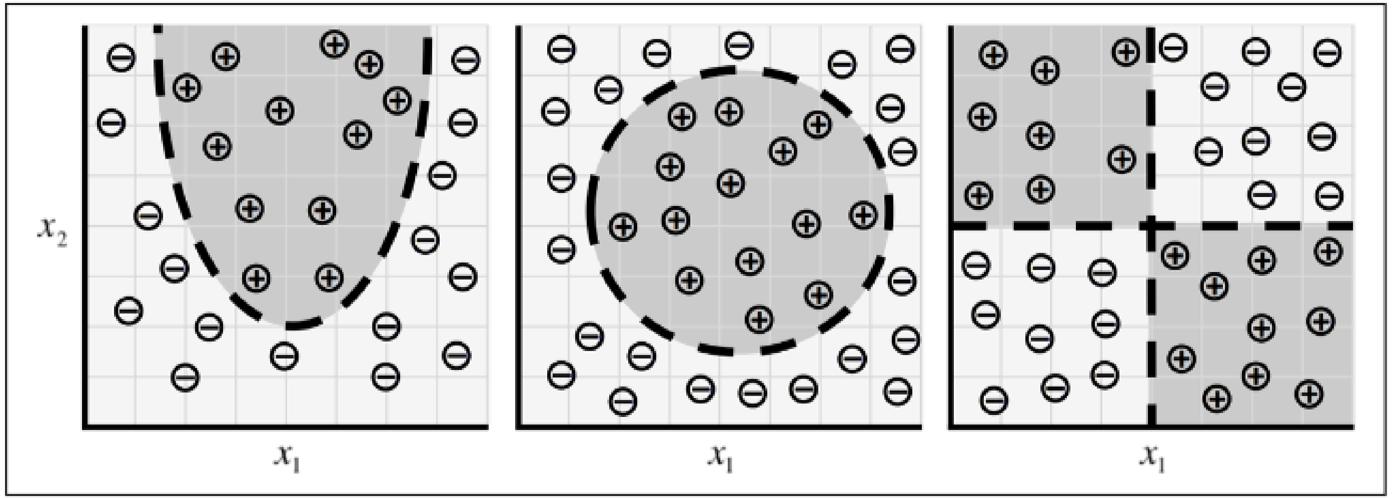
Logic XOR

input 1	input 2	output
1	1	0
1	0	1
0	1	1
0	0	0



Limitation

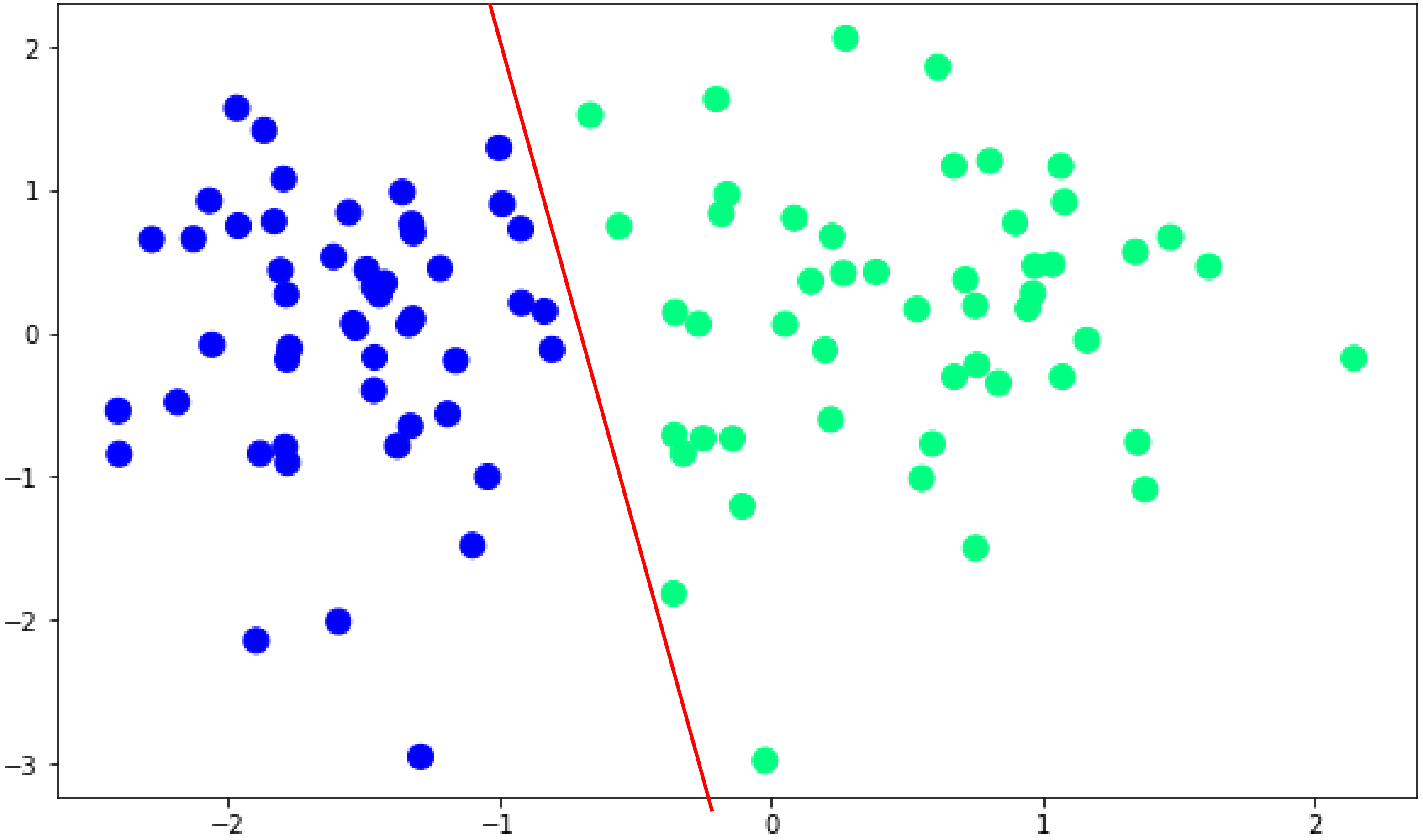
- Works only with linear or “sort-of” linear data



- Tensorflow playground: playground.tensorflow.org

Dataset type	Noise	Learning rate	Activation
Gaussian	15-20	0.01	Sigmoid
Exclusive OR	15-20	0.01	Sigmoid

