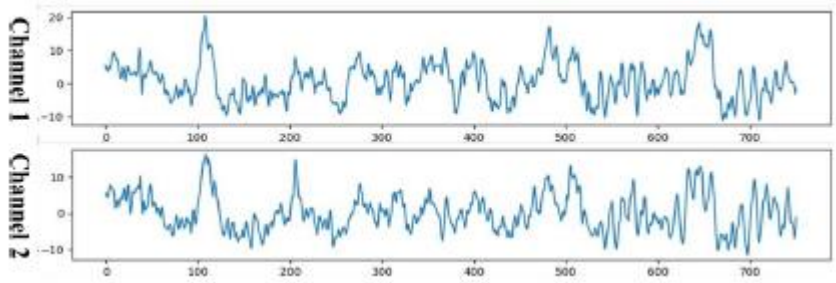
1. Introduction

   Using EEGNet and DeepConvNet to solve a classification problem, the training data is from BCI competition and has a shape of (C=1, H=2, W=750).



2. Experiment set up

   A. The detail of your model

      EEG model:

      EEGNet is a relatively simple Convolutional Neural Network (CNN) architecture optimized for the characteristics of EEG data. It combines depth-wise separable convolutions and spatial convolutions to reduce the number of parameters in the model while effectively capturing spatial information. This allows EEGNet to have lower computational costs when processing EEG data while maintaining good performance. The advantages of EEGNet lie in its ability to efficiently handle time-series EEG data and its ability to achieve good generalization with relatively fewer training samples.

```
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 is a more complex CNN architecture compared to EEGNet. It consists of multiple layers of convolutional, pooling, and batch normalization layers, followed by a fully connected layer and a softmax output layer for classification. The network is designed to automatically learn hierarchical and spatial features from EEG signals, allowing it to capture more complex patterns and representations.The advantage of

DeepConvNet lies in its ability to effectively capture spatial and temporal dependencies within EEG data due to its deeper architecture. However, it may require more computational resources and training data compared to EEGNet.

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:

$$R(z) = \begin{Bmatrix} z & z > 0 \\ 0 & z <= 0 \end{Bmatrix} \qquad R'(z) = \begin{Bmatrix} 1 & z > 0 \\ 0 & z < 0 \end{Bmatrix}$$

Advantage:

The ReLU function is computationally efficient and addresses the vanishing gradient problem in deep neural networks, making it widely used in various architectures. it speeds up the convergence during training since it allows the neuron to be active (outputting a non-zero value) for positive inputs.

Disadvantage:

it can suffer from a problem called "dying ReLU," where neurons might get stuck and stop learning if they consistently output zero.

Leaky ReLU:

$$R(z) = \begin{Bmatrix} z & z > 0 \\ \alpha z & z <= 0 \end{Bmatrix} \qquad R'(z) = \begin{Bmatrix} 1 & z > 0 \\ \alpha & z < 0 \end{Bmatrix}$$

Advantage:

Can solve dying ReLU problem. (x<0 != 0)

Disadvantage:

When x<0 that is linear, it can't be used in the complex classification.

ELU:

$$R(z) = \begin{cases} z & z > 0 \\ \alpha.(e^z - 1) & z <= 0 \end{cases} \qquad R'(z) = \begin{cases} 1 & z > 0 \\ \alpha.e^z & z < 0 \end{cases}$$

Advantage:

It introduces a slight curvature for negative inputs, which helps alleviate the vanishing gradient problem and allows for faster learning.

Disadvantage:

the main downside of ELU is its computational cost, as it involves the exponential function, which is more computationally expensive than simple linear operations like ReLU and Leaky ReLU.

