

HW2

Hands-on with Neural Networks

Jusvin Charles

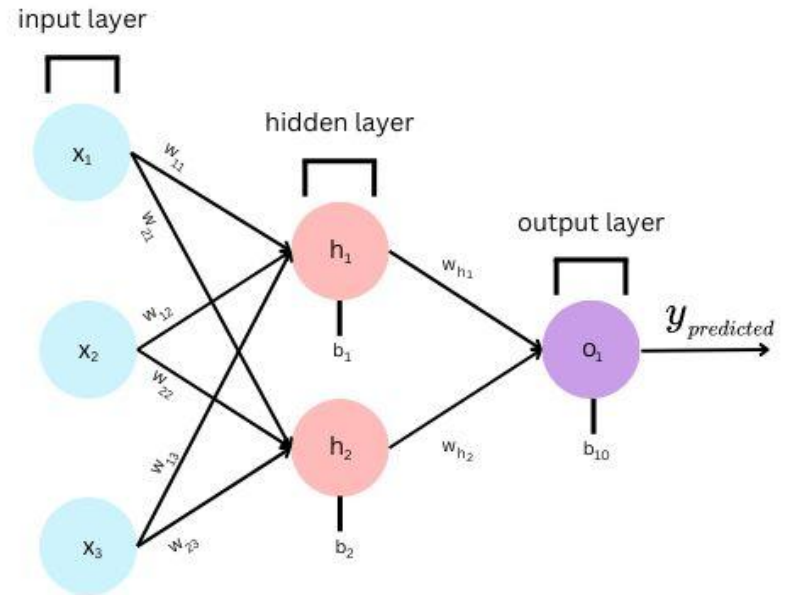
Problem Statement

You are presented with a problem of classifying cakes as chocolate or not chocolate based on certain features of the cake. You are given 3 features,

- X1: Sugar Content (Values ranging from 0-1)
- X2: Cocoa Content (Values ranging from 0-1)
- X3: Flour Content (Values ranging from 0-1)

Exercise 1: Hands-On

To solve this problem, consider the following multi-layer perceptron (MLP) with 3 inputs, 2 hidden neurons and 1 output neuron. The output neuron is binary, where 1 indicates that a cake is chocolate and 0 indicating otherwise. The activation function for the hidden neurons and output neurons are sigmoid. Given a set of weights, biases, and inputs, manually calculate their outputs and the MSE.



$$h_i = \sigma \left(\sum_{j=1}^n (w_{ij}x_j) + b_i \right)$$

$$o_1 = \sigma \left(\sum_{j=1}^n (w_{h_i}h_j) + b_{10} \right)$$

input layer



hidden layer



output layer



$y_{predicted}$

w_{11}

w_{21}

w_{12}

w_{22}

w_{13}

w_{23}

h_1

h_2

b_1

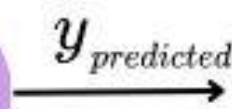
b_2

w_{h1}

w_{h2}

O_1

b_{10}



w11	w12	w13	w21	w22	w23	wh1	wh2	b1	b2	b10
0.14	0.78	0.33	0.91	0.47	0.56	0.65	0.38	1	1	1

x1	x2	x3	y
0.4	0.6	0.8	1
0.8	0.1	0.7	0
0.55	0.65	0.7	1

Exercise 1: Weights, Biases, Inputs & Outputs

Exercise 1: Questions

Answer the following questions in short sentences:

