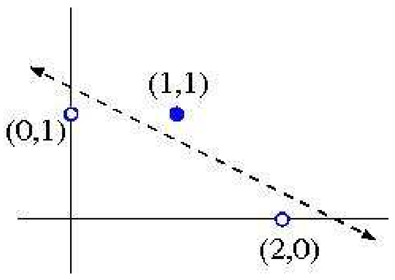
COMP9414: Artificial Intelligence
Solutions 9: Neural Networks/Reinforcement Learning

1. (i) The first step is to plot the data on a 2-D graph, and draw a line which separates the positive from the negative data points:



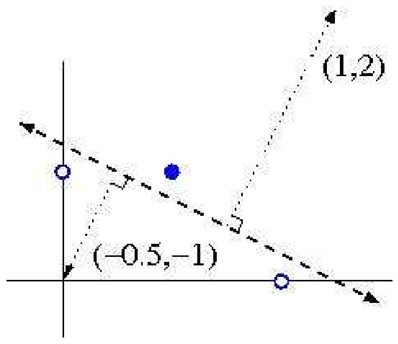
This line has slope $-1/2$ and x_2-intercept $5/4$, so its equation is:

$x_2 = 5/4 - x_1/2$,
i.e. $2x_1 + 4x_2 - 5 = 0$.

Taking account of which side is positive, this corresponds to these weights:

$w_0 = -5$
$w_1 = 2$
$w_2 = 4$

Alternatively, we can derive weights $w_1 = 1$ and $w_2 = 2$ by drawing a vector normal to the separating line, in the direction pointing towards the positive data points:



The bias weight w_0 can then be found by computing the dot product of the normal vector with a perpendicular vector from the separating line to the origin. In this case, $w_0 = 1(-0.5) + 2(-1) = -2.5$.

Note: These weights differ from the previous ones by a normalizing constant, which makes no difference for a perceptron.

(ii)

Iteration	w_0	w_1	w_2	Example	x_1	x_2	Class	$w_0 + w_1x_1 + w_2x_2$	Action
1	-0.5	0	1	a	0	1	$-$	$+0.5$	Subtract
2	-1.5	0	0	b	2	0	$-$	-1.5	None
3	-1.5	0	0	c	1	1	$+$	-1.5	Add
4	-0.5	1	1	a	0	1	$-$	$+0.5$	Subtract
5	-1.5	1	0	b	2	0	$-$	$+0.5$	Subtract
6	-2.5	-1	0	c	1	1	$+$	-3.5	Add
7	-1.5	0	1	a	0	1	$-$	-0.5	None
8	-1.5	0	1	b	2	0	$-$	-1.5	None
9	-1.5	0	1	c	1	1	$+$	-0.5	Add
10	-0.5	1	2	a	0	1	$-$	$+1.5$	Subtract
11	-1.5	1	1	b	2	0	$-$	$+0.5$	Subtract
12	-2.5	-1	1	c	1	1	$+$	-2.5	Add
13	-1.5	0	2	a	0	1	$-$	$+0.5$	Subtract
14	-2.5	0	1	b	2	0	$-$	-2.5	None
15	-2.5	0	1	c	1	1	$+$	-1.5	Add
16	-1.5	1	2	a	0	1	$-$	$+0.5$	Subtract
17	-2.5	1	1	b	2	0	$-$	-0.5	None
18	-2.5	1	1	c	1	1	$+$	-0.5	Add
19	-1.5	2	2	a	0	1	$-$	$+0.5$	Subtract
20	-2.5	2	1	b	2	0	$-$	$+1.5$	Subtract
21	-3.5	0	1	c	1	1	$+$	-2.5	Add
22	-2.5	1	2	a	0	1	$-$	-0.5	None
23	-2.5	1	2	b	2	0	$-$	-0.5	None
24	-2.5	1	2	c	1	1	$+$	$+0.5$	None

2. (i) Set the bias weight to $-1/2$, all other weights to 1.

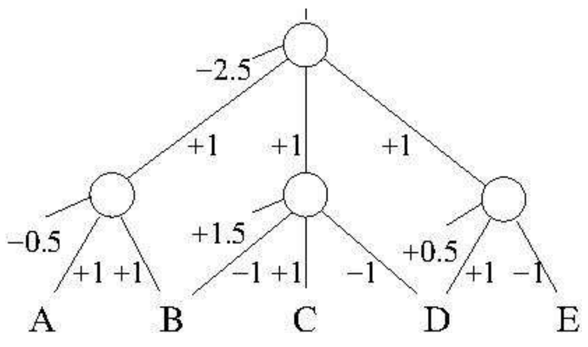
The OR function is almost always True. The only way it can be False is if all inputs are 0. Therefore, we set the bias to be slightly less than zero for this input.

