

## Few-Shot Learning with Generative Adversarial Networks Based on WOA13 Data

Xin Li<sup>1,2</sup>, Yanchun Liang<sup>1,2</sup>, Minghao Zhao<sup>1,2</sup>, Chong Wang<sup>1,2,3</sup> and Yu Jiang<sup>1,2,\*</sup>

**Abstract:** In recent years, extreme weather events accompanying the global warming have occurred frequently, which brought significant impact on national economic and social development. The ocean is an important member of the climate system and plays an important role in the occurrence of climate anomalies. With continuous improvement of sensor technology, we use sensors to acquire the ocean data for the study on resource detection and disaster prevention, etc. However, the data acquired by the sensor is not enough to be used directly by researchers, so we use the Generative Adversarial Network (GAN) to enhance the ocean data. We use GAN to process WOA13 dataset and use ResNet to determine if there is a thermocline layer in a sea area. We compare the classification results of the enhanced datasets of different orders of magnitude with the classification results of the original datasets. The experimental result shows that the dataset processed by GAN has a higher accuracy. GAN has a certain enhancement effect to marine data. Gan increased the accuracy of the WOA dataset from 0.91 to 0.93. At the same time, the experimental results also show that too much data cannot continue to enhance the accuracy of WOA in ResNet.

**Keywords:** Generative adversarial networks (GAN), WOA13, few-shot learning, residual network, convolutional neural network.

### 1 Introduction

In recent years, extreme weather events accompanying the global warming have occurred frequently, which brought significant impact on national economic and social development. The ocean is an important member of the climate system and plays an important role in the occurrence of climate anomalies. The climate anomalies in the coastal zone are closely related to the variation of ocean thermal conditions. Existing studies have shown that the SST anomalies in the oceans will have an impact on the Chinese climate. Therefore, deeply understanding the changing characteristics of the

---

<sup>1</sup> College of Computer Science and Technology, Jilin University, Changchun, 130012, China.

<sup>2</sup> Key Laboratory of Symbolic Computation and Knowledge Engineering, Jilin University, Changchun, 130012, China.

<sup>3</sup> Department of Engineering Mechanics, State Marine Technical University of St. Petersburg, St. Petersburg, 190008, Russia.

\* Corresponding Author: Yu Jiang. [jiangyu2011@jlu.edu.cn](mailto:jiangyu2011@jlu.edu.cn).

global ocean and analyzing the data anomalies at the time of disaster generation are an important basis for oceanography and play an important role in climate prediction.

WOA13 data is the latest marine climatology dataset product [Boyer and Mishonov (2014)] of the NOAA National Ocean Data Center Marine Meteorological Laboratory, which includes global ocean temperature, salinity, dissolved oxygen, phosphate, silicate and other marine element data, and is an integrated data product with a variety of measured data [Jiang, Zhao, Hu et al. (2018)]. The data types are divided into annual average data, monthly average data, and seasonal average data; there are three spatial resolutions: 5°, 1°, and 0.25°. The data can provide 3D information on the marine environment [Jie, Liu and Hong (2016)]; the interpolation method is used on depth, i.e. it is divided into 102 layers from the surface layer to a maximum depth of 5,500 m. Compared with the annual data and quarterly data, the monthly data more accurately reflects the overall change in the global ocean [Deng, Zhou, Liu et al. (2016)]. The monthly average data is a complex dataset. We make a grid map of global ocean data. Use ResNet to determine whether there is a thermocline layer in each square. However, ResNet requires a large dataset. Therefore, we adopt a generative adversarial network to increase the samples WOA data.

The structure of the article is as follows: the research and evaluation criteria of WOA13 will be introduced in Section 2. Neural networks, residual networks and Generative Adversarial Networks will be introduced in Section 3. In Section 4, a comparison is made between the enhanced data of the Generative Adversarial Networks and source data residual network. The last part gives the corresponding conclusions.

