Lab 2 experiment report

Name: 黃兆宇
ID: 0856601

1. Introduction

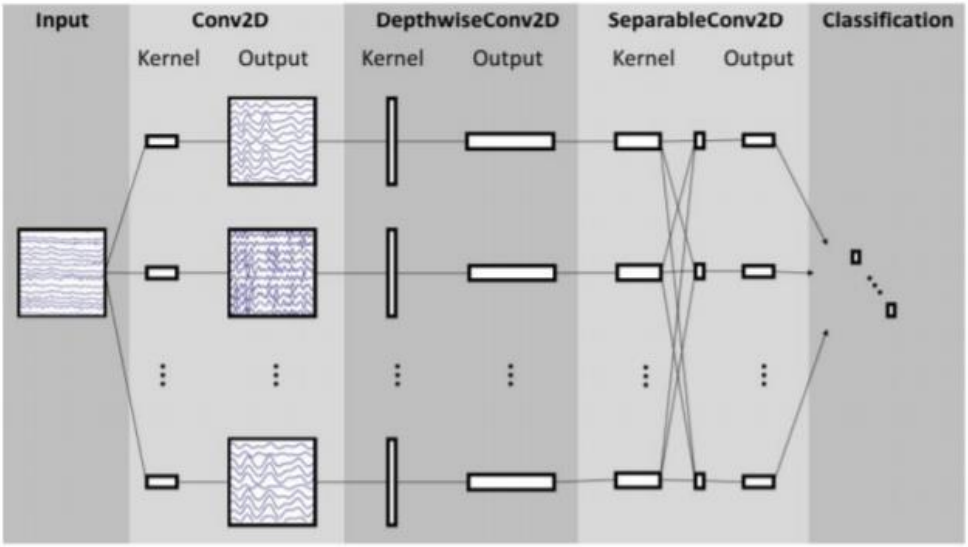
In this homework, we need to train two kinds of deep convolution networks, EEGNet and DeepConvNet, the dataset is BCI Competition III. It is a dataset with 2 classes (left hand, right hand). The dimension of input is [B, 1, 2, 750], the output dimension is [B, 2], and the label dimension is [B].

2. Experiment Set Up

A. The detail of model

- EEGNet

  EEGNet is composed of Conv2D, DepthwiseConv2D, Separable2D and Classification.



More detail of the model

C = number of channels, T = number of time points, F1 = number of temporal filters,
D = depth multiplier (number of spatial filters), F2 = number of pointwise filters,
and N = number of classes.

Block	Layer	# filters	size	# params	Output	Activation	Options
1	Input				(C, T)		
	Reshape				(1, C, T)		
	Conv2D	F_1	(1, 64)	$64 * F_1$	(F_1, C, T)	Linear	mode = same
	BatchNorm			$2 * F_1$	(F_1, C, T)		
	DepthwiseConv2D	D * F_1	(C, 1)	$C * D * F_1$	(D * F_1, 1, T)	Linear	mode = valid, depth = D, max norm = 1
	BatchNorm			$2 * D * F_1$	(D * F_1, 1, T)		
	Activation				(D * F_1, 1, T)	ELU	
	AveragePool2D		(1, 4)		(D * F_1, 1, T // 4)		
	Dropout*				(D * F_1, 1, T // 4)		$p = 0.25$ or $p = 0.5$
2	SeparableConv2D	F_2	(1, 16)	$16 * D * F_1 + F_2 * (D * F_1)$	(F_2, 1, T // 4)	Linear	mode = same
	BatchNorm			$2 * F_2$	(F_2, 1, T // 4)		
	Activation				(F_2, 1, T // 4)	ELU	
	AveragePool2D		(1, 8)		(F_2, 1, T // 32)		
	Dropout*				(F_2, 1, T // 32)		$p = 0.25$ or $p = 0.5$
	Flatten				(F_2 * (T // 32))		
Classifier	Dense	N * (F_2 * T // 32)			N	Softmax	max norm = 0.25

- DeepConvNet

  The structure of DeepConvNet, in this lab, C=2, T=750, N = 2

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. Activation function

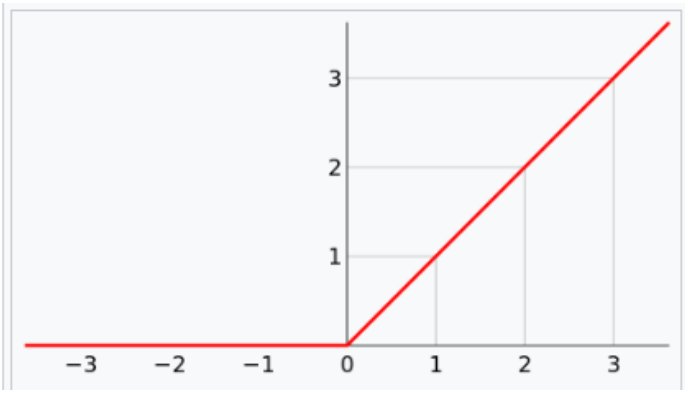
- ReLU

  When x is less than 0, f(x) will be 0, and if x is bigger than 0, f(x) will be x.

$$f(x) = \max(0, x)$$

For the neural network that use the activation function, the neuron output will be:

$\max(0, \mathbf{w}^T\mathbf{x} + b)$ x is the input of neuron



- Leaky ReLU

  When x <= 0, there is a small gradient for x.

$$f(x) = \begin{cases} x & \text{if } x > 0, \\ 0.01x & \text{otherwise.} \end{cases}$$

