深度學習模型的比較研究 – 以 MNIST 為例

開放原始碼論文

陳鍾誠*
國立金門大學資訊工程學系
ccc@nqu.edu.tw

2021/10/15

摘要

本論文的開放原始碼專案網址為：https://github.com/cccresearch/nnModelCompare

不同的神經網路模型，經訓練之後的正確率可能差異很大。本文針對手寫數字辨識的 MNIST 資料庫進行測試，以便觀察模型的表現，並分析其背後的原因。

關鍵字 神經網路 · 深度學習 · MNIST

1 簡介

近幾年深度學習技術讓人工智慧領域有了很大的進展，也吸引到了學術界與產業界共同投入研究，相繼開發出更好，但也相對更複雜的模型。

為何有些模型表現好，有些模型表現差，各個網路層的效用是甚麼，為何需要加入某些層，若拿掉的話會有甚麼不良反應嗎？這就是本研究所想要探討的問題！

2 背景

手寫數字辨識的 MNIST 是影像辨識領域中最常被拿來測試的資料集，而 CNN 卷積神經網路架構的 LeNet 則是 Yann Le Cun 1989 年在研究手寫辨識問題時，提出來的辨識模型，實驗發現 LeNet 在手寫辨識上有相當高的正確率。

不過，其他的模型，像是使用多層感知器，也可以達到 90% 以上的正確率，

[1, 2] and see [3].

The documentation for `natbib` may be found at

`http://mirrors.ctan.org/macros/latex/contrib/natbib/natnotes.pdf`

Of note is the command `\citet`, which produces citations appropriate for use in inline text. For example,

```
\citet{hasselmo} investigated\dots
```

produces

> Hasselmo, et al. (1995) investigated…

`https://www.ctan.org/pkg/booktabs`

*Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

3 方法

簡易的『爬山演算法』如下圖所示 1 所示。

```
Algorithm Hill-Climbing(pi)
 p = pi // 設定粒子 p 為起始粒子 pi
 while not isEnd()
 pn = p.neighbor(step) //選擇粒子 p 的鄰居 pn
 if pn.fitness()>=p.fitness() //如果更好，就接受
 p = pn;
End Algorithm
```

```
def hillClimbing(s, maxGens, maxFails):    # 爬山演算法的主體函數
    print("start: ", s.str())              # 印出初始解
    fails = 0                              # 失敗次數設為 0
    # 當代數 gen<maxGen，且連續失敗次數 fails < maxFails 時，就持續嘗試尋找更好的解。
    for gens in range(maxGens):
        snew = s.neighbor()                #   取得鄰近的解
        sheight = s.height()               #   sheight=目前解的高度
        nheight = snew.height()            #   nheight=鄰近解的高度
        if (nheight >= sheight):           #   如果鄰近解比目前解更好
            print(gens, ':', snew.str())   #     印出新的解
            s = snew                       #     就移動過去
            fails = 0                      #     移動成功，將連續失敗次數歸零
        else:                              #   否則
            fails = fails + 1              #     將連續失敗次數加一
        if (fails >= maxFails):
            break
    print("solution: ", s.str())           #   印出最後找到的那個解
    return s                               #     然後傳回。
```

```python
import torch.nn as nn
import torch.nn.functional as F

class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.fc1 = nn.Linear(28*28, 10)

    def forward(self, x):
        x = x.view(-1, 28*28)
        x = self.fc1(x)
        return x
```

3.1 高度函數如何設計

Measure Measure Measure Measure Measure Measure Measure Measure Measure Measure Measure Measure Measure Measure Measure

$$\xi_{ij}(t) = P(x_t = i, x_{t+1} = j|y, v, w; \theta) = \frac{\alpha_i(t) a_{ij}^{w_t} \beta_j(t+1) b_j^{v_{t+1}}(y_{t+1})}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i(t) a_{ij}^{w_t} \beta_j(t+1) b_j^{v_{t+1}}(y_{t+1})} \tag{1}$$

Table 1: 不同模型的 MNIST 正確率

模型	正確率	說明
fc1	74%	
fc1s	92%	
fc2	10%	損失負無限大
fc2s	92%	
fc2net	95%	
fc2signet	91%	
LeNet	97%	

3.1.1 如何選取好的鄰居？

任何的參數變動，都可以創造出新的鄰居模型，因此、鄰居的選擇性是無限多的，我們面臨的問題是，該如何從無限多的鄰居當中，選擇一個有可能更好的適當鄰居呢？

在此、我們用了一些啟發式法則如下

1. 加一個新層
2. 將一層換成另一層
3. 調整某層的參數

Paragraph Paragraph Paragraph Paragraph Paragraph Paragraph Paragraph Paragraph Paragraph Paragraph Paragraph Paragraph Paragraph Paragraph Paragraph

4 實驗

[1, 2] and see [3].

The documentation for `natbib` may be found at

`http://mirrors.ctan.org/macros/latex/contrib/natbib/natnotes.pdf`

Of note is the command `\citet`, which produces citations appropriate for use in inline text. For example,

```
\citet{hasselmo} investigated\dots
```

produces

Hasselmo, et al. (1995) investigated…

`https://www.ctan.org/pkg/booktabs`

4.1 Figures

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetuer odio sem sed wisi. See Figure 1. Here is how you add footnotes. [2] Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

[2] Sample of the first footnote.

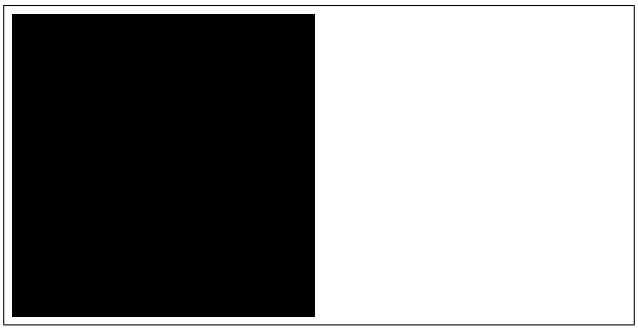

Figure 1: Sample figure caption.

Table 2: Sample table title

Part		
Name	Description	Size (μm)
Dendrite	Input terminal	$\sim$100
Axon	Output terminal	$\sim$10
Soma	Cell body	up to 10^6

4.2 Tables

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consectetuer tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo. See awesome Table 2.

4.3 Lists

- Lorem ipsum dolor sit amet
- consectetur adipiscing elit.
- Aliquam dignissim blandit est, in dictum tortor gravida eget. In ac rutrum magna.

參考文獻

[1] George Kour and Raid Saabne. Real-time segmentation of on-line handwritten arabic script. In *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, pages 417–422. IEEE, 2014.

[2] George Kour and Raid Saabne. Fast classification of handwritten on-line arabic characters. In *Soft Computing and Pattern Recognition (SoCPaR), 2014 6th International Conference of*, pages 312–318. IEEE, 2014.

[3] Guy Hadash, Einat Kermany, Boaz Carmeli, Ofer Lavi, George Kour, and Alon Jacovi. Estimate and replace: A novel approach to integrating deep neural networks with existing applications. *arXiv preprint arXiv:1804.09028*, 2018.