3. Experiment results
   A. The highest testing accuracy

EEGNet:

```
ReLU_train acc: 97.87037037037037
ReLU_test acc: 86.48148148148148
LeakyReLU_train acc: 97.68518518518519
LeakyReLU_test acc: 87.12962962962963
ELU_train acc: 95.27777777777777
ELU_test acc: 83.42592592592592
```

DeepConvNet:

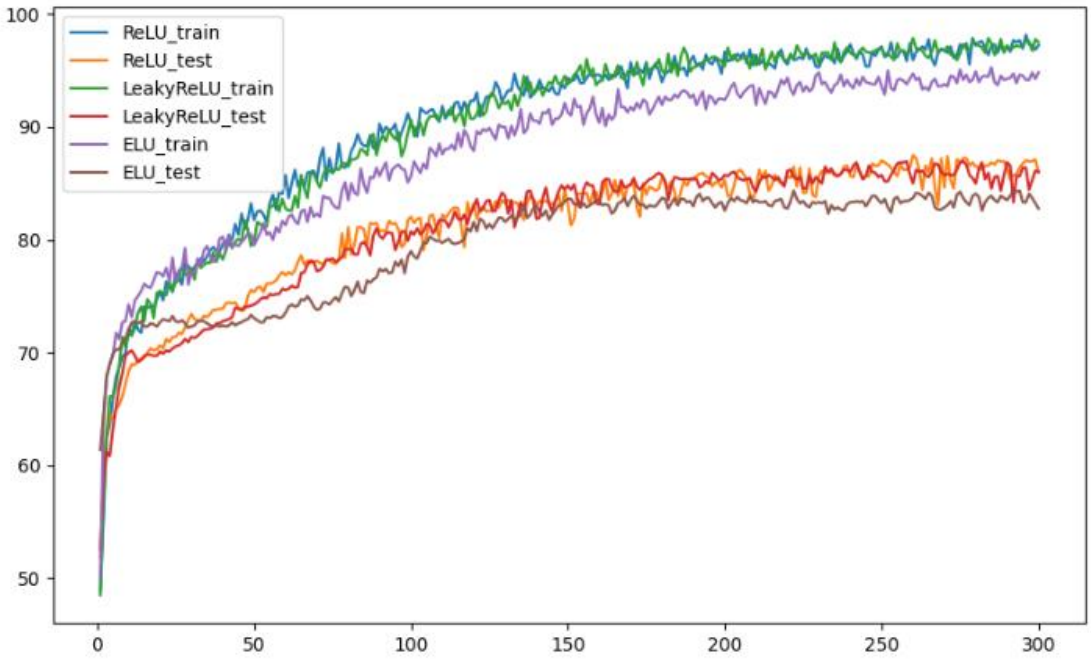
```
ReLU_train max acc: 92.03703703703704
ReLU_test max acc: 81.75925925925925
LeakyReLU_train max acc: 92.31481481481481
LeakyReLU_test max acc: 80.83333333333333
ELU_train max acc: 95.74074074074075
ELU_test max acc: 82.03703703703704
```

	ReLU	Leaky ReLU	ELU
EEGNet	86.48%	87.12%	83.43%
DeepConvNet	81.76%	80.83%	82.04%

   B. Comparison figures

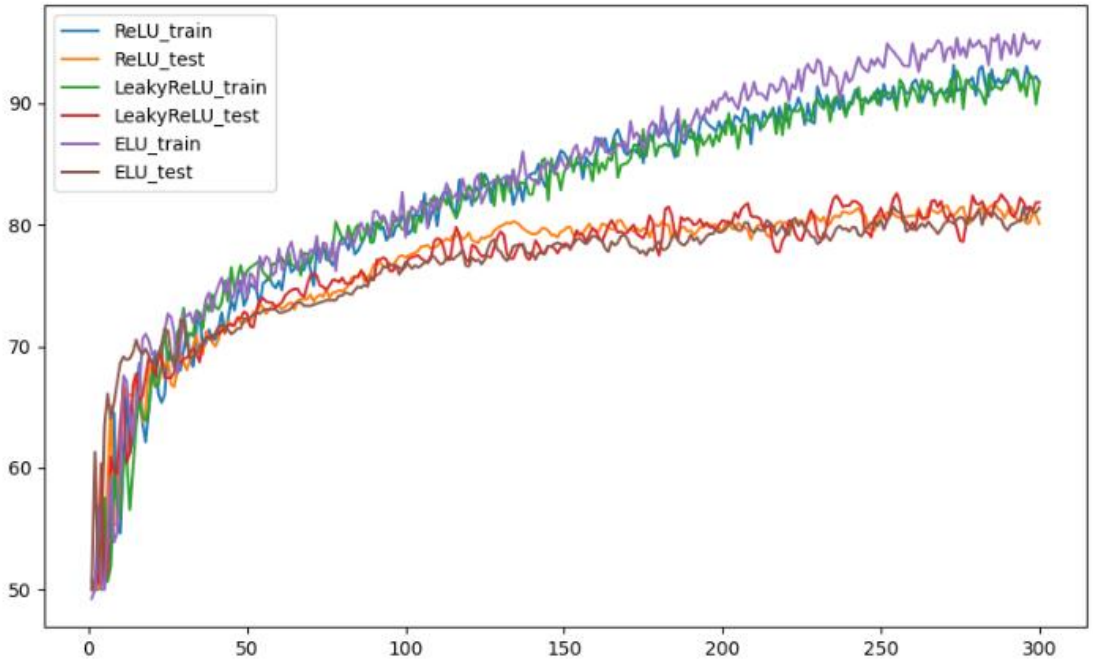
EEGNet:

epcoh 10 loss:0.0005 acc:72.1%	epcoh 10 loss:0.0005 acc:71.7%	epcoh 10 loss:0.0005 acc:74.3%
epcoh 20 loss:0.0005 acc:75.5%	epcoh 20 loss:0.0005 acc:75.5%	epcoh 20 loss:0.0004 acc:77.0%
epcoh 30 loss:0.0004 acc:77.1%	epcoh 30 loss:0.0004 acc:77.5%	epcoh 30 loss:0.0004 acc:77.1%
epcoh 40 loss:0.0004 acc:79.6%	epcoh 40 loss:0.0004 acc:78.2%	epcoh 40 loss:0.0004 acc:80.3%
epcoh 50 loss:0.0004 acc:81.8%	epcoh 50 loss:0.0004 acc:79.8%	epcoh 50 loss:0.0004 acc:79.5%
epcoh 60 loss:0.0003 acc:85.7%	epcoh 60 loss:0.0004 acc:82.8%	epcoh 60 loss:0.0004 acc:81.3%
epcoh 70 loss:0.0003 acc:85.3%	epcoh 70 loss:0.0003 acc:85.2%	epcoh 70 loss:0.0004 acc:82.3%
epcoh 80 loss:0.0003 acc:86.9%	epcoh 80 loss:0.0003 acc:86.5%	epcoh 80 loss:0.0003 acc:83.5%
epcoh 90 loss:0.0002 acc:90.3%	epcoh 90 loss:0.0003 acc:88.6%	epcoh 90 loss:0.0003 acc:85.6%
epcoh100 loss:0.0002 acc:90.3%	epcoh100 loss:0.0002 acc:90.6%	epcoh100 loss:0.0003 acc:85.6%
epcoh110 loss:0.0002 acc:91.9%	epcoh110 loss:0.0002 acc:91.3%	epcoh110 loss:0.0003 acc:87.6%
