FIT5186 Intelligent Systems

SOUTHEAST UNIVERSITY – MONASH UNIVERSITY

JOINT GRADUATE SCHOOL (SUZHOU)

Recognition of Wine Using Neural Networks

Student name: Yingzhen Wu   ID: 2819****

Student name: Jie Liu   ID: 2819****

28 May 2017

Content

Abstract

This study analyzes wines which are grown in the same region in Italy but derived from three different cultivars. Multilayer Feedforward Neural Network (MFNN) models where backpropagation (BP) algorithm is applied is the training model we used to solve the problem. This paper introduces 14 experiments in six rounds conducted by different architectures and parameters to analyze the influence of them.

1. Introduction

Recognition of wine is one of the major technique problems in the wine production. This study analyzes wines which are grown in the same region in Italy but derived from three different cultivars. Chemical analysis on wines can be taken to measure the component indexes. We expect to realize an application of neural networks: learning to distinguish the cultivars of the wines by analyzing relative chemical attributes.

Many similar researches have been done. Penza & Gassano (2004) trained an artificial neural networks (ANNs) based on principal component analysis (PCA) as well as back-propagation method to recognize adulteration of Italian wines. Liu et.al (2008) researched on discrimination of rice wine age. In their study, a basic error back propagation (BP) algorithm was employed in the training of the ANNs. Thus, Neural networks are an ideal tool for solving this problem, since they can learn the feature of the data and the relationships behind. After getting the training data, we will use these data to train Multilayer Feedforward Neural Network (MFNN) models where BP algorithm is applied. In the training process, the desired output is fully used to backward calculate the weight changes for hidden layers. This paper introduces 14 experiments in six rounds conducted by different architectures and parameters to analyze the influence of them.

2. Data Set

The data used for this study was obtained from the machine learning databases of University of California-Irvine. This database has been created in 1998 by Blake. Its website link is http://ftp.ics.uci.edu/pub/machine-learning-databases/wine/. This data set consists of 178 instances, comprising 59 samples belong to cultivar one, 71 belong to cultivar two and 48 belong to cultivar three. Each instance is described by 13 input attributes measured by chemical analysis of wines. All the attributes are numerical in natural order without missing. Thus, the data set does not need to be preprocessed for analysis.

In our model, we decide to use attributes as inputs while use cultivars without inherent order as the outputs. Three columns are used to encode the different cultivars: cultivar one (1 0 0), cultivar two (0 1 0) and cultivar three (0 0 1) using 1-out-of-N Encoding technique. Hold-out is a method employed to divide original data set into training set and test set. When we do the experiment, 107 instances (60% of the data set) will be randomly chosen as the training set where 27 samples belong to cultivar one, 50 belong to cultivar two and 30 belong to cultivar three. The remaining 71 instances make up the test set.

	Attributes	
1)	Alcohol	numerical
2)	Malic acid	numerical
3)	Ash	numerical
4)	Alcalinity of ash	numerical
5)	Magnesium	numerical
6)	Total phenols	numerical
7)	Flavanoids	numerical
8)	Nonflavanoid phenols	numerical
9)	Proanthocyanins	numerical
10)	Color intensity	numerical
11)	Hue	numerical
12)	OD280/OD315 of diluted wines	numerical
13)	Proline	numerical
	Classes	
1)	cultivar one	(1 0 0)
2)	cultivar two	(0 1 0)
3)	cultivar three	(0 0 1)

3. Training Issues

We use NeuroShell 2 to build a MFNN architecture. Generally, three layers (input, hidden and output layer) are sufficient for the vast majority of problems. The classification accuracy of trained neural network is influenced by the selections of input neurons, initial weights, learning rate, hidden neurons, activation function and momentum tern. Thus, control variate method can be used to seek an accurate and efficient condition.

Six rounds with 14 experiments will be taken to observe the influence of each parameter. In Round 1, we check if there are key attributes or interferential attributes that can strongly or weakly affect the performance. Some of the attributes are chosen as the input neurons by their significant degrees. We desire to use the least input neurons to get the best performance. In Round 2, learning rate has been changed to see whether it affects effectiveness and convergence of learning (Hao et.al, 2003). In Round 3, we try three different initial weight values to see if the network results are improved. In Round 4, the numbers of hidden neurons are discussed to seek for less neurons with the best performance. In Round 5, two kind of activation function: Logistic-Sigmoid and Tanh-Sigmoid are applied to see the difference between them. In Round 6, the influence of momentum term is analyzed to verify whether the default value is reasonable (Attoh-Okine, 1999).

