A Weight Initialization Method Based on Neural Network with Asymmetric Activation Function

***Abstract*—Weight initialization of neural networks has an important influence on the learning process, and the selection of initial weights is related to the activation interval of the activation function. Based on the linear interval tolerance method (LIT), the weight initialization problem is transformed into a linear interval tolerance problem, and an improved weight initialization method for asymmetric activation function is first proposed in this paper. Then, a tolerance solution theorem based on neural network system is given and proved. Furthermore, the algorithm to determine the initial weight interval is given. The validity of the theorem and algorithm is verified by numerical experiments. The method in this paper expands the selection range of neural network activation function and improve the convergence rate, the research has certain theoretical significance and practical application value.**

Keywords- Weight initialization, Asymmetric activation function, Interval linear system, Tolerate solution, Neural network

1 INTRODUCTION

The selection of activation function is very important in neural network calculation. Different activation functions have different behaviors, which lead to different learning performance. Activation functions are mainly divided into S-series functions (i.e. Sigmoid function, Tanh function, etc.) and R-series functions (i.e. ReLU, ELU, p-ReLU and L-ReLU functions, etc.). S-series functions are differentiable and their derivatives can be expressed by the functions themselves, but the existence of saturated region will slow down the gradient of back propagation or cause the gradient to disappear, which will affect the convergence speed of the network. R-series functions are characterized by convenient derivation, fast convergence speed and no saturation region. However, the neuron will die and the weight will stop updating when the input data is negative.

Another important issue in neural network computation is the choice of initial weights. The initial value of the parameter has a great influence on the training process. For S-series activation functions, parameters should be initialized in the activation domain of the function as far as possible. Weight initialization methods of neural networks have been discussed and analyzed in many literatures. As early as in the 1990s, some scholars had proposed: To keep the input from falling into the saturated domain of the Sigmoid function, the total input needs to fall between -3 and 3 if there are multiple inputs, and the weight should be selected within a small range, the weight initialization interval is determined by the number of input nodes (Boers E,1992). By analyzing the distribution of the output values of hidden layer nodes, Nguyen D proposed the setting formula of the initial weight which indicated that if the output of each hidden layer node is regarded as a piecewise linear approximation of the objective function, each hidden node should share an active output interval (Nguyen D, 1990). In addition, in order to prevent network overfitting and improve generalization ability, the influence of initial weights on the overall training process and the phenomenon of weight decline was analyzed in the multi-classification problem(Kim M C, 1997). Based upon the dynamic decision boundary, Kim, Y.K. proposed the minimum bound of the weight based on the generalized delta (Kim, Y.K.,1991), this method does not determine the determined range of the weight selection, but only gives the lower bound of the weight. By means of statistical analysis, Drago,G.P. proposed that the scaling factor related to weight is a function of the percentage of dead neurons(PNP), and a weight initialization method based on PNP is given(Drago,G.P.,1992).

Some common initialization methods (i.e. random initialization method, Kim-Ra method (Kim, Y.K, 1991), SCAWI method (Drago, G. P.,1992), Li method (Li, G, 1993), threshold initialization method (Palubinskas, G, 1994), Shimodaira method (Shimodaira, H, 1994) and Yoon method (Ho-Sub Yoon, 1995)) have been compared by measuring convergence speed, generalization ability and convergence success rate(Mercedes Fernández-redondo, 2001), it has shown that the best scheme based on BP algorithm is Shimodaira method, but its disadvantage is that it needs to determine the experimental parameters through repeated tests.

Xavier et al. proposed the weight initialization method commonly used in deep learning in recent years (Bengio Y, 2010), which set the weight of each network layer to the value selected from the bounded random uniform distribution. This method is based on the idea that the activation value of each layer should have an appropriate width, and the variance of activation and back propagation gradient should be maintained. So that the network has a faster convergence speed. This method is based upon the premise that the activation function is linear (because the Sigmoid function and the Tanh function are symmetric, they can be regarded as linear functions near the center),

so the initial Xavier value is suitable for the neural network using the Sigmoid series of functions. And for the ReLU activation function, Kaiming He proposed the Kaiming initialization method (Kaiming He,2016), this method aims to make the activation output of each layer approximate the standard normal distribution, prevent the activation output from being unbounded or disappearing, and improve the convergence of the network. It is widely used in deep learning, especially in convolutional neural networks. At present, the mainstream weight initialization methods are still Xavier method based on Sigmoid series of functions and Kaiming method based on ReLU series of functions. In addition, there are also many other specific initialization methods. For example, the weight initialization method is given in combination with physical related characteristics (Wessels L F A, 1992). A general weight allocation method for neural network was proposed in the parameter setting of the controller in Literature (Jiang N, 2018), and a weight initialization method based on Cauchy inequality and linear algebra was proposed in Literature (Yam J Y F, 2000).

