COMP9414 24T2
Artificial Intelligence

Assignment 1 - Artificial neural networks

Due: Week 5, Wednesday, 26 June 2024, 11:55 PM.

1 Problem context

Time Series Air Quality Prediction with Neural Networks: In this assignment, you will delve into the realm of time series prediction using neural network architectures. You will explore both classification and estimation tasks using a publicly available dataset.

You will be provided with a dataset named "Air Quality," [1] available on the UCI Machine Learning Repository [1]. We tailored this dataset for this assignment and made some modifications. Therefore, please only use the attached dataset for this assignment.

The given dataset contains 8,358 instances of hourly averaged responses from an array of five metal oxide chemical sensors embedded in an air quality chemical multisensor device. The device was located in the field in a significantly polluted area at road level within an Italian city. Data were recorded from March 2004 to February 2005 (one year), representing the longest freely available recordings of on-field deployed air quality chemical sensor device responses. Ground truth hourly averaged concentrations for carbon monoxide, non-methane hydrocarbons, benzene, total nitrogen oxides, and nitrogen dioxide among other variables were provided by a co-located reference-certified analyser. The variables included in the dataset

[1] https://archive.ics.uci.edu/dataset/360/air+quality

are listed in Table 1. **Missing values within the dataset are tagged with -200 value.**

Table 1: Variables within the dataset.

Variable	Meaning
CO(GT)	True hourly averaged concentration of carbon monoxide
PT08.S1(CO)	Hourly averaged sensor response
NMHC(GT)	True hourly averaged overall Non Metanic HydroCarbons concentration
C6H6(GT)	True hourly averaged Benzene concentration
PT08.S2(NMHC)	Hourly averaged sensor response
NOx(GT)	True hourly averaged NOx concentration
PT08.S3(NOx)	Hourly averaged sensor response
NO2(GT)	True hourly averaged NO2 concentration
PT08.S4(NO2)	Hourly averaged sensor response
PT08.S5(O3)	Hourly averaged sensor response
T	Temperature
RH	Relative Humidity
AH	Absolute Humidity

2 Activities

This assignment focuses on two main objectives:

- Classification Task: You should develop a neural network that can predict whether the concentration of Carbon Monoxide (CO) exceeds a certain threshold – the mean of CO(GT) values – based on historical air quality data. This task involves binary classification, where your model learns to classify instances into two categories: above or below the threshold. To determine the threshold, you must first calculate the mean value for CO(GT), excluding unknown data (missing values). Then, use this threshold to predict whether the value predicted by your network is above or below it. You are free to choose and design your own network, and there are no limitations on its structure. However, your network should be capable of handling missing values.

- Regression Task: You should develop a neural network that can predict the concentration of Nitrogen Oxides (NOx) based on other air quality features. This task involves estimating a continuous numerical value (NOx concentration) from the input features using regression techniques. You are free to choose and design your own network and there is no limitation on that, however, your model should be able to deal with missing values.

In summary, the classification task aims to divide instances into two categories (exceeding or not exceeding CO(GT) threshold), while the regression task aims to predict a continuous numerical value (NOx concentration).

2.1 Data preprocessing

It is expected you analyse the provided data and perform any required preprocessing. Some of the tasks during preprocessing might include the ones shown below; however, not all of them are necessary and you should evaluate each of them against the results obtained.

(a) Identify variation range for input and output variables.

(b) Plot each variable to observe the overall behaviour of the process.

(c) In case outliers or missing data are detected correct the data accordingly.

(d) Split the data for training and testing.

2.2 Design of the neural network

You should select and design neural architectures for addressing both the classification and regression problem described above. In each case, consider the following steps:

(a) Design the network and decide the number of layers, units, and their respective activation functions.

(b) Remember it's recommended your network accomplish the maximal number of parameters Nw $<$ (number of samples)/10.

(c) Create the neural network using Keras and TensorFlow.

2.3 Training

In this section, you have to train your proposed neural network. Consider the following steps:

(a) Decide the training parameters such as loss function, optimizer, batch size, learning rate, and episodes.

(b) Train the neural model and verify the loss values during the process.

(c) Verify possible overfitting problems.

2.4 Validating the neural model

Assess your results plotting training results and the network response for the test inputs against the test targets. Compute error indexes to complement the visual analysis.

