COM S 573: Machine Learning
Homework #1

$$
\begin{aligned}
f_2(\lambda u + (1-\lambda)v) &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\lambda u_i + (1-\lambda)v_i)^2 - \left(\frac{1}{n}\sum_{i=1}^{n}(\lambda u_i + (1-\lambda)v_i)\right)^2} \\
&\le \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\lambda u_i)^2 + (1-\lambda)^2 v_i^2 - \left(\frac{1}{n}\sum_{i=1}^{n}(\lambda u_i + (1-\lambda)v_i)\right)^2} \\
&= \sqrt{\frac{1}{n}\sum_{i=1}^{n}\lambda^2 u_i^2 + (1-\lambda)^2 v_i^2 + 2\lambda(1-\lambda)u_i v_i - \left(\frac{\lambda}{n}\sum_{i=1}^{n}u_i + \frac{1-\lambda}{n}\sum_{i=1}^{n}v_i\right)^2}
\end{aligned}
$$

$$
\begin{aligned}
&\le \sqrt{\lambda\frac{1}{n}\sum_{i=1}^{n}u_i^2 + (1-\lambda)\frac{1}{n}\sum_{i=1}^{n}v_i^2 - \lambda(1-\lambda)\left(\frac{1}{n}\sum_{i=1}^{n}u_i + \frac{1}{n}\sum_{i=1}^{n}v_i\right)^2} \\
&= \sqrt{\lambda f_2(u) + (1-\lambda)f_2(v) - \lambda(1-\lambda)|u-v|^2}
\end{aligned}
$$

1. (10 points) Consider the perceptron in two dimensions: $h(\boldsymbol{x}) = \text{sign}(\boldsymbol{w}^T\boldsymbol{x})$ where $\boldsymbol{w} = [w_0, w_1, w_2]^T$ and $\boldsymbol{x} = [1, x_1, x_2]^T$. Technically, $\boldsymbol{x}$ has three coordinates, but we call this perceptron two-dimensional because the first coordinate is fixed at 1.

   (a) Show that the regions on the plane where $h(\boldsymbol{x}) = +1$ and $h(\boldsymbol{x}) = -1$ are separated by a line. If we express this line by the equation $x_2 = ax_1 + b$, what are the slope a and intercept b in terms of w_0, w_1, w_2?
   Sol: We know that $h(\boldsymbol{x}) = \text{sign}(\boldsymbol{w}^T\boldsymbol{x})$ which implies

$$
h(\boldsymbol{x}) = \begin{cases} +1 & \text{if sign}(\boldsymbol{w}^T\boldsymbol{x}) > 0 \\ -1 & \text{if sign}(\boldsymbol{w}^T\boldsymbol{x}) < 0 \end{cases}
$$

   Therefore, $h(\boldsymbol{x}) = \text{sign}(\boldsymbol{w}^T\boldsymbol{x}) = 0$ is a decision boundary which means there exists a line $\boldsymbol{w}^T\boldsymbol{x} = w_0 + w_1x_1 + w_2x_2 = 0$.

$$
w_2x_2 = -w_0 - w_1x_1 \implies x_2 = -\frac{w_1}{w_2}x_1 - \frac{w_0}{w_2}
$$

$$
\therefore \quad a = -\frac{w_1}{w_2} \quad b = -\frac{w_0}{w_2}
$$

   (b) Draw pictures for the cases $\boldsymbol{w} = [1, 2, 3]^T$ and $\boldsymbol{w} = -[1, 2, 3]^T$. Label the positive and negative prediction regions on the pictures.
   Sol:

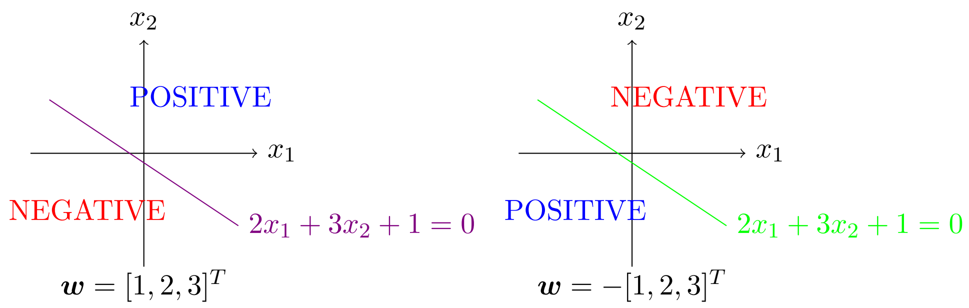

Figure 1: Comparison of perceptron with opposite weights

2. (30 points) In logistic regression (labels are {1, -1}), the objective function can be written as

$$E(w) = \frac{1}{N}\sum_{n=1}^{N} \ln\left(1 + e^{-y_n w^T x_n}\right).$$

Please

(a) (10 points) Compute the first-order derivative $\nabla E(w)$. You will need to provide the intermediate steps of derivation.