Perception Trick

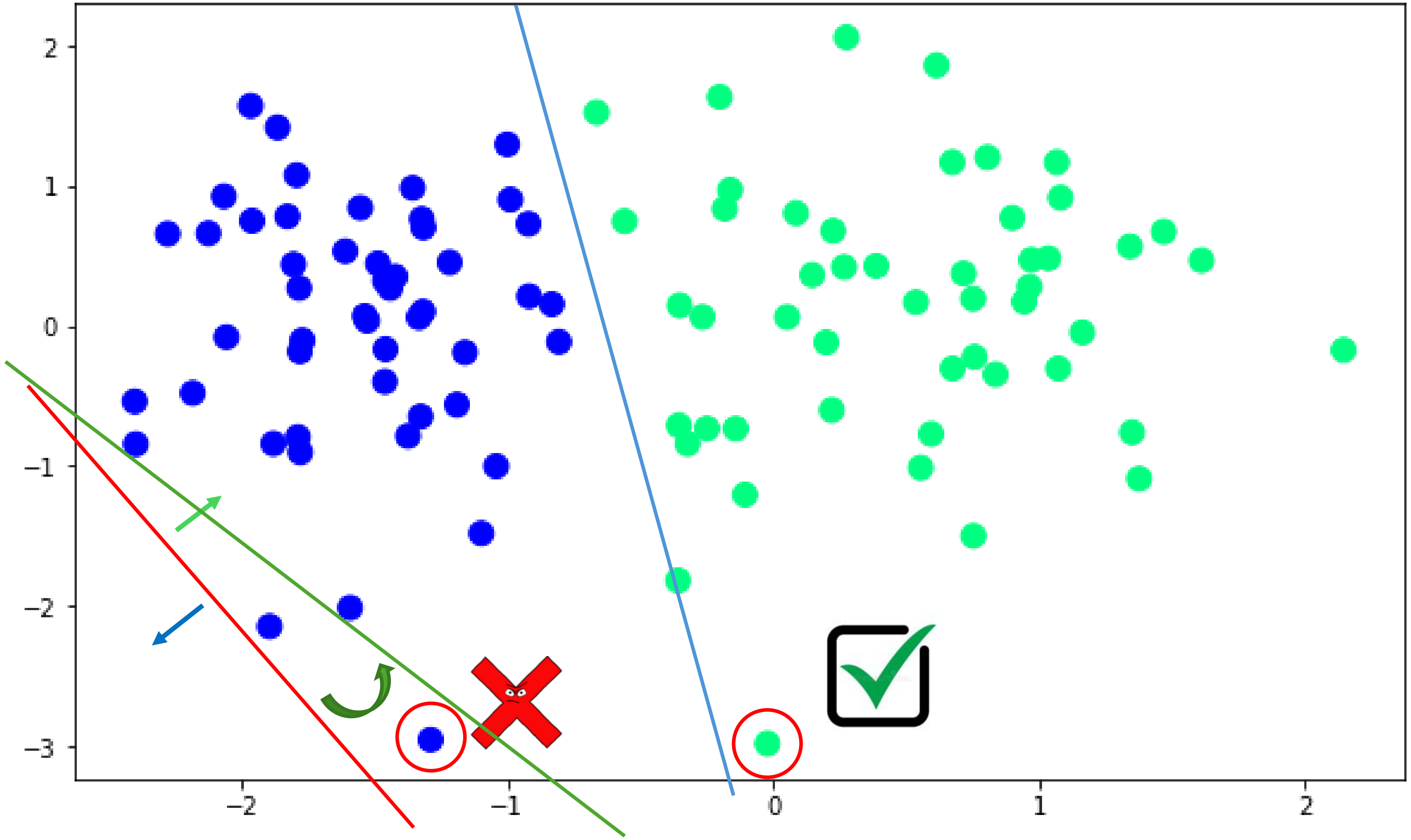


- Main target is to get the decision boundary in the form:

