利用NLP分析句子情感

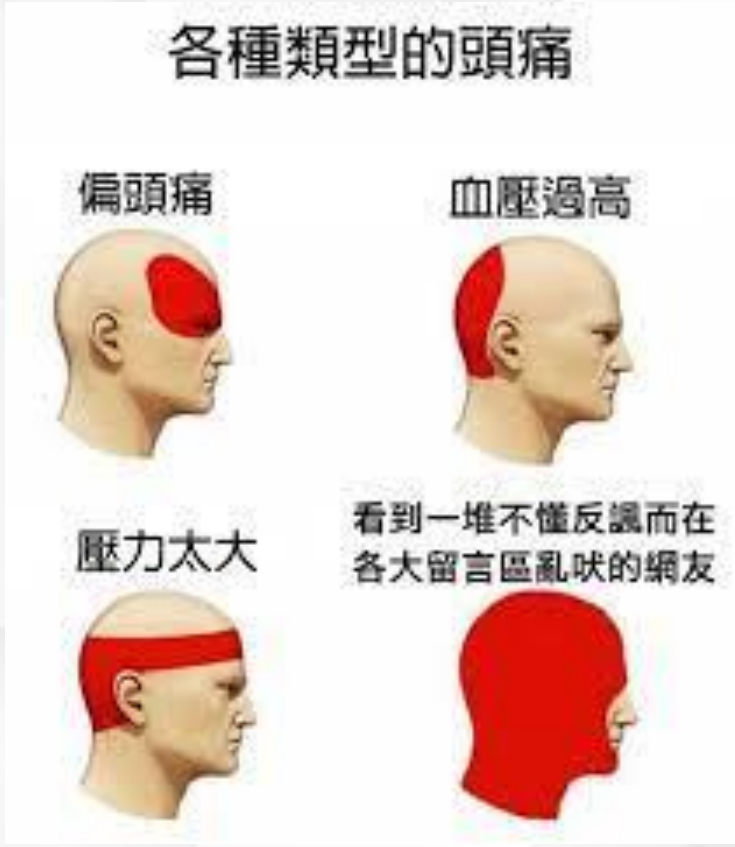
陳彥碩 110034568

目錄

01 背景介紹

01 背景介紹

- 分析某商品或股票的評論時，往往不確定留言到底是正面還是反面的意思。
- 在看留言時，往往不確定留言表達的意思。



01 5W1H

Who 想了解事情並上網查看評論的人

When 任何時刻

What 正確判別文字所要表達的意思

Where 任何地點

Why 避免誤解文字意思，造成錯誤的決策

How 利用RNN和CNN

02 資料前處理

02 資料蒐集

- 資料來自 Sentiment140 dataset，共有160萬 tweets
- 資料包含了
  - target (negative = 0, positive = 4)
  - user id
  - date
  - user
  - text

1	target	ids	date	user	text
2	0	1467810369	Mon Apr 06 22:19:45	_TheSpecialOne_	@switchfoot http://twit
3	0	1467810672	Mon Apr 06 22:19:49	scotthamilton	is upset that he can't up
4	0	1467810917	Mon Apr 06 22:19:53	mattycus	@Kenichan I dived ma
5	0	1467811184	Mon Apr 06 22:19:57	ElleCTF	my whole body feels it
6	0	1467811193	Mon Apr 06 22:19:57	Karoli	@nationwideclass no,
7	0	1467811372	Mon Apr 06 22:20:00	joy_wolf	@Kwesidei not the wh
8	0	1467811592	Mon Apr 06 22:20:03	mybirch	Need a hug
9	0	1467811594	Mon Apr 06 22:20:03	coZZ	@LOLTrish hey long
10	0	1467811795	Mon Apr 06 22:20:05	2Hood4Hollywood	@Tatiana_K nope they
11	0	1467812025	Mon Apr 06 22:20:09	mimismo	@twittera que me mue
12	0	1467812416	Mon Apr 06 22:20:16	erinx3leannexo	spring break in plain c
13	0	1467812579	Mon Apr 06 22:20:17	pardonlauren	I just re-pierced my ea
14	0	1467812723	Mon Apr 06 22:20:19	TLeC	@caregiving I couldn't
15	0	1467812771	Mon Apr 06 22:20:19	robrobbierobert	@octolinz16 It it coun
16	0	1467812784	Mon Apr 06 22:20:20	bayofwolves	@smarrison i would've
17	0	1467812799	Mon Apr 06 22:20:20	HairByJess	@iamjazzyfizzle I wish
18	0	1467812964	Mon Apr 06 22:20:22	lovesongwriter	Hollis' death scene will
19	0	1467813137	Mon Apr 06 22:20:25	armotley	about to file taxes
20	0	1467813579	Mon Apr 06 22:20:31	starkissed	@LettyA ahh ive alwa

