

第一題：

	combination 1	combination 2	combination 3
window sizes	5	10	20
steps	10	15	30
Train loss	31.2149	148.9864	384.6655
Val loss	83.4818	141.9507	68.3722
Best Val loss	83.4818	141.9507	68.3722

結果分析：根據模型結果發現，當window sizes和steps越小，越容易得到越小的loss，造成此結果的可能原因，其中一項是當我們的window sizes和steps越小，代表模型每次學習的資料越小，如此模型更可以針對時間的細微變化做出學習上的調整，因此loss也會越小。

第二題：

(1)當加入Volume的資料作訓練和預測時(window sizes = 5, steps = 10), Train loss提高至1058.0814, Best Val loss提高至175.0916, 造成此現象的原因之一可能是Volume的資料相較於原本的features('Open', 'High', 'Low', 'Close')沒有比較好，反而可能會干擾模型的訓練以及預測。

(2)

	Train loss	Val loss	Best Val loss
'Open', 'High', 'Low'	24.1037	54.0438	54.0432
'High', 'Low', 'Close'	30.7511	2.2214	1.1344
'Open', 'High', 'Close'	31.6869	1.8819	0.9018
'Open', 'Low', 'Close'	30.9053	15.8898	15.8898
'Open', 'Close'	30.9617	25.4472	25.4467

最佳features組合是'Open', 'High', 'Close'，選擇'Open', 'Close'是因為這兩項分別代表時間的開始與結束，對於前一期和後一期的趨勢有明顯的關聯性，然而若只包含這兩項資料，資料數會太少，因此額外添加 'High'，結果顯示這個組合可以很好的作出股市的預測。

第三題：

normalization可以有效的幫助模型達到更好的學習效果，因為它可以將不同尺度的資料變成相同的尺度，如此可以平衡每一個項目對於模型的影響，有助於提高模型的訓練速度、穩定性和性能，並有助於減少過擬合的風險。

第四題：

window size應該要大於step才對，因為如果步長大於窗口大小，則可能會導致窗口之間存在較大的間隔，模型將無法充分利用時間序列中的信息，可能會錯過重要的模式或結構。

參考資料：

<https://bc165870081.medium.com/%E6%99%82%E9%96%93%E5%BA%8F%E5%88%97%E7%9A%84ai%E4%BB%8B%E7%B4%B9-ff250cfc2ff9>

第五題：

在這篇文獻中有提到其中一項解決方法，Augmentation in Time-Frequency Domain，這個方法是利用時間序列資料，轉換為frequency and time-frequency domains，這樣的技術可以在資料中引入不同的模式和變化，從而有可能提高在增強資料上訓練的模型的穩健性和泛化性。

參考資料：<https://arxiv.org/abs/2002.12478>

第六題：

(1)Convolution-based models：在CNN中window size 通常會設定在3*3的大小，也有1*1、5*5的大小，設定為奇數是因為有中心點，較容易對齊確認位置資訊，而且能確保Padding的對稱性。

參考資料：

<https://cinnamonaitaiwan.medium.com/%E6%B7%B1%E5%BA%A6%E5%AD%B8%E7%BF%92-cnn%E5%8E%9F%E7%90%86-keras%E5%AF%A6%E7%8F%BE-432fd9ea4935>

(2)Recurrent-based models：在RNN中，較大的window size可以提高模型效能，然而太大的window size可能會導致過度擬合，從而降低模型效能，實際上該如何決定大小，仍然須根據問題而決定。

參考資料：

<https://medium.com/mindboard/input-window-size-for-deep-recurrent-reinforcement-learning-4e5d2960756b>

(3)Transformer-based models：在Transformer模型中，如果一次輸入較長的window size，會導致在當所需資訊位於上下文中間時，效能顯著下降，運算量太大，因此window size的大小並不是越大越好。

參考資料：

<https://towardsdatascience.com/de-coded-understanding-context-windows-for-transformer-models-cd1baca6427e>