Using neural network for classification of different type of forests

Chongwei Xu

Research School of Computer Science, Australian National University
U6298145@anu.edu.au

Abstract. When training a neural network, using more attributes can usually improve the training result. For the dataset of forests with different types of trees, geographical weighted variables of different type of trees can improve the total accuracy by 3%, from 81% to 84%, which is similar to the result from other research using the same dataset. In addition to adding attributes, reducing traning set can also improve the performance of a nural network both on classification accuracy and training time. By implementing genetic algorithm, we can find some attributes are less important for training, and the total accuracy can be improved to 86%.

Keywords: Neural network, image classification, class related attributes, reducing training set

1 Introduction

Neural network is a important part of artificial intelligence. One of the most important uses of neural network is to classify different type of classes according to attributes of these classes. Neural network is useful in image recognition. In this work I chose a data set about a forest with different types of trees in it. The two main type of trees used to research are ‘Sugi’ and ‘Hinoki’; other data are from mixed deciduous broadleaf natural forest and other land without forest. Since each of these three types of forests has its own economical use and environmental effects and values, mapping of their location is an important job for both economists and environmentalist (Johnson, Tateshi and Xie, 2011).

This dataset uses data from imagery from Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER), which is a satellite that can take high resolution photos in different spectra. In this dataset, the spectra values in different pixels of satellite images and geographically weighted variables are used as attributes. I chose this data set because it is not too complex for neural network training, and it can give clear results to show the usefulness of neural network, as it can classify different images of forests of Sugi, Hinoki or other types of forests.

The main idea of this work is to build a neural network to classify the data into different classes and get the classification accuracy. In addition, what I want to try is to use less attributes to train the network: by different parts of the attributes, I can have a comparison between the results of less and more attributes, in order to know how well the use of additional attributes can improve the training result.

The extra method I used is reducing the size of the training set. There is one hypothesis that a reduction in the complexity of a training set can improve learning, both in training result and training time (T.D. Gedeon and T.G. Bowden). In this work I used different size of training set and testing set to show if the reduction of training set size can improve learning in detail.

2 Method

2.1 Data

This data set can be downloaded from UCI Machine Learning Repository. In detail, this data set has four classes: 's' for 'Sugi' forest; 'h' for 'Hinoki' forest; 'd' for 'Mixed deciduous' forest; 'o' for 'Other' non-forest land. It has 27 attributes: nine of them are spectral information in the green, red, and near infrared wavelengths of satellite image for three dates, Sept. 26, 2010, March 19, 2011 and May 08, 2011; 18 of them are predicted spectral values using the inverse distance weighting interpolation method minus actual spectral values for the 's' class and ‘h’ class in the first nine attributes(Johnson, Tateshi and Xie, 2011), they will be referred to as geographical weighted variables of ‘h’ and ‘s’. This data set has 198 training instances and 325 testing instances.

The training set and testing set are already divided into two files, so they can be used in training and testing separately without preprocessing. The first column of the data is the class, and the next 27 columns are the attributes. By using the pytorch library of python, I can build a neuron network and then use training data to train and testing data to test it.

To get more detailed result, I calculated the number of data of different classes in both training and testing set. In the training set, there are 136 sets of data with class ‘s’, 38 sets of data with class ‘h’, 105 sets of data with class ‘d’ and 46 sets of data with class ‘o’. In the testing set the numbers are59, 48, 54 and 37. These numbers can be used to know the accuracy of learning for different classes.

2.2 Training

The first job I did is building a neural network to train the data set using all the attributes and see the accuracy. For the neural network, the inputs are the 27 attributes, and the outputs are the four classes. I used 50 neurons in the hidden layer, since more hidden neurons cannot give better results. The loss function I used is cross entropy, because it is easy to calculate, and works well for bigger differences between predicted and actual value. The activation function I chose is sigmoid function, because it is easy to use for model building, and this network is not very big so its drawback will not affect the result too much. I set the number of learning epoch to be 1000, that will give a high accuracy in training process and won’t take too much time. In addition to the total accuracy, I also calculated the accuracy of different classes, to see in which class the neural network can do better.

After that, I dropped some of the attributes and trained the network again to see how the accuracy changes. The 27 attributes are made up of spectra values and geographical weighted variables of class ‘h’ and class’s’ separately. By dropping variables for one class, I can compare the accuracy of that class and the other class to see how these data improved the result/

The next work is to try the method of reducing the size of training set. By dropping some of the dataset randomly in a consistent rate during the training, I can train the same neural network using less training data and compare the total and detail accuracy with the results in the first part.

The final step is implementing genetic algorithm for feature selection. In this case I want to choose the features which can give best testing accuracy. The main idea is to drop one attribute in the dataset each time, see which dropped attribute gives the biggest accuracy; then drop that attribute and use the new dataset for another round of dropping. The process will stop if in any round the best accuracy is lower than the average accuracy of datasets with all attributes, or if 9 attributes are dropped in total. In general, the program is doing crossover with one parent, and the number of ‘0’s in the mask increases each time. After all these process we can get which attributes are dropped, and if we run this for several times, we can find which attributes are usually dropped, which means they are less important.