02 資料清理

- 清除不必要的資料，只保留 target(改為 sentiment) 和 text
- 將 sentiment改為 negative 和 positive

```
df = pd.read_csv('C:/Users/Henry-Lab/Desktop/IIE_project 3/training.1600000.processed.noemoticon.csv',
                 encoding = 'latin', header=None)
# print(df.head())

df.columns = ['sentiment', 'id', 'date', 'user_id', 'text']
df = df.drop(['id', 'date', 'user_id'], axis=1)

lab_to_sentiment = {0:"Negative", 4:"Positive"}
def label_decoder(label):
    return lab_to_sentiment[label]
df.sentiment = df.sentiment.apply(lambda x: label_decoder(x))
#print(df.head())

df.to_csv("data1.csv", encoding = "utf-8")
```

		sentiment	text
2	0	Negative	@switchfoot http://twit
3	1	Negative	is upset that he can't up
4	2	Negative	@Kenichan I dived ma
5	3	Negative	my whole body feels it
6	4	Negative	@nationwideclass no,
7	5	Negative	@Kwesidei not the wh
8	6	Negative	Need a hug
9	7	Negative	@LOLTrish hey long
10	8	Negative	@Tatiana_K nope they
11	9	Negative	@twittera que me mue
12	10	Negative	spring break in plain c
13	11	Negative	I just re-pierced my ea
14	12	Negative	@caregiving I couldn't
15	13	Negative	@octolinz16 It it coun
16	14	Negative	@smarrison i would've
17	15	Negative	@iamjazzyfizzle I wish
18	16	Negative	Hollis' death scene will
19	17	Negative	about to file taxes
20	18	Negative	@LettyA ahh ive alwa

02 資料清理

- 清除停用詞(stopwords)
- 提取詞幹
- 清除標記和連結

```
from nltk.corpus import stopwords
from nltk.stem import SnowballStemmer

stop_words = stopwords.words('english')
stemmer = SnowballStemmer('english')
text_cleaning_re = "@\S+|https?:\S+|http?:\S|[^A-Za-z0-9]+"

def preprocess(text, stem=False):
    text = re.sub(text_cleaning_re, ' ', str(text).lower()).strip()
    tokens = []
    for token in text.split():
        if token not in stop_words:
            if stem:
                tokens.append(stemmer.stem(token))
            else:
                tokens.append(token)
    return " ".join(tokens)

df.text = df.text.apply(lambda x: preprocess(x))
```

