HEALTH UNIVERSITY OF UTAH SCHOOL OF MEDICINE

BMI 6015, Fall 2018
Assignment: Neural Network and Keras Basics

Release Date: 11-10-2018
Due Date: 11-22-2018 (11:59 pm)
Marks: 100 points

The purpose of this assignment is to familiarize yourselves with neural networks (basics, calculations, and implementation) and to explore Keras for fully connected neural networks. In this assignment, you may use any python, scikit-learn, and TensorFlow/Keras libraries per question instructions but "copy & paste" from any other resource, including the Web, is totally prohibited. Feel free to email the professor for any question or inquiry. To submit the assignment, you will receive instructions from the professor for where you should upload your solutions. In this assignment, you may want to research part III either alone or in a group of two students per your preference; otherwise, you will be required to answer the other parts by only yourself. The course TA may only help you in any debugging issue. Your solution should be provided using Jupyter notebook. We will skim over the questions in the 11-12-2018 lecture.

Good Luck!
Samir Abdelrahman

A. Part I (30 points): Neural network (NN) – calculations and implementation:

Requirements: You are required to understand NN calculations in (10-29-2019 notebook and BMI_Applied_Machine_Learning_Neural_Network_Example_5Nov-Extra Notes folder). You need to understand and run the NN implementation in 10-31-2018 notebook. In this question, do not use any scikit-learn, Tensorflow or Keras library. You need to implement the solutions yourself from scratch using only python libraries when needed.

Question: Modify the neural network code in 10-31-2018 notebook to solve the example in BMI_Applied_Machine_Learning_Neural_Network_Example_5Nov-Extra Notes folder, as follows:

(1) The values for input is 1, 3, and 7. The initial weights should be taken from the first link in the example references; they should not be randomly generated. The target output is 1, 0, and 0.
(2) Modify the input scaling function to support the input illustrated in this example.
(3) Modify the NN output and weight function to support multiple class and related summation calculations.
(4) Implement any new activation/loss and related derivative functions for this example.
(5) Test your code using three different hyper-parameter settings (learning rate and number of iterations) and include them with related results and your interpretation of each result in your solution notebook.

B. Part II (30 points): Keras fully connected implementation:

Requirements: You are required to study and understand the Keras example in 11-5-2018 and 11-7-2018 lectures to answer the below questions. You may use and modify the codes in these lectures in your answer. You may use only python, scikit-learn, and Keras libraries to help you in your coding.

Question: Use prims.csv (Pima Indians Diabetes Database) data in the 10-31-2018/files/ folder to implement the following tasks:

(1) Load the data.
(2) Shuffle the data randomly.
(3) Split the data into 2/3 and 1/3 for training and testing splits, respectively.
(4) Standardize (0 mean and 1 stdev) the training data and use its information to standardize the testing data.
(5) If needed, use one-hot encoding for categorical data.
(6) Take 20% of the training data as validation data.
(7) Develop your model using Keras models library with different hyper-parameter settings (number of layers, and number of nodes per layer). Compile and fit it with necessary functions with the accuracy metric.
(8) Select the model with the least loss value in Task 7. Plot the training and validation loss curves for the selected model to check the overfitting problem. If over-fitted, select the number of epochs before the occurrence of overfitting. If the curves are too diverge, select the minimum distance between the two curves.
(9) Rebuild the selected model in Task 8 using the whole training dataset and the selected number of epochs. Test the performance on testing data and print the loss and accuracy on training and testing datasets.

Hint: You may want to run five different settings to develop five models in Task 7.

C. Part III (40 points): Research question for some NN basics:

Requirements: You are required to read/understand the concepts in the below web links. To answer the below questions, you may want to search the Web to understand relevant concepts. You are required to summarize your understanding and cite any resource that was used in your answer (no Copy & Paste). You don't need to fully understand the math or the implementation details of each function, but you need to demonstrate your conceptual understanding. You may want to solve this part alone or with one of your course colleagues (indicate the name of your colleague in your answer).

Question: For each of (a) and (b) links below, answer the following questions:

(1) The advantages or when to use each function in (a) and (b)
(2) The disadvantages or the limitations of each function in (a) and (b)

Web Links:

(a) Activation Functions: https://en.wikipedia.org/wiki/Activation_function .
(b) Gradient descent optimizers: http://ruder.io/optimizing-gradient-descent/ .