Application of Bimodal Distribution Removal and GA-Based Feature Selection in Feedforward Neural Network for Multi-class Image Classification

Yi Huang

Research School of Computer Science, Australian National University
U6039034@anu.edu.au

Abstract. Neural networks are widely applied in the multi-class image classification problem. However, the noisy data and redundant features existing in the data set may result in learning problem and hence influence the performance of the neural network. Bimodal Distribution Removal (BDR) aims at cleaning up noisy training data set and hence improving generalisation. In addition, genetic algorithm (GA) could generate the optimal feature subset and eliminate the redundant features. This paper applies BDR and GA-based feature selection to a back-forward neural network designed for multi-class image classification problem respectively. The classification accuracy of the original network could reach 81.2% and the two improved models could achieve 85.5% and 84.9%. The evaluation and comparison conducted in this paper demonstrate both the effectiveness and usefulness of the two improved models.

Keywords: Bimodal Distribution Removal, GA-based feature selection, Feedforward Neural Network, Multi-class Image Classification

1 Introduction

Image classification, which refers to the process of categorizing images into different classes, is a significant computer vision area. Image classification is the fundamental task for image detection, image segmentation and other computer vision applications (Murugeswari & Suruliandi, 2016). Although this task seems simple and natural for humans, it is a challenging problem for an automatic system. This paper designs a feedforward neural network to solve multi-class image classification problem. However, on one hand, such a neural network as a non-parameter estimator may be very sensitive to the noise in the training data. Bimodal Distribution Removal (BDR) was proposed to reduce the noisy data in the training set and was applied for a regression problem (Slade & Gedeon, 1993). This algorithm outperformed Absolute Sriterion Method and Least Median Squares (Joines & White, 1992). Therefore, BDR is applied in this paper to improve the performance of the model. On the other hand, the irrelevant features existing in the data set would also negatively affect the classification. Li (2006) integrated the genetic algorithm (GA) with feature selection which outperformed floating search methods (Pudil, Novovičová & Kittler, 1994) and filter model (John, Kohavi & Pfleger, 1994). Thus, this paper applies GA-based feature selection to enhance the classification performance of the original network.

Statlog (Vehicle Silhouettes) data set (Siebert, 1987) is used to train and test the models proposed in this paper. This data set consists of 846 instances and 18 features. The number of instances and features is large enough to train and test a neural network. The features are extracted from the original images and most required information for classification are maintained. However, the extraction process cannot be perfectly accurate. Thus, the data set would contain noisy data and it is suitable for testing the influence of noisy data to the model introduced in this paper. In addition, some features of this data set seem redundant. For example, both compactness and circularity are the measurement of the area size from different perspectives. Hence, this data set is suited for experiments on the feature selection methods.

Given the data set, the classification problem can be described as to categorize data into 4 classes based on 18 features. I design a feedforward neural network and obtain two improved models by applying BDR and GA-based feature selection respectively. Then, I utilize precision, recall and F1 score to evaluate the original and two improved neural networks. Furthermore, I also compare the improved neural networks with other designed neural networks for the classification problem on the same data set. The evaluation and comparison results demonstrate the effectiveness of the BDR and GA-based feature selection in improving neural network performance.

2 Method

2.1 Data Pre-processing

The input attributes of the data set are integers and have different ranges. For example, the radius ratio ranges from 104 to 333 while the max length aspect ratio only ranges from 2 to 55. I use z-score to normalize the input attributes. The z-score of each raw value can be calculated as

$$z = \frac{x - \mu}{\sigma} \tag{1}$$

where μ and σ are the mean and standard deviation of all values of the attribute. The normalization could ensure all values have the same range and remove bias of lowest and highest values. Besides, I use decimal values to encode the original string representation of the class labels. That is, the four labels (VAN, SAAB, BUS and OPEL) are encoded as 0 to 3, respectively. Then, I randomly divide the data set into 10 pieces to validate the proposed models in this paper using 10-fold cross-validation. The details of validation would be discussed in Section 3.1.

