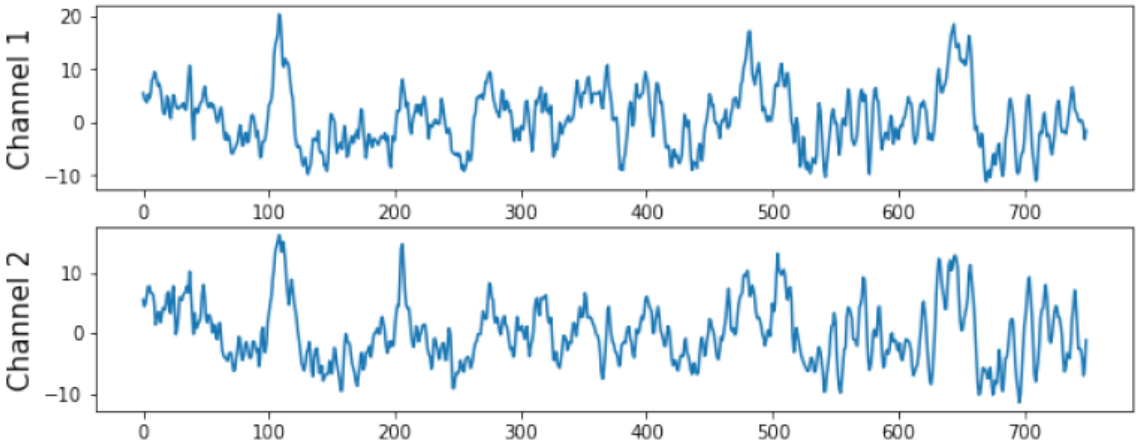
LAB2: EEG classifiction

0851915 徐玉山 2019.7.30

1 Introduction

這個dataset有兩個channel，每一個chanel有750個點，label分為左或是右



使用的網路是
EEGNET 以及 DEEPCONVNET
使用三種不同的激勵函數，ReLu LeakyReLU ELU

Experiment setup

2.1 資料讀取

首先先把 Data 讀入<---- dataloader.py (http://dataloader.py)

從 numpy 到 tensor 是利用 pytorch library

import torch.utils.data as Data

```
    train_dataset = Data.TensorDataset(train_x,train_y)
    train_loader = Data.DataLoader(
        dataset=train_dataset,
        batch_size=BATCH_SIZE,
        shuffle=True,
        num_workers=0
    )
    for epoch in range(MAX_EPOCH):
        for step, (batch_x, batch_y) in enumerate(train_loader):
            optimizer.zero_grad()   # zero the gradient buffers
            output = net(batch_x)
            output = output.float()
            loss = criterion(output, batch_y)
            loss.backward()
            optimizer.step()    # Does the update
```

2.2 網路架構

- DeepConvNet

```
DeepConvNet(
  (modelA): Sequential(
    (0): Conv2d(1, 25, kernel_size=(1, 5), stride=(1, 1))
    (1): Conv2d(25, 25, kernel_size=(2, 1), stride=(1, 1))
    (2): BatchNorm2d(25, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (3): ELU(alpha=1.0)
    (4): MaxPool2d(kernel_size=(1, 2), stride=(1, 2), padding=0, dilation=1, ceil_mode=False)
    (5): Dropout(p=0.5)
  )
  (modelB): Sequential(
    (0): Conv2d(25, 50, kernel_size=(1, 5), stride=(1, 1))
    (1): BatchNorm2d(50, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ELU(alpha=1.0)
    (3): MaxPool2d(kernel_size=(1, 2), stride=(1, 2), padding=0, dilation=1, ceil_mode=False)
    (4): Dropout(p=0.5)
  )
  (modelC): Sequential(
    (0): Conv2d(50, 100, kernel_size=(1, 5), stride=(1, 1))
    (1): BatchNorm2d(100, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ELU(alpha=1.0)
    (3): MaxPool2d(kernel_size=(1, 2), stride=(1, 2), padding=0, dilation=1, ceil_mode=False)
    (4): Dropout(p=0.5)
  )
  (modelD): Sequential(
    (0): Conv2d(100, 200, kernel_size=(1, 5), stride=(1, 1))
    (1): BatchNorm2d(200, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ELU(alpha=1.0)
    (3): MaxPool2d(kernel_size=(1, 2), stride=(1, 2), padding=0, dilation=1, ceil_mode=False)
    (4): Dropout(p=0.5)
  )
  (modelE): Linear(in_features=8600, out_features=2, bias=True)
)
```