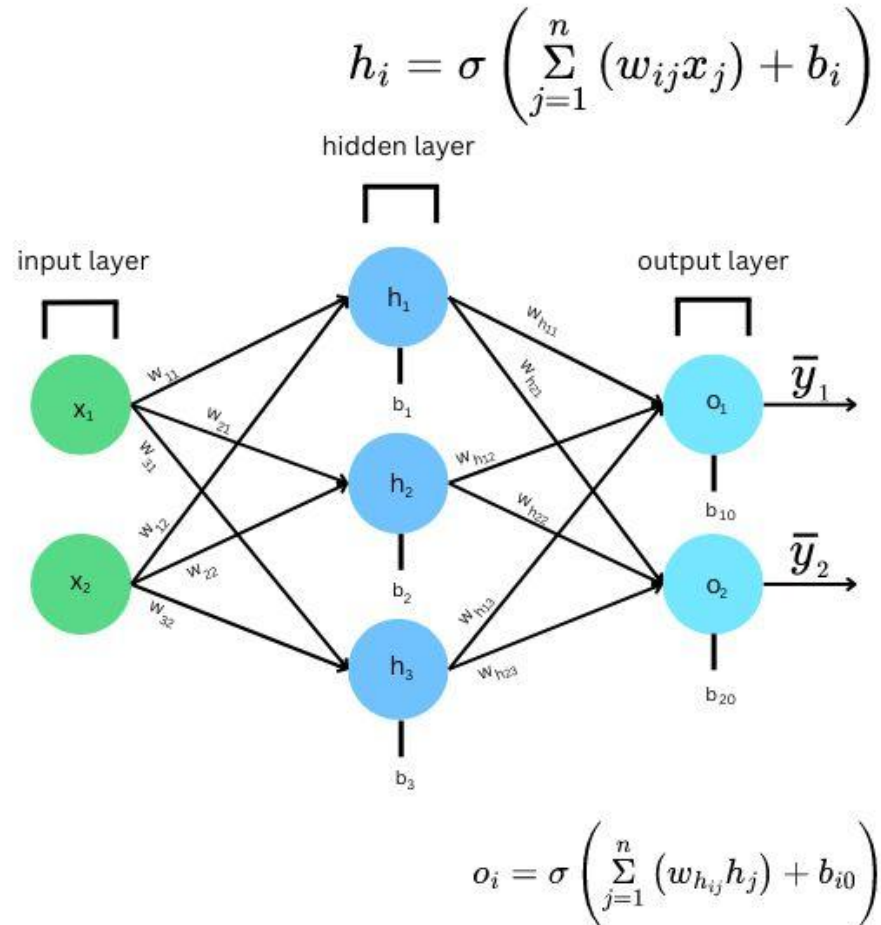
- How well did the following MLP perform?
- Why might the MLP perform as it did, despite the lack of weight updates?
- How would the performance change if backpropagation were introduced?

