A New Handwritten Number Recognition Method Using HMM Based on MNIST

Ruhua Lu¹ and Yalan Li² +
1 .College of Software and Communication Engineering, Xiangnan University, Chenzhou, Hunan 423000, China
2 .School of Electronic Information and Electrical Engineering, Xiangnan University, Chenzhou, Hunan 423000, China
+ Correspondence author: 47616166@qq.com

Abstract. Taking MNIST data-set as research object, the HMM is introduced into the Handwritten Number Recognition for the first time. After the implementation of the classical HMM training algorithm, some optimization methods are proposed for the problems existing in the training. The general random initialization parameters lead to long training time and unstable data. The training of initialization parameters based on observations can speed up the training and avoid data overflow. The number of iterations in the training process is not positively related to the output probability. In order to obtain an optimal model, after the algorithm converges when the cross entropy loss function of the output probability of two adjacent iterations is the minimum, the training ends. Finally, a comparative experiment between the classical method and the optimization method is carried out in the two stages of training and recognition. In the training stage, compared with the classical method, the optimization method has the advantages of short average training time, high average output probability, fast iteration speed and high accuracy. In the test stage, the accuracy of the optimization method is higher than that of the classical method. The experimental results show that the HMM can be effectively applied in the field of Handwritten Number Recognition. And through the optimization method, to a certain extent, the recognition accuracy is improved. In a word, the method in this paper is an effective and feasible method.

1. Introduction
Since the rising of model recognition technology, many scholars have begun to study Handwritten Numeral Recognition methods. How to use computer to automatically recognize Handwritten Numeral has become a hot issue[1]. With the increasing prosperity of information technology, offline or online Handwritten Numeral Recognition becomes more and more important, mainly in the following aspects. First, Handwritten Numeral Recognition is widely used[2]. Financial system, banking system, tax system and other multi field systems need to automatically handle a large number of handwritten numbers, such as handwritten telephone number, handwritten postal code, handwritten ID card number, etc. Compared with previous manual processing, automatic recognition can improve work efficiency thousands of times. Second, Handwritten Numeral Recognition Method is easy to promote. Arabic numbers are the unified symbols in the world, which do not have the national boundaries for researchers, and are easy to be extended to other similar or even dissimilar fields, such as English, Pinyin, Chinese characters, etc. Thirdly, Handwritten Numeral Recognition System is easy to implement. There are only 10 symbols from 0 to 9 in Arabic numbers, and each symbol contains only a few strokes. It is easy to study and implement, which is beneficial for the verification of new methods and further analysis and processing. Therefore, MNIST data-set, which is from American
National Institute of standards and Technology[3], as the research object is taken in this paper. Combining the HMM (Hidden Markov Model) technology, Handwritten Numeral Recognition is realized.

2. HMM

2.1. HMM Parameters
A Markov process with unknown hidden state parameters is described in HMM, which is a statistical model with powerful classification ability. At present, HMM is widely used in fault diagnosis, speech recognition, character recognition and other fields, and has been proved to be a good classification effect[4]. HMM is a special form of the Markov Chain, whose state can’t be observed directly, but can be observed from the observation vector sequence[5]. Each observation vector is represented by various states through some probability density distribution, so the Hidden Markov Model is a dual stochastic process, which is mainly composed of the following three parameters.

First, the initial probability distribution \( \pi = \{\pi_i | 1 \leq i \leq N\} \), indicates the probability of the \( i \)-th state is the initial state, among \( n \) is the number of states.

Second, the state transition matrix \( A = \{a_{ij} | 1 \leq i \leq N, 1 \leq j \leq N\} \), indicates the probability of the \( i \)-th state transition to the \( j \)-th state. Here is a hypothesis that the hidden state at any time only depends on its previous hidden state.