## **2 Preliminaries**

Currently it is lack of ResNet research on WOA13 data and the current research on WOA data mainly focuses on the distribution and prediction of ocean temperature and salinity. In 2016, Huang et al. [Huang, Lu, Wang et al. (2016)] proposed a model. This model can effectively compensate for the inherent defects of XBT such as lack of measured salinity support and insufficient in detection depth. The difference between the estimation value of full depth sound velocity and the measured value of multiple acoustic profile stations is only -0.2~0.35 m/s. In the same year, Dang et al. [Deng, Zhou, Liu, et al. (2016)] studied the characteristics of the Kuroshio temperature front in the East China Sea, and concluded that WOA13 data has a good effect on the extraction of the temperature front information of the East China Sea Kuroshio main-axis, and the flow core structure in winter and spring is most obvious on the PN section. In 2017, Liu Peng et al. [Liu, Zhang and Liu (2017)] concluded by conducting a corresponding study on the temporal and spatial distribution characteristics of the temperature front in the equatorial Atlantic Ocean that in the sea where there is a northern front, summer and autumn seasons with strong front have different effect on the sound velocity profile from winter and spring seasons with small front, while the southern front has a depth at the front. The change rate is consistent along with the depth in each season. The existing research is based on the original data of WOA13. However, the raw data of WOA13 has problems such as excessive data volume and loud noise data [Yuan and Zhao (2017); Cao, Zhang, Zhang et al. (2018)], which brings more troubles to researchers. This paper uses the Generative

Adversarial Networks to carry out the feature learning for WOA13 data, collation and data enhancement processing.

In this paper, the traditional thermocline discrimination method and the information entropy-based thermocline discrimination method are adopted.

The traditional thermocline method adopts the pure numerical calculation method. Those of the temperature gradient exceeding  $0.2^{\circ}/\text{m}$  for shallow seas less than 200 m and the temperature gradient of exceeding  $0.05^{\circ}/\text{m}$  for shallow seas greater than 200 m are the thermoclines. Specifically, the vertical gradient method will cause the discontinuity between the two critical points of shallow water and deep water. Information entropy: an indicator to measure the purity of the collective samples. This paper combines the ‘information entropy method’ [Jiang, Zhang, Gou et al. (2018)] in machine learning with the traditional method for more precise determination. Suppose that the proportion of the first  $k$  classes in the dataset is  $D$ , and  $p_k$  ( $k = 1, 2, 3, \dots, |y|$ ) is the sample.

$$Ent(D) = -\sum_{k=1}^{|y|} p_k \log_2 p_k \quad (1)$$

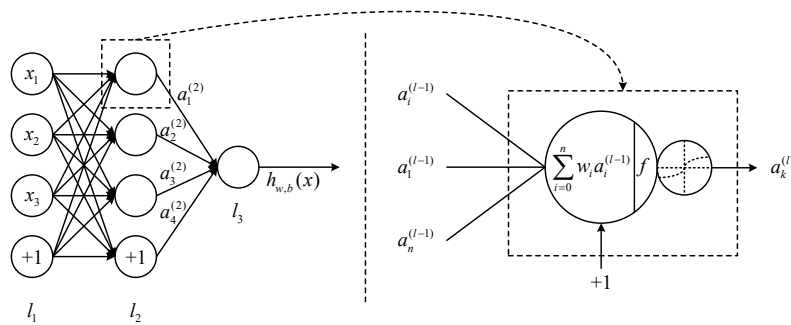
The entropy values increase as the uncertainty of the variables increases.

### **3 Methodology**

ResNet is a deep learning model that can achieve high accuracy in areas such as image recognition. ResNet needs a lot of samples, so we use Convolutional neural network as the core of Generative Adversarial Networks to solve the problem for limited samples.

#### **3.1 Convolutional neural network**

The convolutional neural network is built by simulating the visual perception mechanism of the creature, and can realize the supervised learning and unsupervised learning. The convolution kernel parameter sharing and the sparseness of the inter-layer connection in the hidden layer can make the convolutional neural network study the grid features (such as pixels and audio) with relatively small calculation amount, possessing a stable effect without any additional feature engineering requirements to the data. Structure of the convolutional neural network includes an input layer, a hidden layer and an output layer [Ren, He, Girshick et al. (2017)]. The hidden layer in turn contains a convolutional layer, a pooled layer, and a fully connected layer [Chua and Roska (1993)]. The fully connected layer is a pre-feedback neural network, whose structure is shown in Fig. 1.



