Bidirectional Resnet Transfer Learning for EEG Classification

Yuhang Liu[1]

Research School of Computer Science,
Australian National University
u6920250@anu.edu.au

Abstract. Neural Network technique has been applied in many areas like pattern recognition, bioinformatic etc. and performs well. And the EEG(electroencephalograph) data is an important part of bioinformatic research, especially in the popular BCI(Brain Computer Interface) Research. In this paper, I proposed a new transfer learning method for EEG classification and compared it with 3 different models' performance on classifying the UCI EEG Dataset. Further, I focus on the comparison of the training speed, and develop that Bidirectional method can reduce training epochs to reach convergence and have ideal performance at the same time.

Keywords: EEG Classification · Resnet · Transfer Learning Convolutional Neural Network

1 Introduction

BCI(Brain Computer Interface) has widely attract people's attention these days, especially after Elon Musk's BCI company *Neuralink* Published a video of animals playing games by his brain's signal. BCI could bring an evolutionary change to HCI(Human Computer Interaction) and EEG(electroencephalography) play an significant role in BCI research. Classifying EEG by Neural Network can not only help doctor diagnosis more efficiently but also give us better understanding of how the neural works and the connections between the brain and signal.

In this paper, I use the ImageNet pretrained ResNet to extract the feature and replace the previous Fully Connected Layer in by a Bidirectional Fully Connected Layer. I test my model's performance on classifying the UCI EEG dataset[2] which has 122 subjects (77 diagnosed with alcoholism) and has 64 sensors to capture the brain signal in 3 different frequency bands.

After doing the feature engineering on the dataset, I choose within-subject training for a more robust model. I evaluate the performance of the Bidirectional Resnet by AUC(Area Under Curve) and Accuracy. I also compare the performance of Bidirectional Resnet with other three nerual network(fully connected neural network, bidirectional neural network and bidirectional convlutional neural network) and also record the training time of these models to check the efficiency of the model training.

2 Method

2.1 Dataset

UCI EEG dataset contains 122 subjects(people). Up to 120 trials are performed on each subject, in each trial, they asked the subject to distinguish the difference between two pictures selected from the Snodgrass and Vanderwart picture set[9] and according to the performance, they can label subject **'alcoholism'** and if a subject are diagnosed with alcoholism in one trial, all trials will be labeled as **'alcoholism'**.

And the training of these dataset have two major methods: Within Subject and Cross Subject. Within way first group up the dataset by the subject id and do the train test split in each subject group, see Figure 1 while cross way do the train test split on subject dimension, see Figure 2.In this paper, I choose WITHIN TRAINING and split the dataset in ratio: Train:Validation:Test = 7:2:1

