SILVIA LAVAGNINI

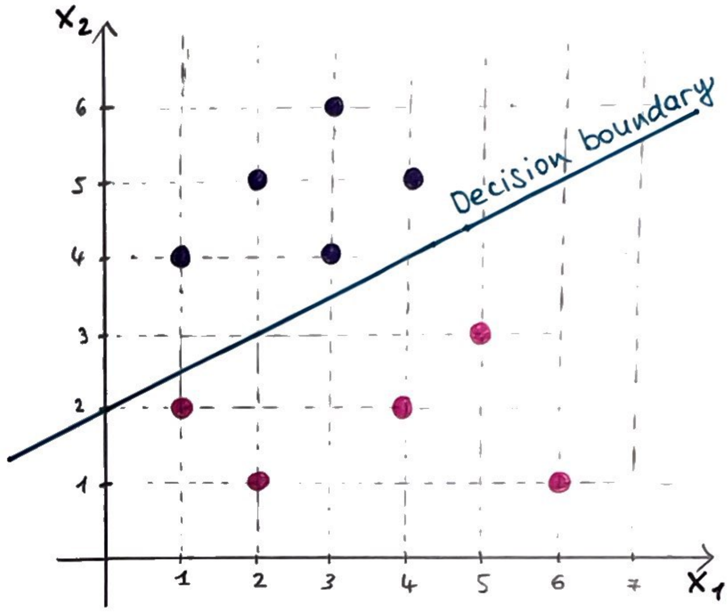
LECTURE 3 : ARTIFICIAL NEURONS FOR CLASSIFICATION

24 FEB 2023

GRA 4150

Let us assume we are given with the following data set:

OBS.	x_1 (Red)	x_2 (Blue)	CLASS (y)
0	1	2	0
1	2	1	0
2	4	2	0
3	5	3	0
4	6	1	0
5	1	4	1
6	2	5	1
7	3	4	1
8	3	6	1
9	4	5	1



- x_1 : amount of red pigment;
- x_2 : amount of blue pigment;
- Class 0 : the observation is classified as "PINK";
- Class 1 : the observation is classified as "PURPLE".

TASK : We want to find the decision boundary to separate pink and purple shades.

FIRST APPROACH : Linear algebra

Assume that:

1. the two classes are separable by a linear boundary (like in the plot above);
2. we know that the linear boundary passes by $(x_1^{(1)}, x_2^{(1)}) = (0,2)$ and $(x_1^{(2)}, x_2^{(2)}) = (2,3)$.

Then we can use the formula

variables!

$$\frac{x_1 - x_1^{(1)}}{x_1^{(2)} - x_1^{(1)}} = \frac{x_2 - x_2^{(1)}}{x_2^{(2)} - x_2^{(1)}} :$$

$$\frac{x_1 - 0}{2 - 0} = \frac{x_2 - 2}{3 - 2} \Rightarrow \frac{x_1}{2} = x_2 - 2 \Rightarrow x_1 - 2x_2 + 4 = 0$$

⚠ There is still one thing to do: I hope you agree that if $x_1 - 2x_2 + 4 = 0$ then also $a \cdot (x_1 - 2x_2 + 4) = 0$ for any possible real number $a \in \mathbb{R}$, so also for $a = -1$.

Hence we have both that

$$x_1 - 2x_2 + 4 = 0$$
$$-x_1 + 2x_2 - 4 = 0$$

I call $z_1 := x_1 - 2x_2 + 4$ and $z_2 := -x_1 + 2x_2 - 4$.

The problem is that we want to write down a decision rule of the following form:

$$\sigma(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{if } z \leq 0 \end{cases}.$$

Obviously, the z_1 and z_2 defined above give two completely opposite decision rules (the pairs (x_1, x_2) that make $z_1 \geq 0$ make obviously $z_2 \leq 0$, and vice-versa). So we need to decide which one is the right one for our specific situation. How? We choose an observation, for example the first one with $(x_1, x_2) = (1, 2)$ and compute z_1 and z_2:

$$z_1 = 1 - 4 + 4 = 1 \geq 0$$
$$z_2 = -1 + 4 - 4 = -1 \leq 0$$

We know that the first observation belongs to the class 0, and accordingly to the decision function σ defined above, we want that for the elements of the class 0 the decision function is equal to 0, hence z must be $z \leq 0$. Hence we choose $z = z_2$.

Finally, we can write down the complete decision rule:

INPUT $\qquad\qquad$ NET INPUT $\qquad\qquad$ DECISION FUNCTION

$$(x_1, x_2) \longmapsto z = -x_1 + 2x_2 - 4 \longmapsto \sigma(z) = \begin{cases} \text{class } 1 & \text{if } z \geq 0 \\ \text{class } 0 & \text{if } z < 0 \end{cases}.$$

we take an observation

we compute the net input

We check the value of the decision function.

We can use the decision rule to distinguish new observations into pink and purple.