2.2 Neural Network Design

After preprocessing the data set, I design the feedforward neural network trained using back-propagation. This process includes the choices of neural network layer, activation function, loss function and optimizer.

2.2.1 Neural Network Layer

There are 18 input neurons corresponding to the 18 input attributes and 4 output neurons corresponding to the 4 predefined classes. I design a two-layer network and a three-layer network for comparison. The two-layer network contains 18 neurons in the hidden layer and the three-layer network contains 20 and 10 neurons in two hidden layers. According to observations on the training results, there do not exist significant differences of both the training speed and validation accuracy. However, the two-layer network is more stable because it has less fluctuation of the validation accuracy. Thus, I choose the two-layer network.

2.2.2 Activation Function

For each hidden neuron, I use a Rectified Linear Unit (ReLU) as activation function. The function can be described as

$$f(x) = max(x, 0) \tag{2}$$

If the input is less than 0, a ReLU will output 0; otherwise, it will output the same as the input. Compared with sigmoid function, the ReLU can reduce the likelihood of the gradient to vanish. Research demonstrates that ReLU activations could increase training speed especially for large networks (Krizhevsk, Sutskever & Hinton, 2012). According to experiments, I find the ReLU could not only decrease the required training time but also increase the validation accuracy. Therefore, I choose the ReLU as the activation function for hidden neurons.

As for output neurons, I use softmax function as the activation function. The mathematic expression of softmax is shown below, z is the input vector to the output layer, K is the number of dimensions of the vector, j is the index of each dimension of z.

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \tag{3}$$

The softmax function calculates a probability distribution over the K dimensions. In the multi-class classification problem, softmax function could convert the output scores of the hidden neurons to the probabilities of classes given the input pattern. The class with the highest probability would be predicted as the category of the input pattern.

2.2.3 Loss Function

For each training epoch, it is required a loss function to measure the error between the predictions of the neural network and the pre-labelled classes. I use the Cross Entropy Loss Function, by which the error is calculated as the average over all losses. Given a prediction x and a label *class,* the loss is:

$$loss(x, class) = -log\left(\frac{exp(x[class])}{\sum_j exp(x[j])}\right) = -x[class] + log\left(\sum_j exp(x[j])\right) \tag{4}$$

In the multi-class classification, the output layer would produce probability distributions and I expect the correct class could have the highest probability. As the above expression (4) shows, the Cross Entropy could mainly focus on how much probability is assigned to the correct class. Hence, Cross Entropy Loss Function is a good measure for the error in the classification problem.

2.2.4 Optimizer

As Section 2.2.3 explains, the loss function can be considered as the difference between the neural network and the expected results. Optimizers are used to optimize the value weights of neural network and hence to minimize the losses. I test different momentums with the Stochastic Gradient Descent (SGD) optimizer proposed by Sutskever, Martens, Dahl and Hinton (2013) and the Adam optimizer designed by Kingma and Ba (2014). Fig. 1 shows the decrease of loss over steps during training with different optimizers. Adam optimizer requires less training time to achieve same results than SGD optimizer with setting momentum to 0 and 0.9. Furthermore, I find the validation accuracy of Adam optimizer also is higher than the other two optimizers. Based on the observations, the Adam optimizer is used in the feedforward neural network.

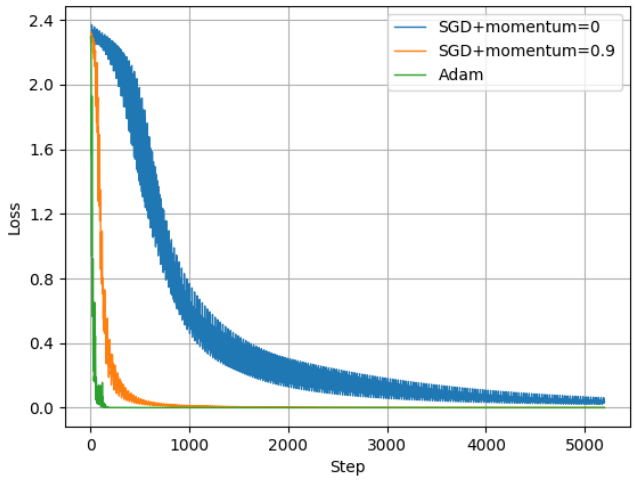

Fig. 1. Comparison of optimizers.