- plays
- played
- playing

} play

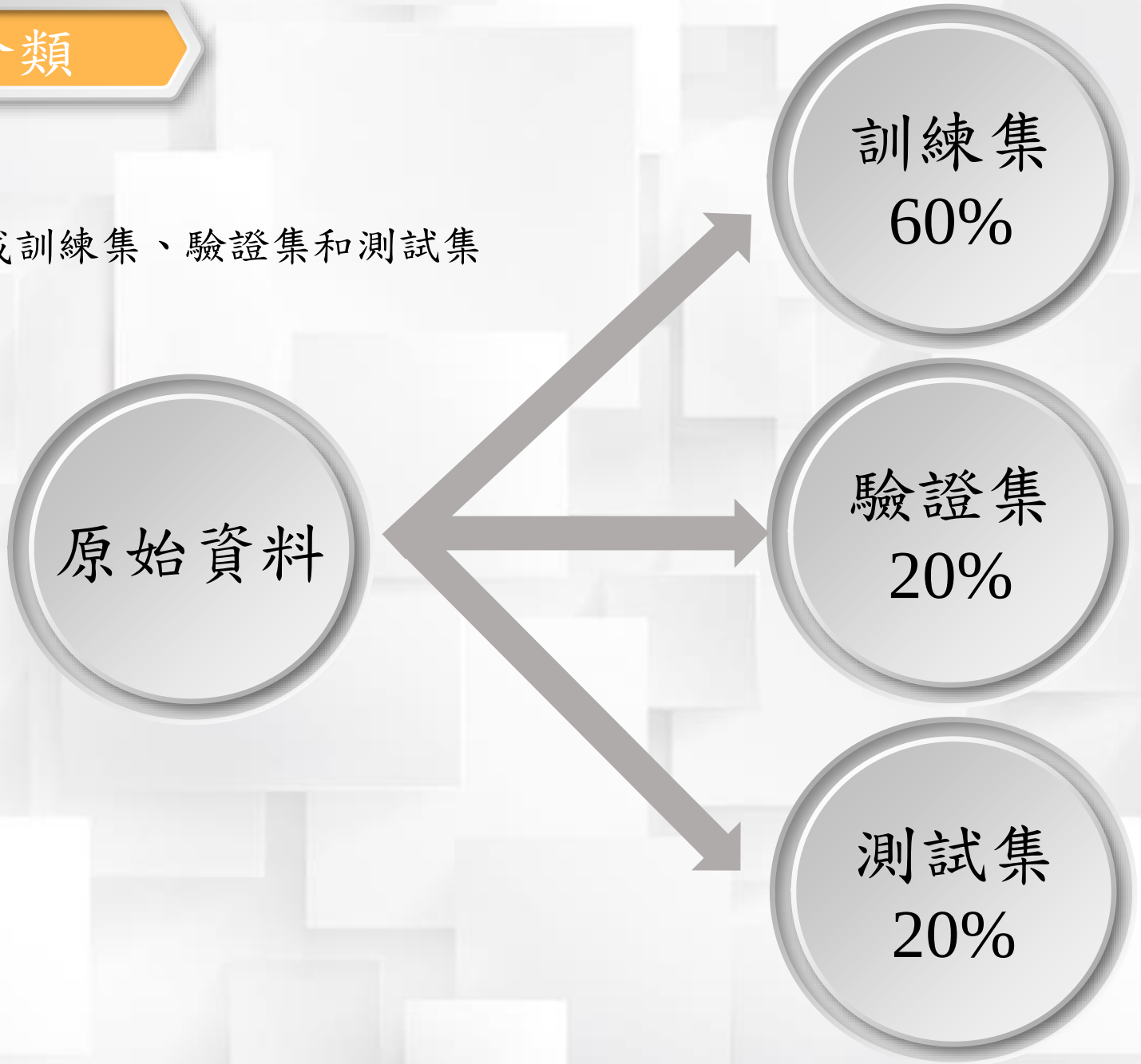
@tea oh! i'm so sorry i didn't think about that before retweeting.

