M2 Biology **26/11/20**

TD7: Learning I

Elie Oriol

`https://github.com/Elieoriol/2021_UlmM2_ThNeuro/tree/master/TD7`

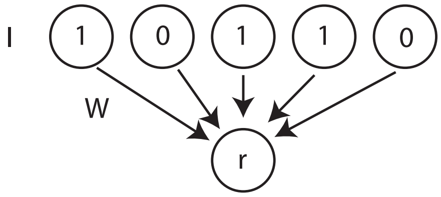
Several learning paradigms exist: mainly supervised, unsupervised and reinforcement learning. In this series of tutorial, we will develop standard models in each of these 3 paradigms, starting with the supervised perceptron. The perceptron model originates with Rosenblatt in 1958 and marks the beginning of what we know today as Artificial Neural Networks, which have in recent years attained unexpectedly high performance in various tasks, placing learning as a central question of Neuroscience.

1 Perceptron model

We consider a neuron receiving input from N other neurons. Each neuron can either be active or not. The input received by the neuron at a given time can therefore be represented by a vector $\vec{I}$ of zeros and ones where I_n is the activity of neuron n.

The synapses connecting the input neurons and the output neuron may have different strengths. They are represented by a vector of weights $\vec{W}$ where W_n is the strength of the synapse connecting neuron n to the output neuron.

The output neuron is active if the total input it receives is larger than a threshold θ.



The output neuron has the task of associating a set of P input patterns $\vec{I_p}$ with a desired output r_p. The parameters $\vec{W}$ and θ may be adjusted so as to solve this task.

1. Give an expression of the total input received by the output neuron when a given pattern $\vec{I_p}$ is presented.

   We sum over each weights:

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

2. Write a condition on the input patterns $\vec{I_p}$, the desired outputs r_p and the parameters $\vec{W}$ and θ such that the task is solved.

   The task is solved if the perceptron is able to categorize each input correctly; for all p, if $r_p = 1$ then $\vec{I_p} \cdot \vec{W} > \theta$; if $r_p = 0$ then $\vec{I_p} \cdot \vec{W} < \theta$.

3. For convenience, consider an imaginary input $I_0 = 1$ which is always turned on and rewrite the condition as a function of $(I_0, \vec{I}_p)$ and $(W_0, \vec{W})$ where $W_0 = -\theta$

   The condition becomes that for all p, if $r_p = 1$ then $\vec{I}_p \cdot \vec{W} > 0$; if $r_p = 0$ then $\vec{I}_p \cdot \vec{W} < 0$.

4. Find a set of input patterns $\vec{J}_p$ such that the condition is equivalent to: for all p, $\vec{J}_p \cdot \vec{W} > 0$

   We can more conveniently define $s_p = 2r_p - 1$. Then for all p, if $s_p = 1$ then $\vec{I}_p \cdot \vec{W} > 0$; if $s_p = -1$ then $\vec{I}_p \cdot \vec{W} < 0$. In other words:

   $$s_p \ \vec{I}_p \cdot \vec{W} > 0 \quad \Rightarrow \quad \vec{J}_p = (2r_p - 1)\vec{I}_p$$

5. What does it mean to have a network which is able to generalize after learning?

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

2 Perceptron algorithm

Now that we have better characterized the inputs to the perceptron, we need to train it in order to correctly associate inputs to their respective output. To do so, we use the perceptron algorithm:

- Randomly pick an input pattern $\vec{J}_p$
- If $\vec{J}_p \cdot \vec{W} > 0$, pick a new input pattern.
- If $\vec{J}_p \cdot \vec{W} < 0$, perform a learning step: $\vec{W}(t+1) = \vec{W}(t) + \epsilon \vec{J}_p$

6. Explain why this algorithm is an implementation of "supervised" learning.

   The correct output is provided by an external "teaching" signal which is used to drive learning in the right direction.

We will show that if there exists a solution $\vec{W}^*$, then this algorithm necessarily finds a solution. For this we consider:

$$\cos[\alpha(t)] = \frac{\vec{W}(t) \cdot \vec{W}^*}{\|\vec{W}(t)\| \|\vec{W}^*\|} \tag{1}$$

with $\alpha(t)$ the angle between $\vec{W}(t)$ and $\vec{W}^*$.

7. Introduce $l = \min_p \vec{J}_p \cdot \vec{W}^* > 0$ and find a lower bound on $\vec{W}(t+1) \cdot \vec{W}^*$ given $\vec{W}(t) \cdot \vec{W}^*$. Considering $\vec{W}(0) = 0$ deduce a lower bound on $\vec{W}(t) \cdot \vec{W}^*$.

   $$\begin{aligned} \vec{W}(t+1) \cdot \vec{W}^* = \vec{W}(t) \cdot \vec{W}^* + \epsilon \vec{J}_p \cdot \vec{W}^* &\geq \vec{W}(t) \cdot \vec{W}^* + \epsilon l \\ \vec{W}(t) \cdot \vec{W}^* &\geq \vec{W}(0) \cdot \vec{W}^* + \epsilon l t = \epsilon l t \end{aligned}$$

8. Introduce $L = \max_p \|\vec{J}_p\|^2$ and find an upper bound on $\|\vec{W}(t+1)\|^2$ given $\|\vec{W}(t)\|^2$. Deduce an upper bound on $\|\vec{W}(t)\|^2$.

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

9. Find a lower bound on $\cos[\alpha(t)]$.

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

10. Explain why the algorithm necessarily finds a solution.

    $\cos[\alpha(t)] < 1$ *therefore* $t < \frac{L}{l}^2$*; the algorithm can only perform a finite number of steps. This means that the algorithm stops after at most* $\frac{L}{l}^2$ *steps. When the algorithm stops, all patterns are correctly classified.*