Third, the observation sequence probability \( B = \{b_j(O_t) | 1 \leq j \leq N\} \) indicates the probability of outputting the observation value \( O_t \) under the \( j \)-th state condition. This is the core parameter of HMM. According to its distribution, HMM can be divided into discrete HMM and continuous HMM, and discrete HMM is used in this paper.

In other words, the HMM is represented by a triple as \( \lambda = (\pi, a, b) \).

2.2. HMM Representation of Handwritten Numbers
The HMM observation sequence is expressed as a set of continuous observation values, and the HMM representation of handwritten numbers is that the handwritten numbers are converted into the observation value sequence in HMM. Handwritten number is an image of \( M1 \times M2 \) pixels, which is converted into a one-dimensional array with the size of \( M1 \times m2 \). After standardization and normalization, it is used as the observation value sequence in HMM, and it is recorded as \( O = \{O_t | 1 \leq t \leq M1 \times M2\} \). In the model training of handwritten number, because the handwritten number involved in the training contains multiple samples, it corresponds to multiple observation value sequences in HMM. If the number of samples is \( L \), the observation value sequence is \( O = \{O^l_t | 1 \leq t \leq M1 \times M2, 1 \leq l \leq L\} \), and \( O^l_t \) is read as the observation value of the \( l \)-th sample and the \( t \)-th time.

3. Handwritten Number Recognition Method Based on HMM

3.1. Overall Working Mode
Overall working mode in this paper is shown in Figure 1.
3.2. Model Training of Handwritten Numeral

HMM training is a continuous iterative process, which generally uses the classic Baum Welch algorithm[6]. The initialization parameters are input into the program, and the new value of the parameters is obtained after one time of the iteration. Then the parameters are input again, the second new value of the parameters is obtained again after the next iteration. These operations are repeated many times until the algorithm converges.

Here, the convergence condition of the algorithm is determined by the probability change of two adjacent iterations. If the output probability and the parameters of the \(i\)-th iteration are set as \( P_i \) and \( \lambda_i = (\pi^i, A^i, B^i) \), the number of samples is \( L \), and the \( l\)-th training sample is recorded as \( O(l) \), then the following formula is obtained.

\[
P_i = \frac{1}{L} \sum_{l=1}^{L} P(O(l)/\lambda_i)
\]  

(1)

In the process of continuous iteration, the output probability will be more and more large, which means that the parameters obtained will be more and more close to the training sample data. However, when the probability increases to a certain extent, it will increase slowly or even decrease, which shows that the algorithm converges and the training should be ended.

3.3. Handwritten Number Recognition

The forward algorithm[7] in HMM is used in Handwritten Numeral Recognition. The handwritten number to be recognized is input into the handwritten number trained model database, and the output probabilities are calculated based on the trained models, which include ten models from the 0-th training model to the 9-th training model. The number corresponding to the maximum output probability is used as the identification result. The ten models from the 0-th training model to the 9-th training model is recorded as \( \lambda_0, \lambda_0, \ldots, \lambda_9 \), then the recognition result is calculated by the formula 2.

\[
\text{index} = \arg \max_{i} (P(O/\lambda_i) | 0 \leq i \leq 9)
\]  

(2)

4. Optimization of HMM Training Algorithm

4.1. Optimization of Model Parameter Initialization

In order to improve the accuracy of Handwritten Numeral Recognition, a large number of training samples are needed, so model training will take a long time, and it is easy that train model fall into local optimum or a denominator value close to zero or even 0. Good initialization parameters alleviate these problems to some extent. Therefore, the general method of random initialization is not adopted here. The initialization parameters are calculated based on the sample data.

The number of hidden states in HMM is \( K \). All the observed value sequences \( O = \{O_t^l | 1 \leq t \leq M1 \times M2, 1 \leq l \leq L\} \) are divided into \( k \) classes according to some rules, corresponding to the \( K \) hidden states respectively. The parameters in the HMM are initialized as follows.