↓

oh sorry think retweet

02 資料分類

- 將資料區分成訓練集、驗證集和測試集



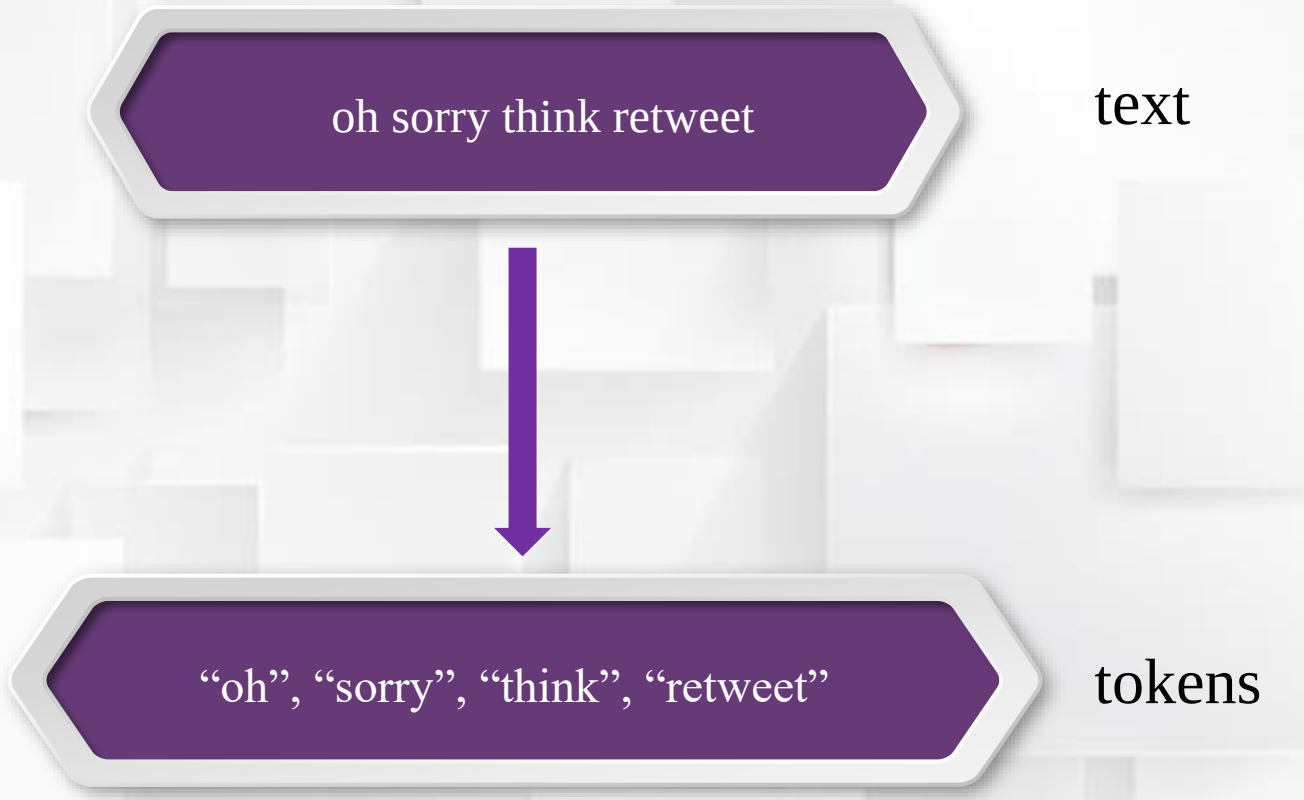
02 資料前處理

- 將每筆資料轉換成 tokens

```
from keras.preprocessing.text import Tokenizer

tokenizer = Tokenizer()
tokenizer.fit_on_texts(train_data.text)

word_index = tokenizer.word_index
vocab_size = len(tokenizer.word_index) + 1
print("Vocabulary Size :", vocab_size)
```



Vocabulary Size = 290575

02 資料前處理

- Vocabulary Size = 290575 ➡ 維度太高
- 利用文字嵌入法降低維度 ➡ 利用史丹佛 GloVe團隊開發的資料庫

```
GLOVE_EMB = 'C:/Users/88697/Desktop/project 3/glove.6B.300d.txt'

embeddings_index = {}

f = open(GLOVE_EMB, encoding = "utf-8")
for line in f:
    values = line.split()
    word = value = values[0]
    coefs = np.asarray(values[1:], dtype='float32')
    embeddings_index[word] = coefs
f.close()

print('Found %s word vectors.' %len(embeddings_index))
```

Found 400000 word vectors.

