Assignment 08
Long-Short Term Memory (LSTM) Networks
EECE-6822 Machine Learning

Cris Ababei
Electrical and Computer Engr., Marquette University

1. Objective
The objectives of this activity include: (1) run several code examples that work with RNN networks; implementations both in Keras/TF2 and Pytorch; and (2) use.

2. Prerequisite Readings
Murphy
- Ch. 15.1-15.2: neural networks for sequences

Geron
- Ch.15: processing sequences using RNNs and CNNs

Raschka
- Ch.15: modeling sequential data using recurrent neural networks

3. Code Examples

Example 1: RNNs in Keras/TF2

This is the example code from Ch.15 from Aurelian Geron's book.
[*B3-Geron] Aurelien Geron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly, 2022.
Open in your google colab and go through the code. You should read first the chapter itself from the book, before going through the code.
https://github.com/ageron/handson-ml3/blob/main/15_processing_sequences_using_rnns_and_cnns.ipynb

The chapter looks at concepts underlying RNNs and how to train them using backpropagation through time. Then, it uses them to forecast a time series, and how to implement them using **Keras**. A dataset that is used is ridership data from Chicago's Transit Authority.

Example 2: RNNs in Pytorch

This code example is from Ch.15 from:
[*B3-Raschka] Sebastian Raschka, Yuxi Liu, and Vahid Mirjalili, Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python, Packt Publishing, 2022.
The source code (both Python code an Jupyter Notebook) is located at the GitHub repository. You should read first the chapter itself from the book, before going through the code.
https://github.com/rasbt/machine-learning-book/tree/main/ch15

To execute the code, I found the easiest usually to actually run directly the Python code (same code as in the notebook) in Anaconda Spyder.

> ***NOTE 1:*** *However, the code in some chapters (like this one) may not run right away due to differences/mismatches in torch, torchtext, torchaudio, torchvision, etc. I had to do quite a few acrobatics to make it work in Anaconda Spyder, including:*
>
> *--Because of mismatches in versions of torch and torchtext, I had to create a new anaconda environment for this chapter code to work.*
>
> *--So, in an anaconda shell (ad Admin) run:*

>conda create -n torch23 python=3.10 -y

>conda activate torch23

>pip install "numpy==2.0.2" "fsspec==2024.12.0" s3fs==2024.12.0

>pip install --index-url https://download.pytorch.org/whl/cpu torch==2.3.0 torchvision==0.18.0 torchaudio==2.3.0 torchtext==0.18.0

*>pip install spyder-kernels==3.0.**

>pip install torchdata==0.7.1

>pip install "portalocker>=2.0.0"

>pip install "numpy==2.0.0"

--Restart Anaconda Spyder, and set as environment "torch23" by doing:

--Go to Tools -> Preferences

--In the Preferences window, select Python interpreter from the left-hand menu.

--Choose the option Use the following Python interpreter:

--Paste the path: C:\anaconda3\envs\torch23\python.exe

--Click Apply and then OK.

***NOTE 2:** The examples in this chapter have significantly longer runtimes (esp. if using cpu and no gpu). Reducing num_epochs could address that in part, but, results will not be as good.*

So, it may be the lesser of a pain to run the Jupyter Notebook in google colab (where you will need to upload additional files, figures, to have all code boxes running smoothly).

This chapter explores recurrent neural networks (RNNs) and their application in modeling sequential data. More specifically, it covers:

- Introducing sequential data
- RNNs for modeling sequences
- Long short-term memory
- Truncated backpropagation through time
- Implementing a multilayer RNN for sequence modeling in PyTorch
- Project one: RNN sentiment analysis of the IMDb movie review dataset
- Project two: RNN character-level language modeling with LSTM cells, using text data from Jules Verne's The Mysterious Island
- Using gradient clipping to avoid exploding gradients

Example 3: RNNs and LSTMs in TF2 and Pytorch

These are examples suggested by K. Murphy on his github repository for the textbook:
https://github.com/probml/pyprobml/tree/master/notebooks/book1/15
Scroll to the bottom of the above page to find several notebook examples with **TF2** (RNNs) and **Pytorch** (LSTM) implementation examples!

