1 Implementation

The first step given by lab instruction is to illustrate the scatter plot of two 100 sample bi-variate Gaussian densities with distinct means $m_1 = \begin{bmatrix}0 & 5\end{bmatrix}^T$ and $m_2 = \begin{bmatrix}5 & 0\end{bmatrix}^T$ and identical covariance matrix $C = \begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix}$. The plot as in Fig. 1.

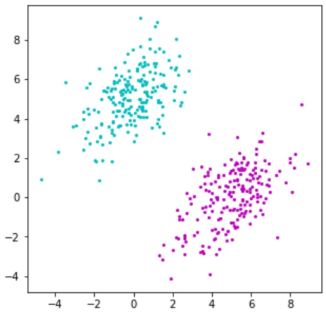

Figure 1: Scatter plot of $m_1 = \begin{bmatrix}0 & 5\end{bmatrix}^T$ and $m_2 = \begin{bmatrix}5 & 0\end{bmatrix}^T$

From the data that display in Fig. 1, when it implement complete it show the results of training a perceptron as the graph in Fig. 2. This graph look similar to the example graph in lab instruction that expected. In the Fig. 2, the result training is implement by use learning rate $\alpha = 0.002$. From equation, $w^{(\tau+1)} = w^{(\tau)} + \alpha y^{(\tau)} x^{(\tau)}$

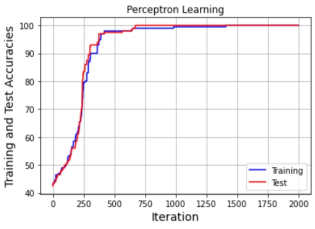

Figure 2: Result training of $m_1 = \begin{bmatrix}0 & 5\end{bmatrix}^T$, $m_2 = \begin{bmatrix}5 & 0\end{bmatrix}^T$ and $\alpha = 0.002$

Next, in the Fig. 3 and Fig. 4 is the result training that change the learning rate to $\alpha = 0.02$ and $\alpha = 0.0002$ respectively. From Fig. 3, when the learning rate is bigger it make algorithm take huge steps that change the direction of weight vector (w) more than the learning rate $\alpha = 0.002$ in Fig. 2. However, when the learning rate change to $\alpha = 0.0002$ as in Fig. 4, the direction of weight vector will change in small different of direction of weight vector.

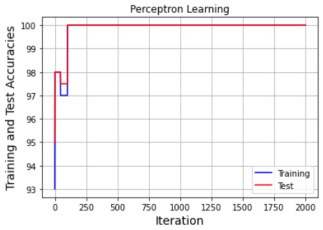

Figure 3: Result training of $m_1 = \begin{bmatrix}0 & 5\end{bmatrix}^T$, $m_2 = \begin{bmatrix}5 & 0\end{bmatrix}^T$ and $\alpha = 0.02$

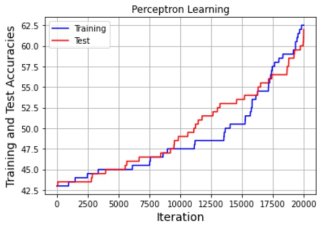

Figure 4: Result training of $m_1 = \begin{bmatrix}0 & 5\end{bmatrix}^T$, $m_2 = \begin{bmatrix}5 & 0\end{bmatrix}^T$ and $\alpha = 0.0002$

In Fig. 5, the distinct means is change to $m_1 = \begin{bmatrix}3 & 5\end{bmatrix}^T$ and $m_2 = \begin{bmatrix}5 & 3\end{bmatrix}^T$. It make the data overlap to each other data class. That make the perceptron algorithm harder to classify each two data as the result of this training in Fig. 6.

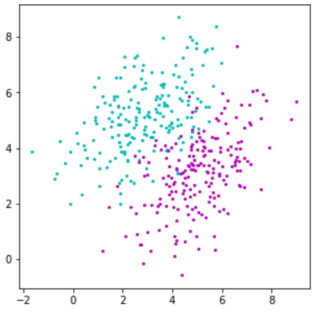

Figure 5: Scatter plot of $m_1 = \begin{bmatrix}3 & 5\end{bmatrix}^T$ and $m_2 = \begin{bmatrix}5 & 3\end{bmatrix}^T$

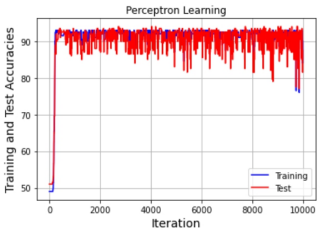

Figure 6: Result training of $m_1 = \begin{bmatrix}3 & 5\end{bmatrix}^T$, $m_2 = \begin{bmatrix}5 & 3\end{bmatrix}^T$ and $\alpha = 0.002$

2 Compare solutions of my own perceptron to scikitlearn

From my code implementation between my own perceptron algorithm and scikitlearn perceptron

algorithm. Scikitlearn algorithm seem have accuracy on test set better than my own algorithm. Therefore scikitlearn perceptron algorithm seem have performance better than my own algorithm.

