ENEL525 Project: Lesson 2 Exercise for Submission

Nov. 24, 2023

For Lesson 2, you only need to submit Exercise #2 that exists in part 2 of the lesson (on slides #41-43). Instructions and some other notes are provided to you below to complete this submission. The exercise itself is also mentioned on the second page of this document for convenience.

Hand in (Midnight Dec 01) via D2L Dropbox (5% of the final project grade): A PDF file that has screenshots of the following:

1. Your entire code running without errors. Warnings are fine as long as the code is running as expected (show the "Problems" tab in PyCharm or the correctly running cells in Jupyter)**.**
2. A screenshot of the model summary printout.
3. A screenshot of the confusion matrix and accuracy printout.
4. A screenshot of the model accuracy and loss plots.

Notes:

- Two CSV files for both the dataset and targets will be provided to you. ("FourClasses.csv" and "Targets_0.csv").
- Jupyter/Google Colab allows you to use python online by connecting to a remote instance of a "server" with limited CPU/GPU resources.
- These instances may time-out and give you "Server error message" if left inactive for long periods of time, you can:

1) Save the code in a text file or .py file.
2) Restart a new instance (https://jupyter.org/try).
3) Continue by copying the code to the new python3.pynb or run the python file from terminal.

- Alternatively, you can just work offline with your own computer with python (3.7.0+) and IDE like Pycharm 2021 (Lesson 1 videos explain how to do so in details).

Useful links:

- Pycharm: https://www.jetbrains.com/pycharm/
- Python: https://www.python.org/downloads/
- Jupyter lab to execute python in browser: https://jupyter.org/try
- Google Colab to execute python in browser: https://colab.research.google.com/?utm_source=scs-index

If you have any feedback to improve future lessons or questions regarding the content/exercises, please email: aliadib.arnab@ucalgary.ca

Exercise 2: 4-Class Classification Neural Network (From Lesson 2 - Part 2 (Slides #41-43)):

You can use the same code as in exercise 1 as an initial draft to be modified to solve this exercise.

- Consider a new dataset with 4 classes (Given to you as "FourClasses.csv").
  - 1000 datapoints centered around (170,90) with target class 0.
  - 1000 datapoints centered around (155,75) with target class 1.
  - 1000 datapoints centered around (130,65) with target class 2.
  - 1000 datapoints centered around (100,50) with target class 3.
- Also, consider a target array that corresponds to this dataset given to you as "Targets_0.csv".

1. Modify based on the designed neural network from Exercise 1, but:
   - Do all the necessary changes for the indices to accommodate for all the 4000 samples.
   - Use 80% of the dataset for training and 20% for testing.
   - Use: y_train = tf.one_hot(y_train,4) to convert targets to one-hot encoding.
   - Change your "output layer" activation (last layer only) to "softmax" with 4 classes/neurons.
   - Also, note the following changes to the model:

```
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
history = model.fit(X_train, y_train, validation_split=0.2, batch_size=24, epochs=100)
```

2. View the model summary.
3. Use:

```
y_pred = np.argmax(y_pred, axis=1) #Decode from tf.one_hot to original
```

4. Change the thresholding criteria using a for loop and if/else statements to covert "y_pred_t" to the pre-defined 4 classes "0", "1", "2",..etc.
5. Evaluate the performance of your classifier with confusion matrix and calculate the accuracy from the confusion matrix (Remember to sum-up all the diagonal values of the 4x4 matrix).
6. Plot the model accuracy and loss:
   - To do so, search online on how you can use the model history for sequential models to plot both the model accuracy and loss.