CS 5102 - Deep Learning

Part A - Convolutional Neural Network using Keras with Tensorflow backend

1. The accuracy of more than 90% needs to be achieved in order for this part of the project to be graded.
2. Implement Convolutional Neural Network on the MNIST digit dataset using Keras with Tensorflow backend.
3. (Abstract) Overview of the Architecture:
    a. The architecture to be followed should have 3 convolution layers followed by pooling layers.
    b. A fully connected layer and softmax layer respectively. Multiple filters are to be used at each convolution layer, for different types of feature extraction.
    c. Dropout rate set to 25%

Note: The files should be submitted as Jupyter Notebooks (in ipynb format)

Deadline: Thursday, 1st November 2018 at 11:59 pm.

Regarding Submission: Send an email with the Jupyter Notebook attached to taz.taimur@gmail.com . OR you can send your python file with a one-page report (containing the following items). The subject should be “Project 1 - Part A - L18-xxxx” before the deadline. The Notebook (or report) should also contain:

1. Architecture built, with specifics such as filter sizes, activation functions, etc
2. Final training and test accuracy
3. Execution parameters such as number of epochs, learning rate etc

Resources:

1. http://yann.lecun.com/exdb/mnist/
2. https://keras.io/
3. https://www.anaconda.com/
4. http://jupyter.org/
5. https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html
6. http://cs231n.github.io/convolutional-networks/

Helpful Tips:

```
In [28]: from keras.datasets import mnist
    ...: (x_train, y_train), (x_test, y_test) = mnist.load_data()
    ...:
```

For Anaconda users, use “conda install <package name>” instead of “pip install <package name>” to install a package such as tensorflow, keras, OpenCV, etc. Moreover, run Anaconda Prompt as an Admin to install a package in Anaconda.

Part B - Tensorboard

Append the Part A of the project and add code for the following:

1. Plot error vs epochs
2. Using Tensorboard, visualize the architecture and plot weights in the Fully Connected Layer and in the Convolution Layer.

Save results of Tensorboard and error vs epoch plot as images or take screenshots and save them as images (JPEG or PNG).

Note: The files should be submitted as Images (Not Compressed) and in Jupyter Notebooks (in ipynb format). It is preferable that you append to the Part A Notebook. This way you'll have a good project to add to your portfolio of projects, and making a single file will help potential recruiters to go through it easily. Moreover, creating a GitHub Repository along the way will also be helpful in the long run. Please add a comment or create a markdown cell which helps in differentiating between part A and B.

Deadline: Thursday, 8th November 2018 at 11:59 pm.

Regarding Submission: Send an email with the Jupyter Notebook and the images attached to taz.taimur@gmail.com . OR you can send the images along with the python file. The subject should be "**Project 1 - Part B - L18-xxxx**" before the deadline. The Notebook (or file) should also contain the code for Part B.

Resource:

https://www.youtube.com/watch?v=eBbEDRsCmv4

Part C - Hyperparameter-Tuning

This part of the project utilizes a different dataset than the Parts A and B. The dataset is a modified version of the Dogs Vs Cats Dataset used in one of our workshops.

This part of the Project has the following requirements:

1. The Dataset will require a different preprocessing technique than the one taught in the Workshop. You need to create a preprocessing function that would prepare the Dataset for the deep learning model.
2. The Dataset is altered, such that the default values for the Hyperparameters will result in a bad accuracy. Using the following architecture, tune the hyperparameters such that the model yields an acceptable accuracy. Your task in this part is to take the given code and parameters and analyze the training behaviour. If you are not satisfied with the training process, you may change any/all of the following parameters until you obtain the required training behaviour:
    a. Batch size
    b. Weight initialization
    c. Learning rate
    d. Momentum
    e. Optimization function
    f. Dropout
3. You are not allowed to add more convolutional layers, and the number of training epochs (should be less than or equal to 100)

Provide all the steps/changes you did to improve training, and justify each change. Also, mention how you reached this correct value/step. Produce the training/test error plots for each change.

The top accuracy will set the bar and all other accuracies will be graded relatively. Minimum accuracy required for the assignment is 80%.

Tune the hyperparameters and show it either in your Jupyter Notebook or in your report about which values of hyperparameters increased the accuracies and which didn't. (penalty for plagiarism)

Architecture:

```
im_width = 64
Im_height = 64
model = Sequential()
model.add(Conv2D(kernel_size=(3,3),filters=3,input_shape=(im_width, im_height,
1),activation="relu",padding="valid"))
model.add(Conv2D(kernel_size=(3,3),filters=10,activation="relu",padding="same"))
model.add(MaxPooling2D(pool_size=(2,2),strides=(2,2)))
model.add(Conv2D(kernel_size=(3,3),filters=3,activation="relu",padding="same"))
model.add(Conv2D(kernel_size=(5,5),filters=5,activation="relu",padding="same"))
model.add(MaxPooling2D(pool_size=(3,3),strides=(2,2)))
model.add(Conv2D(kernel_size=(2,2),strides=(2,2),filters=10))
model.add(Flatten())
model.add(Dropout(0.2))
model.add(Dense(100,activation="sigmoid"))
model.add(Dense(1,activation="sigmoid"))
model.summary()
```

Note: The files should be submitted in Jupyter Notebooks (in ipynb format) or as Python Files. In the case of Python Files, you need to submit a separate PDF Document containing all the outputs, such as the outputs for each epoch, the final accuracies and all the images displayed.

Dataset:
https://drive.google.com/file/d/1fk-rBgurmtJGkgpmdk0QbVB2iZe8fWl9/view?usp=sharing

Deadline: Sunday, 25th November 2018 at 11:59 pm.

Regarding Submission: Send an email with the Jupyter Notebook to taz.taimur@gmail.com . The subject should be "**Project 1 - Part C - L18-xxxx**" before the deadline.