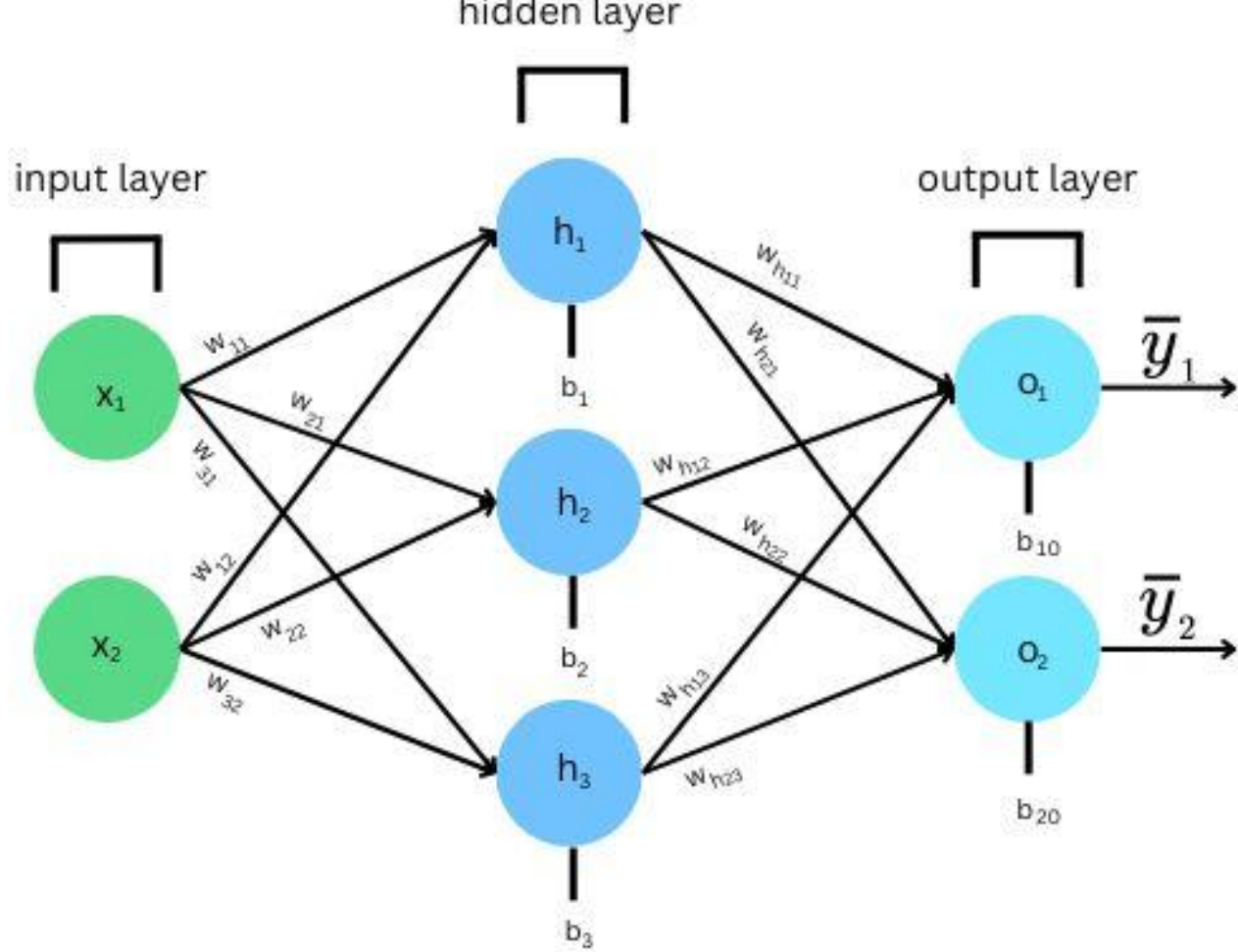
Exercise 1: Coding

Using only the numpy library, create the same multi-layered perceptron in python and initialize using the same parameters and print the MSE. Make sure to structure/generalize your code (either using functions and or classes).

Exercise 2: Hands-On

Let's say we're given the same cake problem and only want to classify using sugar and cocoa content and instead, we will be using another neural network, with 3 hidden neurons and 2 output neurons all using sigmoid as the activation function. If y_1 and y_2 output (0.3,0.7) then it's classified as chocolate cake, but if the output is (0.7,0.3) then it is classified as not chocolate cake. Given a set of weights, biases, and inputs, manually calculate their outputs and the MSE. Additionally, update the weights and biases twice using backpropagation.





$$\begin{pmatrix} n \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

w11	w12	w21	w22	w31	w32	wh11	wh12	wh13	wh21	wh22	wh23	b1	b2	b3	b10	b20
0.27	0.61	0.48	0.13	0.74	0.35	0.89	0.52	0.06	0.97	0.41	0.09	1	1	1	1	1

x1	x2	y1	y2	Outcome
0.3	0.6	0.3	0.7	Chocolate (1)
0.1	0.1	0.7	0.3	Not Chocolate (0)
0.5	0.7	0.3	0.7	Chocolate (1)
0.4	0.5	0.3	0.7	Chocolate (1)
0.2	0.5	0.7	0.3	Not Chocolate (0)

Exercise 2: Weights, Biases, Inputs & Outputs

Exercise 2: Coding Portion

Using only the numpy library, create the same simple neural network in python and initialize using the same parameters. Update its weights and biases 100 times and print out its new weights and MSE. Make sure to structure/generalize your code (either using functions and or classes).

Exercise 2: Questions

Answer the following questions in short sentences:

- How well did the neural network perform?
- Would the neural network perform better or worse if we used categorical cross entropy instead of MSE? What about a different activation function? Explain your reasoning.

Deliverables:

- Word/PDF File
 - Jupyter Notebook File
 - HTML File
-
- The hand-on portions and answers to questions should be in a word document/pdf file. Make sure to show your work! This portion may be typed or written but it must be legible!
 - The coding portions must be submitted with both html and .ipynb files.