S.P.Adam combined weight initialization with tolerance solution of linear interval system in 2014, (S.P.Adam, 2014). He expressed the determination of initial weight of neural network as Linear Interval Tolerance (LIT) problem, which well-integrated interval calculation and neural network weight initialization. Based upon the theory of solving linear interval system, the initialization interval of weights is determined. It only needs to calculate the basic statistical information of the input data, and does not need to do complex preprocessing. It can get the initialization interval of the weight with small calculation cost. At present, it is still rare to study the weight initialization problem with this kind of method. The interval calculation theory used in this paper like the theory and application of interval analysis, the tolerance problem of linear interval and the boundary problem of set was also put forward very early (Hansen E R, 1992, Alefeld G, 2000, Beaumont O, 2001). Some scholars have also applied interval calculation method in the learning process of neural network, and achieved good results (Guan S, 2020, Jinwu Z, 2011).

The LIT method is applicable to the saturated functions（Sigmoid series）, whose unsaturated domain is symmetric. This paper mainly focuses on three situations: First, how to select the initial interval of weight when the activation domain is no longer symmetric; Second, how to improve the initialization method of LIT when the activation domain is still symmetric but the bias is not zero; Third, how to make the input of activation functions located in any given real number interval to avoid neuron necrosis when using unsaturated activation functions. Based upon the above situations, an improved LIT method is proposed and the selection range of weights is given in this paper.

The sections of this paper are organized as follows. Section 2 presents the theory of interval linear systems. Section 3 is devoted to present the transformation of the weight initialization problem and the improvements to the LIT method. The NLIT algorithm is proposed in section 4. Then, numerical simulation results are demonstrated to validate the NLIT method in section 5. Section 6 summarizes this paper with some concluding remarks.

Symbol description: Boldface lower case（such as $\mathbf{x}$, $\mathbf{y}$) stand for the real interval. *IR* stands for the whole real interval set, IR^n stands for the interval vector of n dimension, $IR^{m\times n}$ stands for the interval matrix of $m\times n$ dimension.

2 LINEAR INTERVAL

Interval, namely $\mathbf{I}=[a,b]\subset R$, $a\le b$. it is said to be symmetric if $a=-b$. Also, a point is an interval when $a=b$.

2.1 Arithmetic of the linear interval

Consider two intervals $\mathbf{x}=[x_1,x_2]$, $\mathbf{y}=[y_1,y_2]$, where $x_1,x_2,y_1,y_2\in R$, the operating rules are formulated by

$$\mathbf{x}+\mathbf{y}=[x_1+y_1,x_2+y_2] \tag{1}$$

$$\mathbf{x}-\mathbf{y}=[x_1-y_2,x_2-y_1] \tag{2}$$

$$\mathbf{x}\cdot\mathbf{y}=\left\{\begin{array}{l}\min(x_1y_1,x_1y_2,x_2y_1,x_2y_2),\\ \max(x_1y_1,x_1y_2,x_2y_1,x_2y_2)\end{array}\right\} \tag{3}$$

$$\mathbf{x}/\mathbf{y}=\mathbf{x}\cdot\frac{1}{\mathbf{y}},0\notin[y_1,y_2] \tag{4}$$

2.2 Interval linear systems

Consider the interval linear system $\mathbf{A}x=\mathbf{b}$, where $\mathbf{A}=[A_1,A_2]\in IR^{m\times n}$ stands for an interval matrix in $m\times n$ dimensions, $\mathbf{b}\in IR^m=[b_1,b_2]$ stands for an n −dimensions interval vector . The solution of

$\mathbf{A}x=\mathbf{b}$ is defined as

$$\sum(\mathbf{A},\mathbf{b})=\{x\in IR^n \mid \tilde{A}x=\tilde{b},\forall \tilde{A}\in\mathbf{A},\forall \tilde{b}\in\mathbf{b}\} \tag{5}$$

where $\tilde{A}\in R^{m\times n}$ and $\tilde{b}\in R^n$ stands for real matrices and a real vector respectively after the interval matrix element is assigned to a given interval.

2.3 Tolerance problem and the solution set

Consider the interval equation $F(\mathbf{a},x)=\mathbf{b}$, where $x\in R^n$, $\mathbf{a}$ and $\mathbf{b}$ stand for real intervals, the tolerance solution is defined as

$$\forall \tilde{a}\in\mathbf{a}, F(\tilde{a},x)\in\mathbf{b}. \tag{6}$$

For the linear interval equation $\mathbf{A}x=\mathbf{b}$, the tolerance solution is defined as

$$\sum\nolimits_{\forall\exists}(\mathbf{A},\mathbf{b})=\{x\in R^n \mid \forall \tilde{A}\in\mathbf{A},\exists\tilde{b}\in\mathbf{b},\tilde{A}x=\tilde{b}\} \tag{7}$$

or

$$\sum\nolimits_{\subseteq}(\mathbf{A},\mathbf{b})=\{x\in R^n \mid \mathbf{A}x\subseteq\mathbf{b}\} \tag{8}$$

3 THE WEIGHT INITIALIZATION OF NEURAL NETWORK

3.1 Transformation of the Problem

Consider a multilayer perceptron with a single hidden layer. N, H and O stand for the number of neurons in the input layer, hidden layer and output layer respectively. In this section, symmetric activation interval and asymmetric activation interval are discussed respectively.