- EEGNET

```
EGG(
  (modelA): 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)
  )
  (modelB): 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)
  )
  (modelC): 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)
  )
  (out): Linear(in_features=736, out_features=2, bias=True)
)
```

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

- 三種函數的定義
  - ReLU

$$ReLU(x) = max(0, x) \tag{1}$$

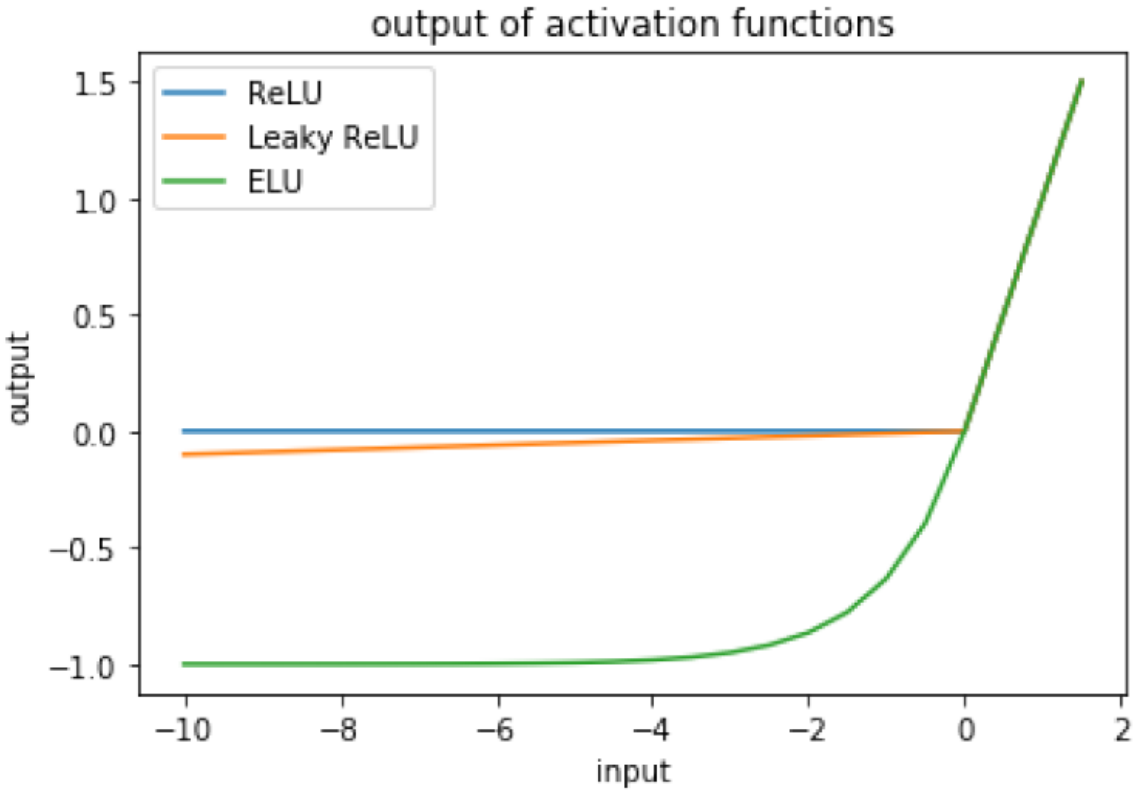
  - Leaky ReLU

$$LeakyReLU(x) = \begin{cases} x, & \text{if x} > 0 \\ \text{negative\_slope} \times x, & \text{otherwise} \end{cases} \tag{2}$$