epcoh120 loss:0.0002 acc:92.3%	epcoh120 loss:0.0002 acc:90.7%	epcoh120 loss:0.0003 acc:87.9%
epcoh130 loss:0.0002 acc:92.8%	epcoh130 loss:0.0002 acc:92.0%	epcoh130 loss:0.0002 acc:90.0%
epcoh140 loss:0.0002 acc:93.4%	epcoh140 loss:0.0002 acc:93.2%	epcoh140 loss:0.0002 acc:90.6%
epcoh150 loss:0.0002 acc:94.4%	epcoh150 loss:0.0002 acc:93.8%	epcoh150 loss:0.0002 acc:91.3%
epcoh160 loss:0.0001 acc:94.6%	epcoh160 loss:0.0002 acc:94.1%	epcoh160 loss:0.0002 acc:90.4%
epcoh170 loss:0.0001 acc:95.7%	epcoh170 loss:0.0001 acc:95.0%	epcoh170 loss:0.0002 acc:91.8%
epcoh180 loss:0.0001 acc:94.0%	epcoh180 loss:0.0001 acc:95.0%	epcoh180 loss:0.0002 acc:91.9%
epcoh190 loss:0.0001 acc:96.3%	epcoh190 loss:0.0001 acc:95.3%	epcoh190 loss:0.0002 acc:92.9%
epcoh200 loss:0.0001 acc:96.7%	epcoh200 loss:0.0001 acc:95.4%	epcoh200 loss:0.0002 acc:92.4%
epcoh210 loss:0.0001 acc:96.4%	epcoh210 loss:0.0001 acc:95.5%	epcoh210 loss:0.0002 acc:91.9%
epcoh220 loss:0.0001 acc:95.6%	epcoh220 loss:0.0001 acc:95.9%	epcoh220 loss:0.0002 acc:94.4%
epcoh230 loss:0.0001 acc:95.3%	epcoh230 loss:0.0001 acc:95.6%	epcoh230 loss:0.0002 acc:94.8%
epcoh240 loss:0.0001 acc:97.3%	epcoh240 loss:0.0001 acc:96.6%	epcoh240 loss:0.0002 acc:93.7%
epcoh250 loss:0.0001 acc:97.1%	epcoh250 loss:0.0001 acc:96.9%	epcoh250 loss:0.0002 acc:93.6%
epcoh260 loss:0.0001 acc:95.9%	epcoh260 loss:0.0001 acc:97.9%	epcoh260 loss:0.0001 acc:94.7%
epcoh270 loss:0.0001 acc:96.8%	epcoh270 loss:0.0001 acc:96.8%	epcoh270 loss:0.0002 acc:93.8%
epcoh280 loss:0.0001 acc:97.2%	epcoh280 loss:0.0001 acc:97.4%	epcoh280 loss:0.0002 acc:94.9%
epcoh290 loss:0.0001 acc:97.1%	epcoh290 loss:0.0001 acc:97.2%	epcoh290 loss:0.0001 acc:94.3%
