

Neural Network Adversarial Attack Method Based on Improved Genetic Algorithm

Yang Dingming¹, Cui Yanrong^{1†}, Yuan Hongqiang²

(1. School of Computer Science, Yangtze University, Jingzhou 434023, China; 2. School of Urban Construction, Yangtze University, Jingzhou 434000, China)

Abstract: Deep learning algorithms are widely used in fields such as computer vision and natural language processing, but they are vulnerable to security threats from adversarial attacks because of their internal presence of a large number of nonlinear functions and parameters leading to their uninterpretability. In this paper, we propose a neural network adversarial attack method based on an improved genetic algorithm. The improved genetic algorithm improves the variation and crossover links based on the original genetic optimization algorithm, which greatly improves the iteration efficiency and shortens the running time. The method does not need the internal structure and parameter information of the neural network model, and it can obtain the adversarial samples with high confidence in a short time by the classification and confidence information of the neural network. The experimental results show that the method in this paper has a wide range of applicability and high efficiency for the model, and provides a new idea for the adversarial attack.

Key words: genetic algorithm; neural network; DCNN; neural network adversarial attack; algorithm improvement

0 Introduction

Szegedy et al [1] showed for the first time that a highly accurate deep neural network can be misled to make a misclassification by adding a slight perturbation to an image that is imperceptible to the human eye, and also found that the robustness of deep neural networks can be improved by adversarial training. Such phenomena are far-reaching and have attracted many researchers in the field of adversarial attacks and deep learning security. Akhtar et al [2] investigated 12 attack methods and 15 defense methods for neural networks against attacks. The main attack methods are finding the minimum loss function additive term [1], increasing the loss function of the classifier [3], methods to restrict the l_0 parametric number [4], and changing only one pixel value [5]. The review literature [6] collates and summarizes and adds to the subsequent research results based on the review literature [2], and proposes new classification methods for classifying adversarial sample attack methods based on white-box and black-box environments. Studying neural networks against attacks is not only beneficial to understand how neural networks work, but also increases the robustness of neural networks by training the adversarial samples.

Nguyen et al [7] continued to explore the question "What is the difference between computer vision and human vision" based on the work of Szegedy et al [1]. They used Evolutionary Algorithm (EA) to iterate over directly encoded images, Evolutionary Algorithm to iterate over CPPN (Compositional Pattern-Producing Network) encoded images, and Gradient Ascent [[8], [17], [18]] (Gradient Ascent (GA) method to generate high confidence adversarial samples (fooling images). They used evolutionary algorithms to obtain high-confidence adversarial samples on a LeNet model pre-trained on the MNIST dataset [19] and on an AlexNet model pre-trained on the ILSVRC 2012 ImageNet dataset [[20],[21]] (provided by the Caffe software package [22]), respectively. of the adversarial samples. The high-confidence adversarial samples generated by the CPPN encoding method are then used as the training set to retrain the LeNet and AlexNet models, and the experimental results show that the LeNet model always generates high-confidence adversarial samples no matter how many times it is trained, while the AlexNet model has difficulty in generating high-confidence adversarial samples after passing the trained adversarial samples.

In this paper, we improve the original adversarial attack model based on the experiments of Nguyen et al [7] by using an Improved Genetic Algorithm (IGA) to combine the traditional Artificial Neural Network (ANN) and modern Deep Convolutional Neural Network (DCN). Convolutional Neural Network (DCNN) as experimental subjects. First, the confidence function in the neural network model is stripped out as the fitness function of the genetic algorithm, then the crossover and variation aspects of the genetic algorithm are improved

Fund project: Hubei Province Technology Innovation Special Major Project (2019AAA011).

Authors' Biographies: Dingming Yang (1997-), male, native of Xinfeng, Jiangxi, master student, main research interests are deep learning, big data mining, cloud computing and distributed systems; Yanrong Cui (1968-), female (corresponding author), native of Xiantao, Hubei, professor, master, PhD, main research interests are network security, high-performance networks (cyanr@yangtzeu.edu. CN); Yuan Hongqiang (1988-), male, Hubei Jingzhou, PhD student, main research interests are structural health monitoring.

to speed up the convergence, and finally the improved genetic algorithm is iterated continuously to generate high confidence adversarial samples. The experimental results show that the directly encoded binarized images and grayscale maps have equivalent properties in the experiments, and the ANN experimental subjects can converge faster than the DCNN experimental subjects.