(a) For the classification task, draw two different plots to illustrate your results over different epochs. In the first plot, show the training and validation loss over the epochs. In the second plot, show the training and validation accuracy over the epochs. For example, Figure 1 and Figure 2 show loss and classification accuracy plots for 100 epochs, respectively.

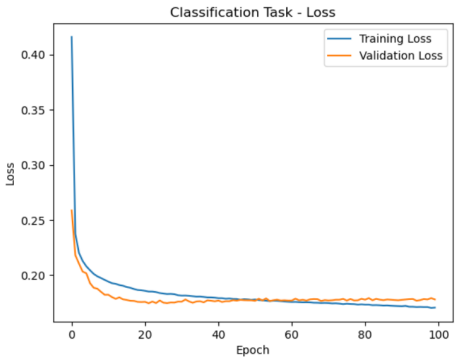

Figure 1: Loss plot for the classification task

Figure 2: Accuracy plot for the classification task

(b) For the classification task, compute a confusion matrix [2] including True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN), as shown in Table 2. Moreover, report accuracy and precision for your test data and mention the number of tested samples as shown in Table 3 (the numbers shown in both tables are randomly chosen and may not be consistent with each other). For instance, Sklearn library offers a various range of metric functions [3], including confusion matrix [4], accuracy, and precision. You can use Sklearn in-built metric functions to calculate the mentioned metrics or develop your own functions.

Table 2: Confusion matrix for the test data for the classification task.

Confusion Matrix	Positive (Actual)	Negative (Actual)
Positive (Predicted)	103	6
Negative (Predicted)	6	75

Table 3: Accuracy and precision for the test data for the classification task.

	Accuracy	Precision	Number of Samples
CO(GT) classification	63%	60%	190

(c) For the regression task, draw two different plots to illustrate your results. In the first plot, show how the selected loss function varies for both the training and validation through the epochs. In the second plot, show the final estimation results for the validation test. For instance, Figure 3 and Figure 4 show the loss function and the network outputs vs the actual NOx(GT) values for a validation test, respectively. In Figure 4 no data preprocessing has been performed, however, as mentioned above, it is expected you include this in your assignment.

(d) For the regression task, report performance indexes including the Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) (see a discussion on [2]), and the number of samples for your estimation of

[2] https://en.wikipedia.org/wiki/Confusion_matrix
[3] https://scikit-learn.org/stable/api/sklearn.metrics.html
[4] https://scikitlearn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html

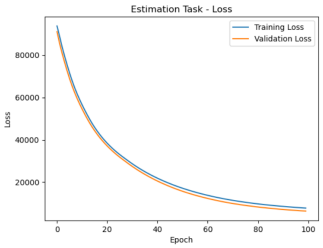

Figure 3: Loss plot for the regression task.

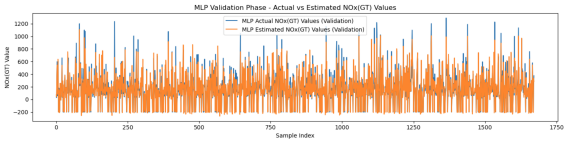

Figure 4: Estimated and actual NOx(GT) for the validation set.

NOx(GT) values in a table. Root Mean Squared Error (RMSE) measures the differences between the observed values and predicted ones and is defined as follows:

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{i=n}(Y_i - \hat{Y}_i)^2}, \tag{1}$$

where n is the number of our samples, Y_i is the actual label and $\hat{Y}_i$ is the predicted value. In the same way, MAE can be defined as the absolute average of errors as follows:

$$MAE = \frac{1}{n}\Sigma_{i=1}^{i=n}|Y_i - \hat{Y}_i|. \tag{2}$$

Table 4 shows an example of the performance indexes (all numbers are randomly chosen and may not be consistent with each other). As mentioned before, Sklearn library offers a various range of metric functions, including RMSE[5] and MAE [6]. You can use Sklearn in-built metric functions to calculate the mentioned metrics or develop your own functions.

Table 4: Result table for the test data for the regression task.

RMSE	MAE	Number of Samples
90.60	50.35	55

[5]https://scikit-learn.org/stable/modules/generated/sklearn.metrics.root_mean_squared_error.html
[6]https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_absolute_error.html

3 Testing and discussing your code

As part of the assignment evaluation, your code will be tested by tutors along with you in a discussion session carried out in the tutorial session in week 6. The assignment has a total of 25 marks. The discussion is mandatory and, therefore, we will not mark any assignment not discussed with tutors.

