Homework 3 – Machine Learning (CS4342, Whitehill, Spring 2021)

1. **Softmax regression (aka multinomial logistic regression)** [35 points]:

   (a) In this problem you will train a softmax regressor to classify images of clothing articles from the Fashion MNIST dataset. The input to the machine will be a 28×28-pixel image (converted into a 784-dimensional vector); the output will be a vector of 10 probabilities (shoes, t-shirts, dresses, etc.). Specifically, the machine you create should implement a function $g : \mathbb{R}^{785} \to \mathbb{R}^{10}$, where the kth component of $g(\tilde{\mathbf{x}})$ (i.e., the probability that input $\tilde{\mathbf{x}}$ belongs to class k) is given by

   $$\frac{\exp(\tilde{\mathbf{x}}^\top \tilde{\mathbf{w}}_k)}{\sum_{k'=1}^{10} \exp(\tilde{\mathbf{x}}^\top \tilde{\mathbf{w}}_{k'})}$$

   where $\tilde{\mathbf{x}} = [\mathbf{x}^\top, 1]^\top$.
   The weights should be trained to minimize the cross-entropy (CE) loss:

   $$f_{\mathrm{CE}}(\tilde{\mathbf{w}}_1, \ldots, \tilde{\mathbf{w}}_{10}) = -\frac{1}{n} \sum_{j=1}^{n} \sum_{k=1}^{10} y_k^{(j)} \log \hat{y}_k^{(j)}$$

   where n is the number of training examples. Note that each $\hat{y}_k$ implicitly depends on all the weights $\tilde{\mathbf{w}}_1, \ldots, \tilde{\mathbf{w}}_{10}$, where each $\tilde{\mathbf{w}}_k = [\mathbf{w}_k^\top, 1]^\top$.
   To get started, first download the Fashion MNIST dataset (including both the training and testing subsets) from the following web links:

   - `https://s3.amazonaws.com/jrwprojects/fashion_mnist_train_images.npy`
   - `https://s3.amazonaws.com/jrwprojects/fashion_mnist_train_labels.npy`
   - `https://s3.amazonaws.com/jrwprojects/fashion_mnist_test_images.npy`
   - `https://s3.amazonaws.com/jrwprojects/fashion_mnist_test_labels.npy`

   These files can be loaded into `numpy` using `np.load`.
   Then implement **stochastic gradient descent** (SGD) as described in the lecture notes. I recommend setting $\tilde{n} = 100$ for this project. Note that, since there are 785 inputs (including the constant 1 term) and 10 outputs, there will be 10 separate weight vectors, each with 785 components. Alternatively, you can conceptualize the weights as a 785×10 matrix.

   (b) Using your SGD implementation for softmax regression, train the classifier. Include in your PDF file a screenshot showing the training loss during the last 20 mini-batches of SGD.

   (c) After optimizing the weights on the training set, compute both the CE loss and the PC accuracy (percent correctly classified images) on the **test** set. Include these results in your submitted PDF.

   (d) Finally, create an image to visualize each of the trained weight vectors $\mathbf{w}_1, \ldots, \mathbf{w}_{10}$ (similar to what you did in homework 2) and include it in the PDF.

2. **Practice with Kaggle** [20 points]: Try your hand at the "Titanic - Machine Learning from Disaster" Kaggle competition (`https://www.kaggle.com/c/titanic/`). First, download the training and testing data from the "Data" tab. The file `train.csv` includes both the features (x) and target (y) values in the same file, whereas the file `test.csv` contains the features but not the target values.

   Using the `Sex`, `Pclass`, and `SibSp` fields, and using your code from part 1 of this homework, train a softmax regression model to predict whether a passenger will survive (as recorded in the `Survived` field). It's up to you whether you model the `Pclass` and `SibSp` as an ordinal (e.g., 3 is "higher" than 2, and 2 is "higher" than 1) or categorical (1, 2, and 3 are distinct categories but otherwise not comparable to each other) variables.

   After you have trained your model, use it to predict the target values ($y \in \{0, 1\}$, i.e., whether a particular person survived) for the test data. Then submit your predictions to Kaggle using the "Submit Predictions" button. Report (in the PDF) the accuracy your model achieved.

In addition to submitting your Python code in a file called `homework3_WPIUSERNAME1.py`, please submit a PDF file called `homework3_WPIUSERNAME1.pdf`.