2.3 Bimodal Distribution Removal

Similar to the bias-variance decomposition of regression problem (Gedeon, Wong & Harris, 1995), the error of classification problem can also be decomposed into bias and variance (Le Borgne, 2005). Simply speaking, bias is the error comes from erroneous assumptions and variance is the error derives from sensitivity to input. A typical issue results from the variance is that the neural network may spend plenty of time training on noisy data and converge to incorrect results. Bimodal Distribution Removal algorithm proposed by Slade and Gedeon (1993) is used to reduce noise and therefore the variance of training data set for regression problem. Based on the above analysis, I find classification problem have the same issue and hence apply Bimodal Distribution Removal to the designed neural network.

As introduced in Section 2.2.3, Cross Entropy is a good measure for the error of classification problem. I use the Cross Entropy to calculate the error instead of the mean square error which is suited for regression problem. The error distribution before training demonstrates the large variance of errors. However, after only 200 epochs, the neural network dramatically reduces the errors. The error distribution is bimodal, which means that the neural network has learnt the patterns of the low peak well while the patterns of the high peak are outliers.

Bimodal distribution removal begins with training on the entire training data set. After 200 training epochs, if the normalization variance of errors over the whole training data set v_{ts} is less than 0.1, the neural network would have reached the bimodal error distribution and patterns can be removed. Extract the patterns whose errors are greater than the mean error $\bar{\delta}_{ts}$ for all patterns to form a subset. Calculate the mean error $\bar{\delta}_{ss}$ and the standard deviation σ_{ss} of the subset. Within the subset, remove the patterns whose errors are greater than $\bar{\delta}_{ss} + \alpha\sigma_{ss}$ from the training set, where α is a customized parameter based on the training data set and its range is from 0 to 1. After each removal, the neural network need to train on the new training data set for 50 epochs. Repeat removal and training until the normalization variance of errors over the whole training data set v_{ts} is below 0.01. The choices of the customized parameter α and the threshold for the normalized variances depend on the choice of data set.

2.4 GA-Based Feature Selection

The irrelevant and redundant features in the data set might unduly complicate the learning task and therefore negatively influence the performance of the neural network. Yang and Honavar (1998) demonstrate that the choice of features could affect several classification aspects, including the time needed for learning from the training data, the classification accuracy and the number of instances required for learning. Research shows the ability of genetic algorithm to obtain the

optimal feature subset and hence enhance the performance of neural network (Sharma & Gedeon, 2013). Thus, I apply GA-based feature selection to the original designed network and the procedure is explained as follows.

2.4.1 Define Chromosome Representation and Initialize Population

In GA-based feature selection, a chromosome represents a feature subset. Fig 2 illustrates the binary representation of the chromosome. F1 to F18 are the 18 input features and the corresponding binary value indicates whether the feature is selected. For example, "0" of F2 indicates F2 is not in the feature subset while "1" of F5 indicates F5 is in the feature subset.

F1	F2	F3	F4	F5	…	F18
1	0	0	0	1	…	1

Fig. 2. Chromosome binary representation example.

At the beginning, each gene of each chromosome is assigned a random value from the allowed domain (0 and 1 for binary representation) to form the initial population. The goal of random assignment is to ensure that the initial population is a uniform representation of the whole search space. The choice of the size of the population depends on the problem.