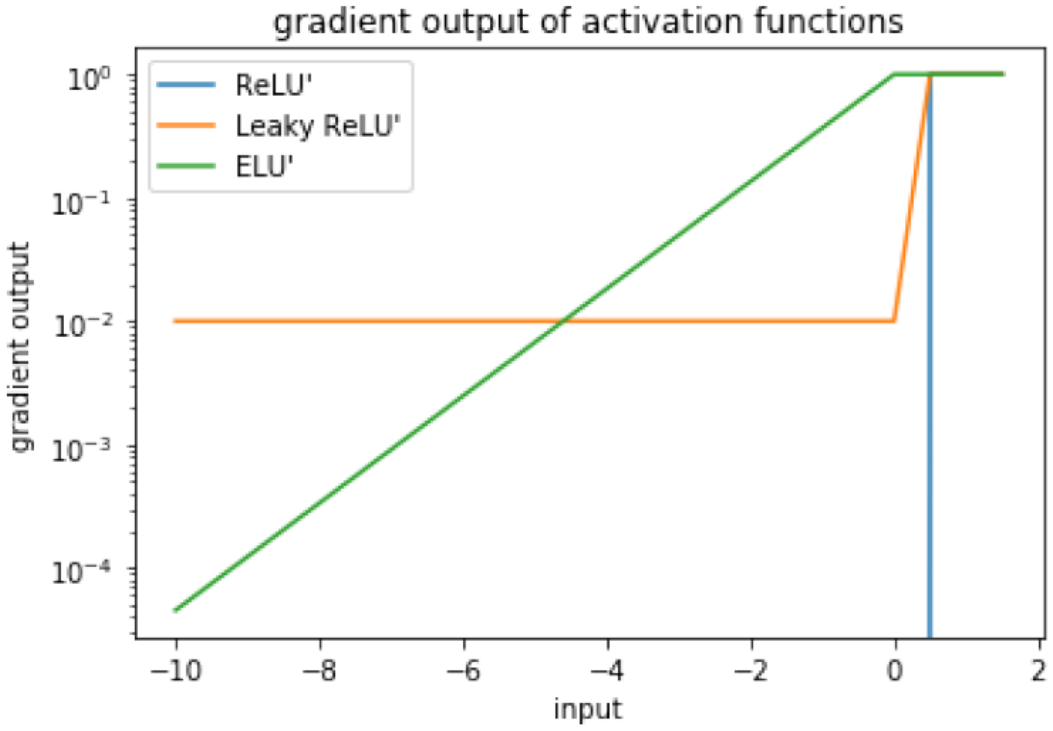
  - ELU

$$ELU(x) = max(0, x) + min(0, \alpha * (exp(x) - 1)) \tag{3}$$

- 以下是三種激勵函數的圖形



- 以下是三種激勵函數之導函數圖形



3. Experimental result

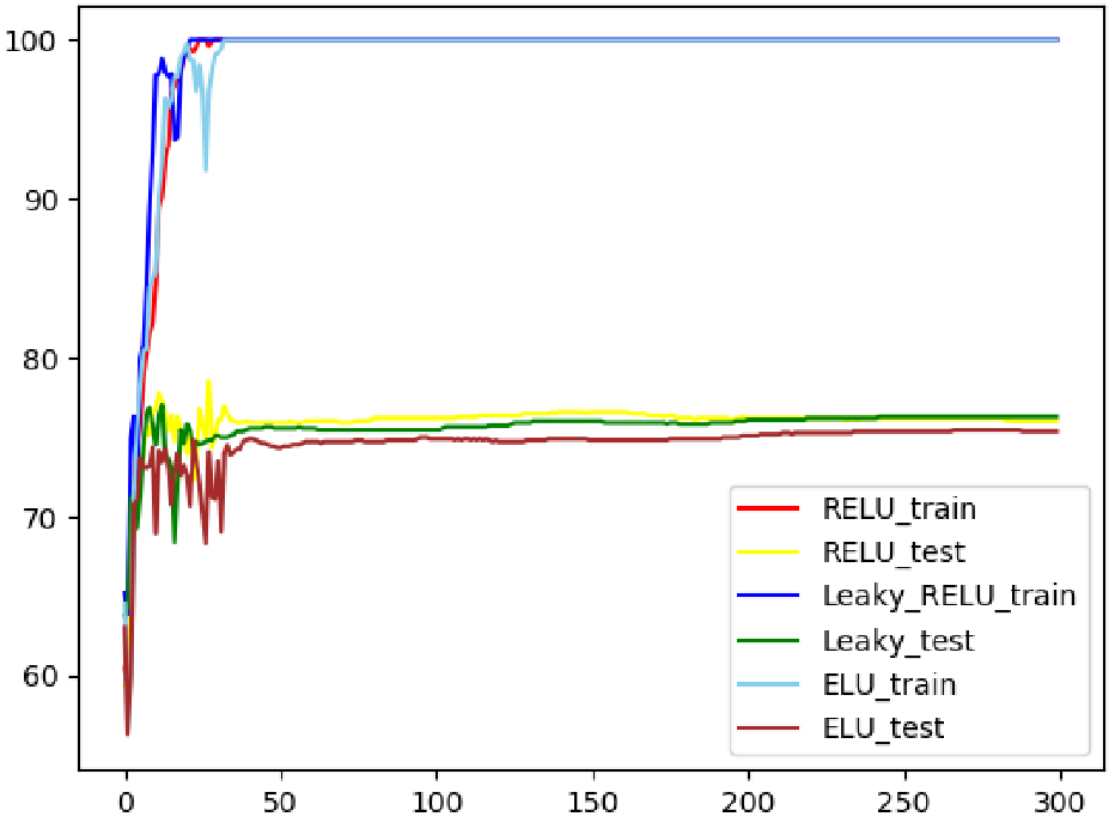
Hyper parameters

```
optimizer : Adam
criterion (loss) : CrossEntropy
epoch size : 300
batch size : 64
learning rate for EEGNet : 0.01
learning rate for DeepConvNet : 0.001
```

