Feature Selection in Convolutional Neural Network with MNIST Handwritten Digits

Zhuochen Wu

College of Engineering and Computer Science, Australian National University
U5842051@anu.edu.au

Abstract. Feature selection is an important technique to improve neural network performances due to the redundant attributes and the massive amount in original data sets. In this paper, a CNN with two convolutional layers followed by a dropout, then two fully connected layers, is equipped with a feature selection algorithm. Accuracy rate of the networks with different attribute input weight as zero are calculated and ranked so that the machine can decide which attribute is the least important for each run of the algorithm. The algorithm repeats itself to remove multiple attributes. When the network will not achieve a satisfying accuracy rate as defined in the algorithm, the process terminates and no more attributes to be removed. A CNN is chosen the image recognition task and one dropout is applied to reduce the overfitting of training data. This implementation of deep learning method proves its ability to rise accuracy and neural network performance with up to 80% less attributes fed in. This paper also compares the technique with other the result of LeNet-5 to see the differences and common facts.

Keywords: CNN, Feature selection, Classification, Real world problem, Deep learning

1. Introduction

Feature selection has been a focus in many study domains like econometrics, statistics and pattern recognition. It is a process to select a subset of attributes in given data and improve the algorithm performance in efficient and accuracy, etc. It is commonly understood that the more features being fed into a neural network, the more information machine could learn from in order to achieve a better outcome. However, among all given data, especially from the real world, many of them could be noisy, redundant or invalid, which cannot accumulate the correct rate for testing but being an interference for machine to recognise the real pattern from valid but small amount data[6].
Furthermore, those useless data increase the time complexity of most algorithms and take up more storage while computing[1]. Hence it is necessary and important to get rid of excessive features during classification. The feature selection process can be considered as choosing the most useful N attributes from the original M attributes, N<M, so that the classification result accuracy can be improved through reducing feature attributes.

A conventional neural network (CNN) with two convolution layers is chosen for the proposed task. This CNN has two conventional layers followed by a dropout, then two fully connected layers. The outputs will be digits from 0 to 9 to indicate the classification results. The CNN is defined to use Log SoftMax and then use the Negative Log Likelihood as loss function. SGD rule is used to update network weights.

It is proposed that using the neural network alone first to carry out the classification task for 10 epochs to complete training and record the results. Then implement the neural network with feature selection prior to learning process and proceed another training, record the results again and compare with the first group to see the performance difference.

To investigating a feature selection problem with handwriting classification, I believe the MNIST data set is appropriate in terms of size, difficulty and application in the real world.

2. Method

2.1 Data Set Selection

The MNIST data set is chosen to practice a simple neural network for multiple reasons. First of all, it has a training set of 60000 examples and a test set of 10000 examples. The data set has been normalized to 20x20 pixels and centered into 28x28 image, which is easy to handle when trying out a new technique using a CNN without spending too much time for data preprocessing and formatting. Secondly, this particular data set is widely used for image pattern recognition. It contains binary images and labels of them, which is suitable for supervised machine learning. Quite amount of paper processing the data set by different machine learning methods can be found and compared to get more insights of my own research. Thirdly, it can be a useful and meaningful implementation in the real world to recognize and convert hand written on tablets to digital texts.

The raw data are hand written pictures of digits form 0 to 9 from approximately 250 writers. They are from Special Database 3 and Special Database 1, and the former data are cleaner and easier to recognize. The MNIST training set is consist of 30000 SD-3 patterns and 30000 SD-1 patterns. Similarly, the test set is divided in half data from these 2 databases. It is also worth mentioning that the sets of writers of training and test sets are selected to be disjoint to make sure the test result is at its most value.

The goal is to train the neural network with the dataset so that the machine can recognize handwritten digits from test set correctly.

2.2 Neural Network Model Design

A CNN with 2 convolutional layers and a dropout is chosen for the proposed task. This network has kernels of size 5 for each filter and take input of 2 channels since the MNIST dataset are black and white images. Two pooling layers after convolutional layers are both set with max-pooling function to extract features. ReLU function is used as the activation function for its simplicity to implement and advantage of faster convergence. At the end of the network, we would like to use SoftMax function to output the possibilities of 10 different labels as a result. However, we are using Negative Log Likelihood as our loss function so we just need to adjust SoftMax to Log SoftMax so that they can be linked up with compatible date type.

To apply feature selection on this network, we also need a way to decide which attributes to be excluded from all features. Here, a simple and straight forward algorithm is used. Given the trained network, accuracy rates are computed when one attribute are excluded[4]. To exclude one attribute, we simply set the input weight of it to zero. Then the accuracy rates of those networks are ranked. When the network can achieve an accuracy not more than R% of decreasing with one more attribute removed, it will remove the attribute and computing again. Else the algorithm will terminate.

Feature selection algorithm

1. Let A = {A1, A2, …An} be the set of input attributes to the CNN. Let R be the acceptable maximum drop of accuracy rate of test set.
2. Train network N to minimise the loss value with A as input so that the accuracy rate of training set is acceptable.
3. For all k = 1, 2, …n, network Nk has the weight from input Ak as zero and weights from other inputs equal to weights of network N.
4. Compute the accuracy rates of training set (Rk) and test set (R'k) respectively.
5. Rank networks Nk by their accuracy rates of training set.
6. Compute the change of accuracy rate of test set, r, for each Nk from k = 1. If r <= R, remove Ak from input set A, and N = N-1. If k < N, k = k+1 and go to repeat. Else stop the algorithm.

3. Result and Discussion

3.1 Result Comparison

Results generated from the original CNN is shown as tuples of average loss and accuracy rate of test set.

test1
(0.2078, 94.08%), (0.1257, 96.20%), (0.0971, 97.07%), (0.0829, 97.49%), (0.0771, 97.67%), (0.0659, 98.00%), (0.0599, 98.09%), (0.0594, 98.21%), (0.0543, 98.29%), (0.0496, 98.40%)
Learning rate = 0.01, channel = 2

test2
(0.1105, 96.70%), (0.0784, 97.51%), (0.0639, 97.93%), (0.0560, 98.29%), (0.0544, 98.48%), (0.0512, 98.53%), (0.0472, 98.56%), (0.0441, 98.70%), (0.0433, 98.66%), (0.0367, 98.78%)
Learning rate = 0.02, channel = 2

test3
(0.1129, 96.54%), (0.0740, 97.79%), (0.0639, 98.03%), (0.0511, 98.50%), (0.0459, 98.65%), (0.0481, 98.48%), (0.0369, 98.90%), (0.0399, 98.80%), (0.0359, 98.97%), (0.0327, 99.03%)
Learning rate = 0.02, channel = 3

It is proved that a slightly higher learning rate can improve the performance slightly but it remains curious that the increase of input channel provided a better result. Given input channel as 3, the highest accuracy rate increased to 99%, which is a good outcome. Considering the data is actually black white with grey scale, I can only guess the reason is that adding one more channel to discriminate features has taken the level of greyness into consideration.

Results from LeNet-5 constructed by LeCun are shown as follow. [11]

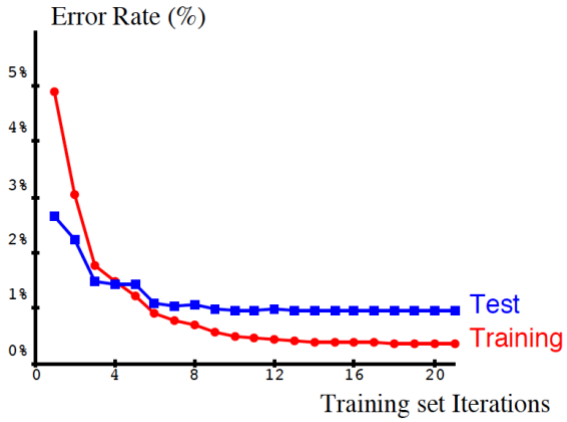

Figure 1. Test and training set error rate of LeNet-5. Convergence is attained after 10 to 12 passes through the training set.

LeNet-5 is a CNN of 7 layers with input of 32x32 images. The larger input is used to capture more detailed features like ending strokes. Convolutional layer1 has 6 feature maps of 28x28 size, sub-sampling layer 2 has 6 feature maps of 14x14 size and convolutional layer 3 has 16 feature maps. Then sub-sampling layer 4 also has 16 feature maps of 5x5 and convolutional layer 5 has 120 feature maps. The loss function used is Maximum A Posteriori criterion. It can not only push down the penalty of correct class but also pull up the penalties of incorrect classes. Then the gradient is computed by back-propagation.

For Regular Database in LeCun's paper, which is the same size of 28x28 image, learning rates are tuned down to 0.0005, 0.0002, 0.0001, 0.00005 and 0.00001. The training of the LeNet-5 was carried out for 20 iterations.

My CNN get a result from 96.54% to 99.03% through 10 iterations and LeNet-5 has a correct rate from 98.25% to 99.00%.

Comparing with LeCun's results of LeNet-5, it is shown that the more complex CNN structure has an advantage of starting from a higher standard and reaching the maximum correction rate earlier, at iteration 6. This indicates that LeNet-5 is more stable and more capable of distinguishing patterns from the beginning. However after reaching the 99%, it is hard to have further improvement due to some limitations, which results in the same highest result as my network. But overall, it is relatively easy for MNIST to achieve a accuracy rate above 98%, so it is the network's learning speed, ability and stability that need to be improved in future work.

3.2 Related Work

Feature selection problem is getting more and more attention in machine learning field and many techniques are developed. There are two types of methods to search for an optimal subset of features, exhaustive and heuristic[2]. An example of exhaustive algorithms is FOCUS algorithm. It starts as an empty set and computes exhaustively until it can find a minimal set of features representing the pattern. The algorithm can also be heuristic, like Relief algorithm. It assigns a relevance weight to each attribute and update the weight. It does not has the ability to remove redundant features[8] and always select most of original features. A probabilistic approach[9] is also possible to filter out the optimal features but can be time consuming regarding to large data sets.

Feature selection techniques are also classified into 3 types-embedded selection techniques, filter techniques and wrapper techniques[10]. Embedded techniques like L1 regularization and decision tree are used to optimize the performance of an algorithm or mode. Filter techniques select the features and then pass them on to an induction algorithm. Such methods including information gain, chi-square test and variance threshold, etc. In wrapper methods, feature selection algorithm works like a wrapper before an induction algorithm. They are more computationally expensive than filter methods. This includes genetic algorithms, recursive feature elimination and sequential feature selection, etc.

4. Conclusion and Future Work

4.1 Conclusion

During this experiment, we combined a simple convolution neural network with a basic feature selection algorithm to achieve better performance.

Through the work described above, it is apparent that feature selection can be a huge favor for neural network classification. In addition to risen the accuracy rate generally, it can usually reduce the feature attributes up to 80%, which increases computation efficiency and suppress the complexity.

In this particular case, it is also hard to get higher accuracy result after reaching 99% for MNIST dataset. Compared with LeNet-5, the latest CNN of LeNet structure, it is also evident that the performance of the network is superior from the beginning, which indicates the feature extraction method of my network still needs to be imporved.

4.2 Further Improvement

With knowing a clear advantage of feature selection in a real world problem, it can be explored that combining different feature selection techniques like SBC with more complex convolutional neural network to achieve a higher prediction accuracy. Construction of feature maps needs to be improved further to look into deeper level features. In this way, the network can collect more detailed features to increase the discrimination ability. Penalty functions may be introduced in this neural network so we can have a better control of the timing to remove an attribute.

Apart from accuracy rate, algorithm complexity of time and space can also be compared. This may give us a more comprehensive understanding of algorithms performance and their advantages and defects, which help us apply them in difference cases appropriately.

References

[1]D. Aha, "Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms", *International Journal of Man-Machine Studies*, vol. 36, no. 2, pp. 267-287, 1992.

[2]H. Liu and R. Sationo, "Incremental Feature Selection", *Applied Intelligence*, vol. 9, pp. 217-230, 1998.

[3]C. Ratanamahatana and D. Gunopulos, "Scaling up the Naive Bayesian Classifier: Using Decision Trees for Feature Selection."

[4]R. Setiono and H. Liu, "Neural-network feature selector", *IEEE Transactions on Neural Networks*, vol. 8, no. 3, pp. 654-662, 1997.

[5]L. Milne, T. Gedeon and A. Skidmore, "CLASSIFYING DRY SCLEROPHYLL FOREST FROM AUGMENTED SATELLITE DATA: COMPARING NEURAL NETWORK, DECISION TREE & MAXIMUM LIKELIHOOD", 2018.

[6]D. Koller and M. Sahami, "Toward optimal feature selection", *Machine Learning: Proceedings of the Thirteenth International Conference*, pp. 284-292, 1996.

[7]J. Schlimmer, "Concept Acquisition through Representational Adjustment", 1987.

[8]K. Kira and L. Rendell, "The feature selection problem: Traditional methods and a new algorithm", *AAAI Press/The MIT Press*, pp. 129-134, 1992.

[9]H. Liu and L. Setiono, "A probabilistic approach to feature selection—a filter solution", *Morgan Kaufmann Publishers*, pp. 319-327, 1996.

[10]A. Blum and P. Langley, "Selection of relevant features and examples in machine learning", *Artificial Intelligence*, vol. 97, no. 1-2, pp. 245-271, 1997.

[11]Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998.