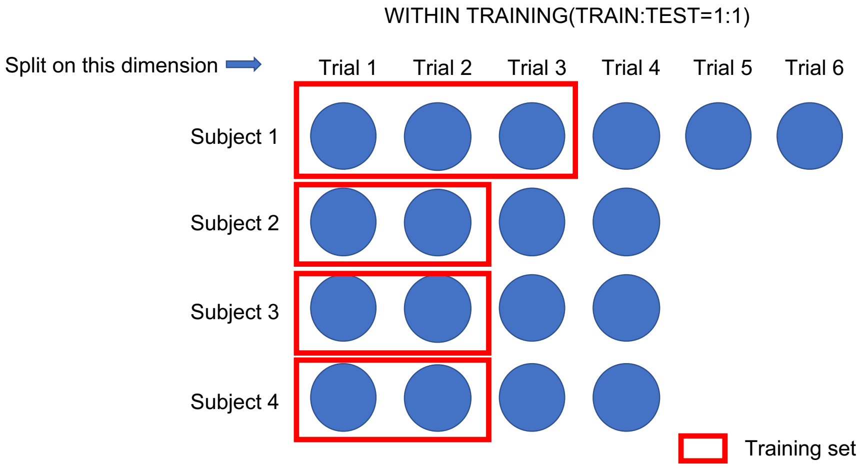

Fig. 1. Within Subject

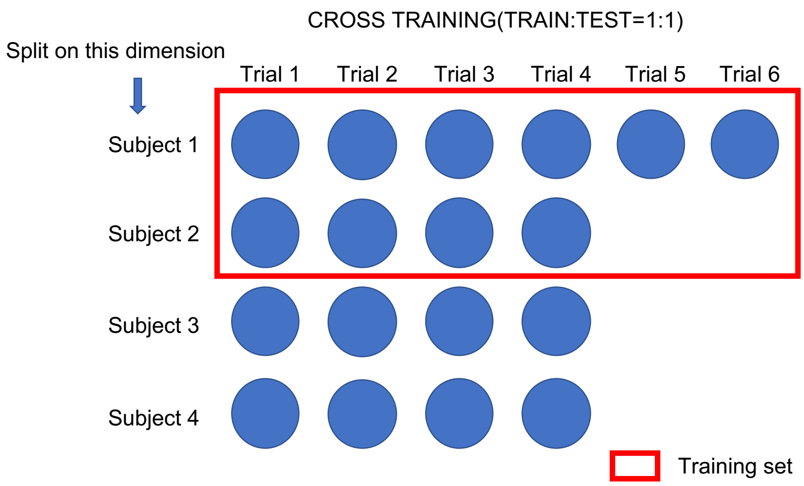

Fig. 2. Cross Subject

The feature are captured by 64 sensor in three different frequency bands, the final out put are concated into a one-dimension vector, i.e. 192 * 1 vector. Moreover, the dataset provides the corresponding stimulus label. After checking the distribution of stimulus type and alcoholism, see Figure 3, I dispose the stimulus type feature due to the dramatically unbalanced data.

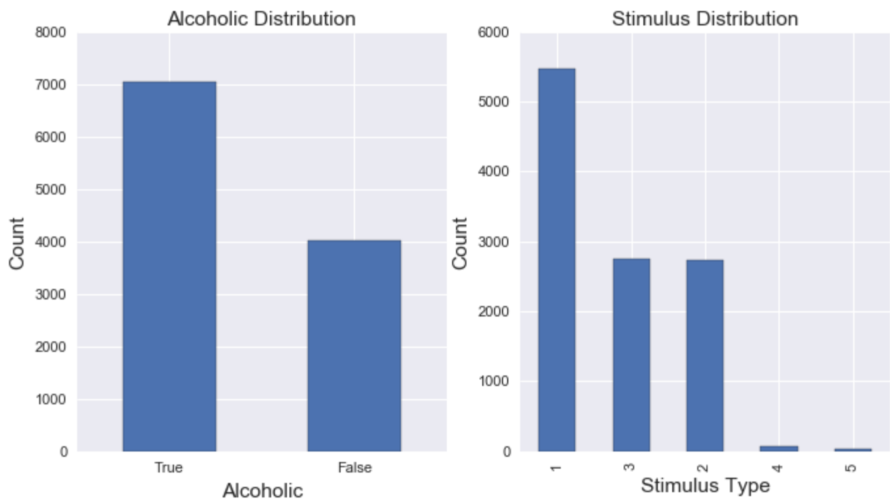

Fig. 3. Within Subject

After the feature engineer, I randomly choose 9 of EEG data and visualize them, found that some colors have specific location(green always located in the left-bottom of the image), which means the location and the value are highly correlated in EEG dataset(because of the fixed location sensor). So extracting some semantic information from part of the image might be helpful for obtaining higher accuracy. ResNet18[3] (pretrained on ImageNet[1]) might be a good choice for feature extraction task. Deep Resdiual Network(ResNet) is a robust model that can avoid gradient explosion and vanishing gradient in deep Neural Network. After training on the ImageNet, the model is able to extract both the detailed information(in shallow layer, smaller reception fields) and the semantic information(deep layer, larger reception field). However, the deep neural network cost lots of computation power to train, we decide to use the transfer learning to do the feature extraction.