1 Related Work

1.1 CNN Model

The concept of neural network was first introduced in 1943 when psychologist W.S. McCulloch and mathematical logician W. Pitts created a computational model of neural network (NN) based on mathematics and an algorithm called threshold logic [9]. In 1986, Rumelhart [10] presented the complete reverse propagation algorithm (Backpropagation, BP) propelled the development of neural networks. Three years after the BP algorithm was proposed, LeCun [11] chose to train a multilayer Convolutional Neural Network (CNN) with the BP algorithm to recognize handwritten digits, which was the prototype of CNN. In 1998, LeCun formally proposed the LeNet-5 model [12], which was the first real CNN model. With the deepening of the layers of CNN models, the DCNN model was a hit at the ImageNet competition in 2012, getting two firsts and outperforming the second place by nearly 10% in correctness [13]. In 2014, the Visual Geometry Group of Oxford proposed the VGG model [14], demonstrating that increasing the depth of the network can somewhat influence the final performance of the network.

1.2 Adversarial Attack

In 2014, Szegedy et al [1] first introduced the concept of adversarial samples. Szegedy pointed out that the input-output mapping of deep neural network learning is largely quite discontinuous and can mislead a highly accurate deep neural network to make a misclassification by adding slight perturbations to the image that are imperceptible to the human eye. Meanwhile, Szegedy's method of finding the minimum loss function additive term (BFGS) proposed in the paper can enable neural networks to make misclassifications.

Some recent approaches to neural network countermeasures against attacks are listed below.

- AdvHat

Komkov et al [15] used ArcFace (alias nsightface) as an experimental subject to deceive the detection of ArcFace by posting a printed image of the adversarial sample on the hat, which was generated by the Fast Gradient Symbolic Method (FGSM) method. The total Loss is shown in Equation 1, TV loss makes the posted adversarial sample and the original image of the face smoother after overlay, and Similarity loss measures the similarity of the two samples.

$$\mathcal{L}_{\text{final}}(x, a) = \mathcal{L}_{\text{sim}}(x, a) + \lambda \cdot TV(x) \quad (1)$$

- OPAD

OPAD (Optical ADversarial attack) [16] is a solid attack and the principle of OPAD is to use structured illumination to change the appearance of the target object. The system consists of a low-cost projector, a camera and a computer. The challenge of the problem is the nonlinearity of the radiometric response of the projector and the spatially varying spectral response of the scene.

- MvUPA

Tang Jing et al [17] proposed a 3D adversarial sample attack scheme based on multi-view depth panoramic view universal perturbation attack (referred to as MvUPA). The MvUPA network framework consists of two parts, branch A for generating shape feature vectors and branch B for generating shape feature vectors with added perturbations. In designing a universal attack method for multi-view retrieval tasks, it is necessary to represent similarity in on the retrieved target and the matching target are disturbed.

2 Materials and Methods

2.1 MNIST Test Dataset

The MNIST dataset (Mixed National Institute of Standards and Technology database) [19] is one of the most famous datasets in the field of machine learning and is used in a variety of applications ranging from simple experiments to published paper research. It consists of handwritten digital images from 0 to 9. The MNIST image data is a single-channel grayscale map of 28*28 pixels, with each pixel taking values between 0 and 255, with 60,000 samples in the training set and 10,000 samples in the test set. The general approach to the use of the MNIST dataset is to first learn with the training set, and then use the learned model metric to what extent it can correctly classify the test set is correctly classified [23].

As shown in Figure 1a, the distribution of Loss values of the 10 classes of samples from the MNIST test set in the ANN used for the

experiments in this paper, it can be seen from the figure that each class of samples has its own specific aggregation area except for the concentrated distribution around the Loss value of 0. As shown in Fig. 1b, a non-uniform exponential sampling with a base of 2 and a sampling interval of 0-784. The horizontal coordinates are the sampling points of 2^x , and the vertical coordinates are the loss values of the "0" samples after the sampling points are converted to binarized images. The images characterize the variation of Loss values in the similar binarized images of single class samples. From the vertical coordinates, the sampling points are concentrated in the vicinity of the Loss value of 3.4-3.6, which is consistent with the distribution in Figure 1a; from the horizontal coordinates, the decimal representation of the binarized image of the sampling points is concentrated in the vicinity of 2^{50} and 2^{700} , and the samples in the middle are not only less similar, but also more scattered or even outlier in the Loss value.

The above analysis of Figure 1 shows that the smaller the Loss value does not mean that the ANN has a higher probability of correctly classifying the image, because even if the image is correctly classified, its Loss value will not be concentrated around a fixed value (the correctly classified test samples have a very large difference in the Loss value between them except for the points with a Loss value of 0). Therefore, this paper uses the confidence output of the neural network as the adaptation degree of the improved genetic algorithm, trying to restore the real input image by the output of the neural network.

