

AI 第二次作業

資管二 A 109403524 洪祥銘

TextGeneration

[Colab Link](#)

```
# 作業之一就是試試看其他本小說

book = ""
with open("./HP1.txt", "r", encoding="utf8") as file:
    for line in file:
        book += line

with open("./HP2.txt", "r", encoding="utf8") as file:
    for line in file:
        book += line

with open("./HP3.txt", "r", encoding="utf8") as file:
    for line in file:
        book += line

with open("./HP4.txt", "r", encoding="utf8") as file:
    for line in file:
        book += line

with open("./HP5.txt", "r", encoding="utf8") as file:
    for line in file:
        book += line

with open("./HP6.txt", "r", encoding="utf8") as file:
    for line in file:
        book += line

with open("./HP7.txt", "r", encoding="utf8") as file:
    for line in file:
        book += line

book_length = len(book)
unique_words = set(book)
print(f"哈利波特全系列共有 {book_length} 字詞")
print(f"包含了 {len(unique_words)} 個獨一無二的字 (含標點符號)\n")
print(book[0:500])

哈利波特全系列共有 2090906 字詞
包含了 4141 個獨一無二的字 (含標點符號)
```

第1章 大難不死的男孩
家住水堀街四號的德思禮夫婦總是得意地說他們是非常規矩的人家。拜託，拜託威農德思禮先生是一家名叫海格摩的公司做主管，公司生產鍋爐。他高犬魁梧，胖德思禮一家什麼都不缺，但他們擁有一個秘密，他們最害怕的就是這秘密會被人發
我們的故事開始於一個晦暗、陰

```
# 計算字數統計
words_count = {}
for w in book:
    if w in words_count:
        words_count[w] += 1
    else:
        words_count[w] = 1

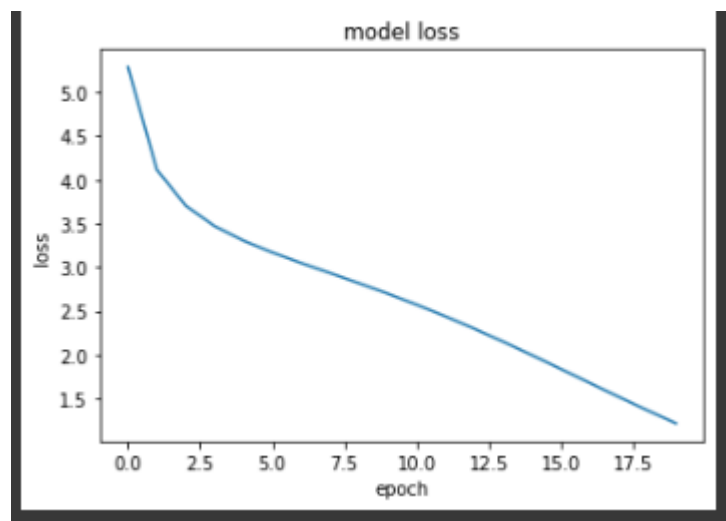
words_count = sorted(words_count.items(), key=lambda x: x[1])

[8] stop_word = 8
unique_words = [w_tup[0] for w_tup in words_count if w_tup[1] > stop_word]
print(f"去除次數小於 {stop_word} 的文字剩餘 : {len(unique_words)}")

去除次數小於8的文字剩餘 : 3040

[9] print(f"原本哈利波特全系列共有 {book_length} 字詞")
print(f"去除不常出現的文字後")
book = [w for w in book if w in unique_words]
print(f"剩餘 {len(book)} 個字")
```

原本哈利波特全系列共有 2090906 字詞
去除不常出現的文字後
剩餘2087281個字



身為一個哈利迷，去侵害哈利波特的著作權是很糟糕的，我很抱歉 XD
我選擇爬了哈利波特全系列把七集都丟進去 Train，結果如下、挺有趣的

```
init_seq = "葛林戴華德是同志嗎"
init_seq_ind = [word_2_index[w] for w in init_seq]
input = init_seq_ind[-seq_len:]

generateWords(input, 2000)
```

