文字探勘初論 期末報告

用澳洲新聞頭條分析經濟指標動向

熊才誠
化工五

彭盛皓
數學四

陳岳緯
財金三

鍾秉瀚
財金三

摘要

這次研究的目標主要是利用課程所學，測試以文字探勘基礎的深度學習模型是否能夠捕捉真實事件為區域性經濟體帶來的增益及虧損。

1 研究目標

我們選定澳洲的 ABC News 的新聞標題為資料集，時間期為 2003 年至 2020 年，一共 17 年，以及澳洲 30 年公債殖利率的 MoM 變動作為經濟指標，期望以深度學習模型觀察出兩者之間的關聯性，以及達到預測的效果。

2 研究方法

2.1 資料選擇

1. 新聞具有社會代表性，新聞代表了當時的社會狀態和社會事件。
2. 新聞標題為了達到摘要的目的，標題通常是對於內文的概括陳述，因此我們選擇使用新聞標題作為我們的資料集。
3. 從 Kaggle Data 下載資料集：
   (A Million News Headlines - News headlines published over a period of 17 Years)

Figure 1: A Million News Headlines

2.2 Vectorizers

1. Sklearn Tfidf (CountVectorizer)：

優點：

能過濾掉一些常見的卻無關緊要的詞語，同時保留影響整個文本的重要字詞。

缺點：

i. 不能反應詞的位置。
ii. TFIDF 並不能用來說明特徵詞的重要與否，只是用來區分不同文檔。
iii. 只有在監督式學習表現較好。

2. Word2vec (gensim)：

優點：

i. 考慮詞與詞之間的關係。
ii. Embedding 方法維度更少，所以速度更快。
iii. 通用性很強，可以用在各種 NLP 任務中。

缺點：

i. 詞和向量是一對一的關係，所以多義詞的問題無法解決。
ii. Word2vec 是一種靜態的方式，雖然通用性強，但是無法針對特定任務做動態優化。

3. BERT (keras_bert, pyTorch)：

優點：

i. 適用無監督學習。

ii. 考慮上下文，因此不容易出現單詞的歧義問題。

缺點：

i. 需要 pre-trained model。
ii. 可能會需要做大量 parameters tuning。

2.3 NN Model

1. CNN

主要針對 DNN 存在的參數數量膨脹問題，對於 CNN，並不是所有的上下層神經元都能直接相連，而是通過“卷積核”作為中介。同一個卷積核在多有圖像內是共享的，圖像通過卷積操作仍能保留原先的位置關係。CNN 之所以適合圖像識別，正因為 CNN 模型限制參數個數並挖掘局部結構的這個特點。

優點：

i. 權重共享策略減少了需要訓練的引數，相同的權重可以讓濾波器不受訊號位置的影響來檢測訊號的特性，使得訓練出來的模型的泛化能力更強
ii. 池化運算可以降低網路的空間解析度，從而消除訊號的微小偏移和扭曲，從而對輸入資料的平移不變性要求不高。

缺點：

i. 深度模型容易出現梯度消散問題。

2. RNN

針對 CNN 中無法對時間序列上的變化進行建模的侷限，為了適應對時序數據的處理，出現了 RNN。在普通的全連接網絡或者 CNN 中，每層神經元的信號只能向上一層傳播，樣本的處理在各個時刻獨立（這種就是前饋神經網絡）。而在 RNN 中，神經元的輸出可以在下一個時間戳直接作用到自身。(t+1) 時刻網絡的最終結果 O(t+1)是該時刻輸入和所有歷史共同作用的結果，這就達到了對時間序列建模的目的。存在的問題：RNN 可以看成一個在時間上傳遞的神經網絡，它的深度是時間的長度，而梯度消失的現象出現時間軸上。