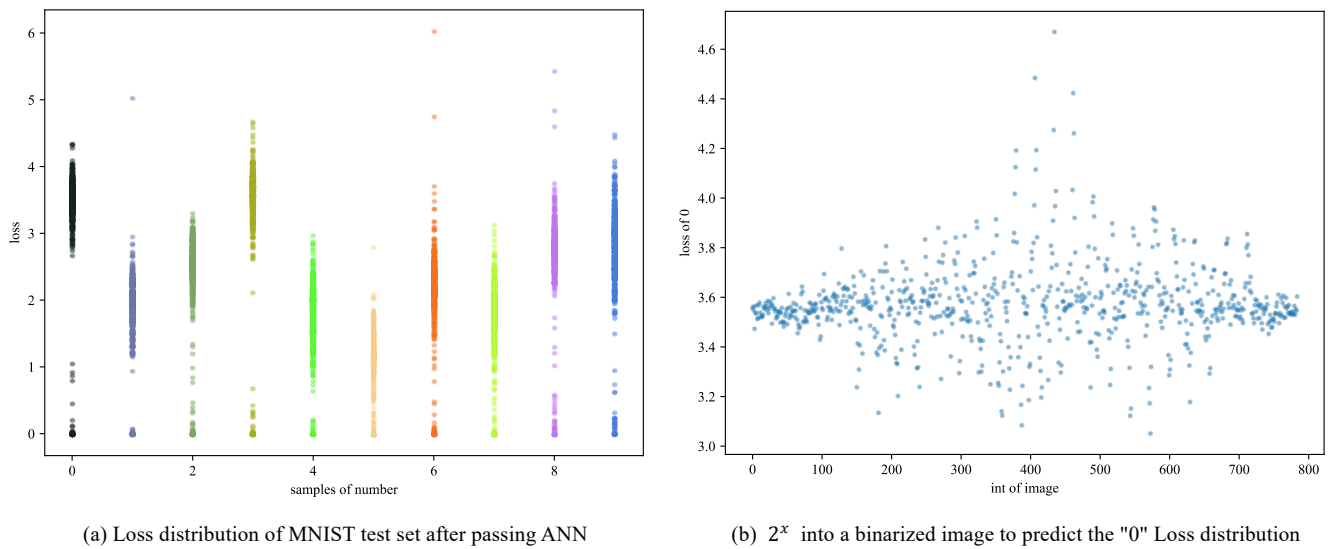


Fig. 1 MNIST test set analysis

2.2 Genetic Algorithm Improvement

"Genetic algorithms are a class of simulated evolutionary algorithms proposed by Professor Holland of the University of Michigan in 1969 and later summarized by DeJong, Goldberg and others" [24]. The genetic algorithm first encodes the problem, then calculates the fitness function, then selects the parent and the mother after roulette, and finally cross-variates to produce the children with high fitness, and then iterates many times to produce the individuals with high fitness, which is the satisfactory or optimal solution of the problem. The Sample Genetic Algorithm (SGA) uses single-point crossover and basic bitwise variation to reflect the information exchange between individuals and local search, and does not rely on gradient information [25], so the SGA is able to find the global optimal solution.

In order to make the confidence converge as soon as possible, speed up the experiment and reduce unnecessary time overhead. In this paper, the crossover and variational links of the SGA are improved (IGA). As shown in Algorithm 1 is the Python like pseudo code of the improved crossover algorithm. single point crossover of SGA is to generate random numbers in the range of parental chromosome lengths, and then intercept the first half of the father's chromosome and the second half of the mother's chromosome to interbreed the offspring according to the generated random numbers. In this paper, we improve the algorithm by trying to cross genes in the parental chromosome length range one by one, calculating the confidence level, and picking the offspring individuals with the highest confidence level. According to the experimental data, such an improvement can speed up the convergence of the confidence level.

