Pattern Reduction on Handwritten Digits

Reilly Francis

Received: 30/05/2018 / Accepted: 30/05/2018

Abstract Handwritten digit recognition is an already established problem that impacts many facets of computer vision and machine learning; with lots of research having already been done in this field. It is a non-trivial task to build a model that can recognise and classify digits as peoples handwriting styles differ greatly from one person to the next. Therefore the model needs to be robust and generalized to remain accurate across the large variations in test cases. This research is the implementation of a generic convolutional neural network with the goal of maximising classification accuracy of the digits 0-9 and then comparing that to two layer feed-forward Neural Network and the impact heurstic pattern reduction has on both networks. I have found that patten reducing the training data for the convolutional neural network maintained accuracy with an 80% reduction without significant drop in testing accuracy. Pattern reducing the feed-forward network maintained accuracy with a 95% reduction without significant drop in testing accuracy. The CNN model's accuracy is similar to other models done on the same dataset and the NN was below similar models; this wasn't the goal of the research and is secondary to the result of pattern reduction.

Keywords MNIST · CNN · Heuristic pattern reduction · Handwritten Digits · Feedforward Neural Network

1 Introduction

Handwritten digit recognition encompasses both recognizing and classifying handwritten digits from zero to nine without human intervention. This problem set includes both on-line and off-line (Fakhr, n.d.) classification. Classifying digits off-line is harder than on-line classification as handwritten style differers greatly from person to person. People also have slight transformations in their digits when drawing the same digit multiple times. The research is focused on both off-line and on-line classification. The end result of both models would be that given a picture of a digit it can classify what digit was drawn.

Both the Convolutional Neural Network(CNN) and feed-forward Neural Network(NN) is trying to achieve the highest classification accuracy on the testing dataset with restrictions in placed on the network. The restrictions include the number of epoch the network is allowed to train in as well as the activation function and optimizer the network can use. This is because the accuracy of the network is secondary to the effect pattern reduction on the training data has on accuracy and speed of the network. Ideally heuristic pattern reduction would improve the model as it improves general classification as shown by previous work done by (Gedeon & Bowden, 1992).

1.1 Handwritten Digits Data Set

The dataset used to train and test the model is the Pen-Based Recognition of Handwritten Digits Data Set(Handwritten Digits). ("Pen-Based Recognition of Handwritten Digits Data Set", n.d.) The dataset is composed of 250 samples from 44 writers. Each writer was asked to write 250 digits in a random order in boxes of 500x500 table pixel resolution. These digits were written with a Wacom table with an integrated LCD display so the writers wrote directly inside the input box with a stylus. The dataset is split into a training set composed of samples written by 30 writers which is used for training, cross-validation and writer dependent testing. The rest of the samples make up the testing set to avoid the model over-fitting the training set. Typically for datasets, especially for handwritten data sets, the data would be preprocessed to avoid the model learning errors due inherent from the sampling method. For the Handwritten dataset it has already been preprocessed. The data has been normalized to a range of 100 for x and y as x typically stayed in that range for the data. The digits have also been re-sampled as a sequence of points regularly spaced in arc length, as opposed to the input sequence, which is regularly spaced in time. ("Pen-Based Recognition of Handwritten Digits Data Set", n.d.). The Handwritten Digits dataset was chosen due to a number of factors. A major factor was that the dataset had no missing data which meant data didn't have to be scrapped in order to avoid training biases into the model from the sampling and not the underlying trends in the data. Another factor was both the training and testing dataset are already stratified which allows testing without sampling bias. This bias can appear if the training dataset has a statistically significant lower number of one category when it's probability is equal to other classes. Lastly handwriting recognition is a very broad field with lots of applications across different industries.

R. Francis
College Of Enginering and Computer Science
Australian National University
Tel.: +61 2 6125 8630
Fax: +61 2 6125 0010
E-mail: u5826312@anu.edu.au

Investigating if reducing the input size of the training set improves the accuracy of the model is also desirable as typically computer vision and language recognition have large datasets and reducing the size needed could have large impacts in the field.This becomes more prevalent as many deep neural techniques usually require lots of data, like a CNN.