4. Results

Six rounds with 14 experiments with different architectures and parameters have been done to analyze the influence of them.

Round 1: Selection of Input Neurons

The first experiment is conducted with all the attributes and other parameters using the initial value. Where, initial weights (w) are a small value around 0.3, learning rate (c) equals to 0.1, number of hidden neurons is 21 and momentum term (α) is 0.1. After training, we can get a related contribution factors result and adjust the order, we can get Table 2.

Table 1 Classification accuracy using MFNN with 13-17-3 architecture

	classified cultivar 1	**classified cultivar 2**	**classified cultivar 3**	**row accuracy**
actually cultivar 1	32	0	0	100%
actually cultivar 2	0	21	0	100%
actually cultivar 3	0	1	17	94.44%
column accuracy	100%	95.45	100%	

Table 2 Related Contribution Attributes in order

Ranking	**Attributes**	**Input Strength**
1	Ash	0.30297
2	Alcohol	0.29324
3	Flavanoids	0.28763
4	Proline	0.28618
5	Color intensity	0.27095
6	OD280/OD315 of diluted wines	0.26669
7	Alkalinity of ash	0.26006
8	Hue	0.24650
9	Malic acid	0.21332
10	Total phenols	0.16203
11	Proanthocyanins	0.15924
12	Magnesium	0.14301
13	Nonflavanoid phenols	0.10819

Thus, some of the attributes are chosen as the input neurons by their significant degrees. For example, top six attributes for experiment 2, top eight for experiment 3 and top ten for experiment 4.

Table 3 Classification accuracy using MFNN with 6-17-3 architecture

	classified cultivar 1	**classified cultivar 2**	**classified cultivar 3**	**row accuracy**
actually cultivar 1	30	2	0	93.75%

actually cultivar 2	0	21	0	100%
actually cultivar 3	0	0	18	100%
column accuracy	100%	91.30%	100%	

Table 4 Classification accuracy using MFNN with 8-17-3 architecture

	classified cultivar 1	**classified cultivar 2**	**classified cultivar 3**	**row accuracy**
actually cultivar 1	32	0	0	100%
actually cultivar 2	0	21	0	100%
actually cultivar 3	0	1	17	94.44%
column accuracy	100%	95.45%	100%	

Table 5 Classification accuracy using MFNN with 10-17-3 architecture

	classified cultivar 1	**classified cultivar 2**	**classified cultivar 3**	**row accuracy**
actually cultivar 1	32	0	0	100%
actually cultivar 2	0	21	0	100%
actually cultivar 3	0	0	18	100%
column accuracy	100%	100%	100%	

Conclusion: From these experiments, we find that the experiment 4 has the highest accuracy, which reduces the cost of chemical component analysis while improving the performance. Thus, the architecture with ten input neurons will be selected to do further experiments.

Round 2: Influence of the learning rate

Learning rate (c) is the step size in the steepest descent algorithm. It affects effectiveness and convergence of learning. In experiments 5 and 6, we use 0.01 and 0.5 as the learning rate of the architectures.

Table 6 Classification accuracy using MFNN with c = 0.01 (10-17-3)

	classified cultivar 1	**classified cultivar 2**	**classified cultivar 3**	**row accuracy**
actually cultivar 1	32	0	0	100%
actually cultivar 2	0	21	0	100%
actually cultivar 3	0	2	18	90%
column accuracy	100%	91.30%	100%	

Table 7 Classification accuracy using MFNN with c = 0.5 (10-17-3)

	classified cultivar 1	**classified cultivar 2**	**classified cultivar 3**	**row accuracy**
actually cultivar 1	32	0	0	100%
actually cultivar 2	0	21	0	100%
actually cultivar 3	0	1	17	94.44%

column accuracy	100%	95.45%	100%	

Conclusion: The accuracy of these two experiments are lower than experiment 4 which c equal to 0.5. For large c in the experiment 5, it quick convergence but may overshoot minimum. For small c in the experiment 6, it slow convergence but true steepest descent locally.

Round 3: Influence of the initial weight

Two different initial weight values have been tried to see the network results are improved. The initial weights of experiment 7 is a small value around 0.03 and experiment 8 is 8.

