Physics 171/271

1 Feb 2011 DK
corrected

Preliminary Notes on Perceptrons

This is the study of feedforward networks, as found at the "front end" of sensory systems. Roughly,

$$y = f\left(\sum_{j=1}^{N \leftarrow \#\text{ of units}} w_j \, x_j - b\right)$$

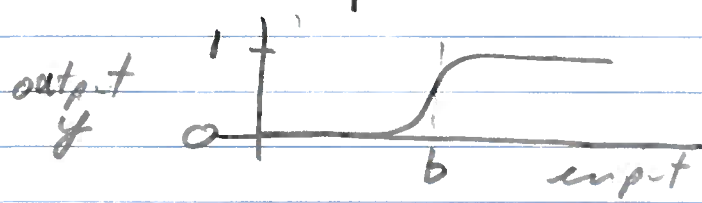
y ↑ output

f ↑ (see plot below)

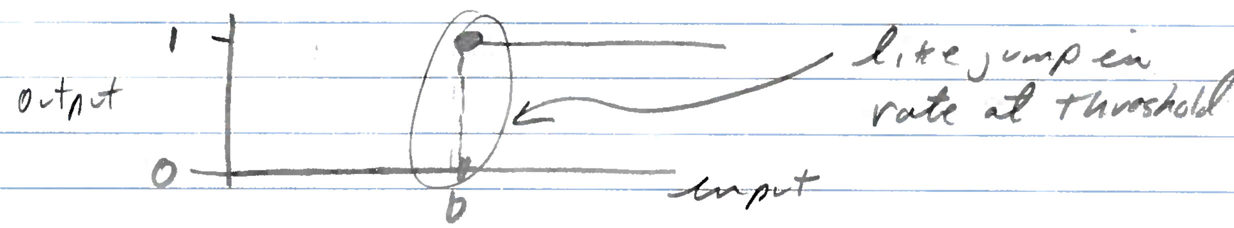
w_j ↑ synaptic weights

x_j ↑ inputs

b ↑ threshold or bias



Consider case of step function, which is not so far from type 2 neurons or Hopf bifurcation.

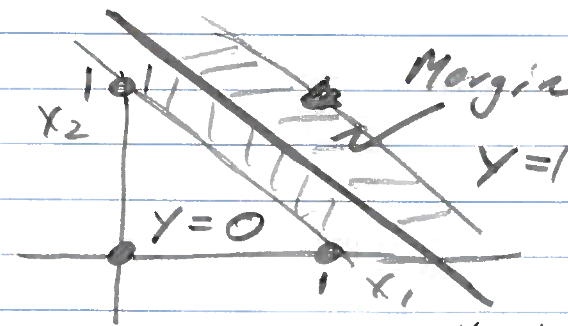


McCulloch & Pitts considered mapping Boolean functions onto neurons in this way. Take the case of "AND"

X_1	X_2	y
0	0	0
1	0	0
0	1	0
1	1	1

The Perceptron has two cells, so
$w_1 x_1 + w_2 x_2 - b = y$ implies:

$0 < b$
$w_1 < b$
$w_2 < b$
$w_1 + w_2 > b$



$w_1 = 1$
$w_2 = 1$
$b = 3/2$

OR function → just move dividing line
so $b = 1/2$
NOR function → cannot be realized

In general, line is $\sum w_j x_j = b$, or
$\vec{w} \cdot \vec{x} = b$, a hyperplane.

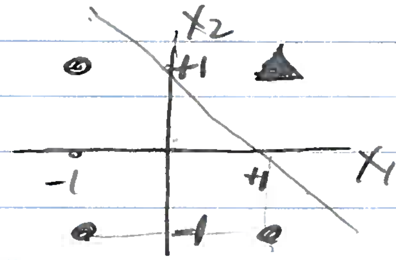
Perceptron Learning (Minimize $\langle \{-y \vec{w}^T \cdot \vec{x}\}^+ \rangle$)

For binary input-output, we can have
an update rule
$$\vec{w} = \vec{w} + y\,\vec{x}$$

