A Comparative Study on Handwriting Digit Recognition Classifier Using Neural Network, Support Vector Machine and K-Nearest Neighbor

Chayaporn Kaensar

Department of Mathematic, Statistic and Computer Science, Faculty of Science,
Ubon Ratchathani University, Thailand
scchayka@ubu.ac.th

Abstract. The aim of this paper is to analyze efficiency of three classifiers which will be experimented and compared to find out the best techniques. They were experimented on a standard database of handwritten digit. However, not only recognition rate is considered, but also other issues (ex. error rate, misclassified image rate and computing time) will be analyzed. The presented results show that SVM is the best classifier to recognize handwritten digits. That is, the highest recognition rates (96.93%) are obtained. But the computing time of training is the main problem for them. Conversely, other methods, like neural networks, give insignificantly worse results, but their training is much quicker. However, all of the techniques also represent an error rate of 1–4% because of confusion with digits 1 and 7 or 3, 5 and 8 respectively.

Keywords: Handwritten Digit Recognition, Artificial Neural Network, Support Vector Machines, K-Nearest Neighbor and Recognition Rate.

1 Introduction

Handwritten digit recognition has attracted a great deal of research and analysis because there are many places where handwritten digit documents still exist, for example, automatic letter sorting at the post office, the cheque processing in the bank, or historical documents. [1][2]. Unfortunately, such writing documents are very hard to recognize even by people. Thus, a system which could aid an automatic recognition of handwritten documents would be very desirable.

Handwritten digit recognition is an important filed of Optical Character Recognition (OCR) and can be seen as a sub problem of OCR. It is the ability of a computer to receive and interpret intelligible handwritten digit input from sources such as documents, image and other devices.

Although more number of proposed system and classification techniques have been developed for this area such as [3][4][5], proper accuracy of predicting the pattern is still questionable.

So the comparison of proper techniques became a challenge and seems difficult to determine the best one because their performance is data-dependent. It also depends

P. Meesad et al. (Eds.): *IC²IT2013*, AISC 209, pp. 155–163.
DOI: 10.1007/978-3-642-37371-8_19

on many factors including high accuracy, low run time, low memory requirement and reasonable training time [6]

Thus, we aimed to study and compare three classifier techniques, Neural Network with BP Algorithm, K-Nearest Neighbor and SVM respectively. The performance evolution (i.e. training time,) is done to analyze the various classification algorithms to select the proper classifier and other issues for handwritten digit recognition.

To do so, we took Optical Recognition of Handwritten Digits Data Set from UCI Machine Learning Repository [7]. It is input data for analyzing the various classification techniques which are normalized from 32x32 bitmap to 8x8 bitmap already.

This paper is organized as follows: In next section, we present the related works on handwritten digit classification. In Section 3, we have discussed the three classification techniques (i.e. ANNs, SVM and K-Nearest Neighbor Classifier) used for recognizing handwritten digits. We also compare the classification accuracy among them and analyze the experimental result on Section 4. Finally, Section 5 contains conclusion and future work discussions.

2 Literature Survey

Handwritten digit recognition is an important problem in optical character recognition and it has been used as a test case for theories of pattern recognition and machine learning algorithms for many years.

It can be classified into two categories: online recognition and offline recognition. [8] The on-line recognition technology, which emerges in recent years, uses the geometry and temporal dynamics information of the users' input. The methods for online recognition relatively pose low resource and processing requirement, and may effectively use many kinds of clues to capture users' input customs. They are effective with good user adaptation. [9]

Inversely, Offline Recognition mainly processes and recognizes the user input handwritten digit based on images (the scanned images of handwritten digit, or the digital images transformed from the real time handwritten). A lot of methods have been proposed to solve offline recognition. [10].

In this paper, we focus on Offline Recognition. We have found that a number of researches have been concerned with the offline recognition of handwritten digits. For example:

In [1] the author addressed some issues in designing high reliability system for hand-written digit recognition using SVM classifiers. However, the presented result shows that it is difficult to achieve the good recognition rates by using only one selection method.

In [11] the author proposed a decision tree learning to classify different writing styles of the identical digits. Several direction features were employed to implement the classification. That is, when the stroke direction of some digits is similar, the decision tree learning can classify them properly. However, it is difficult to manage sets of possibilities as more features.

In [12] the author analyzed the learning rate using BP algorithm of Artificial Neural Network for handwritten digit recognition application. That is, they used various parameters such as learning rate, number of hidden neurons in hidden layer, and momentum term to analyze the learning rate which shows its impact for the performance of application.

The performance comparison is also applied and studied in different techniques ex. [6] [13].

Moreover, the comparison of classification Handwritten Digit Recognition have been published continuously. For example in [14] the authors have performed the experiments by extracting structural features from the handwritten digits by using SVM and tree classifier. The recognition rate of SVM classifier is more than the Tree classifier.

And in [15] they use three stage classifiers for hand written digit recognition, at stage 1 and 2 neural network is used, and at stage 3 support vector machine is used. The recognition rate obtained is among the best on MNIST database. The results were also better than single SVM using the same feature set.

Although there are a number of published research provide high recognition rate, little work has been done on analysis processing time consuming and comparison of three techniques, which provide good result, especially among efficiency classifier like ANNs, K-Nearest Neighbor and SVM. [6]

3 The Classification Method

3.1 BackPropagation Artificial Neural Network

Backpropagation Neural Network (BPN), which was developed by Rumelhart, et al. in 1986, is the most common neural network learning algorithm. It should be noted that input signals propagate forwards through the network, and error signals propagate backwards. Weight adjustments are made to reduce error. [16]

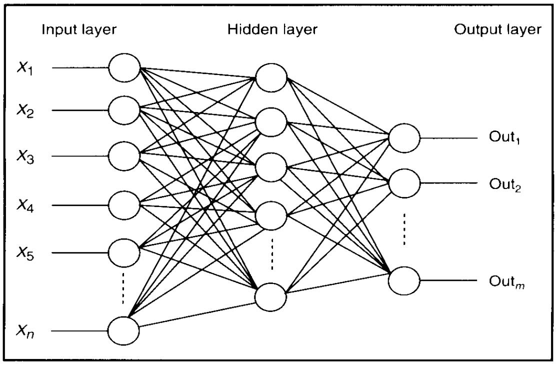

Fig. 1. Structure of backpropagation neural network. (Source: [17], p. 273)

In a BP Neural Network, the learning algorithm has two phases as follows [19][20]:

- Propagation; A training input pattern is presented to the network input layer. The network propagates the input pattern from layer to layer until the output pattern is generated by the output layer.
- Weight Update; If this pattern is different from the desired output, an error is calculated and then propagated backwards through the network from the output layer to the input layer. The weights are modified as the error is propagated.

In this paper, to solve a problem, we have used throughout one hidden layer BP-ANN and designed with three layers, namely an input layer, a hidden layer, and an output layer (Figure 1). Moreover, we used Rule-of-Thumb methods in [18][19] and the result set from [12] approximately define 45 nodes in the hidden layer because Rule-of-Thumb methods is used to calculate the proper node in the hidden layers.

The signal of input from outside spread to the output layer and gives the result through processing layer for layer of neurons in input layer and hidden layer. If the expected output cannot be obtained in output layer, it shifts to the conversed spreading processing and the true value. The error outputted by network will return along the coupled access formerly. The error is reduced by modifying contacted weight value of neurons in every layer and then it shifts to the positive spreading processing and revolves iteration until the error is smaller than the given value. There are no changes of weights in the recognition processing, except the data of the input or output layers. [18] [20]

3.2 K-Nearest Neighbor (KNN)

K-Nearest Neighbor (KNN) is one of the simple methods, which memorize the entire training data and perform classification. The attributes of the test object match one of the training examples exactly. An obvious drawback of this approach is that many test records will not be classified because they do not match any of the training records. A more sophisticated approach, *k*-nearest neighbor (*k*-NN) classification finds a group of *k* objects in the training set that are closest to the test object, and bases the assignment of a label on the predominance of a particular class in this neighborhood. [19][20]

There are three key elements of this approach: a set of labeled objects, e.g., a set of stored records, a distance or similarity metric to compute distance between objects, and the value of *k*, the number of nearest neighbors. To classify an unlabeled object, the distance of this object to the labeled objects is computed, its *k*-nearest neighbors are identified, and the class labels of these nearest neighbors are then used to determine the class label of the object. For the *k*-nearest neighbor classification algorithm is given in [6][19].

3.3 Support Vector Machines (SVMs)

Support Vector Machines (SVM) were introduced as a machine learning method by Cortes and Vapnik (1995). Since SVM is a binary classifier. Thus, the aim of SVM is to find the best classification function to distinguish between members of the two classes in the training data. The metric for the concept of the "best" classification function can be realized geometrically.

In the case of classification, an SVM constructs an optimal separating hyperplane in a high-dimensional feature space. The computation of this hyperplane relies on the maximization of the margin. In this case, we select Non-linear mapping and the kernel function, called Gaussian basis functions (RBF kernel). [6] [19]

This technique defines that the input vectors are only involved through their inner product. Thus, to map the data in a feature space, one does not need to consider the feature space in explicit form. One only has to calculate the inner products of the vectors in the feature space via the kernel function. This is the kernel trick that allows the construction of a decision function that is nonlinear in the input space but equivalent to a linear decision function in the feature space. The equation is described in [6][19].

4 Experimental Results and Evaluation

In this section, we will show the experiments which were performed in order to test our approach. The aim of this experiment is to compare three classifier techniques, Artificial Neural Network using Back Propagation (BP ANNs), K-Nearest Neighbor with Euclidean and SVM with Gaussian kernel, which were experimented and concluded to find out the best techniques. Moreover, the different values, such as error rate, misclassified image rate and processing time for computing step, were also analyzed as well.

To define the classifier for this experiment, we employed three techniques because they are all effective established techniques for pattern recognition and classification area. For example, BPN is one of the most known methods used for character recognition. It is able to segment non-linear separable classes. So, it is a common choice for the digit recognition task.

Another widely accepted technique is the use of SVM. It is beneficial from its generalization power because it is capable of moving the entire problem into a representation of greater dimension, enabling it to separate more complex problems. This is achieved by the use of kernel function, such as the RBF function.

The other one is KNN, which is also commonly used because it is simple and it also produce high recognition rate [13][15][19].

We first download data set, Optical Recognition of Handwritten Digits Data Set, from UCI machine learning repository (http://archive.ics.uci.edu/ml/datasets/) in the form of text file. The data was collected it from a total of 43 people, 30 people contributed for the training set and different 13 people for the test set. Each Digit was written in the form of matrix of 8x8 and contains 64 input attributes of continuous format, which was normalized size from 32x32 bitmap. That is, E. Alpaydin and C. Kaynak, who created this data, used preprocessing programs made available by NIST (The US National Institute of Standards and Technology) to extract normalized bitmaps of handwritten digits from a preprinted form. Some samples data from the UCI database [3] are shown in Figure 2.

In Figure 2, the values of the pixels are normalized and the target values are 16 gray-scale images of size 8x8. That is, they generates an input matrix of 8x8 where each element is an integer in the range 0..16 and the last digit is class code from 0 to 9.

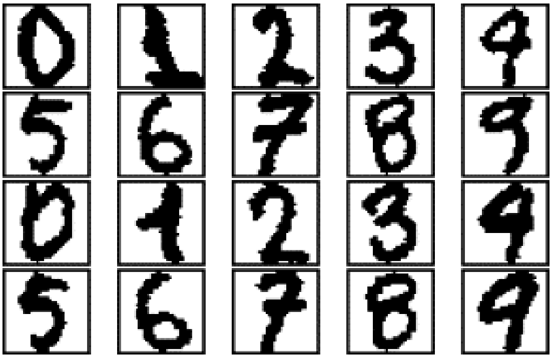

Fig. 2. Example of handwritten digit in the UCI database (Source: [21], p. 2765)

Moreover, the database provides number of items in each class code differently, 389 items in the largest category for class 1 and 3. And 376 items in the smallest category for class 0 and 5 respectively.

The total number of handwritten digits used is 5,620 instances. In the training phase, 3,823 hand written digits (68 percentages) are used as training pattern, 1,797 patterns of different digits (32 percentages) are used as a validation pattern to check. The Classification Techniques such as KK-Nearest with Euclidean, SVM and Artificial Neural Network are applied for Handwritten Digit Recognition.

Three classifiers were tested using the open source Wega Tool Kit [22]. All tests were performed on 1.60GHz Intel CPU T2050 processor under Windows 7.

To do so, we started with experimenting BPN. We set algorithm with one hidden layer and set 45 hidden nodes parameter in Weka because it was proved to be a fast according to Q. Abbas et. al. [12]. Then, the WLSVM software toolbox (i.e. Weka LibSVM - Integrating LibSVM into Weka) was employed for SVM which was seen as a form of implemented LibSVM working in Weka [23]. Moreover, all the running parameters were set as software defaults. Moreover, we also use the implementation of KNN in Weka. That is, the value *k* in all k-related algorithms is set to 1 because G. Daqi. and L. Jie [24] defined the classifier which is approximately equal to the 1-*NN* classifier in classification accuracy.

Table 1. Rates of Different Methods On UCI Data Set

Classifier	Recognition. Rate (%)	Misclassified Image (%)	Error Rate (%)	Recognition. Time. (s)
BP ANN 64-45-10	95.10	0.73	4.17	0.658
SVM with Gaussian kernel	96.93	1.34	1.73	76.112
K Nearest Neighbor	95.66	0.89	3.45	1.034

The result of all the classifier used for hand written digits is shown in the form of Table 1. That is, the individual recognition rate (Rec. Rate), misclassified image (Misc Image), error rates and recognition times (Rec. Time) of the three NNs, SVM and K-NN are presented.

It is obvious that the SVM has a superior recognition rate but it is a level of magnitude slower than the NNs and K-NN. However, an ANNs provide a low misclassification rate. It showed us that SVM is the best classifier but the time is still the main problem. Other methods like K-NN and ANNs give insignificantly worse results, but their time is much quicker.

Table 2. Recognition Rate (%) Per Digit

Digit	*Train (Number of Digits)*	*K-NN (%Recognition Rate)*	*ANNs (%) (% Recognition Rate)*
0	376	97.19	97.12
1	389	97.32	94.32
2	380	96.38	94.65
3	389	95.01	94.16
4	387	94.82	96.18
5	376	95.38	95.02
6	377	93.45	95.94
7	387	95.65	95.32
8	380	94.41	93.13
9	382	96.94	94.74
Average		95.66	95.10

Moreover, the second experiment is to look at the recognition rate obtained for each of the individual ten digits as shown in Table 2. We found that the highest recognition rate was reached just over 97%. And the highest recognition rate obtained was with digit 0 and 1, whereas the lowest recognition rate obtained is for digit 9 and 6. Besides some digits were mainly confused with digits 1 and 7, like digit 3, 5 and 8 due to the similarity in writing these digits when it comes to different handwriting styles.

5 Conclusion and Future Work

This paper concludes that different classifier affects the recognition rate for handwritten digit recognition. To do so, we used three techniques from different good classifier and also used the opensource Wega tool kit for training and testing the dataset, which was from the UCI repository.

The presented results show that SVM is the best classifier to recognize handwritten digits. The highest recognition rates (96.93%) are obtained. But the time of training is the main problem for their use. Conversely, other methods like neural networks give insignificantly worse results, but their training is much quicker. However, to fully evaluate it, further study is necessary.

For future work, we will try to apply some techniques for character recognition in Thai Printed Characters. Moreover, we will consider reasonable factor which affects the recognition rate such as run time, memory requirement, proper parameter and any.

