1. (Haykin 3/e problem 1.3)
   a. The perceptron may be used to perform numerous logic function. Demonstrate the implementation of the binary logic function AND, OR, and COMPLEMENT. (Assume the input to AND and OR is two numbers drawn from {-1,1} representing false and true (respectively) and the input to COMPLEMENT is one number drawn from {-1,1}.

To implement AND, we can use weights $\vec{w} = [-0.5\,1\,1]^T$ for the bias, *x*, and *y* respectively. These values yield the following truth table:

AND

x	*y*	*bias*	*v, y*
1	1	1	1.5, 1
1	-1	1	-.5, -1
-1	1	1	-.5, -1
-1	-1	1	-2.5, -1

To implement OR, we can use weights $\vec{w} = [1\,1\,1]^T$ yielding this truth table:

OR

x	*y*	*bias*	*v, y*
1	1	1	3, 1
1	-1	1	1, 1
-1	1	1	1, 1
-1	-1	1	-1, -1

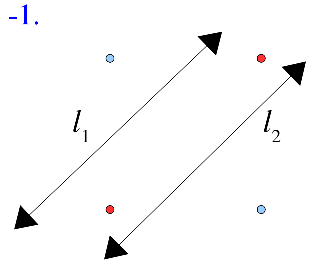
Finally, to implement NOT, we can use weights $\vec{w} = [0\,1]^T$ yielding truth table:

NOT

x	*bias*	*v, y*
1	1	1, 1
-1	1	-1, -1

b.A basic limitation of the perceptron is that it cannot implement the EXCLUSIVE-OR function. Explain the reason for this limitation.

Any line mapping (-1,-1) and (1,1) to value -1 passes either above both points or below them. If it passes above them, like l_1, then it will misclassify (1,-1) by mapping it to -1. If it passes below, like l_2, then it will misclassify (-1,1) by mapping it to -1.



2. Suppose the inputs to an SLP are three numbers x, y, and z having values drawn from {1,-1}. Find a weight vector w of length four (i.e., including a bias) that will yield the following output function: if x=1 and y=-1, or if x=-1 and y=1, or if x=1 and y=1, then the output is z. Otherwise, the output is -1. Describe how you found this weight vector.

   I solved this by making a matrix X from all cases of the inputs x, y, z, and bias, finding its pseudo-inverse (using `pinv` in Matlab) and multiplying by the desired response vector. That is:

$$X = \begin{bmatrix} 1 & -1 & 1 \\ 1 & -1 & -1 \\ -1 & 1 & 1 \\ -1 & 1 & -1 \\ 1 & 1 & 1 \\ 1 & 1 & -1 \end{bmatrix}, \vec{d} = \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}$$

   That yielded weights $\vec{w} = [0\,0\,1]^T$. And

$$X\,\vec{w} = \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \end{bmatrix} .$$

3. (Haykin 3/e problem 1.5)
   Equations (1.37) and (1.38) define the weight vector and bias of the Bayes classifier for a Gaussian environment. Determine the composition of this classifier for the case when the covariance matrix **C** is defined by

$$\mathbf{C} = \sigma^2 \mathbf{I}$$

   Where σ^2 is a constant and **I** is the identity matrix.

   The first observation is that the composition of a classifier comprises its weights and bias. Nothing else is necessary to be able to evaluate the classifier's result. With this in mind, Haykin's Equation (1.37) is $\vec{w} = \mathbf{C}^{-1}(\vec{\mu}_1 - \vec{\mu}_2)$ and since $\mathbf{C} = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}$, we have

   $\mathbf{C}^{-1} = \begin{bmatrix} 1/\sigma^2 & 0 \\ 0 & 1/\sigma^2 \end{bmatrix}$ and thus $\vec{w} = \dfrac{(\vec{\mu}_1 - \vec{\mu}_2)}{\sigma^2}$, and Equation (1.38) is

   $\vec{b} = \dfrac{1}{2}(\vec{\mu}_2^T \mathbf{C}^{-1} \vec{\mu}_2 - \vec{\mu}_1^T \mathbf{C}^{-1} \vec{\mu}_1)$, so $\vec{B} = \dfrac{1}{2\sigma^2}(\vec{\mu}_2^T \vec{\mu}_2 = \vec{\mu}_1^T \vec{\mu}_1)$.