**Figure 1:** Model diagram of neural network

A simple three-layer neural network is shown in Fig. 1 (left), each consisting of several neurons. The first layer is the input layer, and the corresponding neuron is also called the input neuron; the second layer is the hidden layer, and the corresponding neuron is called the hidden neuron; the third layer is the output layer, and the corresponding neuron is called the output neuron. Each neuron is connected to all neurons in its previous layer, therefore such a neural network is also called the fully connected feedforward neural network. A more complex neural network is a complex network consisting of more hidden layers and more neurons.

As shown in Fig. 1 (right), it is the structure of the neuron; the neuron is responsible for receiving and summarizing the information from the previous layer to decide whether to activate accordingly. Assuming the activation signal of No.  $k$  neuron in the No.  $l$  layer is  $a_k^l$ , and its calculation process can be divided into following two steps: (1) summarizing the information; assuming that the signal sent from the No.  $n$  neuron of previous layer, i.e., No.  $l-1$  layer is  $\langle a_1^{l-1}, \dots, a_i^{l-1}, \dots, a_n^{l-1} \rangle$ , and the neuron obtains the summary information by summing, i.e.,  $\sum_{i=0}^n w_i a_i^{l-1}$ , where  $a_0^{l-1} = 0$  represents the offset corresponding to the neuron, corresponding symbol  $+1$ ; such representation facilitates the vector operations and matrix operations. (2) The determination is made based on the summary information whether to activate, i.e. the summary information is input into the activation function  $f$ , which determines whether to generate an activation signal to the next layer of the network and calculate the activation signal strength according to the input signal strength. The neural unit is activated only when the signal strength reaches a certain fixed domain value  $\tau$ .

$$a_k^l = f \left( \sum_{i=0}^{l-1} w_i a_i \right) \quad (2)$$

In summary, the operation of one neuron can be induced as Eq. (1); and so forth, the output signals of all neurons in the No.  $l$  layer and the output signals of all neural units in the neural network can be calculated.

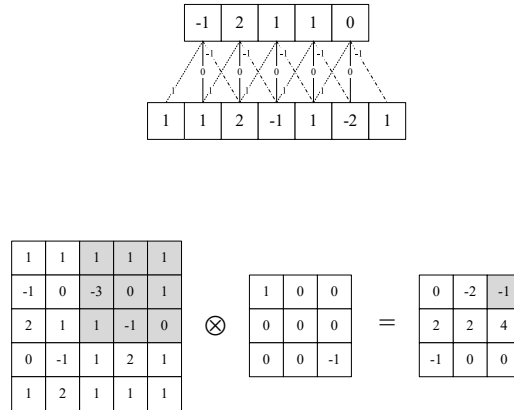
Function of the convolutional layer is to extract the feature from the input data, and contains multiple convolution kernels inside it. Each element composing the convolution kernel corresponds to a weight coefficient and a bias vector, similar to the neuron of a

feed-forward neural network. Each neuron in the convolutional layer is connected with several neurons in areas close to the previous layer, and the size of the area depends on the size of the convolution kernel. The convolution kernel will regularly sweep through the input features during the working, and perform the matrix element multiplication summation for the input feature in the receptive field and superimpose the deviation, as shown in the following formula:

$$Z^{l+1}(i, j) = [Z^l \otimes w^l](i, j) + b = \sum_{k=1}^{K_l} \sum_{x=1}^f \sum_{y=1}^f [Z_k^l(s_0 i + x, s_0 j + y) w_k^{l+1}(x, y)] + b$$

$$(i, j) \in \{0, 1, \dots, L_{l+1}\} \quad L_{l+1} = \frac{L_l + 2p - f}{s_0} + 1 \quad (3)$$

The summation in the above formula is equivalent to solving a cross-correlation.  $b$  is the deviation value,  $Z^l$  and  $Z^{l+1}$  represent the convolution input and output of layer  $l+1$ , also known as the feature map,  $L_{l+1}$  is the size of  $Z_{l+1}$ ; here assuming that the feature map has the same length and width.  $K$  is the number of feature map channels, corresponding to the pixels of the feature map;  $f$ ,  $s_0$  and  $p$  are convolution layer parameters, corresponding to the convolution kernel size, the convolution step size and the number of filling layers. The process of one-dimensional convolution and two-dimensional convolution is shown in Fig. 2.



