

CSE4/510: Applied Deep Learning

Summer 2020

Instructor: Alina Vereshchaka

Project 4 -- Anomaly Detection

Deadline: August 4, Tue, 11:59pm

Description

In this assignment we will practice to detect anomalies in the benchmark dataset. We will explore deep learning approaches that include building a sequence to sequence MLP and autoencoder.

Dataset

NAB (Numenta Anomaly Benchmark) is a novel benchmark for evaluating algorithms for anomaly detection in streaming, real-time applications. It is composed of over 50 data files designed to provide data for research in streaming anomaly detection. It is comprised of both real-world and artificial timeseries data containing labeled anomalous periods of behavior.

<https://github.com/numenta/NAB/tree/master/data>

Tasks

Part I: MLP for Anomaly Detection [50 points]

1. Choose any dataset from NAB (except those used in class) and prepare it for training (normalize, split between train/test/validation). Explore the dataset by visualizing it and showing statistical parameters about it.
2. Build an MLP/LSTM model for predicting a sequence of values (min 5 values). Work with 3 different setups of the window size and the size of the output sequence.
3. Using 3 different loss/distance measures identify the anomalies in the dataset.

Compare the measurements.

4. Discuss the results and provide the graphs, e.g. train vs validation accuracy and loss over time. Show a confusion matrix (normal vs anomaly).

Part II: Autoencoder for Anomaly Detection [50 points]

1. Build a Autoencoder model for predicting a sequence of values. Show 3 different Autoencoder setups (e.g. using Dense/LSTM/Conv1D layers).
2. For one of the model built in 1 show the process of hyperparameters tuning (e.g. thresholds, # of layers, activation functions).
3. Discuss the results and provide the graphs, e.g. train vs validation accuracy and loss over time. Show the confusion matrix.

Submit the Project

- Submit at **UBLearns > Assignments**
- The code of your implementations should be written in Python. You can submit multiple files, but they all need to have a clear name
- All project files should be packed in a ZIP file named **TEAMMATE#1_UBIT_TEAMMATE#2_UBIT_project4.zip** (e.g. **avereshc_neelamra_project4.zip**).
- Your Jupyter notebook should be saved with the results. If you are submitting python scripts, after extracting the ZIP file and executing command `python main.py` in the first level directory, all the generated results and plots you used in your report should appear printed out in a clear manner.
- In your report include the answers to questions for each part. You can complete the report in a separate pdf file or in Jupyter notebook along with your code.
- Include all the references that have been used to complete the project.

Important Information

This project can be done in a team of up to two people. The standing policy of the Department is that all students involved in an academic integrity violation (e.g. plagiarism in any way, shape, or form) will receive an F grade for the course. Refer to the [Academic Integrity website](#) for more information.

Late Days Policy

You can use up to 5 late days throughout the course that can be applied to any project. You don't have to inform the instructor, the late submission will be tracked in UBLearn. If you work in teams the late days used will be subtracted from both partners. E.g. you have 4 late days and your partner has 3 days left. If you submit one day after the due date, you will have 3 days and your partner will have 2 days left.

Important Dates

August 4, 11:59pm - Project 4 is Due