Ömer Morgül $\overline{NAME}$ $\overline{FAMILYNAME}$ $\overline{SECTION}$

EEE 443/543 Neural Networks
Final, Fall 2012-13

No credits will be given for unjustified answers.

Prob. 1 : (30 pt.s) Consider the Hopfield Neural Network given as :

$$o(k+1) = sgn(Wo(k))$$

where sgn is the signum function given as $sgn(v) = 1$ when $v \geq 0$, $sgn(v) = -1$ when $v < 0$. Here we consider a 3 dimensional case, where $o = \begin{pmatrix} o_1 \\ o_2 \\ o_3 \end{pmatrix}$ is the output vector, note that the threshold vector is chosen as $\theta = 0$, and W is a 3×3 weight matrix, $k = 1, 2, \ldots$ is the iteration index. The cost $E(o)$ associated with Hopfield Network is given as $E(o) = -0.5o'Wo$, where the superscript $'$ denotes transpose. Assume that W is given as :

$$W = \begin{pmatrix} 0 & \omega_1 & \omega_2 \\ \omega_1 & 0 & \omega_3 \\ \omega_2 & \omega_3 & 0 \end{pmatrix}.$$

Here $\omega_1, \omega_2, \omega_3$ are constant weights. Our aim is to store the following patterns y_1, y_2

$$y_1 = \begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix} \quad , \quad y_2 = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}.$$

i : Show that for any real number ω, if we choose $\omega_2 = \omega_3 = \omega$ and $\omega_1 < -\mid \omega \mid$, then both y_1 and y_2 become the stored patterns of this Hopfield network (i.e. they become fixed points of the iteration).

ii : For the rest of the problem, use $\omega_1 = -2$, $\omega_2 = \omega_3 = 1$. Show that the cost given above satisfies $-2 \leq E(o) \leq 4$.

iii : Now let $o(1) = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$ be given. By using synchronous update rule, find $o(2)$ and $o(3)$, and find the costs associated with $o(1), o(2), o(3)$. Predict the future iterations, if you can.

iv : Let $o(1)$ be given as in **iii**. By using asynchronous update, find the consecutive outputs. Continue till the output converges to a stored pattern. (If the output does not converge, stop at 3rd update cycle). What is the cost of the last updated output?

Problem 2 : (20 pt.s) This problem is on linear Support Vector Machines. Consider the two class case given below. In this figure $\times$ indicates a class 1 pattern, and a $\circ$ indicates a class 2 pattern. These patterns are also labeled as $z_1, \ldots, z_7$ for class 1 and $y_1, \ldots, y_6$ for class 2 cases, as indicated in the figure. Our aim is to guess the possible support vectors. (Note that in this problem, extensive numerical calculations such as solving KKT equations, Dual-Lagrangian formulation etc. are not required, simple and convincing algebraic and/or geometrical arguments are sufficient.)

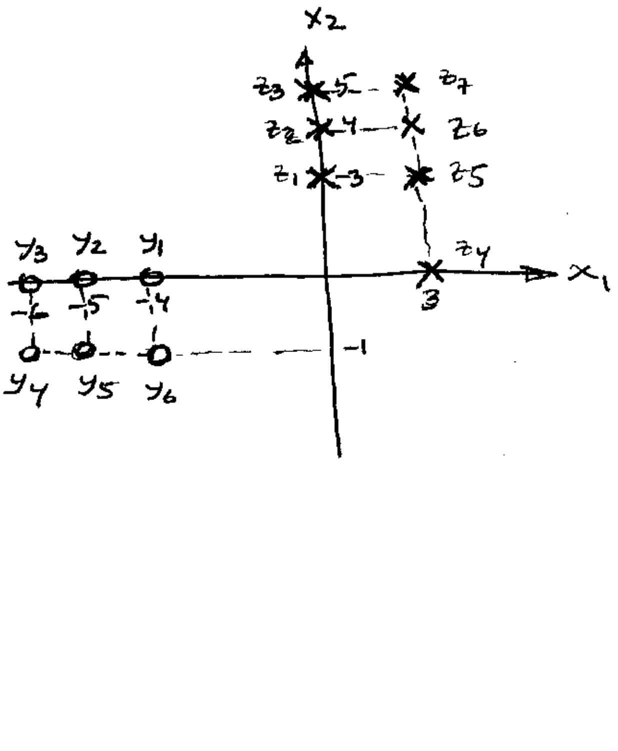
(For the last two part, the following information might be useful : Two parallel lines are given by $w^T x = c_1$, $w^T x = c_2$, the distance between these lines are $\delta = \mid c_1 - c_2 \mid / \|w\|$)

i : Can z_6 be a support vector for class 1? (yes/no with a convincing justification).

