CSE474/574: Introduction to Machine Learning(Fall 2018)

Instructor: Sargur N. Srihari
Teaching Assistants: Mihir Chauhan, Tiehang Duan, Alina Vereshchaka and others

**********September 3, 2018**********

Project 1.1: Software 1.0 Versus Software 2.0

Due Date: Monday, September 17

1 Objective

The project is to compare two problem solving approaches to software development: the logic-based approach (Software 1.0) and the machine learning approach (Software 2.0). It is also designed to quickly gain familiarity with Python and machine learning frameworks.

2 Task

We consider the task of FizzBuzz. In this task an integer divisible by 3 is printed as *Fizz*, and integer divisible by 5 is printed as *Buzz*. An integer divisible by both 3 and 5 is printed as *FizzBuzz*. If an integer is not divisible by 3 or 5 or 15, it simply prints the input as output (for this last case, the input number can be classified as Other in Software 2.0, it should then be handled using Software 1.0, which prints the input as output).

Your programs will be tested on how well they perform in converting integers from 1 to 100 to the FizzBuzz labels.

2.1 Software 1.0

Implement the logic in Python using standard logic (if-then-else statements using modulo arithmetic). With the simple logic that is needed, your program will presumably work perfectly on all 100 input integers.

2.2 Software 2.0

First you need to create a training data set for numbers ranging from 101 to 1000. We avoid training on integers 1 to 100 because that forms the testing set (In machine learning it would be considered cheating to train on the testing set). We present this training set to the program in the form of (input,output) pairs.

To design the learning program, you will have to make decisions on hyper-parameters such as the learning rate, number of epochs, loss function, regularizer, etc. Since

outputs are discrete, you can use cross-entropy as your loss function. Plot the performance of your program for different values of the hyper-parameters.

As mentioned above, the purpose of this first project is to give you an intuitive feeling on how machine learning works and let you discover the wonders of machine learning without worrying too much on the technical details. To better serve this goal, we made a sample implementation available on UBlearns. We encourage you to dive into the sample implementation, get to know the functionality of each line in the code, and share with us your understanding in the form of comments to the code (and include it in the submission zip). There are also other Python/Tensorflow implementations available online. You can look at them and use them, but try and understand the decisions being made. You may also wish to implement it using any of the alternative machine learning frameworks such as Pytorch, Keras and Gluon.

3 Deliverables

There are two deliverables: report and code. After finishing the project, you may be asked to demonstrate it to the TAs, particularly if your results and reasoning in your report are not clear enough.

1. Report

   The report should describe the performance of your program using accuracy measures. Also accuracy for Fizz, Buzz and FizzBuzz. Show how the choice of hyper-parameters affects performance in the form of graphs (for example, you can show how the dropout rate affects the cross entropy loss with a figure where the x axis is the dropout rate ranging from 0.1 to 1 and y axis being the cross entropy). We encourage you to tune the model with different network settings such as different number of layers in the network, different number of nodes in each layer, different optimization methods such as SGD, Adagrad, Adadelta, Rmsprop and Adam, different dropout rates, different activation functions such as sigmoid, tanh, relu and leaky relu etc. Please include in your report the network setting that you get the best performance.

   Submit the PDF on a CSE student server with the following script:

   `submit_cse474 proj1.1.pdf` for undergraduates

   `submit_cse574 proj1.1.pdf` for graduates

   In addition to the PDF version of the report, you also need to hand in the hard copy version on the first class after due date or else your project will not be graded.

2. Code

   The code for your implementation should be in Python only. You can submit multiple files, but the name of the entrance file should be main.py. Please provide necessary comments in the code. Python code, training and testing files should be packed in a ZIP file named `proj1.1code.zip`.

   Submit the Python code on a CSE student server with the following script:

   `submit_cse474 proj1.1code.zip` for undergraduates

   `submit_cse574 proj1.1code.zip` for graduates

After extracting the ZIP file and executing command `python main.py` in the first level directory, the program should output a CSV file (output.csv). Format of output.csv is provided on Ublearns.

4 Scoring Rubric

Conceptual Understanding in Report: 30%
Results in Report - Graphs, Tables etc: 40%
Report Formatting: 5%
Python Code Understanding (Provide comments in your Python Code): 25%