National Tsing Hua University

11220IEEM 513600

Deep Learning and Industrial Applications

Homework 4

Name: 林芮伃　　　　Student ID: 112030509

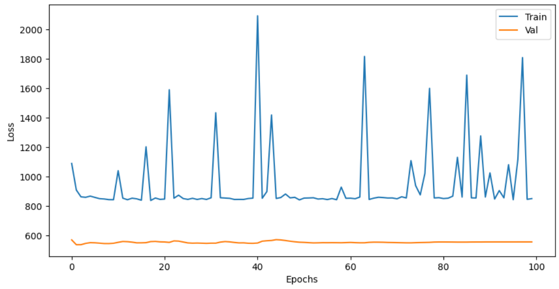
Due on 2024/05/02.
Note: DO NOT exceed 3 pages.

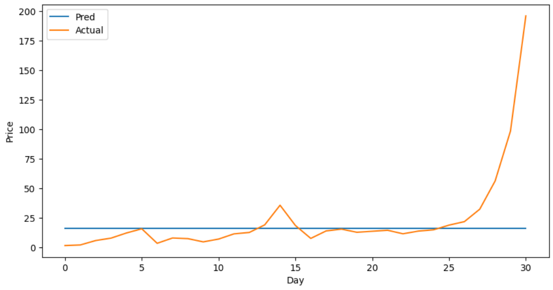
1. (15 points) Experiment with different window sizes and steps. Train the model using **3** different combinations of window size and step. Evaluate the Mean Squared Error (MSE) for each configuration. Report the MSEs using a table and analyze the results. (Approximately 100 words.)

combinations	window size	step	MSE
Original	10	15	109.7450
1	10	20	701.5697
2	5	15	44.8302
3	15	10	109.4222

Increasing the step size typically leads to an increase in the model's MSE, possibly because the increased step size results in the model capturing less information or losing some temporal correlations. Decreasing the window size helps the model capture shorter-term patterns or features, thereby improving the model's predictive accuracy. Increasing the window size while decreasing the step size may lead to some improvement in the model's predictive performance, but the extent of improvement may not be significant.

2. (Approximately 200 words.)
   (i) (15 points) Include 'Volume' as an additional input feature in your model. Discuss the impact of incorporating 'Volume' on the model's performance.





Adding the 'Volume' column as an additional input feature can provide more comprehensive information for understanding market conditions, especially in stock or financial data analysis. However, after adding the 'Volume' column as an extra input feature, if the model's performance graph shows that the val and Pred lines become a straight line and the model fails to predict accurately, it may indicate the need for more detailed data processing, such as standardization or regularization, to ensure consistent scaling among different features. Additionally, over-reliance on this feature could lead to overfitting issues, resulting in poor performance on test data.

(ii) (15 points) Explore and report on the best combination of input features that yields the best MSE. Briefly describe the reasons of your attempts and analyze the final, optimal input combination.

input features	MSE
Open	396.9668
High	866.7266
Low	511.6915
Close	876.0742
Volume	1069.8758
Open, Low	20.5353
Open, Low, High	146.8153
Open, Low, Close	165.8165

After testing each individual feature ('Open', 'High', 'Low', 'Close', 'Volume'), it was found that 'Open' and 'Low' resulted in lower MSE. Therefore, I attempted to combine these two features ('Open' and 'Low') to see if it could improve predictive ability. Indeed, this combination led to a reduction in MSE. I then explored adding a third feature to the combination to see if it further reduced MSE, but found that it actually increased MSE. Thus, the optimal input combination is 'Open' and 'Low'.

3. (15 points) Analyze the performance of the model with and without normalized inputs in Lab 4. You can use experimental results or external references (which must be cited) to support your conclusions on whether normalization improves the model's performance. (Approximately 100 words.)
   Many studies and practical applications have indicated that normalizing input features can often improve the performance of deep learning models. Normalization helps models converge faster, reduces issues like vanishing and exploding gradients, and enhances model generalization. Therefore, based on external references, experimental results in Lab 4 are likely to show that normalizing input features improves model performance, leading to reduced loss functions and increased accuracy.
   external references: https://machinelearningmastery.com/how-to-improve-neural-network-stability-and-modeling-performance-with-data-scaling/

4. (10 points) Why should the window size be less than the step size in Lab 4? Do you think this is correct? If you use external sources, please include references

to support your response. (Approximately 50 words.)
The window size should be smaller than the step size. This ensures that there is overlap between each window, allowing for better capture of features in the time series data. This approach helps improve the accuracy and robustness of the model because each window can utilize information from surrounding windows for a more comprehensive analysis and prediction.

5. (15 points) Describe one method for data augmentation specifically applicable to time-series data. Cite references to support your findings. (Approximately 100 words.)
   One data augmentation method applicable to time-series data is using Generative Adversarial Networks (GANs). GANs are deep learning models composed of a generator and a discriminator, capable of generating realistic synthetic data. In the context of time-series data, GANs can be trained to generate data similar to the original but with slight variations, such as adding noise or transforming the shape of the data, thereby augmenting the dataset. This approach helps increase data diversity and richness, enhancing the model's generalization ability.

   References:https://thesai.org/Publications/ViewPaper?Volume=15&Issue=1&Code=IJACSA&SerialNo=118

6. Discuss how to handle window size during inference in different model architectures (approximately 150 words):
   (i) (5 points) Convolution-based models
   In Convolutional Neural Networks (CNNs), the window size is typically determined by the size of the convolutional kernels and the stride. During inference, the appropriate window size can be chosen based on the specific task and the characteristics of the input data. Smaller window sizes may capture local features better, while larger window sizes can capture global features more effectively.
   (ii) (5 points) Recurrent-based models
   In Recurrent Neural Networks (RNNs) or Long Short-Term Memory networks (LSTMs), the window size is determined by the length of the sequence. During inference, the suitable window size can be selected based on the sequence length and the model's structure. Longer window sizes can capture longer-term dependencies, while shorter window sizes can lead to faster inference.
   (iii) (5 points) Transformer-based models
   In Transformer models, there is no explicit concept of window size; instead, the model uses self-attention mechanisms to handle different parts of the sequence. During inference, the number of layers and attention heads in the Transformer model can be adjusted based on task requirements and input sequence length to accommodate different window sizes. Transformers can effectively handle long-range dependencies, so window size handling during inference is usually not a significant concern.