Advanced Machine Learning (F90AM) Coursework

This assessment accounts for 50% of your final course mark.

Due: 17:00 – UK time, Friday, 28 March 2025 (Week 11)

Overview

The objectives of this coursework are the following

- Apply dimensionality reduction and clustering techniques with Autoencoder and hierarchical clustering to tabular data for identifying patterns or groupings
- Implement and evaluate an MLP classifier for categorising entities based on tabular features
- Design, train, and compare LSTM, CNN-LSTM, and Transformer models for time-series forecasting
- Build and use a Variational Autoencoder (VAE) to generate synthetic time-series data for data augmentation.
- Analyse the impact of augmented datasets on model performance and derive actionable insights.

Please read through the following important points before you begin:

This is an assessed coursework. You are allowed to discuss this assignment with other students, but you should not copy their work, and you should not share your work with other students. Plagiarism is not acceptable. Please review the Herriot Watt's plagiarism policy here. https://www.hw.ac.uk/uk/services/academic-registry/academic-integrity/academic-misconduct.htm We will be carrying out automated plagiarism checks on both code and text submissions.

Special note for re-using existing code or large language models. If you are re-using code that you have not written, then this must be indicated, making clear which parts were not written by you and clearly stating where it was taken from. If your code is found elsewhere by the person marking your work, and you have not mentioned this, you may find yourself having to go before a disciplinary committee and face grave consequences.

Late submission and extensions. Late submissions will be marked according to the university's late submissions policy, i.e. a 30% deduction if submitted within five working days of the deadline and a mark of 0% after that. The deadline for this work is not negotiable. If you are unable to complete the assignment by the deadline due to circumstances beyond your control (e.g. illness or family bereavement), you should complete and submit a mitigating circumstances application: https://www.hw.ac.uk/uk/students/studies/examinations/mitigating-circumstances.htm

(continues on next page…)

Detailed Description

1. Quick Summary

In this coursework, you will implement different machine-learning algorithms to perform different tasks with a given dataset. You will run various experiments, display results, and evaluate and compare performance. You will also write a report to present your work. Your submission must include a Zip file containing all Python code files and the PDF report. The Zip must include Python files and a Jupyter Notebook file to implement the full workflow of the coursework. You will complete this project in groups of two students.

The structure of your report and Jupyter Notebook file should follow the order of the tasks described below, with appropriate subsection titles. For any clarifications on this assessment, please ask Dr Wei Pang (w.pang@hw.ac.uk, Edinburgh Campus), Dr William Yoo (w.yoo@hw.ac.uk, Malaysia Campus), or Dr Hadj Batatia (h.batatia@hw.ac.uk, Dubai Campus)

2. Programming Language

You will need to use Python and PyTorch to complete the coursework. The code can be in separate Python script files. You also use a Jupyter Notebook to import Python files, run your experiments, save, and present your experimental results.

3. Data - World Bank Data

In this coursework, you will analyse data from the World Bank. This repository (https://data.worldbank.org/) has a large volume of macro-economic, social and environmental data from all countries, regions, groups, and organisations for many years. It consists of many (89) databases. We are primarily interested in the "World Development Indicators" database (number 2 in the World Bank series). The repository contains 1500 indicators. Each indicator has an ID and a name; e.g., ID "SP.POP.65UP.FE.ZS" corresponds to the name "Population ages 65 and above, female (% of female population)". A metadata file is provided to describe each indicator.

You can access the full dataset (more than 400 MB) and the metadata file here: CW_handouts. Please note that this is a very large CSV file that cannot be fully opened with Excel due to its volume. You can manipulate it (for example, extract only some indicators) by writing custom code. You can also download selectively (only the indicators and countries of interest) from the World Bank using an API. There are several APIs in different languages (R, Python, Excel…). You can find information on the World Bank web page (https://data.worldbank.org/products/third-party-apps).

We have used the Python API named wbdata (https://wbdata.readthedocs.io/en/stable/) to download a subset of the dataset. You can find this subset on the Canvas page. The subset has 221 countries, and the following indicators recorded yearly over the period 01/01/1980 to 31/12/2023:

- "NY.GDP.PCAP.PP.KD": "GDPpc_2017$", #"GDP per capita, PPP (constant 2017 US$)
- "SP.POP.TOTL": "Population_total"
- "SP.DYN.LE00.IN": "Life_exectancy"
- "SE.ADT.LITR.ZS": "Literacy_rate"
- "SL.UEM.TOTL.ZS": "Unemploymlent_rate"
- "EG.USE.PCAP.KG.OE": "Access_electricity"
- "SP.DYN.TFRT.IN": "Fertility_rate"
- "SI.POV.NAHC": "Poverty_ratio"

- "SE.PRM.ENRR": "Primary_school_enrolmet_rate"
- "EG.USE.PCAP.KG.OE": "Energy_use"
- “NE.EXP.GNFS.KD": "Exports_2017$" #Exports of goods and services (current US$)

The Python script to download this subset is also provided on Canvas. You are encouraged to modify it to download and study other indicators, countries, or periods.

