Deep learning
Lab3 : EEG classification

1. Introduction:
   這次作業要用 pytorch 實作 EEGNet 和 DeepConvNet 替 EEG signals 分類。我利用 dictionary 儲存 activation(acts = {name: function})，可以直接用迴圈執行完所有組合。

2. Experiment:
   A. The detail of your model
      - EEGNet
        模型架構完全採用作業簡報的說明文件，而 loss function 選擇 cross entropy，因為 pytorch 的 cross entropy 有包含 softmax，所以不需要再使用 softmax 分類，optimizer 則是用 Adam (A METHOD FOR STOCHASTIC OPTIMIZATION)

```
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)
  )
)
```

      - DeepConvNet

        DeepConvNet 和 EEG 一樣完全使用簡報的模型架構，loss

        function 選擇 cross entropy，optimizer 用 Adam

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

B. Explain the activation function (ReLU, Leaky ReLU, ELU)
   - ReLU
     把負數都改成 0，正數保持不變。缺點是產生負數的 neuron 不會再對 loss 有貢獻
   - Leaky ReLU
     在 ReLU 的基礎上，對負數乘上一個較小的值，保留負數對 loss 的影響
   - ELU
     在 ReLU 的基礎上，對負數做 exponential

3. Experimental results:
   A. The highest testing accuracy
      下表(圖)是 learning rate = 0.1, batch size = 135, epoch = 1000 的結果

	ReLU	Leaky ReLU	ELU
EEG	**0.83**	**0.88**	**0.89**
DeepConv	0.82	0.84	0.83

eeg - Dictionary (3 elements)

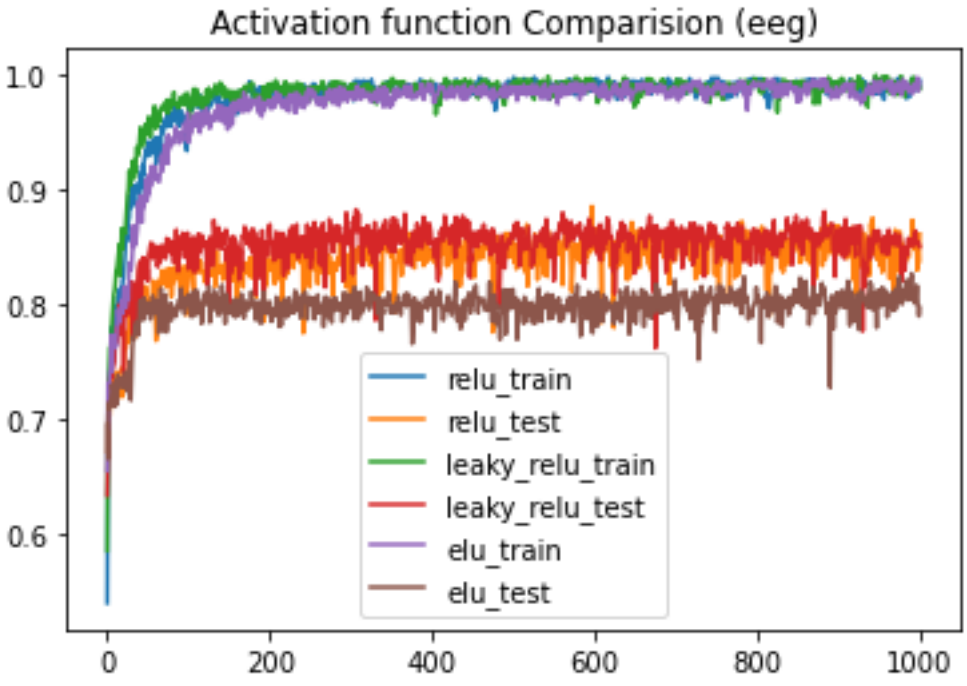
Key	Type	Size	
elu	Array of float32	1	0.82685184
leaky_relu	Array of float32	1	0.88240737
relu	Array of float32	1	0.8851852

deepconv - Dictionary (3 elements)