優點：

i. 模型是時間維度上的深度模型，可以對序列內容建模。

缺點：

i. 需要訓練的引數較多，容易出現梯度消散或梯度爆炸問題；
ii. 不具有特徵學習能力。

3. Linear DNN

神經網絡是基於感知機的擴展，而 DNN 可以理解為有很多隱藏層的神經網絡。多層神經網絡和深度神經網絡 DNN 其實也是同一個東西，DNN 有時也叫做多層感知機（Multi-Layer perceptron, MLP）。

優點：

i. 生成模型學習聯合概率密度分佈，所以就可以從統計的角度表示資料的分佈情況，能夠反映同類資料本身的相似度；
ii. 生成模型可以還原出條件概率分佈，此時相當於判別模型，而判別模型無法得到聯合分佈，所以不能當成生成模型使用。

缺點：

i. 生成模型不關心不同類別之間的最優分類面到底在哪，所以用於分類問題時，分類精度可能沒有判別模型高；
ii. 由於生成模型學習的是資料的聯合分佈，因此在某種程度上學習問題的複雜性更高。
iii. 要求輸入資料具有平移不變性。

這次選用的操作環境以及模型搭建語法為 Pytorch，並且使用到線性的深度學習網路作為學習的基礎架構。不使用其他常見神經網路類型如：卷積神經網路、循環神經網路的原因為：這次研究中，轉換好的新聞標題特徵值為一維空間的特徵向量，故不適用善於處理平面 2D 資料的卷積神經網路。此外，由於這次選用的目標值是領先指標，所以對於資料和目標值的 lagging 基本可以忽略不計，此外我們這次想要實現的目標只是測試以文字探勘基礎的深度學習模型是否能夠捕捉真實事件為區域性經濟體帶來的增益及虧損。

3 研究過程與發現

3.1 Vectorizer 選擇

根據我們資料集的類型和性質，本身是監督式的學習，對於不同的時間有不同的新聞和配合的經濟指標，最後的選擇是 Sklearn Tfidf。

根據不同的參數，提供三種 TFIDF：

1. `min_df = 100, ngram_range = (1,3)`

   最小出現詞頻定在必須大於 100 次，允許 tri-gram 和 bi-gram。

2. `min_df = 100`

   最小出現詞頻定在必須大於 100 次。

3. `default`

min_df 是為了去除一些出現頻率很少的字，對於文章意義不大，通常是一些專有名詞，對於新聞影響不大，限定 document frequency 以降低字典大小，降低維度。Tri-gram, bi-gram 是為了讓一些專有名詞或新聞中常連用的字允許出現在字典裡，例如: child education, abc news, political scandal，因為出現次數還是要大於 100 次所以不會有一些只出現一次的特殊用法。

三種 TFIDF 字典大小分別是：9736, 7551, 97534。

```
term_names_tfidf_tri_min = TFIDF_vectorizer_tri_min.get_feature_names()
term_names_tfidf_min = TFIDF_vectorizer_min.get_feature_names()
term_names_tfidf = TFIDF_vectorizer.get_feature_names()
```

```
print(len(term_names_tfidf_tri_min))
print(len(term_names_tfidf_min))
print(len(term_names_tfidf))
```

```
9736
7551
97534
```

Figure 2: TFIDF 字典大小

最後再把時間資料和 TFIDF 合併在一起，供下游任務使用。

```
data_tfidf_tri_min = []
for i in range (0,1186018):
    data_tfidf_tri_min.append(np.hstack([date[i], TFIDF_vectors_tri_min[i]]))
```

```
data_tfidf_tri_min = np.reshape(data_tfidf_tri_min,(2, 1186018))
```

```
data_tfidf_tri_min
```

```
array([['20030219',
        <1x9736 sparse matrix of type '<class 'numpy.float64'>'
        with 3 stored elements in Compressed Sparse Row format>,
        '20030219', ...,
        <1x9736 sparse matrix of type '<class 'numpy.float64'>'
        with 5 stored elements in Compressed Sparse Row format>,
        '20110208',
        <1x9736 sparse matrix of type '<class 'numpy.float64'>'
        with 3 stored elements in Compressed Sparse Row format>],
       ['20110208',
        <1x9736 sparse matrix of type '<class 'numpy.float64'>'
        with 4 stored elements in Compressed Sparse Row format>,
        '20110208', ...,
        <1x9736 sparse matrix of type '<class 'numpy.float64'>'
        with 1 stored elements in Compressed Sparse Row format>,
        '20191231',
        <1x9736 sparse matrix of type '<class 'numpy.float64'>'
        with 8 stored elements in Compressed Sparse Row format>]], dtype=object)
```

