**Lab2: EEG classification**

**系級:智能系統 學號:312581006 姓名:張宸瑋**

1. Introduction

這次lab要實作出EGGNet以及DeepConvNet來分類EEG訊號，並且在這兩種不同結構的網路中，使用不同的激活函數，並且展示兩種結構的模型在不同的激勵函數下的最高準確率，並且為了更清出的看到訓練中準確率變化的趨勢，我們必須可視化在訓練階段以及測試階段，每個epoch的準確率曲線。

|  |
| --- |
|  |
| 訓練資料為BCI dataset |

這兩種不同架構的網路會根據下面兩張架構圖分別來做實作

|  |
| --- |
|  |
| EEGNet模型架構 |
|  |
| DeepConvNet模型架構 |

1. Experiment setups
2. The detail of your model

下面兩張圖是我在程式中實現的網路架構

|  |
| --- |
|  |
| EEGNet實現細節 |
|  |
| DeepConvNet實現細節 |

1. Explain the activation function(ReLU,Leaky ReLU,ELU)

ReLU 函數:

ReLU是一個簡單且廣泛使用的激活函數。它將所有負數的輸入值設置為0，而保留所有非負數的輸入值不變。數學表示為：f(x) = max(0, x)。ReLU的優點是計算高效，且能夠解決梯度消失的問題。然而，它存在一個稱為"神經元死亡"的問題，即某些神經元可能在訓練過程中永遠不會被激活，導致其梯度始終為0，無法更新權重。

|  |
| --- |
|  |
| ReLU Function示意圖 |
| Leaky ReLU 函數:   |  | | --- | | 為了解決ReLU的"神經元死亡"問題，Leaky ReLU被提出。Leaky ReLU在輸入值為負時引入一個小的斜率，使得當輸入值為負時，其梯度不再為0。數學表示為：f(x) = max(ax, x)，其中a是一個小的正數，通常很小，比如0.01。Leaky ReLU在ReLU的基礎上增加了一個超參數a，可以部分解決"神經元死亡"問題。 | | Leaky ReLU Function示意圖 | |

ELU 函數:

ELU是另一種解決"神經元死亡"問題的激活函數。ELU在輸入值為負時引入了一個負指數項，使得當輸入值為負時，其梯度不僅不會為0，還會保持平滑。數學表示為：f(x) = x (x >= 0)；f(x) = α(e^x - 1) (x < 0)，其中α是一個可選的超參數，控制輸入值為負時的曲率。ELU的主要優點是它能夠解決"神經元死亡"問題，並且在一些情況下可能比ReLU和Leaky ReLU表現更好，但計算成本較高。

|  |
| --- |
|  |
| ELU Function示意圖 |

1. Experimental results
   * 1. The highest testing accuracy

|  |
| --- |
|  |

* + 1. Comparison figures

EGGNet:

|  |
| --- |
|  |

設定參數:epoch:1300, learning rate:0.001, batch size:256, Optimizer: Adam, Loss function: torch.nn.BCEWithLigitsLoss()

DeepConvNet:

|  |
| --- |
|  |

設定參數:epoch:800,learning rate:0.001,batch size:256

Optimizer: Adam,Loss function: torch.nn.BCEWithLigitsLoss()

1. Discussion
   1. 調整Loss Function

|  |  |
| --- | --- |
| DeepConvNet | |
| Cross-Entropy | Binary Cross-Entropy |
|  |  |
| 兩者間不同激勵函數模型的準確率比較圖 | |
|  |  |
| 兩者間不同激勵函數模型的最高準確率 | |

|  |  |
| --- | --- |
| EGGNet | |
| Cross-Entropy | Binary Cross-Entropy |
|  |  |
| 兩者間不同激勵函數模型的準確率比較圖 | |
|  |  |
| 兩者間不同激勵函數模型的最高準確率 | |

討論:

這裡我將損失函數做了更換，將原本的Cross-Entropy更改為Binary Cross-Entropy，會有這個想法是因為網路的輸出僅僅是1跟0，因此相當於我可以把它看成一個二分類問題，因此我認為Binary Cross-Entropy在做法上比較直觀，而這裡我們可以觀察到，再像DeepConvNet這樣複雜的網路中，Binary Cross-Entropy的表現是更好的，我認為是Binary Cross-Entropy只需要計算兩個類別之間的交叉熵，而不需要對所有類別進行計算，因此在計算效率上更高。尤其是當網路比較複雜。這裡要注意的是，因為改成二分類問題，所以在nn.Linear()的out\_features是要更改成1的，並且也要對labels的shape做轉換才行。

* 1. 調整batch size

|  |  |  |
| --- | --- | --- |
| DeepConvNet | | |
| Batch size = 16 | Batch size = 64 | Batch size = 256 |
|  |  |  |
| 不同batch size的準確率比較圖 | | |
|  |  |  |
| 不同batch size的最高準確率 | | |

|  |  |  |
| --- | --- | --- |
| EGGNet | | |
| Batch size = 16 | Batch size = 64 | Batch size = 256 |
|  |  |  |
| 不同batch size的準確率比較圖 | | |
|  |  |  |
| 不同batch size的最高準確率 | | |

討論:

這裡我們可以看到，當betch size較小時，其準確率的表現看起來比較好，而原因我認為是較小的batch size，使更新頻率更高，可以更及時的更新模型參數，使模型更快收斂，但也有可能因為較小的batch size造成更不穩定的梯度估計，造成訓練過程不穩定。而batch size較大，會加快訓練速度，因為每一次迭代處理的樣本數量更多，能更有效利用運算資源，但也不是越大越快，如果太大的話可能會導致內存空間不夠，出現記憶體不足的問題，因此根據狀況選擇一個適合的batch size是很重要的

* 1. 調整drop out

|  |  |  |
| --- | --- | --- |
| EEGNet | | |
| P = 0.1 | P = 0.5 | P = 0.9 |
|  |  |  |
| 不同drop out rate的準確率比較圖 | | |
|  |  |  |
| 不同drop out rate的最高準確率 | | |

|  |  |  |
| --- | --- | --- |
| DeepConvNet | | |
| P = 0.1 | P = 0.5 | P = 0.9 |
|  |  |  |
| 不同drop out rate的準確率比較圖 | | |
|  |  |  |
| 不同drop out rate的最高準確率 | | |

討論:

這裡我們可以看到，如果drop out rate太大，代表大部分的神經元都關閉，可能會導致模型更難訓練複雜的模型，導致欠擬合，而如果drop out rate太小，代表用大多數的神經元來訓練模型，這樣可能會造成模型過擬合，泛化能力會比較差，因此選擇適當的drop out rate是很重要的。

總結:

這此實驗我們實作兩個應用於腦電圖(EEG)信號分類的兩個網路，從架構上來看DeepConvNet是一個相對較深的神經網路，由多個卷積層跟池化層交錯組成，最後透過全連結層進行分類，而EEGNet是一個相對較簡單的網路架構，只由一個混合卷積層，和深度可分離卷積層組合而成，並使用短時間平均池化(STAP)來獲得特徵，而在訓練的資源消耗上，DeepConvNet因為架構較為複雜，參數量較大，需要更多的計算資源以及時間訓練。

而在足夠的訓練次數下，我們可以看到EGGNet相對於DeepConvNet表現更好，那是因為EGGNet架構較為簡單，參數量較少，可以避免過度擬合的問題，但我認為這也是取決於具體的數據集跟任務，在某些方面DeepConvNet可能因為複雜的架構，造成它的表現能力更好。

而我們也可以看到，對於各個不同的網路，其參數的調整是很重要的，參數的調整會直接影響到模型的性能和效果，因此仔細調整參數，找到最適合的配置，也是很重要的一環。