Example 4: mlm tutorials and code examples on NNs

In the last part, we look at several tutorials from **machinelearningmastery (mlm)**. Read the following tutorials and run the code where applicable:

--How to Develop LSTM Models for Time Series Forecasting

https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/

--Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras

https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/

-- How to Develop Multi-Step LSTM Time Series Forecasting Models for Power Usage

https://machinelearningmastery.com/how-to-develop-lstm-models-for-multi-step-time-series-forecasting-of-household-power-consumption/

--Multivariate Time Series Forecasting with LSTMs in Keras

https://machinelearningmastery.com/multivariate-time-series-forecasting-lstms-keras/

--How to Tune LSTM Hyperparameters with Keras for Time Series Forecasting

https://machinelearningmastery.com/tune-lstm-hyperparameters-keras-time-series-forecasting/

4. Assignment

First, you must read and do the following Tutorial 1 (T1):

> How to Develop Multi-Step LSTM Time Series Forecasting Models for Power Usage
>
> https://machinelearningmastery.com/how-to-develop-lstm-models-for-multi-step-time-series-forecasting-of-household-power-consumption/

from the beginning to and including the section "LSTM Model With Univariate Input and Vector Output". This is to learn and develop a good understanding of the model from section **"LSTM Model With Univariate Input and Vector Output"**.

Second, you must read and do Tutorial 2 (T2):

> How to Tune LSTM Hyperparameters with Keras for Time Series Forecasting
>
> https://machinelearningmastery.com/tune-lstm-hyperparameters-keras-time-series-forecasting/

from the beginning to and including the section "**Tuning the Number of Epochs**"; you can stop at section "Tuning the Batch Size".

Finally, you must create a new Python program (or Jupyter Notebook) where you must do the experiment from T2 in section "Tuning the Number of Epochs", but, on the model and dataset from T1, for values of **epochs = {10, 50, 100, 200, 500}**.

To simplify(?) this assignment, you can change the model from T1 to output only one value (i.e., energy for next day) instead of currently 7 values (energy for next 7 days).

You must create a plot similar to the one from T2 (shown in Figure 1 below), but for the new values of epoch and for the new model.

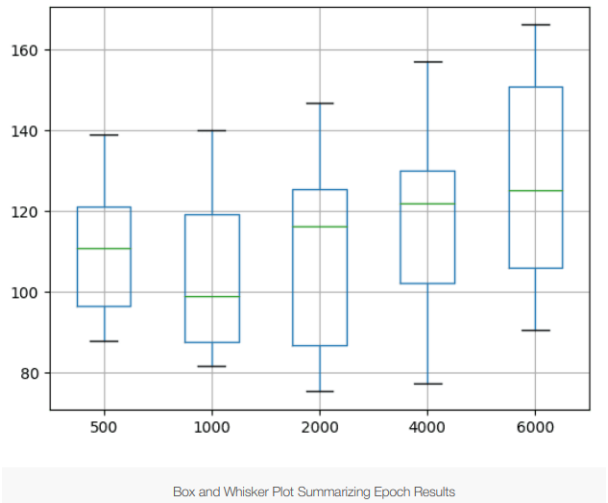

Figure 1: Plot that must be replicated for the problem in this assignment.

5. Deliverables

You must write (typed) a report and upload it as a PDF file on D2L. The report should be named "**hw8_report_LastName.pdf**". You should also create a .zip archive with all your code and implementations of all parts of the assignment. Upload also this archive .zip file with the name "**hw8_implementation_code_LastName.zip**" to D2L. Hence, your D2L should contain two items: the report and the .zip file. **Do not include the report inside the .zip and upload only the .zip. They should be two separate items!**

The report should include the following sections and subsections. Make sure section titles are in bold font and pages are numbered.

1) **Title + course info + your name**
2) **Summary.** Describe in one paragraph what the objective of the assignment is.
3) **Description of Experiments and Discussion.** Describe the experiments you did. All tables and figures should be numbered and should have captions. All plots in all figures should have axes labels and titles. Present a meaningful discussion with the interpretation of the results you obtained. Explain if you expected the results or not; discuss the intuition behind it.
4) **Conclusion.** Present your conclusions; highlight what are your main takeaways that you learned from this assignment. Describe what issues you encountered and how you solved them.
5) **References.** Include all references that you used, as a numbered list. Cite them in the report itself; do not just list them! If your report has References that are not numbered and cited in the report, points will be deducted!