Optimization of Neural Network Training for Wine Quality Classification Using Simulated Annealing

Mingfei Duan*

The Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hongkong, China

ABSTRACT: The backpropagation algorithm is the most used algorithm to train a neural network. However, a simulated annealing algorithm can do that work too. This paper shows the process and results of training neural network by applying simulated annealing algorithm. A Wine Quality Dataset is used to do the experiment. The experiments first extracted the data by feature selection and pre-processing of the dataset. By applying the Principal Component Analysis method, the features in the original data are extracted into a lower-dimensional space. The importance of features will increase significantly, and there is a positive effect on the training of neural networks. Then a variety of neural networks with different structures are constructed and trained with simulated annealing and back propagation respectively. More specifically, neural networks with two-hidden-layer fully-connected neural networks with two, three, and four hidden nodes in each layer are constructed to represent the different architecture of the network. Finally, their respective prediction results are compared to get a conclusion. This paper uses four parameters, accuracy, precision, recall and F1 score respectively, to evaluate the performance of the two target models, in addition to measure their performance in a more holistic way. As a result, the simulated annealing algorithm performs better than the backpropagation method in the context of wine quality classification.

1. INTRODUCTION

Artificial neural networks are consisted of some simple neurons, which are used to simulate some functions of the brain. The architecture of a neural network has a significant impact on its performance. A smaller number of connections may not allow the model to achieve the expected results, while too many connections not only increase the consumption of the training process but also lead to overfitting and reduce the accuracy of the prediction model.

The performance of architecture can be considered as a surface in a multidimensional space composed of its parameters, and finding the optimal architecture is a matter of finding the lowest point in it.

The simulated annealing approach has been widely and successfully used in the machine learning area [2]. In this paper, the main goal is to compare the results of neural network trained by simulated annealing and that trained by back propagation algorithm.

This article will be developed in the following way, starting with a description of how the data is pre-processed and how features are extracted. Then the traditional simulated annealing algorithm will be introduced and the way it is applied to the neural network training problem will be presented. Then, the backpropagation approach used in this paper is presented as a comparison to the simulated annealing approach. After the results are compared, the paper will discuss and summarize the results, and finally, draw the experimental conclusions.

2. METHODOLOGY

2.1. Description of dataset

In this work, the dataset used to train the model is downloaded from Internet [3]. The dataset describes the amount of various chemicals in red wine and its quality. The input variable includes features with fixed acidity, volatile acidity, citric acid, sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, and alcohol. The aim of the neural network work is to predict the quality of a given red wine based on its given various chemical attributes. Preprocessing work is done before the model training to normalize the data. In this work, a robust scaler is used to normalize the data. Its rule is to normalize the data based on the median and quartile of the data. It is possible to effectively scale data with outlier by Robust. If the data contains outliers, they will be rounded off in the scaling. Therefore, using this regularization makes the training process more stable. And to initialize the weights of the network, Xavier initialization is used to get better performance of the model. The basic idea of the Xavier initialization is to keep the variance of the inputs and outputs the same so that all output values do not converge to zero. This can prevent the gradient from converging to zero during backpropagation training, which makes it

*Corresponding author. Email: 20099033d@connect.polyu.hk difficult to update the parameters in the network. In this work, the structure of the neural network is set as a fully connected network with two hidden layers, which can fit the data better. The activation function in the neural network is chosen as ReLU, which is one of the most effective activation functions [4].

2.2. Data preprocessing

Data preprocessing can revise or drop the data before it is used to train the neural network, to ensure or enhance performance. It is also an important step in the data mining process and can ensure the model to make better use of the original data.

After applying data cleaning and normalization, Principal Component Analysis (PCA) method is applied to extract the original data. PCA can find a linear projection of original features in a lower-dimensional space that can maximize the cumulative variance or minimize the cumulative error. In other words, it divides the original data by 'feature importance' criteria. It uses an orthogonal transforming method to linearly transform a series of correlated variables into another series of uncorrelated variables. This operation can be regarded as revealing the internal information of the data, thereby better showing the distribution of the data. If a multivariate dataset is represented by a coordinate system in a high-dimensional data space, then PCA can provide a lower-dimensional image, equivalent to a projection of the dataset on the most informative angle. In this way, principal components of the original data can be used to reduce the dimension of the data, and also can make the most of the original information.

As PCA simply transforms the input data, it can be applied both to classification and regression problems. PCA was set up to reduce the dimensionality and then select the most relevant features by finding the maximum system variability.

