Hindawi
Security and Communication Networks
Volume 2019, Article ID 9580862, 5 pages
https://doi.org/10.1155/2019/9580862

Research Article

Analysis of DES Plaintext Recovery Based on BP Neural Network

Sijie Fan[1,2] **and Yaqun Zhao**[1,2]

[1]*State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, China*
[2]*Information Engineering University, Zhengzhou, China*

Correspondence should be addressed to Sijie Fan; 826733148@qq.com

Received 16 May 2019; Accepted 22 October 2019; Published 11 November 2019

Guest Editor: Leonel Sousa

Backpropagation neural network algorithms are one of the most widely used algorithms in the current neural network algorithm. It uses the output error rate to estimate the error rate of the direct front layer of the output layer, so that we can get the error rate of each layer through the layer-by-layer backpropagation. The purpose of this paper is to simulate the decryption process of DES with backpropagation algorithm. By inputting a large number of plaintext and ciphertext pairs, a neural network simulator for the decryption of the target cipher is constructed, and the ciphertext given is decrypted. In this paper, how to modify the backpropagation neural network classifier and apply it to the process of building the regression analysis model is introduced in detail. The experimental results show that the final result of restoring plaintext of the neural network model built in this paper is ideal, and the fitting rate is higher than 90% compared with the true plaintext.

1. Introduction

The study of cryptography mainly includes two aspects, cryptographic design and cryptanalysis. There are independent and mutually unified relationships between them [1]. Block cipher is an important branch of symmetric cryptography. It uses the same key in encryption and decryption and plays a very important role in information and communication security. We hope to use existing cryptanalysis methods to design cryptanalysis methods that can resist all cryptanalysis methods. At the same time, we also hope to use the updated cryptanalysis methods to find some security flaws in cryptanalysis algorithms.

Modern cryptosystems often use methods to expand the key space or increase the complexity of encryption and decryption, and use some mathematical problems as the theoretical basis, which greatly improves the requirement of computational power in cryptographic deciphering. Sometimes the cost of traditional cryptographic deciphering methods may exceed the value of cryptographic deciphering. Artificial neural network (ANN) is the same discipline as cryptography for studying information processing. Neural network algorithm has the characteristics of nonlinear massively parallel-distributed processing and has strong high-speed information processing and uncertainty information processing capability. Using neural networks to solve cryptographic problems will provide a new research idea for cryptography.

In 2008, Bafghi et al. used a recurrent neural network to solve the problem of finding the least-weight multibranch path between two known nodes in the differential operation graph of block cipher. The main idea was to minimize the loss function of the neural network [2]. In 2010, Alallayah et al. considered the black box characteristics of neural networks, combined with system identification technology and adaptive system technology, simulated the neural model of the cryptanalysis target system and could guess the key from a given ciphertext [3]. In 2012, Alani et al. used a new cryptanalysis attack on DES (Data Encryption Standard) and 3DES (Triple Data Encryption Algorithm) cryptographic algorithms. The attack implemented was a known plaintext attack based on a neural network. In this attack, they trained a neural network to retrieve plaintext from ciphertext without retrieving the keys used in encryption. Compared with other attacks, this method reduced the number of known plaintexts required and reduced the time required to

perform a full attack [4]. The above methods were less able to directly restore the plaintext sequence, and the experimental procedures of the related literature were mostly based on the simplified cryptographic encryption algorithm and had very high requirements on the computing power.

In this paper, we choose DES algorithm as a case study of block cipher. We propose to use BP (backpropagation) neural network algorithm to simulate the mapping relationship between ciphertext and plaintext. The ciphertext obtained by DES encryption is converted into binary string, which is fed to our improved BP neural network as input after processing according to the preprocessing method defined in this paper. The difference between predicted output and true plaintext is compared for the purpose of cryptanalysis. Compared with previous work, the plaintext recovered by this experiment has a better fitting effect with true plaintext. According to the error rate defined in this paper, the experimental error rate can be controlled below 10%.

The second section of this paper briefly introduces the development history and basic working principle of block cipher. The third section briefly introduces the principle of BP algorithm and the modification we have made to it, thus successfully building a regression model. The fourth section shows our experimental process and results.

