Machine Learning and Artificial Intelligence
J.-L. Kim (Ed.)

doi:10.3233/FAIA230762

Encryption and Decryption Using Deep Neural Network

Yun FU[a], Jingyi FU[b], Jinxin WEI[c1]
[a] *Shandong Heze Yellow River Engineering Co. Ltd, 274000, China*
[b] *Yellow River Heze Bureau, YRCC, Heze, 274000, China*
[c] *Vocational School of Juancheng, Juancheng 274600, China*

Abstract. An auto-encoder which can be split into two parts is designed. The two parts can work well separately. The top half is an abstract network which is trained by supervised learning and can be used to classify and regress. The bottom half is a concrete network which is accomplished by inverse function and trained by self-supervised learning. It can generate the input of abstract network from concept or label. It is tested by tensorflow and mnist dataset. The abstract network is like LeNet-5. The concrete network is the inverse of the abstract network. Through test, encryption and decryption can be achieved by abstract network and concrete network add jump connection and negative feedback with absolute function. When binary encoding is used, although the encrypted vector is four bits, the result of decryption has the same quality as one-hot encoding because of the jump connection and negative feedback. The parameter of DNN is secondary key, the architecture of DNN is primary key. Secondary key can be shared by all the people, primary key can be shared by sender and receiver. The key can be generated by training the DNN. When big dataset is used for encryption, the classes are far bigger, the label may be something in the world, numbers, words, or attributes represented by float number. The label can use the mix of one-hot encoding and binary encoding, it is harder to attack. Through analysis, it is safe for most situations.

Keywords. Encryption, decryption, deep neural network, regression, auto-encoder, absolute function, primary key, secondary key, jump connection, negative feedback, one-hot encoding, binary encoding

1. Introduction

Cryptography is concerned with encoding a sensitive piece of data in a form which is unintelligible or meaningless to any human or machine other than the intended party which has the decoding mechanism to regenerate the original data from the encoding. Cryptanalysis on the other hand is used to breach cryptographic security systems and gain access to the contents of encrypted messages [1]. Most of the protocols for encryption in commercial use today are n bit key based or hash functions like RSA, DES, 3DES, AES, SHA. In this paper we are concerned with images as data to be encrypted at source, sent securely and decrypted at destination [2].

[1] Corresponding Author: Jinxin WEI, Vocational School of Juancheng, Juancheng 274600, China. E-mail: wjxabai@163.com.

Deep learning neural networks have been an area of active research as a means for both encryption and decryption for some time. Techniques like Hopfield neural networks, chaotic time delayed neural networks, Autoencoder networks, Generative and GAN based etc have shown encouraging results. However, this research has not found much application outside research where conventional algorithms continue to dominate. One of the reasons for this is that deep neural networks as in autoencoders are computationally expensive to run involving many matrix operations [3]. In this paper, many new methods are proposed, such as absolution function as activation function, jump connection and negative feedback as approximation method, the mix of on-hot encoding and binary encoding as label encoding.

In another paper [4], an auto-encoder which can be split into two parts is designed. The two parts can work well separately. The top half is an abstract network which is trained by supervised learning and can be used to classify and regress. The bottom half is a concrete network which is accomplished by inverse function and trained by self-supervised learning. It can generate the input of abstract network from concept or label. The input can change to any form by encoder and then change it back by decoder through inverse function. A picture can change to label that is encryption, then change it back from label that is decryption. So, encryption and decryption can be realized by the autoencoder.

The abstract network is like LeNet-5. The abstract network is 3 layers convolutional neural network and 2 layers fully connected network, no padding, no maxpooling. The concrete network is the inverse of the abstract network. The architecture is shown in Figure 1.