**Figure 2:** One-dimensional convolution and two-dimensional convolution

For the upper one-dimensional convolution in Fig. 2, the step size 1 and the convolution kernel is 3. For the lower two-dimensional convolution kernel shown in Fig. 2 the step size is 1 and the convolution kernel is  $3 \times 3$ . The convolution layer contains an excitation function helping to express the complex features. For example, the value of each point in the image cannot be negative, and adding the excitation function can effectively eliminate the data outliers. The activation function adopts the following formula form.

$$A_{i,j,k}^l = f(Z_{i,j,k}^l + b) \quad (4)$$

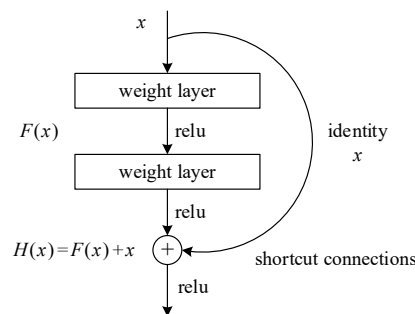
The pooling is required after the valid value is acquired by processing the activation function. After the feature is extracted in the convolutional layer, the output feature map

will be passed to the pooling layer for feature selection and information filtering. The pooling layer contains the preset pooling function whose function is to replace the result of a single point in the feature map with the statistic of the feature map in its neighboring area. The procedure of the pooling layer selecting the pooling area is the same as the convolution kernel scanning the feature map, and is controlled by the pooling size, step size, and filling. The pooling layer sort the  $n*n$  point set into 1 point, effectively scaling the image in  $n$  times. Thus, the features of the image are further extracted.

### 3.2 Residual network

The accuracy of the model will continue to increase along with the increasing of the network layer. When the network layer increases to a certain number, both the training accuracy and the test accuracy will decrease rapidly. This shows that when the network is quite deepened, the deep network becomes more and more difficult in training [Wu, Shen and Hengel (2016)]. Assuming that a relatively shallow network has reached the saturation accuracy, then several constant mapping layers are added following it (i.e.,  $y=x$ , the output is equal to the input); thereby depth of the network will increase. and yet the minimum error will not increase, that is, the deeper network shall not bring an increase in the error of the training set. The idea of directly passing the output of the previous layer to the next one is the source of inspiration for the famous deep residual network.

The residual network borrows the idea of cross-layer linking of the Highway Network [Veit, Wilber and Belongie (2016)], but is improved. Provided that the input of a certain neural network is  $x$ , and the expected output is  $H(x)$ , i.e.,  $H(x)$  is the expected complex potential mapping. The training will be more difficult if such a model will be learnt [Yu, Chen, Dou et al (2017)]; if the accuracy of greater saturation has been learnt (or when the error of the lower layer is found to be larger), the next learning goal will be transformed into learnt the identity mapping, that is, let the input  $x$  approximate to the output  $H(x)$ , for the purpose of ensuring there is no degradation in accuracy of the following layers. The shortcut link method is often used to pass the output value.



**Figure 3:** Shortcut connections

As shown in Fig. 3, the input  $x$  is directly passed to the output as the initial result through the shortcut connection method, and the output result is  $H(x) = F(x) + x$ . When  $F(x) = 0$ ,  $H(x) = x$ , i.e., the identity mapping mentioned above. Therefore, ResNet is equivalent to

changing the learning target, i.e., the target is the difference between the target value  $H(X)$  and  $x$  (i.e., so-called residual  $F(x) := H(x) - x$ ) instead of the complete output. Therefore, the following training goal is to approximate the residual result to 0, so that the accuracy will not decrease as the network deepens.

This kind of residual jump structure breaks the convention that the output of the traditional neural network Layer  $n-1$  can only be passed to Layer  $n$  as an input, therefore the output of a certain layer can be taken as the input of the following certain layer by directly crossing several layers. The meaning is that it provides a new solution for the challenge of increasing the error rate of the whole learning model by superimposing multi layers of network. Therefore, the number of the neural network layers can exceed the previous constraints, reaching dozens, hundreds or even thousands of layers, providing a feasibility for high-level semantic feature extraction and accurate classification. We use the residual network to accurately identify the existence of a thermocline in a marine area.

### 3.3 Basic operation of GAN

The core of GAN consists of two convolutional neural networks [Tang, Tan, Li et al (2017)]. For general calculations of neural networks and convolutional neural networks, please refer to the previous introduction. Here two special operations are introduced, i.e. fractional strided convolution (FSC) [Dosovitskiy, Springenberg and Brox (2015)] and batch normalization [Zeng, Dai, Li et al. (2018)].