03 模型訓練與績效

03 模型架構

- 循環丟棄
- 堆疊循環層
- 雙向循環層

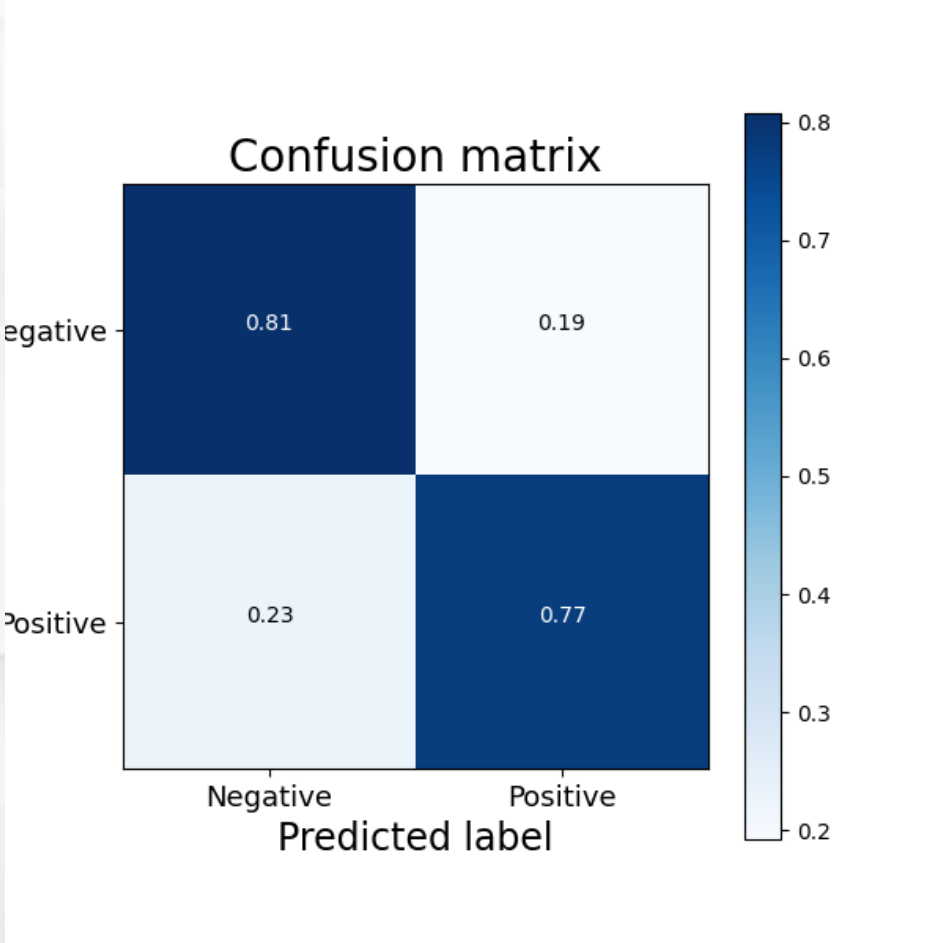
```
x = Bidirectional(LSTM(64,
                       dropout=0.2,
                       recurrent_dropout=0.2,
                       return_sequences=True))(x)
```

- 遞減學習率

```
ReduceLROnPlateau = ReduceLROnPlateau(factor=0.1,
                                      min_lr = 0.0001,
                                      patience = 3,
                                      monitor = 'val_loss',
                                      verbose = 1)
```

03 循環神經網路

- RNN、LSTM、GRU



	Accuracy	Precision	Recall	F1-score
Simple RNN	0.765	0.752	0.790	0.771
LSTM	0.790	0.802	0.770	0.786
GRU	0.785	0.788	0.780	0.784

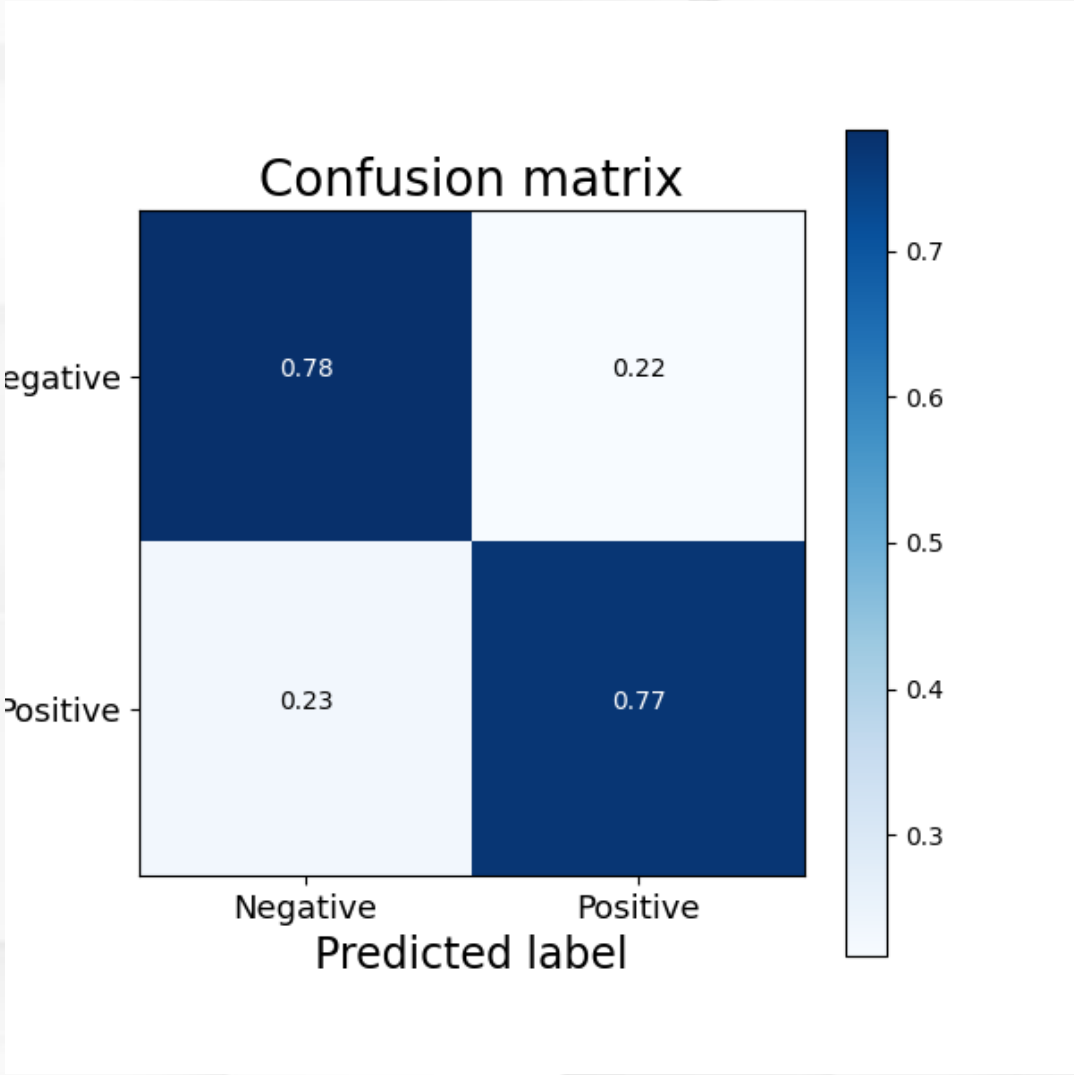
```
int32')


2, return_sequences = True))(x)
```

