Summer 2023 DLP Lab2

312552017 董家鳴

1. Introduction (10%)

In this lab, we need to implement two classification models, EEG and DeepConvNet by using PyTorch to classify 2 classes, left hand and right hand given the motor imagery from BCI Competition III. The data set will have two channels as Figure 1 shown below.

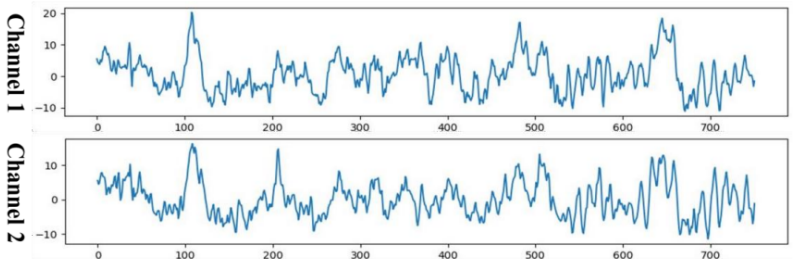

Figure 1: BCI Competition III dataset

The requirements in this lab are listed below:

1. Implement the EEGNet, DeepConvNet with three kinds of activation function including ReLU, Leaky ReLU, ELU.
2. In the experiment results, you have to show the highest accuracy (not loss) of two architectures with three kinds of activation functions.
3. Visualize the accuracy trend, you need to plot each epoch's accuracy (not loss) during the training phase and testing phase.

2. Experiment set up (30%)

A. The detail of your model

The following figure 2 below shows the EEGNet model. Let's break down the model and explain its components except for activation which will be talked about later.

- **First Convolutional Layer (first_conv):** The first layer will perform a 2D convolution over the input signal, with 16 output channels and a kernel size of (1, 51).
- **Depthwise Convolutional Layer (depthwise_conv):** Instead of performing a single convolution operation on all the input channels, In this layer, a depthwise convolu-

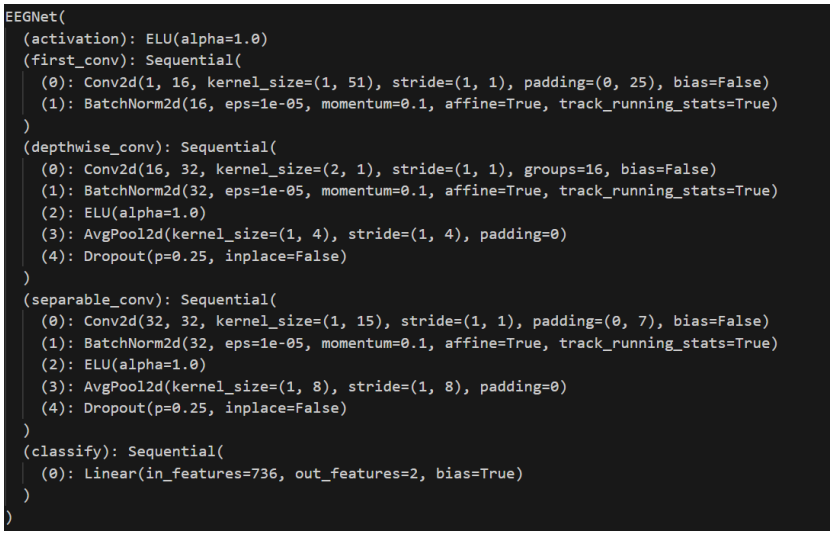

Figure 2: EEGNet

tion applies a different convolution to each input channel separately. This drastically reduces the number of parameters and computations.

- **Separable Convolutional Layer (Separable_conv):** The separable convolution layer is designed to learn temporal filters. This layer captures temporal dependencies in the EEG signals.

