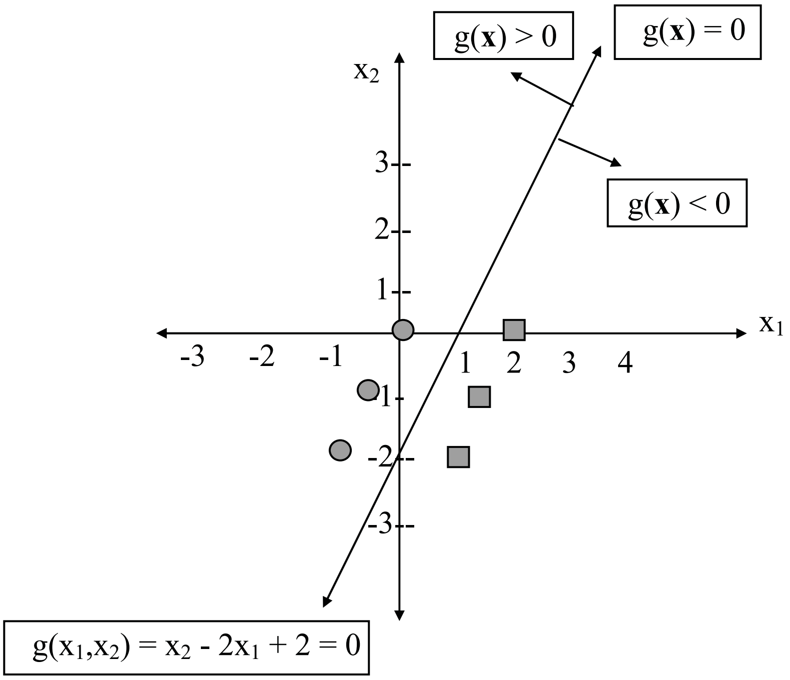
FIT5186 Examples from Lecture 2

Perceptrons as Classifiers

CLASS A: ●
(0,0), (-0.5,-1), (-1,-2)

CLASS B: ■
(2,0), (1.5,-1), (1,-2)



General form of straight line: $x_2 = mx_1 + c$

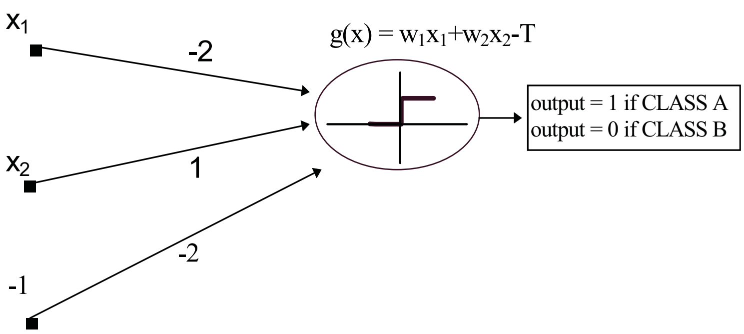
$m = gradient = 2$

$c = vertical\ intercept = -2$ so $x_2 = 2x_1 - 2$ or $x_2 - 2x_1 + 2 = 0$

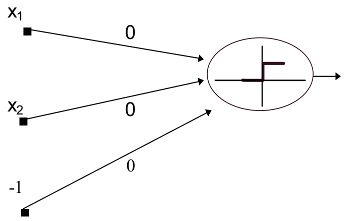
note: there are infinitely many lines to separate these two classes
Choose a line: it is your discriminant function or "decision boundary"

Suppose we have a new point (3,2) and we want to know if it belongs to class A or B.
Evaluate $g(3,2) = 2 - 2 * 3 + 2 = -2$
Since $g(3,2) < 0$, then this new point is below the decision boundary (you could see that from the graph anyway) and so the point is classified as a class B.

PERCEPTRON CLASSIFIER



Perceptron Learning Algorithm



PERCEPTRON LEARNING RULE

STEP 1: Initialise weights to zero

Inputs (x1,x2)	0 0 -1	1.5 -1 -1	-1 -2 -1	2 0 -1	-0.5 -1 -1	1 -2 -1
Outputs (desired)	1	0	1	0	1	0

$$\boxed{\mathbf{w}^0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}}$$

STEP 2: Present input 1 $\quad \mathbf{x} = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} \quad$ d = 1

STEP 3: Calculate output $\quad o = f(0 \times 0 + 0 \times 0 + 0 \times -1) = f(0) = 1$

STEP 4: Since o = d = 1, there is no error, so no weight adaptation is needed

---

STEP 2: Present input 2 $\quad \mathbf{x} = \begin{pmatrix} 1.5 \\ -1 \\ -1 \end{pmatrix} \quad$ d = 0