Figure 3: 合併日期的資料集

3.2 經濟指標選擇

「債券」的長期利率不是由中央銀行決定，大致上由市場決定（不考慮 QE 購債的間接影響）。美國 10 年期公債殖利率普遍被市場認為是零風險利率。因此可以認為債券對於市場反應比較靈敏。

我們這次的資料集主要是集中在澳洲的新聞，因此選用澳洲 30 年公債殖利率的 MoM 變動作為經濟指標，若是景氣正向成長，公債利率通常會上升，反之則會下降。如下圖可見：2019 年底與 2020 年初皆呈現劇烈下降。

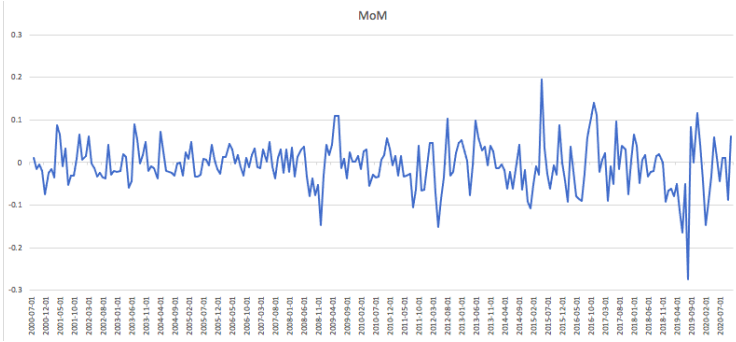

Figure 4: 澳洲 30 年公債殖利率的 MoM 變動

3.3 Feature Selection

目的是在開發預測模型時減少輸入變量的數量，提升效率。

1. Stop Words：

   在造 TFIDF 的時候已經處理。

2. Numbers：

   去除非普遍化變因。

3.4 Linear DNN model 操作

1. 架構

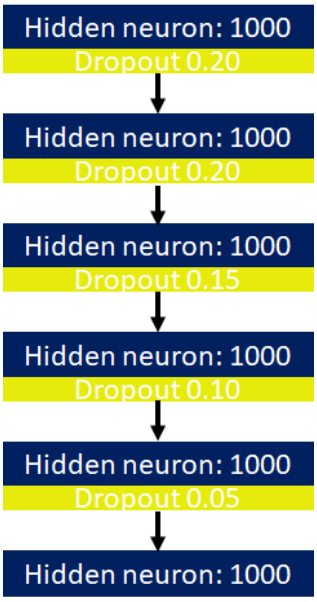

Figure 5: Linear DNN model 架構

其中可以看到我們隨著網路深度不斷推續，使用到的遺忘神經層比重越來越輕，這個設計是為了讓深度神經網路在處理文字相關特徵值的稀疏矩陣所特別設計，可以讓網路更好的了解哪些字詞的計算是對目標值預測沒有幫助的，而到尾端的部分，我們反而放寬遺忘神經層的比重，希望神經網路能夠好好把保存下來的重要特徵和目標經濟指標進行對應。

```
class dnn(torch.nn.Module):
    def __init__(self, n_feature, n_hidden, n_output):
        super(dnn, self).__init__()
        self.linear = nn.Sequential(
            nn.Dropout(0.2),
            nn.Linear(n_feature, n_hidden),
            nn.ReLU(),
            nn.Dropout(0.2),
            nn.Linear(n_hidden, n_hidden),
            nn.ReLU(),
            nn.Dropout(0.15),
            nn.Linear(n_hidden, n_hidden),
            nn.ReLU(),
            nn.Dropout(0.10),
            nn.Linear(n_hidden, n_hidden),
            nn.ReLU(),
            nn.Dropout(0.05),
            nn.Linear(n_hidden, n_hidden),
            nn.ReLU(),
            nn.Linear(n_hidden, n_output))

    def forward(self, x):
        out = self.linear(x)
        return out
```

