Individual Assignment: Applied Deep Learning – Dr. Philippe Blaettchen

Assignment overview

In modern industrial applications, sensors are used to observe key machine characteristics. These can help detect slight deviations and issues to avoid major system failures through targeted repairs while keeping maintenance costs under control.

Such sensors are critical in the operation of wind turbines. Due to fluctuating winds that can negatively impact the turbines and the high costs of maintenance, specifically for offshore turbines, sensor readings need to be reliably converted into an operating mode. That is, given the sensor readings over time, we want to know whether the turbine is operating correctly or whether one of several issues is present. If issues are detected (reliably), targeted efforts can be made to alleviate them before a major system failure occurs.

Your task will be to predict, as accurately as possible, the operating mode of a wind turbine based on the time series data from two sensors.

You find three pickle files as part of the assignment. After running *import pickle*, you can load the files into your notebook as follows:

```
with open('time_series_1.pickle', 'rb') as handle:
    time_series_1 = pickle.load(handle)
with open('time_series_2.pickle', 'rb') as handle:
    time_series_2 = pickle.load(handle)
with open('y.pickle', 'rb') as handle:
    y = pickle.load(handle)
```

The data represent sensor readings and operating modes for 4,000 turbine runs. *time_series_1* and *time_series_2* are NumPy arrays of shape (4000,5000), respectively. Each observation corresponds to 5,000 sensor readings from a turbine over time by one of the two sensors (*time_series_1* measures the pitch angle in each second of operation, and *time_series_2* measures the generator torque). *y* is the operating mode for each of the 4,000 turbine runs (0 if the turbine is healthy, 1 if the generator torque is faulty, 2 if the pitch angle is faulty, and 3 if both are faulty). Note that the dataset is balanced in that each operating mode is represented equally often. The overall objective is to predict the operating mode of a turbine as a function of the sensor readings.

Task description

1. Discuss what type of sequence prediction approach (sequence-to-vector, sequence-to-sequence, or encoder-decoder) is most sensible to predict the operating mode of a turbine based on the two sensor reading time series. Also describe what data shape you need to use for your chosen approach.
2. Create an iterator (ideally, a *tensorflow.data.Dataset*) that produces batches of data formatted in the appropriate way for your chosen approach.

3. Create a neural network in TensorFlow to predict the operating mode of a wind turbine based on the sensor data. Make sure that you try out different layers and elements discussed in class, such as Dense, SimpleRNN, GRU, and Conv1D.
4. We have come across Conv1D layers as a tool for analyzing time series. Different from recurrent layers such as SimpleRNN, LSTM, or GRU, when we apply a Conv1D layer to a part of a sequence, the operation does not depend on the application of the layer to previous parts of the sequence. Discuss in which types of (business) applications Conv1D layers can be particularly useful, and in which you should prefer a recurrent layer.

Another, less frequently used tool for analyzing time-series data is convolutional neural networks with 2D convolutional layers. For this to work, time series need to be converted into "images" (matrices of numbers). The paper

> "Convolutional neural network fault classification based on time series analysis for benchmark wind turbine machine" by Rahimilarki, Gao, Jin, and Zhang (published 2022 in "Renewable Energy" and available through the City-library)

describes how two-dimensional CNNs can be applied to the problem at hand. Consider sections 4 and 5 which depict the process of converting one or multiple time series into "images" used within a CNN.

5. In your own words, explain why the approach outlined here can help analyze time-series data and why it might outperform RNNs.
6. Convert the data for use with a CNN. In particular, following the approach outlined in Scenario 2 (section 5.3 of the paper) and summarized in Figure 18, convert the two time series corresponding to one wind turbine run into a single (100,100,1) array (i.e., a gray-scale image).
7. In TensorFlow, replicate the CNN with three convolutional layers displayed in Figure 12 and train it on your data. Make sure to record your final validation set accuracy.
8. Can you do better by adjusting the CNN? Be creative in your design choices (you might also consider pre-trained CNN architectures) and record your final validation set accuracy.
9. Compare the models you have created so far and select the best model (making sure to justify this). Train that model on a combined training and validation set and evaluate it on your test set. Make sure to record your final test accuracy.

Hints

- Submissions without **individual coursework submission form** filled and attached will receive 0 points.
- Answers to discussion questions (1, 4, and 5) need to be precise and strictly related to the case at hand (as well as other concrete examples for question 4). **Non-specific answers, such as those produced by ChatGPT, will be dismissed entirely.**
- **Don't worry about technical details on wind turbines**, which are not required for creating your classification model.

- Keep in mind that you have two different time series for the same observation, which need to be combined. **If you are stuck trying to combine them, start by creating a model that takes only one of them** and try to extend from there.
- Before creating any neural network, always make sure to **define a simple, yet relevant baseline** to beat.
- When creating a neural network, start with a minimum viable product (a network where the training loss continuously decreases, the validation loss decreases but eventually increases again, and you are **able to beat your baseline**).
- Only once you have completed all steps of the assignment should you go back and see how to improve your models. The performance of your models will matter for evaluation, but **not as much as having a complete answer**.
- When fine-tuning neural networks, while a certain amount of trial and error will be necessary, it is recommended that you systematically **follow the frameworks we discussed in class**. Make sure to record your final validation set accuracy.

Materials to submit

- Individual coursework submission form
- A Jupyter notebook that allows recreating your solutions.
- Your written answers, either within the Jupyter notebook or in a separate PDF. Make sure to create numbered sections within your notebook and your PDF corresponding to the task at hand.
- The trained final model chosen in task 7, as an .h5-file.

Assessment

Your submission will be evaluated against four criteria:

- appropriate use of concepts and frameworks discussed in class
- effectiveness of the proposed answer/solution
- originality and creativity of the proposed answer/solution
- organization and clarity of submitted materials