Sol:

$$\begin{aligned}\nabla E(w) &= \frac{1}{N}\sum_{n=1}^{N} \nabla_w \ln\left(1 + e^{-y_n w^T x_n}\right) \\ &= \frac{1}{N}\sum_{n=1}^{N} \frac{e^{-y_n w^T x_n}}{\left(1 + e^{-y_n w^T x_n}\right)}(-y_n x_n) \\ &= \frac{1}{N}\sum_{n=1}^{N} \left(\frac{-y_n e^{-y_n w^T x_n}}{1 + e^{-y_n w^T x_n}}\right) x_n\end{aligned}$$

$$\nabla E(w) = \begin{bmatrix} \frac{\partial E(w)}{\partial w_1} \\ \frac{\partial E(w)}{\partial w_2} \\ \vdots \\ \frac{\partial E(w)}{\partial w_m} \end{bmatrix} = \frac{1}{N}\begin{bmatrix} \sum_{n=1}^{N} \left(\frac{-y_n e^{-y_n w^T x_n}}{1+e^{-y_n w^T x_n}}\right) x_n(1) \\ \sum_{n=1}^{N} \left(\frac{-y_n e^{-y_n w^T x_n}}{1+e^{-y_n w^T x_n}}\right) x_n(2) \\ \vdots \\ \sum_{n=1}^{N} \left(\frac{-y_n e^{-y_n w^T x_n}}{1+e^{-y_n w^T x_n}}\right) x_n(m) \end{bmatrix}$$

(b) (10 points) Once the optimal w is obtain, it will be used to make predictions as follows:

$$\text{Predicted class of } x = \begin{cases} 1 & \text{if } \theta(w^T x) \geq 0.5 \\ -1 & \text{if } \theta(w^T x) < 0.5 \end{cases}$$

where $\theta(z) = \frac{1}{1+e^{-z}}$.

Explain why the decision boundary of logistic regression is still linear, though the linear signal $w^T x$ is passed through a nonlinear function θ to compute the outcome of prediction.

Sol:

$$\begin{aligned}
\theta(w^T x) &\geq 0.5 \\
\frac{1}{1+e^{-w^T x}} &\geq 0.5 \\
1+e^{-w^T x} &\leq 2 \\
e^{-w^T x} &\leq 1 \\
-w^T x &\leq 0 \\
w^T x &\geq 0
\end{aligned}$$

Similarly, for $\theta(w^T x) < 0.5$, we get $w^T x < 0$. Therefore, $w^T x = 0$ is a decision boundary separating two classes, which is linear in x for a fixed w.

(c) (5 points) Is the decision boundary still linear if the prediction rule is changed to the following? Justify briefly.

$$\text{Predicted class of } x = \begin{cases} 1 & \text{if } \theta(w^T x) \geq 0.9 \\ -1 & \text{if } \theta(w^T x) < 0.9 \end{cases}$$

Sol:

$$\begin{aligned}
\theta(w^T x) &\geq 0.9 \\
\frac{1}{1+e^{-w^T x}} &\geq 0.9 \\
1+e^{-w^T x} &\leq \frac{10}{9} \\
e^{-w^T x} &\leq \frac{1}{9} \\
-w^T x &\leq -\ln 9 \\
w^T x &\geq \ln 9
\end{aligned}$$

Similarly, for $\theta(w^T x) < 0.9$, we get $w^T x < \ln 9$. Therefore, $w^T x = \ln 9$ is a decision boundary separating two classes, which is linear in x for a fixed w.

(d) (5 points) In light of your answers to the above two questions, what is the essential property of logistic regression that results in the linear decision boundary?

Sol: The logistic regression value lies between 0 and 1 making the decision boundary linear.

3. (10 points) Given

$$X = [x_1, x_2, \cdots, x_n] \in \mathbb{R}^{m\times n}$$

^{m} for all i, and

$$Y = \begin{bmatrix} y_1^T \\ y_2^T \\ \vdots \\ y_n^T \end{bmatrix} \in \mathbb{R}^{n\times p}$$

where

$$y_i \in \mathbb{R}^p$$

for all i. Show that

$$XY = \sum_{i=1}^{n} x_i y_i^T.$$

Sol:

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}; Y = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1p} \\ y_{21} & y_{22} & \cdots & y_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n1} & y_{n2} & \cdots & y_{np} \end{bmatrix}$$

$$XY = \begin{bmatrix} x_{11} \cdot y_{11} + \cdots + x_{1n} \cdot y_{n1} & x_{11} \cdot y_{12} + \cdots + x_{1n} \cdot y_{n2} & \cdots & x_{11} \cdot y_{1p} + \cdots + x_{1n} \cdot y_{np} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} \cdot y_{11} + \cdots + x_{mn} \cdot y_{n1} & x_{m1} \cdot y_{12} + \cdots + x_{mn} \cdot y_{n2} & \cdots & x_{m1} \cdot y_{1p} + \cdots + x_{mn} \cdot y_{np} \end{bmatrix}$$

$$XY = \begin{bmatrix} \sum_{i=1}^n x_{1i} \cdot y_{i1} & \sum_{i=1}^n x_{1i} \cdot y_{i2} & \cdots & \sum_{i=1}^n x_{1i} \cdot y_{ip} \\ \sum_{i=1}^n x_{2i} \cdot y_{i1} & \sum_{i=1}^n x_{2i} \cdot y_{i2} & \cdots & \sum_{i=1}^n x_{2i} \cdot y_{ip} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^n x_{mi} \cdot y_{i1} & \sum_{i=1}^n x_{mi} \cdot y_{i2} & \cdots & \sum_{m=1}^n x_{mi} \cdot y_{ip} \end{bmatrix}$$

$$XY = \sum_{i=1}^{n} \begin{bmatrix} x_{1i} \cdot y_{i1} & x_{1i} \cdot y_{i2} & \cdots & x_{1i} \cdot y_{ip} \\ x_{2i} \cdot y_{i1} & x_{2i} \cdot y_{i2} & \cdots & x_{2i} \cdot y_{ip} \\ \vdots & \vdots & \ddots & \vdots \\ x_{mi} \cdot y_{i1} & x_{mi} \cdot y_{i2} & \cdots & x_{mi} \cdot y_{ip} \end{bmatrix} \tag{1}$$

$$\sum_{i=1}^{n} x_i \cdot y_i^T = \sum_{i=1}^{n} \begin{bmatrix} x_{1i} \\ x_{2i} \\ \vdots \\ x_{mi} \end{bmatrix} \times \begin{bmatrix} y_{i1} & y_{i2} & \cdots & y_{ip} \end{bmatrix} = \sum_{i=1}^{n} \begin{bmatrix} x_{1i} \cdot y_{i1} & x_{1i} \cdot y_{i2} & \cdots & x_{1i} \cdot y_{ip} \\ x_{2i} \cdot y_{i1} & x_{2i} \cdot y_{i2} & \cdots & x_{2i} \cdot y_{ip} \\ \vdots & \vdots & \ddots & \vdots \\ x_{mi} \cdot y_{i1} & x_{mi} \cdot y_{i2} & \cdots & x_{mi} \cdot y_{ip} \end{bmatrix} \tag{2}$$

By (1) and (2),

$$XY = \sum_{i=1}^{n} x_i y_i^T .$$

4. (10 points) We show that maximizing log-likelihood is equivalent to minimizing $RSS(\boldsymbol{w})$.
   In particular,

$$\log \mathcal{L}(\boldsymbol{w}|\boldsymbol{x}) \ = -\frac{1}{2}\left(\frac{1}{\sigma^2}RSS(\boldsymbol{w}) + n\log\sigma^2\right) + const$$

   (a) (5 points) Please derive the optimal $\boldsymbol{w}^*$ and σ^*.
   Sol:

$$\begin{aligned} RSS(\boldsymbol{w}) &= (\boldsymbol{y} - \mathbf{X}\boldsymbol{w})^T(\boldsymbol{y} - \mathbf{X}\boldsymbol{w}) \\ &= \boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{y}^T\mathbf{X}\boldsymbol{w} + \boldsymbol{w}^T\mathbf{X}^T\mathbf{X}\boldsymbol{w} \end{aligned}$$

   For optimum $\boldsymbol{w}^*$

$$\begin{aligned} \nabla \log \mathcal{L}(\boldsymbol{w}|\boldsymbol{x}) \ &= \frac{\partial RSS(\boldsymbol{w})}{\partial \boldsymbol{w}} = 0 \\ \implies -2\mathbf{X}^T\boldsymbol{y} &+ 2\mathbf{X}^T\mathbf{X}\boldsymbol{w} = 0 \\ \boldsymbol{w}^* &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{y} \end{aligned}$$

For optimum σ^*

$$\begin{aligned}\nabla_\sigma \log \mathcal{L}(\boldsymbol{w}^*|\boldsymbol{x}) &= \frac{\partial}{\partial \sigma}\left(\frac{1}{\sigma^2} RSS(\boldsymbol{w}^*) + n \log \sigma^2\right) = 0 \\ \implies \frac{-2}{\sigma^3} RSS(\boldsymbol{w}^*) + \frac{n}{\sigma^2}(2\sigma) = 0 \\ \implies \sigma^* = \sqrt{\frac{RSS(\boldsymbol{w}^*)}{n}}\end{aligned}$$

(b) (5 points) What do you observe from the optimal σ^*?
Sol: The variance decreases when we increase the number of examples.

