CS534 — Implementation Assignment 2 — Due 11:59PM Nov 2nd, 2019

General instructions.

1. The following languages are acceptable: Java, C/C++, Python, Matlab

2. You can work in a team of up to 3 people. Each team will only need to submit one copy of the source code and report. You need to explicitly state each member's contribution in percentages (a rough estimate).

3. Your source code and report will be submitted through TEACH

4. You need to submit a readme file that contains the programming language version you use (e.g. python 2.7) and the command to run your code (e.g. python main.py).

5. Please make sure that you can be run code remotely on the server (i.e. babylon01) especially if you develop your code using c/c++ under visual studio.

6. Be sure to answer all the questions in your report. You will be graded based on your code as well as the report. In particular, **the clarity and quality of the report will be worth 10 pts**. So please write your report in clear and concise manner. Clearly label your figures, legends, and tables.

7. In your report, the results should always be accompanied by discussions of the results. Do the results follow your expectation? Any surprises? What kind of explanation can you provide?

Perceptron algorithm for Optical Character Recognition (total points: 80 pts + 10 report pts + 10 result pts)

Task description. In Optical Character Recognition (OCR) we seek to predict a number (between 0 to 9) for a given image of handwritten digit. In this assignment we consider a simplified binary classification task: differentiating two digits **3** versus **5** and you will implement and experiment with variations of the **perceptron** algorithm.

Data. All the handwritten digits are originally taken from http://www.kaggle.com/c/digit-recognizer/data The original dataset contains the sample digits suitable for OCR. We extract samples with only labels 3 and 5. Following a little pre-processings we produce three datasets for this assignment as follows:

(a) **Train Set (pa2_train.csv):** Includes 4888 rows (samples). Each sample is in fact a list of 785 values. The first number is the digit's label which is 3 or 5. The other 784 floating values are the the flattened gray-scale values from a 2d digital handwritten image with shape 28×28.

(b) **Validation Set (pa2_valid.csv):** Includes 1629 rows. Each row obeys the same format given for the train set. This set will be used to select your best trained model.

(c) **Test Set (pa2_test.csv):** Includes 1629 rows. Each row contains only 784 numbers. The label column is omitted from each row.

Important Guidelines. For all parts of this assignment:

(a) Please assign labels +1 to number 3 and -1 to label 5. In your produced predictions, please use only +1 and -1 as labels not 3 and 5.

(b) Please do not shuffle the given data unless instructed to do so. While in practice shuffling should be used to improve training convergence, for this assignment we ask you not to shuffle the data to ensure deterministic results for assessment purpose.

(c) To simplify the notation in this assignment, your load function which loads train, validation and test dataset should add a bias feature to the dataset. The bias feature is a feature with value of 1.0 for all of the samples. Therefore the feature size for each samples will become 785.

Part 1 (20 pts) : Online Perceptron. In the online perceptron algorithm we train a linear classifier with parameter $\boldsymbol{w}$ to predict the label of a sample with equation:

$$\hat{y} = \text{sign}(\boldsymbol{w}^T \boldsymbol{x}) \tag{1}$$

Where $\hat{y} \in \{-1, 1\}$. Algorithm 1 describes the online perceptron.

Algorithm 1 Online Perceptron

procedure OnlinePerceptron
  $\boldsymbol{w} \leftarrow \boldsymbol{0}$
  while iter < iters:
    for all sample $\boldsymbol{x_t}$ in train set: // no shuffling
      if $y_t \boldsymbol{w}^T \boldsymbol{x_t} \leq 0$:
        $\boldsymbol{w} \leftarrow \boldsymbol{w} + y_t \boldsymbol{x_t}$
  return $\boldsymbol{w}$

In this part we are interested in the following experiments for the online perceptron algorithm:

(a) Implement online perceptron as described in Algorithm 1. Set the iters = 15. During the training, at the end of each iteration use the current $\mathbf{w}$ to make prediction on the validation samples. Record the accuracies for the train and validation at the end of each iteration. Plot the recorded train and validation accuracies versus the iteration number. Does the train accuracy reach to 100%? Why?

(b) Use the validation accuracy to decide the best value for $iters$. Apply the corresponding learned model to make predictions for the samples in the test set. Generate the prediction file oplabel.csv. Please note that your file should only contain +1 (for 3) and -1 (for 5) and the number of rows should be the same as pa2_test.csv.

