Implementation of connectionist compression by auto-associative network

Zhe Li
The Australian National University

Abstract. In this report, a dataset contains records of Electroencephalogram (EEG) experiment and MINST handwritten digits dataset is chosen to implement the connectionist compression. After running the original dataset and compressed dataset on the feed forward neural network(FFNN) and convolutional neural network(CNN), the comparison of their results shows that the connectionist compression could reduce the running time and improve the accuracy of the classification of eye-state. It also discusses that the overfitting problem damage the performance of the FFNN result and how to avoid it.

Keywords: Electroencephalogram (EEG), Connectionist compression, Auto-associative, Comparison

1 Introduction

Eye state identification is one of the most popular research topics in recent years. It aims to determine individuals in certain cognition states by analyzing the movements of the eye, while it can be used in several projects, for example, driver's drowsiness detection (DDD) problem and brain computer interface(BCI). Studying this topic has significant meaning since it could provide improvement of safety or convenience to specific population. Meanwhile, it requires efficient algorithm and large calculating power to prove accuracy and speed of the eye state identifier. There are many methods applied in this area including machine learning and artificial neural network. The purpose of this report is implementing the connectionist compression and comparing the results with non-compression data to test the performance of auto-associative network.

This report choose dataset provided by Baden-Wuerttemberg Cooperative State University [1]. A test called Electroencephalography (EEG) is applied to record values, which represent the activities of brain cells and a camera is used to capture the images of the eye state. This dataset is large which is sufficient for the training and testing.

A neural network is constructed to process EEG records and give back the prediction of the eye state. Also, based on the Gedeon's approach [2], the auto-associative network is built to achieve connectionist compression. After the compression, the original data and compressed data are processed by the prior neural network built to compare the results. The comparison with a modern neural network using deep architectures will be conducted.

Another hot research topic in machine learning area is image recognition. It is designed to process and analyze the image captured by the camera and get the right label of the object in the image. The dataset used in the report is MINST database which created by modifying the samples from NIST's datasets and each sample is the image of the handwritten numbers. The whole dataset contains two sub-datasets. One is the training-set collected from American Census Bureau works, while the other one is the testing-set collected from American high school students [3]. Due to it is an ideal database for deep learning method, we use it to test the convolutional neural network(CNN) we will construct. Also, the auto-associative network will be adapted on the MINST dataset to test the performance of compression on image.

2 Method

This EEG dataset contains 14980 instances and each instance has 14 attributes. It is record by EGG headset and stored in chronological order, and the last column in the dataset is added manually with '1' indicates the closed-eye and '0' indicates the opened-eye. The dataset is checked that there is no missing value or duplicated value. The neural network will be conducted on the whole datasets.

A feed forward neural network(FFNN) with 14 input units, 50 hidden units, and 2 output units is constructed as basic neural network. The number of epochs is set to 3000 after testing from 50 to 5000 and batch size is 64. The resilient backpropagation is applied to this neural network to calculate the gradient and weights calculation. The sigmoid function is built in it, as it can reduce the difficulty and time to calculation in hidden layer. Also, the cross-entropy loss function is used as an error measurement between predictions and targets. Before the datasets run through the FFNN, the data need to be normalized by standard score to reduce the redundancy. Then, the FFNN will trained by the 80 percent of the whole dataset by randomly picking the data and return the accuracy and loss rate in each 50 epoches. After the training finished, a confusion matrix is produced to analyze. The rest of 20 percent data will be used for testing, and a confusion matrix as well as the testing accuracy will be generated.

Based on Gedeon's approach [2], the main idea is to construct a special architectural neural network to achieve functional symmetry and a normal feedforward network is shown in Figure 1, given by Gedeon's paper.

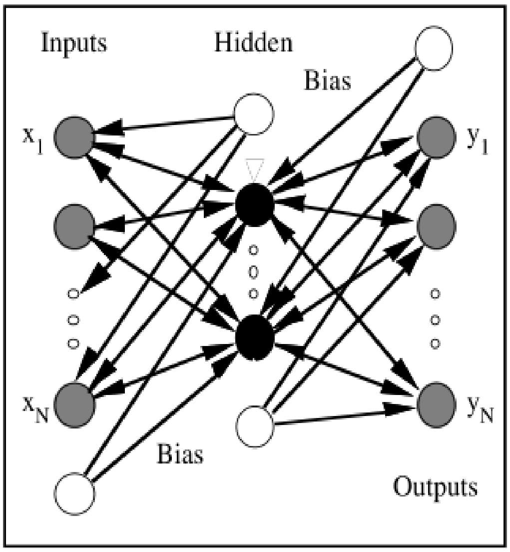

Fig. 1. Normal feedforward network [2]

To achieve connectionist compression, some adjustment need to be implemented in the prior FFNN we build. The number of outputs units need to be modified as same as the number of input units. Hidden size need to be reduced below the number of inputs units to achieve the function of compression and decompression. In our implementation, the hidden size is set as half of the number of input units. For the learning of compressed function, the whole dataset need to run through the auto-associative network to gain the settled initial cost. Then, the whole dataset run through the auto-associative network again and export the values of 7 hidden units for every record. By compressing, the attributes reduction is complete. These records will combine with the targets from the original dataset to a new compressed dataset.