葛林戴華德是同志嗎？」
「當然，當然。」哈利說，想到了榮恩和妙麗，他自己和榮恩一起進了公共休息室。這裡所有的東西都由著他的頭，跟他哥哥的一樣蒼白。哈利不管在哪
「我們必須返回他們的注意，」鄧不利多說，「不然我為什麼要找格里戈維奇。」
「哦，是嗎？」阿不福思粗聲粗氣地問，「為什麼？」
「我認為是你，」哈利說，「但我必須教點不許她分靈體，並教你們的信任，你必須知道。」
「我？」
「我想，他大概認為，魔法部的巫師來說，這是一本書的關心。」
「你的魔杖在這兒，就在我們要面前，你就能——不著——」
「不，」哈利堅決地說，「你必須等著，我們不會回去，去尋找佛地魔的，並遺憾地告訴我們，你是惟一能把他無意為麻瓜的法律的！」
「鄧不利多！」哈利忍著笑向聲音發了一個燦爛的笑容。
「你覺得這是個好主意，」哈利說道，「是個更優秀的巫師，更優秀的男子漢兒了……不可否認為哈利的內心在禁忌森林裡發生了什麼事，接著問道：「
「我認為不會把你算我的，哈利，」鄧不利多說，「所以，他只是感到很糟糕，可現實不過是個傲慢的光，此刻正聚在那一幕。」
「哈利，我想你不會去找那個房子吧，」花兒一邊幫比爾把枕頭塞進了哈利手裡，「她是史萊哲林的。」
「他們會毀掉你，」榮恩一邊說，一邊順著梯子繼續給他們倆上樓，海格仰倒在他身邊。哈利把隱形斗篷緊緊裹在身上，看見馬份急切地、高爾平大的臉
「這不是偷，對嗎，露娜？」金妮問。
「我們知道他在說法堵住了他，」哈利說，「我們也想盡自己的一份，」她朝贊諾皺著眉頭說，「你以為我知道自己在幹什麼……我們以為鄧不利多教授
「我只是試著猜一下萬應室，」榮恩說，「一直在仔細聽我們。」
「什麼？」榮恩和妙麗一起問，聲音都啞了，「你媽媽要是那麼做的？」
「不，應該不是，先生。我是麻瓜出身。」
「赫—米—恩的記文，是嗎？」
「是的。」哈利說。
「不，不是，」金妮說著，聲音啞了，「會開完了。」
「好啊，但你待在這兒。」
「你沒有魔杖——？」衛斯理先生問。
「他們沒有聽見我的話嗎，波特？」念召喚咒的人大聲說，「怎麼樣，赫瑞司？」
「哈利，我想我們該回去了，」路平說，「可是，你必須毫無其事地做一些事情。」
「他已經在那裡了。」哈利說。
「他們說他的名字就會說什麼，」榮恩最後說道，「如果你的魔藥課O. W. L. E. W. Ts課程，我們上節課交通堵塞，一旦密切注視著。」

看到真的快笑死，連我的 model 都知道了><
這個文字生成的真滴讚、有趣，缺點就是 train 的時間長了點，我本來 optimizer 用 ranger、
epoch 設 20 而已，竟然要跑三個多小時，索性就改回去 adam 了。

Stock RNN

[Colab Link](#)

下二為 Model

```
input_shape = (seq_len, 1)
output_shape = [BATCH_SIZE, seq_len, 1]

keras.backend.clear_session()
model = tf.keras.Sequential([
    layers.LSTM(units = 64, input_shape=input_shape, activation="tanh", return_sequences=True),
    layers.Dropout(0.2),
    layers.LSTM(units = 64, input_shape=input_shape, activation="tanh", return_sequences=True),
    layers.Dropout(0.2),
    layers.LSTM(units = 64, input_shape=input_shape, activation="tanh", return_sequences=True),
    layers.Dropout(0.2),
    # layers.LSTM(units = 64, input_shape=input_shape, activation="tanh"),
    layers.LSTM(units = 64, input_shape=input_shape, activation="tanh", return_sequences=True),
    layers.Dropout(0.2),
    layers.TimeDistributed(layers.Dense(1)),
    # layers.Dropout(0.2),
    # layers.Flatten(),
    # layers.GlobalAveragePooling1D(),
    # layers.Dropout(0.2),
    # layers.Dense(1)
])

model.summary()
```

分享一下，activation function 的部分應該可以不用寫，據查到的資料說 Keras default LSTM 的 activaiton func 就是 tanh

但我覺得有待查證，我自己啃 doc 是沒注意到這件事情

```
Model: "sequential"
-----
Layer (type)                Output Shape              Param #
-----
lstm (LSTM)                  (None, 8, 64)            16896
dropout (Dropout)            (None, 8, 64)            0
lstm_1 (LSTM)                (None, 8, 64)            33024
dropout_1 (Dropout)          (None, 8, 64)            0
lstm_2 (LSTM)                (None, 8, 64)            33024
dropout_2 (Dropout)          (None, 8, 64)            0
lstm_3 (LSTM)                (None, 8, 64)            33024
dropout_3 (Dropout)          (None, 8, 64)            0
time_distributed (TimeDistr (None, 8, 1)            65
ibuted)
-----
Total params: 116,033
Trainable params: 116,033
Non-trainable params: 0
-----
```