First, Initialization of \( \pi \). \( M_i \) represents the number of handwritten numbers belonging to the \( i \)-th state at the first time in all observation sequence. Initialization of \( \pi \) is shown in the following formula.

\[
\pi_i = M_i / \sum_{l=1}^{K} M_l \quad 1 \leq i \leq K
\]  

(3)

Second, Initialization of A. \( \theta_{ij} \) represents the number of handwritten numbers belonging to the \( i \)-th state at the \( t \)-th time and belonging to the \( j \)-th state at the \( t+1 \)-th time in all observation sequence. Initialization of A is shown in the following formula.

\[
a_{ij} = \theta_{ij} / \sum_{j=1}^{K} \theta_{ij} \quad 1 \leq i \leq K, 1 \leq j \leq K
\]  

(4)

Third, Initialization of B. First, R discrete values are set. Generally speaking, R discrete values are all the observed values in all observation value sequence. The handwritten numbers belonging to the \( j \)-th state is assigned to the corresponding discrete value from R discrete values, and \( \beta_j \) is noted the
number of handwritten numbers belonging to the j-th state. When the k-th discrete value belongs to the j-th state, it is obvious that

\[ b_j(k) = \frac{1}{\beta_j}, \quad 1 \leq j \leq K, 1 \leq k \leq R \]  

(5)

4.2. Optimization of the Algorithm Convergence Conditions
It is unnecessary to use the differences between two output probabilities to judge the convergence condition. The output probability is not that the greater the better, but that the test model corresponding to the maximum output probability is correct under the condition of ten models from 0 to 9. This problem can be improved is using cross entropy to measure the distribution difference of the two output probabilities. The output probabilities of the i-th sample in the current iteration and the last iteration are recorded as \( P_i \) and \( Q_i \). The cross entropy loss of \( P_i \) and \( Q_i \) is recorded as follows formula.

\[ H(P_i, Q_i) = -P_i \log Q_i \]  

(6)

Therefore, the cross entropy function is defined as

\[ f(\lambda) = \frac{1}{L} \sum_{i=1}^{K} H(P_i, Q_i) = -\frac{1}{L} \sum_{i=1}^{K} P_i \log Q_i \]  

(7)

In the formula 7, \( \lambda \) represents the current model parameter, and an optimal model can be obtained according to the cross entropy loss function \( f(\lambda) \), which has been minimized. At this time, the training algorithm converges and the training iteration can end.

5. MNIST Data-Set
There are ten kinds of numbers from 0 to 9 in the dataset MNIST handwritten number data-set, which is a large handwritten number database collected and organized by American National Institute of standards and technology. There are 60000 handwritten numbers in the training set for model training, and 10000 handwritten numbers in the test set for testing to calculate the accuracy of the training model. Each handwritten number consists of two items: a label and an image. The label refers to the Arabic number represented by the image, and the image is a 28 * 28 pixel. After reading the data, the image is stored in the database as a one-dimensional array with the length of 28 * 28, and the label field is added in the database to store the corresponding label value.

In the training set, 60000 handwritten numbers are divided into ten categories, and each category has the same label value. Then this training data participate in the following HMM training separately. In order to get the real output probability in the training process, each kind of sample data is divided into two parts. In the first part, 80% handwritten numbers are selected from the front to participate in the training. The remaining 20% of handwritten numbers are used to calculate the output probability of iterative process of each step.

6. Handwritten Number Simulation Experiment

6.1. HMM Training Simulation Experiment
There are ten kinds of handwritten numbers from 0 to 9 in this training sample, which participate in HMM training respectively, and ten training models are obtained: model 0, model 1, model 2, model 3, model 4, model 5, model 6, model 7, model 8 and model 9. The original classical method and this optimization method are adopted for the training, and the specific training data of ten kinds of model is shown in Table 1.
Table 1. The training of handwritten numbers

|                                | The Original Classical Method | The Optimization Method |
|--------------------------------|-------------------------------|-------------------------|
| Average Training Time (s)      | 40                            | 15                      |
| Average Times of Iterations    | 45                            | 25                      |
| Average Output Probability (log)| -30                           | -13                     |
| Accuracy                       | 94%                           | 97%                     |