ii : Assume that z_2 is a support vector for class 1. Can it be the **only** support vector for class 1? If yes, explain briefly the reason; if no, which other pattern(s) in class 1 is/are also (possibly) support vector(s)?

iii : If z_1 and z_4 are the **only** support vectors for class 1, which pattern(s) in class 2 is/are support vector(s) for class 2 ? In this case, what is the distance between the support lines H_+ and H_-?

iv : If If z_1 is the **only** support vector for class 1, which pattern(s) in class 2 is/are support vector(s) for class 2 ? In this case, what is the distance between the support lines H_+ and H_-?



Problem 3 : (20 pt.s) The following problem is related to generating **nonlinear** decision boundaries by using a single neuron. Let the input x be given as $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ and consider the nonlinear transformation $y = \phi(x) = \begin{pmatrix} 1 \\ x_1 \\ x_2^2 \end{pmatrix}$. Let us define $\phi_1 = 1$, $\phi_2 = x_1$, $\phi_3 = x_2^2$. Consider the structure given below, where the last part is standard neuron with weights w_1, w_2, w_3. Hence, $v = w_1\phi_1 + w_2\phi_2 + w_3\phi_3$, and $sgn(v)$ is the standard signum function defined as $sgn(v) = +1$ when $v \geq 0$ and $sgn(v) = -1$ when $v < 0$. Consider the standard 2 class classification problem, i.e. we want $o = 1$ when x is in class 1 (C_1), and $o = -1$ when x is in class 2 (C_2).

i : Sketch the possible candidates for C_1 and C_2 which could be classified with this network. Try to give as many possible cases as you can.

ii : Now consider the patterns given in Figure 2. In this figure $\times$ indicates a class 1 pattern, and a $\circ$ indicates a class 2 pattern. Find appropriate weights w_1, w_2, w_3 such that the network given in Figure 1 solves this classification problem.

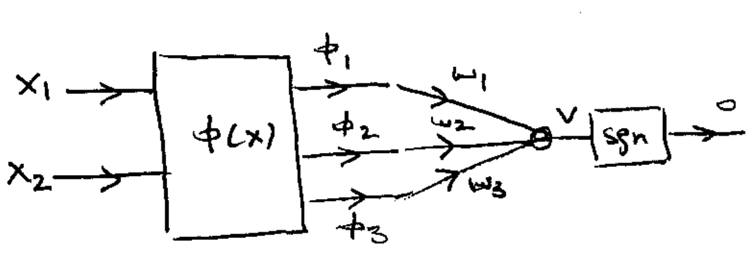
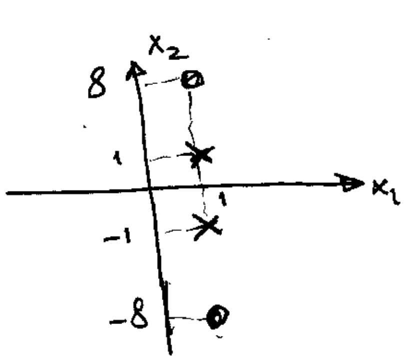


Fig. 1



Problem 4 : (30 pt. s) This problem is related to Kohonen Network, winner-take-all algorithm and one of its generalization which is called Learning Vector Quantization (LVQ). Assume that input y is given as $y = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ and consider the standard Kohonen network given in Figure 1. Consider the standard 2 class classification case in Kohonen network, i.e. we want the first neuron to win the competition if y is in class 1 (C_1), and second neuron to win the competition when y is in class 2 (C_2). Assume that the initial weights are given as $w_1(1) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $w_2(1) = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$ (Here w_1 is the weight for the first neuron, and w_2 is the weight for the second neuron). In the learning algorithms considered below, take the learning constant α as $\alpha = 0.5$.

