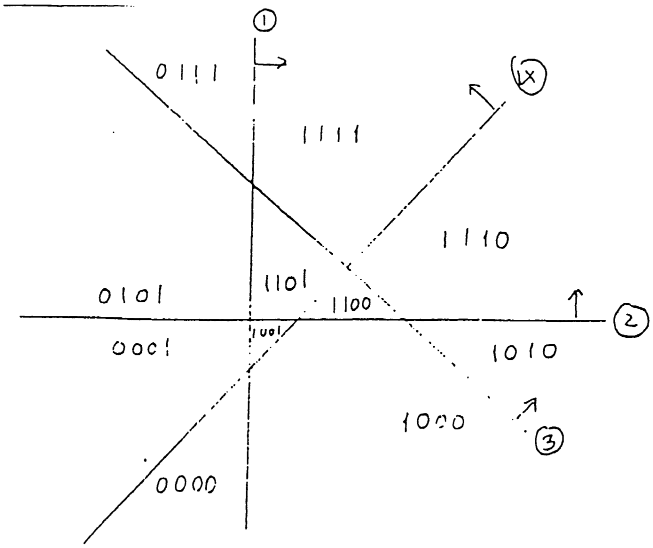
HW #3 Solns., EE556

#1)

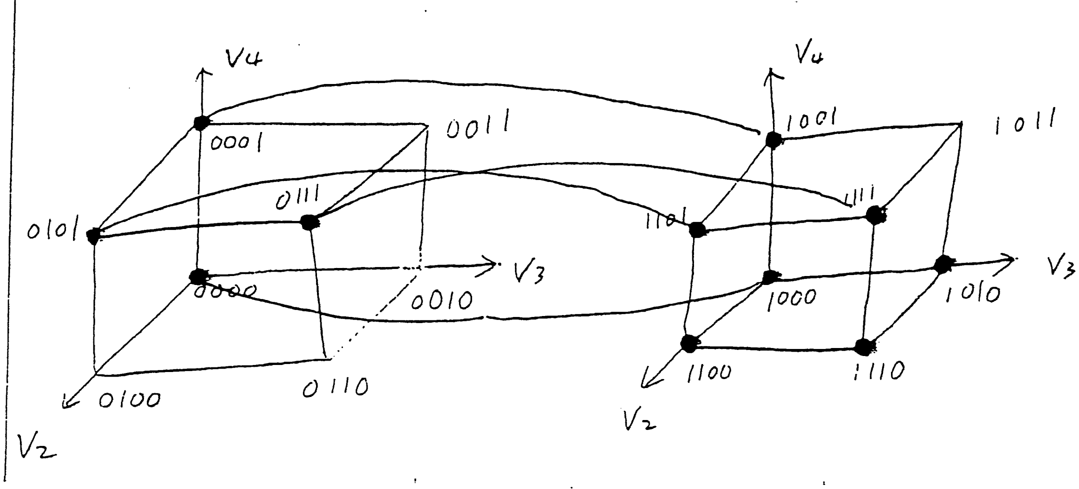
(a)



(b)

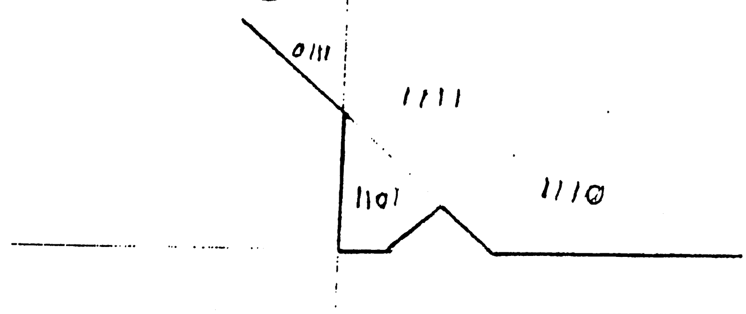
V_1	V_2	V_3	V_4	Class
0	0	0	0	✓
0	0	0	1	✓
0	0	1	0	IMP
0	0	1	1	IMP
0	1	0	0	IMP
0	1	0	1	✓
0	1	1	0	IMP
0	1	1	1	✓
1	0	0	0	✓
1	0	0	1	✓
1	0	1	0	✓
1	0	1	1	IMP
1	1	0	0	✓
1	1	0	1	✓
1	1	1	0	✓
1	1	1	1	✓