2. Brief Introduction to Block Ciphers

Block cipher is one of the important systems in modern cryptography, which is an important part of many cryptosystems. Block cipher usually refers to a kind of cipher algorithm that can only deal with a piece of data of a certain length at a time. Here, the "piece" is called a block. The number of bits in a block is called the block length. Specifically, the principle of block cipher is to divide the plaintext message sequence into a group $(m_1, m_2, \ldots, m_n)$, $(m_{n+1}, m_{n+2}, \ldots, m_{2n}), \ldots$ encrypts it according to a set of fixed encryption algorithms under the control of the key $K = k_1, k_2, \ldots, k_m$, and outputs a group of ciphertext $(c_1, c_2, \ldots, c_n)$, $(c_{n+1}, c_{n+2}, \ldots, c_{2n}), \ldots$. The model is shown in Figure 1.

Under the same key, the block cipher transforms the input plaintext group with length i equally, so it only needs to study the transformation rules for any group [5].

A cryptosystem consists of five parts (plaintext P, ciphertext C, key K, encryption transformation E, and decryption transformation D). It satisfies the following conditions [6]:

(1) $P = \{p_1, p_2, \ldots, p_n\}$ is a limited set of plaintext

(2) $C = \{c_1, c_2, \ldots, c_n\}$ is a limited set of ciphertext

(3) $K = \{k_1, k_2, \ldots, k_m\}$ is a limited set of keys

(4) $E = \{e_1, e_2, \ldots, e_n\}$ is a limited set of encryption change rules

(5) $D = \{d_1, d_2, \ldots, d_n\}$ is a limited set of decryption change rules

(6) $\forall k \in K, \exists e_k \in E, d_k \in D, \text{s.t.}, d_k(e_k(p)) = p$, $(\forall p \in P), e_k : P \longrightarrow C, d_k : C \longrightarrow P$

DES is a method of encrypting 64-bit plaintext m by 16 rounds of encryption processing with 56-bit key and obtaining 64-bit ciphertext. We choose DES as the block cipher for research because it can change the encryption key and network level faster, encrypting at a faster rate and reducing the impact of other factors. The specific description is as follows:

(1) Enter 64-bit plaintext and perform initial replacement IP

(2) Divide the plaintext into two parts, each part of 32 bits, which are represented by L_0 and R_0, respectively

(3) After adding the key, perform 16 rounds of operation f, $f : \{0,1\}^{32} \times \{0,1\}^{48} \longrightarrow \{0,1\}^{32}$

(4) After 16 rounds, the left and right bit strings are exchanged and then connected for inverse replacement

(5) Output 64-bit ciphertext

3. Backpropagation Neural Network

Artificial neural network is a cross-disciplinary field of multidisciplinary research in brain science, neuropsychology and information science. It is a research hotspot in high-tech fields in recent years. Its research goal is to explore the mystery of human intelligence by studying the composition mechanism and thinking mode of the human brain and then to make the machine have human-like intelligence by simulating the structure and working methods of the human brain [7].

BP (backpropagation) neural network usually refers to the multilayer forward neural network based on error rate backpropagation algorithm, and the error rate backpropagation algorithm is the most successful neural network learning algorithm to date. It uses the error rate after output to estimate the error rate of the direct predecessor layer of the output layer and then uses this error rate to estimate the error rate of the previous layer. After such a layer of backpropagation, the error rate estimates of all other layers are obtained [8]. The BP neural network topology includes an input layer, a hidden layer, and an output layer. The model is shown in the following Figure 2.

The BP neural network model is often used for classification. It has high self-learning, self-adaptive, and fault-tolerant ability. That is to say, BP neural network can simulate the mapping relationship between input and output through continuous learning, and this process is reflected in the dynamic adjustment of network weights and thresholds. After repeated training, the error rate is stable in an acceptable range. At this time, the corresponding network parameters can be finally determined to achieve local optimum. If the local nerve unit of BP neural network is damaged, it has little effect on the global training results [9].

Based on the above work, we modify a classifier based on BP algorithm and realize the regression model of BP neural algorithm. A large number of plaintext pairs are fed into the model to get the difference between the output plaintext and the true plaintext. The modified model is as follows.