3.1.1 LIT method[16]

For symmetric activation interval $[-a,a]$, selecting its input in the activation domain means

$$-a\le\sum_i w_{ji}x_i+w_{jb}\le a. \tag{9}$$

where j stands for the j th node of the hidden layer, x_i stands for the i-dimensional input, w_{ji} stands for the connection weight between the i th node in the input layer and the j th node in the hidden layer, and w_{jb} stands for the threshold value. For input data with P patterns, request

$$-a\le\sum_{i=1}^{N} w_{ji}x_i^k+w_{jb}\le a,\quad k=1,2,\cdots,P \tag{10}$$

where x_i^k stands for the *i*-dimensional input data of the k th input, and N stands for the dimension of the input data.

Based upon equation (10), the weight initialization method is to define an interval for the selection of initial weight, if the unknown w_{ji} and w_{jb} are real number in $[-w_{ji},w_{ji}]$ $(1\le i\le N)$ and $[-w_{jb},w_{jb}]$ respectively, then, the equation (9) is

$$\sum_i[-w_{ji},w_{ji}]x_i+[-w_{jb},w_{jb}]\subseteq[-a,a] \tag{11}$$

According to the discussion in section 2.3, equation (11) shows that w_{ji} is the tolerance solution of the following interval linear equation.

$$\sum_i[-w_{ji},w_{ji}]x_i+[-w_{jb},w_{jb}]=[-a,a] \tag{12}$$

The following conclusions has proposed in literature [16]:

a. If the bias is equal to zero, the public initialization interval for all weights between the input layer and the hidden layer should be

$$w^*=\frac{1}{\sum_{i=1}^{N}(\max_{k=1}^{P}\{x_i^k\})}\cdot[-a,a] \tag{13}$$

that is, it satisfies $-a \le \sum_{i=1}^{N} w_{ji} x_i \le a$,where $w_{ji} \in w^*$.

b. If the bias is equal to zero, the public initialization interval for all weights include bias between the input layer and the hidden layer should be $w^{**} = w^* / s_{x_i}$, where s_{x_i} is a statistic providing summary information about the *i*-th input data component x_i such as the third quartile of the input data.

3.1.2 Improvements to the LIT method

First, consider the saturation activation function (i.e. Sigmoid series function), the inequality $-a \le \mathbf{wx} + \mathbf{b} \le a$ should hold if bias is considered based upon the conclusion a in section 3.1.1. If the parameter $\mathbf{b}$ is randomly initialized and the upper and lower bounds of $\mathbf{b}$ are known (for example, uniformly distributed random numbers are selected within [0,1]), then the target is equivalent to solving the linear interval system $a_1 \le \mathbf{wx} \le a_2$. Currently, the interval $[a_1, a_2]$ is no longer symmetric, and the original conclusion is no longer applicable.

Second, based upon the saturated activation function, if the activation domain have a greater change, A simple case is when the Sigmoid function shifts to the left or right (The reason for considering this situation is that most of the functions currently used in more complex deep learning are non-saturated functions , from the point of view of the mean value, the S-series function is more similar to the ReLU function when shifted to the right, because the mean of the ReLU function is not obtained at the origin). In this case, the interval is not symmetric about the origin, and the degree of asymmetry depends on the magnitude of the offset.

3.1.3 NLIT method

Consider an asymmetric interval $[a_1, a_2]$, the following inequation should be hold,

$$a_1 \le \sum_{i=1}^{N} w_{ji} x_i^k \le a_2, \quad k = 1, 2, \cdots, \mathrm{P} \tag{14}$$

based upon inequation (14), the weight initialization method is to select an interval for the initial weight, if the unknown w_{ji} is a real number in $[-w_{ji}, w_{ji}]$ $(1 \le i \le N)$, then w_{ji} is the tolerant solution of the following interval linear equation

