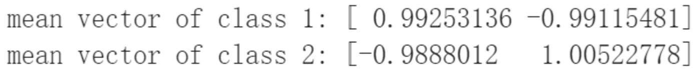
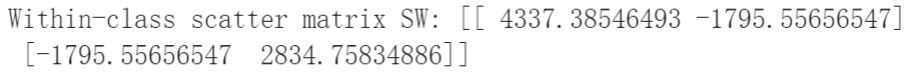
Part. 1, Coding (60%):

1. (5%) Compute the mean vectors m_i (i=1, 2) of each 2 classes on <u>training data</u>



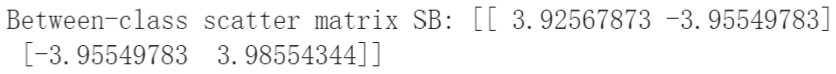
```
mean vector of class 1: [ 0.99253136 -0.99115481]
mean vector of class 2: [-0.9888012   1.00522778]
```

2. (5%) Compute the within-class scatter matrix $S_W$on <u>training data</u>



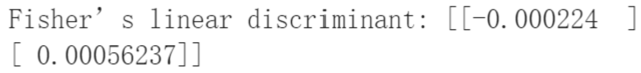
```
Within-class scatter matrix SW: [[ 4337.38546493 -1795.55656547]
 [-1795.55656547  2834.75834886]]
```

3. (5%) Compute the between-class scatter matrix S_B on <u>training data</u>



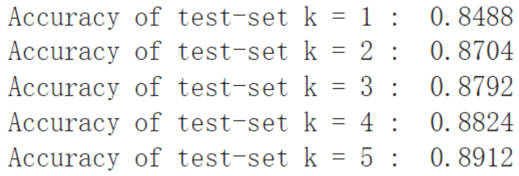
```
Between-class scatter matrix SB: [[ 3.92567873 -3.95549783]
 [-3.95549783  3.98554344]]
```

4. (5%) Compute the Fisher’ s linear discriminant $_w$ on <u>training data</u>



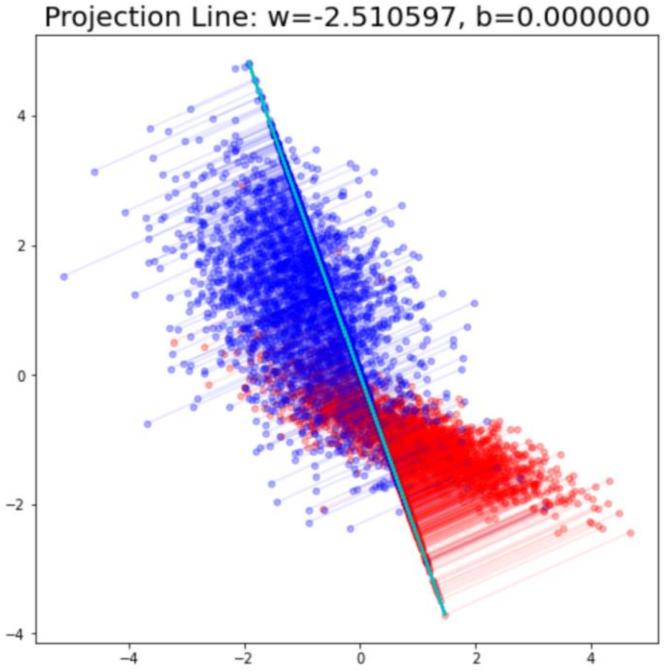
```
Fisher’ s linear discriminant: [[-0.000224   ]
 [ 0.00056237]]
```

5. (20%) Project the <u>testing data</u> by Fisher’ s linear discriminant to get the class prediction by K-Nearest-Neighbor rule and report the accuracy score on <u>testing data</u> with K values from 1 to 5 (you should get accuracy over 0.88)



```
Accuracy of test-set k = 1 :  0.8488
Accuracy of test-set k = 2 :  0.8704
Accuracy of test-set k = 3 :  0.8792
Accuracy of test-set k = 4 :  0.8824
Accuracy of test-set k = 5 :  0.8912
```

6. (20%) Plot the **1) best projection line** on the <u>training data</u> and <u>show the slope and intercept on the title</u> *(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.



