

11040A Neural Networks

Assignment 2

Due Date: May 15, 2025 24:00:00

1. A McCulloch-Pitts neuron generates its output according to the following equation

$$y = \sigma(w_1x_1 + w_2x_2 + b)$$

where σ is a suitable step function, w_1 , w_2 , b are the weights and bias respectively, and x_1 , x_2 are the inputs.

- (a) The McCulloch-Pitts neuron model is motivated from a biological neuron.

- 1) Sketch a dendritic diagram of the *biological* neuron;
- 2) Label the terms of *cell body*, *axon*, *dendrites* and *synapses* on the diagram;
- 3) Describe briefly the functions and how the McCulloch-Pitts neuron models the biological neuron.

- (b) Design the McCulloch-Pitts neuron so that it functions as a

- 1) logic gate AND;
- 2) logic gate OR.

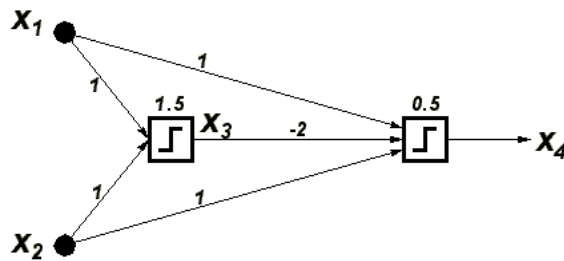
(Derive the rules for determining the values of w_1 , w_2 , b ; Discuss if it is possible to realize logic gate XOR, and if not, why?)

2. The following figure shows an MLP involving a single hidden neuron and jumping connections from the inputs to the output directly.

Suppose *step* function σ is:

$$\sigma(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases}$$

- (a) Construct a truth table for all variables x_1, x_2, x_3 and x_4 . Show that the network solves the XOR problem.
 (b) Plot the decision boundary of x_3 in $x_1 - x_2$ plane.
 (c) Plot the decision boundary of x_4 in $x_1 - x_2$ plane and explain how you derive it.
 (Note that your decision boundary should not be limited to the unit square only.)



Note that the number on top of each neuron is the threshold, and it is *subtracted* from the net input. For instance, the equation for x_3 is $x_3 = \text{step}(x_1 + x_2 - 1.5)$.

3. Write three popular activation functions. And draw them in rectangular plane coordinates.
4. Suppose there is an Artificial Neural Network (ANN) with d neurons in the input layer, h neurons in the hidden layer and c neurons in the output layer. The activation function in the output layer is softmax, while others are ReLU.
 - (a) Suppose the input is a row vector $x \in \mathbb{R}^{1 \times d}$, and you store the weights and biases in 2 matrices $W_1 \in \mathbb{R}^{s_{11} \times s_{12}}$, $W_2 \in \mathbb{R}^{s_{21} \times s_{22}}$ and 2 vectors $b_1 \in \mathbb{R}^{1 \times s_1}$, $b_2 \in \mathbb{R}^{1 \times s_2}$ respectively. Please describe the shapes/sizes of them (s_{11} , s_{12} , s_{21} , s_{22} , s_1 , s_2).
 - (b) Suppose the input is a row vector $x \in \mathbb{R}^{1 \times d}$, please write the function $f(x)$ that computes the output y .
 - (c) Usually, an ANN can compute the output parallelly. Suppose it takes as input a batch of row vectors, which form a matrix $X_{n \times d}$. Write the function $f(X_{n \times d})$ that computes the output $Y_{n \times c}$.

Programming part

Please write a report on the following questions.

The report should include your code & results and the explanation of the code.

(You can submit your programming files along with your report. Or you can organize your program into a markdown file or a Jupiter Notebook file as a report and submit it.)

1. Implement three ANNs with the following structure: (The ANNs take n 2-dimensional row vectors as input, and they form a matrix $X_{n \times 2}$)
 - a) 2 neurons in the input layer and 2 neurons in the output layer (with softmax activation function).
 - b) 2 neurons in the input layer, 8 neurons in the hidden layer (without activation function), and 2 neurons in the output layer (with softmax activation function).
 - c) 2 neurons in the input layer, 8 neurons in the hidden layer (with ReLU activation function), and 2 neurons in the output layer (with softmax activation function).
2. Implement a program to plot the 2-dimensional decision boundary of the ANNs you just implemented. (To plot the decision boundary, you can fill the input space with different colors, which represent different outputs of the ANN.)

Hint: For Python, you may import matplotlib library and use function `pyplot.contourf()`.
3. Initialize your ANNs in the proper way you want, show the decision boundaries, and discuss why it is generally preferable to use the multi-layer perceptron with non-linear activation functions in hidden layers, rather than ANNs with only one layer or those without activation functions in hidden layers.

(Note: You don't need to train your neural network because this is homework for next time.)