Table 1: Number of classifications in NN training set

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

All classification values in the feed-forward two layer neural network training set were summed up

1.2 MNIST Data Set

The dataset used to train and test the model is the the MNIST database of handwritten digits(MNIST) ("THE MNIST DATABASE of handwritten digits", n.d.). The dataset is composed of samples from NIST's Special Database 3 and Special Database 1. SD-1 contains 58,527 images written by 500 different writers. SD-1 is split in two with the first 250 writers making up the training set and the rest making up the testing set. Samples were then added from SD-3 to make a full set of 60,000 training patterns. The dataset is split into a training set that is the full 60,000 training patterns and a testing set that is made up of 10,000 testing patterns. The MNIST dataset has already been preprocessed. The data has been normalized to a 28x28 image from the original 20x20 sizing. The digits have also been greyscaled as a result of the anti-aliasing technique used by the normalization algorithm they used ("THE MNIST DATABASE of handwritten digits", n.d.). The MNIST dataset was chosen due to a number of factors. A major factor was that the dataset is the off-line version of the same classifications. This allows a strong comparison between the two neural networks.

Table 2: Number of classifications in CNN training set

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

All classification values in Convolutional Neural Network training set were summed up

1.3 Heuristic Pattern Reduction

Heuristic pattern reduction is reducing the complexity of a training set to improve generalisation and potentially accuracy. This builds on the work done by (Kruschke, 1989) and (Gedeon & Bowden, 1992). Reducing the network size to some minimal size has been shown to improve the generalisation capabilities of a neural network. This is in part due to eliminating outliers in the data that are irrelevant when learning the patterns inherent in the data. The primary benefit of this technique is it can improve accuracy through data preprocessing. A secondary benefit of this is it speeds up the learning of a network due to running on a smaller dataset.

2 Method

Two two layer feed-forward neural networks with back-propagation was used to train on the Handwritten Digits training set. One with heuristic pattern reduction and the other without. Both NN networks only have 32 hidden layers as there are a limited number of features. The identical neural networks allow for the improvements or lack there of that heuristic pattern reduction had on training the model. Training time is limited to 1000 epochs as this matches the source paper (Gedeon & Bowden, 1992) implementation. This allows a more accurate comparison between experiments. Sigmoid was used as the activation function for the network as it is typically used for classification problems as well, with ReLU being used for more deep neural network applications. The optimization function chosen for the network was SGD. SGD was chosen as it closely matches with what heuristic pattern reduction aims to do, remove redundant data from the pattern or local local minima when contrasted with SGD.

A confusion matrix and number of correctly predicted data compared to targets was used to score both networks. This method was sufficient to test the accuracy of the classification of the 10 different digits as the targets were given along with the test data; which allowed the confusion matrix to check if the result is accurate or not. In order to see if heuristic pattern reduction was improving the accuracy of the network; accuracy of different reduced subsets was measured 10 times and averaged to get the general loss, training accuracy and testing accuracy.

The CNN was composed of 10 filters in a 24x24 convolutional layer with convolutions of a 5x5 kernal. This then feeds into a 10 12x12 pooling layer with a max pooling of 2x2. This connects to another 20 filter 8x8 convolution layer connected to a 20 4x4 pooling layer. After the second pooling layer a dropout of 25% occurs which prevents over fitting. This is then connected to a 50 neuron fully connected layer. Finally this is then passed to the output layer which has 10 neurons, one for each classification and the final results are log Softmaxed to get the probability distribution of the output. This is done as the probability distrubtion closely matches how the network weighs the image into it's classification. The particular hyperparamters were chosen as they closely match existing work done by ("Getting started with PyTorch for Deep Learning (Part 3: Neural Network basics)", n.d.) and hyperparamter tuning were out of scope for this paper.

3 Results and Discussion

Table 3: Accuracy of Heurstic Pattern Reduction on the NN training dataset

Test Set Reduction	Loss	Training Accuracy	Testing Accuracy
1	0.9529	85.99	80.39
0.9	0.9772	86.51	80.75
0.8	0.9679	86.83	81.73
0.7	0.9649	86.06	80.45
0.6	0.9578	86.18	81.37
0.5	0.9661	86.33	81.61
0.4	0.9610	86.76	81.16
0.3	0.9396	87.38	80.96
0.2	0.9559	86.66	80.99
0.1	0.9855	86.40	79.8
0.09	0.9306	88.19	80.07
0.08	0.9426	86.28	78.64
0.07	0.8995	88.68	80.81
0.06	0.9278	88.93	80.19
0.05	0.9297	88.93	79.35
0.04	0.9791	88.46	79.02
0.03	0.9441	89.64	77.52
0.02	0.8906	92.21	76.59
0.01	0.8854	92.84	73.99

Tests were run 10 times and the average value was recorded to avoid outliers in the stratified data. Epoch was set to 1000 with a learning rate of 0.01

Table 4: Accuracy of Overfitting of the Heurstic Pattern Reduction on the NN training dataset