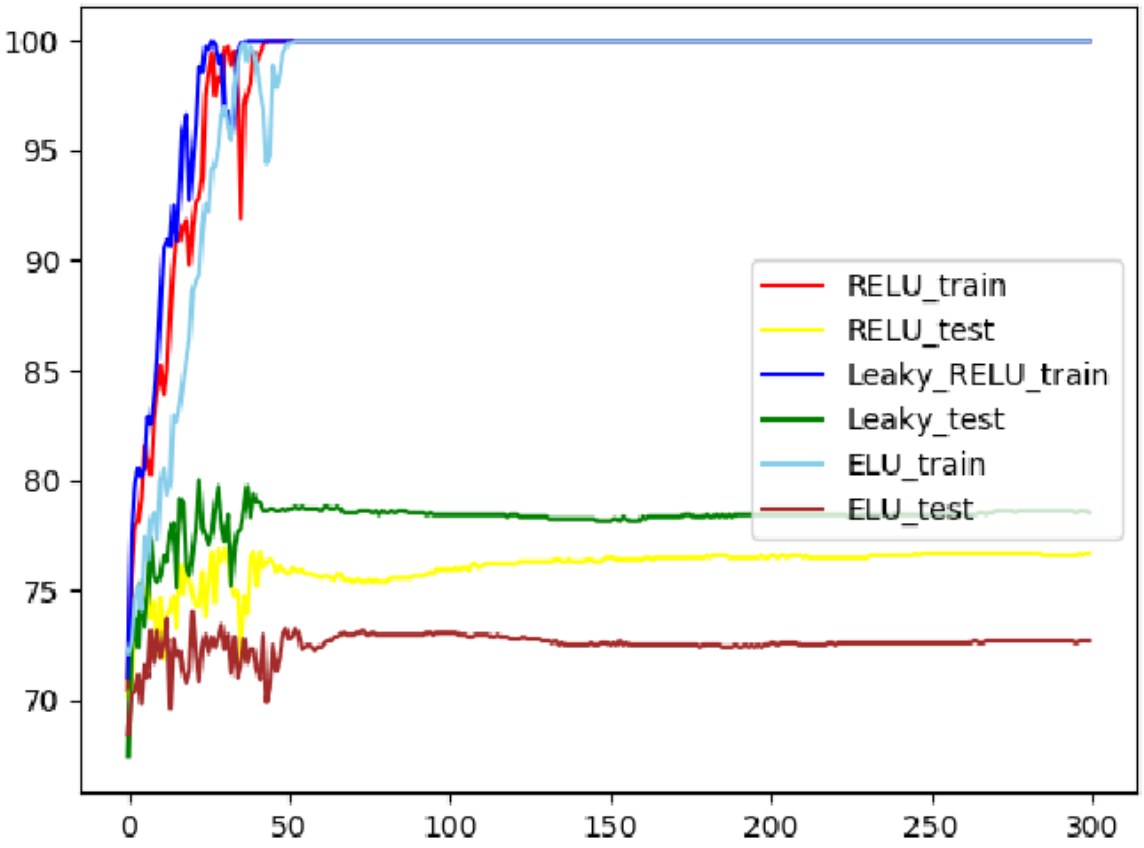
Highest testing accuracy

\	ReLU	Leaky ReLU	ELU
EEG	78.61	86.76(epoch多，有調降batchsize)	78.33
DeepConvNet	76.01	78.52	75.37