3 Consider the problem with specific means

In this problem means is change to $m_1 = \begin{bmatrix}2.5 & 2.5\end{bmatrix}^T$ *and* $m_1 = \begin{bmatrix}10.0 & 10.0\end{bmatrix}^T$ *with the covariance matrices equal and the same as before. Does the perceptron as implemented solve this problem? If not what modification is needed to help solve this problem?*

• When the new means is change and implement it. The scatter plot of data and it result training is illustrate as in Fig. 7 and Fig. 8. That because the weight vector is at the origin it impossible to classify class of this data.

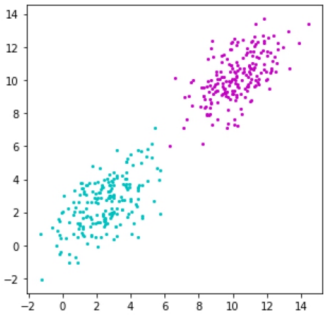

Figure 7: Scatter plot of $m_1 = \begin{bmatrix}2.5 & 2.5\end{bmatrix}^T$ and $m_2 = \begin{bmatrix}10.0 & 10.0\end{bmatrix}^T$

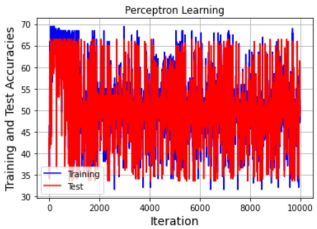

Figure 8: Result training of $m_1 = \begin{bmatrix}2.5 & 2.5\end{bmatrix}^T$ and $m_2 = \begin{bmatrix}10.0 & 10.0\end{bmatrix}^T$

Now, it have other method to fixed this case of data to classify it by change the weight vector into 3-dimensional vector and change data float out of xy-plane as show in Fig. 9. After that the Percentage Correct After Training is back to 100% and result training is show in Fig. 10.

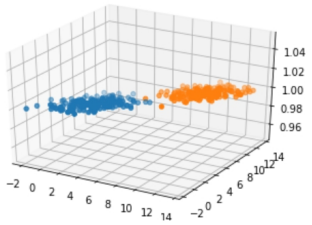

Figure 9: 3D Scatter plot of $m_1 = \begin{bmatrix}2.5 & 2.5\end{bmatrix}^T$ and $m_2 = \begin{bmatrix}10.0 & 10.0\end{bmatrix}^T$

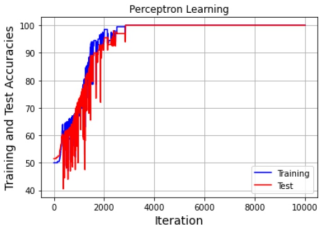

Figure 10: Result training of $m_1 = \begin{bmatrix}2.5 & 2.5\end{bmatrix}^T$ and $m_2 = \begin{bmatrix}10.0 & 10.0\end{bmatrix}^T$ after fixed

4 Try on a dataset from UCL repository

For this point, I use seeds dataset from UCL repository[1]. I prepared the dataset to classify Rosa seeds and Canadian seeds by use it feature that it is area (A) and perimeter (P) of the seeds. The scatter plot of the data represent Fig. 11 and the result training show in Fig. 12. However than dataset isn't have the performance results on this dataset that is binary classification by other researchers to compare with this Perceptron algorithm.

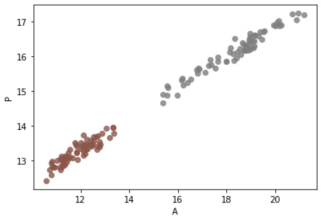

Figure 11: Scatter plot of seeds dataset

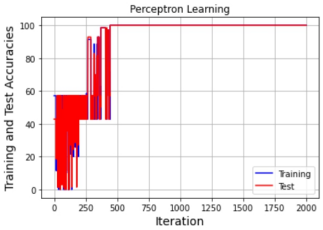

Figure 12: Result training of seeds dataset

Reference

[1] seeds dataset from UCL repository
url :https://archive.ics.uci.edu/ml/datasets/seeds