#### 3.3.1 Fractional strided convolution

Fractional strided convolution, also known as Transpose Convolution or Deconvolution. Generally the convolutional neural network operates in the order of convolution operation first and then pooling operation; while the fractional strided convolution can be regarded as the inverse process of above operations, i.e., the inverse pooling operation first and then the convolution operation. Let the stride of each dimension be  $s$ , input  $X = [w_x, h_x, c_x]$ , input  $Z = [w_z, h_z, c_z]$ . The process of the inverse pooling operation can be described as first mapping each element in the input matrix as a block  $s * s$ ; the upper left corner of this block stores the corresponding element in the original input and the other positions are complemented by 0, thus,  $w_x * h_x$  blocks of size  $s * s$  can be obtained. These blocks are spliced to obtain the output  $Z$ , which is easy to obtain  $w_z = w_x * s$ ,  $h_z = h_x * s$ , and  $c_z = c_x$  the output scale is expanded by  $s$  times compared to the input scale. The role of pooling is to reduce the dimension, while the role of unpooling is to increase the dimension.

#### 3.3.2 Batch normalization

The batch normalization is a normalization operation for a small batch of samples for each training. The advantage is that it can make the training simpler and more efficient, and can use a larger learning rate as the initial value, so that the model is less sensitive to the initial value. It can be used as a normalized operation to avoid using the dropout and accelerate the convergence speed to some extent. The process is as shown in the following formula:

$$\mu_B = \frac{1}{m} \sum_{i=1}^m x^{(i)} \quad (5)$$

$$\sigma_B^2 = \frac{1}{m} \sum_{i=1}^m (x^{(i)} - \mu_B)^2 \quad (6)$$

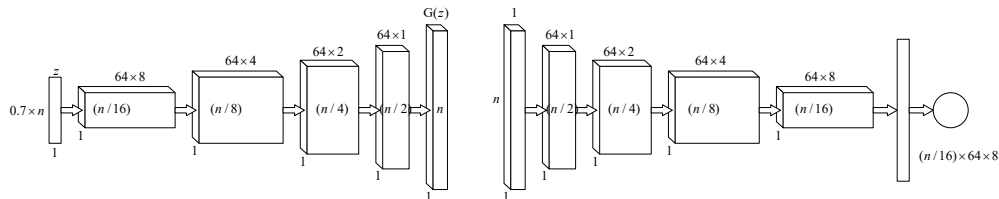
$$x = \frac{x^{(i)} - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} \quad (7)$$

$$z^{(i)} = \gamma x + \beta \quad (8)$$

In the above formula, for each small batch input, the sample mean is calculated by Eq. (6) and the sample standard deviation is calculated by Eq. (7), then the input is normalized by Eq. (8), and finally the return value is calculated by Eq. (9).

### 3.4 Generator and discriminator of GAN

GAN's training process is a game between Generator and Discriminator, and the iterative update is made based on gradient descent and backpropagation algorithms [Yi and Babyn (2018)]. The Generator aims to generate more realistic samples, i.e., to maximize the output probability of the generated samples in the discriminator; the discriminator aims to correctly distinguish the true and false samples, i.e., to maximize the output probability of the real samples while minimize the output probability of generated samples. The Generator and discriminator are shown in Fig. 4.



**Figure 4:** Generator and Discriminator

The structure of GAN's Generator is shown on the left side of Fig. 4. The input is a randomly generated one-dimensional noise vector  $z$  with a length of  $0.7 \cdot n$ , where  $n$  represents the length of the samples in the real dataset (the number of sampling points). First, the input is converted into a 3D matrix by deformation or mapping operation, and then the final output is obtained through a 4-layer fractional convolution operation. The output is a generated sample of length  $n$ . Where each fractional strided convolution operation performs the batch normalization operation. The activation function of the hidden layer is ReLU, and the output layer is a linear unit. The convolution kernel is unified to  $1 \times 5$  scale and the stride is unified to 2. Each fractional strided convolution operation achieves 2 times of the dimension raising. The discriminator of GAN is shown on the right side of Fig. 4. For a sample with an input length of  $n$ , a one-dimensional feature vector is obtained via a 4-layer convolution layer and then a flatten operation.



