Predicting Handwritten Digits using Heuristic Pattern Reduction

Ted Pettigrove
Research School of Computer Science
The Australian National University
Canberra, ACT 0200 Australia

Abstract. Being able to predict handwritten digits allows for new capabilities for human and machine interactions. The use of neural networks assists with this solution and a heuristic pattern reduction is explored as a method to improve the generalization capabilities of the network. Heuristic pattern reduction is applied to both a classical feed-forward neural network and a deep learning convolutional neural network. The neural network was able to classify digits at a 92% accuracy and the convolutional neural network was able to classify at a 98% accuracy. It was found however that the reduction in patterns did not improve the accuracy of the network. Accuracies for both networks were maintained up until a 50% reduction before signs of degradation in quality appeared.

Keywords: Pattern Reduction, Handwritten digits, Neural Network, Backpropagation, Stratified.

Introduction

With the ever-expanding industry for machine and human interaction it will be important for machines to be able to identify hand written character. The introduction of neural networks into this problem allows for machines to have a generalized recognition technique to classify data patterns. In this paper the term "neural networks" will refer to multi layered feed-forward networks trained with back propagation. Gedeon and Bowden (1992) estimate that 70% of the applications of neural networks use some variant of this implementation. They also outline a number of techniques to improve the generalization capabilities of a neural network including node pruning and using validation sets to stop training before generalization is degraded throughout the neural network. In this paper I explored the use of heuristic pattern reduction to reduce the size of the training set to improve the generalization capabilities of the neural network in classifying handwritten digits and further their study by applying it to a deep learning method.

Deep learning is an extension on the neural network to further imitate the ability of a brain to observe, analyze and classify based on features (Rasdi Rere, L.M., Fanany, M.I., Arymurthy, A.M., 2016). The deep learning technique used in this paper is a convolutional neural network. It is a discriminative deep learning model that has significant advantages to pattern recognition due to its ability to extract features and classify specific structures in the network. This model of network will also be able to implement heuristic pattern reduction.

Heuristic Pattern Reduction

Heuristic pattern reduction is the process of removing training data in a manner that does not misrepresent the data. Gedeon, Wong and Harris (1995) found that reducing the size of the data set appropriately can improve the training time of the network without reducing the generalization capabilities. They used the classification of rock types in a mining well to show that the pattern reduction can reduce computation time while maintaining statistical accuracy. Gedeon and Bowden (1992) propose that by reducing the training set, error from outliers can be removed thus increasing the accuracy of the generalized model. Heuristic pattern reduction was used to improve the performance of a neural networks ability to predict the final grade of students based on their first 40% of course grades.

Data

The data used in this paper is the "Pen-Based Recognition of Handwritten Digits" sourced from the University of California, Irvine's machine learning repository (Alpaydin. E, Alimoglu. F,1996). 44 participants were used to collect 11,000 sample digits on a WACOM PL-100V pressure sensitive tablet. The tablet records the user's inputs at x, y coordinates every 100 milliseconds. The coordinates are then normalized to control the representation in relation to translation and scale distortions. Due to the difference in writing speeds of users, spatial resampling is used to create regular distances between points so that all data is represented the same in relation to space and not time. The resampling rate used is 16 points which gave the best trade-off between complexity and accuracy.

The data is then presented as 7494 instances for training and 3498 instances for testing. Each instance has 16 points of attributes and 1 classification attribute used for validation. From the resultant preprocessing the point attributes are integers in the range 0 to 100 and the classification attribute represents the digits 0 to 9. Table 1. shows the distribution of the data used for the training set. This data set was chosen because it both has a substantial amount of varied data and is complete with no missing feature points. This allowed for minimal data manipulation through healing or repairing which makes reuse for replicating the results.

Table 1. Number of instances for each classification in the training set (UCI).

Classification	Instances
0	780
1	779
2	780
3	719
4	780
5	720
6	720
7	778
8	719
9	719

The convolutional neural network requires a much larger dataset and a different representation of the data to get accurate classifications. For this, data was sourced from the MNIST database for handwritten digits (LeCun, Y., Cortes, C., Burges, C, 1998). This dataset has 60,000 instances for training and 10,000 instances for testing. The instances are comprised of a 28 x 28 array that represents the gray levels of the mass of pixels in the original image of the handwritten digit and a label to classify what the image is of. Table 2 shows the distribution of the data used for training. This dataset was used because it closely matched the same domain problem as the UCI dataset and is also a complete set with no missing features.

Table 2. Number of instances for each classification in the training set (MNIST).

Classification	Instances
0	5923
1	6742
2	5958
3	6131
4	5842
5	5421
6	5918
7	6265
8	5851
9	5949

Method

The network architecture used for the neural network was 16 input neurons correlating to the 16 points of attributes, 32 hidden neurons and 10 output neurons correlating with the 10 potential classifications. The neural network uses a stochastic gradient descent optimizer with a learning rate of 0.1. The number of instances in the data set was then stratified and reduced by a desired percentage. The set of percentage used in this paper include [10%, 30%, 50%, 70%, 90%]. Each test of the neural network used 1000 epochs and maintains all parameters other than size of training set.

