TD 8 – Learning II Supervised Learning

1 Perceptron model

① *Total input*
The total input received by the neuron is the weighted sum over all input neurons, which can be written as a scalar product :

$$\sum_j I_{p,j} W_j = \vec{I}_p \cdot \vec{W}$$

② *Condition for solving the task*
The task is solved if the perceptron is able to categorize each input pattern correctly. Each output values r_p requires a relation between the input and the threshold as follows :

$$\forall\, p, \quad \begin{cases} r_p = 1 \implies & \vec{I}_p \cdot \vec{W} > \theta \\ r_p = 0 \implies & \vec{I}_p \cdot \vec{W} < \theta \end{cases}$$

③ *Rewriting the condition*
In order to simplify the condition, the threshold can be modeled as a constant input. To do so, the vectors $\vec{I}_p$ and $\vec{W}$ are appended with a first entry $I_0 = 1$ and $W_0 = -\theta$ respectively, such that :

$$\sum_{j=1}^{N} I_{p,j} W_j > \theta \iff \sum_{j=1}^{N} I_{p,j} W_j - \theta \times 1 > 0 \iff \sum_{j=0}^{N} I_{p,j} W_j > 0$$

The condition becomes :

$$\forall\, p, \quad \begin{cases} r_p = 1 \implies & \vec{I}_p \cdot \vec{W} > 0 \\ r_p = 0 \implies & \vec{I}_p \cdot \vec{W} < 0 \end{cases}$$

④ *Set of input patterns $\vec{J}_p$*
To reduce the condition to a single equation whose result is always positive, the dot product $\vec{I}_p \cdot \vec{W}$ can be multiplied by a scalar of the same sign, expressed as a function of r_p. The goal is to get a variable s_p such that :

$$\forall\, p, \quad \begin{cases} r_p = 1 \implies s_p = 1 \\ r_p = 0 \implies s_p = -1 \end{cases}$$

This can be obtained by introducing $s_p = 2r_p - 1$. Then :

$$\forall\, p, \quad s_p\, \vec{I}_p \cdot \vec{W} > 0$$

Equivalently, this can be obtained by introducing the vectors $\vec{J}_p = (2r_p - 1)\vec{I}_p$.

$$s_p \vec{I}_p \cdot \vec{W} > 0$$

2 Perceptron algorithm

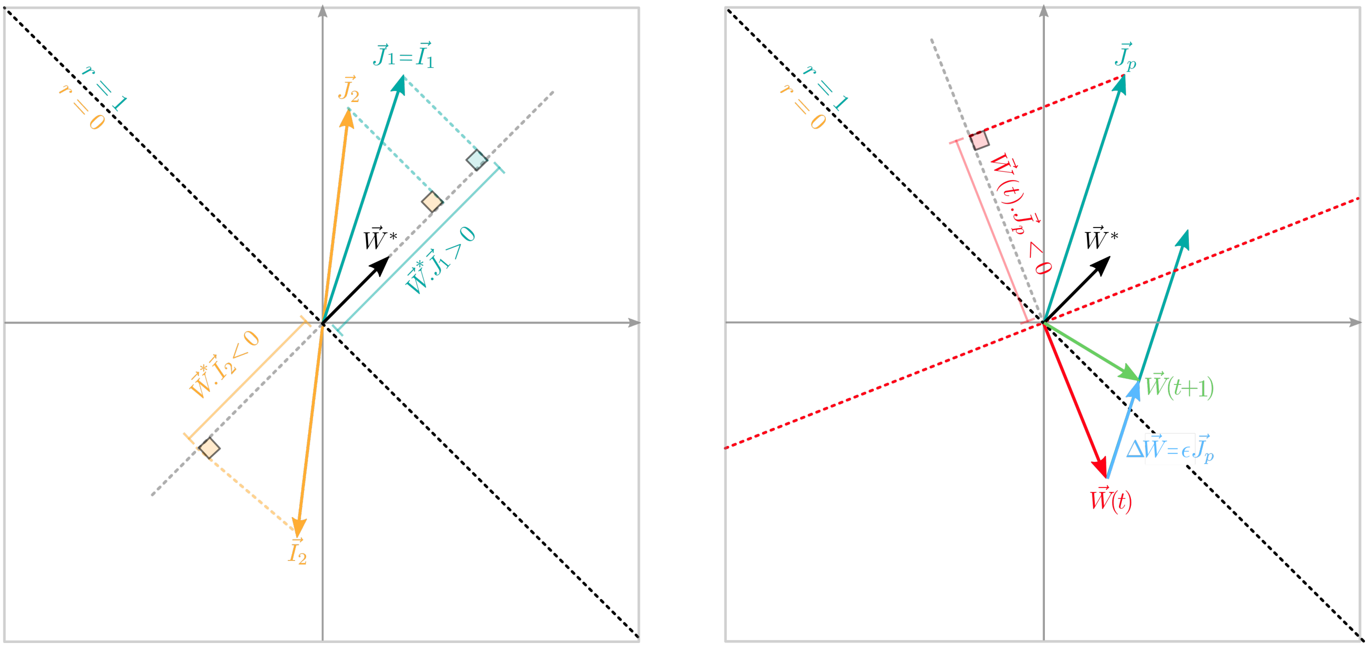
⑤ *Supervised learning*
The correct output is provided by an external "teaching" signal which drives learning in the right direction. The information about the correct output is contained in the vector $\vec{J}_p$ through the appearance of r_p in its expression.

Generalization
Generalization is the ability to correctly classify a new input (i.e. not previously seen) after the network has been trained.

⑥ Graphical representation

Left : The classification line (black dots) splits the plane between the two categories (classes $r = 0$, $r = 1$). One solution for this classification problem is the vector $\vec{W}^*$ which is exactly orthogonal to the classification frontier. Two input patterns $\vec{I}_1$ (class 1) and $\vec{I}_2$ (class 0) are represented : their projections along the vector $\vec{W}^*$ have a sign which reflects the class to with they belong. Their corresponding transformations $\vec{J}_1$ and $\vec{J}_2$ are also represented : their projections onto the vector $\vec{W}^*$ are all positive, indicating a correct classification.

Right : One learning step. The current weight vector $\vec{W}(t)$ is not a solution of the classification problem, since it does not classify correctly the vector $\vec{J}_p$. Indeed, its projection onto $\vec{W}(t)$ is negative, which imposes to modify the vector $\vec{W}(t)$. The modification $\Delta = \epsilon \vec{J}_p$ is aligned with the vector $\vec{J}_p$ (i.e. parallel to it), which brings the new vector $\vec{W}(t+1)$ closer to the solution $\vec{W}^*$.



⑦ *Lower bound*
The goal is to express the *numerator* of the cosine between one solution $\vec{W}^*$ vector $\vec{W(t)}$ at a given learning step t. This can be done by recurrence, which requires to express $\vec{W}(t+1) \cdot \vec{W}^*$ as a function of $\vec{W}(t) \cdot \vec{W}^*$:

$$\begin{aligned} \vec{W}(t+1) \cdot \vec{W}^* &= (\vec{W}(t) + \epsilon \vec{J}_p) \cdot \vec{W}^* \\ &= \vec{W}(t) \cdot \vec{W}^* + \epsilon \vec{J}_p \cdot \vec{W}^* \\ &\geq \vec{W}(t) \cdot \vec{W}^* + \epsilon l \end{aligned}$$

This is an arithmetic sequence, which leads by recurrence to :

$$\vec{W}(t) \cdot \vec{W}^* \geq \vec{W}(0) \cdot \vec{W}^* + \epsilon l t = \epsilon l t$$

with $\vec{W}(0) = \vec{0}$.

8 *Upper bound*
The goal is to express the *denominator* of the cosine between one solution $\vec{W}^*$ vector $\vec{W(t)}$ at a given learning step t. This can be done by recurrence, which requires to express $||\vec{W}(t+1)||$ as a function of $||\vec{W}(t)||$:

$$\begin{aligned}\|\vec{W}(t+1)\|^2 &= \|\vec{W}(t) + \epsilon\vec{J}_p\|^2 \\ &= \|\vec{W}(t)\|^2 + 2\epsilon\vec{J}_p\vec{W} + \epsilon^2\|\vec{J}_p\|^2 \\ &\leq \|\vec{W}(t)\|^2 + \epsilon^2 L\end{aligned}$$

Similarly :

$$\|\vec{W}(t)\|^2 \leq \|\vec{W}(0)\|^2 + t\epsilon^2 L = t\epsilon^2 L$$

9 *Lower bound on* $\cos[\alpha(t)]$
Gathering numerator (question 7) and denominator (question 8) leads to

$$\cos(\alpha(t)) \geq \frac{t\epsilon l}{\sqrt{t\epsilon^2 L}} = \sqrt{t}\frac{l}{L}$$

10 *Solution*
The result obtained at question 9 holds *each time a learning step is performed*. In this case, as $\cos(\alpha(t)) < 1$, it imposes that $t < \frac{L}{l}^2$. Thus, the algorithm can only perform a finite number of steps, which means that the algorithm necessarily stops after at most $\frac{L}{l}^2$ steps. When the algorithm stops, all patterns are correctly classified.