Finally, Sigmoid is used as an output layer to implement a two-category. The output is the probability that the sample is true. No pooling operation is used in the network, and dimensionality reduction is achieved by the convolution operations without padding operations. All hidden layers use LeakyReLU as the activation function and the batch normalization is performed after the convolution operation. The convolution kernel is unified to 1\*5 scale and the stride is unified to 2. Each convolution operation achieves 2 times of the dimension reduction.

## 4 Experimental setup

### 4.1 Data preprocessing and model training

In this paper the dataset selects WOA13 monthly dataset with a resolution of 0.25°. The experimental platform selects ubuntu 16.04, tensorflow 1.11, and sklearn 0.19. We grid the global data in WOA13. A small square of 2° longitude, 1.25° wide (latitude), and the 0-600m depth is selected, and it contains 8\*5\*40 data points. An 8\*5\*40 3D image is converted to a 40\*40 2D image to perform the Generative Adversarial Networks training and the residual network training. All illegal data in the data set will be directly converted to 0. The land area is also marked as 0. In this paper, the Adam gradient descent algorithm is used. The initial learning rate of both the generator and the discriminator is 0.0002, and the number of training iterations is 2000. The size of source samples is 23940 samples of randomly selected sea areas. The dataset contains a total of 7,815 thermocline areas and 16,125 non-thermocline layer areas. Resnet-50 is selected as Resnet Algorithm. The process of gradient descent algorithm training GAN is shown in Algorithm 1:

---

#### Algorithm 1. GAN training based on a stochastic gradient descent algorithm

---

*Data:* update count of generator  $gk$ , update count of discriminator  $dk$ , iteration count of training epoch  
**for**  $i=0; i < \text{epoch}; i++$  **do**  
  **for**  $j=0; j < dk; j++$  **do**  
    Sub-sample a priori  $\{z^{(1)}, \dots, z^{(m)}\}$  from  $P_g(z)$ ;  
    Sample  $\{x^{(1)}, \dots, x^{(m)}\}$  from real sample  $P_{data}(x)$ ;  
    Update the Discriminator based on the gradient descent algorithm:  
    
$$\nabla_{\theta_d} - \frac{1}{m} \sum_{i=1}^m [\log(D(x^{(i)})) + \log(1 - D(G(z^{(i)})))]$$
  
  **end**  
  **for**  $j=0; j < gk; j++$  **do**  
    Sub-sample a priori  $\{z^{(1)}, \dots, z^{(m)}\}$  from  $P_g(z)$ ;  
    Update the Generator based on the gradient descent algorithm:  
    
$$\nabla_{\theta_g} - \frac{1}{m} \sum_{i=1}^m \log(D(G(z^{(i)})))$$
  
  **end**  
**End**

---

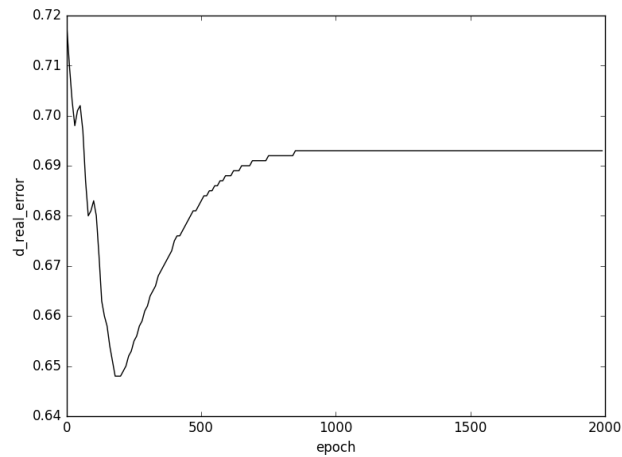
The epoch is the total number of learning cycles in Algorithm 1. The second “for-end” with  $dk$  is the training process of the Discriminator. The third “for-end” with  $gk$  the training process of the Generator. Due to the determination of the existence of the thermocline layer, binary classification cross entropy is selected as the evaluation criteria.

$$\text{loss}(x_i, y_i) = -[y_i \log x_i + (1 - y_i) \log(1 - y_i)] \quad (7)$$

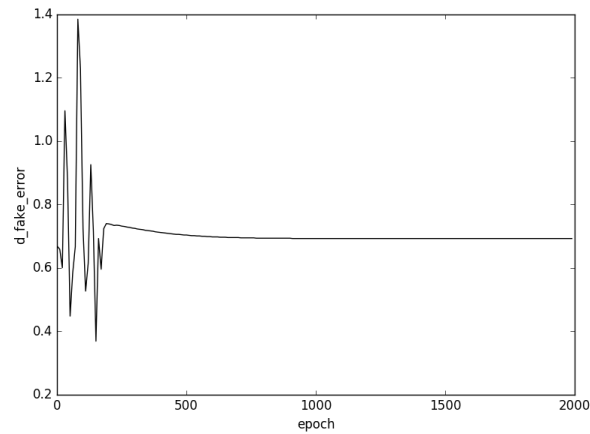
$i$  is only subscript;  $x_i$  represents the probability that the first sample is predicted as a positive example, and  $y_i$  represents the label of the  $i$  sample, and  $w_i$  represents the weight of the item. The lower the loss value, the more accurate the classification result we get.

#### 4.2 Experimental results

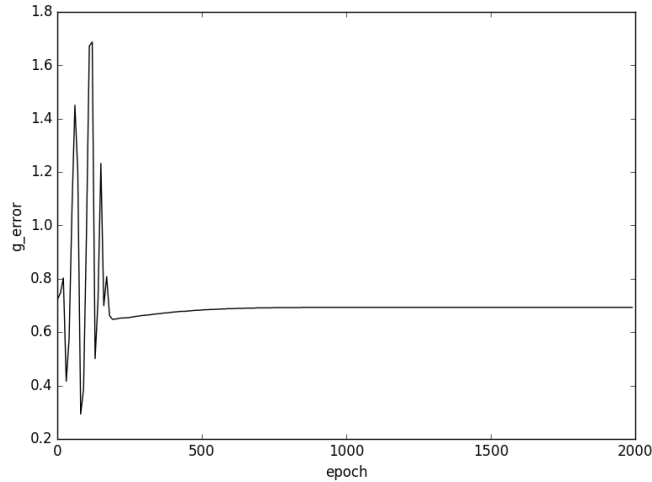
The loss value of the Generator and the Discriminator varies with epoch as shown in Fig. 5, Fig. 6 and Fig. 7.



**Figure 5:** The *loss* of Discriminator in real data



**Figure 6:** The *loss* of Discriminator in real data



**Figure 7:** The Loss of Discriminator in generators

As shown in Fig. 5, the loss of real data is gradually reduced before 250 epochs, indicating that the Generator cannot generate enough “true” data. After 250 epochs, it is explained that the data generated by the Generator is less different from the real data, and the Discriminator is treated. The loss of the Discriminator is stable at 0.693. As shown in Fig. 6, the loss of fake data in the Discriminator is in different fluctuations, after 250 epochs gradually in stability. Loss of fake data of Discriminator stabilized at 0.693. As shown in Fig. 7, the loss value of the Generator is also constantly fluctuating, indicating that the generator is learning the official data and finally stabilizing at 0.693.

We then add the generated data to the original data, test set 1 as 30 percent of the original dataset, and test set 2 as 30 percent of the original dataset with 30 percent of the generated dataset.

**Table 1:** The result of resnet

	source	5000	10000	15000	20000	25000
Acc1	0.9113	0.9175	0.9233	0.9314	0.9371	0.9369
Loss1	0.2081	0.1974	0.1858	0.1740	0.1620	0.1638
Acc2	-	0.9201	0.9289	0.9304	0.9393	0.9454
Loss2	-	0.1978	0.1786	0.1684	0.1503	0.1480

The results are taken 10 times on average as shown in Tab. 1, Acc1 and Loss1 are the accuracy and loss of test set one, and Acc2 and Loss2 represent for the accuracy and loss of test set two. The accuracy of original training set can achieve more than 0.9. With the increase of training set, the accuracy is also increasing. The accuracy and loss of the training set at 20000 and 25000 won’t change significantly. The accuracy of 25000 on test set 2 is higher than 20000, which indicates that there is a difference between the fake samples and the source samples, and that increasing too many generations samples make no effect on the accuracy of whether there is a thermocline.