我這次使用的 Optimizer 是跟 HW1 一樣大顯神威的 Ranger

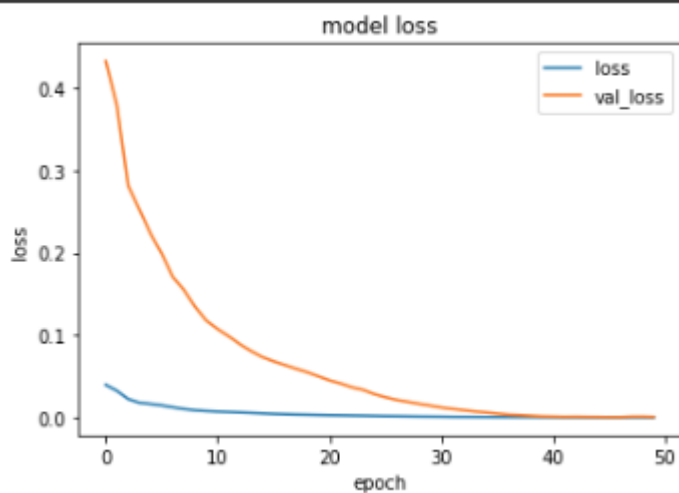
```
epochs = 50

!pip install -U tensorflow-addons
import tensorflow_addons as tfa
from keras import optimizers

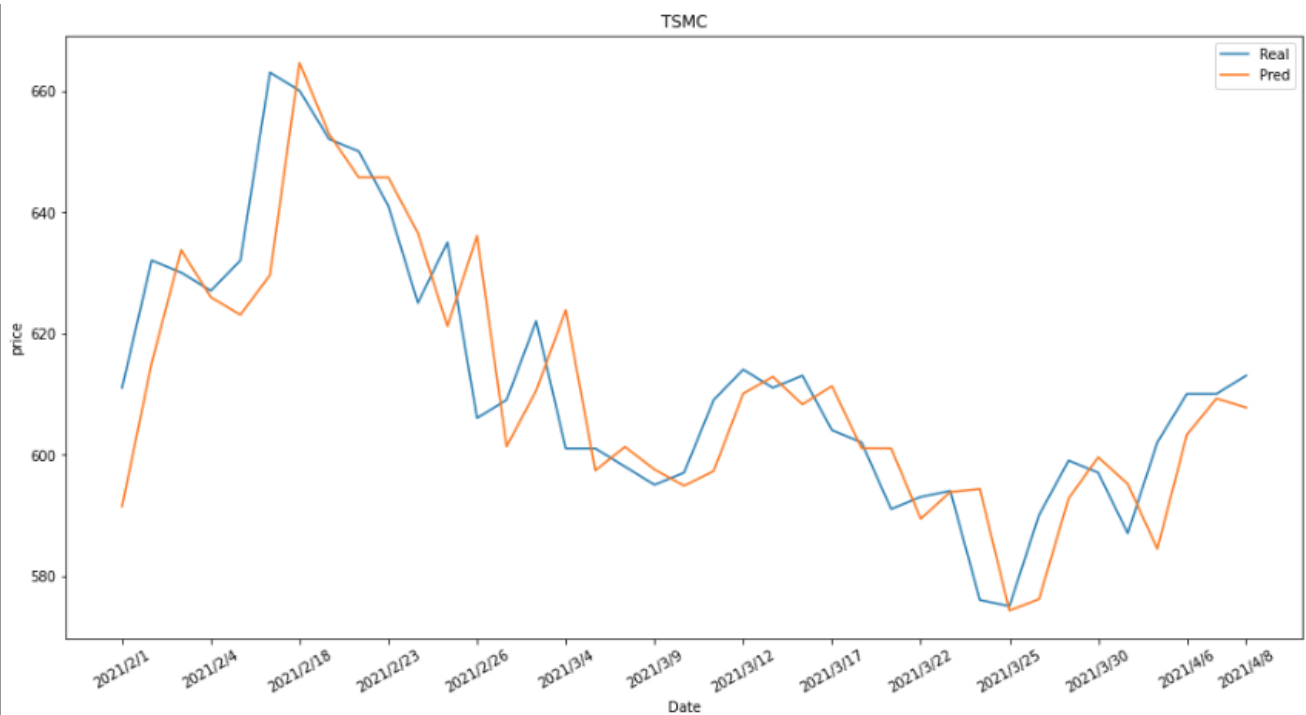
radam = tfa.optimizers.RectifiedAdam(0.001)
ranger = tfa.optimizers.Lookahead(radam, sync_period=6, slow_step_size=0.5)
model.compile(optimizer=ranger, loss='mean_squared_error')

history = model.fit(
    train_ds,
    epochs = epochs,
    validation_data=val_ds,
    shuffle=True
)
```