3. Detailed Tasks

The coursework has a series of tasks to experiment with different machine-learning models. The models can be implemented in any order you wish. However, they should be reported in the order of the tasks described below. In particular, you should delay task 2 on the autoencoder after week 7, waiting for the lecture and lab on the topic.

Task 1. Data pre-processing and exploration [5 marks]

The data come as a time series with countries, years, and yearly values of the economic indicators. The subset has a few indicators. You can add more relevant indicators from the full dataset as you feel relevant from the economic point of view. You must explore the dataset, normalise, and fill in the missing data. You must plot each indicator for the following countries: USA, China, Russia, and Brazil.

To feed the data to the different deep learning models, you should create sequences of 5 years using a sliding window of size 5 years that overlap every 4 years (i.e., you shift the window one year at a time). For ten indicators over 44 years (1980 to 2023), you should obtain 40 sequences per country, each having 50 values.

The report must include a description of the pre-processing and data exploration you carried out.

Task 2. Auto-encoder and clustering [5 marks]

This task aims to cluster countries according to their economic indicators using K-means. An auto-encoder is applied to the data to extract the latent representation. Then, the latent vectors are fed to k-means to perform the clustering. For this, the pre-processed data should be processed further to create one single sequence for each country. This is done by aggregating the 40 sequences. Aggregation can be done by calculating statistics like median, mean, max, variance, or kurtosis of each of the 50 dimensions or by applying a PCA.

You must design and implement an Auto-encoder model to create a latent representation of countries. The model must be trained on the aggregated data to obtain the latent vectors of each country.

You must run a k-means clustering method on the extracted latent vectors to cluster countries.
You should evaluate the quality of the clustering using standard metrics, apply a dimension reduction method like t-SNE or UMAP, and visualise the cluster.

Your report must include descriptions of the autoencoder model, the clustering method, and the results.

Task 3. MLP [5 marks]

This task aims to classify countries into four levels of development. Consider the average GDP of each country. You will label countries according to their level of development. You assign one of the labels “Under-developed”, “Developing”, “Emerging”, or “Developed” to each country by considering their GDP. You do this by dividing the GDP into 4 equal ranges, corresponding to the four classes. And then, label each country by the class corresponding range to which its GDP belongs.

Similarly to the autoencoder, you must aggregate the data (excluding the GDP) but use only statistics and not PCA (as you need to preserve the correlations between the original data).

Implement a Multi-layer Perceptron model and train it using the newly aggregated and labelled data. The dataset must be split into training and testing subsets, and a proper cross-validation procedure must be applied to prevent overfitting and estimate hyperparameters, especially the learning rate.

Your report must describe the MLP, the training and validation procedure and the results. Adequate metrics must be used to evaluate the performance.

Task 4. Time-series forecast [20 marks]

This task aims to forecast the GDP per country for the next 5 years based on the past 10 years using three models: LSTM, CNN-LSTM, and Transformer. The results of these three models must be compared and discussed.

Using the original data, you need to create sequences of input and output. We will consider a window of 10 years for the input, where we include all indicators, including the GDP. You must consider an output window of 5 years, that includes only the target values to forecast, in our case, the GDP for the next 5 years. For each country, use the sliding window to create sequences of the input and output pairs. Each input sequence should be a matrix of size length_of_input_window x number_of_features, in our case, 10 x 10. Each output sequence should be a vector of size length_of_output_window corresponding to the target GDP values, in our case, 5. Normalise the data using Min/Max or Standard scaler across all countries and years. Then, split the data into training, validation and testing sets. [5 marks]

Implement the LSTM, CNN-LSTM and Transformer models [5 marks each]:

- LSTM must have the input size batch_size x length_of_input_window x number_of_features and output of size batch_size x length_of_output_window
- CNN-LSTM must have an input size batch_size x length_of_input_window x number_of_features. First, you must apply CNN layers with 1D convolution filters and 1D pooling along the time axis to extract features. The output size of the CNN would be batch_size x pooled_sequence_length x number_of_filters. This output is fed to the LSTM to model the sequences.
- Transformer takes an input of size batch_size x length_of_input_window x number_of_features. Embeddings, positional encodings and self-attention mechanisms should be used to model temporal dependencies.