i : Consider the patterns given in Figure 2. Here, $y_1 \in C_1$ and $y_2 \in C_2$. Consider the winner-take-all algorithm (with normalization). What can you say about the convergence of the algorithm? Does it converge (i.e. we achieve correct classification), or not? (Extensive numerical calculations are not required, simple and convincing algebraic and/or geometrical arguments are sufficient).

ii : To be concrete, assume that $y_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $y_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$, $C_1 = \{y_1\}$ and $C_2 = \{y_2\}$. Consider the initial weights and the learning constant as given above. Use the winner-take-all algorithm (without normalization). Show that second neuron never wins the competition and that the first neuron weight is always of the form $w_1 = \begin{pmatrix} 1 \\ w \end{pmatrix}$ where $-1 < w < 1$, during the learning phase.

iii : Learning Vector Quantization (LVQ) tries to eliminate this problem. The update rule is the following modification of the winner-take-all rule :

-apply the pattern y. (Note that we know the class y, hence the correct neuron which should win).
- Find the winning neuron.
- If the winning neuron is CORRECT (i.e. the first neuron for C_1, and second neuron for C_2), update the winning neuron weight as in winner-take-all case :
$w \leftarrow w + \alpha(y - w)$
- If the winning neuron is INCORRECT (i.e. the second neuron for C_1, and first neuron for C_2), update the winning neuron weight as :
$w \leftarrow w - \alpha(y - w)$

Now consider the initial weights, C_1, C_2, y_1, y_2 and α as given above. Apply the LVQ algorithm given above for two epochs and find the final weight. Determine whether the final weights solve the classification problem or not.

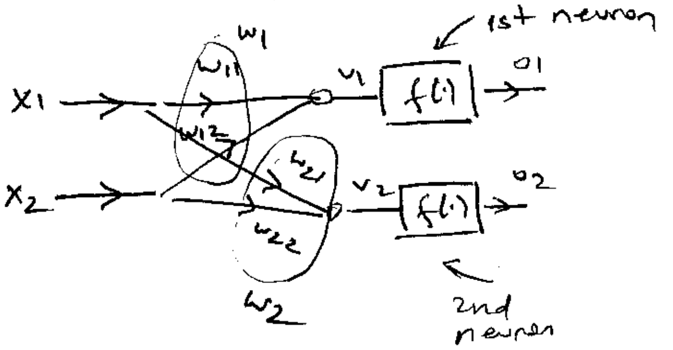
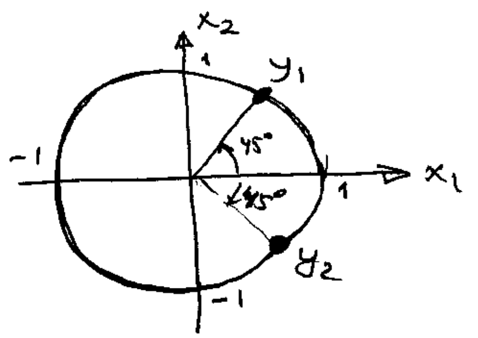


Fig. 1



EEE 443/543 FALL 2012 FINAL SOLUTIONS

P1) i)

12/12

$$W=\begin{pmatrix} 0 & w_1 & w \\ w_1 & 0 & w \\ w & w & 0 \end{pmatrix} \qquad y_1=\begin{pmatrix}-1\\1\\1\end{pmatrix} \qquad y_2=\begin{pmatrix}1\\-1\\1\end{pmatrix}$$

$$y_1=\text{sign}(Wy_1)=\text{sign}\begin{pmatrix} w_1+w \\ w-w_1 \\ 0\end{pmatrix} \Rightarrow \left.\begin{matrix} w_1+w<0 \\ w-w_1\geq 0 \\ 0\geq 0\end{matrix}\right\} \left.\begin{matrix} w_1<-w \\ w_1\leq w\end{matrix}\right\} \text{ satisfied.}$$

