Using PCA and Genetic Algorithm with Casper Neural Network on Alcoholism Classification

Bingchun Huo[1[U7197256]]

Australian National University

Abstract. The EEG signal data is an important data in the check of alcoholism, but the EEG data has very high complexity and contains some unrelated features, using Neural Network seems to be an effective method to solve this problem, this article tries to use a constructive cascade neural network with different optimizer to realize the analysis of the EEG signal and make a decision about the state of the subject, the SARPROP is used as the optimizer in the implement Neural Network, in order to figure out the most related features in the given data, Principal Component Analysis is used to decrease the dimension of the data. The genetic Algorithm is also used to selecct the best fit features to solve the classified problem.

Keywords: Cascade Neural Network · Casper Neural Network · SARPROP · Genetic Algorithm · Principal Component Analysis .

1 Introduction

Nowadays the neural network has been used in more and more aspect to help scientist solve problems. The EEG signal is an important data of the human brain, scientist can know if a person is in the state of alcoholism by analyze his EEG signal data, but the EEG signal data is complexity and the readability is not good, it may contains a lot of noise or unrelated features, and even the related features are hard to analyze, so using the neural network is necessary in this aspect, to realize this application, an constructive cascade neural network is used.

The CasPer algorithm designed to overcome the bad generalization property of CasCor (Cascade correlation algorithm). CasPer uses a Cascade architecture, like CasCor. It uses variation of RPROP, termed Progressive RPROP, to train the network. Hence the name CasPer. CasPer does not use a correlation measure or weight freezing but uses RPROP to train the whole network. It is Smaller networks than CasCor and get better generalization than CasCor.

The CasPer Neural network are separated to 3 regionsk, each with own learning rate, detailed see Fig. 1:

Region L1: Weights connecting to new neuron.

Region L2: Weights connected from new neuron to output neurons.

Region L3: Remaining weights (all weights connected to and coming from the old hidden and input neurons).

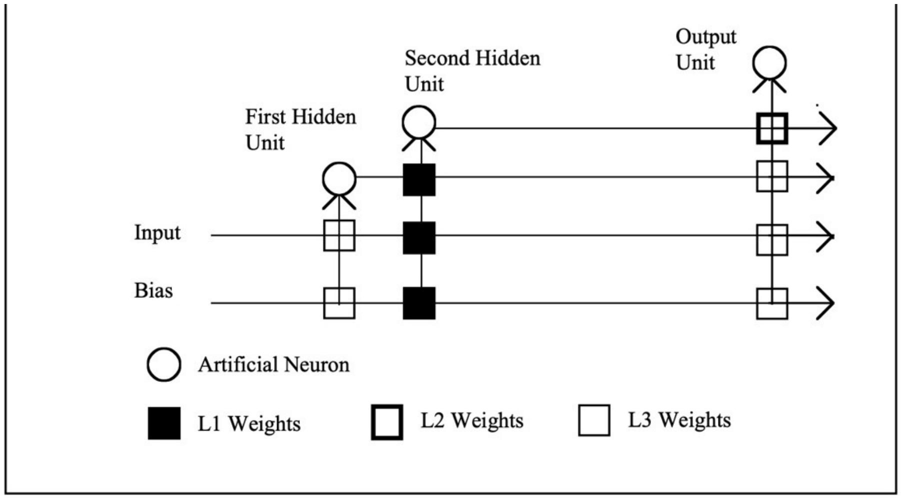

Fig. 1. CasPer Neural Network Architecture

. In my previous essay "Different Optimizer Performance in Constructive Cascade Neural Network on Alcoholism Classification", The SARPROP optimizer has been proved that is best suit for this task, so the main task now is improve the Nerual Net to gain better accuracy.

2 Methodology

In order to optimize and implement the Neural network, some control experiments are made to get the best result on the test data.

Casper The original Neural Network used SARPROP as the optimizer, use Cross Entropy function to calculate loss, and use the tanh function as the output activation function.detailed see Fig. 2:

Casper with PCA PCA Is the most commonly used linear dimension reduction method, its goal is to map the high-dimensional data to the low dimensional space through some kind of linear projection, and expect the maximum variance of the data in the projected dimension, so as to use less data dimensions and retain the characteristics of more original data sites.
Popular understanding, if all the points are mapped together, then almost all the information such as the distance between points are lost, and if the variance is as large as possible after the mapping, then the data points will be scattered, so

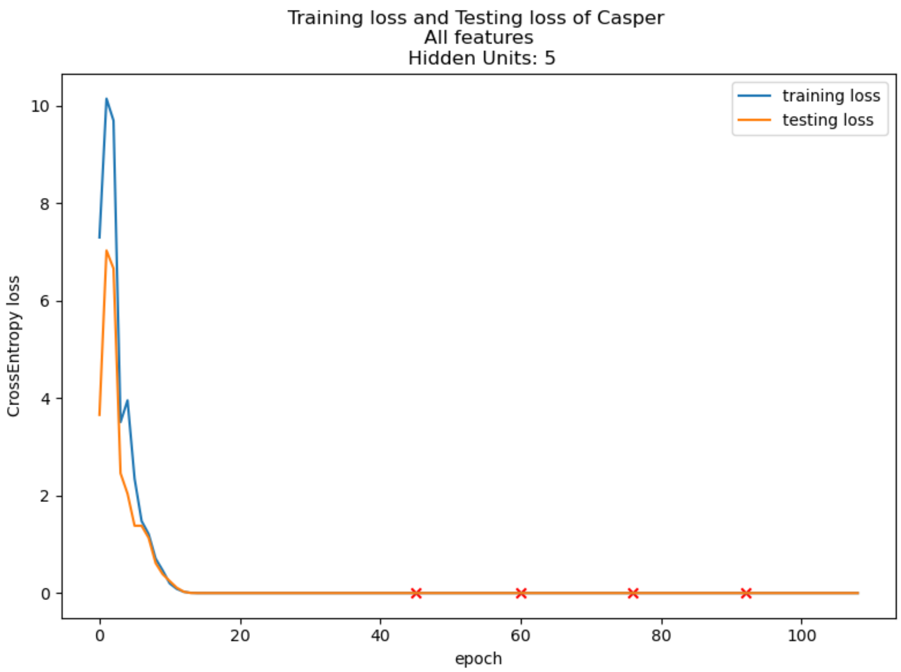

Fig. 2. Casper Model training loss without any dimension reduction.

as to retain more information. It can be proved that PCA is a linear dimension reduction method with the least loss of original data information. We use the PCA first because PCA can avoid lost too many features which caused infomation lost and training crashed, the training loss of the Casper model with PCA detailed in Fig. 3:

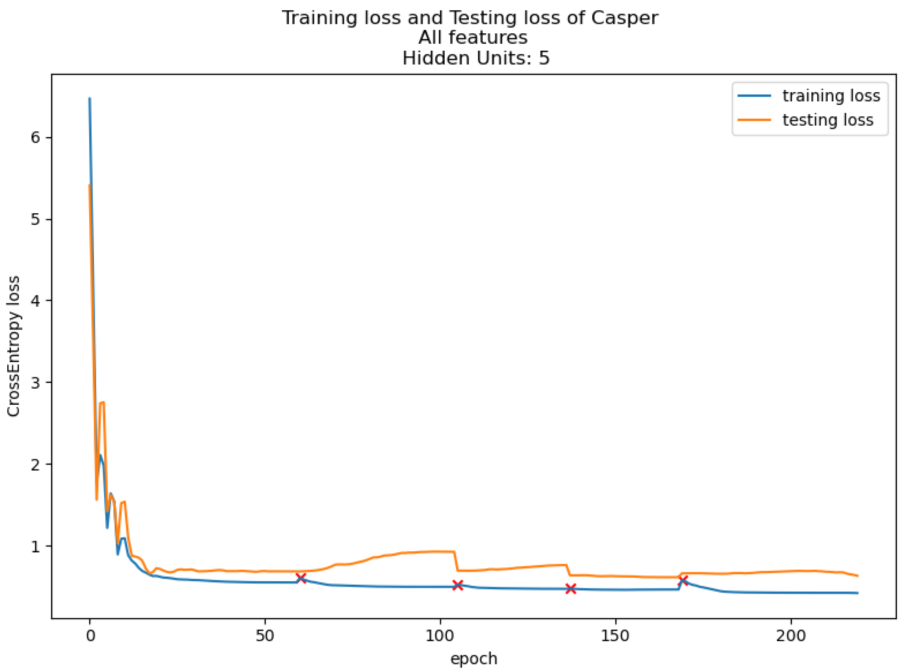

Fig. 3. Casper Model training loss with Principal Component Analysis as dimension reductiona Algorithm.

Casper with GA In order to select related features furthermore, The genetic algorithm is used. The main feature of GA is to operate the structure object directly without the limitation of derivation and function continuity; It has inherent implicit parallelism and better global optimization ability; The probabilistic optimization method can automatically obtain and guide the optimal search space without certain rules, and adjust the search direction adaptively.The GA architecture shows in Fig. 4:

The code realize the function of the GA is showing below

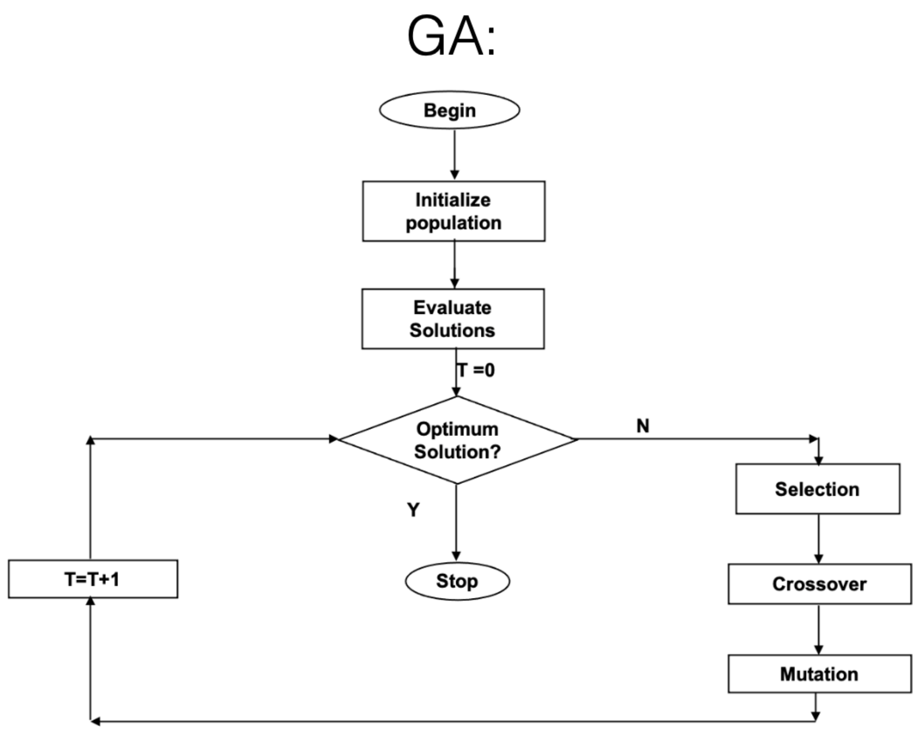

Fig. 4. GA architecture

```
def crossover(self, father, mother):
    """
    @param father: array
    @param mother: array
    @return: two children in list
    """
    if np.random.rand() < self.CROSS_RATE:
        mask = np.random.choice([0, 1], size=(self.DNA_SIZE,)
                                               )
        fa_and_mask = father & mask
        fa_and_not_mask = father & ~mask
        ma_and_mask = mother & mask
        ma_and_not_mask = mother & ~mask
        offspring1 = fa_and_mask | ma_and_not_mask
        offspring2 = fa_and_not_mask | ma_and_mask
        children = [offspring1, offspring2]
    else:
        children = [father, mother]
    return children
```

```
"""
    mutation
    @return: mutated child
    """
    for point in range(self.DNA_SIZE):
        if np.random.rand() < self.MUTATE_RATE:
            child[point] = 1 if child[point] == 0 else 0
    return child
```

```
def evolve(self):
    finish = False
    while not finish:
        self.reproduce()  # create new population
        self.old_candidate_pool = self.new_candidate_pool  #
                                               replace old pool

        self.fitness_function()  # compute fitness values
        self.rank_candidate()  # sort them
        self.update_fame()
        self.reset_new_pool()
        # update mutate
        self.generation_counter += 1
        self.update_mutate_rate()
        self.update_died_pool()

        # check termination
        finish = self.check_terminate()
    self.best_fame = list(self.hall_of_fame.items())[-1]
```

```
    self.best_fame_eval = self.fame_eval[self.best_fame[0]])
    arr = self.str_to_array(self.best_fame[0])
    num = np.sum(arr)
    return self.best_fame, self.best_fame_eval
```

The training results of GA is showing in Fig. 5:

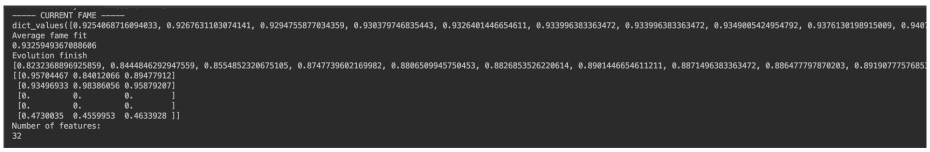

Fig. 5. A figure caption is always placed below the illustration. Please note that short captions are centered, while long ones are justified by the macro package automatically.

3 Data Prepare

Since compare to the original neural network, our new neural network applies for the alcoholism check got completely different input and output, so the data from the dataset need to be cleaning and preprocessing to fitting our net. For the original data, it is stored in a mat file, the following code shows the process of the data prepare, include load the data to pands.dataframe type and then save to xlsx format and transform the data type to torch.tensor.

```
#
def prePareData(data):
    """
    Change mat type data to xlsx format for next processing
    @param data: path of mat type data
    @return:
    """
    data1 = []
    for dataLines in zip(np.transpose(data['y_alcoholic']),np
                              .transpose(data['
                              y_stimulus']),np.transpose
                              (data['subjectid']),np.
                              transpose(data['trialnum']
                              )):
        a = [j.tolist() for j in dataLines]
        a1= []
        for dataPiece in a:
            a1.append(dataPiece[0])
        data1.append(((a1)))
    data_load = pd.DataFrame(data1,columns=('y_alcoholic','
                              y_stimulus','subjectid','
                              trialnum'))
```

```
    data2 = data['data']
    data2_load = pd.DataFrame(data2)
    all_data = pd.concat([data_load, data2_load], axis=1)

    outputpath = './eeg.xlsx'
    all_data.to_excel(outputpath, index=True, header=True)
```

```
#
gsr_data = pd.read_excel('eeg.xlsx').iloc[:, 1:]
all_ft_data = pd.concat([gsr_data], axis=1)

gsr_data = df_to_float_tensor(gsr_data)
all_ft_data = df_to_float_tensor(all_ft_data)
```

4 Findings

By comparing these 3 running results, when use the Casper model only, the loss of the model is unstabel, loss increase always happend, and the accuracy of the model is also unstable, in multiple runs the accuracy located in a interval of [0.5xxx 0.65xxx], which means many features in the input data is unrelated with our classification task.
For the running results of Casper model with PCA, the results is stabel than the pure Casper model, the training loss is decrease in most of the time and the multiple runs accuracy interval is smaller than pure Casper model, but by analysis the loss diagram, the loss is decrease rapidly which shows that the model probabaly overfitting.The Average accuracy of the Casper model with PCA is about 0.63.
The model combined the Genetic Algorithm and the Casper model will run continously until the result change really small, in our experiment, the GA shows that the related features is 32 in that run and it gives accuracy of 0.93 on the test dataset, the number of features is changed but it's keep loacted in interval [30-50].
According to the results of the GA with Casper model, we can get the index of the input data, this may shows that which EEG channel are related with the alcoholic predicting.

5 Future Work

By analyzing the results of models with differnets input, we found that some problems are still exist and caused problems. First of all, another important dimension reduction algorithm LDA is not used, for current version since the LDA need a reformat of the input data, due to the lack of time LDA is not considered, but in the future, this algorithm can also used to test the performance of the model.

The second problem is the potential overfitting risk of PCA, to solve this probelm, first an early stop procedure should be added. After that,the input data should be processed randomly to increase the data compelexity. At last, a new method named Label smoothing should be proposed to solve the above problems. It was first proposed in concept V2, which is a regularization strategy. By "softening" the traditional one hot type label, it can effectively suppress the over fitting phenomenon when calculating the loss value. As shown in the figure below, label smoothing is equivalent to reducing the weight of the real sample label category in calculating the loss function, and ultimately has the effect of restraining over fitting.
For GA model, in current code, the results shows is not very clear, for the data anlyst who is not goood at coding it's may be confusing. a more specific analysis should be output to provide EEG chanel information more obviously.

References

1. Author, F.:Treadgold, N Author, S.:Gedeon, Tom. (2006). A Cascade network algorithm employing Progressive RPROP. 10.1007/BFb0032532.
2. Author, F.:Khoo, Suisin ,Author, S.:Gedeon, Tom. (2008). Generalisation Performance vs. Architecture Variations in Constructive Cascade Networks. 236-243. 10.1007/978-3-642-03040-6-29.
3. Author, F.:Yao, Yue, Author,S.:Plested, Jo,Author, T.:Gedeon, Tom. (2018). Deep Feature Learning and Visualization for EEG Recording Using Autoencoders: 25th International Conference, ICONIP 2018, Siem Reap, Cambodia, December 13–16, 2018, Proceedings, .
4. Author, F.:Treadgold, N.K, Author,S.: Gedeon, Tom. (1998). Exploring architecture variations in constructive cascade networks. 1. 343 - 348 vol.1. 10.1109/IJCNN.1998.682289.
5. Author, F.:Treadgold, Nicholas, Author,S.: Gedeon, Tom. (1999). Exploring constructive cascade networks. IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council. 10. 1335-50. 10.1109/72.809079.
6. https://github.com/luoxiangyi222/Casper-Dynamic-NN.git
7. https://blog.csdn.net/weixin-41697507/article/details/89321586