STEP 3: Calculate output $\quad o = f(0 \times 1.5 + 0 \times -1 + 0 \times -1) = f(0) = 1$

STEP 4: Since o = 1 and d = 0, weight adaptations are needed to correct the error

$$\mathbf{w}^1 = \mathbf{w}^0 + c(d-o)\mathbf{x}$$

$$= \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} + 1 \times -1 \times \begin{pmatrix} 1.5 \\ -1 \\ -1 \end{pmatrix} = \boxed{\begin{pmatrix} -1.5 \\ 1 \\ 1 \end{pmatrix} = \mathbf{w}^1}$$

STEP 2: Present input 3 $\quad \mathbf{x} = \begin{pmatrix} -1 \\ -2 \\ -1 \end{pmatrix} \quad$ d = 1

STEP 3: Calculate output $\quad o = f(-1.5 \times -1 + 1 \times -2 + 1 \times -1) = f(-1.5) = 0$

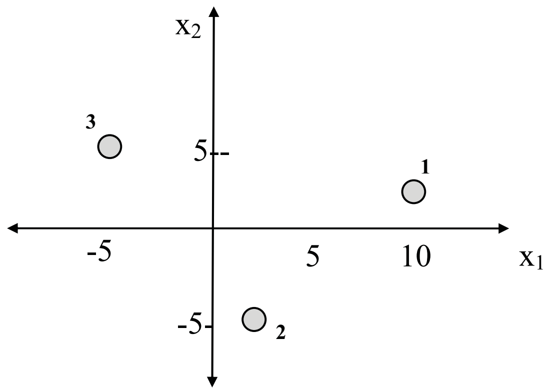
STEP 4: Since o = 0 and d = 1, weight adaptations are needed to correct the error

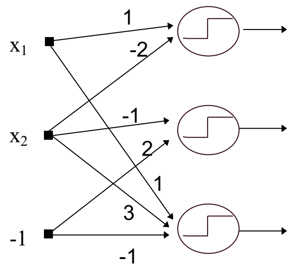
$$\mathbf{w}^2 = \mathbf{w}^1 + c(d-o)\mathbf{x}$$

$$= \begin{pmatrix} -1.5 \\ 1 \\ 1 \end{pmatrix} + 1 \times 1 \times \begin{pmatrix} -1 \\ -2 \\ -1 \end{pmatrix} = \boxed{\begin{pmatrix} -2.5 \\ -1 \\ 0 \end{pmatrix} = \mathbf{w}^2}$$

Present inputs 4, 5 and 6 etc. This is the end of 1 epoch. Continue to present the training set until the weights no longer change: ie. each input results in correct output

<u>Example:</u> R-category Perceptrons





There are three classes, so 3 discrete Perceptrons are used

Inputs (x1,x2)	10 2 -1	2 -5 -1	-5 5 -1
Outputs (desired)	1 0 0	0 1 0	0 0 1

STEP 1:

Initial weight matrix (read from network graph):

$$\mathbf{W}^0 = \begin{pmatrix} 1 & -2 & 0 \\ 0 & -1 & 2 \\ 1 & 3 & -1 \end{pmatrix} = \begin{pmatrix} \mathbf{w}_1^0 \\ \mathbf{w}_2^0 \\ \mathbf{w}_3^0 \end{pmatrix}$$

STEP 2: Present input 1 $\quad \mathbf{x} = \begin{pmatrix} 10 \\ 2 \\ -1 \end{pmatrix}$ with $\mathbf{d} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$

STEP 3: Calculate output vector $\quad \mathbf{o} = f\begin{pmatrix} 6 \\ -4 \\ 17 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$

STEP 4: Adapt weights since $\mathbf{o} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$ and $\mathbf{d} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$ (error in third neuron)

$$\mathbf{w}_3^1 = \mathbf{w}_3^0 + c(d_3 - o_3)\mathbf{x} = \begin{pmatrix} 1 \\ 3 \\ -1 \end{pmatrix} + 1 \times -1 \times \begin{pmatrix} 10 \\ 2 \\ -1 \end{pmatrix} = \begin{pmatrix} -9 \\ 1 \\ 0 \end{pmatrix} = \mathbf{w}_3^1$$

$\mathbf{w}_1^1 = \mathbf{w}_1^0$ and $\mathbf{w}_2^1 = \mathbf{w}_2^0$ (unchanged)

So after adaptation: $\mathbf{W}^1 = \begin{pmatrix} 1 & -2 & 0 \\ 0 & -1 & 2 \\ -9 & 1 & 0 \end{pmatrix} = \begin{pmatrix} \mathbf{w}_1^1 \\ \mathbf{w}_2^1 \\ \mathbf{w}_3^1 \end{pmatrix}$