Model Loss



預測結果



```

# 預測 Test data
# 現在我們取出了選定公司最後seq_len天
# 我們模型透過讀取seq_len天後預測出seq_len+1天的價格

# 將日期保存在datas中 (list)
# 將真實價格保存在real_prices中 (list)
# 將預測價格保存在pred_prices中 (list) *注: 請填入經過正規化後的價格

# input是一個經過**正規化**的真實價格list, 請試著遍歷test_company來填滿上述三個list
# 可以參考TextGeneration的generateWords function, 概念一樣, 但是這次input請全部都填入"真實價格"
# 但是這次input請append "真實價格"
# 但是這次input請append "真實價格"
# 但是這次input請append "真實價格"
# 但是這次input請append "真實價格"
# 但是這次input請append "真實價格"

init_price = company.iloc[-seq_len,:]["price"].values
dates = []
real_prices = []
pred_prices = []

input = scalar.transform(init_price)

print(input)
for index,row in test_company.iterrows():
    # 我已經幫你把real_price和date放入list
    real_price = row["price"]
    real_prices.append(real_price)
    date = row["date"]
    dates.append(date)

    next_input = tf.expand_dims(input,axis=0)
    next_input = tf.expand_dims(next_input,axis=2)
    pred = model.predict(next_input)
    last_output =pred[0,-1]
    pred_prices.append(last_output)
    real_price = scalar.transform_one(real_price)
    input.append(real_price)
    del[input[0]]

```



Predict Function

心得 & StockRNN 的嘗試思路

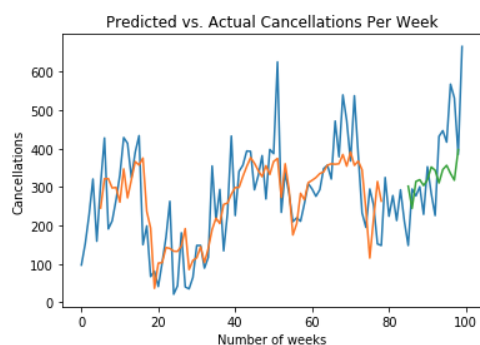
這次作業要寫 code 很少，大部分助教都弄好了，所以可以花很多心思在疊 model，但這次我實在做不出助教提供的預期結果的樣子（太鬼了，趨勢都一樣），在寫心得的過程中我成功了！難道我就是那做 AI 的人才（x）。

起初我是先試試看 HW1 管用的方法能不能在 HW2 再顯神蹟，結果是不行，原本的結果跟垃圾一樣。做了一下功課發現在這個 case，activation func 似乎不能用 ReLU，原因如下（單就圖形解釋，我覺得我 87% 講的是錯的）：

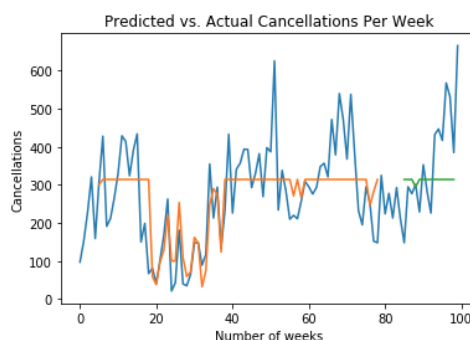
因為 ReLU 的特性會把值限制在大於 0，所以可以解決在 CNN 中使用 sigmoid 使小於 1 的導數連乘導致梯度消失，而相比之下 Tanh 會把值鎖在正負 1 之間，因此下方的比較圖（比較圖原先設定為是否有用 ReLU，但我有在文章讀到 Keras 預設 LSTM 的 activation func 是 Tanh，所以我把它當作 ReLU vs. Tanh）可以看出超過一定大小，那些值都會被河蟹掉變成平的，這對我們預測波動性的數值不好（accuracy 比較高沒錯、但圖形不是我們想要的）。網路上蠻多文章討論為何 CNN 用 ReLU、RNN 用 Tanh，分享一下我讀完的結論，首先是並非 RNN 不能用 ReLU，ReLU 本是要解決 RNN 層數太深導致梯度爆炸的問題而誕生的、所以有些文章根本在瞎說，我讀到比較有料的討論是說在 LSTM 層數不多（不夠深）的時候 Tanh 效果會比 ReLU 好。

Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Tanh		$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$

