113034572_王奕鈞_HW4

1.

Window Size	Step Size	Best Validation MSE
20	5	0.000007
30	5	0.000008
40	10	0.000008

In this experiment, we tested three different combinations of window size and step size for training and validation. The results show that the validation losses (MSE) for all settings were between 0.000007 and 0.000008, indicating that the model effectively captured the trends in the price data. Although increasing the window size is expected to help the model capture longer-term dependencies, the similar performance across different settings suggests that the dataset is relatively stable, and the model is robust to variations in hyperparameter configurations.

2. (i)

Window Size	Step Size	Best Validation MSE
20	5	0.000008
30	5	0.000007
40	10	0.000009

After incorporating 'Volume' as an additional input feature alongside 'Close', the model achieved comparable or slightly better validation MSE across all settings. For instance, with a window size of 30 and step size of 5, the best validation MSE was reduced from 0.000008 (using only Close) to 0.000007 (using Close + Volume). This suggests that 'Volume' contains supplementary information about the market's trading activity, helping the model make more accurate predictions. However, the improvement was relatively minor, implying that the Close price alone already captures most of the essential trend signals. Overall, adding Volume enhanced the model's robustness slightly without introducing instability.

2. (ii)

To identify the best combination of input features, we experimented with four different feature sets: "Close Only", "Close + Volume", "Open+High+Low+Close", and "All Features". We used a fixed window size of 30 and step size of 5 for consistency across experiments. The results showed that the feature set “All Features” (which includes Open, High, Low, Close, and Volume) yielded the lowest validation MSE(0.000006), indicating the best performance.

This is reasonable since including all available features provides the model with richer contextual information about market dynamics. For example, High and Low values reflect volatility, while Volume might suggest investor activity levels. These extra signals help the model better understand the underlying patterns in price movements, leading to improved prediction accuracy.

3.

Feature Set	Normalized	Best Validation MSE
Close Only	True	0.000007
Close + Volume	True	0.000007
Open+High+Low+Close	True	0.000006
All Features	True	0.000006
Close Only	False	0.139164
Close + Volume	False	76.822819
Open+High+Low+Close	False	0.152397
All Features	False	76.247818

We compared model performance with and without normalization. When applying Min-Max normalization, the best validation MSEs ranged from 0.000006 to 0.000007 across different feature sets. Without normalization, the MSEs significantly worsened—up to 76.82 in some cases. This demonstrates that normalization helps stabilize training, prevents features like "Volume" from dominating due to scale, and leads to more accurate learning. These results confirm that normalization is crucial in time-series forecasting using deep learning, consistent with findings in related literature (Zhang et al., 2018).

4.

The window size should typically be greater than or equal to the step size, not less. A larger window provides more temporal context, and a smaller step allows more overlapping samples, improving model generalization. Therefore, I believe requiring window size < step size is incorrect in this context (Bontempi et al., 2013).

Reference：Bontempi, G., Taieb, S. B., & Le Borgne, Y. A. (2013). Machine learning strategies for time series forecasting. European Business Intelligence Summer School.

5.

One effective data augmentation method for time-series data is jittering, which involves adding small random noise to the original signal. This technique helps simulate sensor measurement variability and enhances the model's robustness to slight fluctuations. In financial time-series prediction, jittering can reduce overfitting and improve generalization by introducing variability without altering the underlying pattern. According to Um et al. (2017), jittering is especially effective in deep learning models for time-series classification. It is simple to implement and suitable for applications where the signal's shape is more important than precise values.

Reference：Um, T. T., Pfister, F. M., Pichler, D., Endo, S., Lang, M., Hirche, S., ... & Kulić, D. (2017, November). Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks. In Proceedings of the 19th ACM international conference on multimodal interaction (pp. 216-220).

6.

During inference, handling window size differs across model architectures:

(i) Convolution-based models require a fixed window size that matches the kernel and receptive field. The model expects inputs of a specific length and usually cannot adapt to varying window sizes without resizing or padding.

(ii) Recurrent-based models like LSTMs can handle variable-length windows due to their sequential processing. However, in practice, a fixed window size is often used for consistency and batch processing efficiency.

(iii) Transformer-based models also support variable input lengths, but are limited by memory usage due to self-attention's quadratic complexity. Positional encoding is used to preserve temporal information, and padding may be needed for batching.

In summary, convolutional models need strict window control, while recurrent and transformer models offer more flexibility but come with trade-offs in computational cost or structure.