(ii) Set the bias weight to $1/2 - n$, all other weights to 1.

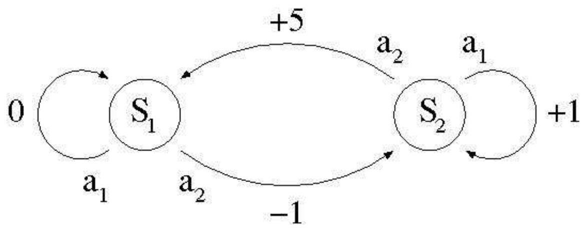
The AND function is almost always False. The only way it can be True is if all inputs are 1. Therefore, we set the bias so that when all inputs are 1, the combined sum is slightly greater than 0.

(iii) Each hidden node should compute one disjunctive term in the expression. The weights should be -1 for items that are negated, $+1$ for the others. The bias should be $k - 1/2$ where k is the number of items that are negated. The output node then computes the conjunction of all the hidden nodes, as in part (ii).

For example, here is a network that computes $(A \vee B) \wedge (\neg B \vee C \vee \neg D) \wedge (D \vee \neg E)$.



3. (i)



(ii) (a) The optimal policy is:

$\pi^*(S_1) = a_2$
$\pi^*(S_2) = a_2$

(b) The optimal value function V^* is calculated as follows.

$V^*(S_1) = -1 + \gamma V^*(S_2)$
$V^*(S_2) = 5 + \gamma V^*(S_1)$

So $V^*(S_1) = -1 + 5\gamma + \gamma^2 V^*(S_1)$
i.e. $V^*(S_1) = (-1 + 5\gamma)/(1 - \gamma^2) = 3.5/0.19 = 18.42$

$V^*(S_2) = 5 + \gamma V^*(S_1) = 5 + 0.9 * 3.5/0.19 = 21.58$

(c) The Q function for the optimal policy is calculated as follows.
$Q(S_1, a_1) = \gamma V^*(S_1) = 16.58$
$Q(S_1, a_2) = V^*(S_1) = 18.42$
$Q(S_2, a_1) = 1 + \gamma V^*(S_2) = 20.42$
$Q(S_2, a_2) = V^*(S_2) = 21.58$

(iii)

Q	a_1	a_2
S_1	16.58	18.42
S_2	20.42	21.58

(iv)

current state	chosen action	new Q value
S_1	a_1	$0 + \gamma * 0 = 0$
S_1	a_2	$-1 + \gamma * 0 = -1$
S_2	a_1	$1 + \gamma * 0 = 1$

At this point, the Q-table looks like:

Q	a_1	a_2
S_1	0	-1
S_2	1	0

If the agent always chooses the current best action, it can have a policy where it always prefers a suboptimal action, e.g. a_1 in state S_2, so will never sufficiently explore action a_2. This means that $Q(S_2, a_2)$ will remain zero forever, instead of converging to the true value of 21.58. With exploration, the next few steps might look like this:

current state	chosen action	new Q value
S_2	a_2	$5 + \gamma * 0 = 5$
S_1	a_1	$0 + \gamma * 0 = 0$
S_1	a_2	$-1 + \gamma * 5 = 3.5$
S_2	a_1	$1 + \gamma * 5 = 5.5$
S_2	a_2	$5 + \gamma * 3.5 = 8.15$

Now we have this table:

Q	a_1	a_2
S_1	0	3.5
S_2	5.5	8.15

From this point on, the agent prefers action a_2 both in state S_1 and in state S_2. Further steps refine the Q value estimates, and in the limit, they converge to their true values.

current state	chosen action	new Q value
S_1	a_1	$0 + \gamma * 3.5 = 3.15$
S_1	a_2	$-1 + \gamma * 8.15 = 6.335$
S_2	a_1	$1 + \gamma * 8.15 = 8.335$
S_2	a_2	$5 + \gamma * 6.34 = 10.70$
$\cdots$	$\cdots$	$\cdots$