LAB 03 EEG Classification Report

1. Introduction

   1.main.py : Training code , can select activation function and model .

   2.test.py : Testing code , can load weight and show accuracy .

   3.EEG.py : Model of EEG , contains three activation function versions .

   4.DeepConv.py : Model of DeepConv , contains three activation function versions .

   5.Loss_show.py : Plot the loss and accuracy which are recorded in the txt file .

2. Experiment Setup

   A. Detail of model:

   Example of EEG (ELU version) :

```
EEG_ELU(
  (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, inplace=False)
  )
  (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, inplace=False)
  )
  (classify): Linear(in_features=736, out_features=2, bias=True)
)
```

   Example of Deep Convolution Network (ELU version) :

```
Deep_ELU(
  (conv1): Sequential(
    (0): Conv2d(1, 25, kernel_size=(1, 5), stride=(1, 1))
  )
  (conv2): Sequential(
    (0): Conv2d(25, 25, kernel_size=(2, 1), stride=(1, 1))
    (1): BatchNorm2d(25, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ELU(alpha=1.0)
    (3): MaxPool2d(kernel_size=(1, 2), stride=(2, 1), padding=0, dilation=1, ceil_mode=False)
    (4): Dropout(p=0.5, inplace=False)
  )
  (conv3): Sequential(
    (0): Conv2d(25, 50, kernel_size=(1, 2), 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, 5), stride=(2, 1), padding=0, dilation=1, ceil_mode=False)
    (4): Dropout(p=0.5, inplace=False)
  )
  (conv4): Sequential(
    (0): Conv2d(50, 100, kernel_size=(1, 2), 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, 5), stride=(2, 1), padding=0, dilation=1, ceil_mode=False)
    (4): Dropout(p=0.5, inplace=False)
  )
  (conv5): Sequential(
    (0): Conv2d(100, 200, kernel_size=(1, 2), 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, 5), stride=(2, 1), padding=0, dilation=1, ceil_mode=False)
    (4): Dropout(p=0.5, inplace=False)
  )
  (classify): Linear(in_features=146000, out_features=2, bias=True)
)
```

   I add the stride in the convolution layer and maxpooling layer for DeepConVNet .
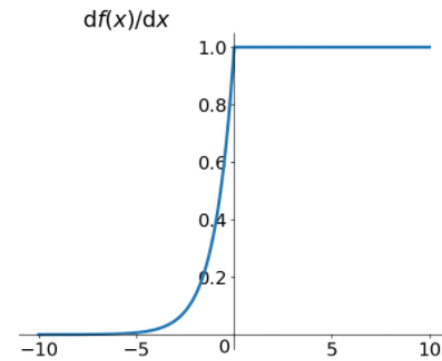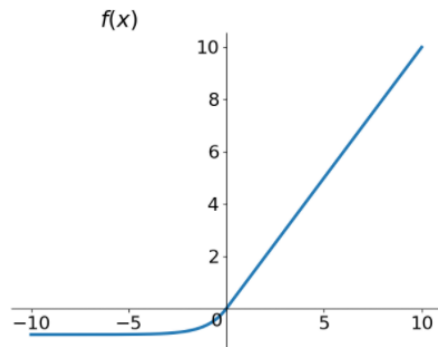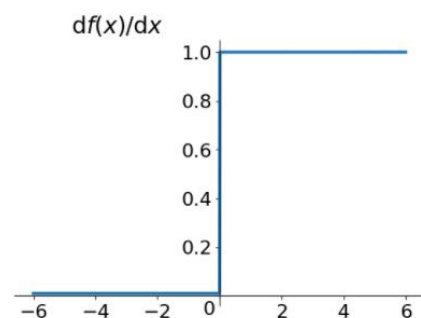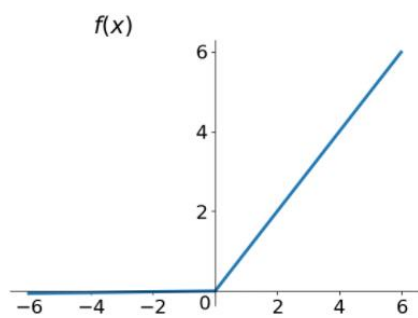
B. Activation:

ReLU :

f(X) = max(0, X)

ELU :
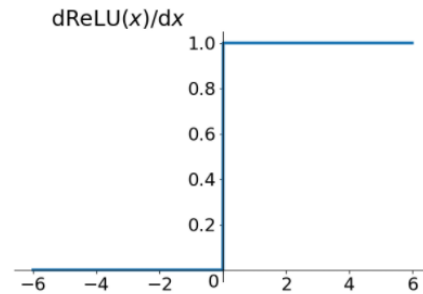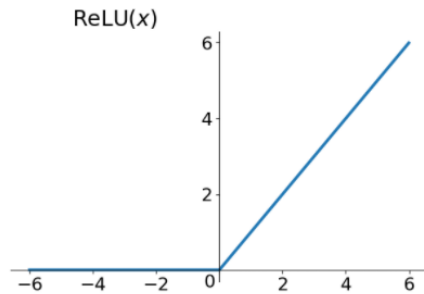
f(X) = X , if x > 0

= alpha*(e^X − 1) , otherwise
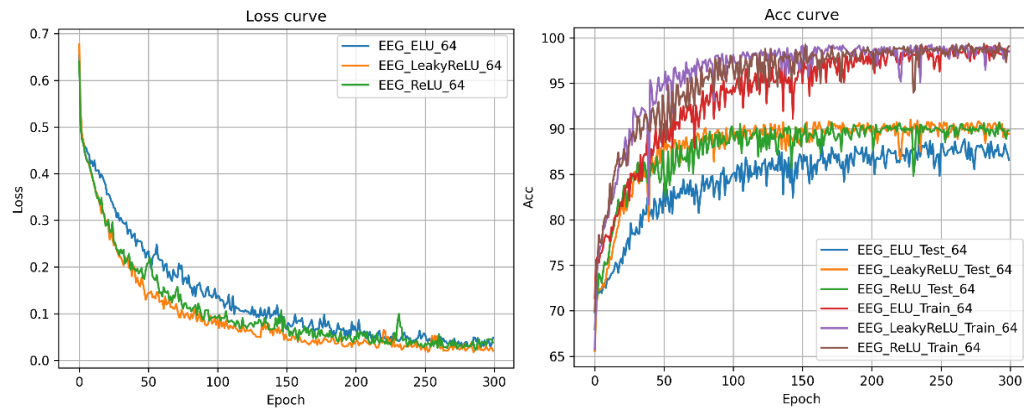

LeakyReLU :

f(X) = max(0.01*X, X)

3. Experiment Result
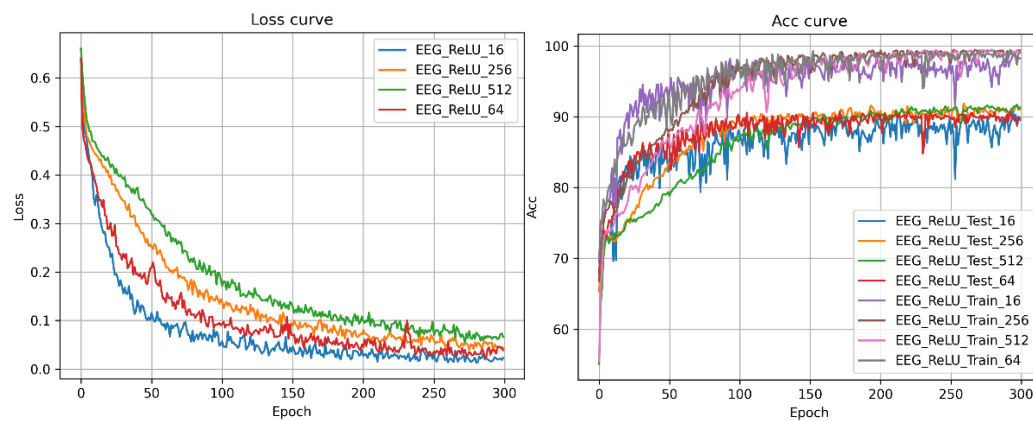
Experimental Method :

Step 1: Comparing three activation functions and choosing the best .

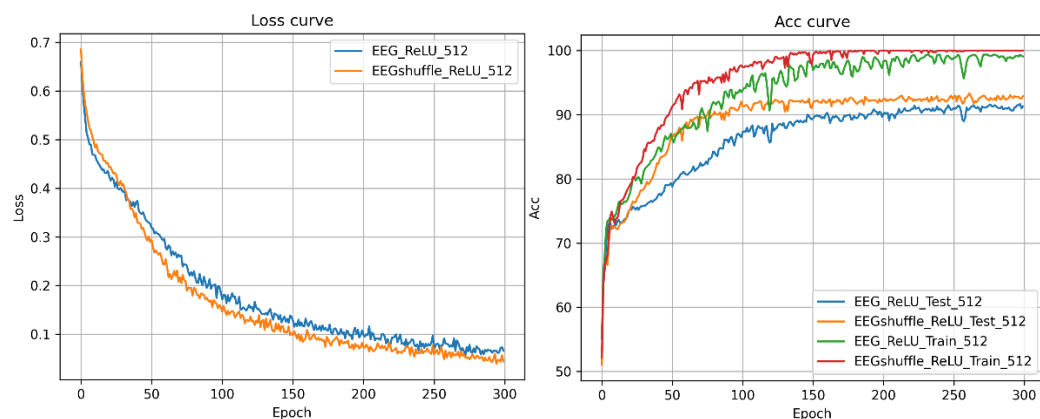Model : EEG / Batch = 64 / Epochs = 300 / Learning rate = 0.001

Then select ReLU (ReLU and LeakyReLu perform almost as well).

Step 2: Comparing several batch size and choosing the best .

Then select 512 .
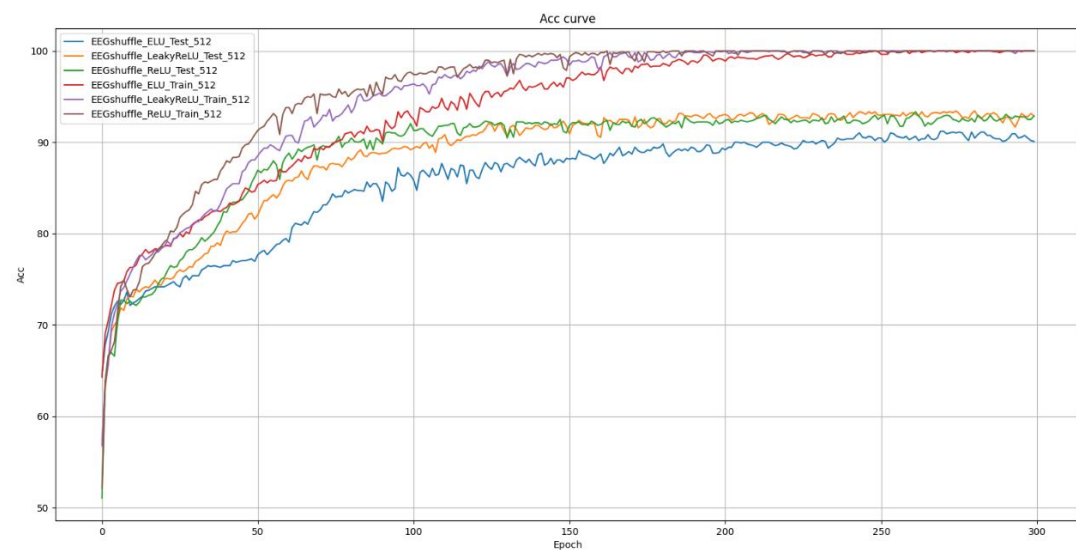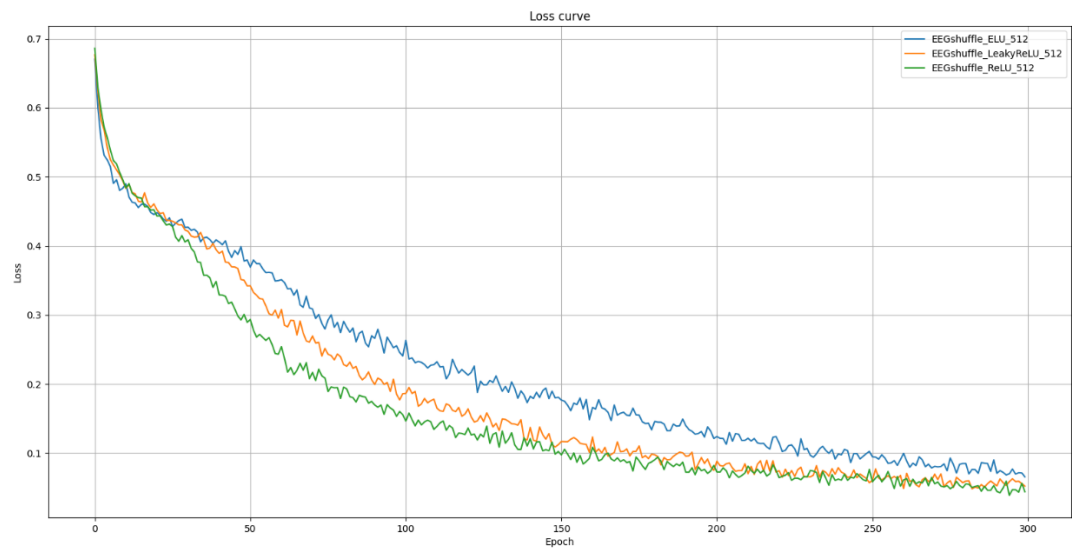
Step 3 : Comparing using shuffle or not .
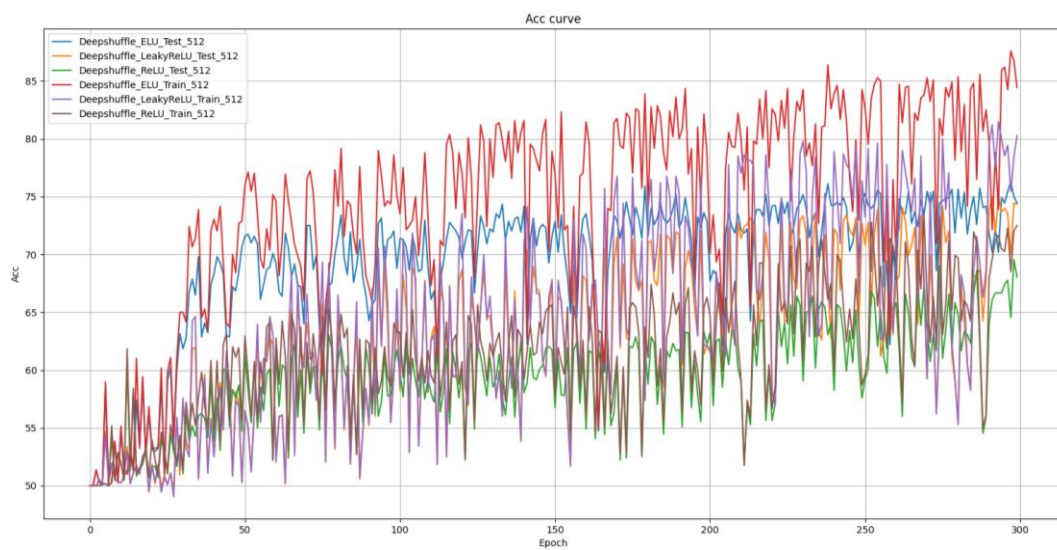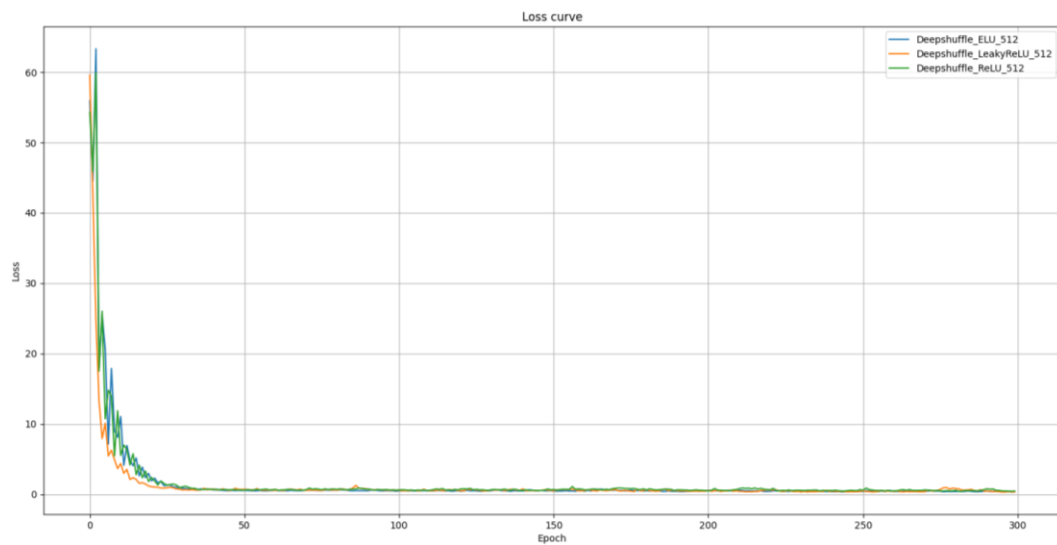

Then us shuffle .

Comparsion of EEG and DeepConV with three activation functions :

Batch = 512 / Epoch = 300 / Learning = 0.001 / Shuffle

EEG :

Loss curve

Acc curve

DeepConV :

Loss curve

Acc curve

Testing accuracy :

	ReLU	LeakyReLU	ELU
EEG	93.33%	93.42%	91.2%
DeepConV	69.53%	75.0%	76.29%