NYCU Introduction to Machine Learning, Homework 2

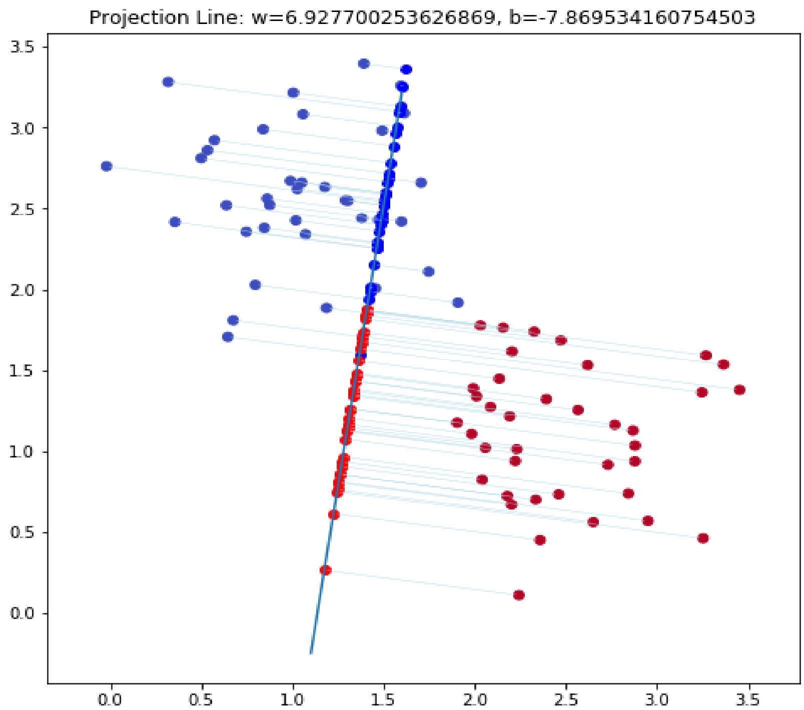
Deadline: Nov. 1, 23:59

Part. 1, Coding (60%):

In this coding assignment, you are required to implement Fisher's linear discriminant by using only NumPy, then train your model on the provided dataset, and evaluate the performance on testing data. Find the sample code and data on the GitHub page https://github.com/NCTU-VRDL/CS_CS20024/tree/main/HW2

Please note that only NumPy can be used to implement your model, you will get 0 point by calling sklearn.discriminant_analysis.LinearDiscriminantAnalysis.

1. (5%) Compute the mean vectors m_i (i=1, 2) of each 2 classes on **training data**
2. (5%) Compute the within-class scatter matrix S_W on **training data**
3. (5%) Compute the between-class scatter matrix S_B on **training data**
4. (5%) Compute the Fisher's linear discriminant w on **training data**
5. (20%) Project the **testing data** by Fisher's linear discriminant to get the class prediction by K-Nearest-Neighbor rule and report the accuracy score on **testing data** with K values from 1 to 5 (you should get accuracy over **0.88**)
6. (20%) Plot the **1) best projection line** on the **training data** and show the slope and intercept on the title *(you can choose any value of* ***intercept*** *for better visualization)* **2) colorize the data** with each class **3)** project all data points on your projection line. Your result should look like the below image (This image is for reference, not the answer)



Part. 2, Questions (40%):

Please write/type by yourself. DO NOT screenshot the solution from others.

(10%) 1. What's the difference between the Principle Component Analysis and Fisher’s Linear Discriminant?

(10%) 2. Please explain in detail how to extend the 2-class FLD into multi-class FLD (the number of classes is greater than two).

(6%) 3. By making use of Eq (1) ~ Eq (5), show that the Fisher criterion Eq (6) can be written in the form Eq (7).

$$y = \mathbf{w}^{\mathrm{T}}\mathbf{x} \qquad \text{Eq (1)}$$

$$\mathbf{m}_1 = \frac{1}{N_1}\sum_{n \in \mathcal{C}_1} \mathbf{x}_n \qquad \mathbf{m}_2 = \frac{1}{N_2}\sum_{n \in \mathcal{C}_2} \mathbf{x}_n \qquad \text{Eq (2)}$$

$$m_2 - m_1 = \mathbf{w}^{\mathrm{T}}(\mathbf{m}_2 - \mathbf{m}_1) \qquad \text{Eq (3)}$$

$$m_k = \mathbf{w}^{\mathrm{T}}\mathbf{m}_k \qquad \text{Eq (4)}$$

$$s_k^2 = \sum_{n \in \mathcal{C}_k} (y_n - m_k)^2 \qquad \text{Eq (5)}$$

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} \qquad \text{Eq (6)}$$

$$J(\mathbf{w}) = \frac{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}\mathbf{w}}{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{w}} \qquad \text{Eq (7)}$$

(7%) 4. Show the derivative of the error function Eq (8) with respect to the activation a_k for an output unit having a logistic sigmoid activation function satisfies Eq (9).

$$E(\mathbf{w}) = -\sum_{n=1}^{N} \{t_n \ln y_n + (1 - t_n)\ln(1 - y_n)\} \qquad \text{Eq (8)}$$

$$\frac{\partial E}{\partial a_k} = y_k - t_k \qquad \text{Eq (9)}$$

(7%) 5. Show that maximizing likelihood for a multiclass neural network model in which the network outputs have the interpretation $y_k(x, w) = p(t_k = 1 \mid x)$ is equivalent to the minimization of the cross-entropy error function Eq (10).

$$E(\mathbf{w}) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{kn} \ln y_k(\mathbf{x}_n, \mathbf{w}) \qquad \text{Eq (10)}$$