After acquiring the compressed dataset, it will be used same as the original dataset on the first FFNN and followed the same step. Then, the accuracy and total running time need to be compared to define the performance of the compressed dataset. Also, it will be compared with a deep learning architecture which implements on the same dataset to test the performance.

The MINST dataset has 70000 examples in total and each piece of data is a gray level image with 28x28 pixels. To achieve the compression on this dataset, some parameters of the FFNN we built before need to be changed to adapt to the handwritten image. The number of epochs is set to 10 and batch size is 64. The number of outputs units need to be modified as same as the number of input units as 784. In this implementation, the hidden size is set as quarter of the number of input units as 196. Same as the compressed function we mentioned above, the MINST dataset need to run through the auto-associative network to gain the settled initial cost. Then, the whole dataset run through the auto-associative network again and export the values of 196 hidden units for every record. For convenience, the original image data is converted to csv format as input and each array contains 784 values which represent the value of each pixel. After the compression, the size of the original image reduced to 14x14.

A CNN with two convolutional layers followed by a dropout and two fully connected layers is construed. The number of epochs is set to 10 as default and batch size is 64. Also, stochastic gradient descent (SGD) algorithm is adopted to perform the optimization and rectified linear unit (ReLU) is adopted to perform the activation function. The training-set is used to train the CNN and the test-set is used to test the prediction of the model. After running this CNN with the original dataset, some parameters of the CNN need to be modified in order to be run on the compressed dataset.

3 Results and Discussion

Table 1. Comparison the accuracy and running time of different type of EGG dataset on different architecture neural network.

Type of dataset	Accuracy	Time for classification
Original dataset on FFNN	29.29%	0.001332seconds
Compressed dataset on FFNN	25.8%	0.001308seconds
Original dataset on dropout NN	94.72%	1.18seconds

The accuracy and running time is shown in the first two rows of the Table 1 above. After compressing the dataset, the running time is shorter than the time taken on the original dataset. The accuracy of classification is also reduced about 3.4 percent. However, the dimensionality reduction should improve the accuracy as expected. Through the observation of the experiment, the overfitting problem has significant influence on the training. During the original dataset on FFNN, the training accuracy is fluctuated between 47.36% and 59.67%, while in the compressed dataset on FFNN, the training accuracy is fluctuated between 55.29% and 63.48%. The compressed dataset improves the training result and it is supposed to give back better prediction, if the overfitting problem could be fixed. To avoid overfitting, data regularization should be applied.

The comparison with dropout NN [4] is also shown in the last row of the Table 1. The dropout NN provides results with much higher accuracy and slow running time. The dropout NN contains 2 layers and 500 units, which is more complex than the FFNN that is built in this report, causing more time usage. The advantage of dropout NN is that it can reduce overfitting efficiently, which is exactly needed in our dataset.

The accuracy of CNN on original MINST dataset is 98.46%, which is quite perfect and acceptable. This is because the CNN we constructed combines several modern and mature algorithms to improve the performance of classification. It cost 1.13 second to run through all the testing-set. While the accuracy of CNN on compressed MINST dataset the accuracy drops. The reason could be that, during the compression, it may damage some key features which could has negative influence on detection and classification. The advantage of using compressed dataset could be found on running time since it reduces the size of the datasets that need to be processed, although the difference could be minor.

4 Conclusion and Future work

By building the CNN and FFNN, the results show that the implementation of connectionist compression is an efficient way to reduce the requirement of classification time. Although, in our test, the result is influenced by the overfitting problem, it still can be find that if the neural network could involve some modern algorithms to avoid certain protentional problem, connectionist compression is supposed to be a good mothed to improve the calculating speed without large negative influence on the accuracy.

Although it can shorten the running time, using associative network to compress the dataset usually cost quit long time to get the result. A more intelligent method should be considered to improve the compression speed. Also, it cannot be avoided to lose key features in compressing process since the method is brutal and forcible. Especially on image dataset, it should be considered more carefully to compress the image and keep the features unbroken so that it can suit the CNN constructed. Based on the view that we own many outstanding methods to compress the image to different format in industry, it is recommended to improve the associative network with the experience from these methods.

5 References

[1] O. Rosler and D. Suendermann(2013), First step towards eye state prediction using EEG, in Proceedings of the International Conference on Applied Informatics for Health and Life Sciences (AIHLS 13), Istanbul, Turkey, 2013.

[2] Gedeon, T. D. (1998, October). Stochastic bidirectional training. In Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on (Vol. 2, pp. 1968-1971). IEEE.

[3] P. Haffner, L. Bottou, P. Howard, P. Simard, Y. Bengio and Y. LeCun: Browsing through High Quality Document Images with DjVu, Proc. of Advances in Digital Libraries 98, 309-318, IEEE, 1998,

[4] Reddy, T. K., & Behera, L. (2016). Online Eye state recognition from EEG data using Deep architectures. 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC).