New method of training two-layer sigmoid neural networks using regularization

IOP Conf. Series: Materials Science and Engineering **537** (2019) 042055 doi:10.1088/1757-899X/537/4/042055

New method of training two-layer sigmoid neural networks using regularization

V N Krutikov[1], L A Kazakovtsev[2,3], G Sh Shkaberina[2] and V L Kazakovtsev[4]

[1]Kemerovo State University, 6 Krasnaya street, Kemerovo, 650000, Russia
[2]Reshetnev Siberian State University of Science and Technology,
31, Krasnoyarsky Rabochy Av., Krasnoyarsk, 660037, Russia
[3]Siberian Federal University, 79, Svobodny Prospect, Krasnoyarsk, Russia
[4]Saint Petersburg National Research University of Information Technologies,
Mechanics and Optics, 49, Kronverksky Av., Saint Petersburg, 197101, Russia

E-mail: levk@bk.ru

Abstract. We propose a complex learning algorithm for sigmoid Artificial Neural Networks (ANN). We introduce the concept of the working area of a neuron for sigmoid ANNs in the form of a band in the attribute space, its width and location associated with the center line of the band to a fixed point. We define of the centers and widths of the working areas of neurons by analogy to the radial ANNs. On this basis, an algorithm for selecting the initial approximation of network parameters, ensuring uniform coverage of the data area with neuron working areas was developed. Network learning is carried out using a non-smooth regularizer designed to smooth and remove non-informative neurons. The results of the computational experiment illustrate the efficiency of the proposed integrated approach.

1. Introduction

Artificial neural networks are widely used to implement artificial intelligence (AI) [1, 2]. Some theoretical results prove the possibility of approximating continuous functions by means of a superposition of nonlinear functions [3]. These results is directly related to the two-layer ANN of the sigmoid type [4]. However, in various applications of the ANN [4] in building a network [4], a complex of problems appears. It is associated with choosing a good initial approximation and preserving its positive properties while learning with the redundancy of description in the form of an excessively large number of neurons and its simultaneous insufficiency approximation of data in certain parts of the data area. Such problems do not allow for providing good generalizing properties of the network [4]. We propose an approach to the development of an integrated learning algorithm where steps that eliminate the problems mentioned above are algorithmically implemented.

It is possible to introduce the concept of the working area of a neuron in the form of a band in the characteristic area, its width, location and reference of the central line of the strip to a fixed point for sigmoid ANNs just as the centers and widths of the working areas of neurons are defined in radial ANNs. The ability to manage the location of bands in the working area of neurons allows us to achieve a uniform coverage of the approximation area by working areas of neurons at the initial stage of setting the parameters of the ANN. The preliminary training of the network is carried out with specific areas of the neurons attached to specific points in the data area to ensure the preservation of such coverage of the

approximation area by the working areas of neurons. This makes it possible to obtain the uniform approximation in the data area at the first stage of training that cannot deteriorate with further training.

While training a network two approaches are used in order to get ANN with good generalizing properties. The first one uses various smoothing functionals, and the training proceeds to the complete minimization of the sum of squared deviations with a regularizing smoothing supplement. This approach is the most universal one, since various minimization algorithms (preferably fast and not resource-intensive) can be used to solve the problem. Also the generalizing properties of the ANN are determined by the value of the regularizing supplement. The advantage of this approach is the possibility of using the obtained solution or its part as the initial approximation in schemes with a changing weight of the regularizer, as well as in changing the network structure as a result of some neurons removal

According to the second approach an algorithm should have the regularizing property. The algorithm stops on the basis of the deterioration criterion of the generalized properties of the ANN. According to this approach in many cases, Levenberg-Marquardt (LM) algorithm [5] is effective, since it uses regularization to minimize a linearized model. The disadvantages of the second approach include the dependence of the quality of data approximation on the minimization algorithm, the inability to use the resulting solution for its subsequent improvement, the inability to use tools to remove low-informative components of the model, and early termination of learning is possible when part of the data area receives relearning with general network training.

We apply the first approach with the use of non-smooth functionals. The smoothing training in the presented algorithm is used both at the stage of calculating the initial approximation and at cyclically repeated stages of network training with the removal of little significant neurons. Smoothing allows obtaining the acceptable approximation even under the condition of excessive description, and the removal of uninformative neurons improves the quality of the approximation. Due to the use of non-smooth regularizers, the sub-gradient minimization method with stretching – space extension of the space is used as a training algorithm [6]. The use of non-smooth regularization allowed eliminating the effects of retraining and effectively removing unimportant neurons. The article presents examples of solving the tasks of training the scientific research system confirming the effectiveness of the integrated approach of training the scientific system.

2. Formulation of the approximation problem

Consider the approximation problem [4]:

$$W^* = \arg\min_W E(\alpha, W, D), \tag{1}$$

$$E(\alpha, W, D) = \sum_{x,y \in D} \left(y - f(x, W)\right)^2 + \alpha R(W),$$

where $D = \{(x^i, y_i) \mid x^i \in R^p,\ y_i \in R^1\},\ i = 1,...,N\}$ is the observation data, α is the regularization parameters, $f(x, w)$ is an approximating function, $x \in R^p$ is a data vector, $W \in R^n$ is a vector of adjustable parameters, *p* and *n* are their dimensions and

$$R(W) = \sum_{i \in I}^{n} (|w_i| + \varepsilon)^{\gamma} \quad (\varepsilon = 10^{-6},\ \gamma = 0.7)$$

is non-smooth regularizer [5]; *I* is a set of vector components involved in regularization. The regularizer $R(W)$ suppresses components of a vector W.

