Assignment 2

Natural Language Processing
Fall 2018
Total points: 90
Issued: 9/29/2018 Due: 10/19/2018

Unless explicitly specified (either here or on Piazza), all the code has to be your own. Here, you are allowed to use the numpy or scipy libraries for basic matrix math (e.g. multiplication, transpose). The code must run on the CAEN Linux environment using Python 3.6 without additional installation or additional files (except for the data files specified in the assignment). To load Python 3.6, on your CAEN terminal do:

```
module load python/3.6
```

You cannot use any library that does not come with CAEN Python, nor can you use libraries or tools that trivialize the assignment.

You can discuss the assignment with others, but the code is to be written individually. You are to abide by the University of Michigan/Engineering honor code; violations will be reported to the Honor Council.

1. [60 points] Language Identification Using Neural Networks.

Implement a language identifier, using a neural network with one hidden layer. The input will be five sequential characters from the text, and the output will be a softmax over three options, determining whether the language is English, French, or Italian. **The neural network should be implemented using matrix operations using numpy, scipy and sklearn. You cannot use sklearn.neural_network, TensorFlow, PyTorch, Keras, etc.**

Neural Network Guidelines

Design your neural network according to the following specifications:

- Let *c* be the number of unique characters present in any of the three languages. Then, the input layer ***x*** will be *5c*-dimensional. It will consist of five one-hot letter encodings concatenated together (each one-hot encoding will be *c* dimensions).
- The weight matrix $\boldsymbol{W}^1 \in \mathbb{R}^{d \text{ x } 5c}$ defines the weights between the input layer and the hidden layer ($\boldsymbol{W}^1[i][j]$ refers to the weight between the *j*th input node and the *i*th hidden layer node). The bias term $\boldsymbol{b}^1 \in \mathbb{R}^d$ defines the bias between ***x*** and ***h***.
- The hidden layer ***h*** will be *d*-dimensional (we will treat *d* as a hyperparameter). It is defined as $\mathrm{h} = \sigma(\mathrm{W}^1\mathrm{x} + \mathrm{b}^1)$ where σ is a function that applies *sigmoid* to each individual element of the array. Here,
  $$sigmoid(z) = \frac{1}{1 + \exp(-z)}$$

- The weight matrix $\boldsymbol{W}^2 \in \mathbb{R}^{3 \times d}$ defines the weights between the hidden layer and the output layer ($\boldsymbol{W}^2[i][j]$ refers to the weight between the jth hidden layer node and the ith output layer node). The bias term $\boldsymbol{b}^2 \in \mathbb{R}^3$ defines the bias between $\boldsymbol{h}$ and $\boldsymbol{y}$.
- The output layer $\boldsymbol{y}$ will be 3-dimensional. It is defined as $\mathbf{y} = \mathbf{W}^2\mathbf{h} + \mathbf{b}^2$.
- To gain the final probability distribution, take a softmax over $\boldsymbol{y}$, where

  $$softmax(\mathbf{y})[i] = \frac{\exp(\mathbf{y}[i])}{\sum_j \exp(\mathbf{y}[j])}$$

Train your neural network according to the following specifications:

- Train the network using basic stochastic gradient descent (though feel free to experiment with more advanced training algorithms for Part 2!). Please do not use external libraries to implement this.
- Use the squared loss, where $\hat{\mathbf{y}}$ is the true output (The correct labels from the data), and y is your predicted output:

  $$L(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{2}||\mathbf{y} - \hat{\mathbf{y}}||_2^2$$

  - ($||\,.\,||^2_2$ denotes L2 Norm)
- Consider two hyperparameters: the size of the hidden layer (d) and the learning rate (η).
- At the beginning, initialize your weight and bias matrices randomly (again, feel free to experiment with other initialization strategies for Part 2!).
- Shuffle your training data to prevent the network overfitting to one output class.
- Train your neural net for at least 3 epochs (an epoch is when you send all of your training data through the neural net once).
- Bear in mind that your neural net may take up to 1.5 hours to train. Start the assignment early so that you have enough time for this!