2.4.2 Evaluate Fitness Values

The fitness value of each chromosome in the population determines the quality of the candidate feature subset. In GA-based feature selection, classification accuracy and the number of selected features are the two criteria used to design a fitness function. The fitness value can be calculated as

$$fitness = accuracy + \alpha \times \frac{F_t}{F_s} \tag{5}$$

where $accuracy$ is the classification accuracy, F_t is the total number of features, F_s is the number of subset features and α is a tuned parameter to compromise between maximizing classification accuracy and minimizing the number of features in the subset.

In the classification problem, every chromosome in the population is used as the input features for the original neural network designed in Section 2.2. The number of input neurons should be modified corresponding to the feature number of the subset and the rest of the network remains the same. Classification accuracy is obtained by training and testing on the new model. Then, each chromosome could be evaluated by the aforementioned fitness function.

2.4.3 Apply Crossover and Mutation to Population

According to the fitness values of chromosomes, crossover is applied to population to reproduce new population. I use proportional selection to select the parents for crossover. That is, the chromosome with higher fitness value has higher chance to be selected as the parent. Then I utilize uniform crossover to attempt to combine the good genes in the parents. As Fig. 3 shows, the offspring is produced by combining the two parents using the randomly created mask.

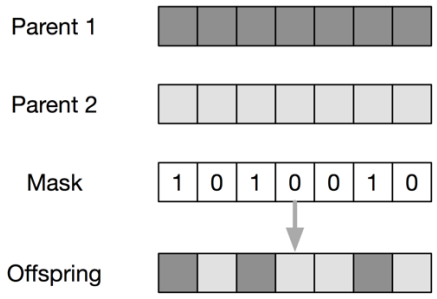

Fig. 3. Uniform crossover.

In addition to crossover, I apply random mutation to introduce new genetic material into the population and hence increase genetic diversity. Mutation is applied at a certain probability to each gene of the offspring produced by crossover. The aim of setting the probability, also referred to as the mutation rate, is to ensure that good chromosomes are not distorted too much. The choice of the mutation rate is problem-dependent.

2.4.4 Select Chromosomes and Terminate Evolution

After applying crossover and mutation to produce new population, I use the fitness function explained in Section 2.4.2 to evaluate fitness values for each new chromosome in the new population. Based on the fitness values, proportional selection is applied to select chromosomes that will proceed to next generation. In other words, the chromosome with higher fitness value has higher probability to survive to next generation.

Furthermore, I set two termination criteria of the evolution. One is reaching the limit of 1000 generations. Another is that search is converged where no improvement is observed over the last 10 consecutive generations. Repeat Section 2.4.3 and Section 2.4.4 until any of the two termination conditions is satisfied.

3 Results and Discussion

3.1 Evaluation Method

Precision, recall and F1 score are common evaluation metrics for classification problem. Precision is a ratio of positive predictions to the total number of positive class values predicted and hence could indicate how often the model is correct. Recall is a ratio of positive predictions to the number of positive class values in the test data and therefore could reflect the effectiveness of a model to identify class labels. The two evaluations could gauge the performance of the neural network from different perspectives. F1 score is the harmonic mean of precision and recall. If either precision and recall is low, the F1 score will be low. Hence, I use the three evaluation metrics to measure the performance of the neural network. The three evaluation values could be easily calculated from the confusion matrix.

In addition, 10-fold cross validation is used to validate models. As Section 2.1 introduces, the data set is randomly divided into 10 pieces. I treat one of the pieces as the test data and fit the model to the other nine-tenths of the data. Then, apply the model to the test data and calculate the aforementioned three evaluation values. Repeat this procedure for all 10 pieces of the data and average the evaluation values derived in all ten cases of cross validation. 10-fold cross validation has many advantages. First, 10-fold cross validation could check whether the model is overfitting. If the performance metrics at each of the 10 times are close to each other that could prove the model is not overfitting. Second, the influence caused by the randomness of the initial weight values for the neural network could be reduced. Finally, 10-fold cross validation could diminish the possible negative effect derived from splitting training and test data.