$$\sum_i [-w_{ji}, w_{ji}] \cdot x_i = [a_1, a_2]. \tag{15}$$

3.2 Theorem of tolerance solution based on NLIT method

Theorem. Consider the interval linear system $\sum_{i=1}^{N} \mathbf{w}_{ji} x_i^k = [a_1, a_2]$,where $0 \le x_i^k \le 1$, $k = 1, 2, \cdots, \mathrm{P}$, set $\mathbf{w}^* = (1 / \sum_{i=1}^{N} E(X_i)) \cdot [a_1, a_2]$, (where X_i stands for the random variable corresponding to column *i* of the coefficient matrix of the system, and $E(X_i)$ stands for the mathematical expectation of the random variable X_i), then, $\sum_{i=1}^{N} \mathbf{w}^* x_i^k$ converges to $[a_1, a_2]$ when N is large enough. In other words, the tolerance solution of the system is

$$\mathbf{w}^* = (1 / \sum_{i=1}^{N} E(X_i)) \cdot [a_1, a_2] \tag{16}$$

in the probability sense.

Proof. The coefficient matrix of the interval system $\sum_{i=1}^{N} \mathbf{w}_{ji} x_i^k = [a_1, a_2]$ is

$$\mathbf{X}=\begin{pmatrix} x_1^1 & x_2^1 & \cdots & x_N^1 \\ x_1^2 & x_2^2 & \cdots & x_N^2 \\ \vdots & \vdots & & \vdots \\ x_1^{\mathbf{P}} & x_2^{\mathbf{P}} & \cdots & x_N^{\mathbf{P}} \end{pmatrix} \tag{17}$$

each column of $\mathbf{X}$ is corresponding to the sequence of random variables $X_1,\cdots,X_N$, and $X_i(i=1,2,\cdots,N)$ is independent of each other. According to Chebyshev's Law of Large Numbers, $(\sum_{i=1}^{N} x_i^k)/N$ is densely clustered around its mathematical expectation $\sum_{i=1}^{N} E(X_i)/N$ when N is large enough, namely,

$$\lim_{N\to\infty} P\{|\sum_{i=1}^{N} x_i^k/N-\sum_{i=1}^{N} E(X_i)/N|<\varepsilon\}=1 \tag{18}$$

thus,

$$\sum_{i=1}^{N}\mathbf{w}^* x_i^k=\mathbf{w}^*\sum_{i=1}^{N} x_i^k=(1/(\sum_{i=1}^{N} E(X_i)))\cdot[a_1,a_2]\cdot\sum_{i=1}^{N} x_i^k=(\sum_{i=1}^{N} x_i^k/\sum_{i=1}^{N} E(X_i))\cdot[a_1,a_2] \tag{19}$$

also

$$\lim_{N\to\infty} P\{|\sum_{i=1}^{N} x_i^k/\sum_{i=1}^{N} E(X_i)-1|<\varepsilon\}=1,\cdot \tag{20}$$

thus, $\sum_{i=1}^{N}\mathbf{w}^* x_i^k$ converges to $[a_1,a_2]$ in probability, that is, $\mathbf{w}^*=(1/\sum_{i=1}^{N} E(X_i))\cdot[a_1,a_2]$ is the tolerant solution of the interval linear system in probability sense.

In practical application, the sample mean $(\sum_{k=1}^{P} x_i^k)/P$ is the unbiased estimate of $E(X_i)$, and $E(X_i)$ is substituted by $(\sum_{k=1}^{P} x_i^k)/P$, namely

$$\mathbf{w}^*=(1/\sum_{i=1}^{N}((\sum_{k=1}^{P} x_i^k)/P))\cdot[a_1,a_2]\ . \tag{21}$$

4 NLIT ALGORITHM

Step1. Initialize the parameters of the network;
Step2. Obtain the input data;
Step3. Scale the input data to the interval $[0,1]$;
Step4. Give the upper and lower bounds a_1 and a_2 according to the specific activation function;
Step5. Calculate the initial weight between the input layer and the hidden layer：

$$\mathbf{w}^*=\frac{1}{\sum_{i=1}^{N}((\sum_{k=1}^{P} x_i^k)/P)}\cdot[a_1,a_2] \tag{22}$$

Step6. Calculate the initial weight between the hidden layer and the output layer, randomly select the normal distribution in $[-3/\sqrt{N},3/\sqrt{N}]$ [1];
Step7. Output the initial weights;
Step8. End.

5 NUMERICAL EXPERIMENT

To illustrate the effections of the proposed NLIT algorithm, four groups numerical example are demonstrated in this section. Mnist and Poker-Hand data sets in Experiment 1 and Experiment 2 are from the UCI machine learning database; The data of Experiment 3 is from Literature [22] and the data of Experiment 4 is from the field test of the intelligent end cover production line in Liaoning Key Laboratory of Information Physics Fusion and Intelligent Manufacturing for CNC Machine.