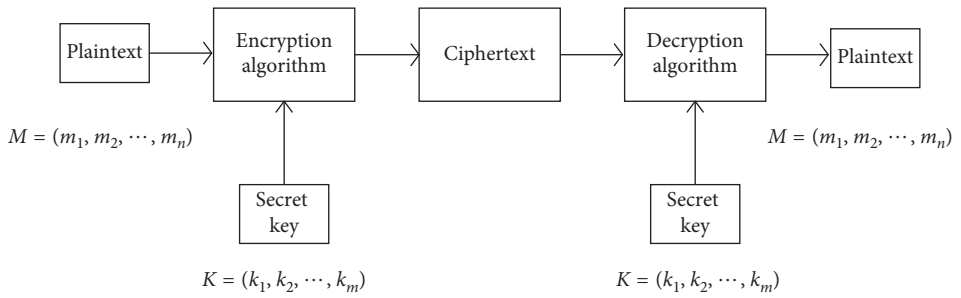

Figure 1: Block cipher workflow.

3.1. Forward Propagation

$$\begin{aligned} &\text{Input Layer}: a_0 = x, \\ &\text{Layer 1}: z = \sigma(w_1 x + b_1), \\ &\text{Layer 2}: y = \sigma(w_2 z + b_2). \end{aligned} \tag{1}$$

3.2. Backpropagation

$$\text{Loss function}: L = \frac{1}{2}[f(x) - y]^2,$$

$$\text{Layer 2}: y = k_2, \quad k_2$$

$$\text{Error} = \delta_2$$

$$\begin{aligned} \frac{\partial L}{\partial b_2} &= \frac{\partial L}{\partial k_2}\frac{\partial k_2}{\partial b_2} \\ \frac{\partial L}{\partial w_2} &= \frac{\partial L}{\partial k_2}\frac{\partial k_2}{\partial w_2} \end{aligned} \tag{2}$$

$$\text{Layer 1}:$$

$$\begin{aligned} \text{Error} = \delta_1 &= \frac{\partial L}{\partial k_1} \\ &= \frac{\partial L}{\partial k_2}\frac{\partial k_2}{\partial z}\frac{\partial z}{\partial k_1} \\ &= (y - f(x)) w_2 \odot \sigma'(z) \odot \sigma'(k_1), \end{aligned}$$

where $k_1 = w_1 x + b_1$ and $\odot$ represents the multiplication of the corresponding position of the matrix.

$$\begin{aligned} \frac{\partial L}{\partial b_1} &= \frac{\partial L}{\partial k_1}\frac{\partial k_2}{\partial b_1} = \delta_1, \\ \frac{\partial L}{\partial w_1} &= \frac{\partial L}{\partial k_1}\frac{\partial k_1}{\partial w_1} = \delta_1 x. \end{aligned} \tag{3}$$

4. Experimental Process

We use DES blocks with variable plaintext and constant keys to convert variable plain text to binary text, which are stored in different document texts. Each document has a 3.2 million-bit binary number. We use it as plaintext, after DES (electronic codebook mode) encryption, the corresponding cipher is obtained. In the data collection phase, data from the California Institute of Technology Caltech-256 dataset are selected, and the data into 1001 files are stitched as plaintext, and the size of files are 512 KB, ten of them were selected as our experimental data [10]. The intercepted image of the plaintext file and the ciphertext file obtained by encrypting them are shown respectively in Figures 3 and 4.

Since the single plaintext document is too large and the computer computing power is limited, we do some of the same processing for each ciphertext text and compress the ciphertext structure. We take an 8-bit binary number from start to finish in turn to convert it to a decimal number, so that each ciphertext becomes a 100000×1 matrix. Similarly, for each plaintext, we take a 8-bit binary number from start to finish in turn to convert it to a decimal number, so that each plaintext becomes a 100000×1 matrix. Then we input all the samples of the preprocessed ciphertext and the plaintext into the modified BP neural network and obtain the output. We compare this output with the expected plaintext to make the BP neural network a simulated decryption system. Convert each decimal digit of the output of the neural network into binary. If the number of digits is not enough, the high digits are filled with 0 and then all of them are connected to restore the plaintext effect.

We define the mean squared error of the output matrix and the processed matrix representing the true plaintext as the evaluation criteria for the experimental effect. That is, the output matrix of the modified BP neural network is $A' = (a'_{ij})$, the processed plaintext text matrix is $A = (a_{ij})$, and the evaluation criterion is

$$\text{error} = \sum \frac{\left(a_{ij} - a'_{ij}\right)^2}{n}. \tag{4}$$

The experimental process is as follows.

