Classify alcoholic or not using Simple Residual Neural Network

Tao Yu[1]

Research School of Computer Science,
Australian National University
u7044148@anu.edu.au

Abstract. Nowadays, medical data has become more and more important for human's health accompanied by the Emergence of new diseases. EEG (electroencephalography) is a very common medical data that is active in various accepts. That's the reason why it is valuable to study and analyze EGG data. Though Deep learning access like Image Processing and Natural Language Processing can be used to solve a lot of problems in analyzing data, it is difficult for us to use CNN and LSTM to build complex network structure if data is insufficient or devices are not good enough. In this paper, we randomly shuffled our dataset, then build our network structure using Residual Block. By trying different thresholds and training methods like GA, we found that using BP and set threshold as 0.6 can achieve better results. Finally, we can prove that the result of our simple neural network is acceptable by setting effective components.

Keywords: Neural Network · EEG Classification · residual · GA

1 Introduction

In recent years, with the rise of Deep Learning Neural Networks, more and more people start to use this most trendy technology to solve many important aspects of people's daily life. Since health is the most vital thing that people pay attention to, how to use neuron network to read medical data correctly is a problem, especially for EEG data. Electroencephalogram is a graph obtained by magnifying and recording the spontaneous biopotential of the brain from the scalp through sophisticated electronic equipment. It is the spontaneous and rhythmic electrical activity of brain cell groups recorded by electrodes. There are indeed a variety of EEG related networks nowadays, but most of them focus on Natural Language Processing and Conevrntultional Neuron Network. Both NLP and CNN can bring better performance in people's model, however, there may be some situation that people can not use the complex network to build models. In this paper, we will show you based on limited data and limited devices, how we build our neuron networks simply but get a final acceptable result at last. Based on data characteristic we design a special network structure. We also implement a Residual block in our model to overcome the overfitting problem and we choose a good loss function and optimizer to help our model achieve better performance. To enhance the validity of the experiment, we also set some ablation studies like using GA as another training strategy and adjust the threshold to reduce the false-positive rate.

2 Related Work

2.1 EEG Network

In Yao's[6] paper, He introduced 2 kinds of autoencoder that focus on EEG data, one is called Image-wise autoencoders and the other is Channel-Wise autoencoders. In the Image-Wise autoencoder, images are inputs and CNN is the main network for extracting 3 frequency bands. To transfer EEG data to images, they use Azimuthal Equidistant Projection (AEP) to map 3 dimensional 64 channel position into 2-dimensional positions on a surface, finally, a trial 64 * 256 EEG signals are transformed to 32 * 32 * 3 colour pictures. For Channel-Wise autoencoders, the key idea is to split channels information respectively first and combine features in the last FC layer. Yao's result achieves the best result compare with other models before 2017 and proves that autoencoder based feature learning is discriminative and robust for EEG data.

2.2 Residual Block

As a new neuron network called Resnet, which proposed by He[2], caused a huge contribution to Deep Learning. He's idea is to let the internal structure of the model at least have the ability of identity mapping to ensure that in the process of stacking the network, the network will at least not be degraded due to the continued stacking. The network is designed as H(x) = F(x) + x, that is, the identity mapping is directly regarded as a part of the network. The problem can be transformed into learning a residual function F(x) = H(x)-x. As long as F(x)=0, an identity map H(x) = x is formed. Moreover, fitting residuals is at least much easier than fitting identity. Balduzzi's research also shows that [1] even if the mode of the gradient stabilizes within the normal range after BN, the correlation of the gradient is actually attenuated by the increase in the number of exchange layers. It has been proven that ResNet can effectively reduce the attenuation of this correlation

2.3 ClassifyDryGIS

In Milne's paper, he compared the difference performance between Decision Tree Classification like C4.5, Maximum Likelihood Classification and Neural Network Classification[5]. For the task like classification for dry sclerophyll forest supratype, Milne must reduce False Positive percentage to satisfy its requirements. So in his idea, he adapts the threshold in the neuron network.

2.4 Genetic Algorithms

Genetic Algorithm follows the principle of survival of the fittest and survival of the fittest. It is a randomized search algorithm for natural selection and natural genetic mechanisms in the biological world. It has mechanisms such as selection, crossover, and mutation. It retains a group of individuals in multiple conversions. Repeating this process, after multiple generations of evolution, ideally its fitness reaches an approximate optimal state [4]. Since the genetic algorithm was proposed, it has been widely used, especially in the fields of function optimization, production scheduling, pattern recognition, neural network, adaptive control, etc., genetic algorithm has played a great role and improved the improvement of some problems. effectiveness.

