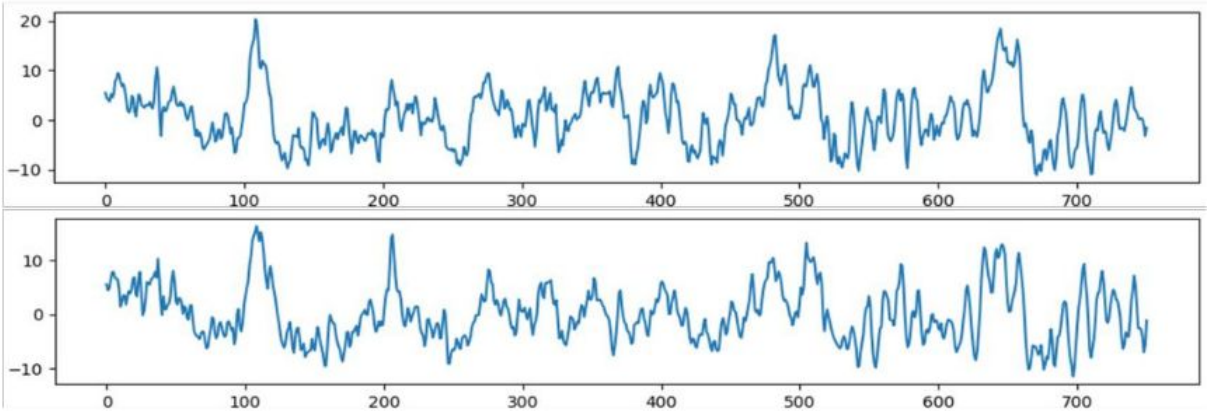
Lab2

資工碩一 吳承翰 0856105

1.Introduction

使用EEGNet與DeepConvNet解決分類的問題，訓練資料為BCI competition
shape為(C=1,H=2,W=750)



2.Experiment Setup

a.detail of my model

EEGNet 模型:

```
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)
  )
)
```

使用了depthwise-seperable convolution,
是基於傳統convolution的輕量化版本，可以降低常數倍的參數數量,
提昇訓練與evaluate的速度，但右不至於影響太多accuracy

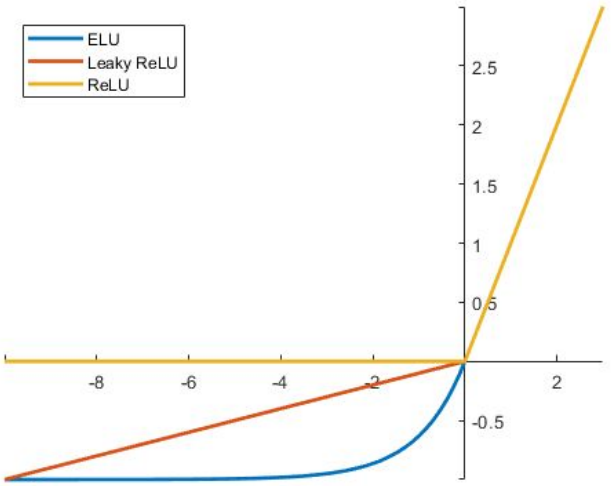
DeepConvNet 模型:

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 * C + 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 * C + 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 * C + 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

為傳統的CNN架構

C->(CBAPD)->(CBAPD)->(CBAPD)->(CBAPD)->fully connected
C: convolution
B: batchnormalized
A: activation function
P: pooling
D: dropout