Parameter Settings are shown in Table 1. Four activation functions are used in the experiment, including Sigmoid function, Sigmoid function shifted 5 units to the right, Sigmoid function shifted 7 units to the right and

ReLU function. (In general, the larger the offset of Sigmoid function, the faster the network convergence, and the more obvious the effect of NLIT method. This paper mainly uses the Sigmoid function with the right offset of 5 and 7 units for the experiment, and the left offset can also be used.) The activation regions of the above four functions are $[-4,4]$, $[3,11]$, $[1,9]$ and $[1,5]$ respectively. The experimental results of NLIT method will be compared with Boersk method [1], Xavier method [11], random number method following U(0,1), and NW method [2].

TABLE I. NETWORK PARAMETER SETTING TABLE

attribute	Value			
	(Experiment 1)	(Experiment 2)	(Experiment 3)	(Experiment 4)
Layers	3	3	3	3
Number of input layer nodes	784	10	3	4
Number of hidden layer neurons	50	50	5	10
Number of nodes in the output layer	10	10	2	1
Number of training samples	60000	25000	20	30
Number of test samples	10000	13118	--	--
Mini-batch size	100	100	--	--
learning rate	0.1	0.1	0.3	0.1
maximum iterations	10000	10000	20	400

5.1 Numerical example 1 (Recognition of MNIST data set)

The Mnist data set consists of images of ten numbers from 0-9, a total of 70,000 images. There are a total of 784 input attributes and 1 target attribute after expanding the two-dimensional image matrix into one-dimensional data. Use 1-10 to stand for 10 different categories. The mini-batch method is used to train the network, and the network parameter settings are shown in Table 1. Compare the test results of different initialization methods when the activation function is changed, the results are shown in Table 2.

TABLE II. COMPARISON OF CLASSIFICATION ACCURACY OF TEST SAMPLES UNDER DIFFERENT INITIALIZATION METHODS

Initialization method	Accuracy rate			
	Sigmoid ($b \in [0,1]$)	Sigmoid offset 5 to the right	Sigmoid offset 7 to the right	Relu
NLIT	94.20% (5)	93.79% (6)	93.02% (7)	96.72% (2)
Boersk	94.46% (3)	93.11% (6)	88.35%	96.91% (2)
Xavier	94.62% (3)	92.91% (7)	87.44%	96.99% (2)
Rand~ U(0,1)	94.57% (3)	86.27%	11.35%	96.90% (2)
NW	11.35%	11.35%	11.35%	19.92%

In Table 2, the numbers in brackets stand for the number of epochs passed through when the identification accuracy of test samples reached more than 90%, and the accuracy without marking was all lower than 90%. It has been shown that the larger the offset of Sigmoid function, the higher the accuracy of NLIT method, NLIT has obvious advantages in learning and convergence speed. However, the NLIT method does not have obvious advantages when the non-offset Sigmoid function is biased or the ReLU function is used as the activation function. It is because the NLIT method could only ensure that its input falls in any given positive interval for the Relu function, which avoids necrosis of neurons. Compared with the conventional method, its utility is limited. For the case of using bias, the effect on network training is not obvious because the value range of bias is small. The NLIT method finally converges with probability, and it cannot guarantee that all inputs fall in the activation domain. But in these two cases, NLIT has little error compared with the test results of other methods, which is within an acceptable range. It has been shown that the stronger the asymmetry of the sigmoid function, the more obvious the advantage of the NLIT method.

5.2 Numerical example 2 (Recognition of Poker-hand data set)

Take 38118 groups of data in the Poker-Hand data set, which have 10 training attributes and 1 target attribute, and 10 different categories are represented by 10 numbers from 0 to 9. The Mini-Batch method was used to train the network, and the test results of different initialization methods were compared when the activation function was changed, as shown in Table 3.

TABLE III. COMPARISON OF CLASSIFICATION ACCURACY OF TEST SAMPLES UNDER DIFFERENT INITIALIZATION METHODS

Initialization method	Accuracy rate			
	Sigmoid ($b \in [0,1]$)	Sigmoid offset 5 to the right	Sigmoid offset 7 to the right	Relu
NLIT	55.18%	56.66%	56.48%	57.94%

Boersk	54.81%	55.67%	50.60%	59.02%
Xavier	54.34%	50.60%	50.60%	58.27%
Rand~ U(0,1)	50.60%	50.60%	50.60%	54.52%
NW	51.86%	56.63%	55.63%	57.49%

It has been seen that the recognition accuracy of several methods is above 50%, and NLIT performed better overall.

5.3 Numerical example 3(Passenger and freight traffic problems of car companies)

Take the passenger traffic statistics of a certain automobile company from 1990 to 2009. There are 20 sets of samples. The input data has 3 attributes and the output data has 2 attributes. Due to the limited number of data samples, all samples are used Train and test the network in order to reflect the network training process. Compare the error between the network output value and the true output value. Table 4 shows the training effect comparison of each initialization method when using different activation functions.