Start with $\vec{w} = (0,0)$ — Convergence in one step!
Put on $\vec{x} = (1,1)$
$$\vec{w} = (0,0) + 1(1,1) = (1,1)$$
Other values for $\vec{x}$ do not contribute

$$b = b + y\,w^T x - \langle y \rangle$$
$$\text{"} = 0 + 1\,(1,1)\begin{pmatrix}1\\1\end{pmatrix} - 1/2$$
$$\text{"} = 3/2$$

All of this becomes a bit easier if we switch to $\left\{ \begin{matrix} y = \pm 1 \\ x = \pm 1 \end{matrix} \right\}$ notation



$$y = \frac{1}{N} \vec{\omega} \cdot \vec{x} = \frac{1}{2} [w_1 x_1 + w_2 x_2]$$

Consider n sets of Boolean functions

We also map threshold onto input

$$\text{Input}(n) \equiv [-1, x_1(n), x_2(n), \ldots, x_N(n)]$$

$$\text{Output}(n) = y(n) = \pm 1$$

Training constitutes the learning of $\left\{ \vec{x}(n), y(n) \right\}$ sets

Class 1 : $w^T(n)\, x(n) \geq 0$

Class 2 : $w^T(n)\, x(n) \leq 0$

Calculate $\underbrace{\hat{y}(n)}_{\text{calculated output}} \equiv \underbrace{\vec{w}^T(n)}_{\text{current weights}}\, \underbrace{x(n)}_{\text{from trainingset}}$

Update Rule

$$\underbrace{\vec{w}(n+1)}_{\text{weights on } n^{th} \text{ iteration}} = \vec{w}(n) + \frac{1}{2}\Big[\underbrace{y(n)}_{\pm 1} - \underbrace{\vec{w}^T(n) \cdot \vec{x}(n)}_{\pm 1}\Big]\, x(n)$$

in response to n^{th} $(y, \vec{x})$ pair

$\therefore$ $\hat{y}(n) = y(n)$ implies $\vec{w}(n+1) = \vec{w}(n)$

$\hat{y}(n) \neq y(n)$ implies $\begin{cases} \vec{w}(n+1) = \vec{w}(n) + \vec{x}(n) & \text{Class 1} \\ \vec{w}(n+1) = \vec{w}(n) - \vec{x}(n) & \text{Class 2} \end{cases}$

where we have two training sets

Set w/ class 1 $(y(k)\ \vec{x}(k))$ $y(k) = 1$ $\forall k$

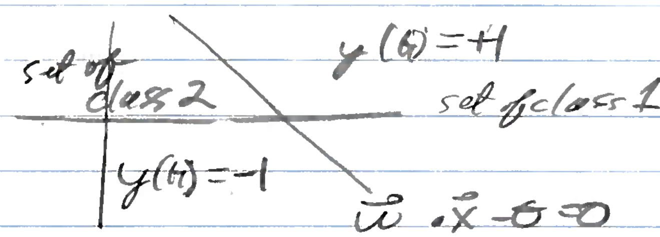
Set of class 2 $(y(k)\ \vec{x}(k))$ $y(k) = -1$ $\forall k$

Note $\vec{w} = [b, w_1, w_2, \ldots]$

$\vec{x} = [-1, x_1, x_2, \ldots]$

$\vec{w} \cdot \vec{x} \equiv \vec{w}^T \vec{x} = w_1 x_1 + w_2 x_2 + \cdots - b$

Categorization by Perceptron



Correct categorization:

$\vec{w}(n+1) = \vec{w}(n)$

Wrong categorization:

Learning makes weights an average over (+1) examples

$\vec{w}(n+1) = \vec{w}(n) + y(n)\, x(n)$

$\qquad '' \quad = \begin{cases} \vec{w}(n) + \vec{x}(n) & \text{Set 1:} \\ \vec{w}(n) - \vec{x}(n) & \text{Set 2:} \end{cases}$

Convergence theorem.