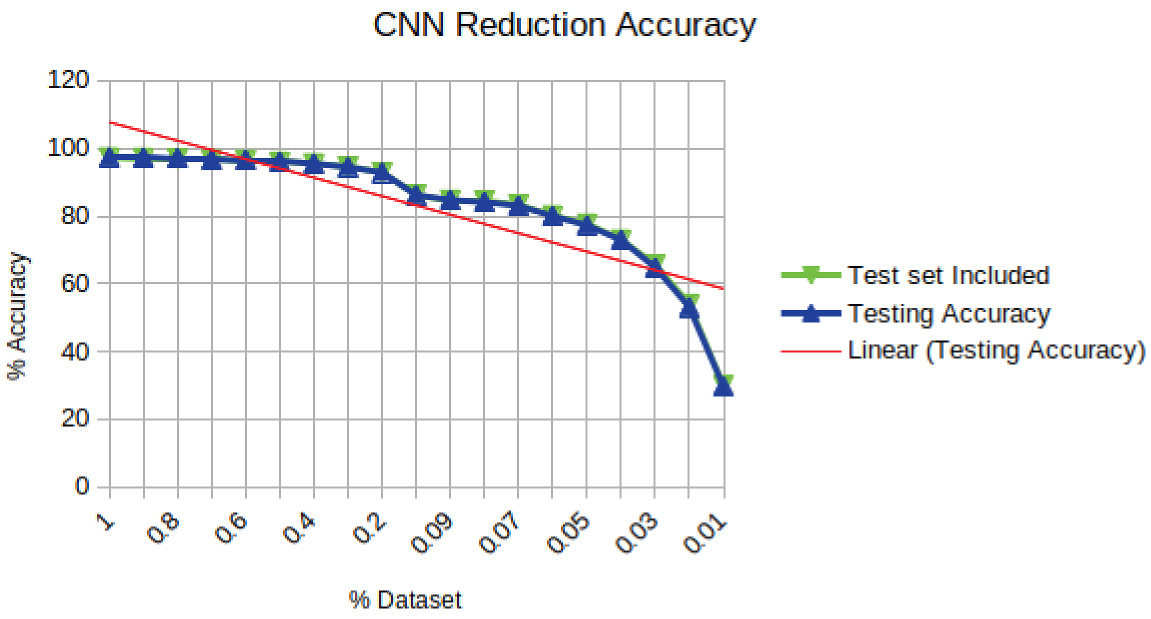
Test Set Reduction	Loss	Training Accuracy	Testing Accuracy
0.1	0.2213	96.13	87.61
0.09	0.2252	96.66	87.81
0.08	0.2497	96.16	87.20
0.07	0.2335	96.01	87.36
0.06	0.2348	96.06	86.09
0.05	0.2297	96.36	85.48
0.04	0.2231	96.09	84.76
0.03	0.2373	96.43	83.20
0.02	0.1711	98.52	82.62
0.01	0.1683	98.38	78.45

Tests were run 10 times and the average value was recorded to avoid outliers in the stratified data. Epoch was set to 5000 to find the point in which over-training occurred. Learning rate of 0.01

Table 5: Accuracy of CNN reduced training dataset

Test Set Included	Loss	Training Accuracy	Testing Accuracy
1	0.0864	97.25	97.33
0.9	0.0944	97.13	97.24
0.8	0.0978	96.91	97.05
0.7	0.1089	96.64	96.67
0.6	0.1134	96.49	96.54
0.5	0.1318	96.00	96.01
0.4	0.1491	95.48	95.47
0.3	0.1852	94.56	94.45
0.2	0.2627	92.92	92.85
0.1	0.5140	86.22	86.13
0.09	0.5523	84.70	84.73
0.08	0.6072	84.38	84.08
0.07	0.6602	83.29	83.01
0.06	0.7551	80.11	79.97
0.05	0.8547	77.63	77.19
0.04	1.0114	72.97	72.87
0.03	1.2340	65.49	64.70
0.02	1.6150	53.91	52.99
0.01	2.1600	29.85	29.71

Tests were run 10 times and the average value was recorded to avoid outliers in the stratified data. Epoch was set to 10 with a learning rate of 0.01



The results are rather interesting. In Table 3 we can see a tread that reduction improves the testing accuracy until we get to a reduction of approximately 95%. This trend then stops as we go below a reduction of 94% due to over-fitting. This trend isn't as strong as in (Gedeon & Bowden, 1992) work. This could be that there isn't any outliers present in the training dataset that when reduced out improve generalisation of the model. This trend isn't continous and is quite sporatic for if the testing accuracy will improve or not with the reduction. I believe this is due to the random nature in which the dataset is reduced. In Gedeon's paper they remove outliers which is why their trend of accuracy improvement is more proment. In order to test the effects of over-fitting of the NN model and to get an accuracy that more closely mimics related work on the dataset, the same tests were applied with an epoch of 5000. What is interesting is that on this particular dataset, table 4, the training data can be reduced up to 98% with a minimal impact on overall accuracy if the epoch number is increased.