3 Method

3.1 Dataset

We use an abridged version EGG signals dataset from UCI, Neurodynamics Laboratory at the State University of New York [3]. In our abridged version EEG signals dataset, we have data that contains 11057 trials and 192 different features. These features will become our training data and features are concatenated in order of 3 different channels: theta, alpha and beta. As we are focusing on solving classify alcoholic problems, our targets are the corresponding alcoholic or not for 11057 trials. No time series is provided in this dataset and the amount of our training data is less than UCI EEG training data. This is a challenging dataset in a way for our model to do the classification problem since no enough features and training samples are provided in advance.

To begin with our task, we first shuffled our dataset randomly to reduce variance and make sure that models remain general and overfit less. In this way, our training and test sets are representative of the overall distribution of the data, which ensured the effectiveness of our data. Then, we divided our dataset into training and test sets with the proportion as 8:2. We don't have a validation set in our simple neural network models since we need more training samples rather than using a validation set to adapt hyperparameters. Since after the data is normalized and standardized, the solution speed of gradient descent can be accelerated, we use BatchNorm to makes it possible to use a larger learning rate for more stable gradient propagation, and even increase the generalization ability of the network.

3.2 Network Structure

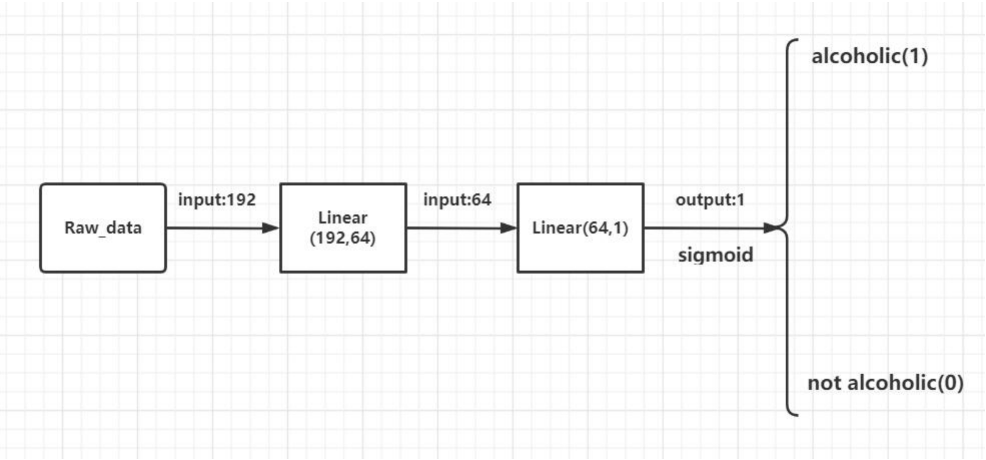

Fig. 1. Our simple Neuron Network Structure

Since we aim to implement a simple neural network, there is no CNN or RNN used in our model. Inspired by Yao's channel-wise autoencoder's structure[6], due to the particularity of our training data: 3 channels (theta, alpha, beta) and each channel have 64 features, we set the first hidden layer parameters as 64, which is corresponding to different 64 features for each channel. We then use BatchNorm to process these features, which reduce the dependence on parameter initialization and increases the generalization ability to a certain extent. After that, we have a linear function to map 64 features to 1, try to extract information from different features. Finally, we use the sigmoid function to map our output between 0 and 1. Here is a straightforward figure that shows our neuron network structure.

3.3 Residual block

In our model, we use a non-linear function like ReLU to help our model to study features and avoid overfitting in a way. However, the overfitting problem is a common risk for people who are training a neural network. The best method to solve this problem is to increase training samples. Because of the limited size of the dataset, we can not get additional data to train and help our model learn more, we implement an easy residual block to fix overfitting problems [2].