2.3. Introduction of the simulated annealing algorithm

The simulated annealing algorithm consists of a series of iterations. In each iteration, the algorithm will change the current solution to a new neighborhood of it. Once the new solution is created, we compare the cost of it to decide whether we accept the new solution to replace the former solution. If the cost of the new solution is less than that of the previous solution, then we can accept the new solution directly. Otherwise, it is accepted according to Metropolis's criterion [5]: if the new solution costs more than the original solution, a random number is generated, which will decide whether the new solution should be accepted or not. If the random number is less than a special value $\exp(-\Delta E/T)$, where ΔE is the change in terms of cost and T is the current temperature, then the new solution is accepted and will replace the original solution. If not, the new solution will be rejected and the current solution keeps unchanged. Before the iterations, initial temperature is set to be T0, and in the iterating process, the temperature will keep reducing until it is reduced to zero.

To implement the simulated annealing algorithm for training of the neural network of the red wine classification problem, the following four principal choices that must be made:

The first is the representation of solutions X. We assume that the initial state is a fully connected neural network, where each connection has a connectivity property, when it is 1 it means that the connection exists in the current neural network model, and on the opposite when it is 0 it means that the connection does not exist. In addition to the topology of the neural network, the initial weights of the network will also have an impact on the training results, so each connection will also be assigned a parameter w to represent its weights.

Second, the definition of the cost function E should be determined, too. To maximize the accuracy of the model, the cost function needs to use the classification error as a parameter. And, to minimize the complexity of the model, the total number of connections used needs to be used as one of the parameters, too. Most importantly, we need to ensure that the generated neural network structure is valid, when the solution is an invalid network, a new neighbor solution is generated.

The third is the definition of the generation mechanism for the neighbors Δx. The new neighborhood solution can be generated as follow: first change the connectivity bit according to a given probability like 20%. This can change the topology of the neural network. Then add a random number between -1 and 1 to the weight w to change the connection weights. After applying these two steps, a neighborhood of the current solution will be generated.

Finally is the definition of the cooling schedule ΔT. The cooling scheme for simulated annealing generally adopts a stepwise cooling strategy in which the geometric cooling strategy sets the cooling rate α, where α is less than but close to 1 and the new temperature will be equal to the previous temperature multiplied by α.

In this work, the geometric cooling rule is applied. The initial temperature is set to be 1, and the cooling rate α is set to be 0.9, and the temperature decreases every 20 iterations. The limit of iteration is set to be 1000.

2.4. Introduction of the backpropagation algorithm

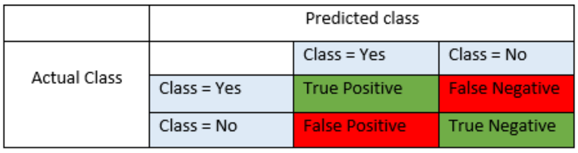
The main process of the backward propagation training method is that the error is obtained from the forward incoming input value, and then the error is backpropagated according to the error to obtain the error value of each neuron, and finally, the weights are updated according to the inverse of the error value and e, which achieves the correction of the whole network.

In this work, the hyperparameter learning rate is set as 0.001, and the limit of iteration is set to be 5000. Activation function ReLU is applied to the network.

3. RESULTS

In this work, three different architectures were set as initial topological graph: two-hidden-layer fully-connected neural networks with respectively two, three, and four hidden nodes in each layer, all nodes in each layer have connections with all the nodes in adjacent layers. Robust scaler is used to normalize the data, and uses dropout with a probability of 80% to prevent overfitting. For each initial network, 10 different random weight initializations were used, and the initial weights were generated by using Xavier initialization. For each initial network, 30 training of simulated annealing were applied. The best and the worst 10 runs were excluded, and remains 10 runs which were considered for the results.

Four indicators, accuracy, precision, recall and F1 score are used to evaluate the performance of each model [6].



	Predicted class		
Actual Class		Class = Yes	Class = No
	Class = Yes	True Positive	False Negative
	Class = No	False Positive	True Negative

Figure 1. Model result

True positive and true negatives are the observations that are correctly predicted and therefore shown in green. False positives and false negatives should be minimized, so they are shown in red color.

Accuracy is the most intuitive performance measure, and it is simply a ratio of correctly predicted observations to the total observations.

Accuracy = TP+TN/TP+FP+FN+TN.

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

Precision = TP/TP+FP.

The recall is the ratio of correctly predicted positive observations to all observations in the actual class.

Recall = TP/TP+FN.

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account.

F1 Score = 2*(Recall * Precision) / (Recall + Precision).