Figure 6: DNN 實際操作

2. 學習曲線

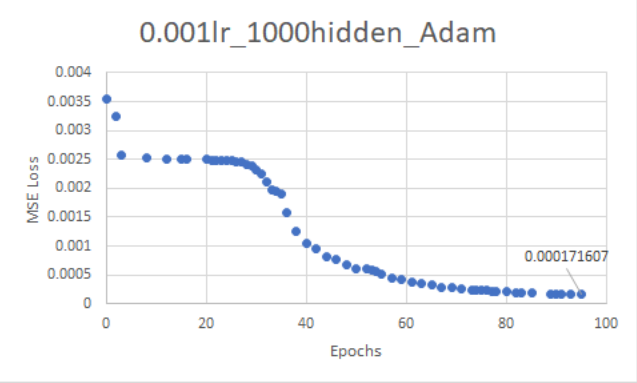

Figure 7: 學習曲線

圖中為使用 Adam 梯度下降演算法、學習率 =0.001 的訓練過程。可以發現訓練過程非常成功，基本上在 80 epoch 的時候就收斂的很好了，並且達到 1e-4 的損失表現。值得一提的是這樣的訓練表現在套用進 2019/2020 年的數據進行測試後誤差更是只有 5e-3 的表現，而 2019/2020 年的公債殖利率變化基本都在正負 0.1 之間震盪，意即我們的模型對於一段時間內發生的新聞事件為區

域性經濟體的領先指標影響可以捕捉將近 95%的預測。可以說是非常成功。

結論與展望

1. 加強 Feature Selection，求出更精準的訓練資料集。

對於這次選用的 TFIDF，我們認為還是保留太多太長的特徵訊息(27219 個特徵)，相信有更多不同的特徵擷取技巧可以套用，一方面減輕神經網路學習的困難一方面減少硬體計算成本及時間成本。

2. 嘗試不同國家之經濟指標以及訓練集，查看適配性。

這點經過老師在期末 demo 中有特別提到對於不同的經濟指標其實可能有不同的適用神經網路模型，這點的問題雖然在這次研究範圍內有刻意避開，但在往後類似的研究操作上想必是個必然會遇到的問題。若是處理到時間相關的資料特徵時，可能會需要以 lagging feature 相關的技術進行 data augmentation，或者先操作過 ARIMA related model 再進行深度學習。

3. 找尋除了經濟指標外能反映國家發展的指標做新模型。

成員參與

熊才誠(30%)：建立深度學習模型、提供想法、上台報告、參與討論。

彭盛皓(30%)：資料處理、處理文字前處理、書面報告製作、參與討論。

鍾秉瀚(20%)：上台報告、簡報製作、參與討論。

陳岳緯(20%)：經濟指標探索、參與討論。

參考資料

[1] Rohit Kulkarni (2020). A Million News Headlines - News headlines published over a period of 17 Years. https://www.kaggle.com/therohk/million-headlines/tasks?taskId=2747.
[2] Scikit-learn, TfidfVectorizer.
[3] Pytorch, Linear DNN Model.
[4] 財經 M 平方，澳洲-30 年期公債殖利率。https://www.macromicro.me/charts/14573/australia-30-year-government-bond-yield
[5] Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2020). Deep learning based text classification: A comprehensive review. arXiv preprint arXiv:2004.03705.