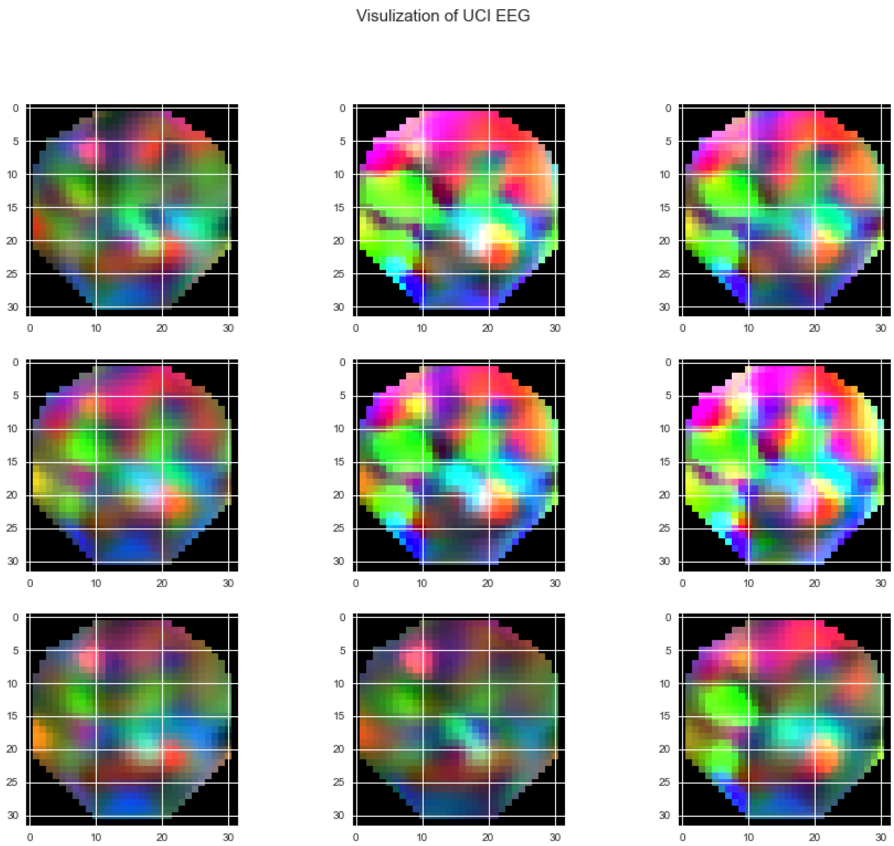

Fig. 4. EEG Data Visualization

2.2 Training Tricks

Transfer Learning Transfer learning reused the previous trained result as the starting point for a model on a second task. In this paper, I first visualize the EEG dataset shiyaLiu's method and then input the visualized images to the Imagenet Pretrained Resnet-18 model and replace the last fully connected layer to the bidirectional fully connected layer. During backpropogation, the parameters of ResNet is frozen. i.e. only the Bidirectional part do the BP, which shorten the training time a lot

Hyper Parameters The size of dataset is relatively small, we can input the whole dataset to model and do Back Propagation. I use SGD with base learning rate 0.02 to optimize this model and use Cross Entropy Loss as the loss function.(In Pytorch, the function *Cross Entropy* Loss contains the softmax and the Cross Entropy)

Learning Rate Decay Learning rate decay (lrDecay) is a de factor technique for training modern neural networks,where we adopt an initially large learning rate and then decay it by a certain factor after pre-defined epochs[10]
I use the following formula to perform the lrDecay:

$$currentLearningRate = baseLearningRate * (1 - currentEpoch/allEpochs)^{decayPower}$$

I use the base learning rate 0.02 and decayPower 0.9 to train 10000 epochs, the change of learning rate with epochs are shown in Figure 4.

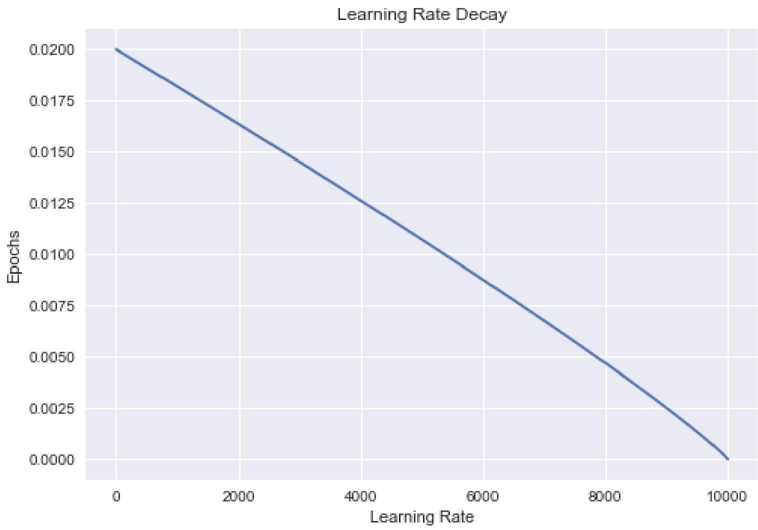

Fig. 5. Learning Rate Decay

Using the relatively high learning rate at start can help the model converge faster.As the training proceed, the gradient is approaching the minimum, which means the gradient becomes more and more sensitive, decreasing the learning rate can ensure the gradient does not miss the global minimum.