$$\sum_{i=0}^{n} w_i x_i = 0$$

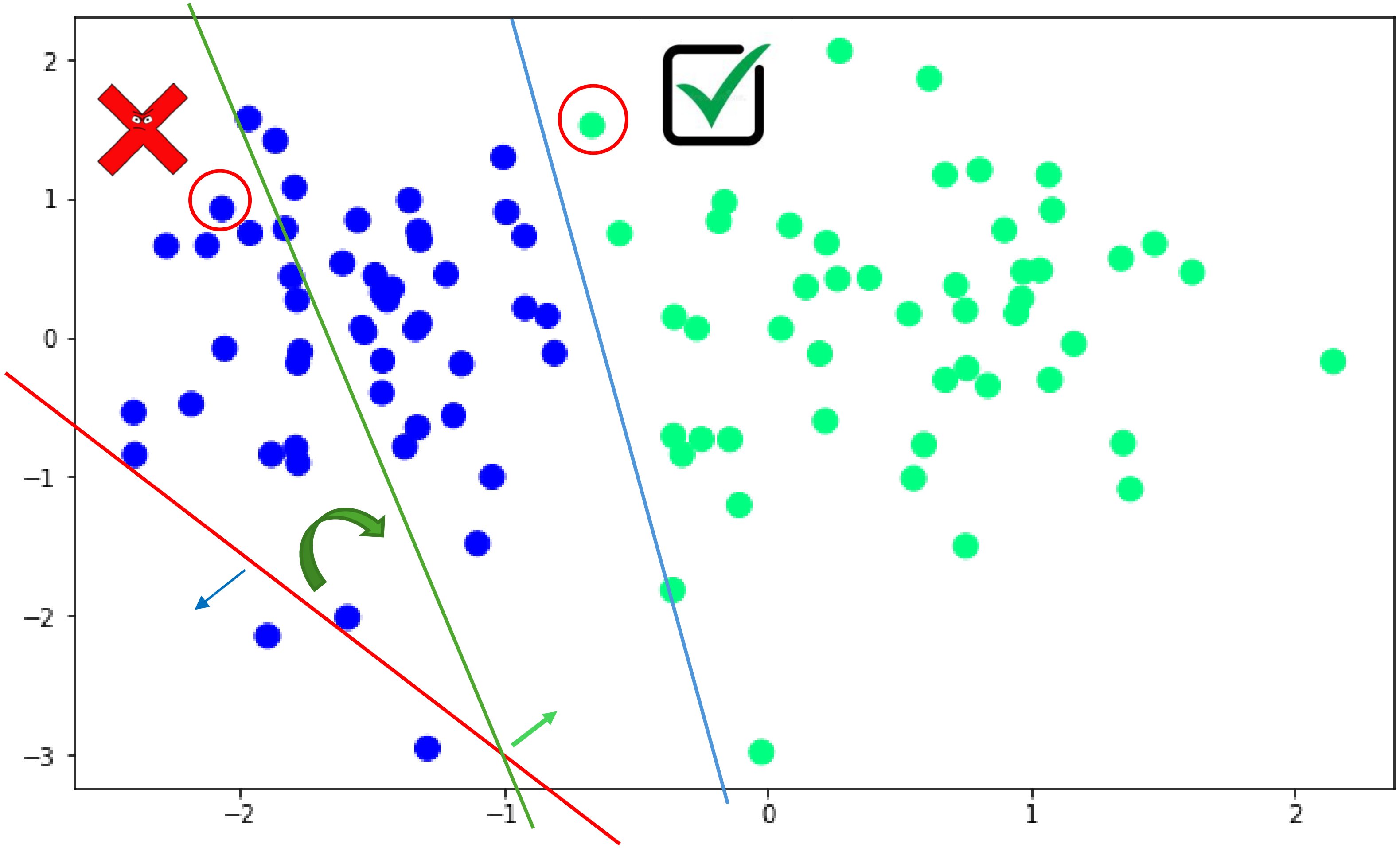
Steps - 1

- Initialize:
- A = 1, B = 1, C = 0
- Randomly select one sample



