Pruning Algorithm on Neural Network Based on Distinctiveness Operator

Qifan Cao
Research School of Computer Science, Australian National University
u6013647@anu.edu.au

Abstract. Recently, neural network technology has been widely used to deal with classification and aggregation problems. On the basis of ensuring the accuracy, in order to reduce the complexity of convolutional neural network (CNN), and better determine the number of initial hidden units for constructing other CNN models in similar scale, many metrics for judging the necessary degree of hidden units have been proposed. Among them, the distinctiveness is one of the methods that can easily locate hidden units that need to be removed. In this paper, we will use the distinctiveness property to prune on trained CNN. The results show that CNN, which is pruned using the property of distinctiveness, can largely retain the accuracy of the original CNN while reducing the complexity of CNN.

Keywords: Classification, CNN, Back-propagate algorithm, Hand writing recognition, Distinctiveness

1 Introduction

A handwritten digit database named MNIST was used in this paper, which contains 60,000 examples of training set and 10,000 examples in testing set. The digits have been size-normalized and centered in a fixed-size image. Hence no more preprocessing and formatting operations are required for use. In terms of rule of thumb, this database has only ten outputs. One-layer CNN is competent for ideal results, while the simplicity of CNN reduces the work load of analyzing effectiveness of different CNN construction methods, to what extent connection pruning will affect the correction of CNN.

There are four reasons account for using MNIST as dataset. First, all the data in dataset are explainable without requiring relevant background knowledge. Unlike stock relevant data, the meaning of values in handwritten database is obvious and explainable. Second, the data type of all the data is numeric identical. No operations of unifying data formation are required. The number of features and instances are both adequate. More generalized results can be drawn from 70,000 examples. The last reason is this topic is sufficient of previous experiments. Experience can be referred to and results can be easily compared.

A fully connected CNN will be constructed at first, then importance of all the connections will be assessed by distinctiveness operator, and the ones with distinctiveness value smaller than a certain threshold will be discarded. Remaining connections will construct a new CNN classification based on the fully connected CNN. Testing accuracies of training set and testing set will be implemented on both CNN model.

This paper will describe as following outline. Section 2.1 gives a description of how the data format in database. A brief introduction of evaluation the effect of prediction will be covered in section 2.2. Section 2.3 will focus on applying distinctiveness technique to find hidden units for pruning in detail. The efficiencies and drawbacks of distinctiveness will be discussed in section 3. Conclusion and results will be covered in section 4. A comparison of testing accuracies between models in this paper and in previous papers will be shown in section 5.

2 Method

2.1 Data structure and problem description

Design of the problem model: For this Target variables in this dataset is ten different digital handwritten numbers. The first value is the "label", that is, the actual digit that the handwriting is supposed to represent, such as a "7" or a "9". It is the answer to which the neural network is aspiring to classify. The subsequent values, all comma separated, are the pixel values of the handwritten digit. The size of the pixel array is 28 by 28, so there are 784 values after the label. This dataset is used for predict the digit values from gray value of pixels.

In construction of CNN, we need to define the number of input, output and hidden unit. For simplicity, assume all connections between input layer and hidden layer, hidden layer and output layer are simple and full. We use sigmoid function as the activation function of hidden units. There is no activation function of output layer. Training is executed in epoch format instead of incremental format.

2.2 CNN evaluation

In this paper, CNN's training and testing results is described with accuracy, which is the proportion of correctly classified input examples out of the entire input examples.

$$Arc = \frac{correct(d)}{correct(d)+wrong(d)}$$

Where *d* stands for all the entire dataset, *correct(d)* is the number of correctly classified examples in *d*, while *wrong(d)* is the number of the rest of the examples in *d*. In both training and testing process, this formula can be used for evaluation of CNN.

Another evaluation operator used is cross-entropy. As the number of training set increases, the prediction of accuracy will get improved. However, the degree of improvement will decrease. As lose function, cross-entropy is able to evaluate the difference between the actual label and prediction, and the information gain from training set.

2.3 Connection pruning in CNN

After we have test the accuracies of constructed CNN on training set and testing set, we started using distinctiveness property for network pruning. In pattern space, the activation outputs of each hidden unit form a n-dimension vector. Vectors in pairs will form an angle. If the angle is over-small (e.g. < 15° here), the two hidden units function similar in pattern space. Hence, one corresponding hidden unit in pairs can be removed, and add the weight of this pair to the remaining hidden unit. If the angle is over-large (e.g. > 165° here), the two hidden units function oppositely in pattern space. Hence, both of the corresponding hidden units can be removed in pairs. In particular, if the length of a vector is small, it means that this vector has almost no effect in the forward delivery process and can be deleted. That means this hidden unit is redundant. After calculating the angle of the vectors in pairs, print out the length distribution of the vector. The result shows that all the module lengths are floating between 50 and 60. No outliers are detected. So it is not necessary to delete the corresponding hidden unit decisively.