2.3 Model Structure

The following networks are all implemented by Pytorch. For each models, I trained 10000 epochs and I record and plot the training time, training loss, validation AUC and validation accuracy to compare models, and I store the model's parameters with best AUC as the final model to predict on test set. Please check the plot for detailed model structure.

Transfer Learning Bidirectional Nerual Network The structure of this network are shown in Figure 4

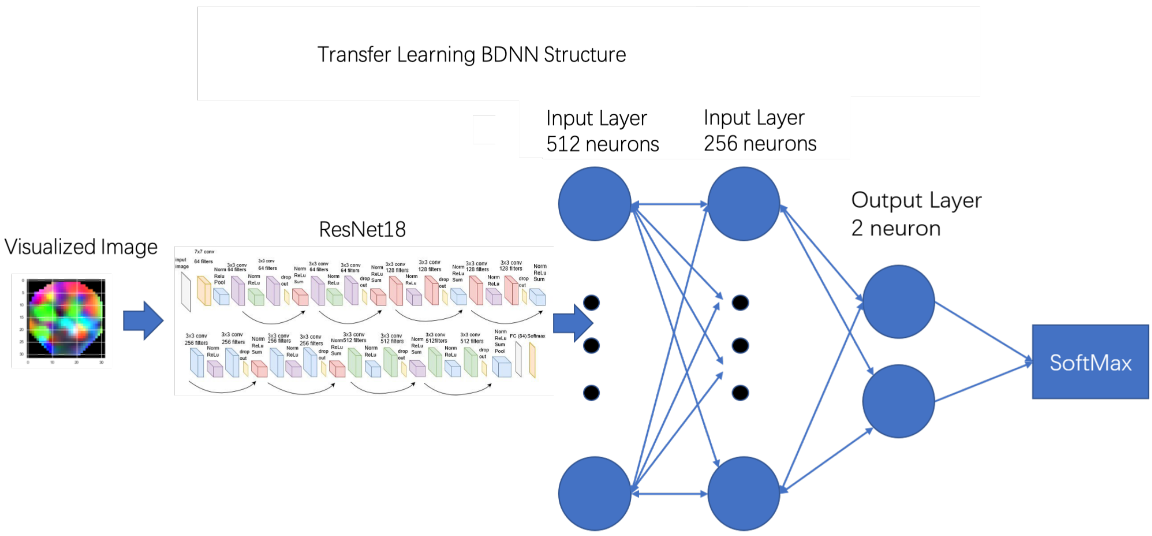

Fig. 6. Transfer Learning Network Structure

Because the memory limiit of my computer, the training process are killed every 300 epoches, I need to load the most recent check-point and activate the training script 3 times to make the model convergence, which cause the strange peak of the loss in Fig.7

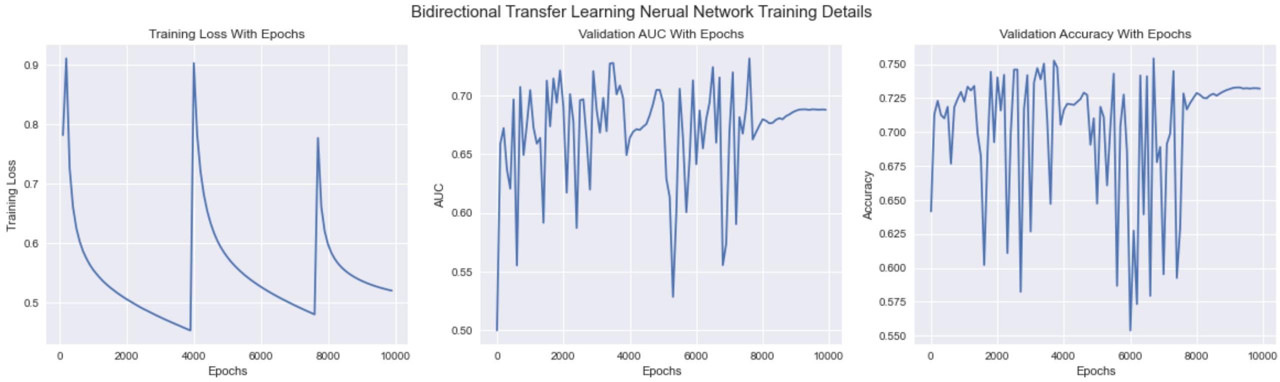

Fig. 7. Transfer Learning Network Structure

Fully Connected Nerual Network(Baseline) The structure of this network are shown in Figure 8

