THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

Machine Learning II
Homework #4
Due: *Oct-05-2020*

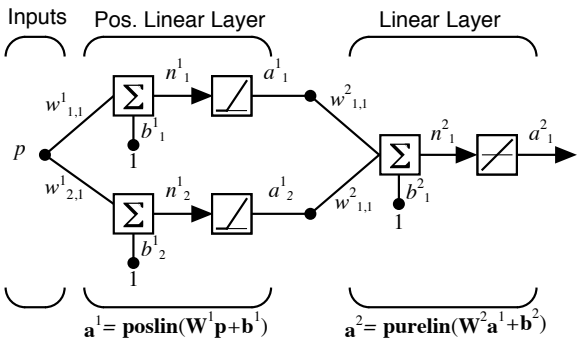
Homework 4:

- **Show ALL Work, Neatly and in Order.**
- **No** credit for Answers Without Work.
- Submit a single pdf file includes all of your solutions.
- **DO NOT** submit individual files or images.
- For coding questions, submit **ONE** .py file and include your comments.

Note 1: Use python to check your results that you draw.

E.1:

Sketch the response of the following network when the weights and biases are $\mathbf{W}^1 = \begin{bmatrix}-1 & 1\end{bmatrix}^T, \mathbf{b}^1 = \begin{bmatrix}0.5 & 1\end{bmatrix}^T, \mathbf{W}^2 = \begin{bmatrix}1 & 1\end{bmatrix}, \mathbf{b}^2 = [-1]$ for $-2 < p < 2$.



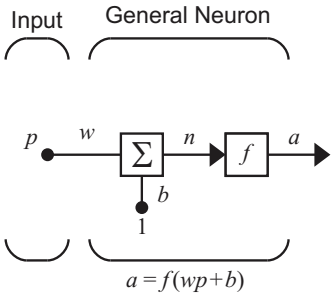
E.2:

Consider the following neuron.
Sketch the neuron response (plot a versus p for $-2 < p < 2$) for the following cases.

i. $w = 1 b = 1$ $f = hardlims$

ii. $w = -1 b = 1 f = hardlim$

iii. $w = 2 b = 3 f = purelin$

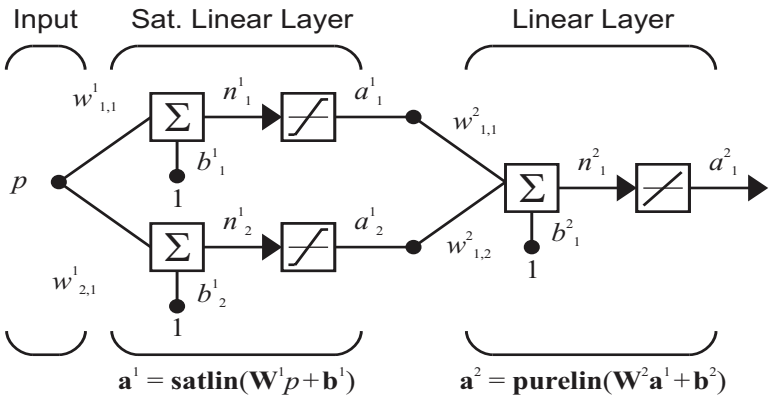


iv.$w = 2b = 3f = satlins$

v. $w = -2b = -1f = poslin$

E.3:

Consider the following neuron.



$$w^1_{1,1} = 2,\ w^1_{2,1} = 1,\ b^1_1 = 2,\ b^1_2 = -1,\ w^2_{1,1} = 1,\ w^2_{1,2} = -1,\ b^2_1 = 0$$

Sketch the neuron response (plot a versus p for $-3 < p < 3$) for the following cases (Use Satlins as Transfer function).

i. n^1_1

ii. a^1_1

iii.n^1_2

iv.a^1_2

v. n^2_1

vi. a^2_1

E.4:

We have four categories of vectors:

$$\text{Category I}: \left(\begin{bmatrix} -2 \\ 0 \end{bmatrix}, \begin{bmatrix} -2 \\ 2 \end{bmatrix} \right), \qquad \text{Category II:} \left(\begin{bmatrix} -1 \\ 3 \end{bmatrix}, \begin{bmatrix} 1 \\ 3 \end{bmatrix} \right)$$

$$\text{Category III}: \left(\begin{bmatrix} 2 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 \\ 2 \end{bmatrix} \right), \qquad \text{Category IV:} \left(\begin{bmatrix} -1 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right)$$

i. Design a two-neuron perceptron network (single layer) to recognize the four categories of vectors. Select the "BEST" decision boundaries, and explain what "BEST" means. Sketch the decision boundaries, and find the weights and biass. SHOW ALL WORK. Use the following targets:

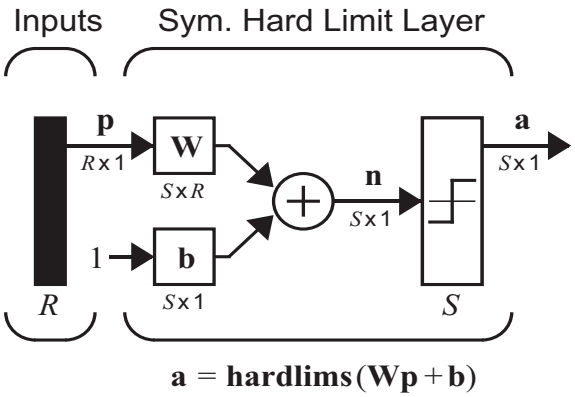
$$\text{Category I}: \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \text{Category II:} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \text{Category III:} \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \text{Category IV:} \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

ii. Draw the network diagram. Use the exact abbreviated notation that is used in class and textbook.

iii. Suppose the following vector is added to category IV. Perform one iteration of the perceptron rule with this vector. (Start with the weights you determined in part i.) $\begin{bmatrix} -3 \\ 1 \end{bmatrix}$

E.5:

Consider the following perceptron network.



$$\mathbf{W} = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix} -2 \\ 0 \end{bmatrix}$$

i. How many different classes can this network classify?

ii. Draw a diagram illustrating the regions corresponding to each class. Label each region with the corresponding network output.

iii. Calculate the network output for the following input. $p = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

iv. Plot the input from part iii in your diagram from part ii, and verify that it falls in the correctly labeled region.

Note 1: Write a Python Script to solve E.6. This problem is to implement the Perceprton learning rule. Please read the summary page in chapter 4 and implement the learning rule.

E.6: Python Exercise

The vectors in the ordered set defined below were obtained by measuring the weight and ear lengths of toy rabbits and bears in the Fuzzy Wuzzy Animal Factory. The target values indicate whether the respective input vector was taken from a rabbit (0) or a bear (1). The first element of the input vector is the weight of the toy, and the second element is the ear length. **Note**: You are not allowed to use any NN packages like Sklearn and etc, you need mostly numpy package.

$p_1 = \begin{bmatrix} 1 \\ 4 \end{bmatrix}, t_1 = 0; \quad p_2 = \begin{bmatrix} 1 \\ 5 \end{bmatrix}, t_2 = 0; \quad p_3 = \begin{bmatrix} 2 \\ 4 \end{bmatrix}, t_3 = 0; \quad p_4 = \begin{bmatrix} 2 \\ 5 \end{bmatrix}, t_4 = 0;$

$p_5 = \begin{bmatrix} 3 \\ 1 \end{bmatrix}, t_5 = 1; \quad p_6 = \begin{bmatrix} 3 \\ 2 \end{bmatrix}, t_6 = 1; \quad p_7 = \begin{bmatrix} 4 \\ 1 \end{bmatrix}, t_7 = 1; \quad p_8 = \begin{bmatrix} 4 \\ 2 \end{bmatrix}, t_8 = 1;$

i. i. Use Python to initialize and train a network to solve this ”practical” problem.

ii. Use Python test the resulting weight and bias values against the input vectors.

iii. Please plot the inputs and check your trained weight vector and validate your results by plotting the trained weight and bias.