First section contains requirements/ steps to run this program.

Second section contains answers to the questions.

Requirements for this module:

1. python3.4.6
2. conda install numpy
3. conda install scikit-learn
4. conda install nltk

Steps to run:

(assuming you are in a conda environment with python 3.4.6 or equivalent)

1. Unzip the zip file.
2. Cd into the folder logRegressionMithun.
3. Type : python initializer_cs539.py
4. Follow instructions on screen
5. If you choose 2: training and testing
    a. You will be asked to enter two values
        i. <maxMiniBatchSize>
        ii. <maxNoOfEpochs>
    b. maxMiniBatchSize: an int value which specifies the maximum mini batch size. The code will run for each of the mini batch sizes from 1 to maxMiniBatchSize and print the corresponding accuracies.
    c. maxNoOfEpochs: an int value which specifies the maximum number of epochs you want my code to run. the code will run from 1 to so many number of epochs. Data is shuffled at the beginning of each epoch.
    d. For your chosen value of max number of Epochs and max number of Batch Size the code will train on the attached training data set, test on the testing data and print corresponding accuracy.
    e. Please note that for every batch size the pickle will be stored to hard disk as
        i. trainedWeights.pkl
        ii. Vectorizer.pk
    f. A good combination to test is
        i. maxMiniBatchSize:2
        ii. maxNoOfEpochs:2

Assumptions:

1. If you decide to test with another test data please keep the data files with the same name as in d2l in a folder called data
2.

Files attached herewith:

1. Report
2. Code
3. 2 pickle files
    a. trainedWeights_golden.pkl
    b. vectorizer_golden.pkl

Ling/CSC 439/539: Assignment #2 (75 pts) (Graduate students)

Due by 11:59 P.M., September 24 (upload all materials to D2L)

Last modified on: 8/26/2017

Requirements for the submission

You must submit code and a written report for this assignment. The code and report must follow these requirements:

1. Each programming question must be answered through code that is executed with a single command in the terminal. Please include one command for each of the requirements stated below in the assignment. Submissions that must be run through an IDE (e.g., Eclipse, IntelliJ, etc.) are not accepted.

2. Similarly, if your code requires compilation (e.g., it is written in Java), a single

command line for compiling the code must be provided.

3. If your code requires certain dependencies (e.g., specific libraries, version of the

Python language), these have to be clearly stated with instructions for installation.

4. Your report must include clear instructions for all the above issues.

5. The code for this assignment cannot use ML libraries such as TensorFlow, but may use libraries such as numpy for the linear algebra in the assignment such as managing vectors, and NLP libraries such as NLTK (http://nltk.org) or spaCy (https://spacy.io) for the tokenization of the text.

Points will be taken off if the above requirements are not met. Additionally, your code must compile (if required by the programming language), run, and produce the correct output. Points will be taken off in any of these issues are violated.

Data

In this assignment you will use the SMSSpamCollection dataset (available in D2L). This dataset contains SMS messages labeled either as spam or ham. The files contain one message per line. Each line is composed by two columns: one with label (ham or spam) and other with the raw text. Here are some examples:

ham What you doing?how are you? ham Ok lar... Joking wif u oni... ham dun say so early hor... U c already then say... spam FreeMsg: Txt: CALL to No: 86888 & claim your reward of 3 hours talk time

to use from your phone now! ubscribe6GBP/ mnth inc 3hrs 16 stop?txtStop spam Sunshine Quiz! Win a super Sony DVD recorder if you canname the capital

of Australia? Text MQUIZ to 82277. B spam URGENT! Your Mobile No 07808726822 was awarded a L2,000 Bonus

Caller Prize on 02/09/03! This is our 2nd attempt to contact YOU! Call 0871-872-9758 BOX95QU

The dataset was split by the instructor in 3 files:

- SMSSpamCollection.train – The partition to be used for the training of your learn- ing algorithm;
- SMSSpamCollection.devel – The partition to be used for the tuning of hyper pa- rameters; and
- SMSSpamCollection.test – The testing partition to be used solely for the evaluation of your algorithm.

Problem 1 (35 points)

Implement a binary logistic regression (LR) algorithm and use it to train a spam classifier using the SMSSpamCollection dataset. Your LR algorithm must contain implementations from scratch for the sigmoid/logistic function, as well as for stochastic gradient descent (SGD). That is, you are not allowed to use an optimizer from another machine learning library. You may use numpy vectors in your code.

