COM S 573: Machine Learning
Homework #1

1. Please put required code files and report into a compressed file "HW#_FirstName_LastName.zip"
2. Unlimited number of submissions are allowed on Canvas and the latest one will be graded.
3. No later submission is accepted.
4. Please read and follow submission instructions. No exception will be made to accommodate incorrectly submitted files/reports.
5. All students are required to typeset their reports using latex. Overleaf (`https://www.overleaf.com/learn/latex/Tutorials`) can be a good start.

---

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?

   (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.

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.

   (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.

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

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

   (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?

3. (10 points) Given

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

where $x_i \in \mathbb{R}^m$ 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

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

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^*.

   (b) (5 points) What do you observe from the optimal σ^*?

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

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

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.

   (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.

   (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.

Deliverable: You should submit (1) a report (along with your write-up for other questions) that summarizes your results and (2) the "solution.py" file to the Canvas.

Note: Please read the "Readme.txt" file carefully before you start this assignment. Please do NOT change anything in the "main.py" and "helper.py" files when you program.