STEP 2: Present input 2 $\mathbf{x} = \begin{pmatrix} 2 \\ -5 \\ -1 \end{pmatrix}$ with $\mathbf{d} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$

STEP 3: Calculate output vector $\mathbf{o} = f\begin{pmatrix} 12 \\ 3 \\ -23 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$

STEP 4: Adapt weights since $\mathbf{o} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$ and $\mathbf{d} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$ (error in first neuron)

$$\mathbf{w}_1^2 = \mathbf{w}_1^1 + c(d_1 - o_1)\mathbf{x} = \begin{pmatrix} 1 \\ -2 \\ 0 \end{pmatrix} + 1 \times -1 \times \begin{pmatrix} 2 \\ -5 \\ -1 \end{pmatrix} = \begin{pmatrix} -1 \\ 3 \\ 1 \end{pmatrix} = \mathbf{w}_1^2$$

$\mathbf{w}_2^2 = \mathbf{w}_2^1$ and $\mathbf{w}_3^2 = \mathbf{w}_3^1$ (unchanged)

So after adaptation: $\mathbf{W}^2 = \begin{pmatrix} -1 & 3 & 1 \\ 0 & -1 & 2 \\ -9 & 1 & 0 \end{pmatrix} = \begin{pmatrix} \mathbf{w}_1^2 \\ \mathbf{w}_2^2 \\ \mathbf{w}_3^2 \end{pmatrix}$

STEP 2: Present input 3 $\mathbf{x} = \begin{pmatrix} -5 \\ 5 \\ -1 \end{pmatrix}$ with $\mathbf{d} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$

STEP 3: Calculate output vector $\mathbf{o} = f\begin{pmatrix} 19 \\ -7 \\ 50 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$

STEP 4: Adapt weights since $\mathbf{o} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$ and $\mathbf{d} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$ (error in first neuron)

$$\mathbf{w}_1^3 = \mathbf{w}_1^2 + c(d_1 - o_1)\mathbf{x} = \begin{pmatrix} -1 \\ 3 \\ 1 \end{pmatrix} + 1 \times -1 \times \begin{pmatrix} -5 \\ 5 \\ -1 \end{pmatrix} = \begin{pmatrix} 4 \\ -2 \\ 2 \end{pmatrix} = \mathbf{w}_1^3$$

$\mathbf{w}_2^3 = \mathbf{w}_2^2$ and $\mathbf{w}_3^3 = \mathbf{w}_3^2$ (unchanged)

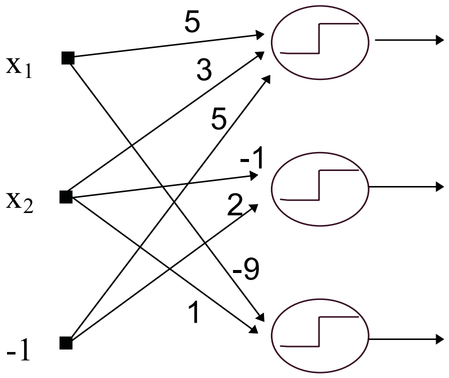
So after adaptation: $\mathbf{W}^3 = \begin{pmatrix} 4 & -2 & 2 \\ 0 & -1 & 2 \\ -9 & 1 & 0 \end{pmatrix} = \begin{pmatrix} \mathbf{w}_1^3 \\ \mathbf{w}_2^3 \\ \mathbf{w}_3^3 \end{pmatrix}$

<u>END OF 1ST EPOCH</u>

Repeat with input 1, etc.

Only the first row of the weight matrix appears to be changing (the first neuron still has not learnt the mapping, but the other neurons get the right answers).

The final weights (after several epochs) are: $\mathbf{W} = \begin{pmatrix} 5 & 3 & 5 \\ 0 & -1 & 2 \\ -9 & 1 & 0 \end{pmatrix} = \begin{pmatrix} \mathbf{w}_1^3 \\ \mathbf{w}_2^3 \\ \mathbf{w}_3^3 \end{pmatrix}$



Equations of boundary lines:

CLASS 1: $5x_1 + 3x_2 - 5 = 0 = g_1$

CLASS 2: $-x_2 - 2 = 0 = g_2$

CLASS 3: $-9x_1 + x_2 = 0 = g_3$

Where a new input gives $g_1>0$, $g_2<0$, $g_3<0$ then it's CLASS 1

Where a new input gives $g_1<0$, $g_2>0$, $g_3<0$ then it's CLASS 2

Where a new input gives $g_1<0$, $g_2<0$, $g_3>0$ then it's CLASS 3

"Indecision regions" form when more than one neuron outputs "1", or no neurons output "1"
ie. $g_1>0$, $g_2>0$, $g_3<0$ (neurons 1 and 2 output "1")
or $g_1<0$, $g_2<0$, $g_3<0$ (no neurons output "1")