The convolutional neural network architecture included an input layer, two convolutional layers, two pooling layers, a fully connected layer and an output layer. The input layer holds the 28x28 image. The convolutional layers compute the output of neurons connected to the regions in the output based on 10 filters for the first layer and 20 for the second. The pooling layers downsize the region to create a further subsample using 10 filters for the first layer and 20 for the second. The fully connected layer computes the scores over 50 neurons, correlating the results to the possible 10 classifications in the output layer. In a similar fashion to the neural network, the data is then stratified and reduced on the same set of percentages. Each test of the convolutional neural network uses 10 epochs however as to mitigate overfitting.

There are 3 measures looked at when evaluating both the networks: The cross-entropy loss, the training accuracy and the testing accuracy. The cross-entropy loss measures the performance of the classification model. The higher the value, the further the result is from the correct classification. I also consider the percent of correct classifications to incorrect classifications to determine the training accuracy and compare to the cross-entropy loss. Finally, once the neural network is trained, I evaluate the testing accuracy and consider the percentage of correct classifications from the testing set.

Results and Discussion

The results from the runs are shown in Figure 1, Figure 2 and Figure 3.

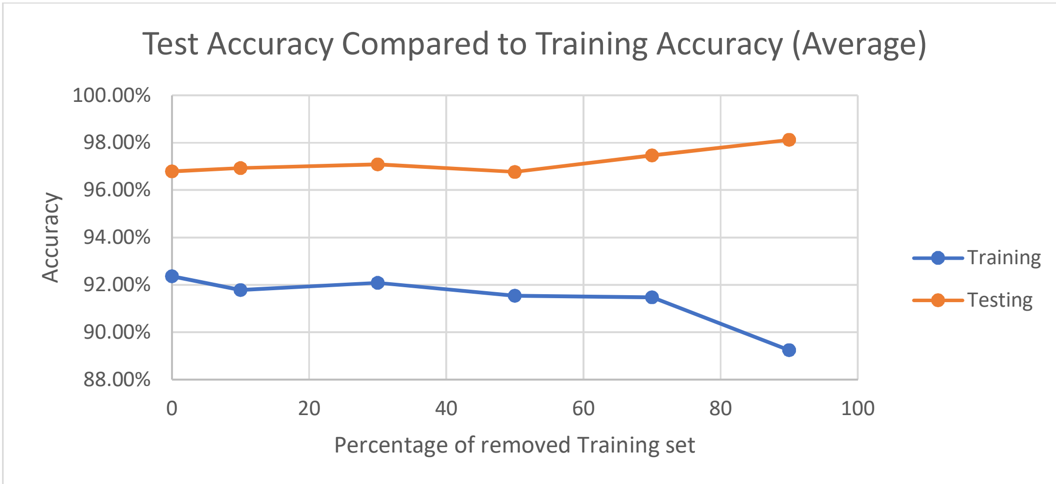

Fig 1. Shows the relationship between the training and test accuracies based on the average percentage from the runs.

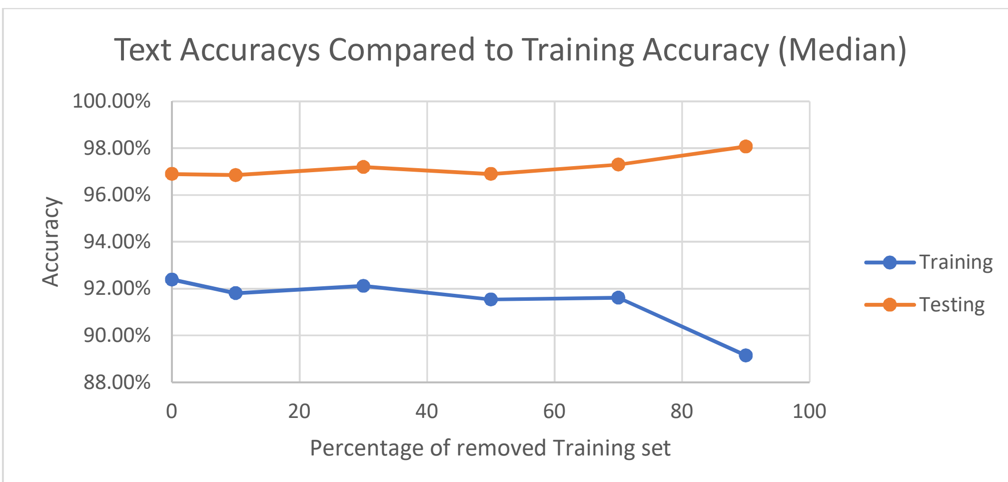

Fig 2. Shows the relationship between the training and test accuracies based on the median percentage from the runs.