Predictions without ReLU



Predictions with ReLU



真的有夠哭，我在那邊試很久才想到會不會是 activation func 的問題，年輕人中就是年輕人啊。

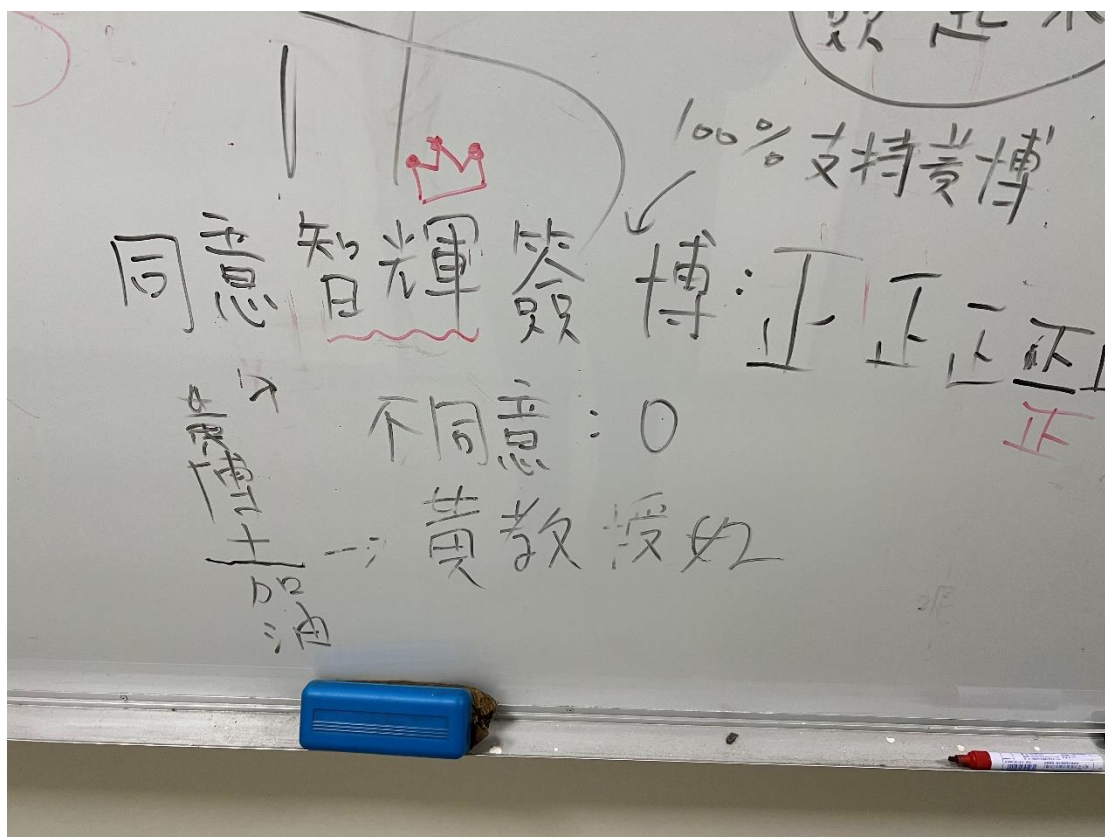
我在每一層 layer 都加上 Dropout 加速收斂，並且在最後加上了黑魔法 TimeDistributed（我努力了、我真的看不懂），我的 model 基本就完成了，看起來效果還不錯。另外在疊 model 的時候也有學到原來 LSTM 要轉 Dense 的時候不用做 Flatten，因為本身的 output_shape 就是 Dense 可以接受的格式所以就不用做平坦化了，另外還有發現一點是在轉 Dense 的時候，只轉一次的效果最好。也不能使用在 CNN 效果很好的 GlobalAveragePooling，它會使原先

設置的 `seq_len` 長度直接被砍掉變成 1 (如下圖)。

```
global_average_pooling1d (G (None, 1) 0  
lobalAveragePooling1D)
```

這次一樣啃了一堆文件，相較於 HW1 我這次在 `activation func` 撞牆最久、偏偏中文資源又少 (剛好強迫去啃 `document`)，剛好來練英文 XD。意外發現上面那堆關於 `ReLU`、`Tanh` 的爭論，自己的心得是看文件或教學真的要多看幾篇、有時候以為很有道理的東西其實也是某個人瞎姬芭說的害死人，以後有機會做研究真的要很小心。

最後的最後，在這邊我要大聲說「支持智輝簽博」，這麼讚ㄉ人才一定要留下來造福學弟妹 \ 智輝 /\ 智輝 /\ 智輝 /



同意阿，哪次不同意