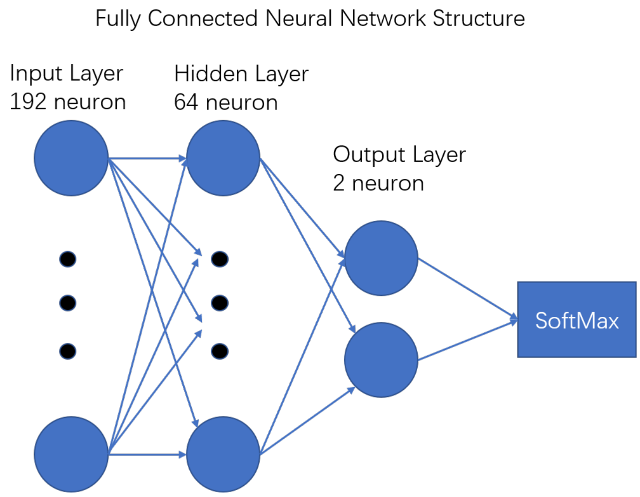

Fig. 8. Fully Connected Network Structure

From Figure 6, we can see the simple 2-Layer Fully Connected reach convergence after about 6000 epochs and get a stable performance on validation set.

Fig. 9. Fully Connected Network Training

Bidirectional Nerual Network To implement this technique, I create 2 network(original one and reversed one), the original one do the BP(Back Propagation) first and assign the updated weight to the reversed network, then the reversed net do the BP and assign the updated value to the original net. Models only share weights not bias[6], so I only assign the weight after each BP. The structure are shown as followed, the only difference between this model and the previous one is in one epoch, it does BP twice(original and reversed). I use MSE loss as the reversed BP's loss function and SGD with base learning rate 0.01 as the optimizer

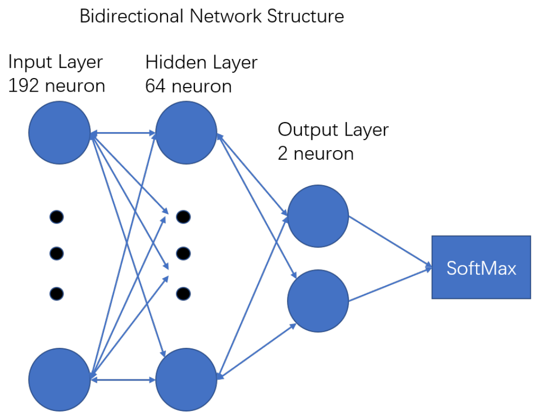

Fig. 10. Bidirectional Nerual Network Structure

In the training process, I developed that this bidirectional model's amplitude is much larger than simple Fully Connected Nerual Net and it reach convergence at about 5000 epochs, which is earlier than FC NN 1000 epochs. Moreover, the elbow point appear earlier than previous one.