03 卷積神經網路

- 使用1D卷積神經網路來處理
- 與RNN、LSTM、GRU相比，1D卷積網路雖然驗證準確率較低，但可擁有較快的運算時間。

```
sequence_input = Input(shape=(MAX_SEQUENCE_LENGTH,), dtype='int32')
embedding_sequences = embedding_layer(sequence_input)
x = SpatialDropout1D(0.2)(embedding_sequences)
x = Conv1D(64, 5, activation='relu')(x)
x = MaxPool1D()(x)
x = Conv1D(64, 5, activation = 'relu')(x)
x = GlobalMaxPool1D()(x)
outputs = Dense(1, activation='sigmoid')(x)
model = tf.keras.Model(sequence_input, outputs)
```



03 RNN + CNN

- 結合RNN與CNN進行訓練
- 利用1D卷積神經網路萃取資料特徵，再送給RNN進行訓練以減少運算時間。

	Accuracy	Precision	Recall	F1-score
Simple RNN	0.700	0.679	0.760	0.717
LSTM	0.695	0.676	0.750	0.711
GRU	0.695	0.673	0.760	0.714

```
sequence_input = Input(shape=(MAX_SEQUENCE_LENGTH,), dtype='int32')
embedding_sequences = embedding_layer(sequence_input)
x = SpatialDropout1D(0.2)(embedding_sequences)
x = Conv1D(64, 5, activation='relu')(x)
x = MaxPool1D()(x)
x = Conv1D(64, 5, activation = 'relu')(x)
x = MaxPool1D()(x)
x = Bidirectional(LSTM(64, dropout=0.2, recurrent_dropout=0.2, return_sequences=True))(x)
x = Dense(512, activation='relu')(x)
x = Dropout(0.5)(x)
x = Dense(512, activation='relu')(x)
x = Dense(256, activation = 'relu')(x)
outputs = Dense(1, activation='sigmoid')(x)
model = tf.keras.Model(sequence_input, outputs)
```

03 績效衡量

	Accuracy	Precision	Recall	F1-score
Simple RNN	0.765	0.752	0.790	0.771
LSTM	0.790	0.802	0.770	0.786
GRU	0.785	0.788	0.780	0.784
1D CNN	0.775	0.780	0.770	0.775
Simple RNN	0.700	0.679	0.760	0.717
LSTM	0.695	0.676	0.750	0.711
GRU	0.695	0.673	0.760	0.714

結合CNN

04 結論與未來展望

04 結論與未來展望

- 結論
  - 現有績效可達到79%
  - 結合CNN與RNN績效沒有比單純RNN好
- 未來展望
  - 可以多嘗試不同參數提升準確率
  - 應用於股票分析或商品評論

Thank you for listening