References

1. Gorgevik, D., Cakmakov, D.: Combining SVM Classifiers for Handwritten Digit Recognition. In: 16th International Conference on Pattern Recognition, vol. 247, pp. 529–551. IEEE Press (2002)
2. Oliveria, A.L.I., Mello, C.A.B., Silva, E.R.: Optical Digit Recognition for Images of Handwritten Historical Documents. In: Proceedings of the Ninth Brazilian Symposium on Neural Networks, pp. 166–171. IEEE Press (2006)
3. Khalighi, S.: A Novel OCR System for Calculating Handwritten Persian Arithmetic Expressions. In: International Conference on Machine Learning and Applications, pp. 755–758. IEEE Press (2009)
4. Khalighi, S., Tirdad, P., Rabiee, H.R., et al.: Handwritten Digit Recognition Using State-of-the-Art Techniques. In: International Conference on Machine Learning and Applications, pp. 320–325 (2009)
5. Deng, L.: The MNIST Database of Handwritten Digit Images for Machine Learning Research. IEEE Signal Processing Magazine, 141–142 (2012)
6. LeCun, Y., et al.: Learning Algorithm for classification: A comparison on handwritten digit recognition. CiteSeerX (1995)
7. UCI Machine Learning Repository – Optical Recognition of Handwritten Digits Data Set, `http://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits`
8. Cheng, L.L., Nakashima, K., Hiroshi, S., et al.: Handwritten digit recognition: benchmarking of state of the art techniques. Pattern Recognition, 2271–2285 (2003)
9. Artieres, T., Marchand, J.M., Gallinari, P., et al.: Stroke Level Modeling of On-Line Handwriting Through Multi-Modal Segmental Models. In: The 10th International Workshop on Frontier in Handwriting Recognition, pp. 1–6 (2000)
10. Khorsheed, M.S.: Off-Line Arabic Character Recognition, A Review. Pattern Analysis & Application 5, 31–45 (2002)
11. Jiang, W.L., Sun, Z.X., et al.: User-Independent Online Handwritten Digit Recognition. In: Proceedings of the Fifth International Conference on Machine Learning and Cybernetics, pp. 3359–3364. IEEE Press (2006)

12. Abbas, Q., Ahmad, J., Bangyal, W.H.: Analysis of Learning Rate using BP Algorithm for Hand Written Digit Recognition Application. In: The 2nd International Conference on Information Engineering and Computer Science, pp. 1–4. IEEE Press (2010)
13. Gorgevik, D., Cakmakov, D., Radevski, V.: Handwritten Digit Recognition Using Statistical and Rule-Based Decision Fusion. In: The 11th Mediterranean Electrotechnical Conference, pp. 131–135 (2002)
14. Garg, N.K., Jindal, S.: An Efficient Feature Set For Handwritten Digit Recognition. In: The 15th International Conference on Advanced Computing and Communications, pp. 540–544. IEEE Press (2007)
15. Gorgevik, D., Cakmakov, D.: An Efficient Three-Stage Classifier for Handwritten Digit Recognition. In: Proceedings of the 17th International Conference on Pattern Recognition, pp. 507–510. IEEE Press (2004)
16. Ngu, S.C., Cheung, C., Leung, S.H.: Fast Convergence for Back Propagation Network with Magnified Gradient Function. In: International Joint Conference on Neural Networks, pp. 903–908. IEEE Press (2003)
17. Changa, T.C., Chaob, R.J.: Application of back-propagation networks in debris flow prediction. Engineering Geology 85, 270–280 (2006)
18. Zhang, C., Huang, X.: Off-line Handwritten Digit Recognition Based on Improved BP Artificial Neural Network. In: The IEEE International Conference on Service Operations and Logistics, and Informatics, vol. 1, pp. 626–629. IEEE Press (2008)
19. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Pearson Addison-Wesley (2006)
20. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach, 2nd edn. Prentice-Hall (2002)
21. Kim, H.C., et al.: Constructing support vector machine ensemble. Pattern Recognition 36, 2757–2767 (2003)
22. Weka home page, `http://www.cs.waikato.ac.nz/-mJ/weka/index.html`
23. Integrating LibSVM into Weka Environment, `http://www.cs.iastate.edu/~yasser/wlsvm`
24. Daqi, G., Jie, L.: Kernel Fisher Discriminants and Kernel Nearest Neighbor Classification: A Comparative Study for Large-Scale Learning Problems. In: The 2006th International Joint Conference on Neural Networks, pp. 1333–1338. IEEE Press (2006)