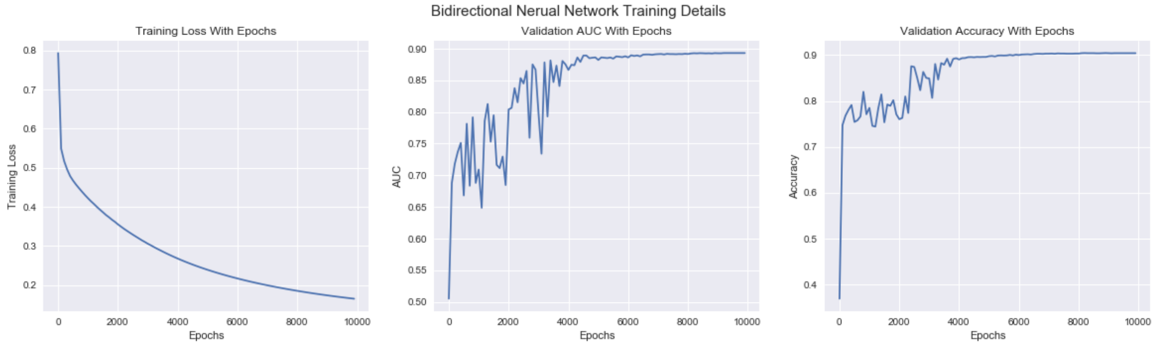

Fig. 11. Bidirectional Nerual Network Training

Bidirectional Convolutional Nerual Network Convolutional layer force the extraction of local features[4].Due to the nearby sensors might have similar signals , the convolution layer can improve the model performance by fusion the location information. And Using Convolutional Layer can also reduce the input size of Fully Connected Layer, the previous model has unbalanced input and output, which will make it difficult for Fully Connected Layer to recover the data from output to input when do reversed propogation. In this dataset, each sample has 192 feature(64 sensors * 3 channels), I transform the data to 2 dimension (192→3*64) and place 1D Convolution before the Bidirectional Fully Connected Layer. After the data passing Convolutional Layer, the channel reduces to 1 and the size remain unchanged. When do backpropogation, only the fully connected layer do the bidirectional propagation

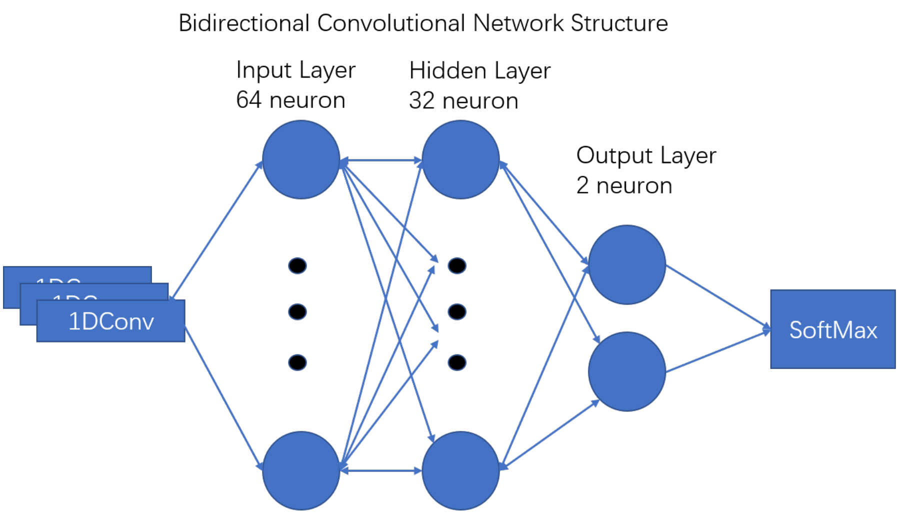

Fig. 12. Bidirectional Convolutional Nerual Network Structure

Due to the poor performance of my computer, I only trained this model for 10000 epochs and find that it does not converge. We can see that the Convolutional Neural is harder to converge and the training speed is relatively slow due to the incereasement of the parameters.

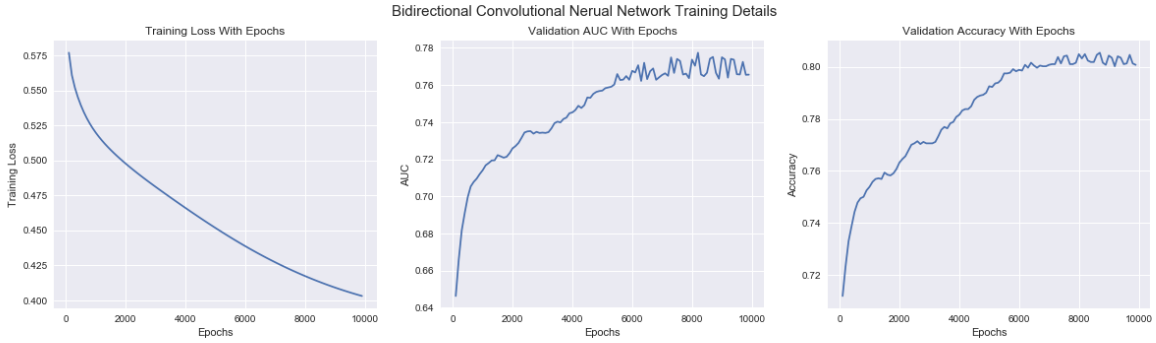

Fig. 13. Bidirectional Convolutional Training

3 Result and Discussion

All experiments were done on an i5-7300 CPU, 8g RAM and Windows machine with python and pytorch CPU version. The following table show result of WITHIN SUBJECT TEST

Table 1. Past Method Classification accuracy – within-subject tests[9]