(c) $V_1 = 0$ $V_1 = 1$



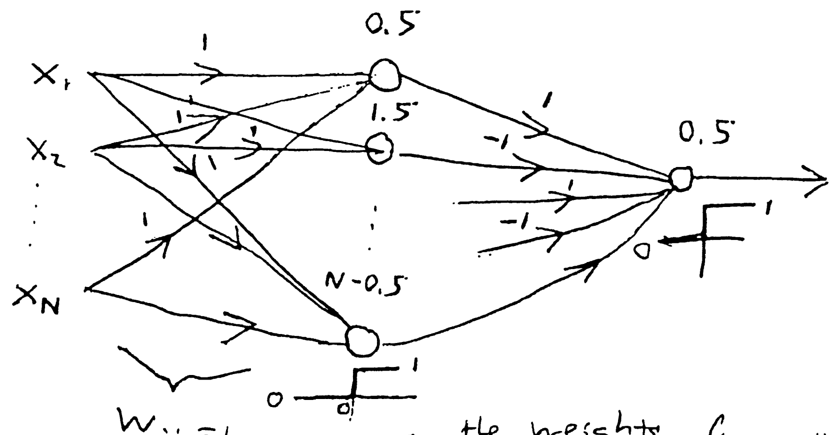
(d) Let $\omega_1 = \omega_2 = \omega_3 = \omega_4 = 1$, $\omega_5 = -2.5$

then $y = u(v_1 + v_2 + v_3 + v_4 - 2.5)$ classifies a region that is not convex nor is it the compliment of a convex region as shown below.



#2)

Consider the following network:



- the weights from input to each hidden neuron are all '1' ⇒ each hidden neuron counts the # of ones; neuron i then compares the count to the threshold $i - 0.5$ and uses a 0-1 threshold activation.
- "odd" neurons use "1" weights to the output; "even" neurons use "-1" weights to the output neuron.
- The output is "1" if the parity is odd & "0" if even.

#3) i) Space (storage) complexity:

# of adjustable weights from input to hidden layer is: $I \cdot J$

# of adjustable weights from hidden to output layer is: JK

Assuming thresholds are all at zero, total # weights =

$$I \cdot J + J \cdot K$$

During learning, however, one also has to store the gradients $\Longrightarrow$ total storage for weights + their gradients is: $2(IJ + JK)$.

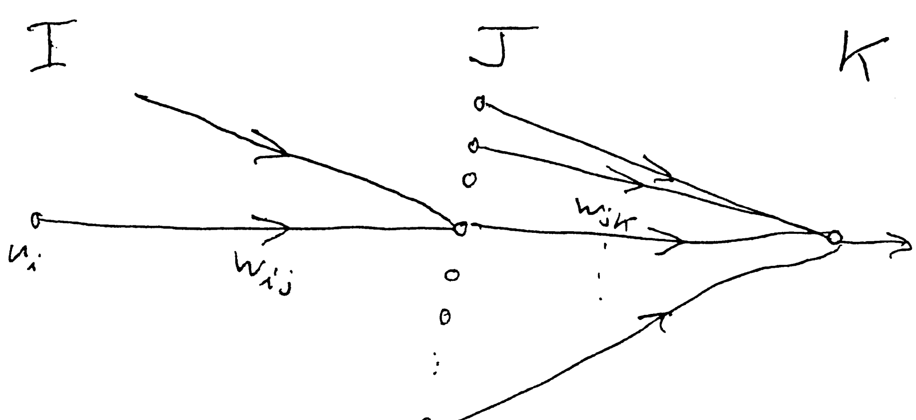
We also need to store the training patterns $\Longrightarrow$

$$(\underbrace{I}_{\text{input}} + \underbrace{1}_{\text{output dim.}})\underbrace{T}_{\text{\# of training examples}}$$

$\Longrightarrow$ space complexity $= 2(IJ + JK) + (I+1)T$

ii) <u>Computational complexity of BP</u>:

Refer to the BP derivation from lecture:



(assume single output, y)