Steps - 2

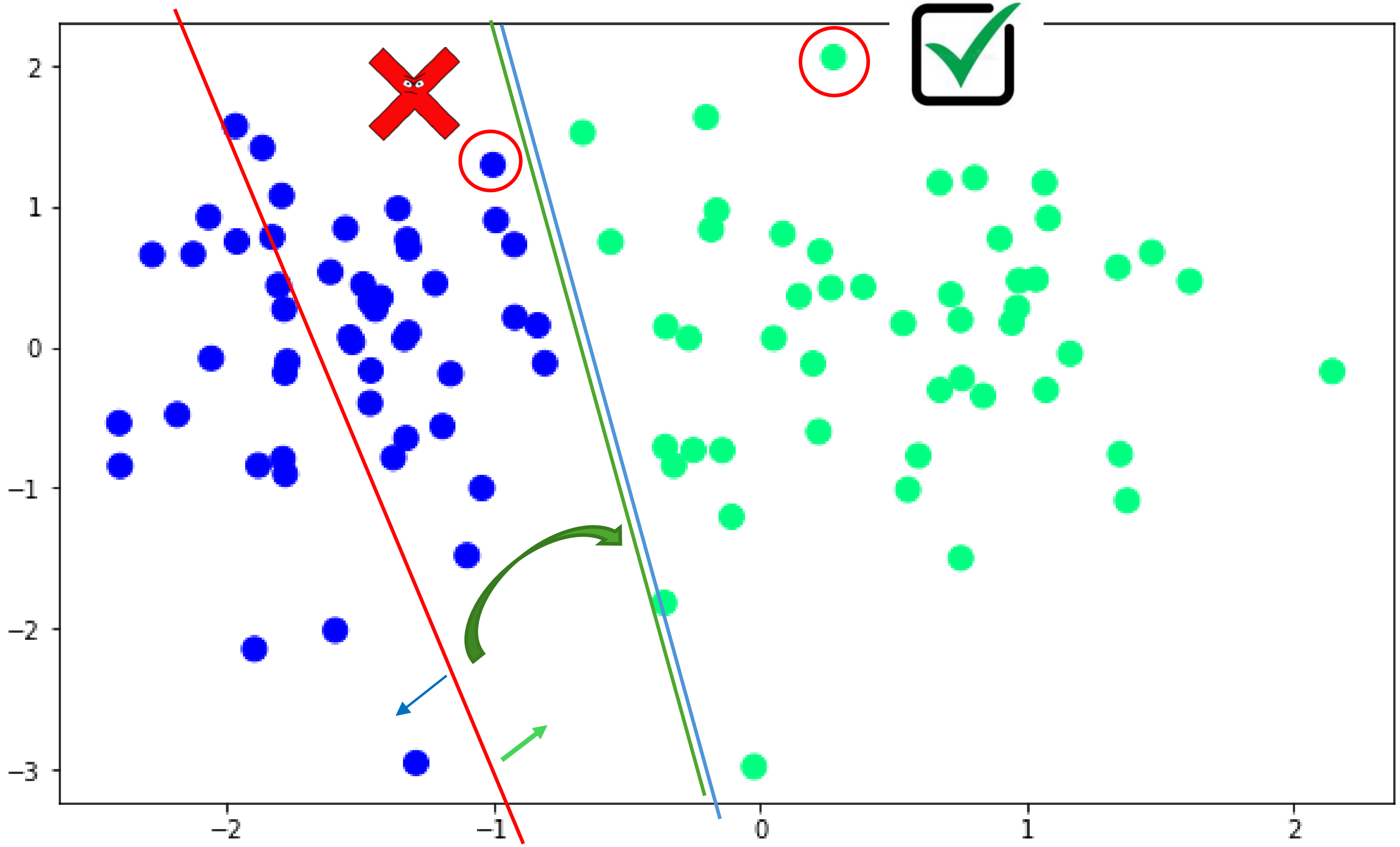
- Initialize:
- A = 2, B = 1.5, C = 0.4
- Randomly select one sample