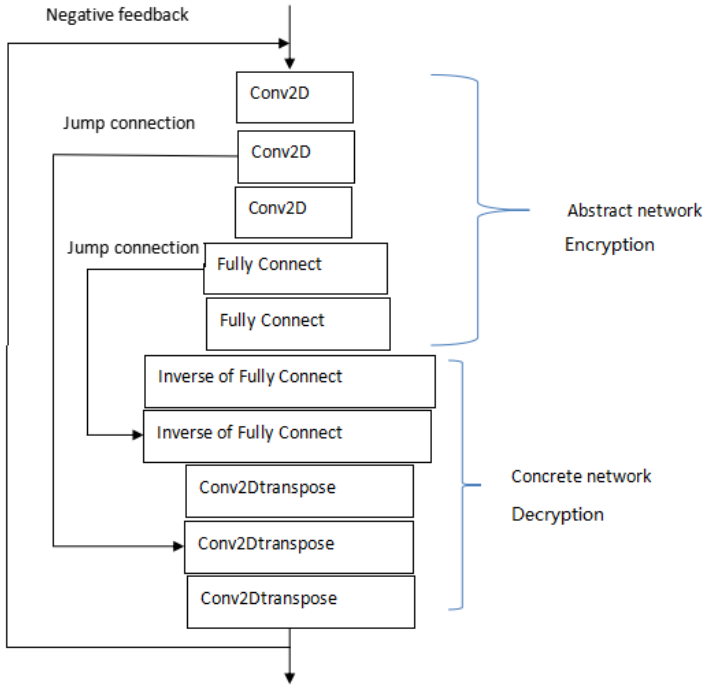

Figure 1. The Architecture of Network.

2. The Inverse Function

The following are inversions of the functions. The function of fully connected layer is

$$y = wx + b \tag{1}$$

so the inverse function is

$$y = w^{-1}x - w^{-1}b \tag{2}$$

Because the linear function's inverse function is also linear, so the fully connected layer's structure of concrete network is the same as abstract network layer's. Because w is a matrix, we need the inverse of w, so the w needs to be square matrix. But the real situation is that the dimension of w is determined by the neuron numbers of the two-layer next to each other, so this network cannot reproduce the inputs, but approximate the inputs.

The function of absolute [5] is

$$y=|x| \tag{3}$$

The inverse function is

$$y=|x| \tag{4}$$

Because the inverse of the softmax function which used by classification is a approximation method [4], it can change the distribution of the probability, so don't use softmax in order to generate the input well. You can read my another paper [4] for the detail. Most deep learning frameworks all have the transpose of convolution function, just use conv2Dtranspose layer.

3. Test

3.1. Test on mnist dataset and one-hot encoding

The test is on mnist dataset and autoencoder which architecture is showed in Figure 1. Activation function is absolute function [5]. Loss function is mse [6], optimizer is adam [6]. We train the top half first, then set it as untrainable, then train the bottom half. The results are the following pictures (Figure 2(a)-Figure 2(c)). The left image is input of abstract network, the label of input is the argmax function of the prediction of output of abstract network, the right image is the output of concrete network. The encryption is achieved by regression which treats classification as regression, the label is represented by vector of float number. The decryption is achieved by regression. In Figure 2(a), the output is not very similar to the input, the background of the output is a little dark.

In order to generate the output which is very similar to input, jump connection and negative feedback (which are in Figure 1) can be used. The output of one layer of abstract network which can be seen as the knowledge of features it has learned before is connected to the input of the symmetrical layer of the concrete network, then take the

mean as the new input. If the output of layer two is B, the input of layer nine is $B \pm \xi$, ξ is the error because the inverse function which used before is the approximation function. So $\frac{1}{2}(B+B\pm\xi)=B\pm\frac{1}{2}\xi$, the error decreases. The more jump connection (more knowledge about features), the less training time, the more similarity. It fits the process of learning. Figure 2(b) shows the result of layer two jump connecting with the layer nine. Figure 2(c) shows the result of jump connection and negative feedback. It shows that the background is no longer dark when we use negative feedback. Why is negative feedback? Inspired by principle of automatic control, negative feedback is added. Because the whole network is like the proportional integral differential parts of automatic control, our aim is to make output very similar to input, so let the subtraction of input and output as the new input will decrease the difference of output and input.