The three models must be trained, validated, and tested on the same data sets. Your report must describe the three implemented models, the performance metrics (MSE or MAE, or preferably MAPE), and the actual results in tables and curve plots.

Task 5. Data augmentation [7 marks]

This task aims to learn an explicit probability distribution of the latent space of the data. The original data is used to train a variational auto-encoder to learn the probability distribution of the latent space. In the next task of this coursework, the decoder will be used to generate synthetic data instances to augment data.

Using the same training data (pairs of input and output sequences) fed to the LSTM, you must concatenate each input sequence and its corresponding output sequence and flatten the result. The dimension of each sequence should be length_of_input_window * number_of_features + length_of_output_window, in our case 10 x 10 + 5 = 105.

Implement a variational autoencoder model. Train the VAE to reconstruct the concatenated and flattened sequences.

Your report must describe the variational autoencoder architecture and its performance. You must include the reconstruction loss, the KL divergence, and the ELBO. In addition, you must illustrate the distribution of the latent space. For this purpose, you should extract the latent representations from the VAE and apply t-SNE to reduce the dimension to 3. You must visualise the reduced dimensions (3D) using a 3D scatter plot.

Task 6. Comparing forecasts with and without data augmentation [8 marks]

This task aims to augment the original data. The VAE decoder, trained in task 5, must be used to generate several synthetic sequences. You must split the generated sequences to make the input and output pairs. Combine the original and generated data to retrain the LSTM, CNN-LSTM, and Transformer models. Testing must be done using the same set (made of original data) used for the previous LSTM, CNN-LSTM, and Transformer in task 4.

Your report must include the performance of the LSTM, CNN-LSTM, and Transformer to forecast the GDP for the next 5 years. You must draw combined plots with predictions using original data and combined data. You must also include a table to compare the performance of the three models with and without data augmentation.

Your report must include a discussion section where you reflect on the results obtained from the different tasks. You must use references from the literature to support your discussion and interpretation of the results. [3 marks]

4. Submission

You must submit the following items to CANVAS:

1. Your report is in PDF format. Your report should
   - Submit the “Declaration of Authorship” quiz on Canvas and include “The Declaration of Authorship” on the first page of your report or as a separate file within the zip. Note: No marks will be awarded without the declaration of authorship.
   - The first page must provide the names, IDs and contributions of the group members. Contributions must be reported as a list of tasks carried out by each member, with possibly a percentage of work if the tasks are shared.
   - Be no more than eight pages in length (max of 3000 words, not including the declaration of authorship and references). You should take this into account when planning your experiments. If you have more results than you have space for, select the results you consider the most insightful and briefly mention other experiments you carried out.
   - Be written in Arial or a similar font, with a minimum font size of 12.
   - Include useful references to the wider literature. For instance, you might use references to books or papers to justify implementation or hyperparameter choices, or you could compare your findings to those reported elsewhere. Use standard referencing styles for this.

2. Python source code files.
   - All source code files should be compressed as a single zip file and submitted to CANVAS. The zip file should be named "F90AM_GXX_1H0XXXXXXX_2H0XXXXXXX.zip", where GXX is the group number, 1H0XXXXXX stands for the ID number of student 1, and 2H0XXXXXX for the ID number of student 2.
   - The source code files should include Python script files for the different models (each implemented only once), Python files to pre-process the data, and a Jupyter Notebook file named "F90AM_GXX_1H0XXXXXXX_2H0XXXXXXX.ipynb" that includes the overall flow of processing. The Notebook must import the Python files.

5. Marking Criteria

- Marks for the different tasks are assigned based on the code, the quality of the results, and the presentation of the report.
- If the contributions of the group members are significantly different, adjustments to the marks will be made to reflect the contributions.
- We will look at how you implement and configure your algorithms, how you pre-process the data, and how you implement the evaluation of the performance.
- We will also examine clarity and reproducibility: how easily a third person can understand and reproduce your results based on your report and code. Specifically,
  - The quality of your report, including organisation, quality of writing, and brevity.
  - The quality of your code, including quality of documentation, organisation, readability, efficiency, and ease of maintenance.
- We will look at the design and report of your experiments (including the appropriate use of tables and figures) and the depth of your analysis, including critical thinking.
- We will award going the extra mile, for example, an advanced hyperparameter tuning algorithm was used to finetune the hyperparameters so that the machine learning model can achieve better performance.
- In specific circumstances (e.g., if we suspect submitted coursework was not done by the students), we may call the group for a technical interview to assess their comprehension of the submitted work. If a student fails to demonstrate a good understanding of their submitted work, their mark may be reduced; in case of severe violations, we may report the case to the school for disciplinary investigation.