Homework #2

1. Below are class-conditional PDF and prior probabilities for binary classification.

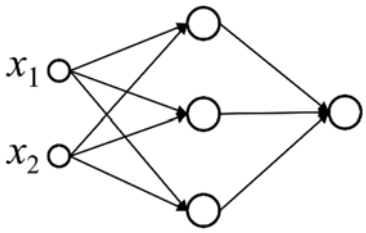
$$p(\mathbf{x} \mid S_1) = N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1),\ p(\mathbf{x} \mid S_2) = N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$$

$$\boldsymbol{\mu}_1 = [1,0]^T, \boldsymbol{\Sigma}_1 = \begin{bmatrix} 1.0 & -0.5 \\ -0.5 & 2.0 \end{bmatrix}, \boldsymbol{\mu}_2 = [0,0]^T, \boldsymbol{\Sigma}_2 = \begin{bmatrix} 1.0 & -0.5 \\ -0.5 & 2.0 \end{bmatrix}$$

$$P(S_1) = P(S_2) = 0.5$$

Determine and sketch Bayes decision boundary. Also sketch the decisio boundary when the prior probabilities are modified to $P(S_1) = 0.8,\ P(S_2) = 0.2$.

2. (hand-caculation) Consider a neural network for binary classification shown below.



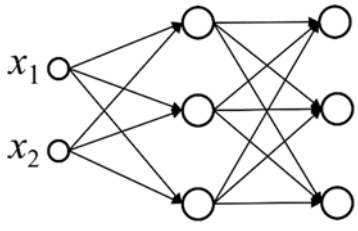
When model parameters are initialized as follow (c = constant),

$$\mathbf{W}^{(1)} = \begin{bmatrix} c & c \\ c & c \\ c & c \end{bmatrix}, \mathbf{b}^{(1)} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \mathbf{W}^{(2)} = \begin{bmatrix} c & c & c \end{bmatrix}, \mathbf{b}^{(2)} = 0$$

Prove that $\mathbf{W}^{(1)}$ and $\mathbf{b}^{(1)}$ are updated to the following forms after one forward propagation and one back-progapation. Note that this is the result shown in lecture notes.

$$\mathbf{W}^{(1)} = \begin{bmatrix} g_1 & g_2 \\ g_1 & g_2 \\ g_1 & g_2 \end{bmatrix}, \mathbf{b}^{(1)} = \begin{bmatrix} g_3 \\ g_3 \\ g_3 \end{bmatrix} \quad \text{where } g_i = \text{constant}$$

3. (electric calculator can be used) Consider the following neural network for 3-class classification.



The activation functions of the 1st layer and the 2nd layer are ReLu and softmax, respectively. Input vector is $\mathbf{x} = [1,\ -1]^T$. The model parameters are assumed to have the following values after sufficient iterations.

$$\mathbf{W}^{(1)}=\begin{bmatrix}0.0 & 0.5\\ 0.5 & -0.5\\ 1.0 & 0.0\end{bmatrix},\ \mathbf{b}^{(1)}=\begin{bmatrix}0\\ 0\\ 0\end{bmatrix},\ \mathbf{W}^{(2)}=\begin{bmatrix}0.5 & 0.0 & -1.0\\ -0.5 & 1.0 & -0.5\\ -0.5 & 0.5 & 0.5\end{bmatrix},\ \mathbf{b}^{(2)}=\begin{bmatrix}0\\ 0\\ 0\end{bmatrix}$$

a. Obtain output $\mathbf{y}=\mathbf{a}^{(2)}$ by applying a forward propagation using the above weight matrices and bias vectors. Your final answer should be in a form of length-3 vector.

b. Calculate cross-entropy-based cost function for each of two labels $\mathbf{y}^*=[1,\ 0,\ 0]^T$ or $[0,\ 0,\ 1]^T$.

c. Insted of single training sample assumed in **a** and **b**, now assume that the mini-batch size is 2 and total sample size is 20. How many gradient descent update will occur during 5 epochs ?

4. Let the weight matrix at a certain layer be initialized with

$$\mathbf{W}(0)=\begin{bmatrix}0.1\\ -0.1\end{bmatrix}$$

When the gradient of cost function with respect to **W** is computed as follow for the first three iterations,

$$\frac{\partial C}{\partial \mathbf{W}}(1)=\begin{bmatrix}1.0\\ 0.0\end{bmatrix},\ \frac{\partial C}{\partial \mathbf{W}}(2)=\begin{bmatrix}0.5\\ -0.5\end{bmatrix},\ \frac{\partial C}{\partial \mathbf{W}}(3)=\begin{bmatrix}0.0\\ 1.0\end{bmatrix}$$

Obtain **W**(3) (the updated weight matrix at the 3rd iteration), using the gradient descent with momentum. Assume that the momentum vector is initialized as $\mathbf{v}_m(0)=[0,\ 0]^T$, step size (learning rate) α = 0.1, and momentum coefficient β = 0.9.

5. Coding exercise for binary classification neural network (Open question)

- Set up Google Colab environment using your gmail account.
- Study the provided reference code 'shallowNN_reference.ipynb' implementing 1-hidden layer neural network based on Pytorch as much as possible.
- In the data file 'data_spiral.csv', each row represent one training sample, the first two columns represent 2D data, and the 3rd colume represent class label (0 or 1).

a. Upgrade the structure and parameters of the referfence neural network so that you can obtain more reasonable decision boundary. All the labeled data can be used for training only, i.e. you don't need to do validation test.

b. In your neural network obtained in **a**, remove non-linear activations of all hidden layers (i.e. affine transformation only in every layer). Show how the decision boundary is changed, and explain why using mathematical equations.