Algorithm 1 crossover with confidence as evaluation

Input: father's gene, mother's gene, confidence function

Output: child's gene

1: **function** crossover(father, mother, confidence)

```

2:  best_confidence=float.MIN_VALUE
3:  best_child=np.zeros(father.size)
4:  for i in range(father.size):
5:      current_child=np.zeros(father.size)
6:      current_child=np.append(father[0:i],mother[i:])
7:      current_confidence=confidence(current_child)
8:      if current_confidence>best_confidence then
9:          best_confidence=current_confidence
10:         best_child=current_child.copy()
11:     return best_child
12: end function

```

The pseudocode of the improved variation algorithm is shown in Algorithm 2. The basic bit variation of SGA sets a relatively large variation rate, and when the generated random number is smaller than the variation rate, the variation is performed on any one gene of the incoming offspring chromosome. In this paper, we improve the algorithm by first setting a small variation rate, and then selectively mutate each gene of the incoming offspring chromosome, i.e., when the generated random number is smaller than the variation rate, the gene is mutated, and when the traversed gene position is larger than half of the chromosome length, the second half of the gene has relatively less influence on the result, so the variation rate is set to twice the original one. This ensures that the first half of the gene and the second half of the gene have equal chance of mutation, and can mutate at the same time. The variability of the whole chromosome is $1 - (1 - 0.025)^{392} \times (1 - 0.05)^{392}$, which can greatly improve the species diversity and ensure the stability of the species at the same time, and according to the experimental data, it can accelerate the convergence of the confidence.

Algorithm 2 mutate child with alter each gene if rand number less than mutate rate

Input: child's gene

Output: mutated child's gene

```

1: function mutate(child)
2:   _mutate_rate=.025
3:   for i in range(child.size):
4:       if i>child.size//2 then
5:           _mutate_rate=.05
6:       if random.random()<_mutate_rate then
7:           child[i] = 0. if child[i]==1 else 1.
8:   return child
9: end function

```

Figure 2 shows the speed of convergence of the adaptation (confidence) of the improved genetic algorithm (GA with A1+A2) and the original genetic algorithm (GA), where the red line shows the original genetic algorithm (GA), the orange line shows the addition of the improved algorithm 1 (GA with A1) in this paper to the original genetic algorithm, and the green line shows the improved genetic algorithm (GA with A1+A2) after the improvement of algorithm 1 and The green line is the improved genetic algorithm (GA with A1+A2) after improving Algorithm 1 and Algorithm 2 based on the original genetic algorithm. The horizontal coordinate is the number of iterations, and the vertical coordinate is the confidence level of the all-black binarized image after passing through the neural network. From the figure, we can learn that the improved genetic algorithm converges best, and converges after the second iteration; followed by the improved genetic algorithm after Algorithm 1, although the convergence is not as good as the improved genetic algorithm, there is a significant increase in the confidence level after each iteration, and the confidence level is 58.19% after 12 iterations; the original genetic algorithm relies on the diversity of the initial population, in which the initial population The original genetic algorithm relied on the diversity of the initial population, and evolution was almost suspended in the case of consistent chromosomes, and the confidence increase was not significant.

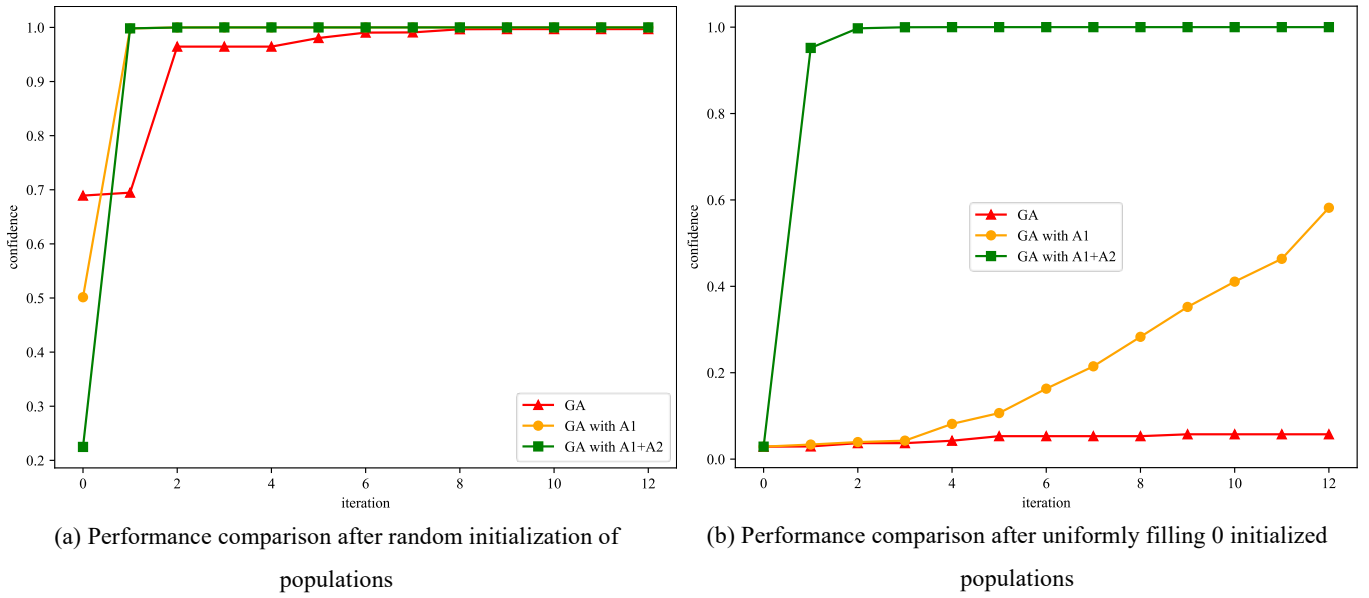


Fig. 2 Performance comparison of improved genetic algorithm

3 Experiment

3.1 Experiment Preparation

In this paper, a pre-trained neural network is used as the experimental object, and the training process is not described. The classification accuracy and Loss values of the experimental subjects are shown in Table 1.

Table 1 The parameter of object for experiment

Subject	Accuracy	Loss
ANN	96.84%	0.1007
DCNN	99.35%	0.9632

The flow chart of the experiments in this paper is shown in Figure 3, where the population size of a specific size (set to 100 in this paper) is first generated and then input to the neural network to obtain the confidence level of the specified labels. In order to reduce the computational overhead and to reflect the difference between the experiments in this paper and those of Nguyen et al [7], the input is simplified to a binarized image of 28*28 in this paper. Fathers and mothers with relatively high confidence levels are selected among 100 individuals by roulette selection, and then the children are generated using the improved crossover link in this paper, and the children form new populations by improving the variation link until iterating to the specified rounds. Finally, the individual with the highest confidence level is picked from 100 individuals, which is the binarized image with the highest confidence level after passing the neural network.

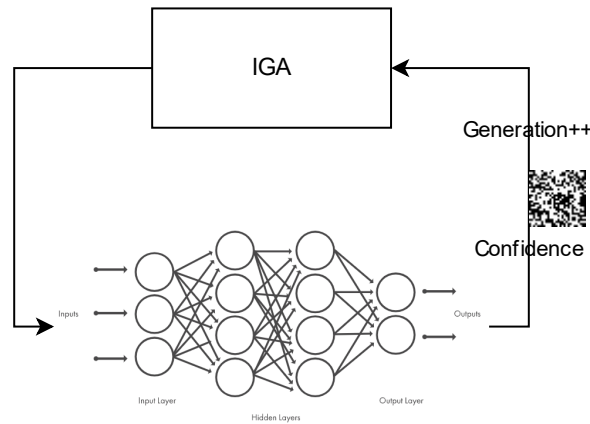


Fig. 3 The process of experiment

3.2 Using ANN as Experimental Subject

A simple two-layer neural network structure is shown in Figure 3, where a 28*28 grayscale map is first flattened, regularized and fed into the neural network. After a fully connected layer of 50 "neurons", a nonlinear expression is added by the ReLU function, and after a fully connected layer of 10 "neurons", the 10 outputs are scaled to the interval of 0-1 by the Softmax function to characterize the The final

output, the cross-entropy loss function, is used to train the neural network. The mathematical description of the neural network is shown in Equation 2.

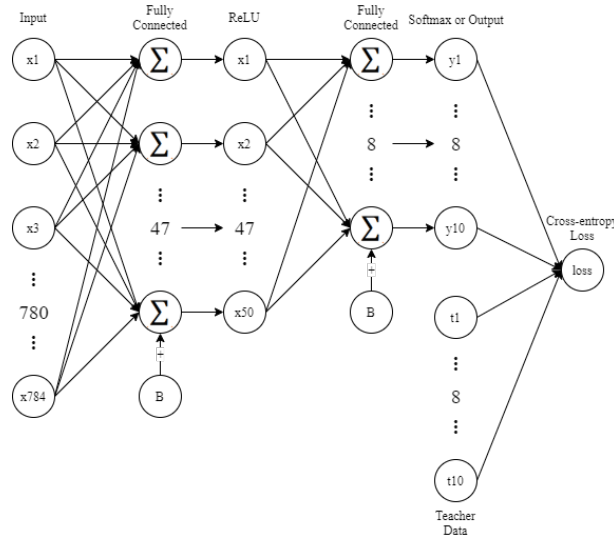


Fig. 4 The structure of ANN for experiment

$$Confidence = \text{Softmax}(\sum_{j=1}^{50} (w_j \times \text{ReLU}(\sum_{i=1}^{784} (w_i \times x_i) + b_i)) + b_j) \quad (2)$$

3.3 Using DCNN as Experimental Subject

Here, considering that in the traditional neural network the fully connected layer is not spatially perceptive to the image data after spreading, which may lead to a large difference between the results and the training set samples, the author again used DCNN as the experimental object in the same environment. The core idea of DCNN is to combine three structural ideas: local perceptual field, weight sharing (or weight replication), and temporal or spatial subsampling. The feature map extracted by the convolutional layer in DCNN contains the relative position information of the original map [26].