Train your algorithm using only the training partition. For features use unigrams (i.e., individual words in the messages). Tune any hyper parameters you have (I expect you will have at least two: number of epochs, and size of the minibatch) on the development partition. Include functionality to save the model learned to disk (after training), and to load a pre-trained model (before testing).

Ans:

1. Code attached herewith has a from-scratch implementation of binary logistic regression (LR) alongwith sigmoid and SGD implemenations.
2. It trains on the training data set SMSSpamCollection.train
3. Features are created using CountVectorizer from SciKit which counts the frequency of words.
4. The trained weights are stored after training as : “trainedWeights.pkl" at the same location as the initializer code .
5. An instance of CountVectorizer is also stored after training as : “vectorizer.pkl" at the same location as the initializer code .

Answer the following questions:

1. What were your best hyper parameters according to the analysis on the development Partition?

Ans ;

I tuned against the development data set and found that :

a. **Best number of epochs : 16**
b. **Best minibatch size : 10**

2. What was the classification accuracy (i.e., percentage of messages classified correctly) of your algorithm on the test partition, using the best hyper parameters from the development partition?

Ans:

86.9604%

To answer these questions, your submission must contain:

• Code to train, tune, and evaluate the LR algorithm. In the report, include: (a) a single command line to train the algorithm and save the resulting model; (b) a single command line to tune the algorithm, and (c) a single command line to run the algorithm on a test partition, using a pre-trained model, and specific hyper parameters.

• Include your best model in the submission, so the instructors can run the algorithm on a test partition without retraining.

• In the written report, include the best values for your hyper parameters (question 1), as well as data to support your choices (e.g., classification accuracy on the devel- opment partition as you vary the hyper parameters).

• Include the accuracy on testing in the written report.

Problem 2 (10 points)

Add bigrams (i.e., contiguous sequences of two words) to the features used by your algorithm. How does performance change after adding these features?

Include in your submission the same materials as above, but adapted for a classifier that uses both unigrams and bigrams as features.

Ans:

1. I used the ngram_range feature of CountVectorizer to specify my code to use only bi grams
    a. vectorizer = CountVectorizer(tokenizer=normalize, stop_words='english',ngram_range=(2, 2))
2. Just with BIgrams the accuracy reduced to 73.23%
3. Then I ran my code with both unigram and bigram
4. vectorizer = CountVectorizer(tokenizer=normalize, stop_words='english',ngram_range=(1, 2))
5. The accuracy increased to : 91.34%
6. The code can be found at :from utils.process_input_data import tokenizeWithBothUniBigrams
7. This is called in the function:train(filename,miniBatchSize,maxNoOfEpochs) In the file:trainData.py

Problem 3 (10 points)

Filter your features (i.e., unigrams and bigrams) based on their frequency in the training partition. For example, in one model keep only features seen more than 1 time in training, in another only features seen more than 2 times in training, etc. How does classification accuracy change for the different filtering thresholds (try at least five values for the filtering threshold)?