Steps - 3

- Initialize:
- A = 4, B = 1.5, C = 0.4

- Randomly select one sample



Line Transformation

- Shown in desmos.com/calculator
- Ax+By+C=0

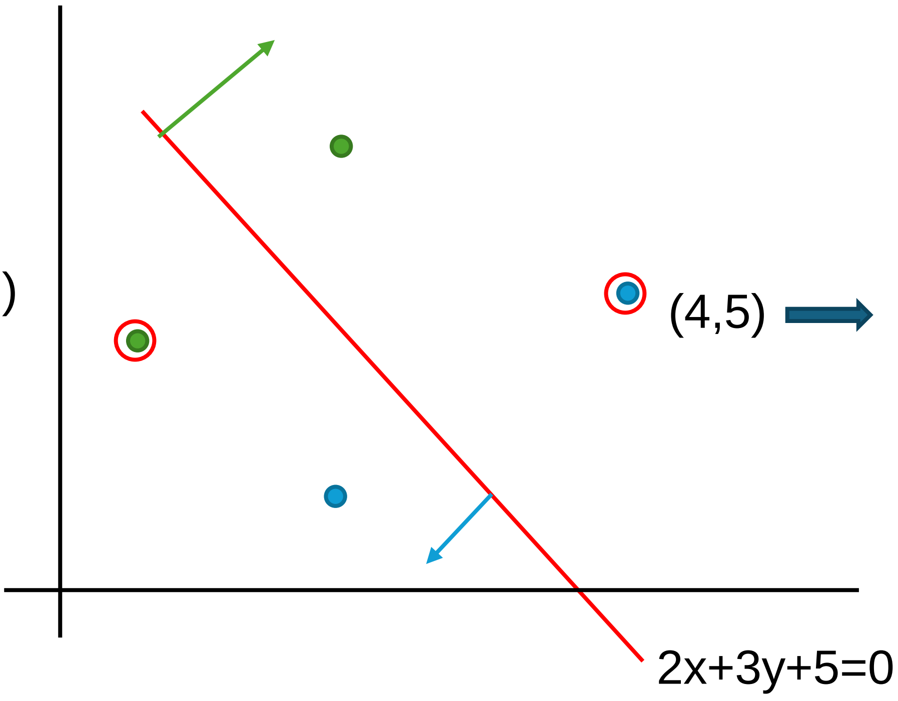
Main equation: 2x+3y+5=0			Effect
Change in c	2x+3y+10=0	2x+3y+0=0	
Change in A	4x+3y+5=0	x+3y+5=0	
Change in B	2x+6y+5=0	2x+y+5=0	

How much to transform?

