Spring 2022: CSEE5590/490 – Special Topics

Python and Deep Learning Module-2 - ICP-8

Lesson Overview:
In this lesson, we are going to have an introduction to Deep Learning programming on Keras. Before to that, we will introduce some of the applications of the Deep Learning in the area of vision and NLP.

Use Case Description:
Predicting the diabetes disease

Programming elements:
Keras Basics

Source Code:
Provided in the assignment and GitHub repo.

Assignment:

1. Use the use case in the class (DL_Lesson_1.ipynb & diabetes.csv):
   a. Add **more Dense layers** to the existing code and check how the accuracy changes.
   b. Add the **validation_data**=(X_test, Y_test) attribute to .fit() method.
   c. Plot the **accuracy** for training and validation (one plot for both: train and test accuracy).
   d. Plot the **loss** for training and validation (one plot for both: train and test loss).
   e. **Normalize** the data before feeding to the model and check how the normalization changes your accuracy sue the `StandardScaler()` from scikit learn.

2. Change the data source to Breast Cancer.csv available in the source code folder and make required changes. Report accuracy of the model.
   a. Add **more Dense layers** to the existing code and check how the accuracy changes.
   b. Add the **validation_data**=(X_test, Y_test) attribute to .fit() method.
   c. Plot the **accuracy** for training and validation (one plot for both: train and test accuracy).
   d. Plot the **loss** for training and validation (one plot for both: train and test loss).
   e. **Normalize** the data before feeding to the model and check how the normalization changes your accuracy sue the `StandardScaler()` from scikit learn.

Breast Cancer dataset is designated to predict if a patient has Malignant (M) or Benign = B cancer

** Follow the IPC rubric guidelines.

Submission Guidelines:

1. Once finished document your code and make sure all parts of the assignments are completed.
2. Push your code to your GitHub repo and update the ReadMe file, add your info, and partner info.
3. Submit the assignment on Canvas.
4. Present your work to TA during class time to prove the execution and complete submission.

After class submission:

1. Once finished document your code and make sure all parts of the assignments are completed.
2. Push your code to your GitHub repo and update the ReadMe file, add your info, and partner info.
3. Submit the assignment on Canvas before the deadline.
4. Record a short video (1~3) minute, proof of execution and complete assignment.
5. Add video link to ReadMe file.

Note: *Cheating, plagiarism, disruptive behavior, and other forms of unacceptable conduct are subject to strong sanctions in accordance with university policy. See detailed description of university policy at the following URL: https://catalog.umkc.edu/special-notices/academic-honesty/*