The output probability changes in the iteration process affect the quality of the model. Take the number 1 which is easier to identify and the number 5 which is harder to identify as examples, the output probability changes in the iteration process are shown in Figure 2 and Figure 3. It can be concluded from the two figures that the output probability of the optimization method does not increase at 23 iterations, and the classical method does not converge until 43 iterations. Compared with the output probability, the output probability of the optimization method is greater than that of the corresponding classical method. The superiority of the optimization method is proved.

![Figure 2. The output probability of number 1](image1)

![Figure 3. The output probability of number 5](image2)

6.2 Handwritten Number Test Simulation Experiment

After ten kinds of model are trained, unknown handwritten numbers can be recognized. In order to verify the effectiveness of this method, 10000 handwritten numbers in MNIST test set are used in this test experiment. The accuracy obtained is shown in Table 2.

Table 2. The test accuracy of handwritten numbers

|         | 0    | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | Average |
|---------|------|------|------|------|------|------|------|------|------|------|---------|
| The Original Classical Method | 88%  | 97%  | 85%  | 85%  | 80%  | 93%  | 95%  | 98%  | 85%  | 88%  | 90%     |
| The Optimization Method       | 88%  | 100% | 90%  | 94%  | 83%  | 95%  | 96%  | 100% | 88%  | 88%  | 93%     |

The experimental results show that the accuracy of the original classical method and the optimization method is 90% and 93% respectively. These accuracy rates are ideal. It shows that HMM can be applied to handwritten number recognition, and the optimization method in this paper is more effective.

7. Conclusion

HMM is a classical pattern recognition method, which has a strong classification ability. In this paper, HMM is introduced into handwritten number recognition, and the classical HMM training algorithm and the optimization algorithm are used to train handwritten number pattern. Experimental results show that the optimization algorithm is effective. However, MNIST is also the research object, and the
accuracy of deep learning method is higher. Finding out these problems will be my research goal in the next stage.

8 Acknowledgement
This work has been supported by The Scientific Research Project of Hunan Provincial Education Department(19C1716,19C1723), The Science and Technology Plan Projects of Chenzhou ([2018]102), The Xiangnan University Youth Talent Support Program, Xiangnan University[2019]124 and the Scientific Research Project of Xiangnan University(2018XJ24).

9 References
[1] Ashif Ali and Shaista Khan 2020 Multiscript Handwritten Numeral Recognition. Global Sci-Tech 12 1.
[2] Weiwei JIANG and Le ZHANG 2020 Edge-SiamNet and Edge-TripleNet: New Deep Learning Models for Handwritten Numeral Recognition. IEICE Transactions on Information and Systems E103.D 3.
[3] Emma Lejeune 2020 Mechanical MNIST: A benchmark dataset for mechanical metamodells. Extreme Mechanics Letters 3 6.
[4] Jun Wang, Huiwen Zheng, Yang Yang, Wanyue Xiao, Taigang Liu and Quan Zou 2020 PredDBP-Stack: Prediction of DNA-Binding Proteins from HMM Profiles using a Stacked Ensemble Method. BioMed Research International 20 20.
[5] Putland R L, Ranjard L and Constantine RI 2018 A hidden Markov model approach to indicate Bryde's whale acoustics. Ecological Indicators 84 479-487.
[6] Sassi, I. Anter, S. and Bekkhoucha, A 2020 A new improved baum-welch algorithm for unsupervised learning for continuous-time hmm using spark. International Journal of Intelligent Engineering and Systems 13 214-226.
[7] Andreas Sand, Martin Kristiansen, Christian NS Pedersen and Thomas Mailund 2013 zipHMMlib: a highly optimised HMM library exploiting repetitions in the input to speed up the forward algorithm. BioMed Central 14 10.