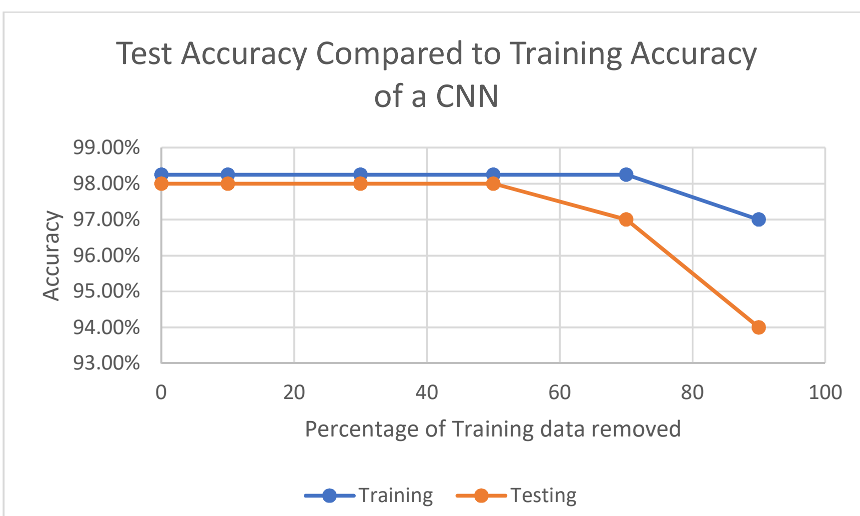

Fig 3. Shows the relationship between the training and test accuracies based on the average percentage from the runs of a Convolutional Neural Network.

From the results of Figure 1. and Figure 2. it can be identified that the testing accuracy of the neural network without any heuristic pattern reduction is approximately 92%. As pattern reduction is applied there is minimal change to the results until approximately 70%. Once the training patterns approach and exceed a 70% reduction, the neural network appears to overfit the data. This can be associated with the increase in training accuracy compared to the reduction in testing accuracy. There is no present increase to the generalization capacities of the neural network from heuristic pattern reduction.

Figure 3. depicts the convolutional neural network results with the network implementation without heuristic pattern reduction averaging 98%. The heuristic pattern reduction has no impact on the network until 50% reduction, where the testing accuracy begins to degrade. Unlike the classic neural network, the structure of the convolutional neural network shows that when the pattern reduction reaches 90% the training accuracy is also hindered.

Alimoglu (1994) performed a number of methods on the same data set none of which included heuristic pattern reduction. The neural network architecture however was also a multilayer perceptron with an input layer, hidden layer and output layer. When testing his network, he was able to achieve a 94.25% success rate for classification. He was able to further improve this accuracy to 97% using a simple voting heuristic. The discrepancy between Alimoglu (1994) and this paper's neural network implementation lies within the adaptive optimizer used to train the network. The stochastic gradient descent algorithm is comparatively inferior to the adaptive moment estimation algorithm (Adam). The function allows to learning rates to not be static and to be applied adaptively to the network for each parameter. This difference between the methods used allows for Alimoglu (1994) to achieve a higher success rate on the same classification model.

Conclusion and Future work

I have shown the effects of heuristic pattern reduction using stratification on a neural network and a convolutional neural network. The findings of this paper show that heuristic pattern reduction did not improve the accuracy of a neural network. In turn, after reaching a threshold it can hinder the overall performance of the network. The results also show that a neural network and convolutional neural network act differently when patterns are reduced below 70%. A side note identified from this and presented by Gedeon, Wong and Harris (1995) is that by reducing the training patterns, the computation time for the neural network can be decreased while maintaining its present accuracy. This is an area that could be further investigated and applied to other improving techniques.

To further improve the classification of handwritten digits, I recommend a composite approach of methods applied to a neural network. As seen from the papers cited in this paper, the best methods include a combination of techniques slightly improving different areas of the network architecture. Heuristic pattern recognition affects the inputs to a network architecture but within the network other techniques such as node pruning and cascading models could further improve the generalized capabilities of the network.

References

1. Alimoglu, F (1994) Combining Multiple Classifiers for Pen-Based Handwritten Digit Recognition, MSc Thesis, Institute of Graduate Studies in Science and Engineering, Bogazici University
2. Alpaydin. E, Alimoglu. F (1996). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml/datasets/Pen-Based+Recognition+of+Handwritten+Digits]. Department of Computer Engineering, Bogazici University, 80815 Istanbul Turkey
3. Gedeon, T. D., & Bowden, T. G. (1992). Heuristic pattern reduction. In International Joint Conference on Neural Networks (Vol. 2, pp. 449-453).
4. Gedeon, T. D., Wong, P. M., & Harris, D. (1995, June). Balancing bias and variance: Network topology and pattern set reduction techniques. In International Workshop on Artificial Neural Networks (pp. 551-558). Springer, Berlin, Heidelberg.
5. LeCun, Y., Cortes, C., Burges, C. (1998). THE MNIST DATABASE of handwritten digits. Courant Institute, NYU
6. Rasdi Rere, L.M., Fanany, M.I., Arymurthy, A.M. (2016). Metaheuristic Algorithms for Convolution Neural Network. Machine Learning and Computer Vision Laboratory, Faculty of Computer Science, Universitas Indonesia, Depok 16424, Indonesia