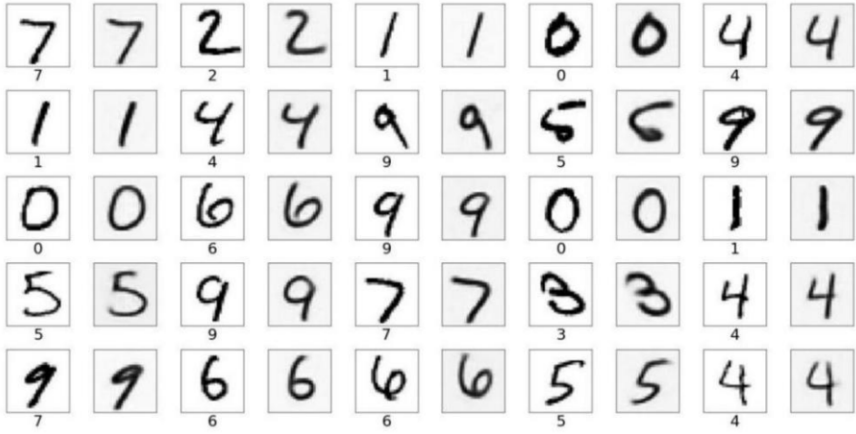

Figure 2. (a) Absolute Function.

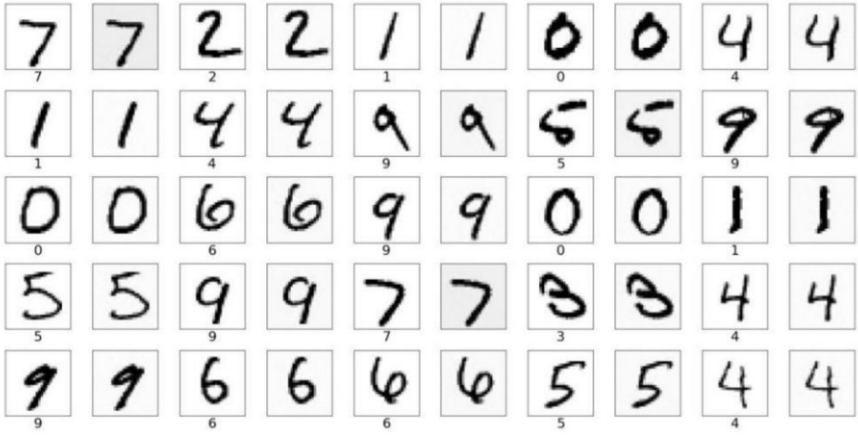

Figure 2. (b) Absolute Function and Jump Connection.

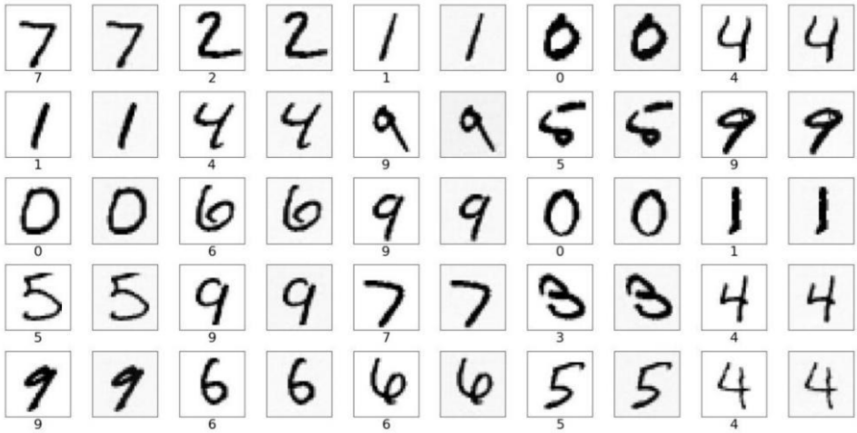

Figure 2. (c) Absolute Function and Jump Connection and Negative Feedback, One-hot Encoding.

3.2. Test on modified mnist dataset and binary encoding

The label of the Figure 2 is one-hot encoding, there is another view when we use binary encoding [7]. When binary encoding used, the length of the label is 4bits vs 10bits with one-hot encoding. For example, number 4 is 0000100000 for one-hot encoding while it is 0010 for binary encoding. Binary encoding is suit for large dataset which has large classes. When the classes are 1 million, there is 20 bits for the label which uses binary encoding. Figure 3 shows the result of binary encoding. The left image is input of abstract network, the label of input is the round function of the prediction of output of abstract network, the right image is the output of concrete network. In Figure 3, although the encrypted vector is four bit, the result of decryption has the same quality as one-hot encoding because of the jump connection and negative feedback.