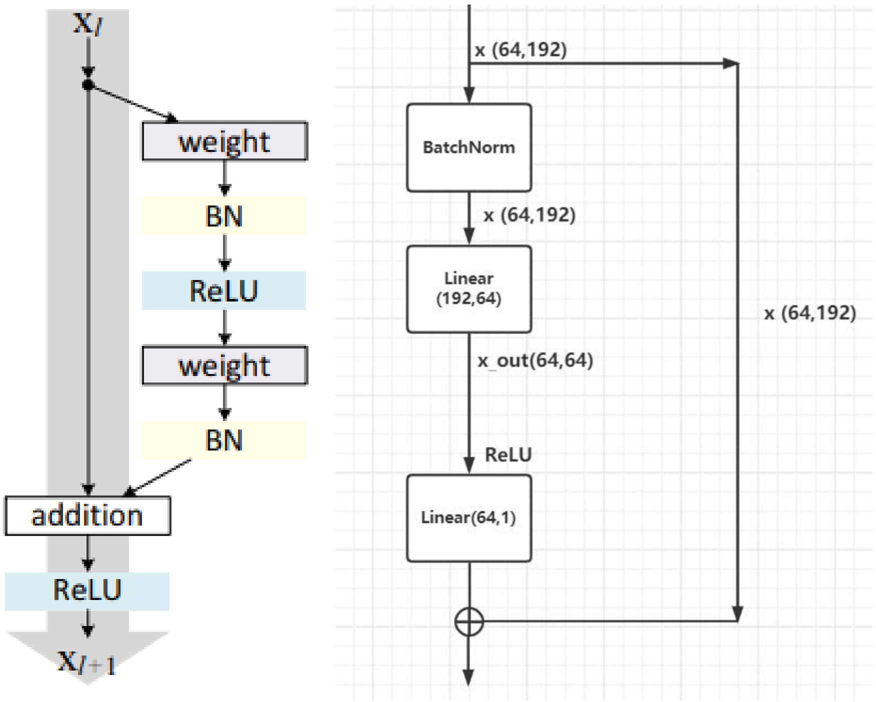

Fig. 2. Left image is Common Residual Block, Right image is the residual block used in our model

As Figure 2 shows: our model first input data x which contains three different channels' features. According to our network structure, the final linear function will output 1 single result as usual. However, the original input data also will become a part of the input in our residual block. Because the dimension may be different from our additional output and origin output, we use downsample to map our additional output to the same size as our origin output. After that, we add 2 outputs together and get our final output.

However, residual block works better on deep layers network. Since our network is quite simple, it will enhance the model's performance in a way but not too much.

3.4 Loss Function and Optimizer

As our task is to classify whether the people in a trial is an alcoholic or not and our target label is always 0 or 1, so cross-entropy can be a good choice to solve this problem especially BCE loss function. In our model, we use the sigmoid activation function + cross-entropy loss function. The sigmoid activation function ”normalizes” the vector output by the neural network into the form of a probability distribution and the cross-entropy describes the difference between the two probability distributions.

For optimizer, we choose Adam optimizer since it based on the mean value of the nearest magnitude of the weight gradient, the learning rate is adaptively reserved for each parameter. This means that the algorithm has excellent performance on non-steady-state and online problems. The empirical results prove that the Adam algorithm has an excellent performance in practice and has great advantages over other kinds of random optimization algorithms

4 Results and Discussions

4.1 BP strategy and adjust theta

We trained our model for 100 epochs, we set batch size as 64, learning rate as 0.01. The code was written in python and PyTorch. All experiments were done on an i7-9750H CPU, Nvidia GTX1650, 8gRAM and Windows environment. Figure 3 shows the loss and accuracy for both training and test set. It is true that our model gets a test accuracy of 0.939 and test loss of 0.2535.

To reduce the false-positive percentage, we draw a figure to see the relationship between different thresholds, FP percentage and accuracy. Finally, we find that 0.6 may be an appropriate threshold for our classification task [5].