When calculating the included angle of the vector, all the angles obtained by the combination of the vector obtained by all hidden units have a certain include angle greater than the threshold degree and need to be deleted in pairs.

Also, the values of some angles are less than the threshold, the corresponding unit weights need to be added together and delete one hidden unit. Finally, for different number of hidden unit when the training is iterated 6,000 times, the accuracy of test set before and after pruning and the number of hidden units to be clipped are shown in the following table.

Table 1. Accuracy of testing set before and after pruning

No. of epoch	Testing accuracy on testing set before pruning (%)		Testing accuracy on testing set after pruning		Accuracy increase (%)		No. of removed hidden units	
1	11.35	11.35	15.06	13.63	-32.69	-20.09	48	47.33
1	11.35		14.48		-27.58		47	
1	11.35		11.35		0		47	
2	52.06	52.89	54.17	48.68	-4.05	3.81	13	18.33
2	61.46		57.76		-3.7		10	
2	45.15		34.11		-3.68		32	
3	83.49	75.01	77.38	69.40	-6.11	-5.61	5	10
3	59.84		49.12		-10.72		20	
3	81.69		81.69		0		5	
4	83.88	83.79	82.36	83.28	-1.52	-0.51	4	5
4	82.75		82.75		0		5	
4	84.74		84.74		0		6	
5	89.33	88.88	89.33	88.73	0	-0.002	3	4
5	89.27		88.83		-0.005		5	
5	88.04		88.04		0		4	
6	90.51	90.02	90.22	89.93	-0.29	-0.10	7	3.67
6	89.32		89.32		0		2	
6	90.24		90.24		0		2	
7	92.23	91.70	92.23	91.70	0	0	0	3
7	91.79		91.79		0		7	
7	91.09		91.09		0		2	
8	92.74	92.28	92.74	92.24	0	-0.03	2	2.67
8	91.92		91.92		0		1	
8	92.17		92.07		-0.1		5	
9	92.97	93.45	93.02	93.48	0.05	0.03	5	5
9	93.78		93.78		0		4	
9	93.60		93.64		0.04		6	
10	94.04	93.6	93.83	93.49	-0.22	-0.12	3	3
10	93.18		93.18		0		3	
10	93.58		93.45		-0.14		3	

Several findings can be concluded from this table.

1) The accuracy on testing set before pruning increase with the increase of epoch iteration times. By studying the patterns of training set over times, the accuracy of prediction on training set will definitely increase. However, with the random factors of collected data, the patterns of testing set can vary from training set in certain degree, which accounts for differences of prediction accuracies on training set and testing set. In general, the improve tendencies should be close. During training process, when patterns with significance are

learned, only patterns with with less importances are learned in latter iterations of training process. Hence, the speed of accuracy increase slows down with the increase of epoch iteration times.

2) The accuracy on testing set after pruning increase with the increase of epoch iteration times. In terms of the descriptions of how to determine connections for pruning based on distinctiveness, only connections studies detailed patterns on training set are removed. After training process, CNN model may reach the over-fitting point, where further study on training set will learn too much details of training set instead of the entire dataset. This can lead to improvement on accuracy because these over-fitting factors are eliminated in CNN. On the other hand, CNN model may have not reach the over-fitting point, which means further study on training set is required for optimal accuracy and generalization function on the entire data set. In this case, any pruning operations are likely reduce the prediction accuracy on training set and testing set. This gives a good explanation that in repeated testing with the same number of epochs, prediction accuracy can get improved, impaired or remained.

3) The difference between the accuracies on testing set, before and after pruning, decrease with the increase of epoch iteration times. When training process is repeated in 3 times, the difference of prediction accuracies before and after pruning differ a lot. This can be explained by that in the initial training process, CNN model only studies limited patterns on training set, connection parameters in CNN model function similarly. After getting trained with enough times, the functions of different connections get specified. Hence, in initial training epochs, pruning operation affects more significant than in latter training epochs.

4) Less hidden units are likely to be removed with the increase of epoch iteration number. In terms of distinctiveness operator, if the angle value between two vectors is too large or too small, corresponding hidden units are to be removed. Connections in CNN are initialed with small random values, resulting vector lengths are small. In initial iterations of training process, both the directions and length of vectors may get changed dramatically. After learning major patterns of training set, these vectors only changes in much preciser range. If the number of hidden units are ideal, each hidden unit functions differently, and the chance of two hidden units function totally oppositely or over similarly can be decreased. And this can explain why less hidden units are to be removed with the process of training.