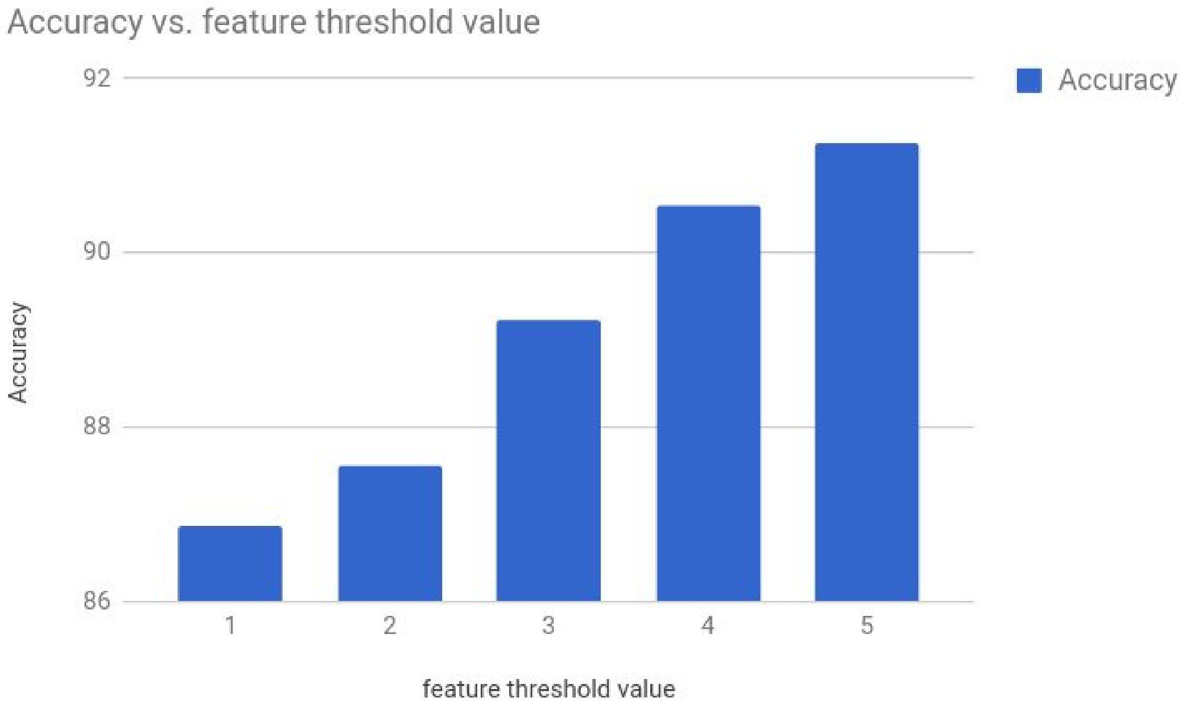
Include in your report a chart that plots the classification accuracy for the different feature threshold values. Your code must contain the functionality to filter features. Include in your report pointers to where this functionality is implemented.

Ans:

1. I achieved this using the min_df parameter of the CountVectorizer[5]
2. Eg: vectorizer = CountVectorizer(tokenizer=normalize, stop_words='english',ngram_range=(1, 2), min_df=1)
3. The code can be found at :from utils.process_input_data import tokenizeWithBothUniBigrams
4. This is called in the function:train(filename,miniBatchSize,maxNoOfEpochs) In the file:trainData.py
5. This was run for No. of epochs=2

6. Minbatch size of 2

feature threshold value	**Accuracy**
1	86.87
2	87.56
3	89.23
4	90.54
5	91.27



Problem 4 (10 points)

What are the top 20 features associated with the spam class, and the top 20 features associated with the ham class? For this analysis, you may use the best model you obtained as a result of the first three problems. What do you observe in this analysis? Do you believe your algorithm is overfitting, or not?

Problem 5 (10 points)

Find a research paper in the ACL Anthology (http://aclweb.org/anthology/) that uses text categorization and summarize it in your written report. What task is the paper addressing? What is the approach implemented? What are the main results and how do they compare with other approaches in the same space? What are the limitations of the proposed work?

\

Ans:

I am reviewing the work "Very Deep Convolutional Networks for Text Classification" by Conneau et al[6] . This paper addresses the task of classification of news documents into groups using VDCNN, a Convolutional Neural Network based architecture which operates at the character level. Their key selling point.novelty is that they go beyond the traditional LSTM and other shallow architectures and implement a deep architecture. They run their architecture on datasets like Yelp Full, Yahoo Answers and Amazon Full. Their primary discovery is that depth improves performance. Their architecture which has a depth of 29 has only a classification error of 37% when compared to the current benchmark of 40.43%. Also they discover that max pooling of size 3 performs better than convolutions with a stride 2. They compare their approach with a similar architecture that was used in Computer Vision[7]. They say that the closest to their work is that of Zhang et al[8], using six convolutional layers. Thus their main selling point is that this is the first time the benefit of depth has been shown for CNN in NLP

References:

1. Numpy home page
    a. https://www.python-course.eu/numpy.php
2. Pandas home page
    a. https://pandas.pydata.org/pandas-docs/stable/10min.html#getting-data-in-out
3. Scikit home page
    a. http://scikit-learn.org/stable/install.html
4. Numerical operations on NumPy array
    a. https://www.python-course.eu/numpy_numerical_operations_on_numpy_arrays.php
5. http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html
6. http://aclweb.org/anthology/E/E17/E17-1104.pdf
7. Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In ICLR, San Diego, California, USA.
8. Xiang Zhang and Yann LeCun. 2015. Text understanding from scratch. arXiv preprint arXiv:1502.01710.