The following figure 3 below shows the DeepConvNet model. Let's break down the model and explain its components except for activation which will be talked about later. There are four convolutional blocks (conv_block1, conv_block2, conv_block3, conv_block4) with each containing:

- **Conv2d:** This is a 2-dimensional convolutional layer. Each Conv2d layer has an increasing number of output channels (25, 50, 100, 200), which means it's extracting more features from the input as it passes through the layers.
- **BatchNorm2d:** Batch normalization is a technique for improving the speed, performance, and stability of neural networks. It standardizes the inputs to a layer for each mini-batch, which has the effect of stabilizing the learning process and dramatically reducing the number of training epochs required to train deep networks.
- **MaxPool2d:** Max pooling is a procedure that takes the maximum value over the window defined by the kernel size for each dimension along the features axis whose window is shifted by strides in each dimension.
- **Dropout:** Dropout is a regularization technique for reducing overfitting in neural networks. It works by randomly setting the outgoing edges of hidden units (neurons) to 0 at each update of the training phase. Here, a dropout rate of 0.5 is used, meaning approximately half of the input units to a dropout layer are dropped.
- **Fully Connected Layer:** Its role is to perform classification on the features extracted by the convolutions. The input to these layers is flattened into a single dimension. The

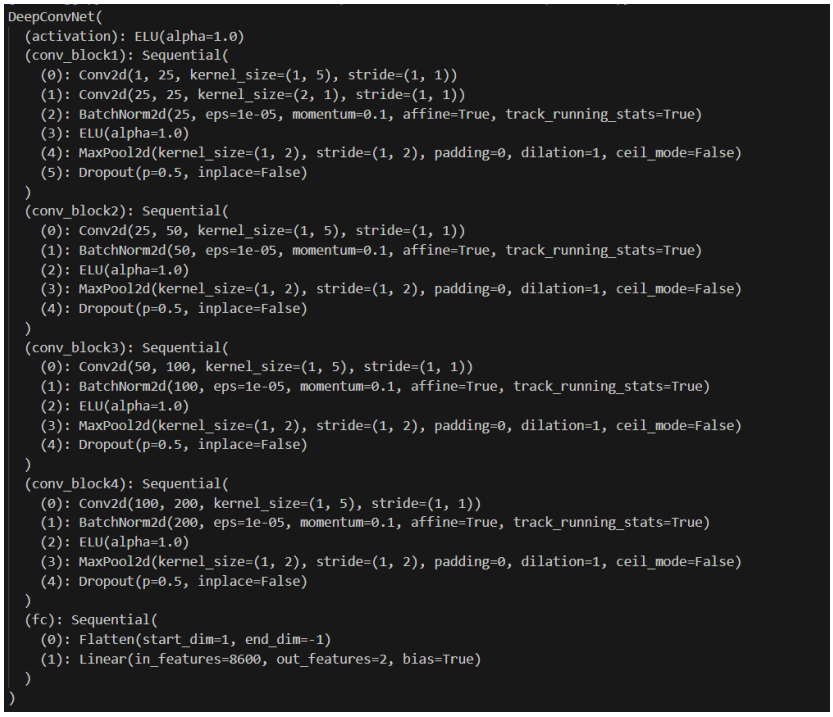

Figure 3: DeepConvNet

final output has 2 features, which suggests that the network is designed to classify inputs into one of two classes.

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

- **ReLU (Rectified Linear Unit)**: Relu is very fast and doesn't have the problem of gradient vanishing, but because it's always 0 when it's less than 0, the gradient is also 0 so loss can't be optimized.

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

- **Leaky ReLU**: Because of the previous drawback, Leaky Relu was invented to solve the problem that everything less than zero is zero. But there is still a drawback that it is not possible to do gradient descent when x=0.

$$LeakyReLU(x) = \max(\alpha \cdot x, x)$$

- **ELU**: ELU is a modification of the above two types of Relu, which does not have its disadvantages, but requires more computational resources because of its complexity.

$$ELU(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha(e^x - 1) & \text{if } x \leq 0 \end{cases}$$

3. Experimental results (15%)

A. The highest testing accuracy is **87.50%** in my experiment as the table 1 shows. The α value of LeakyReLU is 0.01, and the α value of ELU is 1.