The structure of DCNN used for this experiment is shown in Figure 5. First, the input 28*28 grayscale map is zero padded around (Zero Padding) and then passed through three units composed of convolution and pooling, then two fully connected layers, and finally the confidence information of the specified label is output by the Softmax function.

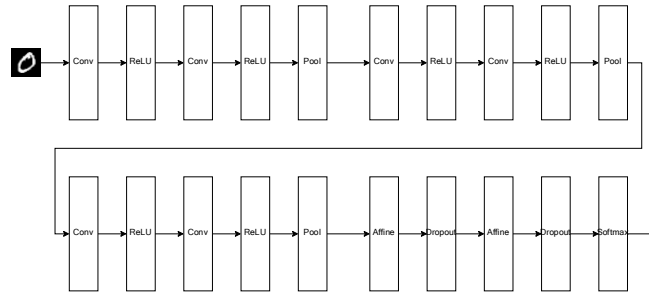


Fig. 5 The structure of DCNN for experiment

3.4 Analysis of Experimental Results

Figure 6 shows the confidence change of the adversarial samples generated by the improved genetic algorithm after iterating through the neural network. As shown in Fig. 6a, the confidence level of the ANN after 12 iterations is shown, and the confidence level of samples "2-8" reaches more than 90% after the first iteration. Sample "5" has the fastest convergence, reaching 99.99% confidence after the first iteration, while sample "1" has the slowest convergence. Figure 6b shows the confidence change of DCNN after 99 rounds of iterations. "After 99 iterations, the confidence level of sample "6" and sample "4" is 78.84% and 68.97%, respectively.

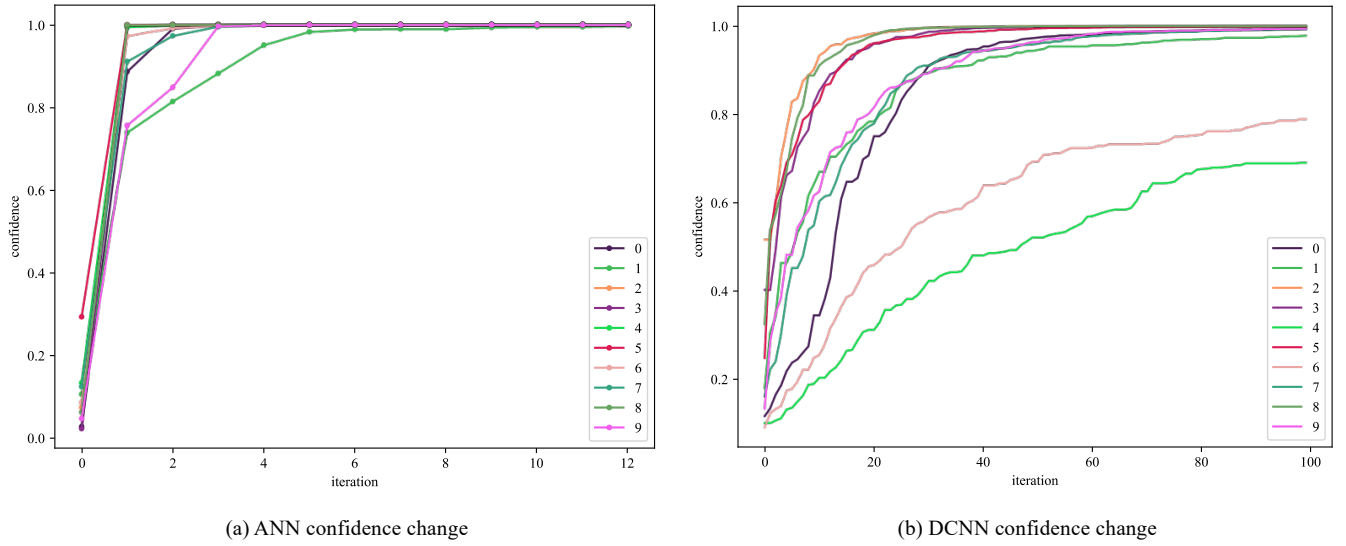










Fig. 6 The confidence change of the binary image after iteration

As shown in Table 2, the ANN experimental group is the experimental group of traditional artificial neural network, and the confidence of sample "0" generated after 12 iterations is 99.999996%, which is much higher than that of sample "0" of ANN control group. However, the binarized image displayed is a "snowflake mess" that cannot be recognized by the naked eye. In the DCNN experimental group, the binarized image of sample "1" generated after 999 iterations has 99.94% confidence after passing DCNN, which is much higher than the confidence of sample "1" in the MNIST test set in the DCNN control group. The same is true for the "snowflake garbled code", which is incomprehensible to the human eye.