Dataset Guidelines

The dataset to be used for this assignment is
languageIdentification.data.zip, available from the Files section on Canvas, under Assignments/.

Once you unpack the data archive, you will obtain a folder called languageIdentification.data/.
Store this data folder in the same folder as your program. There are four files included in the languageIdentification.data/ folder: one file with data to use for training (*train*), one for optimizing hyperparameters (*dev*), one for testing (*test*), and one with the solutions to the test dataset (*test_solutions*). The files *train* and *dev* are structured in the same way: each line contains a separate sentence. The first token of each line is the language of the sentence (either ENGLISH, FRENCH, or ITALIAN). The file *test* only contains the separate sentences to test on (it does not include the identifying language of each sentence). The file *test_solutions* contains the correct language identifications for each sentence in the *test* file.

It is very important that you use the data in these files in the way that it is intended (e.g. training or optimizing hyperparameters on the test set is bad practice!).

Programming Guidelines

Write a Python program called languageIdentification.py. This program will take three command line arguments: the training data file (first argument), the dev data file (second argument) and the test data file (third argument). After reading the arguments, the program should perform the following sequence of steps:

1. For Part 1 of the assignment, use the hyperparameters *d=100* and η*=0.1*. (For this part of the assignment, you will not use the dev data set, but you will use it for Part 2 of the assignment.)
2. Train the neural network with the hyperparameters provided. Only use the training data provided as the first command line argument.
3. Before you begin training the neural net, calculate the accuracy of the classifier on the training data and the accuracy of the classifier on the dev data. While you are training the neural net, at the end of each epoch, calculate these accuracies again. Finally, produce a graph of training epoch v. accuracy (plot training accuracy and dev accuracy on the same graph). Save this graph as “accuracy.png”, and submit this file with your assignment.
4. Open the test file, provided as the third argument on the command line, and for each line in the test file, use the neural network to identify the probable language for that sentence.
5. Produce a file called languageIdentificationPart1.output, with the following content:

   Line1 Language1
   Line2 Language2
   …
   LineN LanguageN

   (where Line*i* represents the content of the *i*th line in the test file, and Language*i* is the language determined as most likely). Submit this file with your assignment.

The languageIdentification.py program should be run using a command like this:

```
% python languageIdentification.py languageIdentification.data/train
     languageIdentification.data/dev languageIdentification.data/test
```

Write-up and Submission Guidelines

Create a text file called languageIdentification.answers, and include the following information under “Part 1”:

- A short explanation of how you used the neural network to identify the probable language of an individual sentence
- Accuracy of your language identifier (the percentage of predictions that are correct)

2. [30 points] Hyperparameter Optimization.

For this part of the assignment, use the dev data to optimize your hyperparameters. That is, using at least five different sets of values for the hyperparameters *d* and η, train the network on the training data, and see which set of hyperparameters performs best on the dev data. Then, use this set of hyperparameters to perform an evaluation on the test data.

Write-up and Submission Guidelines

Include the following information under “Part 2” of languageIdentification.answers:

- A short explanation of how you decided which sets of hyperparameters to try
- A list of the sets of hyperparameters that you tried on the dev data
- The best performing set of hyperparameters (as determined by the dev data)
- Accuracy of your best language identifier (measured on the test data)

General Submission Instructions

- Make sure all your programs run correctly on the CAEN machines.
- Include all the files for this assignment in a folder called [your-uniqname].Assignment2/ (this should include languageIdentification.answers, languageIdentificationPart1.output, languageIdentification.py, accuracy.png, and any other python files that you created).
- Do not include the data folder languageIdentification.data/.
- Archive the folder using zip and submit on Canvas by the due date.
- Include your name and uniqname in each file that you submit.