For the CNN network the there is no notable improvement in the accuracy of the model as the training dataset is reduced. This strengthens (Gedeon & Bowden, 1992) notion that ”This improvement is most likely due to the simplification of the error surface in pattern space traversed by the network as it attempts to locate the minimum. That the minima found after simplification can be better than those found with the original pattern set indicates that none of the significant features of the original pattern set have been lost”. For the MNIST dataset applied to a CNN there appears to be no outliers whose remove allows for easier traversal to the minimum through the backpropogation as seen by the CNN Reduction Accuracy graph. I believe this is due to the preprosessing and curation of the training set already done by the authors of the (“THE MNIST DATABASE of handwritten digits”, n.d.) dataset. This technique still had merit to a deep neural network as with a reducton of 50% the average accuracy of the testing dataset fell 1.22%. This means that computation times of large datasets can be greatly improved with minimum impact on the accuracy of the network. This reduction in accuracy could be prevented with an increase in epochs as demonstrated in the NN overfitting test.

While heuristic pattern reduction doesn’t improve accuracy as much as originally thought it does greatly reduce the time each epoch takes. This can then be used on particularly big datasets to run larger epochs without spending extra time on them. The accuracy of a simple two-layer neural network comes close to similar work done by (LIANG, 2013). Training our model using SGD as the optimizer and sigmoid as the activation function reached a testing accuracy of 90.17% given 40,000 epochs. A higher accuracy could be achieved using this method given more training data as the training accuracy reached 100%. For the accuracy of the CNN with a 50% reduction, 98.42% with 20 epochs, is similar to the 99.01% accuracy achieved by (Lecun, Bottou, Bengio, & Haffner, 1998) on the same dataset.

4 Conclusions and Futurework

I have shown that while heuristic pattern reduction has been shown to greatly improve generalisation accuracy of a model it depends on the dataset that the model is being trained on. For the Handwritten Digits dataset heuristic pattern reduction on it’s own didn’t significantly improve the testing accuracy of the neural network. What heuristic pattern reduction has done is maintain testing accuracy with no significat loss up to a reduction of 94% with slight accuracy increases depending on which data samples were randomly removed. With this reduction the speed at which we can train the neural network as well as how many epochs we can train greatly improves. This trend is not present with the CNN on the NMIST dataset. This is possibly due to the data having no stastical outliers that when removed speed up the gradient decent of the optimizer. Both networks demonistrated that the training dataset can be reduced by a large margin with only a minimal drop in testing accruacy, which could greatly speedup the learning rate of large neural networks. Future direction of this research would be a deeper investication into how heurstic pattern reduction behaves on unstructured data where finding stastical outliers is non-trival with large datasets. Another path this research could take would be how heurstic pattern reduction behaves when used on modern optimizers such as Adam or RMSprop. This interaction is unknown as heurstic pattern reduction relies on removing data that slows down minimum convergence.

References

Fakhr, M. (n.d.). On-line handwriting recognition.

Gedeon, T. & Bowden, T. (1992). Heuristic pattern reduction. In *International joint conference on neural networks* (Vol. 2, pp. 449–453).

Kruschke, J. (1989). Improving generalization in back-propagation networks with distributed bottlenecks. *Int. Joint Conf. on Neural Networks*, *1*, 443–447.

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998, November). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324. doi:10.1109/5.726791

LIANG, T. W. (2013). *Computational intelligence methods for medical image understanding, visualization, and interaction* (Doctoral dissertation).

CNN Hyperparameters. (n.d.). Getting started with pytorch for deep learning (part 3: neural network basics). (n.d.). Retrieved May 30, 2018, from https://codetolight.wordpress.com/2017/11/29/getting-started-with-pytorch-for-deep-learning-part-3-neural-network-basics/

Hadnwritten. (n.d.). Pen-based recognition of handwritten digits data set. (n.d.). Retrieved April 27, 2018, from http://archive.ics.uci.edu/ml/datasets/Pen-Based+Recognition+of+Handwritten+Digits

Handwritten. (n.d.). The mnist database of handwritten digits. (n.d.). Retrieved May 30, 2018, from http://yann.lecun.com/exdb/mnist/