Table 2 Statistical table of experimental results

Figure	Group	Label	Iteration	Confidence
	ANN Control	0	—	99.987613%
	ANN Experimental	0	12	99.999996%
	ANN Control	1	—	99.89%
	ANN Experimental	1	12	99.59%
	DCNN Control	0	—	96.71%
	DCNN Experimental	0	204	99.61%
	DCNN Control	1	—	99.44%
	DCNN Experimental	1	999	99.94%

As shown in Table 3, the statistics of the results after initializing the population with the MNIST test set, because the overall confidence level of the population initialized with the test set is relatively high, the increase in confidence level during iteration is smaller. "The confidence level of the sample in the DCNN control group is 99.56%, and after 10 iterations, the confidence level of the sample is 99.80%, and the number "1" becomes vertical; after 89 iterations, the confidence level of the sample is 99.98%. After 89 iterations, the confidence level is 99.98%, and the number "1" tends to "decompose" gradually.

As shown in Figure 7, the reason for this situation is probably that the confidence level as a function of the image input is a multi-peak function, and the interval where the test set image distribution is located is not the highest peak of the confidence level function. This causes the initial population of the test set to "stray" from some pixels in the images generated by the improved genetic algorithm.

Table 3 Statistical table of experimental results after using the MNIST test set to initialize the population

Figure	Group	Iteration	Confidence
	ANN Control	—	99.959%
	ANN Experimental	4	99.961%
	DCNN Control	—	99.56%
	DCNN Experimental	10	99.80%
	DCNN Experimental	89	99.98%

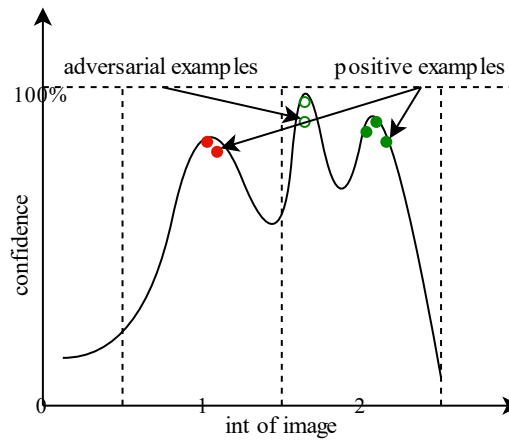


Fig. 7 The confidence curve of a binary image

4 Conclusion

By analyzing the experimental results, we can draw the following two conclusions: 1. the generated binarized image can obtain the same high confidence level as the grayscale map, and does not have a lower confidence level than the grayscale map due to the loss of grayscale information. 2. the grayscale map generated by iteration after initializing the population of the grayscale map of the test set has a higher confidence level than the test set, but with the increase of the number of iterations there is a gradual "decomposition" that does not produce an "average picture" of the test set. Here we question the confidence calculation method of most neural networks nowadays, a good confidence function should be single-peaked.

This paper has the following three contributions:

- The MNIST dataset was analyzed from the perspective of Loss value distribution, and the non-uniform exponential sampling method was used to reflect the nature of the neural network to some extent.
- The crossover and variation aspects of the original genetic algorithm were improved to produce the IGA algorithm which not only solved the problem of traditional genetic algorithm relying on the initial population diversity, but also greatly accelerated the convergence of fitness and reduced the time overhead of computation.
- The experiments of Nguyen et al [7] are complemented by experiments on the convergence of the confidence level of the improved genetic algorithm on ANN and DCNN, whether the binarized image can obtain the same confidence level as the grayscale map, and whether the test set as the initial population can obtain a discernible "average portrait".

5 Discussion and Outlook

With the wide application of artificial intelligence and deep learning in the field of computer vision, face recognition has excellent performance in access control systems and payment systems, which require fast response to the input face image, but this has instead become a drawback to be hacked. For face recognition systems without live detection, using the method in this paper only requires output labels and confidence information to obtain high confidence images quickly. In summary, neural networks have many pitfalls due to their uninterpretability and still need to be considered carefully for use in important areas.

