The Perceptron Algorithm

Perceptron (Frank Rosenblatt, 1957)

• First learning algorithm for neural networks;

• Originally introduced for character classification, where each character is represented as an image;
Perceptron (contd.)

Output unit performs the function: (activation function):

\[
H(x) = \begin{cases} 
1 & \text{if } x \geq 0 \\
0 & \text{if } x < 0 
\end{cases}
\]

Total input to output node:

\[
\sum_{j=1}^{n} w_j x_j
\]

Perceptron: Learning Algorithm

• **Goal**: we want to define a learning algorithm for the weights in order to compute a mapping from the inputs to the outputs;

• **Example**: two class character recognition problem.
  – **Training set**: set of images representing either the character 'a' or the character 'b' (supervised learning);
  – **Learning Task**: Learn the weights so that when a new unlabelled image comes in, the network can predict its label.

• **Settings**:
  - Class 'a' $\mapsto 1$ (class C1)
  - Class 'b' $\mapsto 0$ (class C2)
  - $n$ input units (intensity level of a pixel)
  - 1 output unit

The perceptron needs to learn $f : \mathbb{R}^n \rightarrow \{0,1\}$
Perceptron: Learning Algorithm

The algorithm proceeds as follows:

- Initial random setting of weights;
- The input is a random sequence \( \{x_k\}_{k \in \mathbb{N}} \);
- For each element of class C1, if output = 1 (correct) do nothing, otherwise update weights;
- For each element of class C2, if output = 0 (correct) do nothing, otherwise update weights.

A bit more formally:

\[ x = (x_1, x_2, ..., x_n) \quad w = (w_1, w_2, ..., w_n) \]

\( \theta \) : Threshold of the output unit

\[ wx^T = w_1x_1 + w_2x_2 + ... + w_nx_n \]

Output is 1 if \( wx^T - \theta \geq 0 \)

To eliminate the explicit dependence on \( \theta \):

Output is 1 if:

\[ \hat{w}\hat{x}^T = \sum_{i=1}^{n+1} \hat{w}_i\hat{x}_i \geq 0 \]

\( x_0 = 1 \)
Perceptron: Learning Algorithm

• We want to learn values of the weights so that the perceptron correctly discriminate elements of $C_1$ from elements of $C_2$:

• Given $x$ in input, if $x$ is classified correctly, weights are unchanged, otherwise:

$$w' = \begin{cases} w + x & \text{if an element of class } C_1 (1) \text{ was classified as in } C_2 \\ w - x & \text{if an element of class } C_2 (0) \text{ was classified as in } C_1 \end{cases}$$

Perceptron: Learning Algorithm

• 1st case: $x \in C_1$ and was classified in $C_2$

The correct answer is 1, which corresponds to: $\hat{w} \hat{x}^T \geq 0$

We have instead: $\hat{w} \hat{x}^T < 0$

We want to get closer to the correct answer: $w x^T < w' x^T$

$$w x^T < w' x'^T \iff w x^T < (w + x) x^T$$

$$(w + x) x^T = wx^T + xx^T = wx^T + \|x\|^2$$

because $\|x\|^2 \geq 0$, the condition is verified
Perceptron: Learning Algorithm

- In summary:

1. A random sequence \(x_1, x_2, \ldots, x_k, \ldots\) is generated such that \(x_j \in C_1 \cup C_2\)

2. If \(x_k\) is correctly classified, then \(w_{k+1} = w_k\)
   otherwise

\[
w_{k+1} = \begin{cases} w_k + x_k & \text{if } x_k \in C_1 \\ w_k - x_k & \text{if } x_k \in C_2 \end{cases}
\]
Perceptron: Learning Algorithm

Does the learning algorithm converge?

Convergence theorem: Regardless of the initial choice of weights, if the two classes are linearly separable, i.e. there exist \( \mathbf{w} \) s.t.

\[
\begin{align*}
\hat{w} \mathbf{x}^T & \geq 0 \text{ if } \mathbf{x} \in C_1 \\
\hat{w} \mathbf{x}^T & < 0 \text{ if } \mathbf{x} \in C_2
\end{align*}
\]

then the learning rule will find such solution after a finite number of steps.

Representational Power of Perceptrons

- Marvin Minsky and Seymour Papert, “Perceptrons” 1969:
  “The perceptron can solve only problems with linearly separable classes.”
- Examples of linearly separable Boolean functions:

AND

OR
Representational Power of Perceptrons

Perceptron that computes the AND function

Perceptron that computes the OR function

Representational Power of Perceptrons

• Example of a non linearly separable Boolean function:

EX-OR

The EX-OR function cannot be computed by a perceptron