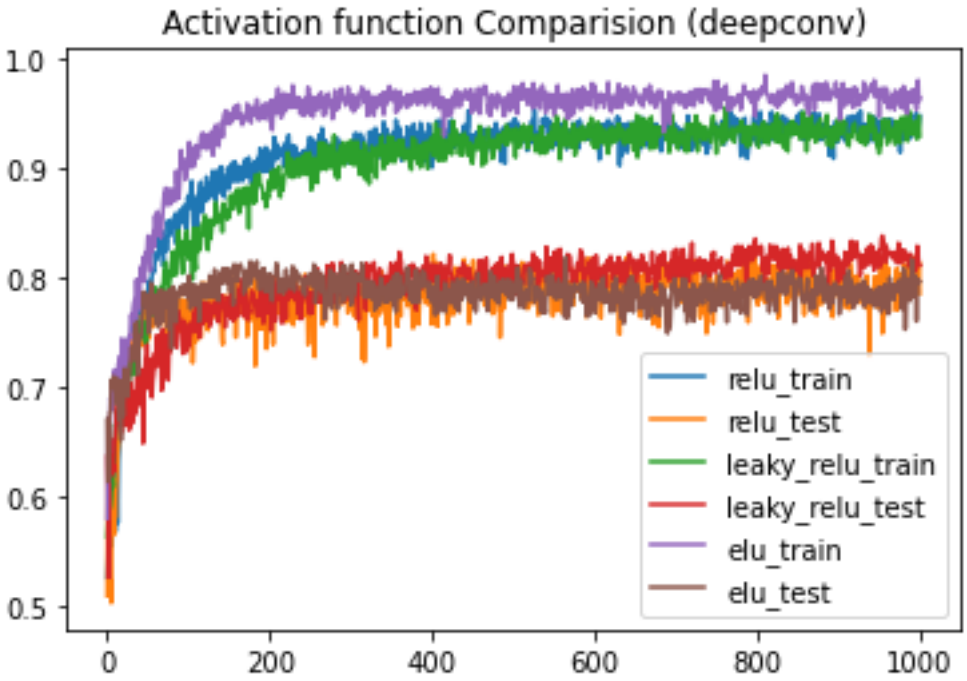
Key	Type	Size	
elu	Array of float32	1	0.8185185
leaky_relu	Array of float32	1	0.8379629
relu	Array of float32	1	0.8259259

B. Comparison figures

- EEGNet

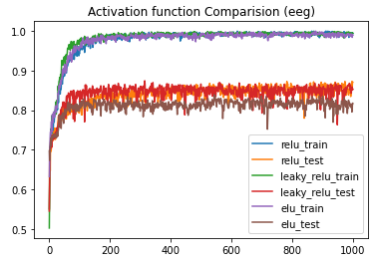
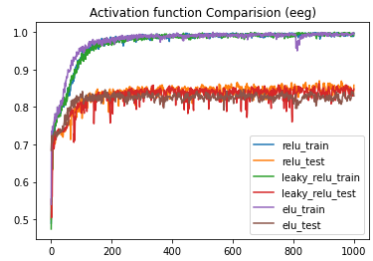
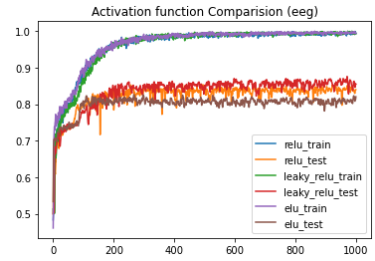
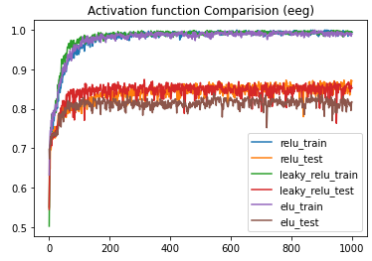
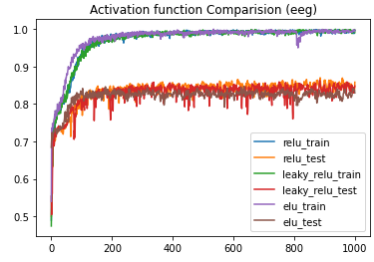
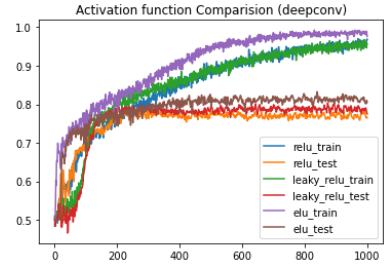


- DeepConvNet



4. Discussion:

比較不同 batch size 對 accuracy 的影響

Size	270	540	1080
EEG	 Key / Type / Size elu — Array of float32 1 — 0.8157407 leaky_relu — Array of float32 1 — 0.8796296 relu — Array of float32 1 — 0.8898148	 Key / Type / Size elu — Array of float32 1 — 0.8472222 leaky_relu — Array of float32 1 — 0.86296296 relu — Array of float32 1 — 0.8703703	 Key / Type / Size elu — Array of float32 1 — 0.8259259 leaky_relu — Array of float32 1 — 0.8759259 relu — Array of float32 1 — 0.85462964
Deep	 Key / Type / Size elu — Array of float32 1 — 0.8064815 leaky_relu — Array of float32 1 — 0.8185185 relu — Array of float32 1 — 0.81203705	 Key / Type / Size elu — Array of float32 1 — 0.8203704 leaky_relu — Array of float32 1 — 0.81296295 relu — Array of float32 1 — 0.7916666	 Key / Type / Size elu — Array of float32 1 — 0.8324074 leaky_relu — Array of float32 1 — 0.80370367 relu — Array of float32 1 — 0.787963