epcoh300 loss:0.0001 acc:97.2%	epcoh300 loss:0.0001 acc:97.5%	epcoh300 loss:0.0001 acc:94.8%



DeepConvNet:

epcoh 10 loss:0.0009 acc:54.6%	epcoh 10 loss:0.0009 acc:56.0%	epcoh 10 loss:0.0010 acc:59.7%
epcoh 20 loss:0.0006 acc:67.9%	epcoh 20 loss:0.0005 acc:68.1%	epcoh 20 loss:0.0007 acc:69.4%
epcoh 30 loss:0.0005 acc:69.7%	epcoh 30 loss:0.0005 acc:73.1%	epcoh 30 loss:0.0005 acc:72.7%
epcoh 40 loss:0.0005 acc:70.6%	epcoh 40 loss:0.0005 acc:73.2%	epcoh 40 loss:0.0005 acc:74.3%
epcoh 50 loss:0.0005 acc:74.0%	epcoh 50 loss:0.0005 acc:76.2%	epcoh 50 loss:0.0004 acc:74.0%
epcoh 60 loss:0.0005 acc:75.4%	epcoh 60 loss:0.0004 acc:77.0%	epcoh 60 loss:0.0004 acc:78.1%
epcoh 70 loss:0.0004 acc:76.4%	epcoh 70 loss:0.0004 acc:76.9%	epcoh 70 loss:0.0004 acc:78.0%
epcoh 80 loss:0.0004 acc:79.0%	epcoh 80 loss:0.0004 acc:78.9%	epcoh 80 loss:0.0004 acc:78.4%
epcoh 90 loss:0.0004 acc:80.2%	epcoh 90 loss:0.0004 acc:78.6%	epcoh 90 loss:0.0004 acc:81.1%
epcoh100 loss:0.0004 acc:80.2%	epcoh100 loss:0.0004 acc:79.6%	epcoh100 loss:0.0004 acc:79.0%
epcoh110 loss:0.0004 acc:79.9%	epcoh110 loss:0.0004 acc:81.2%	epcoh110 loss:0.0004 acc:82.2%
epcoh120 loss:0.0003 acc:82.5%	epcoh120 loss:0.0003 acc:83.2%	epcoh120 loss:0.0004 acc:82.4%
epcoh130 loss:0.0003 acc:83.7%	epcoh130 loss:0.0003 acc:83.3%	epcoh130 loss:0.0003 acc:83.6%
epcoh140 loss:0.0003 acc:84.3%	epcoh140 loss:0.0003 acc:82.5%	epcoh140 loss:0.0003 acc:83.7%
epcoh150 loss:0.0003 acc:85.9%	epcoh150 loss:0.0003 acc:84.3%	epcoh150 loss:0.0003 acc:84.9%
epcoh160 loss:0.0003 acc:85.2%	epcoh160 loss:0.0003 acc:85.8%	epcoh160 loss:0.0003 acc:87.2%
epcoh170 loss:0.0003 acc:85.7%	epcoh170 loss:0.0003 acc:85.1%	epcoh170 loss:0.0003 acc:86.0%