TABLE IV. COMPARISON OF MEAN SQUARE ERRORS OF TRAINING SAMPLES UNDER DIFFERENT INITIALIZATION METHODS

Initialization method	MSE			
	Sigmoid ($b \in [0,1]$)	Sigmoid offset 5 to the right	Sigmoid offset 7 to the right	Relu
NLIT	4.62e-07	1.24e-04	6.24e-05	6.03e-05
Boersk	5.58e-06	7.89e-06	6.41e-06	1.50e-04
Xavier	6.53e-06	6.37e-06	6.35e-06	1.38e-05
Rand~ U(0,1)	1.49e-10	6.34e-06	6.36e-06	5.58e-06
NW	1.60e-05	5.82e-06	6.34e-06	1.38e-03

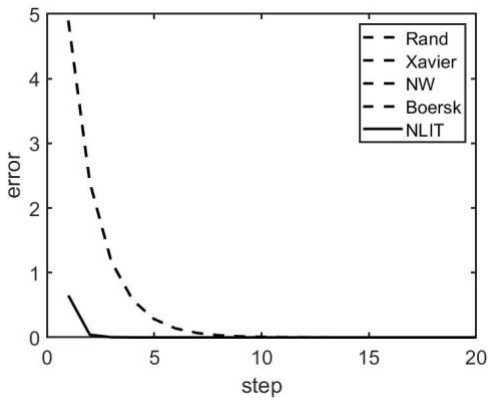

Fig.1 Comparison of convergence rates of different methods (the Sigmoid function is offset 7 units to the right)

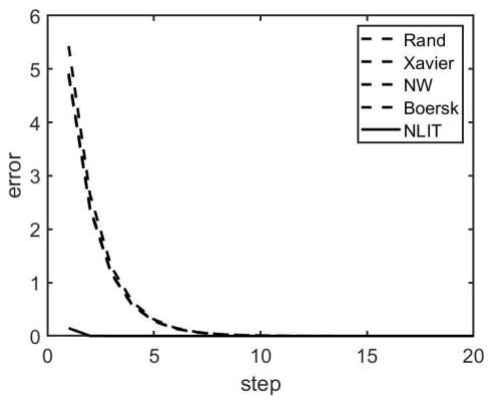

Fig.2 Comparison of convergence rates of different methods (the Sigmoid function is offset 5 units to the right)

It has been shown in Table 4 that the errors are all acceptable after the network training under the five initialization methods. Fig. 1 and Fig. 2 show the comparison of network convergence speed when the Sigmoid function is offset 7 units and 5 units to the right. The convergence effect of NLIT method is shown in the line. Due to the closeness of the values, the convergence curves of the other four methods overlap, as shown by the dotted line. It has been shown that the NLIT method converges rapidly from the second step and has obvious advantages.

5.4 Numerical example 4(Cutting energy consumption of end cover production line equipment)

End cover is a back cover installed in the motor or other chassis, widely used in machine tool headstock and motor reducer, the production line consists of two parts: processing area and measuring area. The autonomous perception of equipment processing parameters is an important issue of machine autonomous intelligence. The main processing parameters of machine tools are spindle speed, feed rate, cutting depth and cutting length, etc. Taking the cutting energy consumption in the process of the end cover production line as the object of investigation, the current changes of the machine tool host reflect the changes in cutting energy consumption. Studying the changing law of the current with respect to the above four parameters has certain significance for the equipment's autonomous perception and monitoring of energy consumption information. In the field experiment, 30 sets of data corresponding to the cutting current measured in the end cover machining process with the change of spindle speed, feed rate, cutting depth and cutting length are shown in Table 5.

TABLE V. THE CUTTING PROCESS PARAMETERS OF THE END COVER

Seq.	Spindle rotation speed(r/s)	Feed rate (mm/r)	Cutting depth (mm)	Cutting length (mm)	Electricity (A)	Seq.	Spindle rotation speed(r/s)	Feed rate (mm/r)	Cutting depth (mm)	Cutting length (mm)	Electricity (A)

1	800	0.12	0.4	10	2.3	16	1400	0.12	0.9	10	4.5
2	800	0.12	0.4	14	2.3	17	500	0.185	0.65	12	2.2
3	800	0.12	0.9	14	2.3	18	1700	0.185	0.65	12	5.3
4	800	0.25	0.9	10	2.3	19	1100	0.055	0.65	12	3.3
5	800	0.25	0.9	14	2.3	20	1100	0.315	0.65	12	3.7
6	800	0.25	0.4	14	2.4	21	1100	0.185	0.15	12	2.4
7	800	0.25	0.4	10	2.6	22	1100	0.185	1.15	12	6.5
8	800	0.12	0.9	10	3.2	23	1100	0.185	0.65	8	4.5
9	1400	0.12	0.4	10	2.3	24	1100	0.185	0.65	16	4.5
10	1400	0.12	0.4	14	3.8	25	1100	0.185	0.65	12	4.5
11	1400	0.12	0.9	14	3.8	26	1100	0.185	0.65	12	4.5
12	1400	0.25	0.9	10	4.9	27	1100	0.185	0.65	12	4.5
13	1400	0.25	0.9	14	4.9	28	1100	0.185	0.65	12	4.5
14	1400	0.25	0.4	14	3.1	29	1100	0.185	0.65	12	4.5
15	1400	0.25	0.4	10	3.1	30	1100	0.185	0.65	12	4.5

