Lab 4-1: EEG classification

310551083 許嘉倫

1 Introduction

In this lab, we will need to implement simple EEG classification models which are EEGNet, DeepConvNet with BCI competition dataset. Additionally, we need to try different kinds of activation function including ReLU, Leaky ReLU, ELU.

1.1 Dataset

- BCI Competition III – IIIb
- 2 classes, 2 bipolar EEG channels
- Training data: S4b_train.npz, X11b_train.npz
- Testing data: S4b_test.npz, X11b_test.npz
- Input: [B, 1, 2, 750] Output: [B, 2] Ground truth: [B] B means batch size

There are 1080 data in both training data and testing data. Figure 1 is the visualization of BCI dataset.

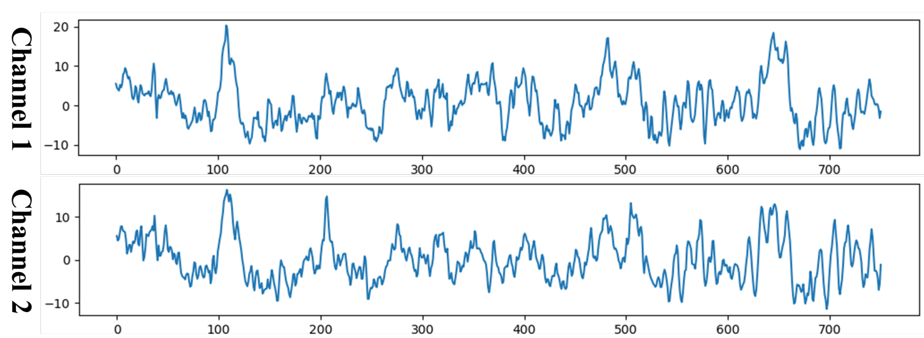


Figure 1: BCI Dataset

2 Experiment set up

Both EEGNet and DeepConvNet will try different activation ReLU, ELU, LeakyReLU. My hyper parameter is shown below.

- Batch size: 64
- Learning rate: 1e-3
- Epochs: 300
- Optimizer: Adam
- Loss function: cross entropy

2.1 EEGNet

EEGNet use depthwise separable convolution to replace conventional convolution. This way can reduce the computational complexity. It separates conventional convolution into depthwise and separable convolution. Depthwise convolution will learn the correlation between different signal channel. Separable convolution will learn how to combine each feature map. Figure 2

2.2 DeepConvNet

DeepConvNet is a CNN model which implement with conventional convolution. Figure 3

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

- $ReLU(x) = max(0, x)$
- $ELU(x) = max(0, x) + min(0, \alpha * (e^x - 1)$
- $LeakyReLU(x) = \begin{cases} x, & if\ x \geq 0 \\ \alpha \cdot x, & if\ x < 0 \end{cases}$

When input value is negative, ReLU will set it to zero. This way make the gradient of negative be zero which means the model can not update weights. But ELU and LeakyRelu can avoid this problem because of the design at negative input value. They design slope at negative part which make the gradient not be zero.

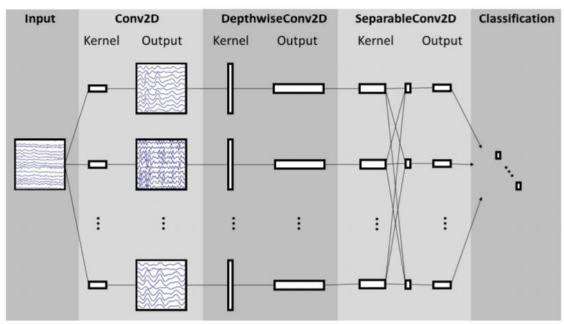


Figure 2: EEGNet architecture

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

Figure 3: DeepConvNet architecture

3 Experimental results

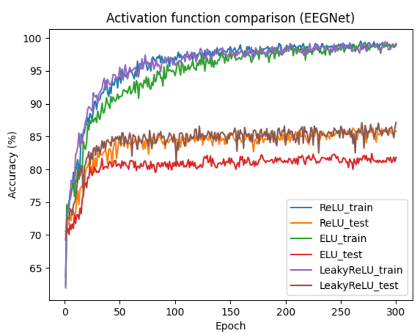
3.1 The highest testing accuracy

Both EEGNet and DeepConvNet use same hyper parameter. My hyper parameter is shown below. The following table shows the accuracy of EEGNet and DeepConvNet.

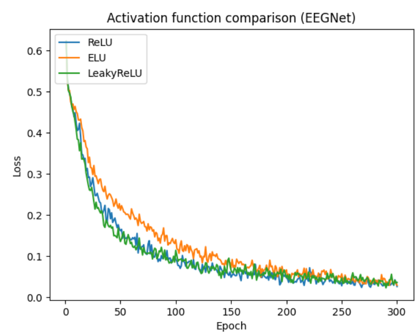
- Batch size: 64
- Learning rate: 1e-3
- Epochs: 300
- Optimizer: Adam
- Loss function: cross entropy

	ReLU	ELU	LeakyReLU
EEGNet	86.3%	82.4%	**87.1%**
DeepConvNet	82.4%	81.6%	82.6%

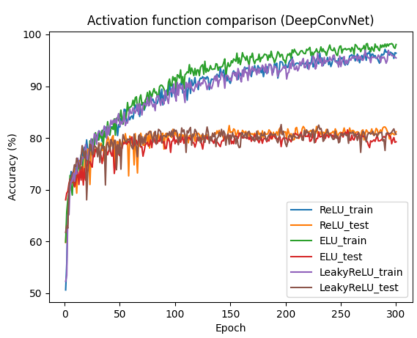
3.2 Comparison figures



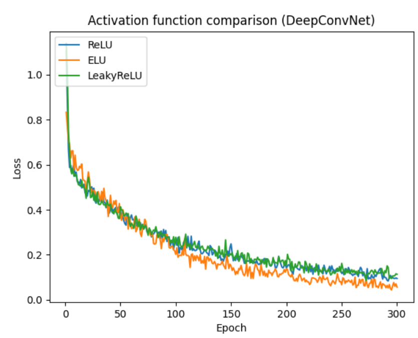
(a) EEGNet accuracy



(b) EEGNet loss



(c) DeepConvNet accuracy



(d) DeepConvNet loss

Figure 4: Comparison

4 Discussion

4.1 Activation function

In section 3, we can see LeakyReLU in both EEGNet and DeepConvNet has best accuracy, but ELU does not improve accuracy even go down the accuracy.

4.2 The number of parameters

The following table shows the number of parameters in EEGNet and DeepConvNet.

	total parameters
EEGNet	17,874
DeepConvNet	150,977

The big difference between EEGNet and DeepConvNet is because EEGNet use depthwise separable convolution. This way can efficiently reduce computation of convolution operation and reduce the number of parameters.