If the confidence function is single-peaked, using the method in this paper can quickly generate real input images (recognizable to human eyes) from the label and confidence output of the neural network, which means that the idea of implementing neural networks against attacks by the method in this paper will also fail. The design of a new single-peak confidence calculation function is the focus of future work in this paper.

6 References

- [1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus. Intriguing properties of neural networks[J]. arXiv preprint, 2014, arXiv:1312.6199.
- [2] Naveed Akhtar, Ajmal Mian et al. Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey[J]. arXiv[cs.CV], 2018, 1801.00553.
- [3] A. Kurakin, I. Goodfellow, S. Bengio. Adversarial examples in the physical world[J]. arXiv preprint, 2016, arXiv:1607.02533.
- [4] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, A. Swami. The Limitations of Deep Learning in Adversarial Settings[C]//In Proceedings of IEEE European Symposium on Security and Privacy, 2016.
- [5] J. Su, D. V. Vargas, S. Kouichi. One pixel attack for fooling deep neural networks[J]. arXiv preprint, 2017, arXiv:1710.08864.
- [6] Zhang Jianan, Wang Yixiang, Liu Bo, Chang Xiaolin. Survey of adversarial attacks of deep learning[J]. Cyberspace Security, 2019, 10(07):87-96.
- [7] Nguyen A, Yosinski J, Clune J. Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images[C]//In Computer Vision and Pattern Recognition (CVPR '15), IEEE, 2015.
- [8] D. Erhan, Y. Bengio, A. Courville, and P. Vincent. Visualizing higher-layer features of a deep network[C]//Dept. IRO, Universite de Montr' eal, Tech. Rep ' , 2009.
- [9] McCulloch W S, Pitts W. A logical calculus of the ideas immanent in nervous activity[J]. The bulletin of mathematical biophysics, 1943, 5(4): 115-133.
- [10] Rumelhart D E, Hinton G E, Williams R J. Learning internal representations by error propagation[R]. California Univ San Diego La Jolla Inst for Cognitive Science, 1985.
- [11] LeCun Y, Boser B, Denker J S, et al. Backpropagation applied to handwritten zip code recognition[J]. Neural computation, 1989, 1(4): 541-551.
- [12] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.
- [13] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. Advances in neural information processing systems, 2012, 25: 1097-1105.
- [14] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.
- [15] Komkov S, Petiushko A. Advhat: Real-world adversarial attack on arcface face id system[C]//2020 25th International Conference on Pattern Recognition (ICPR). IEEE, 2021: 819-826.
- [16] Gnanasambandam A, Sherman A M, Chan S H. Optical Adversarial Attack[J]. arXiv preprint arXiv:2108.06247, 2021.
- [17] TANG Jing, PENG Wei-long, TANG Ke-ke, FANG Mei-e. MvUPA: Universal Perturbation Attack against 3D Shape Retrieval based on Multi-view Networks[J/OL]. JOURNAL OF GRAPHICS:1-8[2021-09-22].<http://kns.cnki.net/kcms/detail/10.1034.T.20210901.1425.006.html>.
- [18] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps[J]. arXiv preprint arXiv:1312.6034, 2013.
- [19] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks[J]. arXiv preprint arXiv:1312.6199, 2013.
- [20] Y. LeCun and C. Cortes. The mnist database of handwritten digits[J]. 1998.
- [21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. FeiFei. Imagenet: A large-scale hierarchical image database[C]//In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pages 248–255. IEEE, 2009.
- [22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge[J]. arXiv preprint arXiv:1409.0575, 2014.

-
- [23] Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding[J]. arXiv preprint arXiv:1408.5093, 2014.
- [24] Saito Yasue. Deep Learning from Scratch[M]. Lu Yujie, translated. Beijing: Posts and Telecom Press, 2018. 224-228.
- [25] Ge Jike, Qiu Yuhui, WU Chunming, PU Guolin. Summary of genetic algorithms research[J]. Application Research of Computers, 2008(10):2911-2916.
- [26] WANG Qiong , LV Wei , REN Wei-jian. Immune genetic algorithm and applications in optimization[J]. Application Research of Computers, 2009, 26(12):4428-4431.
- [27] Jiang Zhubo. CNN Introduction: How does the convolution layer extract features[EB/OL]. <https://zhuanlan.zhihu.com/p/31657315>
- [28] Yang Dingming. Using improved genetic algorithm to find the maximum value of multi-peak function[EB/OL]. <https://github.com/huangyebiao/MaxFunVal>