Table 8 Classification accuracy using MFNN with w = 0.03 (10-17-3)

	classified cultivar 1	**classified cultivar 2**	**classified cultivar 3**	**row accuracy**
actually cultivar 1	32	0	0	100%
actually cultivar 2	0	21	0	100%
actually cultivar 3	0	0	18	100%
column accuracy	100%	100%	100%	

Table 9 Classification accuracy using MFNN with w = 3 (10-17-3)

	classified cultivar 1	**classified cultivar 2**	**classified cultivar 3**	**row accuracy**
actually cultivar 1	32	1	0	96.97%
actually cultivar 2	1	20	0	95.23%
actually cultivar 3	0	1	16	94.12%
column accuracy	96.97%	90.91%	100%	

Conclusion: Initialization of weight strongly affects the final result. Generally, the optimal initial weights are small random values. From the result, we find that the performances of small value weights are good (experiment 4 and 7) while the performance of large value is relatively bad (experiment 8).

Round 4: Selection of hidden neurons

The network does converge when the number of hidden neurons is 17 in the experiment 4. Thus, we try to remove some neurons for faster running. In experiment 9, 10 and 11, we changed the number of hidden neurons from 17 to 14.

Table 10 Classification accuracy using MFNN with 10-14-3 architecture

	classified cultivar 1	**classified cultivar 2**	**classified cultivar 3**	**row accuracy**
actually cultivar 1	32	1	0	96.97%
actually cultivar 2	1	20	0	95.23%
actually cultivar 3	0	1	16	94.12%
column accuracy	96.97%	90.91%	100%	

Conclusion: The accuracy results remain 100% until 14 hidden neurons applied, which classification accuracy is shown in the table 10. So, we think 15 is the modest number of hidden neurons.

Round 5: Influence of activation function

Two of the most common used activation functions in traditional neural networks are Logistic-Sigmoid and Tanh-Sigmoid. In the above experiments, we all use Logistic-Sigmoid as the activation function. Thus, we will try to use Tanh-Sigmoid in the experiment 12 using the architecture 10-17-3.

The classification accuracy of them are both 100%. However, the length of time on training was different. 13 seconds for Logistic-Sigmoid while 17 seconds for Tanh-Sigmoid.

Round 6: Influence of momentum term

$\alpha$is the momentum factor where $0<\alpha<1$. A default value of 0.1 is commonly used, for example experiment 4. For experiments 13 and 14, the momentum of them have changed to be 0.01 and 0.8. The accuracy results of them are the same, missing classification of one term. So, we think it may be better to use the default value.

5. Limitations

The result of classification accuracy is surprisingly high, because the data set is small in size and highly distinguished. Thus, we cannot train the network to be well performed since the initial model already has 100 percent accuracy. We instead change the architectures and parameters to see their influence of the network results. In addition, many other techniques have not been tried to classification of this problem for lack of space.

6. Conclusion

This study analyzes wines which are grown in the same region in Italy but derived from three different cultivars. Data set is used to train Multilayer Feedforward Neural Network (MFNN) models where BP algorithm is applied. This paper introduces 14 experiments in six rounds conducted by different architectures and parameters to analyze the influence of them. After all experiments, we find that using features top ten attributes in the table 2 as input will result in the highest accuracy. Hence, we conclude that these ten features are key features and influence the classification result most. The 10-15-3 architecture with initial parameters is an efficient model for training the classifier. However, due to the size of the data set is too small, the result of these experiments may be relatively inconvincible. In spite of these limitations, we still can conclude that these experiments show the feasibility to do the classification using neural networks.

References

[1] Penza, M., & Cassano, G. (2004). Recognition of adulteration of Italian wines by thin-film multisensor array and artificial neural networks. *Analytica Chimica Acta*, 509(2), 159-177.

[2] Liu, F., Cao, F., Wang, L., & He, Y. (2008). Discrimination of rice wine age using visible and near infrared spectroscopy combined with BP neural networks. *IEEE Congress on Image and Signal Processing* (CISP 08), Vol. 5, pp. 267-271.

[3] Hao, P., Yuan, J. L., & Zhong, L. (2003). Probing modification of BP neural network learning-rate. Proceedings of the *IEEE International Conference on Machine Learning and Cybernetics*, Vol. 1, pp. 307-309.

[4] Attoh-Okine, N. O. (1999). Analysis of learning rate and momentum term in backpropagation neural network algorithm trained to predict pavement performance. Elsevier Science Ltd.