- $(x_1, x_2) = (5, 5)$

$$z = -5 + 10 - 4 = +1 \rightarrow \sigma(z) = 1 \rightarrow \text{"PURPLE"}$$

- $(x_1, x_2) = (4, 3)$

$$z = -4 + 6 - 4 = -2 \rightarrow \sigma(z) = 0 \rightarrow \text{"PINK"}$$

GEOMETRICAL INTERPRETATION

The ~~input~~ net input can be rewritten in vector form:

$$z = -x_1 + 2x_2 - 4 = [-1 \quad 2] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - 4 = w^T x + b$$

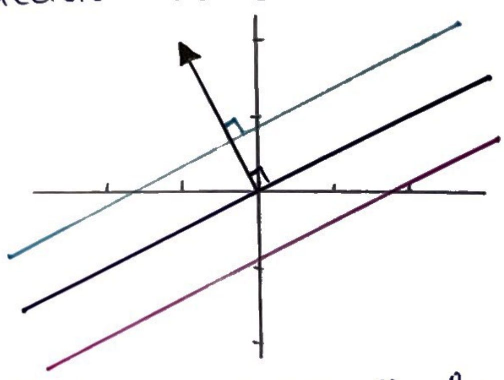
where

$$w := \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} -1 \\ 2 \end{bmatrix} \in \mathbb{R}^{2 \times 1}$$

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathbb{R}^{2 \times 1}$$

$$b = -4 \in \mathbb{R}$$

- What does w represent? It represent a vector perpendicular to the decision boundary. Since we are working in $\mathbb{R}^2$, w is the vector perpendicular to the decision line.



How to draw a vector?

- w_1 corresponds to the x_1- direction;
- w_2 corresponds to the x_2- direction.

However, we have infinitely many lines that are perpendicular to this vector. Which one is ours?

It is the value of b which identifies the right line.

The value of b tells us where our line intersects the axis :

- $x_1 = 0 \rightarrow 2x_2 - 4 = 0 \rightarrow 2x_2 = 4 \rightarrow x_2 = 2$

OR

- $x_2 = 0 \rightarrow -x_1 - 4 = 0 \rightarrow x_1 = -4$

So it is the line passing by $(x_1, x_2) = (0, 2)$ or $(x_1, x_2) = (-4, 0)$ (one of this is enough since we already know the direction.

To conclude :

- w gives us the direction of the boundary ;
- b gives us the location of the boundary.

⚠ There exists **ONLY** one line with a given w and b!

SECOND APPROACH : Perceptron model

Now we remove assumption 2, that means we still know that the two classes are separable by a linear boundary, but we don't know anything about it.

Remember that, in order to identify the boundary, we need to find the values for w and b. Since we are working in dimension 2, we need in practice w_1, w_2, b.

Implementation of the perceptron learning rule

1. Initialize the weights and bias unit to 0 or to small random numbers.
2. For each training example $x^{(i)}$ $(i = 0, 1, \dots, 9)$
   a) Compute the output value $\hat{y}^{(i)}$;
   b) Update the parameters accordingly to the rule :

$$\boxed{\begin{cases} w_1 = w_1 + \Delta w_1 \\ w_2 = w_2 + \Delta w_2 \\ b = b + \Delta b \end{cases} \quad \text{where} \quad \begin{cases} \Delta w_1 = \eta\,(y^{(i)} - \hat{y}^{(i)})\,x_1^{(i)} \\ \Delta w_2 = \eta\,(y^{(i)} - \hat{y}^{(i)})\,x_2^{(i)} \\ \Delta b = \eta\,(y^{(i)} - \hat{y}^{(i)}) \end{cases}}$$

Here

- η is the learning rate, it tells "how big" I want the step to be;
- $y^{(i)}$ are the observations, so for each $i = 0, 1, \dots, 9$ $y^{(i)}$ is either 0 or 1 (last column of the dataset);
- $\hat{y}^{(i)}$ is the class predicted by the model, given as input the i-th observation $x^{(i)} = (x_1^{(i)}, x_2^{(i)})$.

Since it is a binary problem (we predict either "class 0" or "Class 1") we have two possibility: either we predict the right class or we predict the wrong class:

I) We predict the <u>right class</u>: <u>NO WEIGHTS UPDATE</u>

(i) If $y^{(i)} = 0$ and we predict $\hat{y}^{(i)} = 0$:

$$\Delta w_1 = \eta \cdot (0-0) \cdot x_1^{(i)} = 0, \quad w_1 = w_1$$
$$\Delta w_2 = \eta \cdot (0-0) \cdot x_2^{(i)} = 0, \quad w_2 = w_2$$
$$\Delta b = \eta \cdot (0-0) = 0, \quad b = b$$

(ii) If $y^{(i)} = 1$ and we predict $\hat{y}^{(i)} = 1$:

$$\Delta w_1 = \eta \cdot (1-1) \cdot x_1^{(i)} = 0, \quad w_1 = w_1$$
$$\Delta w_2 = \eta \cdot (1-1) \cdot x_2^{(i)} = 0, \quad w_2 = w_2$$
$$\Delta b = \eta \cdot (1-1) = 0, \quad b = b$$

II) We predict the <u>wrong class</u>: <u>YES WEIGHTS UPDATE</u>

(i) If $y^{(i)} = 0$ and we predict $\hat{y}^{(i)} = 1$:

$$\Delta w_1 = \eta (0-1) x_1^{(i)} = -\eta x_1^{(i)}$$
$$\Delta w_2 = \eta (0-1) x_2^{(i)} = -\eta x_2^{(i)}$$
$$\Delta b = \eta (0-1) = -\eta$$

(ii) If $y^{(i)} = 1$ and we predict $\hat{y}^{(i)} = 0$:

$$\Delta w_1 = \eta (1-0) x_1^{(i)} = \eta x_1^{(i)}$$
$$\Delta w_2 = \eta (1-0) x_2^{(i)} = \eta x_2^{(i)}$$
$$\Delta b = \eta (1-0) = \eta$$

- The weights update is proportional to $x_1^{(i)}$ for w_1 and proportional to $x_2^{(i)}$ for w_2;
- Bias and weights are updated all simultaneously!