epcoh180 loss:0.0003 acc:87.3%	epcoh180 loss:0.0003 acc:86.5%	epcoh180 loss:0.0003 acc:87.2%
epcoh190 loss:0.0003 acc:88.0%	epcoh190 loss:0.0003 acc:87.1%	epcoh190 loss:0.0002 acc:88.1%
epcoh200 loss:0.0003 acc:87.7%	epcoh200 loss:0.0003 acc:88.1%	epcoh200 loss:0.0002 acc:90.4%
epcoh210 loss:0.0003 acc:89.1%	epcoh210 loss:0.0003 acc:88.8%	epcoh210 loss:0.0002 acc:91.8%
epcoh220 loss:0.0002 acc:88.5%	epcoh220 loss:0.0002 acc:90.0%	epcoh220 loss:0.0002 acc:91.8%
epcoh230 loss:0.0002 acc:91.4%	epcoh230 loss:0.0002 acc:90.4%	epcoh230 loss:0.0002 acc:93.6%
epcoh240 loss:0.0002 acc:89.7%	epcoh240 loss:0.0002 acc:90.1%	epcoh240 loss:0.0002 acc:91.8%
epcoh250 loss:0.0002 acc:89.8%	epcoh250 loss:0.0002 acc:91.0%	epcoh250 loss:0.0002 acc:93.2%
epcoh260 loss:0.0002 acc:91.7%	epcoh260 loss:0.0002 acc:91.4%	epcoh260 loss:0.0001 acc:94.9%
epcoh270 loss:0.0002 acc:90.6%	epcoh270 loss:0.0002 acc:90.6%	epcoh270 loss:0.0002 acc:93.4%
epcoh280 loss:0.0002 acc:90.6%	epcoh280 loss:0.0002 acc:91.2%	epcoh280 loss:0.0001 acc:94.8%
epcoh290 loss:0.0002 acc:91.7%	epcoh290 loss:0.0002 acc:92.8%	epcoh290 loss:0.0001 acc:94.8%
epcoh300 loss:0.0002 acc:91.8%	epcoh300 loss:0.0002 acc:91.8%	epcoh300 loss:0.0001 acc:95.1%

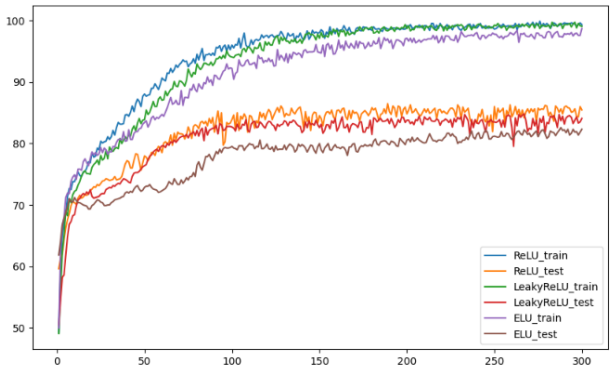


4. Discussion

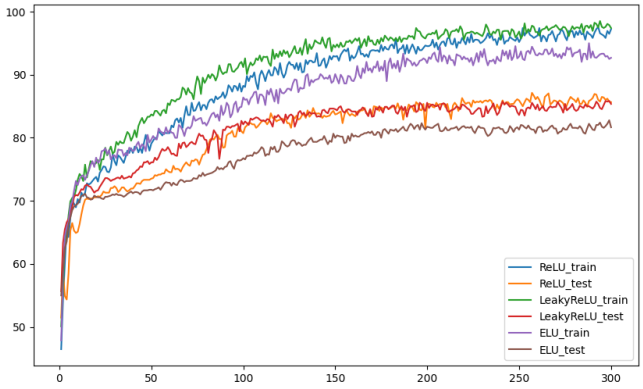
A. Anything you want to share

Different Dropout: Dropout is a regularization technique commonly used in neural networks to prevent overfitting. It works by randomly setting a fraction of the neurons to zero during training, effectively dropping them out of the network for that particular forward and backward pass. This helps the network to become more robust and less reliant on specific neurons, reducing the risk of overfitting the training data.