Method	Accuracy
Normal Channel-wise Autoencoders	0.864
Shared weight Channel-wise Autoencoders	0.858
Normal Image-wise Autoencoders	0.917
Shared weight Image-wise Autoencoders	0.897
EEGNet (Lawhern et al. 2016)	0.878
SyncNet (Li et al. 2017)	0.923
DE (Zheng and Lu 2015)	0.821
PSD (Zheng and Lu 2015)	0.816
rEED (O'Reilly et al. 2012)	0.702

Table 2. My Method Classification accuracy – within-subject[5] tests

Method	Accuracy
Transfer Learning	0.8017
Fully-Connected Neural Network	0.9768
Bidirectional Neural Network	**0.9818**
Bidirectional Convolutional Neural Network	0.8535

From the table, we can conclude that the **Bidirectional Neural Network** performs the best and reach the accuracy of 98%. We can see my three methods are better than most of the past methods.
I also compare the convergence time and epochs time in Figure 11. We can see the BDNN reach the convergence in least epoch numbers, but because it need to do BP twice in one epoch, the training time is longer than Fully Connected Layer

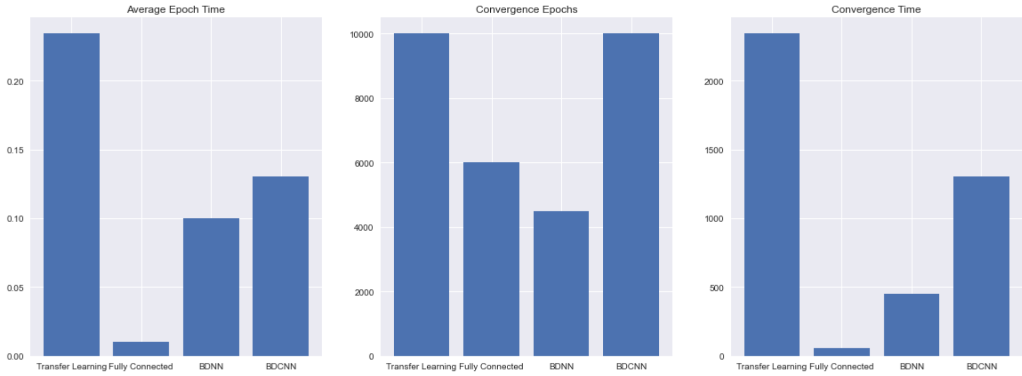

Fig. 14. Training Time and Epochs

4 Conclusion and Future Work

Bidirectional Resnet Transfer Learning has poor performance on EEG classification task, while the Bidirectional Neural Network has the best performance and least training epochs in UCI EEG Dataset Within Subject Test. In future, There are three potential directions:

- Use semantic segmentation model to do the feature extraction
  Compared with ResNet baseline, some SOTA model in semantic segmentation like High Resolution Net-OCR[8] area focus more on the picture's semantic information, which are similar in this situation.
- add attention mechanism:
  Due to the location and value are highly correlated in this problem, attention mechnism might be helpful for model to recognize the specif EEG image.
- Consider the time series feature:
  The prepossessed data provided average the time series dimension, which might cause the loss of information. And some previous Time Series Data Technique like Transformer[7] might be useful in this situation

References

1. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
2. Dua, D., Graff, C.: UCI machine learning repository (2017), http://archive.ics.uci.edu/ml
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
4. LeCun, Y., Bengio, Y., et al.: Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks **3361**(10), 1995 (1995)
5. Liu, Y.: Bidirectional resnet transfer learning for eeg classification. Australian National University (2021)
6. Nejad, A.F., Gedeon, T.D.: Bidirectional neural networks and class prototypes. In: Proceedings of ICNN'95-International Conference on Neural Networks. vol. 3, pp. 1322–1327. IEEE (1995)
7. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. arXiv preprint arXiv:1706.03762 (2017)
8. Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., et al.: Deep high-resolution representation learning for visual recognition. IEEE transactions on pattern analysis and machine intelligence (2020)
9. Yao, Y., Plested, J., Gedeon, T.: Deep feature learning and visualization for eeg recording using autoencoders. In: International Conference on Neural Information Processing. pp. 554–566. Springer (2018)
10. You, K., Long, M., Wang, J., Jordan, M.I.: How does learning rate decay help modern neural networks? arXiv preprint arXiv:1908.01878 (2019)