3 Discussion of pruning algorithm based on distinctiveness property

3.1 Benefits

- Easy to understand and apply.
- When available instances are abundant, pruning works better.

3.2 Drawbacks

- Accuracy decrease. When over-fitting training data, because of the training set's own characteristics, it will lead to some necessary hidden unit over-training the training set. The too small or too large angle may lead to unnecessary corresponding hidden unit deletion, causing accuracy decrease after pruning. Also, although the complexity of CNN can be reduced, the accuracy rate will always decrease more or less.
- Pruning requires some experience and skill to set the angle threshold and relevant parameters in pruning algorithms. When the range of the reserved angle is too large, the pruning effect is not obvious. When the range of the reserved angle is too small, too less vectors remains, the function of CNN can be over general to capture some necessary patterns. In the pruning process, if various parameters are not properly selected and before the optimal network structure is reached, the effect of pruning will not be achieved because of a local minimum in a medium-sized CNN.
- Depends on training set. There are three kinds of training result before pruning: local optimal result of prediction is reached, before and after the local optimal point. Pruning can eliminate the over-learned factors on training set of CNN model, which works best when CNN has reached the local optimal. However, if training set cannot capture all the major features of the entire dataset, or the distribution of data points with same features in training set and the entire dataset differs too much, the CNN model before pruning is not good enough for prediction on data with new features. To improved the prediction, CNN model should continue study the features of new data.

4 Result and conclusion

Experiments show that the pruning algorithm based on distinctiveness can solve the problem of over-fitting when training CNN model to a certain extent, and it is an effective CNN scale optimization tool. According to the actual operation experience, during the application of the pruning algorithm, attention should be paid:

The default parameter in distinctiveness operator is the range of angle. When more space of CNN required to be saved, the range of vector angle to keep can be smaller. While when more precise results are required, the range of vector angle to keep can be larger. But for improving predict accuracy, it depends. More connections only stand for higher predicting accuracy on training set but not the entire dataset. In practice, features of training set and dataset should be taken into consideration.

5 Comparison with previous result

The previous research paper whose training results are used for comparison is Gradient-Based Learning Applied to Document Recognition from Yann LeCun. The error rate of CNN in this paper and the CNN in previous paper are shown in plots below. Both error rate decrease with the increase in training set iterations times. The CNN model in previous converged before iterations reached 12. On the contrary, CNN model in this paper never converged in 30 times of iterations. In addition, error rate of CNN in this paper is always greater than that of CNN model in previous paper. The reason is that CNN model constructed in this paper is much simpler than the previous one. More model parameters are used, and more accuracy improvements are taken, better prediction accuracy CNN model will get.

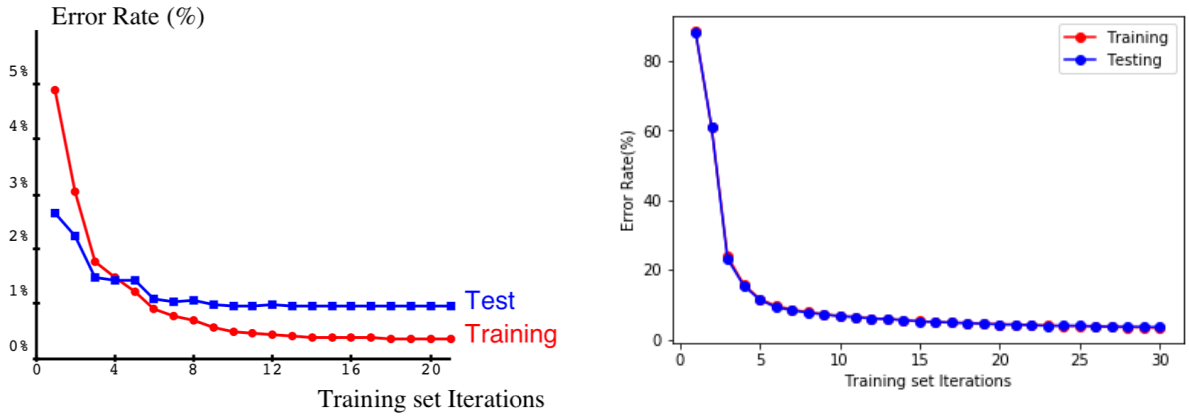

Fig. 1. The image on the left shows the error rate on testing set and training set in previous paper, the image on the right shows the error rate on testing set and training set in this paper.

Reference

1. Y. LeCun, L. Bottou, Y. Bengio and P. Haffner: Gradient-Based Learning Applied to Document Recognition, Proceedings of the IEEE, 86(11):2278-2324, November 1998

2. Gedeon, T. D., & Harris, D. (n.d.). *Network reduction techniques.* Department of Computer Science, Brunel University.