These four indicators can show the performance of a specific model, in most cases, the larger these indicators are, the better performance the model hhas.

Table 1. Confusion matrix

Classifier algorithm	Precision	Recall	F1 Score	Accuracy
2 nodes each layer + backpropagation	61.54	78.08	68.83	75.76
2 nodes each layer + SA	66.91	67.77	67.34	67.20
3 nodes each layer + backpropagation	87.73	88.21	87.97	87.47
3 nodes each layer + SA	89.30	98.30	94.16	90.53
4 nodes each layer + backpropagation	89.93	89.50	89.67	90.20
4 nodes each layer + SA	96.0	98.72	96.54	96.32

Comparing these results, by comparing the precision, recall, F1 score, and accuracy value of each model, one can see in all number of nodes in each layer, training the neural network with simulated annealing performs better than that using backpropagation. Therefore, in the context of the wine classification problem, simulated annealing algorithm performs much better in finding minimal network architectures work than network trained by the backpropagation algorithm.

4. DISCUSSION

The reasons for the superiority of the simulated annealing approach over the backpropagation algorithm under these conditions are broad as follows. Firstly, the traditional backpropagation algorithm has the problems of slow convergence and easily falling into the local optimum, and the traditional backpropagation algorithm uses instantaneous gradient descent to modify the weights, which utilizes less information and therefore easily falls into the local optimum solution. In contrast, the simulated annealing algorithm uses a global optimization approach, using global information to optimize the network and modify the weight vectors, so it can avoid falling into local minima. Secondly, the training speed of the backpropagation algorithm is extremely slow. As the backpropagation algorithm is essentially a gradient descent method, the objective function it has to optimize is very complex, and therefore the "sawtooth phenomenon" will occur, making the backpropagation algorithm inefficient. The backpropagation algorithm is also paralyzed, as the objective function is complex, it is bound to have some flat areas where the neuron output is close to 0 or 1, and in these areas, the weight error changes very little, making the training process almost stagnant, while the simulated annealing algorithm, after setting the initial temperature and step size, the number of iterations is confirmed, and in the training of simple networks, the time complexity of simulated annealing will you complexity would outperform the backpropagation algorithm.

However, simulated annealing still has many shortcomings when training more complex neural networks, first of all, the fixed training method, because its training method is more fixed, the simulated annealing method has fewer training methods, while the backpropagation algorithm has different update methods, so simulated annealing algorithm is more difficult to implement specific optimization for a specific problem, in the face of a specific problem, backpropagation algorithm will have better performance. Secondly, because the simulated annealing algorithm uses global information for iteration, it converges relatively slowly at larger network sizes, and its time advantage over the backpropagation algorithm becomes smaller due to the rapid increase in the number of iterations required. Theoretically, the algorithm always finds the global optimal solution, but for some problems, it runs very slowly and in practice it will become a problem to determine the rate of temperature reduction.

5. CONCLUSION

In this paper, results of neural network applied simulated annealing for the optimization of network weights and architectures have been presented and analyzed. It was shown that simulated annealing algorithm can train networks with lower complexity and better performance than the back propagation algorithm for the red wine classification work. It proves the simulated annealing algorithm in neural networking can be very important and can be applied in a wide range of applications.

One possible future work is the implementation of a simulated annealing algorithm for other artificial neural network tasks, to explore more possible uses of simulated annealing algorithm in the artificial intelligence area.

REFERENCES

1. Rumelhart, D., Hinton, G. & Williams, R. Learning representations by back-propagating errors. Nature 323, 533–536 (1986). https://doi.org/10.1038/323533a0
2. S. Chalup and F. Maire, "A Study on Hill Climbing Algorithms for Neural Network Training", Proceedings of the 1999 Congress on Evolutionary Computation (CEC'99) July 6–9 1999, vol. 3, pp. 2014-2021, 1999.
3. P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. Modeling wine preferences by data mining from physicochemical properties. In Decision Support Systems, Elsevier, 47(4):547-553, 2009. https://archive.ics.uci.edu/ml/datasets/wine+quality
4. Agarap, A. F. (2018). Deep learning using rectified linear units (relu). arXiv preprint arXiv:1803.08375.
5. N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller and E. Teller, "Equation of state calculations by fast computing machines" in J. of Chem. Phys., vol. 21, no. 6, pp. 1087-1092, 1953.
6. Taha, A.A., Hanbury, A. Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. BMC Med Imaging 15, 29 (2015). https://doi.org/10.1186/s12880-015-0068-x