以下都是用固定batchsize = 64 跑出來的 實際上我訓練的時候會調降batchsize

- DeepConvNet LeakyReLU max: 78.52
- 

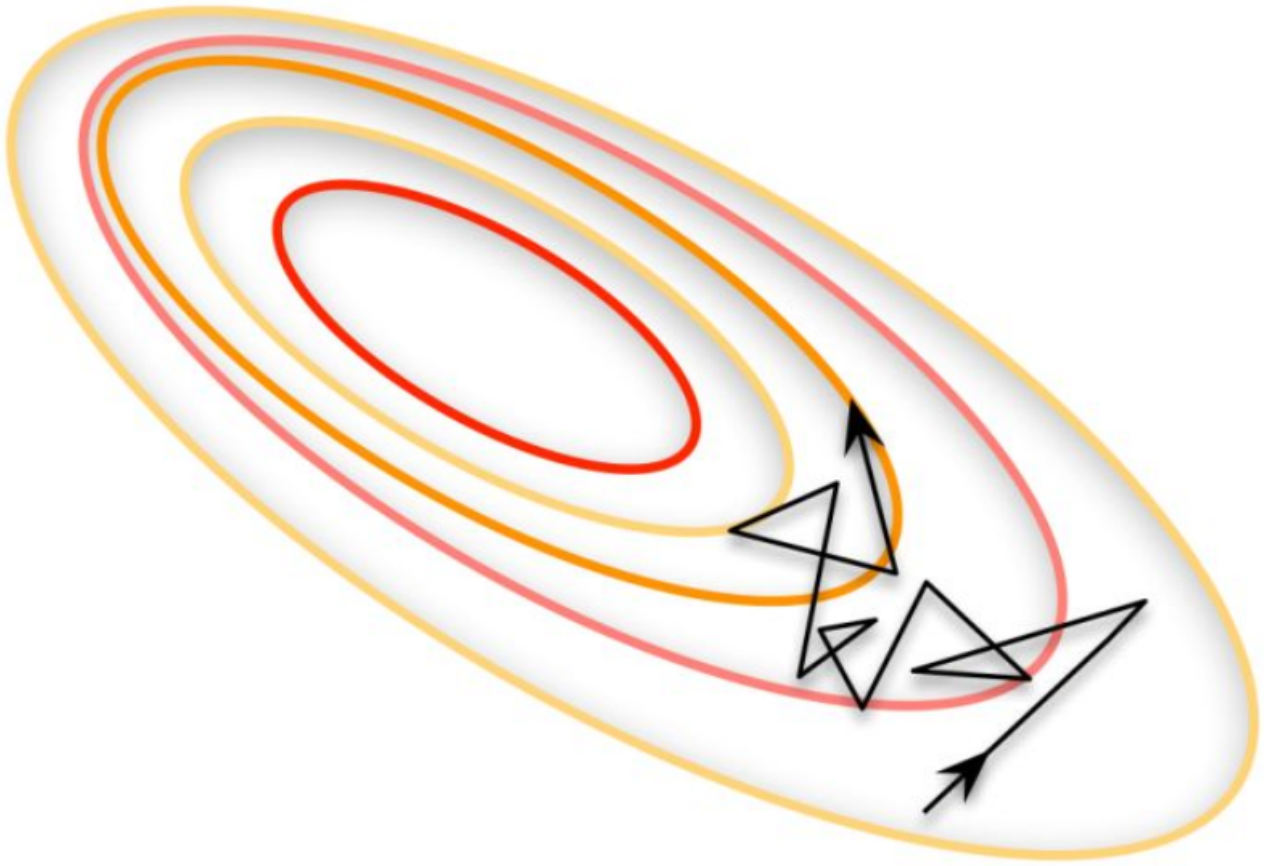
- EEGNET Leaky_RELU max:86.57 (下面的圖只是跑 300 epoch的結果)

- 

4. Discussion

Batchsize

Batchsize 大會讓收斂速度變快，但是精度下降。
Batchsize 小會讓每次修正方向已各自樣本梯度方向修正，各自為政，收斂慢。



BatchNorm

BatchNorm layer，估計這個batch的平均和標準差

Dropout

Dropout 層會讓 output 每次不一定一樣
Dropout rate 變大=>更多的input會變成0
反之亦然
Dropout layer 可以讓train和test之間的差距變少