Part 2 (20 pts) : Average Perceptron. In this part you will implement and experiment with average perceptron, which is described in Algorithm 2.

Algorithm 2 Average Perceptron

procedure AveragePerceptron
  $\boldsymbol{w} \leftarrow \boldsymbol{0}$ //initialize weight vector
  $\bar{\boldsymbol{w}} \leftarrow \boldsymbol{0}$ // initialize average weight vector
  $s \leftarrow 1$ //initialize the example counter to 1
  while iter < iters:
    for all sample $\boldsymbol{x_t}$ in the train set: // no shuffling
      if $y_t \boldsymbol{w}^T \boldsymbol{x_t} \leq 0$:
        $\boldsymbol{w} \leftarrow \boldsymbol{w} + y_t \boldsymbol{x_t}$
      $\bar{\boldsymbol{w}} \leftarrow \frac{s\bar{\boldsymbol{w}} + \boldsymbol{w}}{s+1}$ // update $\bar{\boldsymbol{w}}$ regardless of $\boldsymbol{w}$ udpate
      $s \leftarrow s + 1$
  return $\bar{\boldsymbol{w}}$

Note that in Algorithm 2, the running average $\bar{\boldsymbol{w}}$ always gets updated every time an example is process, regardless whether the current $\boldsymbol{w}$ correctly classify it or not [1].

Perform the following experiments:

(a) Apply your implemeted average perceptron to learn from the training data. Plot the train and validation accuracies versus the iteration number for iters $= 1, ..., 15$.

[1] If you are interested, there is a slightly different but equivalent version of this algorithm that only performs update $\bar{\boldsymbol{w}}$ when $\boldsymbol{w}$ is updated, presented in Chapter 3 of A course in machine learning (Algorithm 7). You can choose to implement either version.

(b) What are your observations when comparing the training accuracy and validation accuracy curves of the average perceptron with those of the online perceptron? What are your explanation for the observations?

(c) Use the validation accuracy to decide the best value for *iter* and apply the corresponding learned model to make predictions for the test data. Please name the predicted file as aplabel.csv.

Part 3 (40 pts). Polynomial Kernel Perceptron. The online/average perceptron in Algorithm(1 and 2) are linear models. In this part we will consider kernelized perceptron, as described in Algorithm 3, with a polynomial kernel k_p of degree p:

$$k_p(\boldsymbol{x}_1, \boldsymbol{x}_2) = (1 + \boldsymbol{x}_1^T \boldsymbol{x}_2)^p \tag{2}$$

Algorithm 3 Kernel (polynomial) Perceptron

procedure KERNELPERCEPTRON
- $\alpha_i \leftarrow 0$ for $i = 1, ..., N$
- compute the gram matrix $K(i, j) = k_p(\boldsymbol{x}_i, \boldsymbol{x}_j)$
- while iter < iters:
  - for all sample $\boldsymbol{x_t}$ in training set: // no shuffling
    - $u = \sum_i \alpha_i K(i, t) y_i$
    - if $y_t u \leq 0$:
      - $\alpha_t \leftarrow \alpha_t + 1$

Implement the kernelized perceptron with polynomial kernel and perform the following experiments:

(a) Apply the kernelized perceptron with different p values in [1, 2, 3, 4, 5]:
1) For each p value, run your algorithm with $iter = 15$. At the end of each iteration use the current model (aka α's) to make prediction for the validation set. Record the train and validation accuracy for each iteration and plot the train and validation accuracies versus the iteration number.
2) Record the best validation accuracy achieved for each p over all iterations. Plot the recorded best validation accuracies versus p. How do you think p is affecting the train and validation performance?

(b) Use your best model (the best one you found over all p values and all iterations above) to make prediction for the test set. Please name the predicted file as kplabel.csv.

Submission. Your submission should include the following:
1) Your source code with a short instruction on how to run the code in a readme.txt.
2) Your report only in pdf, which begins with a general introduction section, followed by one section for each part of the assignment.
3) Three prediction files oplabel.csv, aplabel.csv and kplabel.csv. These prediction file will be scored against the ground truth y values and 10% of the grade will be based on this score.
4) Please note that all the files should be in one folder and compressed only by .zip.