3 Results and Discussion

3.1 Training using full or partial attributes

Table 1 shows the confusion matrix of one of the tests of the dataset using all the attributes. The total accuracy of all data is 84.62%, and in detail the accuracy of class ‘s’ is the highest of all. The total accuracy changes in different training processes, but the value is always around 85%. To compare, the total accuracy of training after 1000 epochs is 97.98%, while the accuracy of all classes are nearly 100%.

Confusion matrix using all attributes						
Predicted class	S	H	D	O	Total	Accuracy
Actual class						
S	123	11	2	0	136	90.44%
H	7	31	0	0	38	81.58%
D	10	2	85	8	105	80.95%
O	2	0	8	36	46	78.26%
Total accuracy				84.62%		

Table 1. Testing confusion matrix of data using all attributes, S stands for Sugi; H stands for Hinoki; D stands for mixed deciduous broadleaf; O stands for other.

Table 2 and Table 3 shows the confusion matrix of one of the tests using attributes without geographical weighted variables of ‘s’ and ‘h’. They show that ignoring variables of one particular class cannot really improve the performance of the network on the other class, and the accuracy of each class actually various a lot in different training processes, and the total accuracy changes too.

Confusion matrix using attributes without variables of S						
Predicted class	**S**	H	D	O	Total	Accuracy
Actual class						
S	125	9	2	0	136	91.91%
H	8	30	0	0	38	78.95%
D	12	1	83	9	105	79.05%
O	2	0	8	36	46	78.26%
Total accuracy				84.62%		

Table 2. Testing confusion matrix of data using attributes without variables of S, S stands for Sugi; H stands for Hinoki; D stands for mixed deciduous broadleaf; O stands for other.

Confusion matrix using attributes without variables of H						
Predicted class	**S**	H	D	O	Total	Accuracy
Actual class						
S	112	17	7	0	136	82.35%
H	5	33	0	0	38	86.84%
D	2	2	97	4	105	92.38%
O	2	0	13	31	46	67.39%
Total accuracy				84.00%		

Table 3. Testing confusion matrix of data using attributes without variables of H, S stands for Sugi; H stands for Hinoki; D stands for mixed deciduous broadleaf; O stands for other.

Finally, table 4 shows the confusion matrix of training using only first nine attributes. The total accuracy I chose here is 80.62%, and this value in most training processes is around 81%.

Confusion matrix using only spectra values as attributes						
Predicted class	**S**	H	D	O	Total	Accuracy
Actual class						
S	117	16	1	2	136	86.03%
H	5	33	0	0	38	86.84%
D	8	2	81	14	105	77.14%
O	3	0	12	31	46	67.39%
Total accuracy				80.62%		

Table 4. Testing confusion matrix of data using only spectra value attributes, S stands for Sugi; H stands for Hinoki; D stands for mixed deciduous broadleaf; O stands for other.

According to the four confusion matrixes and their accuracies, the first conclusion is that using geographical weighted variables can improve the total accuracy: for the examples I used, the accuracy improved from 80.62% to 84.62%. Compare to the results of Johnson, Tateshi and Xie, they used the same data set and got the total accuracy of 82.2% and 85.9% (Johnson, Tateshi and Xie, 2011). My accuracies are lower than theirs, but both results show an improvement after using more attributes. In addition, according to the total accuracy in table 2 and 3, using part of geographical weighted variables can still improve the total accuracy, although the type of tree they related to seems have no difference.

3.2 Training using reducing training set

The next part is compare the result of total accuracy using full training set and reducing training set. By shuffling the training set and using different reducing rate to reduce the training set, the training set is reduced each 100 epochs. To analyze the performance of the neural network, I run the network using 0.95, 0.9, 0.85 as reducing rate to reduce the size and unchanged training set for 10 times each. Figure 1 is the line graph of the accuracy results, the results using the same reducing rate are in the same color.

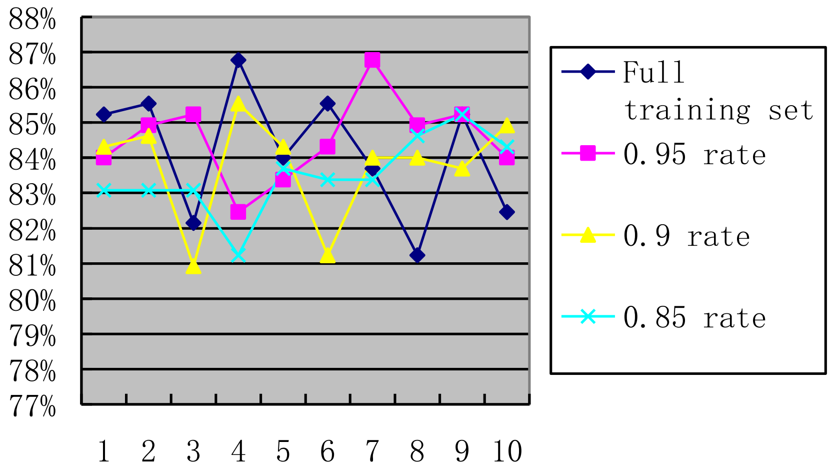

Fig. 2. Line graph of accuracy in 10 performances of the same neural network with different sized training set. X axis is the number of performances and Y axis is the total accuracy.

