Deep Feature Learning for EEG Recording Using Autoencoders

Yue Yao*
Research School of Computer Science
The Australian National University
*yue.yao@anu.edu.au

Abstract. In this era of deep learning and big data, the transformation of biomedical big data into recognizable patterns is an important research focus and a great challenge in bioinformatics. An important form of biomedical data are electroencephalography (EEG) signals, which are generally strongly affected by noise and there exist notable individual and environmental differences. In this paper, we focus on learning discriminative features. Inspired by traditional image compression techniques to learn a robust representation of an image, we introduce and compare three strategies for learning features from EEG using three specifically designed autoencoders. Channel-wise autoencoders focus on features in a certain channel while Image-wise autoencoders instead learn features from the whole trial. The GAN-based autoencoders are further designed for EEG data filtering. While in this paper we will only focus on elimination of some information of EEG images using GAN-based autoencoders. We used a UCI EEG dataset. Our results show that using both Channel-wise and Image-wise autoencoders achieve good performance for discriminating a classification problem with nearly state of art accuracy. A further experiment using shared weights showed that the shared weights technique only slightly influenced learning but it reduced training time significantly. The experiment on GAN-based autoencoders shows it can it can successfully remove the disease information in nearly 70% EEG images.

Keywords: Deep Learning, Neural Network, Autoencoders, Convolutional Neural Network, EEG, Brain-Computer Interface, Image Translation, Generative Adversarial Nets

1 Introduction

As an import part of Brain-Computer Interface (BCI), EEG has found a variety of interesting and useful applications for users and has become increasingly important in various areas. Especially for the medical field, diagnosis of epilepsy for example, EEG has shown great success [7, 14]. Gathered from the scalp, the EEG is a kind of signal that contains information about the electrical activity of the brain. It uses electrodes placed on the scalp to get electrical information. Since it is the overall measurement of human brain electrical activity, it may contain a wealth of information. This is the reason why EEG can be applied to diverse areas like personal recognition, disease identification [7], sleep stage classification, rebuild the picture from personal eyes [6], and so on. On the other hand, being full of information also often means full of noise and interference, making it very hard to extracted readable features from it. Furthermore, full of information also often means full of personal privacy issues. For example, if we would like to use EEG for personal recognition task. The only information we would like to upload is personal identity. But since there still not exist a successful information filtering algorithm, hackers will be able to get our other information like disease information, emotion information and so on.

For feature filtering and elimination task for EEG, there hasn't been an applicable method for this. But if with such feature filtering algorithm realized, many applications using EEG could be finally achieved and change our life. Still taking personal recognition for example, compared with fingerprint or face recognition, EEG has more advantage on identifying different people. For instance, if one person's fingerprint is stolen or one person's face is reconstructed by others, it is basically irreparable because the fingerprint and face model are irrevocable. But for EEG data, if it is hacked by others, users can still reset a new EEG pattern because the EEG recognizer can identify a person by both personal details and personal brain action.

For feature learning tasks, a wide range of traditional machine learning algorithms have been applied and achieved success. In some areas, such as for bio-signals like EEG, many well-known algorithms have been applied like support vector machine, random forests, Bayesian networks, and hidden Markov models [5]. Good performance of conventional machine learning algorithms relies heavily on features extracted [10]. Traditional learned features are not always as good as we want. For this reason, we need algorithms that can learn features from big data automatically.

Deep learning, a neural network based technique, is a new branch of machine learning. It has achieved great success in computer vision (CV) and natural language processing (NLP) in recent years. Unlike CV and NLP which have many successful algorithms and datasets using deep learning, bio information areas have no widely accepted learning algorithms or even a well-known and popular dataset like imagenet in CV. Human brain waves have commonalities and differences, and it is these commonalities and differences which are exactly the features that we want. We believe that only when all these features are better understood can we make it possible to design a robust and recognized deep learning method in this area, because even deep learning approaches need some understanding of the structure of the data to extract features well.

As a result, deep learning based approaches are utilized in this paper. Two autoencoder-based techniques are used for feature learning and dimensionality reduction and one other is used for data filtering. They are referred to as Channel-wise autoencoders, Image-wise autoencoders and GAN-based autoencoders. Channel-wise autoencoders are inspired by

one-dimensional convolutions. For EEG data, the number of channels is often significantly less than the timescale length, forming an unbalanced matrix input. So, for applying convolutional neural networks based techniques, it is usual to perform one-dimensional convolution first, then followed by two-dimensional convolution for feature extraction [5]. For this reason, we design a group of channel-wise autoencoders witch only focus on features from a certain channel using simple fully connected layers. The Image-wise autoencoders are designed based on fast fourier transform (FFT) and CNN. Using FFT, we can obtain theta, alpha, beta EEG brain frequency bands, then we use these frequency bands to achieve a colored picture.Then, a CNN based autoencoder is used to extract features from these colored pictures. GAN-based autoencoder is inspired by Image-to-Image translation [19]. It is designed to map one image distribution to another image distribution. In our paper, such translation mechanism can be used for feature filtering.

2 Related work

Convolutional neural network (CNN) is a feature extraction network proposed by Lecun [11], based on the structure of the mammalian visual cortex – thus providing structural information about the data via the network topology. The difference between convolution neural networks and the traditional neural networks is the convolution layer. The convolution layer is a feature extraction layer. In general, a convolutional layer contains multiple convolution kernels. Each kernel performs convolution operations on each channel of the image in forward propagation. The result is concatenated along the third dimension to form a new matrix for further calculation. Multiple convolution and pooling layers and some final fully connected layers form the entire network.

Traditional neural networks treat each pixel in the image as an independent variable ignoring the 'connection' between adjacent pixels. The convolution operation utilizes the association between pixels in neighborhoods, so the result is more robust and the generalization ability is stronger. During training, the kernels will remember patterns as local features. During testing, those features will be detected by kernels and do classification according to these features.

We consider the convolution layers as feature extractor. Then, the fully connected layer serves as a 'classifier' trying to find decision boundaries between each class. From another point of view, the role of the fully connected layer is similar to the kernel method, warping the high-level feature space to make each class approximately linearly separable.

Other than autoencoders, there is other CNN based research. Depending on the type of the kernel, CNN based work can be divided into normal CNN as well as frequency-based CNN. Normal CNN takes the raw EEG as the input while the frequency based CNN takes frequency features as the inputs. Examples of normal CNN approaches include Deep4net [9] and EEGNet [8],the and the SyncNet [4] is the latest example of a frequency based CNN. An interesting commonality is that one-dimensional convolutions should be applied at the begging of all convolution procedures.

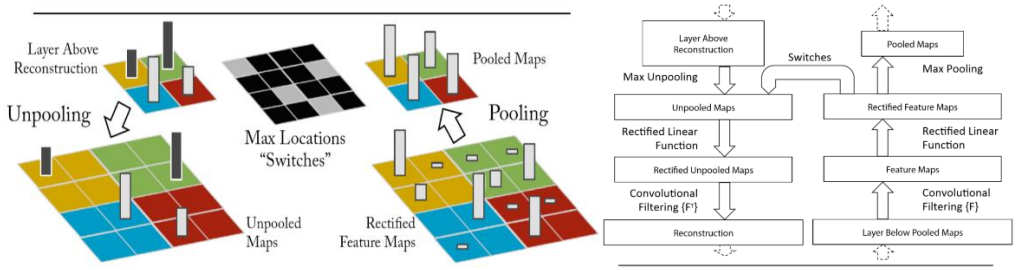

Figure. 1 The Function of Switches and the Structure of DNN [12]

Deconvolution neural network (DNN) was first used for visualization by Zeiler and Fergus [12]. A high-level feature map with many high dimensional features is difficult to interpret intuitively. A DNN projects the response value of the specified convolution layer into the input pixel space by reversing the CNN, thus revealing the contribution made by each pixel of the input image to the response value, thus creating a more comprehensible feature map visualization. These operations are shown in the right of Fig. 1 The right side is the process of forward propagation, while the left side is the process of mapping the response value back to the input pixel space.

From the left of Fig. 1, it can be seen that the output size of the DNN is consistent with the input size of the CNN. In DNN, up-sampling is difficult to implement because only the maximum value in each pooling area is preserved during the down-sampling process without original coordinates. The solution to this problem is to use a "Switches" structure to store the coordinates during down-sampling. As shown in the left of Fig. 1, where the right side is down-sampling, the four locations of the maximum value are saved to the "Switches". So, on the left, DNN retrieves the "Switches" and determines the location of the value.

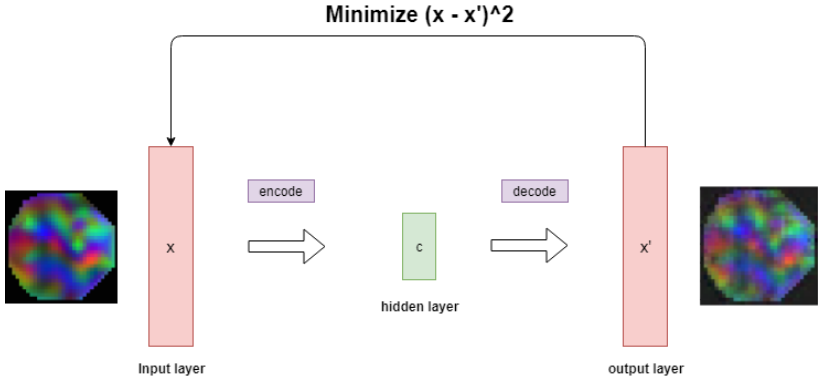

Figure. 2 Structure of Autoencoder

Autoencoder is a sort of compression algorithm, or dimension reduction algorithm, which is similar to Principal Components Analysis (PCA). But compared with PCA, the autoencoder has no linear constraints. The autoencoder structure has been widely used for image compression, for example [1], which inspired us to try an autoencoder based learning algorithm. From Fig. 2, an autoencoder can be divided into two parts, an encoder and a decoder. The number of nodes in the hidden layer is generally less than the nodes in the input layer and the output layer. That is, the original input is compressed to a smaller feature vector. In equation 1 below, ϕ and ψ stand for encoder and decoder, respectively, and L means squared loss. The objective of the autoencoder is to minimize the difference between the input and the generated output. A CNN based autoencoder uses convolution operations as the encoder and deconvolution operations for the decoder, making it possible to operate on image data.

$$\phi, \psi = \operatorname{argmin}_{\phi,\psi} L\ (X,\ (\phi \circ \psi)X) \tag{1}$$

One of the major purpose of our work it to help advance the state of art in feature abstraction of signal analysis for biosignals like EEG. Prior to our work, a number of autoencoder related deep learning methods have been applied to EEG signals. For autoencoders, Sebastian and Avital use convolutional autoencoders with custom constraints to learn features and improve generalization across subjects and trials [2]. Hajinoroozi uses fully connected stacked autoencoders on the output of a supervised trained CNN [13]. They all achieved commendable results but all use CNN autoencoders directly on the raw whole trial EEG signal without effort on preprocessing to reduce the high inherent noise in EEG signals.

Generative adversarial networks (GAN) is a system of two neural networks contesting with each other in a minimax game framework [21]. It achieves a great success in image generation area [22]. It mainly includes two parts, namely generator and discriminator. The generator is mainly used to learn the distribution of the real image in order to fool the discriminator and the discriminator needs to accept the real image while rejecting the generated image. Throughout this process, the generator strives to make the generated image more realistic, while the discriminator strives to identify the real image. The key part of GAN is the adversarial loss [20]. For image generation task, the adversarial loss is very powerful for images in one domain transformed to the other domain. The co-operation and confrontation between neural networks have become a hot topic since the breakthrough of GAN related work. From the bio-inspired point of view, human beings are living in an environment full of co-operation and competition. As a result, there should be also a good option for a neural network system to do the same thing in order to learn robust features.

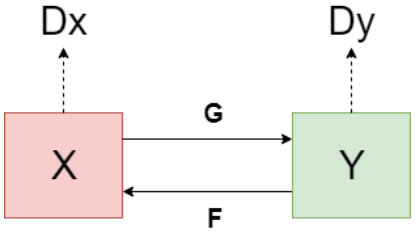

Figure. 3 CycleGAN Structrue

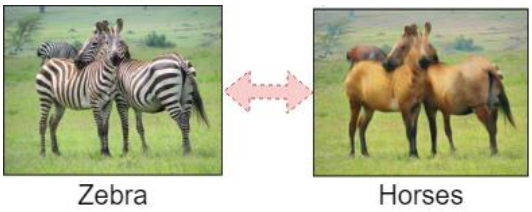

Figure. 4 Image Translation Example [20]

Image-to-image translation is a kind of system that can learn the mapping between input image distribution and output image distribution using two separate images domains [19]. Shown in Fig. 3, given a source distribution X, we are aiming to use a generation model G to map our source distribution X to target distribution Y. An example is shown in Fig. 4, though it is not perfect, the translation system has successfully transferred the most important features between zebra and horses like the skin color. In this translation system, we do not explicitly tell the neural network to change some features. Instead, we have the prior knowledge of two separated image distribution. As a result, it is possible for us to extract the stylistic difference between two image distribution and then directly translate them from one domain to the other domain.

Cycle-Consistent Adversarial Networks (CycleGAN) is a well-known image-to-image translation for unpaired images [20]. It overcomes the difficulty of getting the paired image and form an autoencoder like structure to achieve image translation. In Figure. 3, G is such generator that generates a domain Y image from domain X while F is the generator that generates a domain X image from the domain Y. Dx and Dy are two discriminators that used to justify whether the coming image really belongs to domain X and domain Y respectively. The training procedure could be separated into two symmetric part. One is $X \rightarrow G(X) \rightarrow F(G(X))$. In this autoencoder-like loop, the training loss come from two part, One is the discriminator loss comes from Dy to judge whether G(X) is really from domain Y and another

one is the reconstruction loss to judge whether F(G(X)) is the same as X or not. The other loop $Y \rightarrow F(Y) \rightarrow G(F(Y))$ is in the same principle.

But all these GAN methods is based on two hypothesis. One is that it is possible to build a strong classifier that can discriminate such feature and the other one is to use a reliable generator that can filter out original feature and rebuild target feature. For the first hypothesis, if we cannot train a strong classifier in normal labeled training, it will be almost impossible for us get a strong discriminator in adversarial training because adversarial training itself is not designed to help training with discriminator. That should be one of the reasons why GAN base method has achieved great success in CV area because the current most popular datasets like mnist [23] and cifar-10 [24] have already achieved more than 90% accuracy using different CNN. In contrast to CV, since NLP area has not own a universally recognized text classification method, current GAN method for NLP like Seqgan [25] and Leakgan [26] does not have a strong discriminator to guide the generator. For the hypothesis two, building a strong generator is strongly related the given type of data. For the image translation area, convolution and deconvolution-based method is often used and the further U-net [27] based method achieve the state of art [19].

Furthermore, the evaluation method for GAN is still remained to be solved. From the long time after the original GAN paper published, the generated result in CV area is still be judged by manual selection [28]. But after the critical job from the google brain [28]. FID and F1 score is introduced to judge the generation quality of a GAN. But since the FID and F1 score requires a strong pre-trained classifier on other datasets, making to become very difficult to implement on other areas like NLP or bio-signal.

EEG2Image is a work which provides a method which transfers EEG signal to images and they further applied recurrent-convolutional network for the image features [3]. It forms part of the base of our approach so it will be discussed later.

3 Methodology

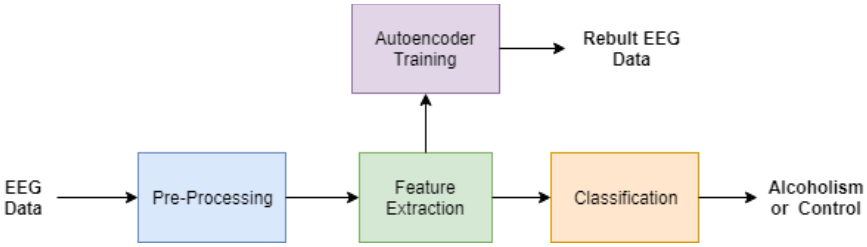

Figure. 5 Structure of General Procedure for Learning Discriminative Feature

The block diagram shows the general procedure for Channel-wise Autoencoders and Image-wise Autoencoders, as depicted in Fig. 5. We first pre-process the raw EEG data into a useable form. Then feature extraction and dimensionality reduction are done by using autoencoders. Finally, two fully connected layers are utilized to do classification and evaluation. We extract features prior to applying the classification. To achieve this, two kinds of autoencoders are used to enhance features. That is, Channel-wise autoencoders and Image-wise autoencoders.

3.1 Dataset

This dataset we use is from UCI, the EEG dataset from Neurodynamics Laboratory at the State University of New York. It has a total of 122 subjects with 77 diagnosed with alcoholism and 45 control subjects [4,15]. Each subject has 120 separate trials. If this subject is labeled with alcoholism, all 120 trails belong to that subject will be labeled as alcoholism. The stimuli they use are several pictures selected from the Snodgrass and Vanderwart picture set. One trail of EEG signal is of one second length and is sampled at 256Hz with 64 electrodes. Models are evaluated using data within subjects, which is randomly split as 7:3 for training and testing for one person [8]. The classification task is to recover whether the subject has been diagnosed with alcoholism or is a control subject. Also, we noticed that this dataset is not a balanced dataset. It is a two-task classification but alcoholism trials account for more than 70% of the data.

The usual challenges of handling EEG make it more difficult to apply deep learning methods compared with computer vision data or natural language processing data. The UCI EEG dataset is not an exception. First, a label is usually applied to one trial. But as one trial contains 64 channels and 256 time serious data, making it become a 64×256 large matrix. In other words, a signal EEG trial has 64×256 attributes, making it become impossible to be directly fed into a neural network. Otherwise, it will be too large to compute efficiently. Second, EEG is a kind of time-series data but it lacks recognizable patterns in single time slices (1/sampling rate) compared with natural language processing. Third, as previous work has shown, if we consider an EEG signal as a picture and directly use a convolution neural network, there is always a serious problem to determine the size of the kernels at each stage [5]. That is because the original features could be distributed with different time differences in a single trial depending on the scenario (different classification task for example). To address these difficulties, we used two kinds of autoencoders as described in the following.

3.2 Channel-wise Autoencoders

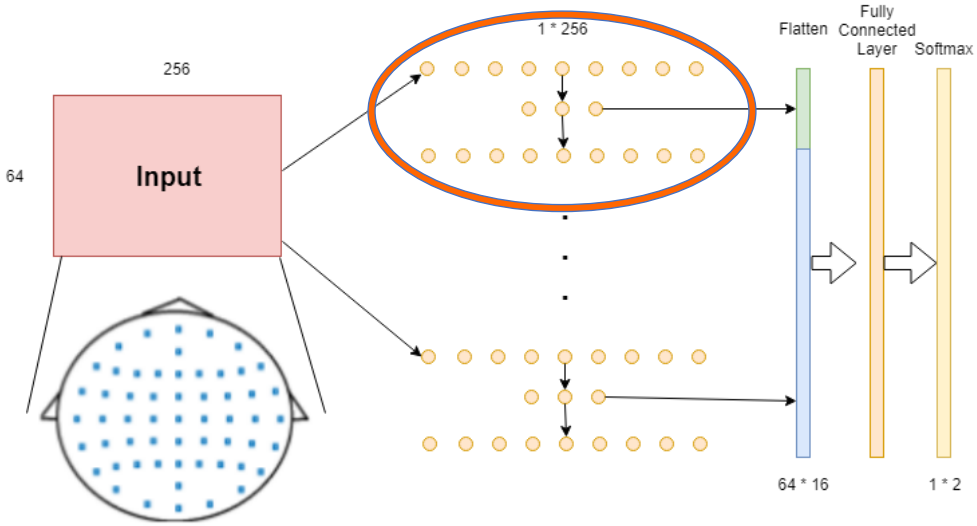

Figure. 6 Structure of Channel-wise Autoencoders

The key idea of applying channel-wise autoencoder is to separate the feature extraction procedure into two parts, the channel-wise autoencoder only focuses on features in one channel while the final fully connected layer will combine features across channels to make a final prediction. As shown in Fig. 6. An EEG trial with the 64 × 256 dimensions will be separated into 64 1×256 signals, then each signal will be input for one autoencoder only designed for that channel. These 64 autoencoders are just 2 layers of a fully connected neural network with 16 hidden units in the middle. The input of autoencoders will be normalized to [-1, 1] and there is tanh activation function for the output layer to match the output to [-1,1] as well. The shared weight technique derived from image compression [1] is also used for signal compression, which takes the transpose of encoder weights for the decoder weights.

3.3 Image-wise Autoencoders

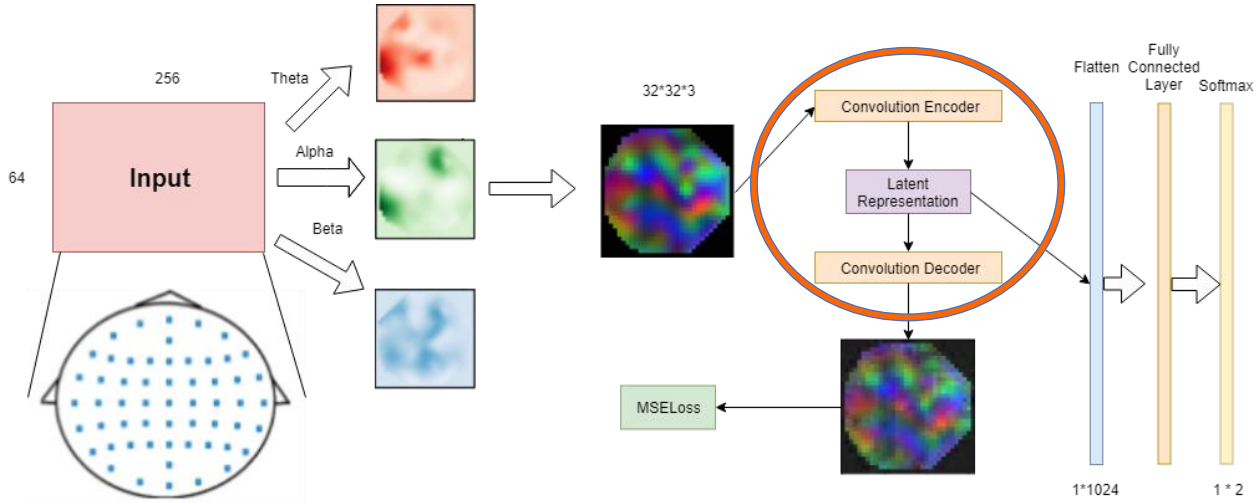

Figure. 7 Structure of Image-wise Autoencoder

The image-wise autoencoders take the images as input while using convolution neural network to extract features. The whole procedure is shown in Fig. 7 and below is some further explanation.

A. EEG to Image:

Figure. 8 EEG Signal to Image Example

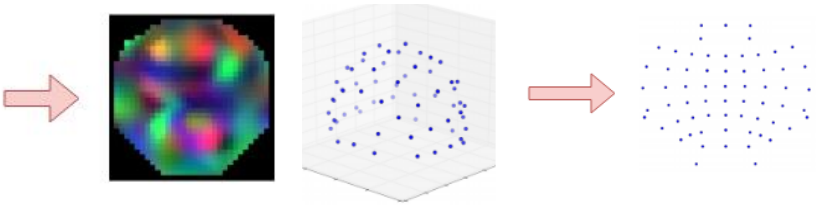

Figure. 9 Transform 3-D Coordinate to 2-D Coordinate [3]

The EEG to image method is derived from the Bashivan's work [3]. As illustrated in Fig. 8, it is a method that combines the time-series information and spatial channel locations information over the cortex in a trial of EEG signal. A fast fourier transform is performed on the time series to estimate the power spectrum of the signal for each trial (64×256). Then, three frequency bands of theta (4-7Hz), alpha (8-13Hz), and beta (13-30Hz) are extracted and the sum of squared absolute values in these frequency bands are used, forming a 64×3 map. To form an RGB EEG image, the theta frequency will be the red channel, Alpha is the green channel and the Beta is the blue channel. For each frequency band (64×1), shown in Fig. 9, Azimuthal Equidistant Projection (AEP) also known as Polar Projection is used to map the three dimensional 64 channel position into two dimensional positions on a flat surface. That is, all EEG electrodes positions are mapped into a

consistent 2-D space because the original EEG electrodes are distributed over the scalp is in a three dimensional fashion. In this way, each 64×1 frequency band can be mapped to a 32×32 mesh, forming 32×32×3 data. The CloughTocher scheme is used for estimating the values in-between the electrodes over the 32 × 32 mesh. Finally, a trial of 64×256 EEG signal is transformed to a 32×32×3 color pictures.

B. Autoencoder design:

Encoder	Decoder
Input 32×32×3 Color Image	Input 16×8×8 Matrix
3×3 conv, 2×2 max-pooling ReLU, 0.25 dropout 3×3 conv, 2×2 max-pooling ReLU, 0.25 dropout 3×3 conv, ReLU	3×3 deconv, 2×2 max-un-pooling ReLU, 0.25 dropout 3×3 deconv, 2×2 max-un-pooling ReLU, 0.25 dropout 3×3 deconv

Table. 1 The Detailed Encoder and Decoder Structure

The design of this CNN based autoencoder is inspired by the CNN for cifar-10 [16]. The CIFAR-10 dataset consists of 60,000 32x32 color images in 10 classes, with 6,000 images per class, and has the same input dimension as our generated EEG pictures. Our encoder takes one input layer, three convolution layers, three activation layers as well as two pooling layers. For our decoder, it takes one output layer, three deconvolution layers, three activation layers as well as two pooling layers. Our shared weight CNN Autoencoder is in the same structure as the normal CNN autoencoder but the weight of the three deconvolution layers is fixed and derived from encoder's convolution layer. The Rectified Linear Unit (ReLU) is used for activation layers to speed up the training process while dropout is performed after every activation layer to make the model more robust, since it forces all the layers before the dropout to extract redundant representations. In principle, we do not impose a particular order for pooling layers and ReLU since they are known to be the same as expressed in equation 2.

$$\max(\mathrm{ReLU}(x1), \mathrm{ReLU}(x2)) = \mathrm{ReLU}(\max(x1, x2)) \tag{2}$$

3.4 Classification Task

The features extracted from channel-wise autoencoder and image-wise autoencoder will be flattened into a long vector, composed of 16 hidden unit representations x 64 autoencoders in the channel-wise case. Then we use a feedforward network with two hidden layers. Before the output layer, a dropout layer with 0.5 dropout rate is added. Also, a ReLU layer is added after the first fully connected layer.

3.5 GAN-Based Autoencoder

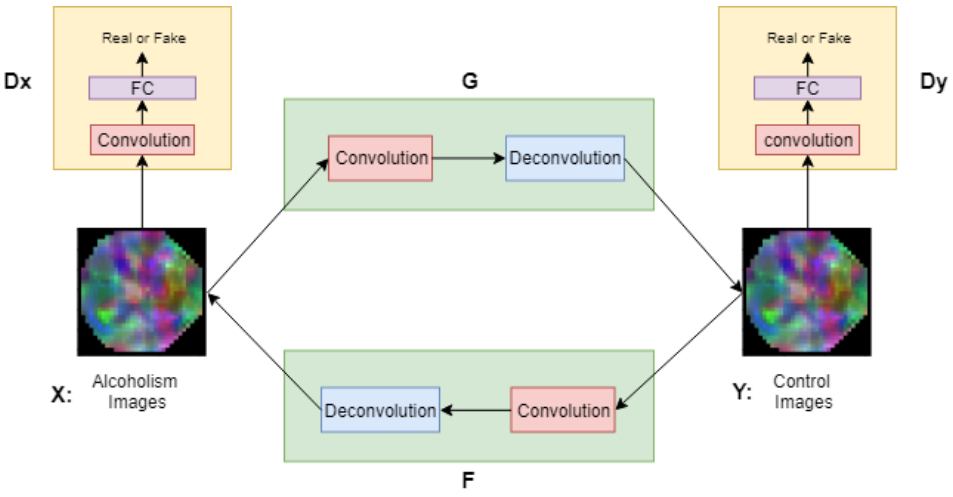

Figure. 10 Structure of GAN-based Autoencoder

The GAN-Based Autoencoder is mainly used for data filtering. The GAN-Based Autoencoder is in the same structure as CycleGAN structure. We call it GAN-based Autoencoder mainly because it principally still in a data->latent representation->original data structure and use reconstruction loss. So in this autoencoder design, we take this latent representation as our filter result. As introduced before. The training procure can be split into two separate training loop and each loop has two separate loss. The detailed loss definition is as follows.

A. Adversarial Loss:

The adversarial loss is mainly designed to judge whether the coming image really belongs to a certain distribution. So take loop $X \to G(X) \to F(G(X))$ for example, it is designed to map distribution X to distribution Y using generator G. The adversarial loss for this loop is defined as:

$$\mathrm{L_{GAN}}(\mathrm{G}, \mathrm{D_Y}, \mathrm{X}, \mathrm{Y}) = \mathrm{E_{y\sim pdata(y)}}[\log\ \mathrm{D_Y}(\mathrm{y})] + \mathrm{E_{x\sim pdata(x)}}[\log\ (1 - \mathrm{D_Y}(\mathrm{G}(\mathrm{x})))] \tag{3}$$

It is just common GAN loss where G(x) is trying to fool the discriminator D_Y to make the generated image become more similar to image distribution Y. The similar adversarial loss is introduced for loop $Y \rightarrow F(Y) \rightarrow G(F(Y))$.

B. Autoencoder Loss:

The autoencoder loss (reconstruction loss) is mainly used as a regulation term to make sure the generated image is not from random selection because the target distribution could have multiple choice. The autoencoder loss will help the generator to choose a target image which also maintains some feature from the original image in order to help reduce the reconstruction loss. Also, take loop $X \rightarrow G(X) \rightarrow F(G(X))$ for example, It is defined as:

$$L_{AL}(G, F) = E_{x \sim pdata(x)}[||F(G(x)) - x||_1] \quad (4)$$

The Autoencoder Loss is just the same as common autoencoder mean squared loss to judge whether F(G(x)) is really like x or not. Similar autoencoder loss is introduced for loop $Y \rightarrow F(Y) \rightarrow G(F(Y))$ as well.

Shown in Fig. 10, in this paper, we are trying to use EEG images with alcoholism condition and then map it to an EEG image with the control condition. By doing this way, we hope we can get rid of alcoholism information from an EEG image while still maintaining its personal identity. Since our input image is still 32x32 colorful image, we are still using convolution and deconvolution as our generator and convolution with 1 fully connected layer as our discriminator.

4 Results and Discussion

Our experiment was to compare the classification accuracy using normal channel-wise autoencoders, shared weight channel-wise autoencoders, normal Image-wise autoencoders and shared weight Image-wise autoencoders. The code is written in python and pytorch. All experiments were done on an i5-7500 CPU, Nvidia GTX1050Ti, 8g RAM and Windows environment. Both autoencoders' versions are manually selected by the least test loss. Here is classification result after 1,000 epoch training for the classification task.

Method	Accuracy	Final Loss	Training Time
Shared weight Image-wise Autoencoders	0.897	0.00026	**132.99s**
Normal Image-wise Autoencoders	**0.904**	**0.00019**	150.68s

Table. 2 Comparison between Two Image-wise Autoencoders

Method	Accuracy
Normal Channel-wise Autoencoders	0.864
Shared weight Channel-wise Autoencoders	0.858
Normal Image-wise Autoencoders	**0.908**
Shared weight Image-wise Autoencoders	0.897
EEGNet [8]	0.878
SyncNet [4]	**0.918**
DE [17]	0.821
PSD [17]	0.816
rEED [18]	0.702

Table. 3 Classification Accuracy Comparison between Different Methods in Within-Subject Test

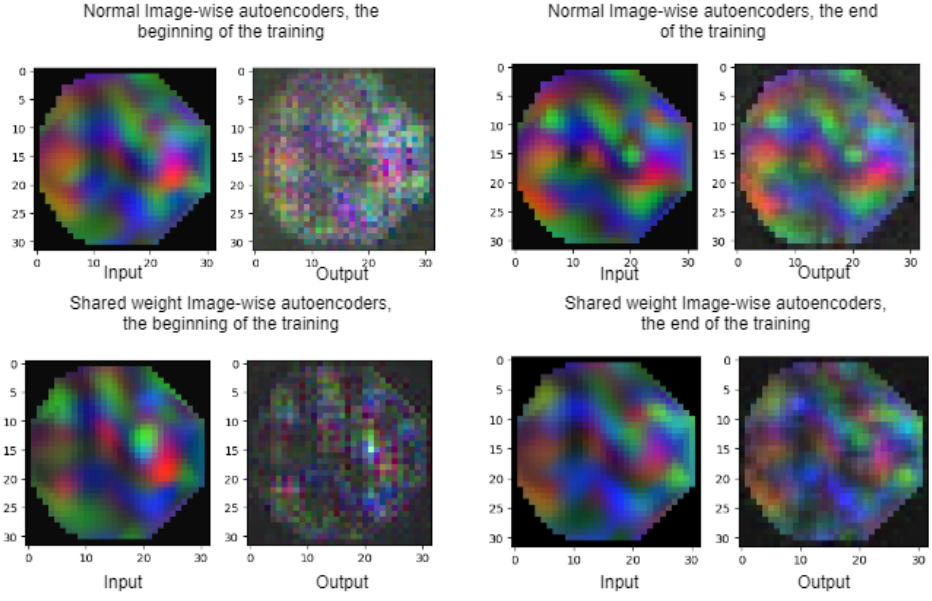

Figure. 11 Image-wise Autoencoders Performance

The accuracy of prediction on UCI EEG datasets, from a variety of methods, is given in Table 3. The accuracy of other methods is derived from Li's paper [4]. From the result above, we can see that the accuracy of our autoencoder based method is better than most of the past method except SyncNet [4] published last year. Image-wise autoencoders perform better than Channel-wise autoencoders while the normal autoencoders perform slightly better than shared weight autoencoders. From the Table 2, we can see the normal Image-wise autoencoder has better accuracy and less final test loss than the shared weight Image-wise autoencoders. Also from Fig. 11, the picture generated by normal Image-wise autoencoder is slightly clearer and more similar to the original image. But on the other hand, shared weight Image-wise autoencoders have less training time. This is an advantage of shared weight technique because it cuts half of the parameters. From all these results above we can see that the Image-wise autoencoders get the best discriminative features in our methods. We think this is because frequency-based feature learning method can obtain more discriminative information since both Image-wise Autoencoders and SyncNet are frequency based and they achieved the best performance. Also, all our autoencoder method achieve quite a high test accuracy. I think that because autoencoder can encourage feature extracted is not overfitting and it will prevent the model perform badly in the coming data.

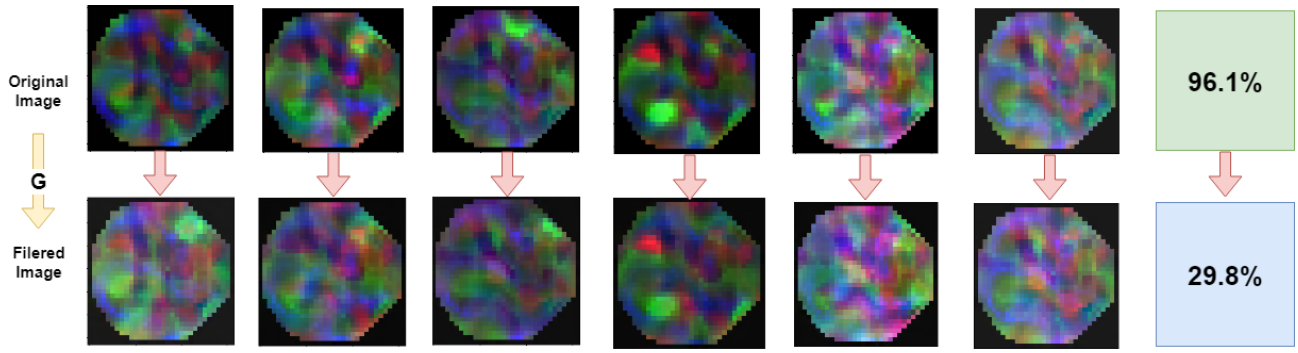

Figure. 11 GAN-based Autoencoders Performance

The picture generated by GAN-based autoencoder is shown in Fig. 11. These are six generation example randomly selected from all generation pairs. We can see our GAN works some slight modification to them but the fact that we cannot find interpretable features from these transformations so generated result from GAN cannot be manually checked. So we turn our eyes to digital indicators. So in here, we only evaluate whether our generated image is really getting rid of feature we don't want using the normal Image-wise autoencoder with a classification net. Form the Fig. 5. We can see that 96.1% original image is classified as alcoholism but after our GAN-based autoencoder only 29.8% image is classified as alcoholism. That is nearly 70% of images have their alcoholism information filtered out. Also, from the figure above, it seems that our GAN-based autoencoder do not change our EEG images too much but it is already filtered out one feature of the original EEG image. That is a part very interesting and further study is needed to figure out reasons. As a conclusion, It turns out our GAN-based autoencoder can filter out alcoholism information to some extent.

But the limitation of using alcoholism accuracy only is also very obvious because there could be various ways to make accuracy down like adding random noise. One potential solution to this is to check whether our autoencoder is still maintaining our desired information when doing filtering operation. But since we cannot get strong classifier designed for other features. It is not reliable to apply them for GAN evaluation. We have tried using our model to discriminate personal identity (120 classification problem). But it get quite low accuracy (nearly 10%) so it cannot be used to evaluate our GAN model.

5 Conclusion

Feature extraction for EEG data is very challenging because EEG signals contain a lot of noise. This report introduces three types of autoencoders. Two types of autoencoders is test for feature extraction and both of them achieve more than 85% accuracy. The experiment result demonstrates the autoencoder based feature learning is discriminative. Also, the shared weight technique can significantly reduce the training time but it may lead to a tiny discriminative information loss. GAN-based autoencoder is designed for feature filtering and it turns out that it can successfully filter out unwanted features. Limited by time, the potential of these models are not fully revealed, with further adjustment and fine-tuning, the accuracy will be higher.

