Assignment 03 (Due date: Oct. 27, 2019)

Neural Networks

Assignment 03

Due: Oct. 27, 2019

The purpose of the this assignment is to create multi-layer neural networks using Tensorflow (without using Keras) .

To begin this assignment download the Kamangar-03.zip and unzip it to your computer.

Implementing Model and Training

- In this assignment you will implement a class which can be used to create multi-layer neural networks.
- Your code should be using Tensorflow (without using Keras).
- There are two files in the unzipped directory. The "multinn.py" file is where you are to implement the network model and all necessary helper functions. The "test_multinn.py" file includes the unit test modules.
- You DO NOT need to rename these two files according the submission guidelines. Just modify the first four lines of these files according to the assignment submission guidelines.
- The "MultiNN" class is 'stubbed out' and your task is to implement the unimplemented functionality **within the given structure**.
- **DO NOT** alter the structure that is given. You may introduce additional helper methods, but do not alter function names or argument structures.
- The API structure of the given file is inspired by (but not an exact copy of) modules such as **scikit-learn** and **keras** that are very commonly used in practice.
- The comments and docstrings provide additional information that should help with your implementation.
- Methods that you must implement:
  - `add_layer` - This is where you add a layer to the network. Note that this function adds a layer after that last layer of the network. In other words this function appends a layer.
  - `predict` - Given array of inputs this method predicts array of corresponding outputs for the multi-layer network.
  - `train` - Given array of inputs, desired outputs as class indexes, and other parameters, this method adjusts the weights using the partial derivatives of the loss function with respect to weights and biases. Note that this method should implement batch training. You do not need to implement regularization in this assignment.
  - `get_weights_without_biases` - Given the layer number. This function returns the weight matrix for the given layer.
  - `get_biases` - Given the layer number. This function returns the biases for the given layer.
  - `set_weights_without_biases` - Given the layer number. This function sets the weight matrix for the given layer.
  - `set_biases` - Given the layer number. This function sets the biases for the given layer.
  - `calculate_percent_error` - Given array of inputs and desired outputs as class indexes this method calculates percent error.
  - `calculate_confusion_matrix` - Given array of inputs and desired outputs as class indexes array this method calculates the confusion matrix

Notes:

- The test_multinn.py file includes a very minimal set of unit tests for the multinn.py module. Part of the assignment grade will be based on your code passing these tests (and some other unspecified tests)
- **You may modify the "test_multinn.py" to include more tests. You may add additional tests to help you during development of your code. The changes that may make to the "test_multinn.py" file will not be graded.**

- You may run these tests using the command: `py.test --verbose test_multinn.py`

Grading Criteria

- Passing Unit Tests - 80 points Note: Not all tests are given.
- Qualitative Evaluation - 20 points (Grader may examine your code and subjectively award as many as 20 points.)

Submission Guidelines

- Modify the first four lines of the two Python files according to the assignment submission guidelines.

# Your name (last-name, first-name)

# Your student ID (100x-xxx-xxx)

# Date of submission (yyyy-mm-dd)

# Assignment-nn-kk

- You **DO NOT** need to rename the two Python files. The name of the two Python files should remain as "multinn.py" and "test_multinn.py" .
- Create a directory and name it according to the submission guidelines and include your files in that directory.
- Zip the directory and upload it to Canvas according to the submission guidelines.

Top