Average accuracy and variance		
Reducing rate	Average accuracy	Variance
1(not changed)	84.16%	0.28%
0.95	84.52%	0.13%
0.9	83.75%	0.20%
0.85	83.51%	0.11%

Table 5. Average accuracy and variance of 4 groups of accuracy results, using not changed training set, reducing rate 0.95,0.9 and 0.85.

Table 5 shows the data of these results. From figure 1 we can see the lines representing results using reducing rate seems more stable than the not changed one; and in table we can see the variance, the variance of reducing rate 0.95, 0.9 and 0.85 are all smaller than that of the full training set. In terms of average accuracy, the one of reducing rate 0.95 is slightly bigger than the initial one. This may indicates that using reducing training set have a positive effect on the training result. Aside from the result, reducing training set has the advantage of saving time. This could be very useful in bigger training sets, and lower reducing rate can be used.

3.2 Using genetic algorithm for feature selection

According to the results above, the average accuracy for full attributes is about 84%. To make sure dropping attributes improves the result I set the lower bound of accuracy to be 85%. As I described in previous section, the program for this part drop one attributes each time to make maximum accuracy for 9 times, which means it will finally drop 9 attributes or stop if the accuracy is lower than 85%. Table 6 gives the attributes dropped during 10 processes.

Results of Feature Selection										
Attribute:	1	2	3	4	5	6	7	8	9	accuracy
1	B4	B5	B8	B6-H	B7-H	B8-H	B9-H	B4-S	B6-S	85.84%
2	B1	B3	B8	B9	B7-H	B9-H	B3-S	B5-S	B7-S	86.77%
3	B2	B8	B9	B4-H	B7-H	B1-S	B2-S	B6-S	B7-S	86.15%
4	B1	B2	B4	B7	B7-H	B8-H	B9-H	B1-S	None	84.92%
5	B2	B3	B5	B9	B4-H	B6-H	B7-H	B9-H	B2-S	87.38%
6	B3	B5	B8	B4-H	B5-H	B7-H	B9-H	B4-S	B7-S	87.38%
7	B2	B8	B2-H	B5-H	B7-H	B8-H	B9-H	B1-S	B5-S	87.38%
8	B2	B3	B5	B6	B8	B2-H	B5-H	B5-S	None	84.92%
9	B1	B2	B5	B3-H	B7-H	B1-S	B3-S	B5-S	B7-S	85.84%
10	B5	B6	B1-H	B2-H	B4-H	B7-H	B9-H	B1-S	B4-S	87.07%

Table 6. Results of feature selection, each row shows attributes dropped during the process and the accuracy after all these attributes are dropped. None attribute means the process stopped earlier because of low accuracy.

From this table we can find some attributes that are dropped in most of the 10 cases. They are B5, B8, B7-H and B9-H: all of them are dropped in more than 5 cases. It indicates that they may not be important for the classification. In addition, in most cases, more –H attributes are dropped compare to –S attributes. To compare, another training on dataset without there features are performed, and the total accuracy was improved to 86%, which indicates the feature selection process is useful.

3 Conclusion and Future Work

This work shows that using additional data as attributes can improve the training result, even some of these attributes are not very related. When using neural network to map some objects, it will be useful to use theoretical predicted values as attributes. In addition, reducing the training process can improve the training result, and it can save much time when dealing with large number of datasets.

By using genetic algorithm, I dropped some attributes in the dataset and got a better result. This showed the usefulness of feature selection, and can help find useless or negative data in the dataset, which can benefit the research to the forest itself.

In the future, I want to do a research on what data can improve the accuracy of a particular class, since in the dataset I used the data related to one type of tree cannot improve the accuracy of this class considerably. I will use bigger datasets and apply the reducing training set method, to research the effectiveness of that method both on accuracy and learning time. I need more tests and different dropping sequence in order to find a better way for feather selection. In addition, finding a better function for reducing the training set is another task for me in the future.

References

1. Brian Johnson, Ryutaro Tateshi and Zhixiao Xie (2011), “Using geographically weighted variables for image classification”, Remote Sensing Letters, Vol. 3, No. 6, November 2012, 491–499
2. T.D. Gedeon and T.G. Bowden, “Heuristic Pattern Reduction”
3. Brian Johnson, “Forest type mapping Data Set”, http://archive.ics.uci.edu/ml/datasets/Forest+type+mapping