6 Limitation and Future Work

Many further experiments can be done. First of all, the UCI dataset also contains other labels that can be classified. We should further test our extracted features on that to ensure our extracted features are discriminative for multiple task classification. Second, as mentioned before, we just test whether our GAN-based autoencoder can filter our unwanted information or not. We should also test when doing such filter operation, whether our desired information is really maintained or not. But all of this if based on whether we can get a strong classifiers on other labels or not. For the generator of GAN-based autoencoder, we can test U-net since it is the current state of art method for image translation. Then we can apply PCA for each channel to compare the result with our Channel-wise autoencoders. The objective of using PCA is to justify whether the linear transformation of features is worse for EEG feature extraction task or not. Then we can

turn our attention to more frequency based methods, since both Image-wise Autoencoders and SyncNet are frequency based. Finally, we may try LSTM based work because there also exists many RNN feature extractors for EEG data. If we have realized all these feature extraction methods, we will try to do more visualization procedures. Unlike pictures, the features of EEG signal are not obvious, so visualization will be a good choice for understanding EEG features. Our ultimate goal is to get a deep understanding of the EEG features and make it possible to design stronger feature extractors and classifiers.

7 References

[1] Gedeon, T. D., Catalan, J. A., & Jin, J. Image Compression using Shared Weights and Bidirectional Networks. *In Proceedings 2nd International ICSC Symposium on Soft Computing* (SOCO'97) (pp. 374-381).

[2] Stober, S., Sternin, A., Owen, A. M., & Grahn, J. A. (2015). Deep feature learning for EEG recordings. *arXiv preprint* arXiv:1511.04306.

[3] Bashivan, P., Rish, I., Yeasin, M., & Codella, N. (2015). Learning representations from EEG with deep recurrent-convolutional neural networks. *arXiv preprint* arXiv:1511.06448.

[4] Li, Y., Dzirasa, K., Carin, L., & Carlson, D. E. (2017). Targeting EEG/LFP Synchrony with Neural Nets. *In Advances in Neural Information Processing Systems* (pp. 4623-4633).

[5] Min, S., Lee, B., & Yoon, S. (2017). Deep learning in bioinformatics. Briefings in bioinformatics, 18(5), 851-869.

[6] Palazzo, S., Spampinato, C., Kavasidis, I., Giordano, D., & Shah, M. (2017, October). Generative Adversarial Networks Conditioned by Brain Signals. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3410-3418).

[7] Truong, N. D., Nguyen, A. D., Kuhlmann, L., Bonyadi, M. R., Yang, J., & Kavehei, O. (2017). A Generalised Seizure Prediction with Convolutional Neural Networks for Intracranial and Scalp Electroencephalogram Data Analysis. *arXiv preprint* arXiv:1707.01976.

[8] Lawhern, V. J., Solon, A. J., Waytowich, N. R., Gordon, S. M., Hung, C. P., & Lance, B. J. (2016). Eegnet: A compact convolutional network for eeg-based brain-computer interfaces. *arXiv preprint* arXiv:1611.08024.

[9] Schirrmeister, R. T., Springenberg, J. T., Fiederer, L. D. J., Glasstetter, M., Eggensperger, K., Tangermann, M., ... & Ball, T. (2017). Deep learning with convolutional neural networks for EEG decoding and visualization. Human brain mapping, 38(11), 5391-5420.

[10] Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). Deep learning (Vol. 1). Cambridge: MIT press.

[11] LeCun, Y., & Bengio, Y. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10), 1995.

[12] Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In European conference on computer vision (pp. 818-833). Springer, Cham.

[13] Hajinoroozi, M., Mao, Z., Jung, T. P., Lin, C. T., & Huang, Y. (2016). EEG-based prediction of driver's cognitive performance by deep convolutional neural network. *Signal Processing: Image Communication*, 47, 549-555.

[14] Thodoroff, P., Pineau, J., & Lim, A. (2016, December). Learning robust features using deep learning for automatic seizure detection. *In Machine Learning for Healthcare Conference* (pp. 178-190).

[15] Sykacek, P., & Roberts, S. J. (2003). Adaptive classification by variational Kalman filtering. *In Advances in Neural Information Processing Systems* (pp. 753-760).

[16] Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images.

[17] Zheng, W. L., & Lu, B. L. (2015). Investigating critical frequency bands and channels for EEG-based emotion recognition with deep neural networks. *IEEE Transactions on Autonomous Mental Development*, 7(3), 162-175.

[18] O'Reilly, D., Navakatikyan, M. A., Filip, M., Greene, D., & Van Marter, L. J. (2012). Peak-to-peak amplitude in neonatal brain monitoring of premature infants. *Clinical Neurophysiology*, 123(11), 2139-2153.

[19] Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. arXiv preprint.

[20] Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. arXiv preprint arXiv:1703.10593.

[21] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In Advances in neural information processing systems (pp. 2672-2680).

[22] Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.

[23] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

[24] Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images.

[25] Yu, L., Zhang, W., Wang, J., & Yu, Y. (2017, March). SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. In AAAI (pp. 2852-2858).

[26] Guo, J., Lu, S., Cai, H., Zhang, W., Yu, Y., & Wang, J. (2017). Long Text Generation via Adversarial Training with Leaked Information. arXiv preprint arXiv:1709.08624.

[27] Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.

[28] Lucic, M., Kurach, K., Michalski, M., Gelly, S., & Bousquet, O. (2017). Are GANs Created Equal? A Large-Scale Study. arXiv preprint arXiv:1711.10337.