5. (10 points) Show that sigmoid function and softmax function are the same in the binary case.

$$\text{sigmoid}(y) = \frac{1}{1+e^{-y}}$$
$$\text{softmax}(y_j) = \frac{e^{y_j}}{\sum_{i=1}^{c} e^{y_i}}$$

Sol:

$$\begin{aligned}\text{softmax}(y_1) &= \frac{e^{y_1}}{\sum_{i=1}^{2} e^{y_i}} \\ &= \frac{e^{y_1}}{e^{y_1}+e^{y_2}} \\ &= \frac{1}{1+e^{-(y_1-y_2)}} \\ &= sigmoid(y_1 - y_2)\end{aligned}$$

$$\begin{aligned}\text{softmax}(y_2) &= \frac{e^{y_2}}{\sum_{i=1}^{2} e^{y_i}} \\ &= \frac{e^{y_2}}{e^{y_1}+e^{y_2}} \\ &= \frac{1}{1+e^{-(y_2-y_1)}} \\ &= sigmoid(y_2 - y_1)\end{aligned}$$

6. (30 points) **Perceptron for Handwritten Digits Recognition**: The handwritten digits files are in the "data" folder: train.txt and test.txt. The starting code is in the "code" folder. In the data file, each row is a data example. The first entry is the digit label ("1" or "5"), and the next 256 are grayscale values between -1 and 1. The 256 pixels correspond to a 16 × 16 image. You are expected to implement your solution based on the given codes. The only file you need to modify is the "solution.py" file. You can test your solution by running "main.py" file. Note that code is provided to compute a two-dimensional feature (symmetry and average intensity) from each digit image; that is, each digit image is represented by a two-dimensional vector before being augmented with a "1" to form a three-dimensional vector as discussed in class. These features along with the corresponding labels should serve as inputs to your Perceptron algorithm. You are expected to use Python3.

(a) (5 points) Familiarize yourself with the data by completing the *show_images* function. Include the images you plotted in your report.

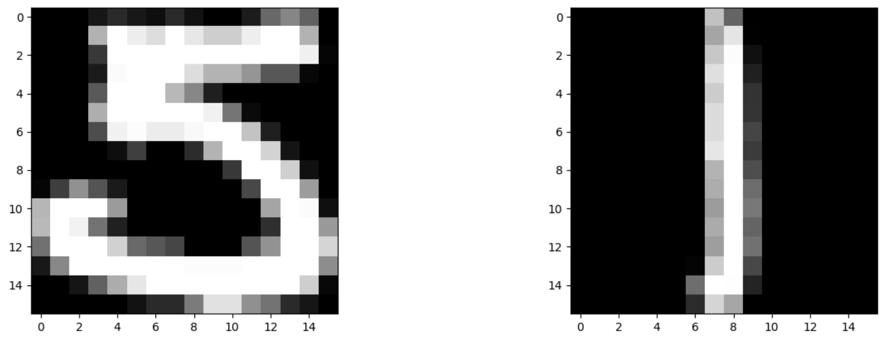

Figure 2: Digit five and one

(b) (5 points) In this assignment, we have already extracted two features, symmetry and average intensity, to distinguish between 1 and 5. Familiarize yourself with the features by completing the *show_features* function and include the 2-D scatter plot into your report. For each sample, plot the two features with a red $*$ if the label is 1 and a blue $+$ if the label is 5.

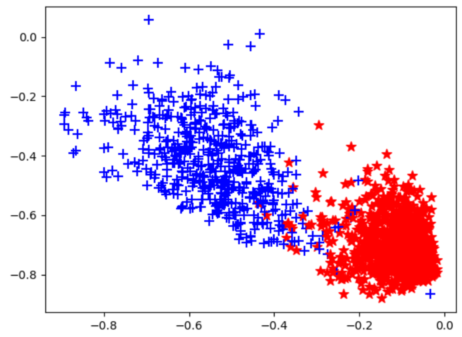

Figure 3: Scatter plot of the given data

(c) (10 points) Complete the *Perceptron* class. You can test your accuracy results using the "test_accuracy" function in "main.py".

(d) (10 points) Complete the *show_result* function to plot the test data with the separators. Include the images you plotted into your report.

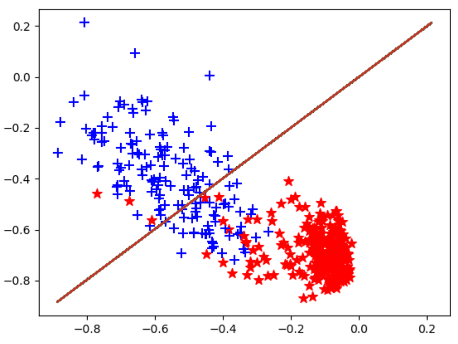

Figure 4: Linear model with optimal W