4.1. Input Plaintext. The known plaintext (binary text) is encrypted according to des (ecb mode) to obtain the corresponding ciphertext, and the known ciphertext is processed and converted into a format that can be fed into the neural network;

4.2. Neural Network Training. Modify the BP neural network model, change it from the classification model to the

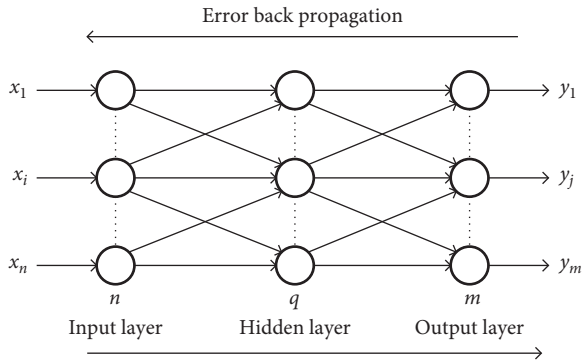

Figure 2: Backpropagation neural network model.

```
00010001101111011000000111011111011111000000001100000000010000
00001000000000010000000100000010100010111101110000011000000100
00000010101101100100111110100000011011010111001000111000101000
10011100010111101110000000110000011010000101000000010000100011
01111011000000111011111011111000000001100000000001000000010000
00000010000000100000010000010111011101100000110000000100000001
01011010101010001011010000001100010100000100011111010100111111
00000000100001011011011000001100000100100000010000000101101110
```

Figure 3: Plaintext sample.

```
00011001101000000010011011000101001001011001010001001000111010 1
01001111111011001110011111100011010110111111001001000110111000 0
11011111001001100011101110001001110100011110110110011111101000 1
10101100001001101010001110111011111111101010001110000000100001 0
11010011101011111001110111000011110001001011101111101111101010 11
00000001000000001010110101010100010111010000001100010100000100 0
11111010100111111000000001000010111011110010011000111011100010 0
```

Figure 4: Ciphertext sample.

regression model, and constantly adjust the internal parameters of the neural network until the best training effect is achieved.

4.3. Output Result. The experimental error rate is stable at an acceptable level, and the improved model is used to achieve plaintext recovery.

5. Results and Discussion

After repeated training, the parameters of BP neural network are constantly adjusted. We used the sigmoid activation function, and the number of trainings exceeds 1000. The error rate can be stabilized at about 10%. The experimental results are shown in the following Table 1:

The correlation between data is too large, which will inevitably lead to unreliable and unpredictable networks. Therefore, our input is encrypted by DES blocks of constant keys, but special attention should be paid to the number of trainings for controlling neural networks. Overtraining the network, the network may become over-fitting [11, 12], so it may not be possible to accurately predict the plaintext outside the training set, resulting in an excessive error rate.

Table 1: Experimental results.

Number of experiments	Error rate
1	0.3632983
2	0.3375012
3	0.2215154
.	.
.	.
.	.
998	0.1022956
999	0.1022163
1000	0.1021363

The modified BP neural network optimizes the weight in backpropagation by the steepest descent method, which converges straight down to the local minimum point in the weight space. However, in addition to trying several different weight initial values, there is no better suggestion than the difference in the output of the neural network [13]. Correct selection of the learning rate effectively controls the size of the step size used to modify each weight in the multidimensional weight space [14]. If the selected learning rate is too large, the local minimum may be continually overrun, causing oscillations and slowly converge to a lower error rate. If the learning rate is too low, the number of iterations required may be too large, resulting in a slow neural network performance [15].

6. Conclusion

In this experiment, we propose to apply the modified BP neural network algorithm to cryptanalysis and implement it. Here, we define mean-square error for analyzing output. Efficiency can be further improved by increasing the number of samples used to train the neural network and adjusting the weights and biases of the neurons in each layer.

Although the DES algorithm is no longer used in new commercial and public applications, the reason why we choose the DES algorithm for cryptanalysis is that the design structure of the DES algorithm is also reflected in other cryptographic algorithms, such as the gost algorithm and the camella algorithm. In addition, many software still compatible with DES algorithm, because there is no real way to completely crack DES algorithm.