## 5 Conclusion

With the enhancement of Generative Adversarial Networks, the classification results of ResNet are stronger than the original data results. When accuracy peaked, continued increases in fake data did not continue to increase accuracy. There are some unusual samples in the Woa13 dataset, such as ocean storms, and increasing the sample does not increase the accuracy of such data. In the next work, we will adjust the structure of GAN, while expanding the ResNet-50 to resnet100. We will also explore what kinds of data are incorrectly classified, and whether increasing the sample of these data can improve accuracy.

**Acknowledgement:** The work was supported in part by the National Natural Science Foundation of China (51679105, 61872160, 51809112, 51409117).

## References

- Boyer, T.; Mishonov, A.** (2014): World ocean atlas 2013 product documentation. <http://www.nodc.noaa.gov/OC5/indprod.html>.
- Cao, Z.; Zhang, Y.; Zhang, X.; Liu, Z.** (2018): Analysis on the influence of sea area location data on acoustic waveguide in north Atlantic based on WOA13 data. *Marine Forecasts*, vol. 35, no. 3, pp. 30-40.
- Chua, L. O.; Roska, T.** (1993): The CNN paradigm. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and, Applications*, vol. 40, no. 3, pp. 147-156.
- Deng, Z.; Zhou, L.; Liu, J.; Dong, H.; Niu, H. et al.** (2016): Analysis on the characteristics of the kuroshio temperature front in the east China sea based on WOA13 data. *Journal of Ocean Technology*, vol. 35, no. 4, pp. 26-30.
- Dosovitskiy, A.; Springenberg, J. T.; Brox, T.** (2015): Learning to generate chairs with convolutional neural networks. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1538-1546.
- Huang, C.; Lu, X.; Wang, K.; Shen, J.; Zhang, B. et al.** (2016): A method of exactly determining the sound velocity in deep water based on salt information from WOA13 model and XBT data. *Marine Science Bulletin*, vol. 35, no. 5, pp. 554-561.
- Jiang, Y.; Zhang, T.; Gou, Y.; He, L.; Bai, H. et al.** (2018): High-resolution temperature and salinity model analysis using support vector regression. *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-9.
- Jiang, Y.; Zhao, M.; Hu, C.; He, L.; Bai, H. et al.** (2018): A parallel FP-growth algorithm on world ocean atlas data with multi-core CPU. *Journal of Supercomputing*, pp. 1-14.
- Jie, Q.; Liu, J.; Hong, Y.** (2016): Study on the seasonal variation characteristics of the sub-antarctic front based on WOA13 data. *Journal of Ocean Technology*, vol. 35, no. 2, pp. 83-86.
- Liu, P.; Zhang, Y.; Liu, J.** (2017): Analysis on temporal and spatial distribution characteristics of temperature front in the equatorial Atlantic based on WOA13 data. *Marine Forecasts*, vol. 34, no. 5, pp. 47-57.

**Ren, S.; He, K.; Girshick, R.; Sun, J.** (2017). Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 39, no. 6, pp. 1137-1149.

**Tang, W.; Tan, S.; Li, B.; Huang, J.** (2017). Automatic steganographic distortion learning using a generative adversarial network. *IEEE Signal Processing Letters*, vol. 24, no. 10, pp. 1547-1551.

**Veit, A.; Wilber, M.; Belongie, S.** (2016): Residual networks behave like ensembles of relatively shallow networks. *Advances in Neural Information Processing Systems*, pp. 550-558.

**Wu, Z.; Shen, C.; Hengel, A.** (2019): Wider or deeper: revisiting the ResNet model for visual recognition. *Pattern Recognition*, pp. 1-36.

**Yi, X.; Babyn, P.** (2018). Sharpness-aware low-dose ct denoising using conditional generative adversarial network. *Journal of Digital Imaging*, vol. 31, no. 5, pp. 655-669.

**Yu, L.; Chen, H.; Dou, Q.; Qin, J.; Heng, P. A.** (2017). Automated melanoma recognition in dermoscopy images via very deep residual networks. *IEEE Transactions on Medical Imaging*, vol. 36, no. 4, pp. 994-1004.

**Yuan, P.; Zhao, S.** (2017): Synthesis of non-uniform linear array antenna based on WOA. *Radio Engineering*, vol. 47, no. 10, pp. 53-58.

**Zeng, D.; Dai, Y.; Li, F.; Sherratt, R.; Wang, J.** (2018): Adversarial learning for distant supervised relation extraction. *Computers, Materials & Continua*, vol. 55, no. 1, pp. 121-136.