3.2 Evaluation of the Original Network

As explained in Section 3.1, I utilize 10-fold cross validation to assess the performance of the designed neural network. The variance of the evaluation results for 10 folds is low, which proves the model does not overfit. Table 1 shows the average evaluation results of the designed neural network over the ten folds. All of the three average evaluation values are above 0.8. High precision proves the correct prediction ability of the neural network while high recall reflects the effectiveness of the neural network. Hence, these evaluation results demonstrate that this neural network performs well in this classification problem. However, the evaluation values of some classes are relatively lower such as class 1 and class 3. This may result from the presence of noisy data and redundant features in the data set.

Table 1. Evaluation of the original network.

Class	Precision	Recall	F1 score
0	0.94	0.97	0.95
1	0.70	0.68	0.68
2	0.96	0.97	0.97
3	0.66	0.66	0.65
Average	0.82	0.82	0.81

3.3 Evaluation of Bimodal Distribution Removal

Similar to the evaluation of the original network, 10-fold cross validation is used and low variance of evaluation values for 10 folds demonstrates the model is not overfitting. Table 2 shows the average evaluation results of the neural network combined with Bimodal Distribution Removal. Compared with table 1, the evaluation values of most classes have increased and hence the overall performance has improved. This improvement reflects the positive influence of the application of bimodal distribution removal to the original neural network.

Table 2. Evaluation of bimodal distribution removal.

Class	Precision	Recall	F1 score
0	0.96	0.97	1.00
1	0.73	0.72	0.81
2	0.98	0.98	1.00
3	0.72	0.72	0.69
Average	0.85	0.85	0.87

Besides, the average normalized variance of prediction errors of the original training data set is 0.82 while the variance of the data set after Bimodal Distribution Removal is only 0.008. The Bimodal Distribution Removal could remove the noisy data in the training set hence reduce the variance of the training set. The performance of the improved neural network shows the positive effect of Bimodal Distribution Removal on cleaning training data set and improving performance of neural networks. However, the average size of training data set after Bimodal Distribution Removal is 746 while the original size is 762. Only a small part of the training data has been removed which indicates the original training data set does not contain too much noise. Because the performance of Bimodal Distribution Removal is largely influenced by the choice of the data set, this algorithm might perform better on other nosier data set.

3.4 Evaluation of GA-Based Feature Selection

Similar with the evaluation explained before, I use 10-fold cross validation and experiment results prove the model with GA-based feature selection is does not overfit. Table 3 summarizes the average evaluation values over the 10 folds. As Table 3 shows, evaluation values of major classes are higher and therefore the average performance is better compared to Table 1. In addition, the size of the feature set decreases from 18 to 15. The decrease of feature numbers and increase of evaluation results demonstrate the ability of GA-based feature selection to eliminate redundant features and improve the original neural network. Furthermore, I conduct the experiment 3 times and I find two features would always be removed. Thus, these two features could be considered as noisy features. However, there are always only 3 features removed from the initial set. This means the original feature set does not consist of too many irrelevant and redundant features.

Table 3. Evaluation of GA-based feature selection.

Class	Precision	Recall	F1 score
0	0.95	0.98	0.96
1	0.73	0.72	0.72
2	0.97	0.97	0.97
3	0.73	0.72	0.72
Average	0.85	0.85	0.84

According to Table 3 and Table 2, there are no significant differences in the evaluation values between Bimodal Distribution Removal and GA-based feature selection. Because the two methods attempt to improve the model by processing the data set from different perspectives, it is not reasonable to conclude which method is better.

3.5 Comparison with Other Networks

Parekh, Yang and Honavar (2000) also solve the classification problem on the same data set through a basic neural network and two constructive networks using MPyramid-real algorithm and MTiling-real algorithm respectively. These three neural networks are used as baseline to evaluate the proposed models by this paper. Parekh, Yang and Honavar (2000) utilize test accuracy as the evaluation metric. Table 4 shows the comparisons of feature number, hidden neuron number and test accuracy among the three baseline networks and the two improved models. The test accuracy of the two improved models are higher than the baseline. These results show that the two improved models outperform the others. This may result from four reasons. First, the basic neural network does not use enough hidden neurons to learn from the training. Second, the model constructed by MPyramid-real algorithm contains too many hidden neurons and hence is overfitting. Third, the improved network by Bimodal Distribution Removal could lessen the noise in the training data and enhance the performance of the neural network. Finally, the model after GA-based feature selection removes redundant features that unduly complicate the learning task and hence improves the performance.

Table 4. Comparison of neural networks

Parameter	Basic NN	MPyramid-real	MTiling-real	BDR+NN	GA+NN
#features	18	18	18	18	15
#hidden neurons	4	35	19	18	18
Test accuracy	79.7%	78.2%	77.5%	85.5%	84.9%

4 Conclusion and Future Work

The application of Bimodal Distribution Removal and GA-based feature selection in feedforward neural network could improve the performance in solving multi-class image classification problem. The evaluation and comparison demonstrate that Bimodal Distribution Removal is a feasible method to reduce noise and hence the variance of training data. The evaluation and comparison also prove that genetic algorithm is able to eliminate redundant features and therefore to improve the classification performance.

Future work could be carried out towards four directions. First, Bimodal Distribution could be used on some nosier data set to further evaluate the effectiveness. The second is applying Bimodal Distribution Removal to other type of classifiers that may encounter bias-variance tradeoff such as support vector machines. Third, some other data set containing more irrelevant features can be utilized to evaluate the improvement by GA-based feature selection. Finally, genetic algorithm could be applied to determine not only the optimal feature subset but also the optimal neural network structure such as the number of hidden neurons.

References

Gedeon, T. D., Wong, P. M., & Harris, D. (1995). Balancing bias and variance: Network topology and pattern set reduction techniques. *International Workshop on Artificial Neural Networks,* 551-558.

Le Borgne, Y. (2005). Bias-variance trade-off characterization in a classification problem: What differences with regression. *Machine Learning Group, Univ. Libre de Bruxelles, Belgium*.

Li, T. S. (2006). Feature selection for classification by using a GA-based neural network approach. *Journal of the Chinese Institute of Industrial Engineers, 23*(1), 55-64.

John, G. H., Kohavi, R., & Pfleger, K. (1994). Irrelevant features and the subset selection problem. *Machine Learning Proceedings,* 121-129.

Joines, J. A., & White, M. W. (1992). Improved generalization using robust cost functions. *Neural Networks International Joint Conference,* (3), 3911-918.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems,* 1097-1105.

Murugeswari, G., & Suruliandi, A. (2016). Fuzzy based roughness feature for image classification and segmentation. *International Journal of Tomography and Simulation, 29*(3), 48-63.

Parekh, R., Yang, J., & Honavar, V. (2000). Constructive neural-network learning algorithms for pattern classification. IEEE Transactions on neural networks, 11(2), 436-451.

Pudil, P., Novovičová, J., & Kittler, J. (1994). Floating search methods in feature selection. *Pattern recognition letters, 15*(11), 1119-1125.

Sharma, N., & Gedeon, T. (2013). Hybrid genetic algorithms for stress recognition in reading. *European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, 117-128.

Siebert, J. P. (1987). Vehicle recognition using rule-based methods. Turing Institute, Glasgow, Scotland.

Slade, P., & Gedeon, T. D. (1993). Bimodal distribution removal. *International Workshop on Artificial Neural Network,* 249-254.

Sutskever, I., Martens, J., Dahl, G., & Hinton, G. (2013). On the importance of initialization and momentum in deep learning. *International conference on machine learning,* 1139-1147.

Yang, J., & Honavar, V. (1998). Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems, 13*(2), 44–49.