Need to show that correction grows faster than error. Do this in 2 steps

1) Consider correction first. Spse we make n errors, leading to n updates

That is, $\vec{w}^T(n)\vec{x}(n) < 0$ for set of class 1 yet $y(n) = +1$

$$\therefore \vec{w}(n+1) = \vec{w}(n) + X(n)$$

$$\uparrow \; \vec{w}(n) = \vec{w}(n-1) + \vec{x}(n)$$

$\uparrow$ interate down to $\vec{x}(1)$

$$'' \quad = \sum_{k=1}^{n} \vec{x}(k) + \cancel{\vec{w}(0)}^{\,0}$$

$\uparrow$ no loss to take $\vec{w}(0) = 0$

Consider a potential class 1 solution denoted by $\vec{w}_1$

Then
$$\vec{w}_1^T \, \vec{w}(n+1) = \sum_{n=1}^{n} \underbrace{\vec{w}_1^T \vec{x}(k)}$$

$$\alpha \equiv \min_{\vec{x}(n) \in \text{set } 1} \vec{w}_1^T \vec{x}(n)$$

$$'' \quad \geq n\alpha$$

Cauchy-Schwartz $\therefore \|\vec{w}_1^T \vec{w}(n+1)\|^2 \geq (n\alpha)^2$

but $\|\vec{w}_1\|^2 \|\vec{w}(n+1)\|^2 \geq \|\vec{w}_1^T \vec{w}(n+1)\|^2$

$$\therefore \|\vec{w}_1\|^2 \|\vec{w}(n+1)\|^2 \geq (n\alpha)^2$$

$$\|\vec{w}(n+1)\|^2 \geq \frac{\alpha^2}{\|w_1\|^2} n^2$$

$\therefore$ weight correction scales as square of the number of iterations.

2) Now look at growth of error

As above, the change in weight upon update for $k=1,\ldots n$ is

$$\vec{w}(k+1) = \vec{w}(k) + \vec{x}(k)$$

$$\therefore \|\vec{w}(k+1)\|^2 = \|\vec{w}(k) + \vec{x}(k)\|^2$$

$$= \|\vec{w}(k)\|^2 + \|\vec{x}(k)\|^2 + \underbrace{2w^T(k)\, x(k)}_{<0 \text{ by definition}}$$

$$\therefore \|\vec{w}(k+1)\|^2 \leq \|\vec{w}(k)\|^2 + \|\vec{x}(k)\|^2$$

$$\therefore \|\vec{w}(k+1)\|^2 - \|\vec{w}(k)\|^2 \leq \|\vec{x}(k)\|^2$$

Now iterate from $k=n$ downward

$$\|\vec{w}(n+1)\|^2 - \|\vec{w}(n)\|^2 \leq \|\vec{x}(n)\|^2$$

$$\Sigma \quad \|\vec{w}(n)\|^2 - \|\vec{w}(n-1)\|^2 \leq \|\vec{x}(n-1)\|^2$$

$$\|\vec{w}(n+1)\|^2 - \|\vec{w}(\cancel{0})\|^2 \leq \sum_{k=1}^{n} \|x(k)\|^2$$

0

$$\| \vec{w}(n+1) \|^2 \le \underbrace{\sum_{k=1}^{n} \| x(k) \|^2}$$

$\beta \equiv \max_{\vec{x}(n) \in \text{set } 1} \| x(k) \|^2$

= example w/ largest norm

" $\le \beta n$ ⟵ linear growth of error

We now have two constraints.

(1) gave $\| \vec{w}(n+1) \|^2 \geq \frac{\alpha^2}{\| w_1^T \|^2} n^2$

(2) gave $\| \vec{w}(n+1) \|^2 \le \beta n$

These are equal, i.e., correction beats error, for a finite value of n.

$$\frac{\alpha^2}{\| w_1 \|^2} n^2 = \beta n$$

or $n = \frac{\beta}{\alpha^2} \| w_1 \|^2$ = some number

↑

minimum learning steps

Fini! Even if the example didn't allow the $\vec{x}$ space to be split exactly with a hyperplane, the Perceptron will converge to that solution.