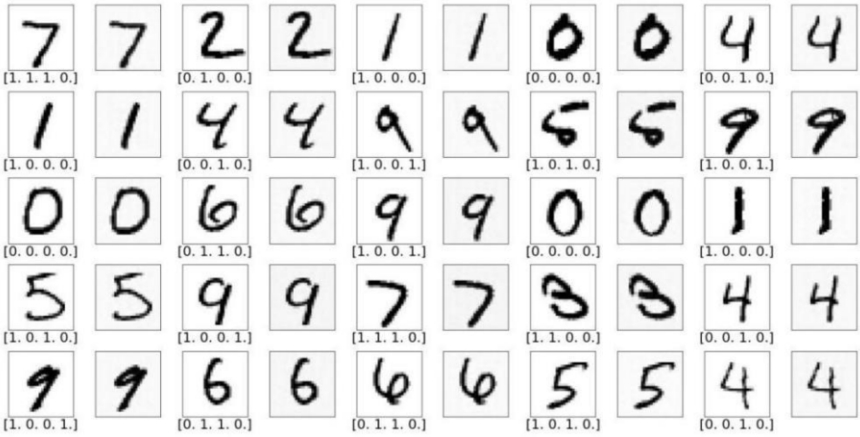

Figure 3. Absolute Function and Jump Connection and Negative Feedback, Binary Encoding.

3.3. Analysis

How to design the encryption and decryption? The parameter of DNN is the secondary key, and the architecture of DNN is the primary key. Secondary key can be shared by all the people, while primary key can only be shared by sender and receiver. The key can be generated by training the DNN. When you have the secondary key but without the primary key, you can't encrypt. It is because you don't know the architecture of the model such as how many layers? what's the type of layers? how many units in one layer? What kind of activation function? The parameter is a list of numbers, you don't know how to split it into meaningful numbers without the architecture of the model. If you have the primary key but without the secondary key, you can't encrypt either. You have the architecture of the model, then you can train the DNN to get the parameter. But every time you run the DNN, the parameter is different from the previous one. The reason is that the training dataset which through change the order of label is different, and the parameter initializes as different numbers. Only when you have the primary key and the secondary key, you can encrypt and decrypt. The key can change by training the DNN. You can change the architecture of the model to get different primary key, meanwhile the secondary key is changing. You can change the order of the label to increase the number of the encryption message. The mnist is simple, when big dataset is used for encryption and decryption, the classes are far bigger. The label may be something in the world like numbers, words, or attributes [8] represented by float number. The label can use the mix of one-hot encoding and binary encoding, and it is more difficult to attack.

4. Conclusion

Through test, encryption and decryption can be achieved by abstract network and concrete network add jump connection and negative feedback with absolute function. The absolute function is suit for generation tasks, so the output is more similarity with the input. The jump connection and negative feedback is approximation method, so the quality of decryption is very good. The parameter of DNN is secondary key, the architecture of DNN is primary key. Secondary key can be shared by all the people, primary key can be shared by sender and receiver. The key can be generated by training the DNN. When big dataset is used for encryption and decryption, the label can use the mix of one-hot encoding and binary encoding, it is harder to attack. Through analysis, it is safe for most situations.

References

[1] Schaefer E. An introduction to cryptography and cryptanalysis. California's Silicon Valley: Santa Clara University, 2009.
[2] Stinson DR. Cryptography: theory and practice. Chapman and Hall/CRC, 2005.
[3] Ian G, Yoshua B, Aaron C. Deep Learning, MIT Press, 2016, p. 505-526.
[4] Wei JX, Ren QY. A Functionally Separate Autoencoder, Proceedings of the Future Technologies Conference (FTC) 2020, Volume 1. p. 553-560.
[5] Wei JX, Hou Z. Activation Function: Absolute Function, One Function Behaves more Individualized, TechRxiv, 2021. Preprint. https://doi.org/10.36227/techrxiv.17639525.v3
[6] Tensorflow Tutorials and Apis, https://tensorflow.google.cn/learn, last accessed 2022/10/20.

[7] Wei JX, Hou Z. Binary-encoding for Label, TechRxiv, 2022. Preprint. https://doi.org/10.36227/techrxiv.21353763.v1
[8] Wei JX, Ren QY. Multi-attribute Recognition, the Key to Universal Neural Network, Proceedings of the Future Technologies Conference (FTC) 2020, Volume 1. p. 599–605.