03 03

$$y_2=\text{sign}(Wy_2)=\text{sign}\begin{pmatrix} w-w_1 \\ w+w_1 \\ 0\end{pmatrix} \Rightarrow \left.\begin{matrix} w-w_1\geq 0 \\ w+w_1<0 \\ 0\geq 0\end{matrix}\right\} \left.\begin{matrix} w_1<-w \\ w_1\leq w\end{matrix}\right\} \text{ satisfied.}$$

03 03

$\Rightarrow$ with $w_2=w_3=w$ and $w_1<-|w|$, y_1, y_2 becomes memory patterns.

ii) $E=-\frac{1}{2}o^T W o \qquad o=\begin{pmatrix}o_1\\o_2\\o_3\end{pmatrix} \Rightarrow Wo=\begin{pmatrix}0 & -2 & 1\\ -2 & 0 & 1\\ 1 & 1 & 0\end{pmatrix}\begin{pmatrix}o_1\\o_2\\o_3\end{pmatrix}=\begin{pmatrix}-2o_2+o_3\\ -2o_1+o_3\\ o_1+o_2\end{pmatrix}$

06

$$\Rightarrow E=-\frac{1}{2}(o_1\ o_2\ o_3)\begin{pmatrix}-2o_2+o_3\\ -2o_1+o_3\\ o_1+o_2\end{pmatrix}=-\frac{1}{2}(-2o_1o_2+o_1o_3-2o_1o_2+o_2o_3+o_1o_3+o_2o_3)$$

02

$$=-\frac{1}{2}(-4o_1o_2+2o_1o_3+2o_2o_3)=2o_1o_2-o_1o_3-o_2o_3=2o_1o_2-(o_1+o_2)o_3$$

$o_i=\mp 1$ if $o_1=o_2$ and $o_3=-o_1=-o_2 \Rightarrow E=4$

if $o_1\neq o_2 \Rightarrow o_1+o_2=0$ and $E=-2$

$$\boxed{-2\leq E\leq 4}$$ 04

iii) 06/06

$$o(1)=\begin{pmatrix}1\\1\\1\end{pmatrix} \qquad o(2)=\text{sgn}(Wo(1))=\text{sgn}\begin{pmatrix}-1\\-1\\2\end{pmatrix}=\begin{pmatrix}-1\\-1\\1\end{pmatrix}$$ 01

$$o(3)=\text{sgn}(Wo(2))=\text{sgn}\begin{pmatrix}1\\1\\-2\end{pmatrix}=\begin{pmatrix}1\\1\\-1\end{pmatrix}$$ 01 $$o(4)=\text{sgn}(Wo(3))=\begin{pmatrix}-1\\-1\\1\end{pmatrix}$$ 01

$$\Rightarrow o(2)=o(4)=\cdots=\begin{pmatrix}-1\\-1\\1\end{pmatrix} \qquad o(3)=o(5)=\cdots\begin{pmatrix}1\\1\\-1\end{pmatrix}$$ 03

$E(o(1))=0 \qquad E(o(2))=4 \qquad E(o(3))=4$ 02

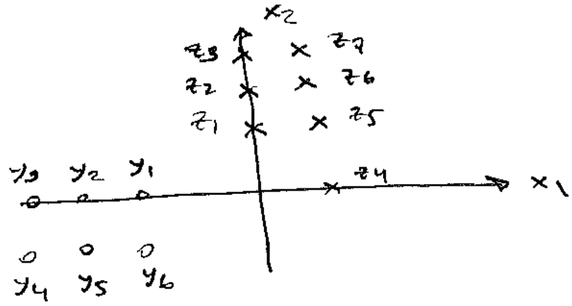
iv) 06/06

$o(1)=\begin{pmatrix}1\\1\\1\end{pmatrix} \qquad v_1=-1 \Rightarrow$ update $o_1 \rightarrow \begin{pmatrix}-1\\1\\1\end{pmatrix}$ 01 $\rightarrow v_2=3 \Rightarrow$ update $o_2 \rightarrow \begin{pmatrix}-1\\1\\1\end{pmatrix}$ 01

$\rightarrow v_3=0 \Rightarrow$ update $o_3 \Rightarrow \begin{pmatrix}-1\\1\\1\end{pmatrix}=y_1$ 01 MEMORY PATTERN! update stops 01

