Assignment Title: Solar Forecasting using LSTM

Overview and Objective:

In this project, you will work with a baseline LSTM (Long Short-Term Memory) model provided to forecast solar power generation. Your task is to understand the existing code, identify potential areas for improvement, and implement modifications to enhance the model's forecasting accuracy in the testing phase.

The objective includes:

- To gain hands-on experience with LSTM networks and their application in time series forecasting.
- To practice data preprocessing techniques suitable for LSTM models.
- To improve upon a provided LSTM model by experimenting with its architecture, hyperparameters, and data preprocessing steps.

Provided Materials

- Baseline LSTM Model Code: A Python script using TensorFlow/Keras that defines, trains, and evaluates an LSTM model on a solar power generation dataset.
- Solar Power Generation Dataset: A dataset containing historical data on solar power output along with relevant features (e.g., weather conditions, time of day). The dataset is pre-split into training, validation, and testing sets.
- Evaluation Metrics: The baseline model's performance is evaluated using metrics such as MAE (Mean Absolute Error) and RMSE (Root Mean Squared Error).
- Check **solar_predict.py** and requirements.txt under "\Homework2_baseline\"

Tools Required:

- Python

- Visual Studio Code (VS Code)

- Python packages: tensorflow, pandas, numpy, matplotlib, keras, etc.

Tasks

- 1. Run and Analyze the Baseline Model (Month 1):
    - Execute the provided code to understand the workflow and the baseline performance metrics on the testing set.
- 2. Data Preprocessing Exploration (Month 1)
    - Experiment with different data preprocessing techniques. Consider normalizing the data, engineering new features, or handling missing values differently.
- 3. Model Architecture Adjustment (Month 2):
    - Modify the LSTM model's architecture. You might add more LSTM layers, adjust the number of units in each layer, or experiment with dropout for regularization.
- 4. Hyperparameter Tuning (Month 2):
    - Tweak hyperparameters such as learning rate, batch size, and the number of epochs. Document how each change affects model performance.
- 5. Feature Engineering (Month 3):
    - Investigate the impact of adding or removing features from the model. Consider whether certain time-based or weather-related features could improve the model's accuracy.
- 6. Re-evaluate the Model (Month 3):
    - After making your modifications, retrain the model on the training set and evaluate its performance on the testing set using the same metrics as the baseline.

Deliverables (Due on 05/24)

- Modified Code: Submit your enhanced Python script with comments explaining the changes you made to the baseline model.
- Report: A brief report that includes:
    - An overview of the modifications you implemented.
    - A comparison of the baseline and improved model's performance based on the evaluation metrics.

- A discussion on why the changes you made improved (or did not improve) the model’s performance.

Evaluation Criteria

- Accuracy Improvement: The degree to which your model's performance surpasses the baseline metrics.
- Innovation and Justification: The creativity of your approach and the rationale behind the modifications you chose.
- Clarity of Report: The quality of your report's writing, structure, and ability to convey your process and findings.

References

Below is a list of references/tutorial that may be helpful:

1. https://medium.com/grid-solutions/ai-photovoltaic-power-forecasting-using-lstm-networks-74c157649c9e

2. https://github.com/samchaaa/capstone_repo

3. https://github.com/Microsoft/CNTK/blob/master/Tutorials/CNTK_106B_LSTM_Timeseries_with_IOT_Data.ipynb