In the future, a lot of work on weight selection and adaptation (training) of neural networks still needs to be completed, especially the possibility of hardware implementation is still an area worthy of further study. There may be different types of neural networks for cryptanalysis [16], resulting in unexpected results.

Data Availability

The data used to support the findings of this study are included within the article. Data can be used for free by everyone to verify the experimental results.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by National Key Research and Development Project 2016–2018 (2016YFE0100600), State Key Laboratory of Information Assurance Technology Open Fund Project (KJ-15-008), and State Key Laboratory of Cryptography and Science.

Supplementary Materials

The supplementary material contains experimental data, including 10 plain text data, with a total size of 24 MB, and 10 cipher text data, obtained by encrypting ten plain text data with DES encryption algorithm, with a total size of 40 MB. The plain text data source is from the California Institute of Technology Cal tech-256 data set. (*Supplementary Materials*)

References

[1] Cheng and Hai and Qun Ding, "Overview of the block cipher," in *Proceedings of the 2012 Second International Conference on Instrumentation, Measurement, Computer, Communication and Control*, IEEE, Harbin, China, December 2012.

[2] A. G. Bafghi, R. Safabakhsh, and B. Sadeghiyan, "Finding the differential characteristics of block ciphers with neural networks," *Information Sciences*, vol. 178, no. 15, pp. 3118–3132, 2008.

[3] K. M. Alallayah, A. H. Alhamami, W. AbdElwahed, and M. Amin, "Applying neural networks for simplified data encryption standard (SDES) cipher system cryptanalysis," *International Arab Journal of Information Technology*, vol. 9, no. 2, pp. 163–169, 2012.

[4] M. M. Alani, *Neuro-Cryptanalysis of DES and Triple-DES. Neural Information Processing*, Springer, Berlin, Germany, 2012.

[5] C. de Canniere, A. Biryukov, and B. Preneel, "An introduction to block cipher cryptanalysis," *Proceedings of the IEEE*, vol. 94, no. 2, pp. 346–356, 2006.

[6] D. Mills, "Review of cryptography: theory and practice by D. R. Stinson," *Cryptologia*, vol. 31, no. 1, pp. 87-88, 2007.

[7] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, no. 1, pp. 59–69, 1982.

[8] T. Kohonen, *Self-Organizing Feature Maps. Self-Organization and Associative Memory*, Springer, Berlin, Germany, 1988.

[9] H. D. Landahl, W. S. Mcculloch, and W. Pitts, "A statistical consequence of the logical calculus of nervous nets," *The Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 135–137, 1943.

[10] G. Griffin, A. Holub, and P. Perona, *Caltech-256 Object Category Dataset*, California Institute of Technology, Pasadena, CA, USA, 2007.

[11] I. V. Tetko, A. E. P. Villa, T. I. Aksenova et al., "Application of a pruning algorithm to optimize artificial neural networks for pharmaceutical fingerprinting," *Journal of Chemical Information and Computer Sciences*, vol. 38, no. 4, pp. 660–668, 1998.

[12] I. V. Tetko, T. I. Aksenova, V. V. Volkovich et al., "Polynomial neural network for linear and non-linear model selection in quantitative-structure activity relationship studies on the internet," *SAR and QSAR in Environmental Research*, vol. 11, no. 3-4, pp. 263–280, 2000.

[13] H. R. Guo and Z. M. Li, "A method of improving generalization ability for neural network based on genetic algorithm," in *Proceedings of the 2010 IEEE International Conference on Intelligent Computing & Intelligent Systems*, October 2010.

[14] J. P. Yang, Q. Li, Z. Liu, and X. L. Yuan, "Research of improved bp algorithm based on self-adaptive learning rate," *Computer Engineering & Applications*, vol. 45, no. 11, pp. 56–58, 2009.

[15] S. Wermter, C. Weber, W. Duch et al., *Artificial Neural Networks and Machine Learning—ICANN 2014*, Springer, Berlin, Germany, 2014.

[16] S. Aditya and N. Nadir, "Cryptography based on artificial neural networks and chaos theory," *International Journal of Computer Applications*, vol. 133, no. 4, pp. 25–30, 2016.