	LeakyReLU	**ELU**	**ReLU**
EEGNet	**87.50%**	83.43%	86.39%
DeepConvNet	81.76%	81.76%	81.85%

Table 1: Accuracy of two architectures with three kinds of activation functions

Batch size	Epochs	Learning rate	Optimizer	Loss function
64	300	1e-3	Adam	CrossEntropy

Table 2: Hyperparameters of the model

B. Comparison figures are depicted in figures 4 and 5.

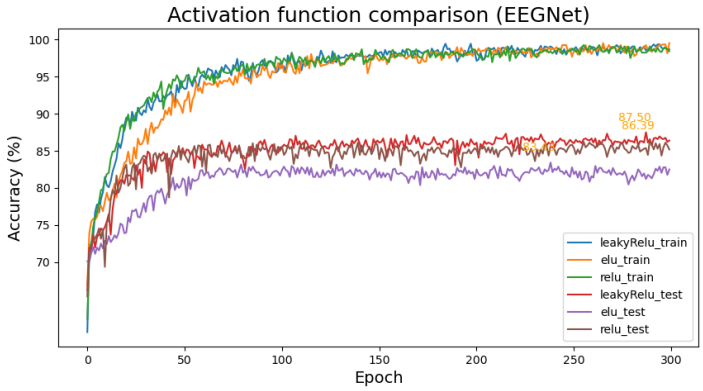

Figure 4: Comparison figures of EEGNet

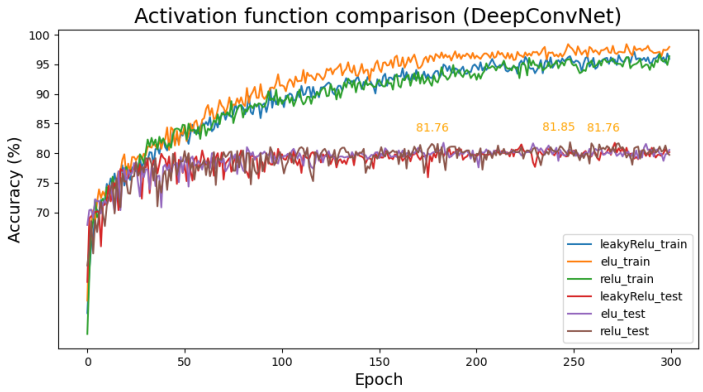

Figure 5: Comparison figures of DeepConvNet

4. Discussion (45%)

In this part, I want to discuss why the EEGNet performs better than DeepConvNet.

A. Design of DeepConvNet

- According to 2.2.2. Comparison with existing CNN approaches [1]. The DeepConvNet architecture was designed to be a general-purpose architecture not restricted to specific feature types.
- Table 3 shows the number of parameters compared to EEGNet. The architecture was not designed to extract frequency features, so the performance was lower when frequency power is the dominant feature.

	Total Parameters
DeepConvNet	150 977
EEGNet	17 874

Table 3: Total Parameters in Different Models

B. Design of EEGNet

- The authors of EEGNet have focused on feature explainability, which is crucial for validating that the classification performance is driven by relevant features and not by noise or artifacts in the data.
- According to the 2.4. EEGNet feature explainability [1]. The EEGNet uses depth-wise and separable convolutions to extract features from EEG signals by summarizing averaged outputs of hidden unit activations, which limits the connectivity of the convolutional layers, so it is possible to interpret the temporal convolution as narrow-band frequency filters.

C. Summary of the difference between two models

- Because DeepConvNet has too many model parameters, there is no way to capture a specific frequency (the frequency of body movements), so the performance is not good.
- EEGNet summarizes the averaged outputs of hidden unit activations, which provide additional insights into the spatial localization of narrow-band frequency activity.

References

[1] Vernon J Lawhern et al. "EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces". In: *Journal of Neural Engineering* 15.5 (July 2018), p. 056013. DOI: 10.1088/1741-2552/aace8c. URL: https://doi.org/10.1088%2F1741-2552%2Faace8c.