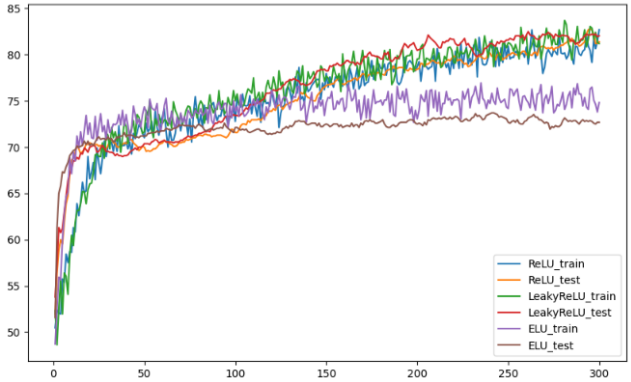
0.1



0.25

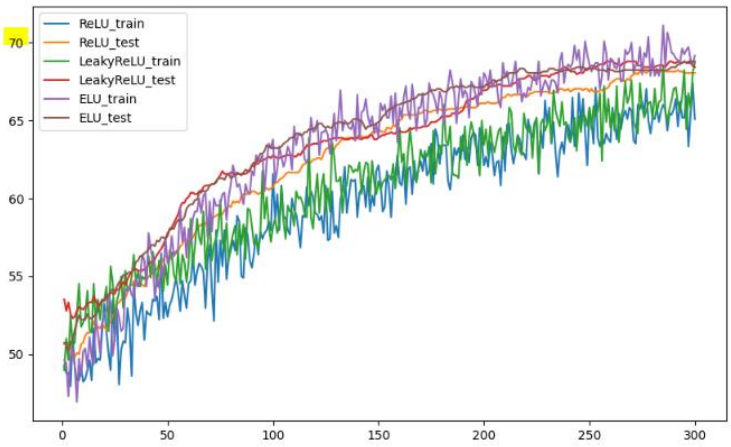


0.8

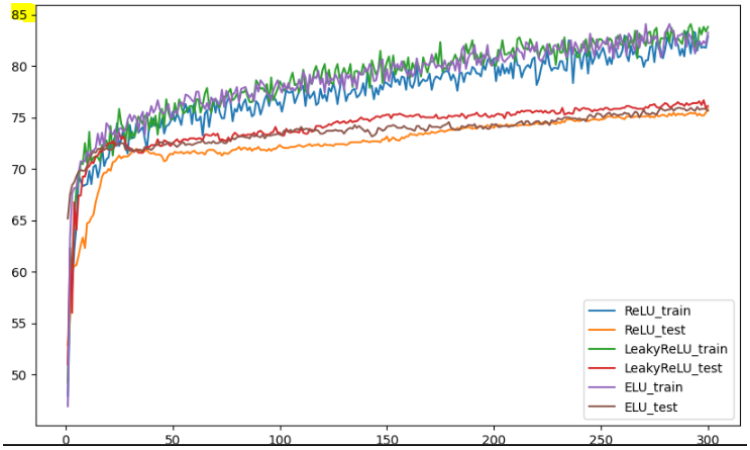


Different optimizer:

SGD:



Adagrad:



RMSprop: