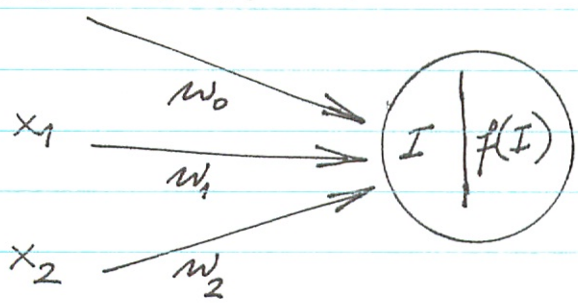
Train a single PE

bias

omitting PE index

$x_0 = \text{const} = 1$



$\longrightarrow y = f(I)$

$$I = \sum_{j=0}^{2} w_j x_j$$

Transfer function :

$$y = f(I) = \begin{cases} 1 & I > 0 \\ -1 & I \le 0 \end{cases}$$

Learning rule:

learning parameter

$$w_j(t+1) = w_j(t) + \alpha\left(y^{target} - y\right) \cdot x_j$$

Train this PE to learn 'OR'

k (pattern #)	x_0	x_1	x_2	x_1 OR x_2 (y^{target})
	input pattern			
1	1	1	1	1
2	1	1	−1	1
3	1	−1	1	1
4	1	−1	−1	−1

truth table

1: true

−1: false

Training procedure:

1) Initialize the weights, set $t=0$, set $\alpha = \text{const.}$
2) Select one input pattern randomly
3) Compute y
4) Apply the learning rule to update the weights.

Repeat ~~until~~ 2)-4) until the output (y) is correct for all input patterns.

---

1) Initialize (set) weights at $t=0$ to

$w_0(0) = 0.1 \qquad w_1(0) = 0.2 \qquad w_2(0) = 0.5$

(pretend we picked random values, in $[-1, 1]$ for example)

Set the learning rate, $\alpha = 0.5$

(We will learn later how to choose good values.)

Now do the training cycle 2)-4)

TOPIC:

DATE:

FILE UNDER: For example

p#2 $I(1) = 0.1 + 0.2 - 0.5 < 0$ $y = -1 \neq y^{target}$

$(1, 1, -1; 1)$ $(y^{target} - y) \cdot \alpha = (1 - (-1)) * 0.5 = 1$

$w_0(1) = 0.1 + 1 \cdot 1 = 1.1$

$w_1(1) = 0.2 + 1 \cdot 1 = 1.2$

$w_2(1) = 0.5 + 1 \cdot (-1) = -0.5$

p#1

$(1, 1, 1; 1)$ $I(2) = 1.1 + 1.2 - 0.5 > 0$ $y = 1 = y^{target}$

no change, $w_j(2) = w_j(1)$

p#3

$(1, -1, 1; 1)$ $I(3) = 1.1 - 1.2 - 0.5 < 0$ $y = -1 \neq y^{target}$

$(y^{target} - y) \cdot \alpha = (1 - (-1)) \cdot 0.5 = 1$

$w_0(3) = 1.1 + 1 * 1 = 2.1$

$w_1(3) = 1.2 + 1 * (-1) = 0.2$

$w_2(3) = -0.5 + 1 * (1) = 0.5$

p#4

$(1, -1, -1; -1)$ $I(4) = 2.1 - 0.2 - 0.5 > 0$ $y = 1 \neq y^{target}$

$(y^{target} - y) \alpha = -1$

$w_0(4) = 2.1 + (-1) \cdot 1 = 1.1$

$w_1(4) = 0.2 + (-1) \cdot (-1) = 1.2$

$w_2(4) = 0.5 + (-1)(-1) = 1.5$

p#1 $I(5) = 1.1 + 1.2 + 1.5 > 0$ $y = 1 = y^{target}$

no change

p#2 $I(6) = 1.1 + 1.2 - 1.5 > 0$ $y = 1 = y^{target}$

no change

p#3 $I(7) = 1.1 - 1.2 + 1.5 > 0$ $y = 1 = y^{target}$

Lucky ... fast converged.

But try picking the patterns in any order — you will succeed in a few steps.

Observe:

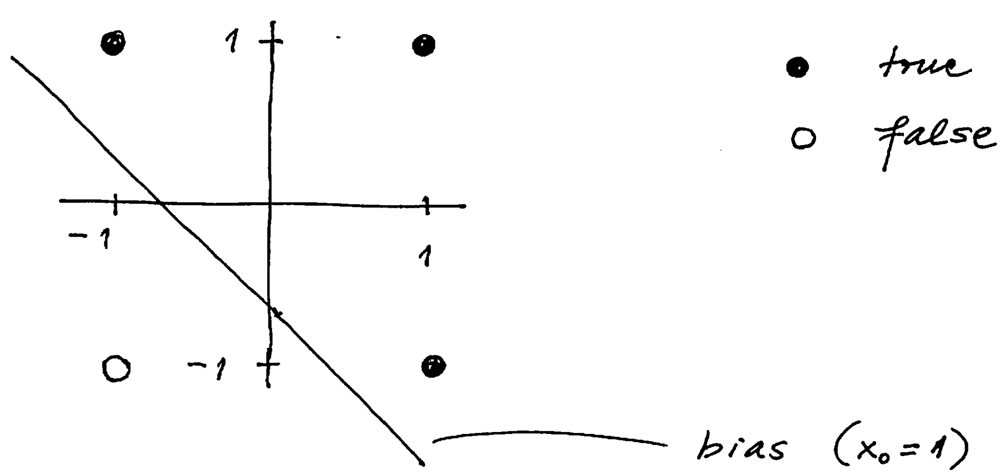
1)
A PE produces one number (a scalar) as output, at any given time step. This is the level of activation, the PE's response to the input pattern (stimulus) at that time step. Depends on the weights!

2)
A PE stores a set of weights (that change during training but are fixed once the learning finished):

$$y = \sum_{j=0}^{n} w_j x_j = w_0 \underbrace{x_0}_{=1} + w_1 x_1 + \ldots + w_n x_n$$

n-d line (hyperplane), separates two halves of the n-d space.

'OR'

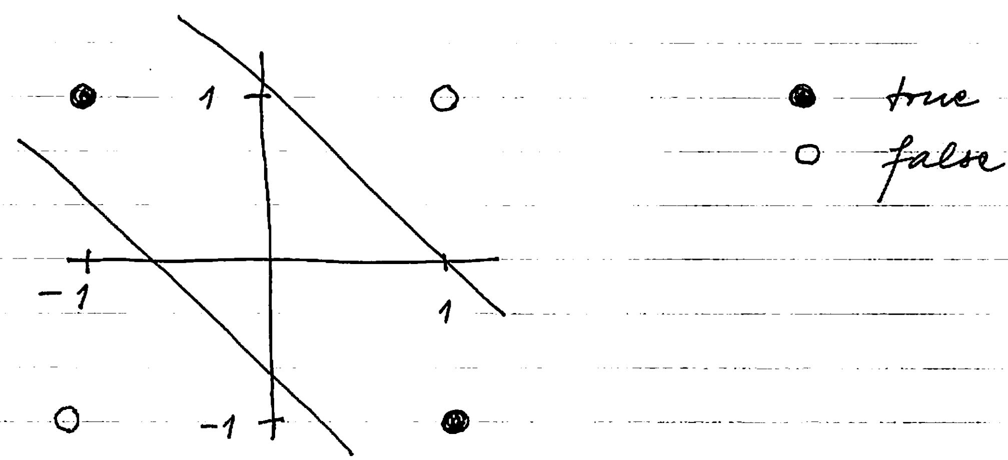


bias ($x_0 = 1$) is needed to allow shifting the line from the origin

Now try to train for XOR

x_0	x_1	x_2	x_1 XOR x_2
1	1	1	-1
1	1	-1	1
1	-1	1	1
1	-1	-1	-1

'XOR'



Two lines are needed to separate true from false!
One PE can't do it.

Note:

In this example, we train with all possible (four) input patterns (cases), essentially learning a lookup table. This is not generally the case & goal. (Generally, we want the ANN to learn to generalize – apply what it learns from training data, to new data – but here there are no "new" data.) We work this simple example to be able to complete a training manually.