There are 30 sets of samples in Table 5, and the input data has 4 attributes (spindle speed, feed rate, cutting depth and cutting length), and the output data has 1 attribute (cutting current). The number of samples is limited, so all the samples are still used to train and test the network. Table 6 shows the comparison of the training effects of each initialization method when the network uses different activation functions.

TABLE VI. COMPARISON OF MEAN SQUARE ERRORS OF TRAINING SAMPLES UNDER DIFFERENT INITIALIZATION METHODS

Initialization method	MSE			
	Sigmoid ($b \in [0,1]$)	Sigmoid offset 5 to the right	Sigmoid offset 7 to the right	Relu
NLIT	5.86e-07	1.39e-04	2.29e-04	1.18e-29
Boersk	5.64e-05	2.39e-08	3.07e-11	1.30e-04
Xavier	8.42e-08	4.69e-10	2.55e-12	1.40e-04
Rand~ U(0,1)	4.92e-10	5.40e-17	2.82e-19	4.06e-11
NW	1.61e-05	1.79e-08	2.16e-13	3.68e-07

As can be seen from Table 6, the convergence speed advantage of NLIT is not obvious when Rule is used as activation function. It is determined by the characteristics of ReLU function, and the final network error is acceptable. The convergence rate of NLIT is advantageous when other activation functions are used. NLIT achieved the effect of 28 iterations of other methods after only 2 steps of training after shifting the Sigmoid function by 7 units to the right.

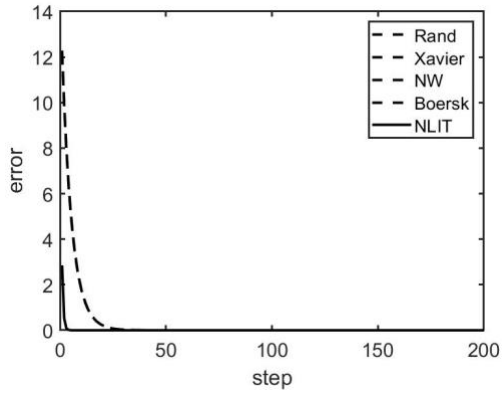

Fig.3 Comparison of convergence rates of different methods (the Sigmoid function is offset 7 units to the right)

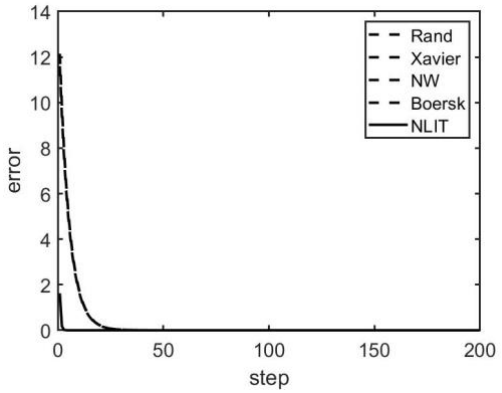

Fig.4 Comparison of convergence rates of different methods (the Sigmoid function is offset 5 units to the right)

In order to further illustrate the convergence speed advantage of NLIT method, Fig. 3-4 shows the network convergence speed comparison diagram when the Sigmoid function is shifted to the right by 7 units and 5 units. In order to see the initial speed change more clearly, only the first 200 steps of training results are taken. The convergence effect of NLIT method is shown as the line. The convergence curves of the remaining 4 methods are shown by the dotted line. there is overlap phenomenon due to the close values. It has been shown that NLIT method has obvious advantages in training speed for right-offset S-shaped functions.

6 CONCLUSION

From the point of view of solving the interval tolerance solution of linear systems, this paper improves the LIT method proposed by S.P.Adam et al. through theoretical analysis, and proposes a neural network weight initialization method (NLIT) which is suitable for the asymmetric activation function and has low computational cost. The method proposes the interval of neural network weight initialization, and its theoretical proof and experimental verification are also carried out. The NLIT method makes the input data fall into the preset activation interval as much as possible, which speeds up the convergence rate of the network, further expands the selection range of activation function, and realizes the determination method of weight initialization interval based on asymmetric activation function.