$E(o(2))=-2 \leftarrow$ minimum! 02

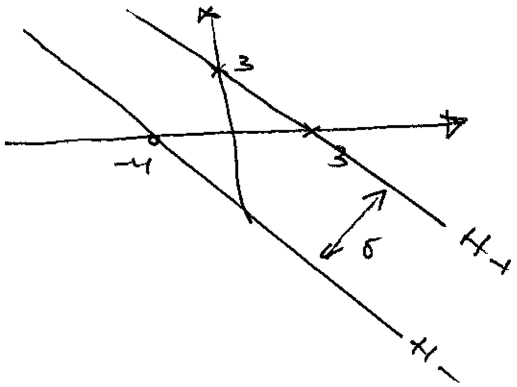
P2)



i) z_6 cannot be a support vector. Any line passing thru z_6 cannot separate C_1 and C_2

ii) z_2 cannot be the only SV for C_1. Otherwise, it cannot separate C_1 and C_2. The only possibility is a line passing thru z_1, z_2, z_3 (x_2 axis). $\Rightarrow$ z_1 and z_3 should also be SV's.

iii) If z_1 and z_4 are the only SV's for C_1, then in C_2 only y_1 could be a support vector (just slide the line)

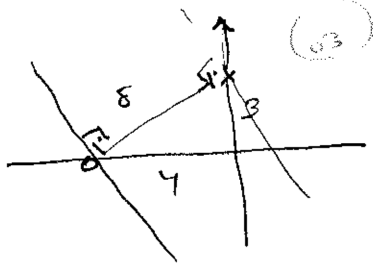


$H_+ : x_1 + x_2 = 3$

$H_- : x_1 + x_2 = -4$

$$\delta = \frac{|3+4|}{\sqrt{2}} = \frac{7}{\sqrt{2}}$$

iv) If z_1 is the only SV for C_1 then for the same reason, y_1 is the only SV for C_2. In this case we have a min. distance classifier.



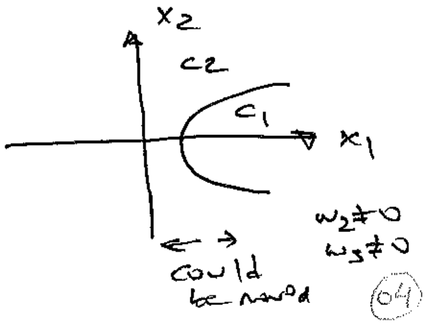
$\underline{\underline{\delta = 5}}$! (3-4-5 triangle)

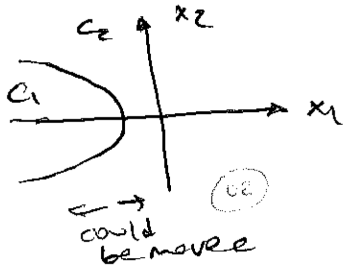
P3) i) $V = w_1\phi_1 + w_2\phi_2 + w_3\phi_3$

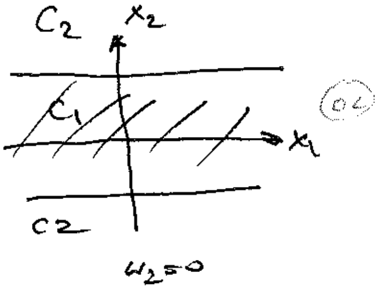
$= w_1 + w_2 x_1 + w_3 x_2^2$

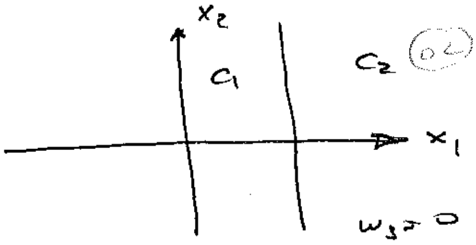
LOOK AT $V=0$ curves (SEPARATING CURVES)

10/10



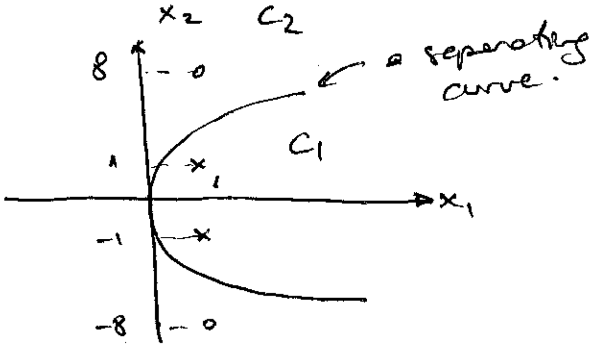






. . . .

ii)



(10) ONE EXAMPLE

$V = 0 + 2x_1 - x_2^2$

$\boxed{\begin{array}{l} w_1 = 0 \\ w_2 = 2 \\ w_3 = -1 \end{array}}$

ONE EXAMPLE (there are inf. many)

or OR (10)

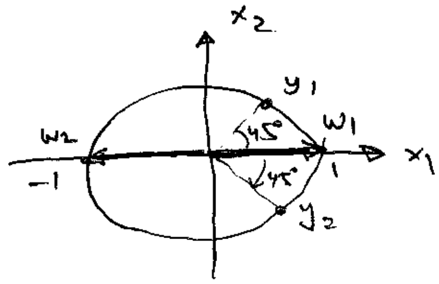
$\boxed{\begin{array}{l} w_1 + w_2 + w_3 \geq 0 \\ w_1 + w_2 + w_3 \geq 0 \\ w_1 + w_2 + 64w_3 < 0 \\ w_1 + w_2 + 64w_3 < 0 \end{array}}$

Any w_1, w_2, w_3 satisfying these are solutions.

(Find ONE!)

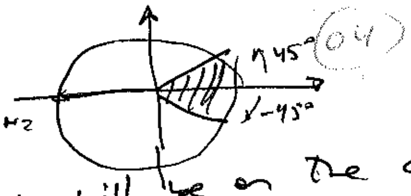
(there are inf. many)

P4) i) 03



when y_1 is presented $\Rightarrow w_1$ wins (has smaller angle with y_1)
w_1 moves $\to y_1$
when y_2 is presented still w_1 wins because has smaller angle with y_2

$\Rightarrow w_1$ will always be in the cone



04

w_2 is far away (anglewise)

04

so ALWAYS w_1 wins and the weight w_1 will be on the cone
w_2 will never win and stays at $w_2 = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$.
classification is not achieved.

ii) Apply $y_1 \to$ $v_1 = w_1(1)^T y_1 = 1$, $v_2 = w_2(1)^T y_1 = -1$ } 1st wins $\to$

10

$$w_1 \leftarrow w_1 + \tfrac{1}{2}(y_1 - w_1) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \tfrac{1}{2}\left(\begin{pmatrix} 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \end{pmatrix}\right) = \begin{pmatrix} 1 \\ 1/2 \end{pmatrix}$$ 02

Now assume that $w_1(\cdot) = \begin{pmatrix} 1 \\ w \end{pmatrix}$ $\quad -1 < w < 1$

Apply $y_1 \to$ $v_1 = 1 + w = w_1(\cdot)^T y_1$, $v_2 = -1 = w_2(1)^T y_1$ } w_1 wins

$$w_1 \leftarrow \begin{pmatrix} 1 \\ w \end{pmatrix} + \tfrac{1}{2}\left[\begin{pmatrix} 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 \\ w \end{pmatrix}\right] = \begin{pmatrix} 1 \\ \tfrac{1}{2}(1+w) \end{pmatrix}$$ 05

$\to 0 \le \frac{1}{2}(1+w) = \hat{w} < 1$

Apply y_2 $\quad v_1 = w_1^T y_2 = 1 - w$ — wins
$v_2 = -1$

$$w_1 \leftarrow \begin{pmatrix} 1 \\ \hat{w} \end{pmatrix} + \tfrac{1}{2}\left(\begin{pmatrix} 1 \\ -1 \end{pmatrix} - \begin{pmatrix} 1 \\ \hat{w} \end{pmatrix}\right) = \begin{pmatrix} 1 \\ -\frac{1}{2} + \frac{1}{2}\hat{w} \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{1}{2}(\hat{w} - 1) \end{pmatrix}$$

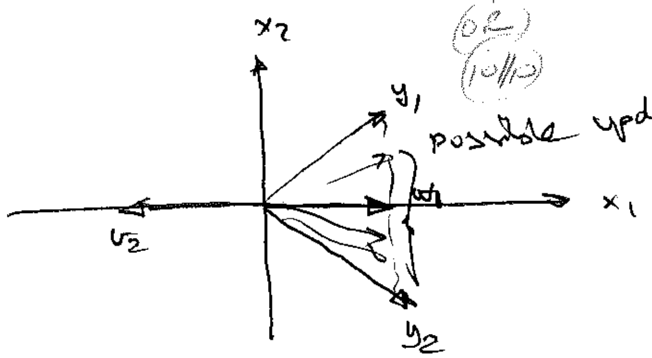
$-1 < \frac{1}{2}(\hat{w} - 1) < 0$ 04

Hence always 1st neuron wins and $w_1 \to \begin{pmatrix} 1 \\ w \end{pmatrix}$ $\boxed{-1 < w < 1}$

GEOMETRICALLY

02

(10/10)



possible updated w_1 weights
Always in the form
$w_1 \to \begin{pmatrix} 1 \\ w \end{pmatrix}$

P4) iii) LVQ method. $w_1(1) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ $w_2(1) = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$

12

Apply y_1 $v_1 = 1$ $\rightarrow$ wins. AND CORRECT

$v_2 = -1$

$$w_1 \leftarrow w_1 + \frac{1}{2}(y_1 - w)$$
$$= \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \frac{1}{2}\left(\begin{pmatrix} 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \end{pmatrix}\right) = \begin{pmatrix} 1 \\ 1/2 \end{pmatrix}$$

02

Apply y_2 $v_1 = 1/2$ $\rightarrow$ wins BUT INCORRECT

$v_2 = -1$

$$w_1 \leftarrow w_1 - \frac{1}{2}(y_2 - w_1)$$
$$= \begin{pmatrix} 1 \\ 1/2 \end{pmatrix} - \frac{1}{2}\left(\begin{pmatrix} 1 \\ -1 \end{pmatrix} - \begin{pmatrix} 1 \\ 1/2 \end{pmatrix}\right) = \begin{pmatrix} 1 \\ 5/4 \end{pmatrix}$$

02

$\rightarrow$ EPOCH 1.

Apply y_1 $v_1 = 1 + \frac{5}{4} = \frac{9}{4}$ $\rightarrow$ wins AND CORRECT

$v_2 = -1$

$$w_1 \leftarrow w_1 + \frac{1}{2}(y_1 - w_1)$$
$$= \begin{pmatrix} 1 \\ 5/4 \end{pmatrix} + \frac{1}{2}\left(\begin{pmatrix} 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 \\ 5/4 \end{pmatrix}\right)$$
$$= \begin{pmatrix} 1 \\ 9/8 \end{pmatrix}$$

02

Apply y_2 $v_1 = 1 - \frac{9}{8} = -\frac{1}{8}$ $\rightarrow$ wins BUT INCORRECT

$v_2 = -1$

$$w_1 \leftarrow (w_1) - \frac{1}{2}(y_2 - w_1)$$
$$= \begin{pmatrix} 1 \\ 9/8 \end{pmatrix} - \frac{1}{2}\left(\begin{pmatrix} 1 \\ -1 \end{pmatrix} - \begin{pmatrix} 1 \\ 9/8 \end{pmatrix}\right)$$
$$= \begin{pmatrix} 1 \\ 35/16 \end{pmatrix}$$

02

— END OF SECOND EPOCH.

$w_1 = \begin{pmatrix} 1 \\ \frac{35}{16} \end{pmatrix}$ $w_2 = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$

Apply y_1 $\rightarrow$ $v_1 = 1 + \frac{35}{16}$ $\Rightarrow$ WINS AND CORRECT

$v_2 = -1$

04

Apply y_2 $v_1 = 1 - \frac{35}{16} = -\frac{19}{16}$

$v_2 = -1$ $\Rightarrow$ WINS AND CORRECT

$\Rightarrow$ CORRECT CLASSIFICATION IS ACHIEVED!