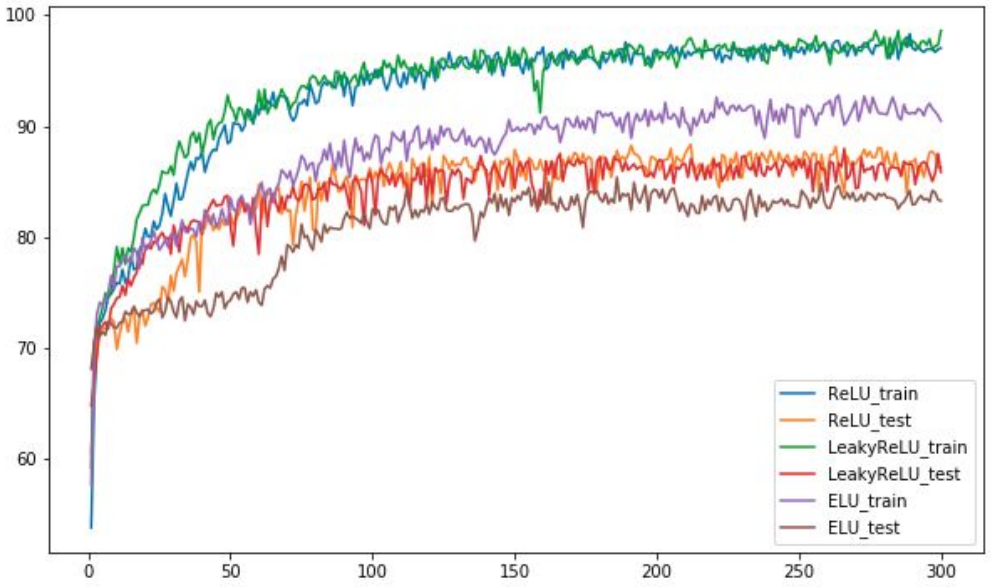
b.activation function:



在backpropagate時，LeakyReLU與ELU比較容易訓練，因為value一旦小於0的話還是有一個梯度存在

3.Experimental results

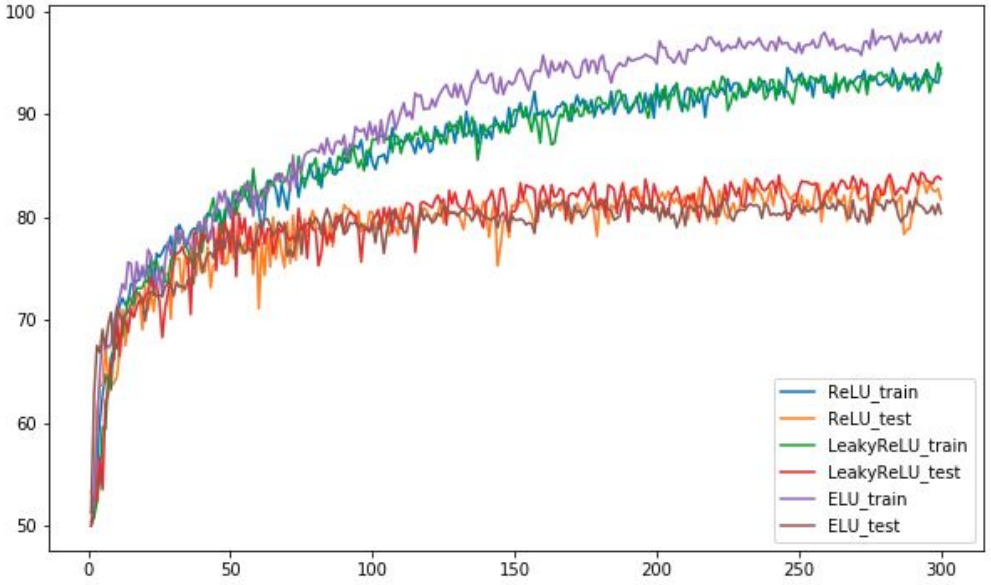
EEGNet:



訓練300個epochs

accuracy 可以達到85%左右

DeepConvNet:



訓練300個epochs
accuracy 可以達到82%左右

	RELU	Leaky ReLU	ELU
EEGNet	**88.33%**	**87.96%**	**85.37%**
DeepConvNet	83.70%	84.35%	81.85%

4.Discuss

(1)一開始不是很清楚dataloader怎使用，後來才知道要先把資料先放到Tensordataset()再放到DataLoader()
(2)在forwarding model時，data與model都要.to(device) 放到gpu才可以跑，tensor可以在cpu與gpu上運行，numpy只能在cpu上運行
(3)target y的datatype必須是torch.long
(4)加入regularization term之後acc可以大幅提昇至88%，原本的acc只能到82%