The NLIT method has further significance. In the current deep learning, the gradient vanishing problem is very serious if the saturated activation function of Sigmoid series is used. Meanwhile, the Relu function widely used is usually a linear function, which is closer to the shape of the S-system function after shifting to the right. Although the Relu functions can avoid the disappearance of the gradient during the learning process and improve the convergence speed, the network will have a better nonlinear fitting effect if an S-shaped curve is used instead of a straight line. In theory, NLIT method can make the single hidden layer neural network to avoid gradient disappear after using offset s-shaped activation function. The next research direction is how to initialize all inter-layer weights for multi-hidden layer neural network to avoid the disappearance of large area gradient as far as possible, so that S-series functions can have better application effect in deep learning.

REFERENCES

1. Boers E, Kuiper H. Biological Metaphors and the Design of Modular Artificial Neural Networks[M]. Master's thesis, Leiden University Netherlands. 1992: 6-19.

2. Nguyen D, Widrow B. Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights[C]. In Proceedings of the international joint conference on neural networks, IJCNN'90, IEEE, 1990,3:21-26.

3. Kim M C, Choi C H. A New Weight Initialization Method for the MLP with the BP in Multiclass Classification Problems[J]. Neural Processing Letters, 1997, 6(1-2): 11-23.

4. Kim, Y.K., & Ra, J.B. Weight value initialization for improving training speed in the back-propagation network[C]. In Proceedings of the international joint conference on neural networks, IJCNN'91 Seattle, WA,1991, 3 :2396–2401.

5. Drago, G. P., & Ridella, S. Statistically controlled activation weight initialization (SCAWI). IEEE Transactions on Neural Networks, 1992, 3, 627–631.

6. Mercedes Fernández-redondo, Carlos Hernández-espinosa. Weight initialization methods for multilayer feedforward[C]. European Symposium on ESANN, Bruges Belgium, 2001: 119-124.

7. Li, G., Alnuweiri, H., Wu, Y. Acceleration of Backpropagations through Initial Weight Pre-Training with Delta Rule[C]. Proc. of the IEEE Int. Conference on Neural Networks, ICNN'93, 1993, 1: 580-585.

8. Palubinskas, G. Data-driven Weight Initialization of Back-propagation for Pattern Recognition[C]. Pro. of the Int. Conf. on Artificial Neural Networks, 1994, 2: 851-854.

9. Shimodaira, H. A Weight Value Initialization Method for Improved Learning Performance of the Back Propagation Algorithm in Neural. Networks. Proc. Of the 6th International Conference on Tools with Artificial Intelligence,1994: 672-675.

10. Ho-Sub Yoon, Chang-Seok Bae, Byung-Woo Min. Neural networks using modified initial connection strengths by the importance of feature elements. Int. Joint Conf. on Systems, Man and Cybernetics, 1995,1: 458-461.

11. Bengio Y, Glorot X. Understanding the difficulty of training deep feed forward neural networks. Proc. AISTATS, 2010, 9: 249-256.

12. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jia Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification [C]. ICCV, 2016: 1-11.

13. Wessels L F A, Barnard E. Avoiding false local minima by proper initialization of connections[J]. IEEE Transactions on Neural Networks, 1992, 3(6): 899-905.

14. Jiang N, Xu J, Zhang S. Neural network control of networked redundant manipulator system with weight initialization method[J]. Neurocomputing, 2018, 307(9): 117-129.

15. Yam J Y F, Chow T W S. A weight initialization method for improving training speed in feedforward neural network[J]. Neurocomputing, 2000, 30(1/4): 219-232.

16. S.P.Adam, D.A.Karras, G.D.Magoulas, M.N.Vrahatis. Solving the linear interval tolerance problem for weight initialization of neural networks[J]. Neural Networks, 2014, 54: 17-37.

17. Hansen E R. Bounding the Solution of Interval Linear Equations[J]. Siam Journal on Numerical Analysis, 1992, 29(5): 1493-1503.

18. Alefeld G, Günter Mayer. Interval analysis: theory and applications[J]. Journal of Computational and Applied Mathematics, 2000, 121(1-2): 421-464.

19. Beaumont O, Philippe B. Linear Interval Tolerance Problem and Linear Programming Techniques[J]. Reliable Computing, 2001, 7(6): 433-447.

20. Yang D, Li Z, Wu W. Extreme learning machine for interval neural networks[J]. Neural Computing & Applications, 2016, 27(1):3-8.

21. Guan S, Zhang Z, Cui Z. Modeling uncertain dynamic plants with interval neural networks by bounded-error data[J]. IEEE Access, 2020(8): 9809-9820.
22. Jinwu Z. The application of MATLAB in mathematical modeling. [M]. Beijing：Beijing University of Aeronautics and Astronautics Press, 2011: 121-123.