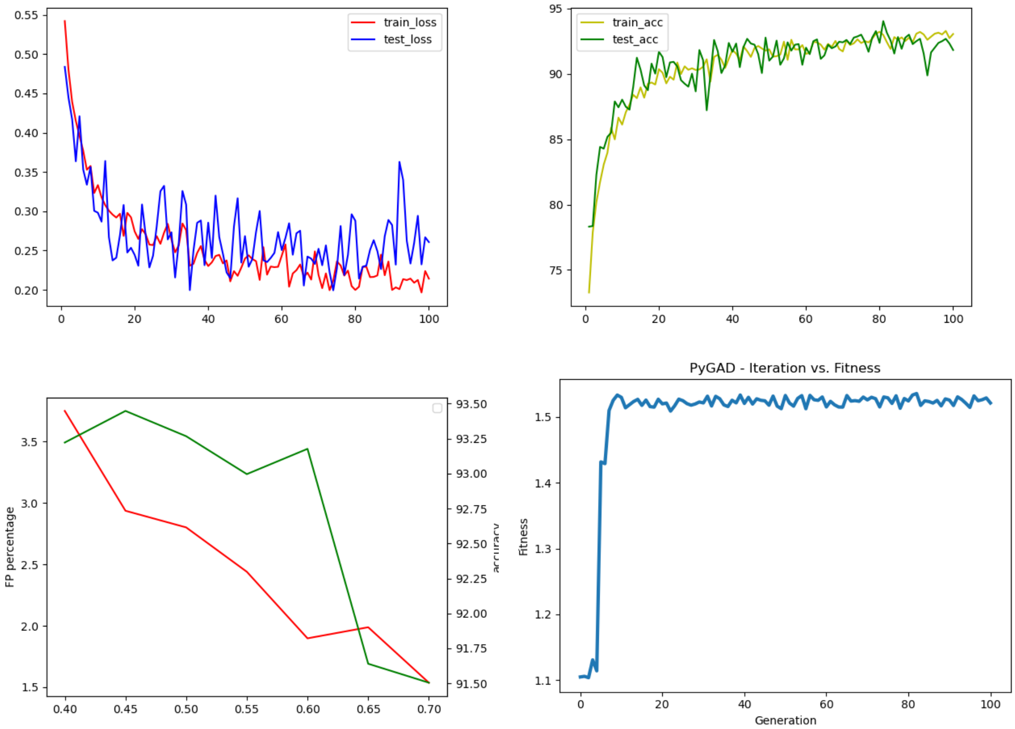

Fig. 3. The first figure: Red Line for Training Loss, Blue line for Test Loss. The second figure: Light Green for training accuracy, Green Line for test accuracy. The third figure: Red line for FP, green line for accuracy. The last figure: changes in fitness during training

4.2 GA strategy

We trained our GA model for 100 generations, set a number of solutions to be selected as parents as 5, Percentage of genes to mutate as 15, type of the crossover operator as a single point and keep all parents in each generation. The last figure in Figure 3 shows the GA's training process. Finally GA model achieves a test accuracy of 0.683 and test loss of 0.645. Obviously, GA's result is not good as BP's result. So we finally use BP as our training strategy rather than GA.

For optimization problems where the problem structure is clear, the target gradient can be easily obtained, and when there is sufficient information about the optimization problem to use, gradient algorithms generally have advantages. Take this EEG signal and alcoholic as an example, there may be a relationship between each other and the whole optimization problem is not an NP-hard problem, we would like to use common Mathematical optimization like gradient descent to solve the problem.

What is more, The calculation speed of the genetic algorithm is relatively slow. When the running time of the algorithm tends to infinity, it tends to the global optimum with probability 1. The result of this theoretical analysis is better than nothing. Because no one sets the running time to infinity when running the algorithm. In this way, if we don't care about the adjustment of algorithm parameters and computational complexity, using GAs to find a better network design is helpful, especially in finding the initial weights of parameters and avoiding local minimal. It still can get an acceptable result in a way.

4.3 Results comparison

Although the dataset we used in our neuron network is not a normal one, we still can compare experiment result with other EEG network in a way. Below is the classification result for the different network. The accuracy of other methods is listed from Li's paper[3].

Methods	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
Ours	0.939

Compared with other EEG network, our final result is the best, which can prove the advantages of our network .

5 Conclusion and Future Work

5.1 Conclusion

In conclusion, to improve the accuracy of the EEG neuron network, the design of a simple model really make promotion in a way. With our network structure, the network can study different kinds of features which can extract better information. With Residual, our model won't afraid of overfitting anymore but to improve its performance; Adam and BCE Loss Function is also helpful for training EEG data which brings external benefits. Finally, we also compare the performance with different training strategy and increase our model's threshold to gain a lower False Positive.

5.2 Future work

The total size of our data is too small (11097 trials). What is more, Due to the characteristic of our data, we can not use common data augmentations like affine, flip and colour jitter which we commonly used in Image Processing to increase data. Need to find a better way to help our model to get more data in order to enhance its performance.

There may be a good Loss function for processing EEG data, instead of just using the BCE Loss function. So is the optimizer.

More special hidden layers needed to replace current network layers. Maybe using conv1d to process raw data instead of just using the linear function can achieve better results.

References

1. Balduzzi, D., Frean, M., Leary, L., Lewis, J., Ma, K.W.D., McWilliams, B.: The shattered gradients problem: If resnets are the answer, then what is the question? (2018)
2. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition (2015)
3. Li, Y., Murias, m., Major, s., Dawson, g., Dzirasa, K., Carin, L., Carlson, D.E.: Targeting eeg/lfp synchrony with neural nets. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017), https://proceedings.neurips.cc/paper/2017/file/7993e11204b215b27694b6f139e34ce8-Paper.pdf
4. Man, K., Tang, K., Kwong, S.: Genetic algorithms: concepts and applications [in engineering design]. IEEE Transactions on Industrial Electronics **43**(5), 519–534 (1996). https://doi.org/10.1109/41.538609
5. Milne, L., Gedeon, T., Skidmore, A.: Classifying dry sclerophyll forest from augmented satellite data: Comparing neural network, decision tree maximum likelihood (01 1995)
6. Yao, Y., Plested, J., Gedeon, T.: Deep feature learning and visualization for eeg recording using autoencoders. In: Cheng, L., Leung, A.C.S., Ozawa, S. (eds.) Neural Information Processing. pp. 554–566. Springer International Publishing, Cham (2018)