(1,3,1) ⟵ (1,3)

$$\begin{array}{rrrr} & 2 & 3 & 5 \\ (+) & 1 & 3 & 1 \\ \hline & 3 & 6 & 6 \end{array}$$

<u>Plus operation</u> to bring the wrongly “negative” point to the correct “positive” zone.



2x+3y+5=0

(4,5) ⟶ (4,5,1)

$$\begin{array}{rrrr} & 2 & 3 & 5 \\ (-) & 4 & 5 & 1 \\ \hline & -2 & -2 & 4 \end{array}$$

<u>Minus operation</u> to bring the wrongly “positive” point to the correct “negative” zone.

Live Desmos demonstration

2x+3y+5=0	(5,2)	(-3,-2)

Learning rate

- The **learning rate** is a small number that controls **how fast or slow** a machine learning or deep learning model updates its internal parameters (like weights) during training.
- "It’s like the step size your model takes while learning. Too big, and it may trip and fall. Too small, and it may take forever to learn."
- New coef = coef – learning rate * coef
- Why it's important:
  - If the **learning rate is too high** → the model may skip over the best solution and never settle.
  - If the **learning rate is too low** → the model will learn very slowly, taking a long time to improve (or getting stuck).

Algorithm

- epoch = 1000, η = 0.01

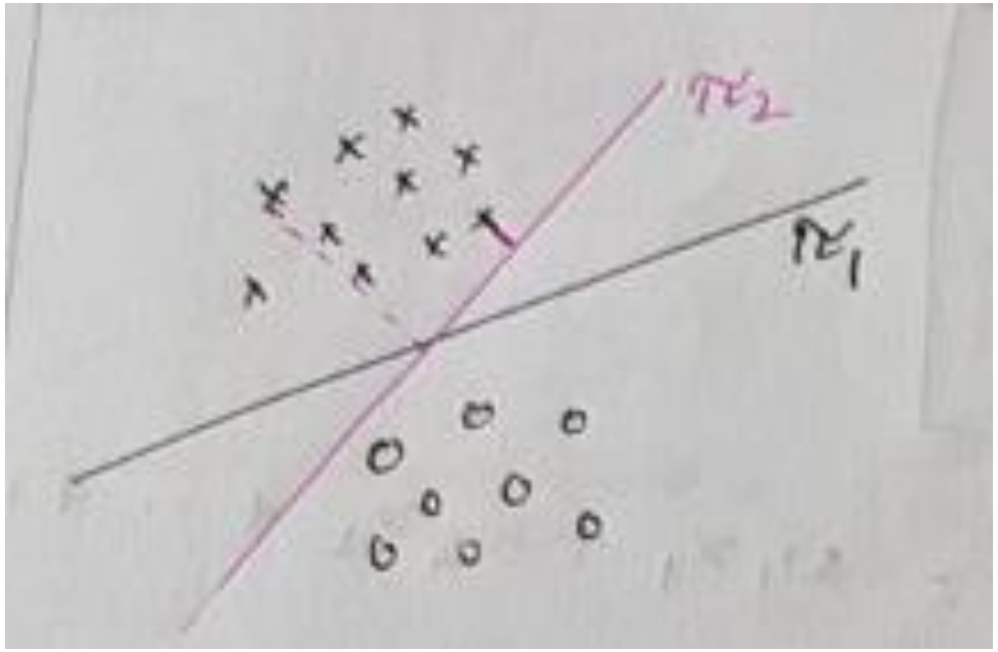
```
for i in range(epoch):
        randomly select a point
        if x_i ∈ N and $\sum_{i=0}^{2} w_i x_i \geq 0$
                $w_{new} = w_{old} - \eta\, x_i$
        if x_i ∈ P and $\sum_{i=0}^{2} w_i x_i < 0$
                $w_{new} = w_{old} + \eta\, x_i$
```

```
for i in range(epoch):
        randomly select a point
        $w_{new} = w_{old} + \eta(y_i - \hat{y}_i)x_i$
```

y_i	$\hat{y}_i$	$y_i - \hat{y}_i$
1	1	0
0	0	0
1	0	1
0	1	-1

Problem with Perceptron Trick

- Which decision boundary is better?
- Quantify the result
- Convergence

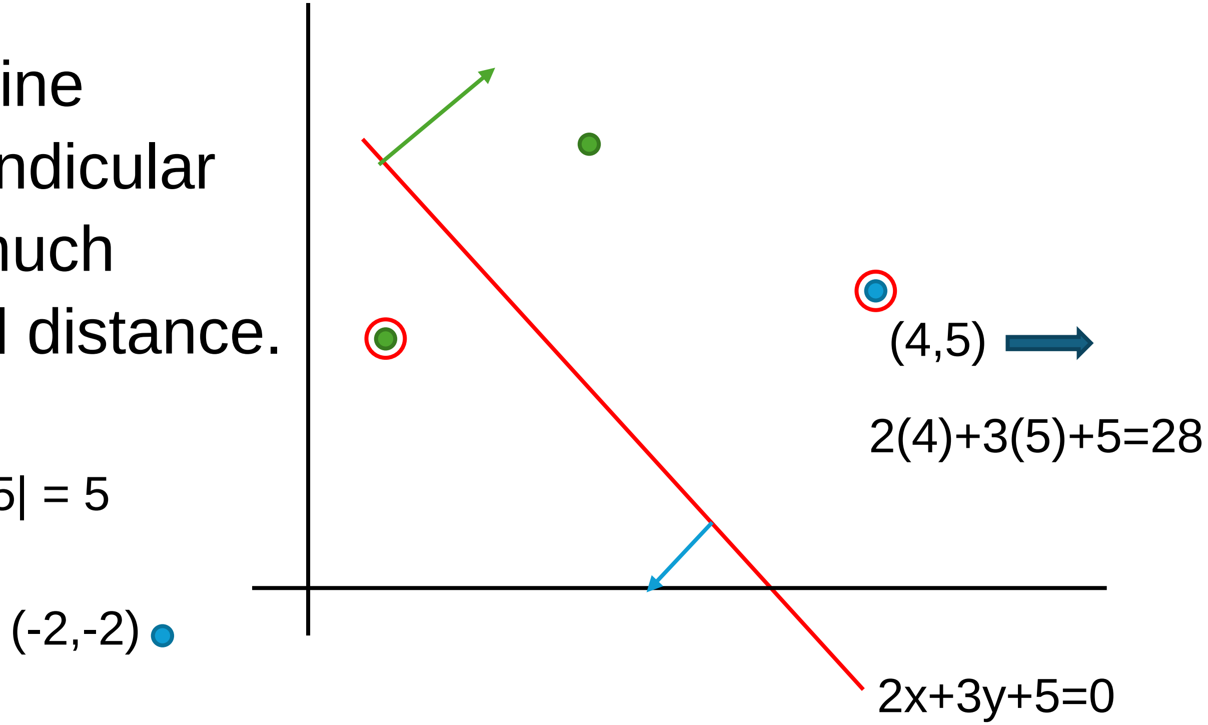


Loss Function

- An **error function** (also called a **loss function**) measures how far off a machine learning or deep learning model's predictions are from the actual target values.
- It gives the model a numeric value that reflects its performance—**lower values mean better predictions**.
- The error function **guides the learning process** by telling the optimizer how to adjust the model’s parameters (like weights in a neural network) during training.
- $f(w_1, w_2, b)$

Perceptron Loss Function

- Number of misclassified points
- (Perpendicular) Distance of the misclassified points
- (In practice)
  - Take the point and put it on the line
  - This is proportional to the perpendicular distance but the mathematics is much simpler than calculating the actual distance.



(4,5)

2(4)+3(5)+5=28

2(-2)+3(-2)+5= |-5| = 5

(-2,-2)

2x+3y+5=0

More Loss Functions

- If activation function is Sigmoid:
  - Loss is Binary cross entropy (used in logistic regression)
  - So when activation function is sigmoid then perceptron is basically logistic regression
- Multi-class classification:
  - Activation: Softmax
  - Loss: Categorical Cross Entropy
- Regression:
  - Activation: Linear (no activation)
  - Loss: MSE

Reference and further reading

1. “Deep Learning”, Ian Goodfellow, et al.
2. Pramoditha, Rukshan. “The Concept of Artificial Neurons (Perceptrons) in Neural Networks.” *Medium*, Towards Data Science, 29 Dec. 2021, towardsdatascience.com/the-concept-of-artificial-neurons-perceptrons-in-neural-networks-fab22249cbfc. Accessed 21 Jan. 2025.