National Tsing Hua University

11220IEEM 513600

Deep Learning and Industrial Applications

Homework 4

Name: Albert Student ID: 112030512

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.)

Window Size	Step	MSE
10	15	10.7319
20	10	11.2435
5	10	7.0636

When the window size is 10 and the step size is 15, the MSE is 10.7319. This suggests that using a smaller window size and a larger step size leads to a higher prediction error. This could be because there is less overlap between input sequences, and the model cannot fully capture continuous temporal dependencies.

For the second and third case we can find that increase the window size may not improve prediction performance, when window size = 5, it might perfectly capture the whole trend of stock price, since the stock price is from Monday to Friday (5 days).

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.

Window Size	Step	MSE
10	15	43.7362
20	10	43.0149
5	10	86.7494

After include 'Volume' as an additional input feature, the model’s performance seems to drop drastically, I think it might be 2 reasons below :

1. The relationship between changes in trading volume and stock price is not clearly correlated or consistent. We can see from the data that when the trading volume surges, the stock price doesn't change much, while other times, the stock price fluctuates significantly even when the trading volume remains relatively stable.
2. The scale and distribution of the volume data differ significantly from the price data. This can make it challenging for the model to effectively learn and utilize these two types of features with different scales.

(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.

Set the Window Size, Step to be 5, 10, since it has the best performance on question 1

input features	MSE
Open, High, Low, Close	7.0636
Open, Close	7.3763
Close, Volume	89.3980

Open, High, Low, Close: By incorporating these four features, the model can capture more comprehensive price dynamics.

Open, Close: By focusing on these two features, the model can capture the essential daily price movement without being influenced by intraday fluctuations. The difference between the open and close prices can provide insights into the overall market trend and investor sentiment for that day.

Close, Volume: Try to find the relationship between Close, Volume and target value, since the Volume may influence Close or target value.

For the result, the Open, High, Low, Close has the best performance, I think is because of its Comprehensiveness.

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.)

Set the Window Size, Step to be 5, 10, the sheet below is question 2’s setting after normalize

input features	MSE
Open, High, Low, Close	8.1035
Open, Close	7.9954
Close, Volume	87.6503

Analyze the impact of input normalization on the model's performance:

In time series prediction, normalizing the input data often improves the model's performance. Normalization scales features with different ranges to a similar range, making it easier for the model to learn and converge. Although the provided code does not explicitly perform input normalization, the model still achieves good results using nn.MSELoss() as the loss function. However, the result does not really suggest that there is significant improvement, but a slightly decrease on performance it may because of some loss of information, may need to do more experiments.

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.)

I think it might be correct although generally the window size should not be smaller than the step size. I think the purpose of this is to use smaller window size and larger step, so the model can focus more on local features and variations rather than relying heavily on long-term dependencies. This could be beneficial in short-term prediction tasks as we need in Lab4.

5. (15 points) Describe one method for data augmentation specifically applicable to time-series data. Cite references to support your findings. (Approximately100 words.)

One data augmentation method suitable for time series data is Time Series Generation.

This method generates new synthetic time series by transforming and combining the original time series. Specifically, the following transformations can be applied to the original series:

- Adding Gaussian noise
- Random time shifting
- Random amplitude scaling
- Random time cropping
- Random time reversal

By combining one or more of the above transformations, multiple new time series can be generated, expanding the training dataset. This approach can improve the model's generalization ability and mitigate overfitting.

6. Discuss how to handle window size during inference in different model architectures (approximately 150 words):

(i) (5 points) Convolution-based models

1. Convolutional models can directly input the entire time series for inference. The window size depends on the kernel size and receptive field, which should capture important patterns. During inference, the same window size as training can be used, or adjusted based on the desired output length.

(ii) (5 points) Recurrent-based models

2. In recurrent models, the window size corresponds to the input sequence length. During inference, the input sequence can be slid using the same window size as training, generating outputs step by step; or longer input sequences can be used by concatenating multiple windows for more context. Hidden states should be propagated between windows to maintain temporal dependencies.

(iii) (5 points) Transformer-based models

3. Transformer models typically divide the input sequence into fixed-length segments, each corresponding to a window, which can be processed in parallel. The window size can be chosen based on task requirements - longer windows for long-term dependencies, shorter windows for local features. Similar to recurrent models, position encoding should remain consistent between windows to maintain temporal order.

Reference：

1. Brownlee, J. (2022). Machine learning mastery. Machine Learning Mastery.
2. Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., & Muller, P. A. (2018). Data augmentation using synthetic data for time series classification with deep residual networks. *arXiv preprint arXiv:1808.02455*.