Projection Line: w=-2.510597, b=0.000000

Part. 2, Questions (40%):

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

Principle Component Analysis (shorten as PCA) is an unsupervised dimentionality reduction technique which ignores class labels. PCA projects the higher dimensional data into lower dimension while preserving the significant information.
While Fisher’s Linear Discriminant (shorten as FLD) is a supervised dimentionality reduction technique that acheives data classification simutaneously. FLD maximize the separation among classes while minimize the separation between classes.
Both of them looks for linear combinations of the features that explain the data best.
PCA is efficient in terms of detection of faults while FLD is optimal in terms of diagnosing the faults.

(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).

To extend 2-class FLD into multi-class FLD, assume that there are N classes instead of 2, the i-th class is denoted as $C_i, i \in (1, N)$.

Then we can redefine S_w, S_b as $S_w = \Sigma_{i=1}^{N} S_{wi}$, where $S_{wi} = \Sigma_{x \in X_i} \left(x - \mu_i\right)\left(x - \mu_i\right)^T$, and $S_b = \Sigma_{i=1}^{N} C_i \left(\mu_i - \mu\right)\left(\mu_i - \mu\right)^T$, μ_i is the mean of each class i.
The objective $J(w)$ from 2-class cannot extend to multi-class directly, so let the objective be $J(w) = Tr\left(WS_wW^T\right)^{-1}\left(WS_bW^T\right)$. The columns of the optimal W are the eigenvectors of $S_w^{-1}S_b$ that correspond to the $D´$ $(D´ < N - 1)$ largest eigenvalues.
With the formulas above, we can find the projection of the multi-class FLD.

(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).

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

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

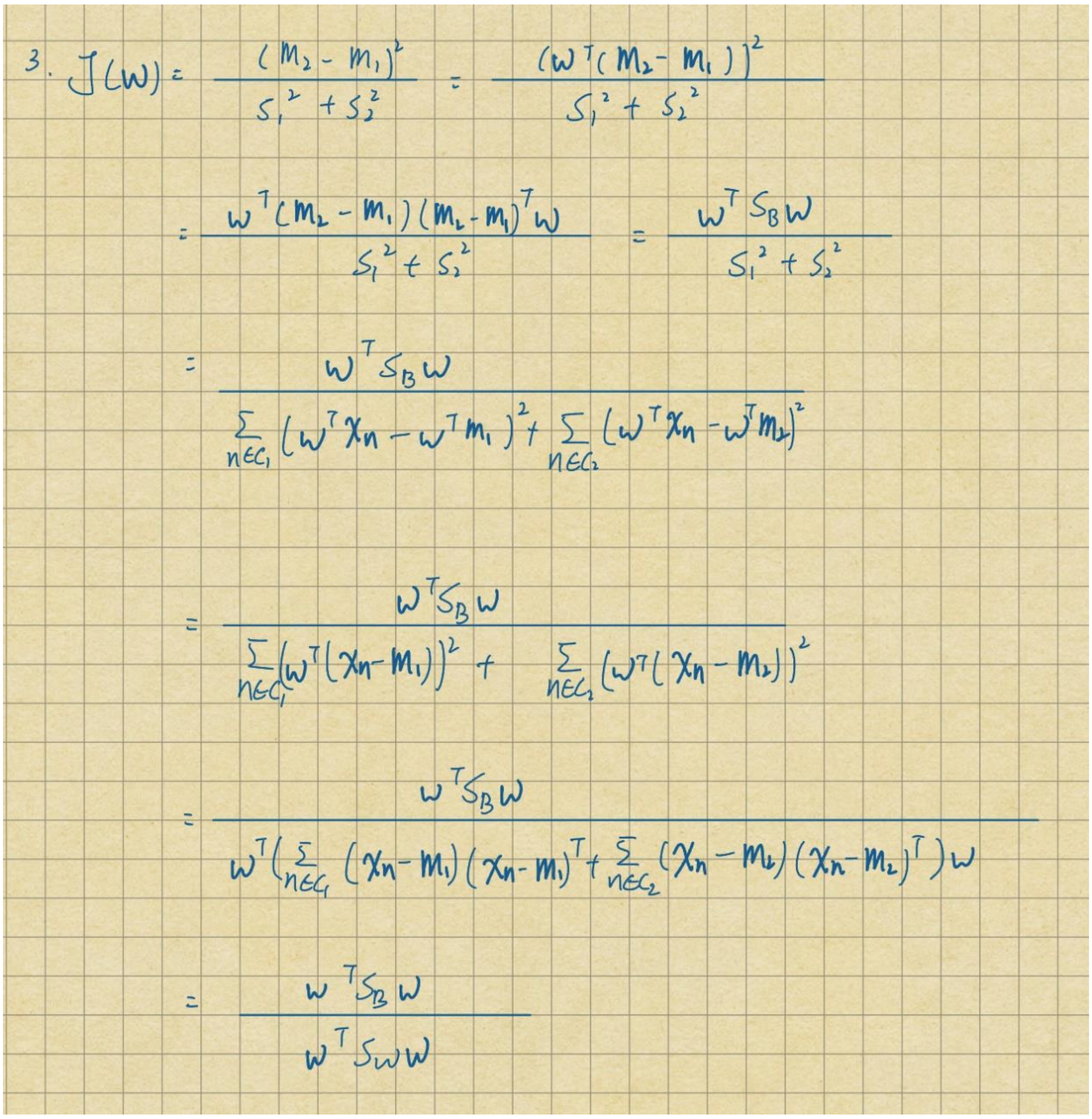
$$m_2 - m_1 = \mathbf{w}^{\mathrm{T}}(\mathbf{m}_2 - \mathbf{m}_1) \qquad \text{Eq (3)}$$

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

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

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

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



3. $$J(W) = \frac{(M_2 - M_1)^2}{S_1^2 + S_2^2} = \frac{(W^T(M_2 - M_1))^2}{S_1^2 + S_2^2}$$

$$= \frac{W^T(M_2 - M_1)(M_2 - M_1)^T W}{S_1^2 + S_2^2} = \frac{W^T S_B W}{S_1^2 + S_2^2}$$

$$= \frac{W^T S_B W}{\sum_{n\in C_1}(W^T X_n - W^T M_1)^2 + \sum_{n\in C_2}(W^T X_n - W^T M_2)^2}$$

$$= \frac{W^T S_B W}{\sum_{n\in C_1}(W^T(X_n - M_1))^2 + \sum_{n\in C_2}(W^T(X_n - M_2))^2}$$

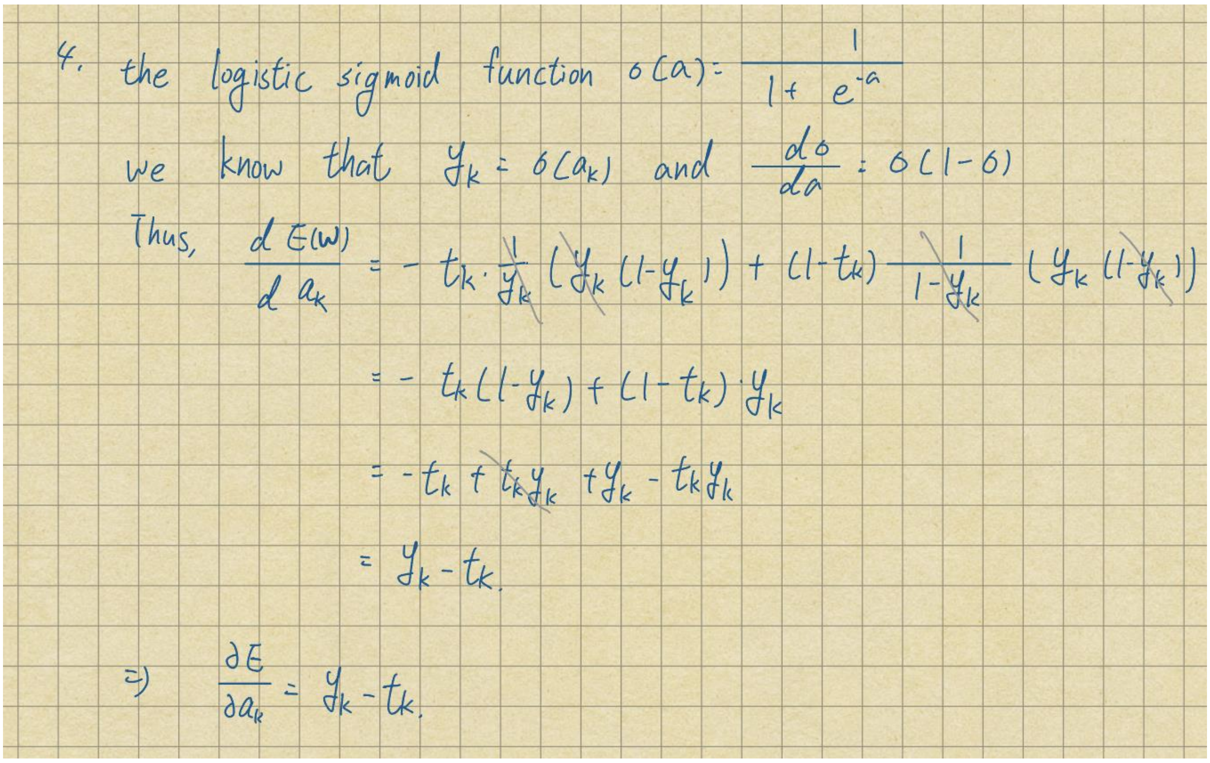
$$= \frac{W^T S_B W}{W^T\left(\sum_{n\in C_1}(X_n - M_1)(X_n - M_1)^T + \sum_{n\in C_2}(X_n - M_2)(X_n - M_2)^T\right)W}$$

$$= \frac{W^T S_B W}{W^T S_W W}$$

(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).

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

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



4. the logistic sigmoid function $\sigma(a) = \frac{1}{1+e^{-a}}$

we know that $y_k = \sigma(a_k)$ and $\frac{d\sigma}{da} = \sigma(1-\sigma)$

Thus, $\frac{d\,E(w)}{d\,a_k} = -t_k \cdot \frac{1}{\cancel{y_k}}(\cancel{y_k}(1-y_k)) + (1-t_k)\frac{1}{\cancel{1-y_k}}(y_k\cancel{(1-y_k)})$

$= -t_k(1-y_k) + (1-t_k)\cdot y_k$

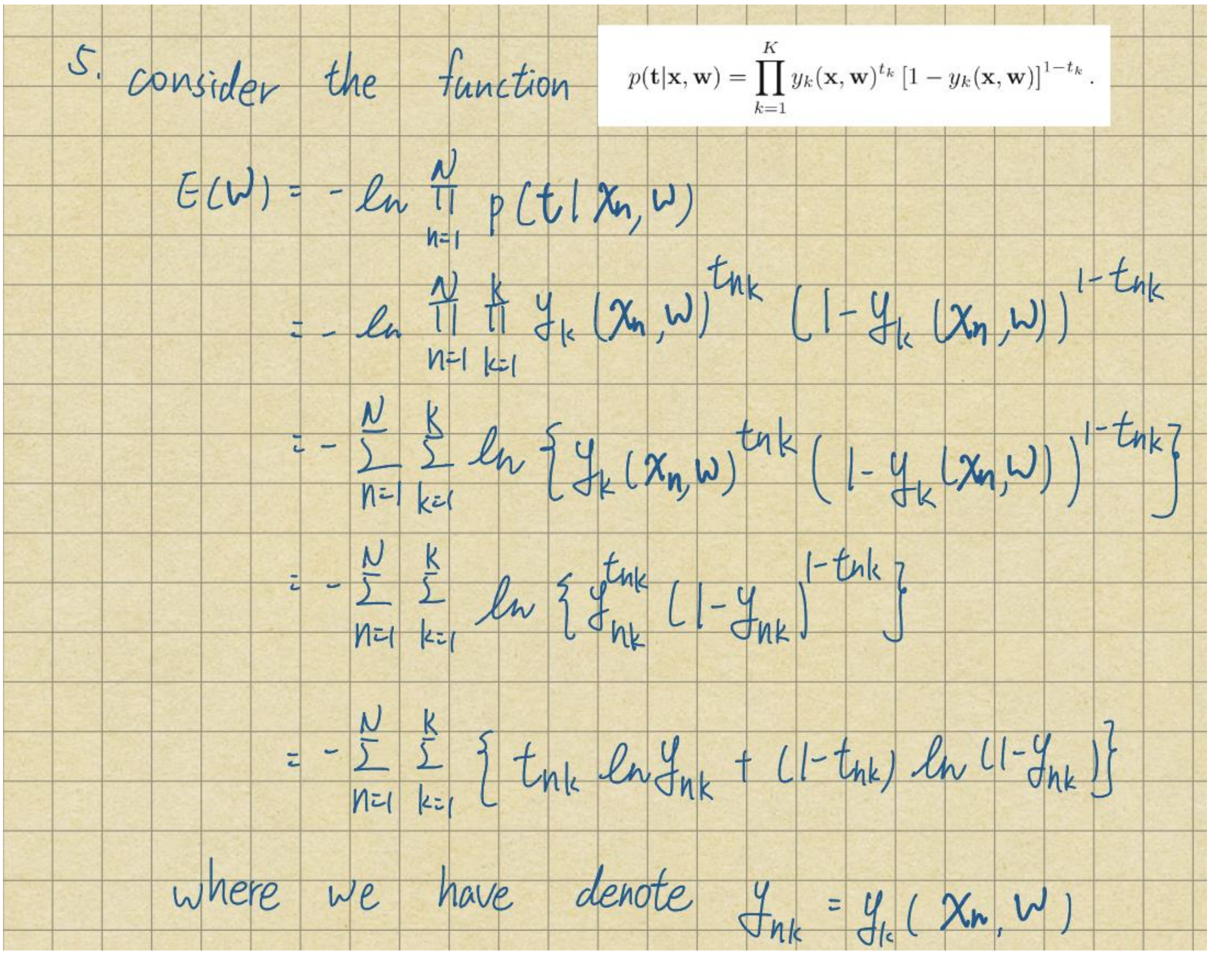
$= -t_k + \cancel{t_k y_k} + y_k - t_k y_k$

$= y_k - t_k.$

$\Rightarrow \frac{\partial E}{\partial a_k} = y_k - t_k.$

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

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



5. consider the function $p(\mathbf{t}|\mathbf{x}, \mathbf{w}) = \prod_{k=1}^{K} y_k(\mathbf{x}, \mathbf{w})^{t_k} \left[1 - y_k(\mathbf{x}, \mathbf{w})\right]^{1-t_k}.$

$$E(W) = -\ln \prod_{n=1}^{N} p(t | x_n, w)$$

$$= -\ln \prod_{n=1}^{N} \prod_{k=1}^{K} y_k(x_n, w)^{t_{nk}} (1 - y_k(x_n, w))^{1-t_{nk}}$$

$$= -\sum_{n=1}^{N} \sum_{k=1}^{K} \ln \left\{ y_k(x_n, w)^{t_{nk}} (1 - y_k(x_n, w))^{1-t_{nk}} \right\}$$

$$= -\sum_{n=1}^{N} \sum_{k=1}^{K} \ln \left\{ y_{nk}^{t_{nk}} (1 - y_{nk})^{1-t_{nk}} \right\}$$

$$= -\sum_{n=1}^{N} \sum_{k=1}^{K} \left\{ t_{nk} \ln y_{nk} + (1 - t_{nk}) \ln (1 - y_{nk}) \right\}$$

where we have denote $y_{nk} = y_k(X_n, W)$