You are expected to propose and build neural models for classification and regression tasks. The minimal output we expect to see are the results mentioned above in Section 2.4. You will receive marks for each of these subsection as shown in Table 5, i.e. 7 marks in total. However, it's fine if you want to include any other outcome to highlight particular aspects when testing and discussing your code with your tutor.

For marking your results, you should be prepared to simulate your neural model with a generalisation set we have saved apart for that purpose. You must anticipate this by including in your submission a script ready to open a file (with the same characteristics as the given dataset but with fewer data points), simulate the network, and perform all the validation tests described in Section 2.4 (b) and (d) (accuracy, precision, RMSE, MAE). It is recommended to save all of your hyper-parameters and weights (your model in general) so you can call your network and perform the analysis later in your discussion session.

As for the classification task, you need to compute accuracy and precision, while for the regression task RMSE and MAE using the generalisation set. You will receive 3 marks for each task, given successful results. Expected results should be as follows:

- For the classification task, your network should achieve at least 85% accuracy and precision. Accuracy and precision lower than that will result in a score of 0 marks for that specific section.
- For the regression task, it is expected to achieve an RMSE of at most 280 and an MAE of 220 for unseen data points. Errors higher than the mentioned values will be marked as 0 marks.

Finally, you will receive 1 mark for code readability for each task, and your tutor will also give you a maximum of 5 marks for each task depending on the level of code understanding as follows: **5. Outstanding, 4. Great, 3. Fair, 2. Low, 1. Deficient, 0. No answer**.

Table 5: Marks for each task.

Task	Marks
Results obtained with given dataset	
Loss and accuracy plots for classification task	2 marks
Confusion matrix and accuracy and precision tables for classification task	2 marks
Loss and estimated NOx(GT) plots for regression task	2 marks
Performance indexes table for regression task	1 mark
Results obtained with generalisation dataset	
Accuracy and precision for classification task	3 marks
RMSE and MAE for regression task	3 marks
Code understanding and discussion	
Code readability for classification task	1 mark
Code readability for regression task	1 mark
Code understanding and discussion for classification task	5 mark
Code understanding and discussion for regression task	5 mark
Total marks	25 marks

4 Submitting your assignment

The assignment must be done individually. You must submit your assignment solution by Moodle. This will consist of a single .ipynb Jupyter file. This file should contain all the necessary code for reading files, data preprocessing, network architecture, and result evaluations. Additionally, your file should include short text descriptions to help markers better understand your code. Please be mindful that providing clean and easy-to-read code is a part of your assignment.

Please indicate your full name and your zID at the top of the file as a comment. You can submit as many times as you like before the deadline – later submissions overwrite earlier ones. After submitting your file a good practice is to take a screenshot of it for future reference.

Late submission penalty: UNSW has a standard late submission penalty of 5% per day from your mark, capped at five days from the assessment deadline, after that students cannot submit the assignment.

5 Deadline and questions

Deadline: Week 5, Wednesday 26 June of June 2024, 11:55pm. Please use the forum on Moodle to ask questions related to the project. We will prioritise questions asked in the forum. However, you should not share your code to avoid making it public and possible plagiarism. If that's the case, use the course email `cs9414@cse.unsw.edu.au` as alternative.

Although we try to answer questions as quickly as possible, we might take up to 1 or 2 business days to reply, therefore, last-moment questions might not be answered timely.

6 Plagiarism policy

Your program must be entirely your own work. Plagiarism detection software might be used to compare submissions pairwise (including submissions for any similar projects from previous years) and serious penalties will be applied, particularly in the case of repeat offences.

Do not copy from others. Do not allow anyone to see your code. Please refer to the UNSW Policy on Academic Honesty and Plagiarism if you require further clarification on this matter.

References

[1] De Vito, S., Massera, E., Piga, M., Martinotto, L. and Di Francia, G., 2008. On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario. Sensors and Actuators B: Chemical, 129(2), pp.750-757.

[2] Hodson, T. O. 2022. Root mean square error (RMSE) or mean absolute error (MAE): When to use them or not. Geoscientific Model Development Discussions, 2022, 1-10.