- ELU

The *exponential linear unit* (ELU) with $0 < \alpha$ is

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha\left(\exp(x) - 1\right) & \text{if } x \leq 0 \end{cases}, \quad f'(x) = \begin{cases} 1 & \text{if } x > 0 \\ f(x) + \alpha & \text{if } x \leq 0 \end{cases}.$$

α value for the ELU formulation.Default: 1.0

3. Experiment results

Accuracy screenshot of two models:

Highest accuracy: **87.31** (EEGNet with activation function ReLU)

EEGNet

```
Train Accuracy: 98.52 %
Epoch:  294 | train loss: 0.1114
Test Accuracy: 82.59 %

Train Accuracy: 99.07 %
Epoch:  295 | train loss: 0.0294
Test Accuracy: 82.31 %

Train Accuracy: 98.52 %
Epoch:  296 | train loss: 0.0513
Test Accuracy: 80.93 %

Train Accuracy: 98.70 %
Epoch:  297 | train loss: 0.0067
Test Accuracy: 82.22 %

Train Accuracy: 99.17 %
Epoch:  298 | train loss: 0.0225
Test Accuracy: 82.22 %

Train Accuracy: 99.07 %
Epoch:  299 | train loss: 0.0089
Test Accuracy: 83.52 %

Highest Test Accuracy (ELU): 82.59 %
Highest Test Accuracy (ReLU): 87.31 %
Highest Test Accuracy (LReLU): 84.07 %
```

DeepConvNet

```
Train Accuracy: 92.87 %
Epoch:  294 | train loss: 0.0784
Test Accuracy: 73.33 %

Train Accuracy: 90.46 %
Epoch:  295 | train loss: 0.6395
Test Accuracy: 73.43 %

Train Accuracy: 92.31 %
Epoch:  296 | train loss: 0.1575
Test Accuracy: 73.06 %

Train Accuracy: 90.19 %
Epoch:  297 | train loss: 0.2564
Test Accuracy: 71.94 %

Train Accuracy: 91.94 %
Epoch:  298 | train loss: 0.1954
Test Accuracy: 71.02 %

Train Accuracy: 91.11 %
Epoch:  299 | train loss: 0.0819
Test Accuracy: 73.98 %

Highest Test Accuracy (ELU): 78.52 %
Highest Test Accuracy (ReLU): 76.30 %
Highest Test Accuracy (LReLU): 77.31 %
```

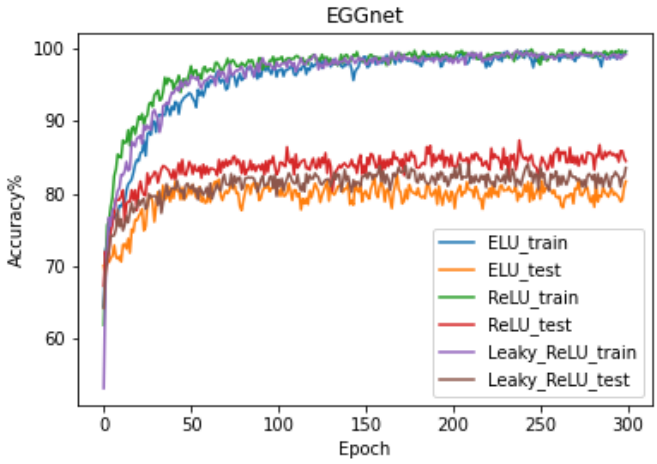
Hyper parameter of EEGnet:

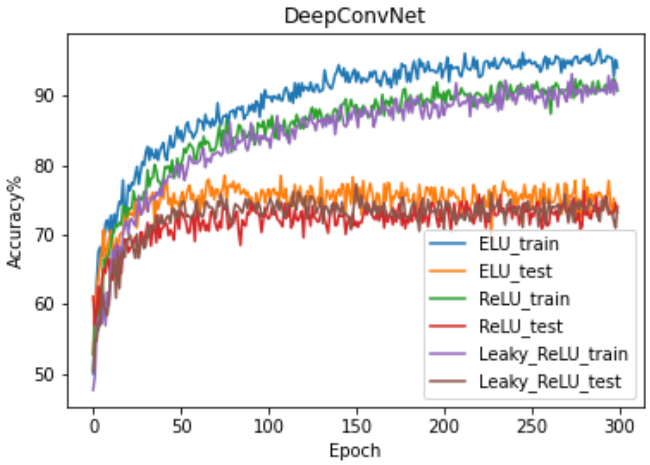
```
EPOCH = 300
LR = 0.005
BATCH_SIZE = 128
```

Hyper parameters of DeepConvNet:

```
EPOCH = 300
LR = 0.01
BATCH_SIZE = 64
```

Comparison figures of models:





4. Discussion

In this Lab, I learned how to build two different kinds of deep convolution networks, I tried to use different parameters to train the models, and find out that when setting batch size:128, learning rate:0.005 and use the activation function ReLU on the EEGNet Model can get the best result(87.31). When building the model, I also found out that between the last linear layer and the last convolutional layer, it's important to use a function view(-1, in_features of linear layer) to make the output of convolutional layer fit the input of linear layer.

5. Reference:

EEGNet
https://arxiv.org/pdf/1611.08024.pdf
ELU
https://arxiv.org/pdf/1511.07289.pdf
ReLU
https://en.wikipedia.org/wiki/Rectifier_(neural_networks)
Leaky ReLU
https://zh.wikipedia.org/wiki/%E7%BA%BF%E6%80%A7%E6%95%B4%E6%B5%81%E5%87%BD%E6%95%B0