

Lab3 EEG classification

311553030 林亮丞

1. Introduction (20%)

在這次作業中我們需要在 BCI competition dataset 上使用 EEGNet, DeepConvNet 去解決分類問題，同時去比較在 2 種 model 下使用 3 種不同的 activation function(ReLU, Leaky ReLU, ELU)的不同結果。

• **Input: [B, 1, 2, 750]** **Output: [B, 2]** **Ground truth: [B]**

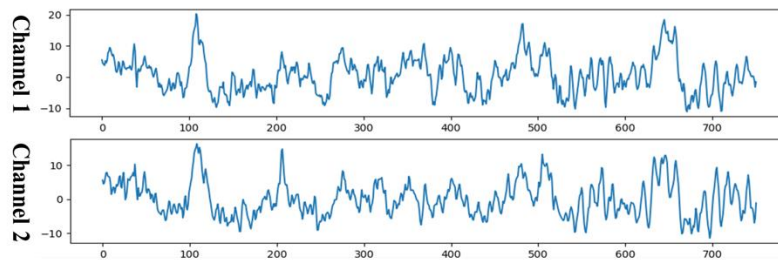


Fig1: BCI dataset

2. Experiment set up (20%)

2.1. The detail of your model

EEGNet 和 DeepConvNet 我皆使用 3 種 activation function 以及使用以下的 Hyperparameter:

- Batch_size: 256
- Learning_rate: 0.01
- Epochs: 300
- Optimizer: Adam
- Loss: CrossEntropy

2.1.1. EEGNet

EEGNet 使用了 depthwise 和 separable 代替了 conventional convolution，depthwise 有效減少了需要的參數量，而 separable 在減少了需要的參數量的同時，還可以擴展 feature map 的數目並加以組合。架構圖如下：

```
EEGNet(  
  (firstconv): Sequential(  
    (0): Conv2d(1, 16, kernel_size=(1, 51), stride=(1, 1), padding=(0, 25), bias=False)  
    (1): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
  )  
  (depthwiseConv): Sequential(  
    (0): Conv2d(16, 32, kernel_size=(2, 1), stride=(1, 1), groups=16, bias=False)  
    (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
    (2): ELU(alpha=1.0)  
    (3): AvgPool2d(kernel_size=(1, 4), stride=(1, 4), padding=0)  
    (4): Dropout(p=0.25)  
  )  
  (separableConv): Sequential(  
    (0): Conv2d(32, 32, kernel_size=(1, 15), stride=(1, 1), padding=(0, 7), bias=False)  
    (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
    (2): ELU(alpha=1.0)  
    (3): AvgPool2d(kernel_size=(1, 8), stride=(1, 8), padding=0)  
    (4): Dropout(p=0.25)  
  )  
  (classify): Sequential(  
    (0): Linear(in_features=736, out_features=2, bias=True)  
  )  
)
```

2.1.2. DeepConvNet

DeepConvNet 就是傳統的 CNN 架構，架構圖如下：

Layer	# filters	size	# params	Activation	Options
Input		(C, T)			
Reshape		(1, C, T)			
Conv2D	25	(1, 5)	150	Linear	mode = valid, max norm = 2
Conv2D	25	(C, 1)	$25 * 25 * C + 25$	Linear	mode = valid, max norm = 2
BatchNorm			$2 * 25$		epsilon = 1e-05, momentum = 0.1
Activation				ELU	
MaxPool2D		(1, 2)			
Dropout					p = 0.5
Conv2D	50	(1, 5)	$25 * 50 * 5 + 50$	Linear	mode = valid, max norm = 2
BatchNorm			$2 * 50$		epsilon = 1e-05, momentum = 0.1
Activation				ELU	
MaxPool2D		(1, 2)			
Dropout					p = 0.5
Conv2D	100	(1, 5)	$50 * 100 * 5 + 100$	Linear	mode = valid, max norm = 2
BatchNorm			$2 * 100$		epsilon = 1e-05, momentum = 0.1
Activation				ELU	
MaxPool2D		(1, 2)			
Dropout					p = 0.5
Conv2D	200	(1, 5)	$100 * 200 * 5 + 200$	Linear	mode = valid, max norm = 2
BatchNorm			$2 * 200$		epsilon = 1e-05, momentum = 0.1
Activation				ELU	
MaxPool2D		(1, 2)			
Dropout					p = 0.5
Flatten					
Dense	N			softmax	max norm = 0.5

2.2. Explain the activation function (ReLU, Leaky_ReLU, ELU)

$$\text{ReLU}(x) = \max(0, x)$$

$$\text{Leaky_ReLU}(x) = \begin{cases} x, & x \geq 0 \\ \text{negative_slope} * x, & \text{otherwise} \end{cases}$$

$$\text{ELU}(x) = \max(0, x) + \min(0, \alpha * (\exp(x) - 1))$$

ReLU 的優點是計算簡單，不容易出現梯度消失的情況，但是當 x 小於 0 時只會輸出 0，導致權重更新不了。

Leaky_ReLU 的優點是負數時有梯度，但缺點是因為有了負數的輸出導致非線性的程度降低，在分類任務上效果可能比 sigmoid 和 ReLU 更差。

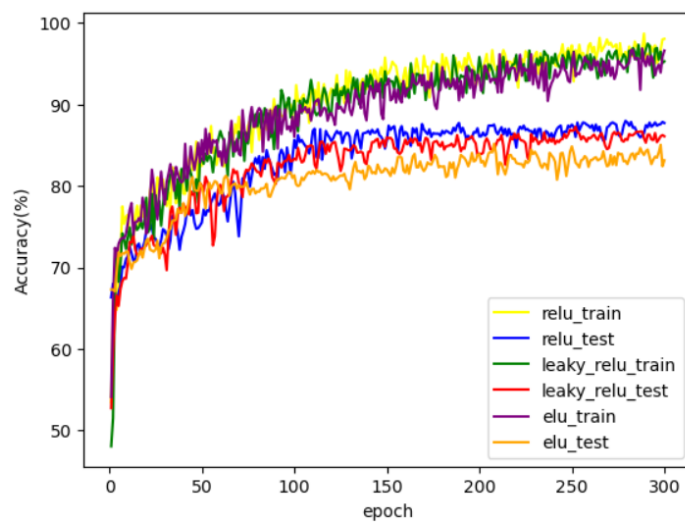
ELU 的優點也是負數時有梯度，以及會較緩慢的變的平滑，缺點是包含指數運算所以計算時間較長。

3. Experimental results (30%)

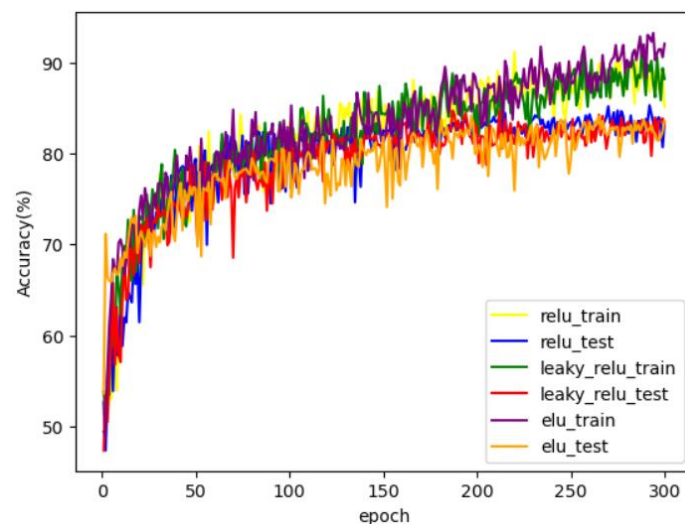
3.1. The highest testing accuracy

	ReLU	Leaky ReLU	ELU
EEGNet	87.98%	87.27%	85.17%
DeepConvNet	85.32%	84.73%	84.20%

3.2. Comparison figures



(a) EEGNet accuracy



(b) DeepConvNet accuracy

4. Discussion (30%)

我比較了多種 `batch_size` 下的變化，發現在不同的 `batch_size` 下，它們最好的準確率大多是采用 ReLU，偶爾使用 Leaky ReLU，但兩者的差別通常不到 1%。至於 ELU 的表現則一直是最差的。我認為這是因為在使用激活函數之前的輸出值中，正值占據了主導地位，而負值較少。但這些負值可能正好位於 ELU 負值變化較大的區域，導致使其準確率較低。

且整體訓練下來準確率蠻不穩定的，舉 EEGNet 的 `batch_size` 是 64 來說好了，經過多次重新訓練來測試，有時最高只會有 86，有時卻能到 88，我認為這是因為數據集資料太少，且分布不均，導致在 shuffle 取資料訓練時經常會把不適合一起訓

練的資料放在一起訓練，導致準確率變化幅度大，不過在 `batch_size` 調高後準確率變化幅度就比較小。