In the problem of approximation it is required to train a two-layer sigmoid neural network [7] of the following type (to estimate its unknown parameters *w*) by a network of direct propagation according to data D,

$$f(x, W) = w_0^{(2)} + \sum_{i=1}^{m} w_i^{(2)} \varphi(s_i), \tag{2}$$

where

$$\varphi(s) = 1/(1+exp(-s)), \tag{3}$$

$$s_i = w_{i0}^{(1)} + \sum_{j=1}^{p} x_j \, w_{ij}^{(1)} \;, \; i = 1,2,...,m \,, \tag{4}$$

x_j is vector components $x \in R^p$, $W = ((w_i^{(2)}, i = 0,...,m), (w_{ij}^{(1)}, j = 0,...,p, i = 1,...,m))$ is a set of unknown parameters that are to be estimated according to (1), $\varphi(s)$ is a neuron activation function, m is a number of neurons. In (2) one can use an arbitrary sigmoid activation function.

We will use the relaxation sub-gradient method with stretching – space extension (SMDM) to solve optimization problems (1) [6].

3. Algorithm for finding the initial approximation

The working area of the neuron (3) is located in some neighborhood of the hyper plane $s_i=0$ (4), and in other areas the values $\varphi(s_i)$ are close to their asymptotes. The standard practice is to select the initial approximation (IA) randomly [7].

In the proposed algorithm, the ANN approximation is at the fixed position of the working areas of the neurons with the help of the given centers $c_i \in R^p$, $i = 1,2,...,m,$ in the approximation area determined by the data. In this case, the neurons a free unit is excluded, and in (2) the expression will be used:

$$s_i = \sum_{j=1}^{p} (x_j - c_{ij}) w_{ij}^{(1)}, \;\; i = 1,2,...,m \,, \tag{5}$$

where the components of the vector w for neurons do not contain free units.

Centers c_i can be found by some data clustering algorithm. That will ensure that neurons will be located in areas with high data density. We use the maximin algorithm [8] where two data points that are the most distant from each other are selected as the first two centers.

Initially, problem (1) is solved with the formulation (2), (3), (5) using the found centers. This allows covering the entire data area with working areas at the first stage. The presence of regularization, even with an excessive number of parameters compared to the amount of data, allows getting an acceptable solution at this stage.

Having solved the problem (1) with fixed centers, it is necessary to return back to the original description of the network in the form (2), (3), (4) by forming a free member of the neuron, and leave the remaining parameters unchanged:

$$w_{i0}^{(1)} = -\sum_{j=1}^{p} c_{ij} \, w_{ij}^{(1)}, \;\; i = 1,2,...,m \,. \tag{6}$$

Such an algorithm for finding the initial approximation of the sigmoid ANN ensures that the data area will be covered by the working areas of the neurons.

4. Compound Training Algorithm

In the algorithm described below, the initial approximation is first found with the fixed working areas of neurons, and then operations of removing slightly significant neurons are alternately performed, then the truncated network learning is fulfilled.

Algorithm 1. Network Training Algorithm

Step 1. Form centers $c_i \in R^p$, $i = 1,2,...,m$, where m is the initial number of neurons with the data D, using the maximin algorithm. Set α, a regularization parameter. Determine the initial parameters $w_{ij}^{(1)}$, $j = 1,...,p$, $i = 1,...,m$, in (5) and the parameters $w_i^{(2)}$, $i = 0,...,m,$ in (2) by a sensor of uniformly distributed random numbers for each neuron.

Step 2. Solve the problem of estimating the parameters W of the neural network (1) for the ANN (2), (3), (5) with the fixed centers, with the following regularizer:

$$R(W)=\sum_i^m(\left|w_i^{(2)}\right|+\varepsilon)^\gamma \ (\varepsilon=10^{-6},\ \gamma=0.7). \tag{7}$$

To exclude expressions from the ANN model and transferring from the expression (5) to (4) for centers, we form additional parameters by formula (6). The resulting set of parameters is denoted by W^0.

Step 3. For k=0, 1, ... follow steps:

Step 3.1. Put $S_0=S(W^k,D)$, where

$$S(W,D)=\sqrt{\sum_{x,y\in D}(y-f(x,w))^2/N}\ . \tag{8}$$

Excluding the corresponding variables from the parameters W^k, remove successively neurons for which the inequality $S(W,D)\le(1+\varepsilon ps)S_0$ holds, where $\varepsilon ps=0.1$. If none of the neurons could be removed, then a neuron, resulting in the smallest increase in the value (8) is removed. If the number of neurons is less than three, then finish the operation of the algorithm.

Step 3.2. Using the parameters of the neural network obtained at the previous stage as the initial ones, obtain a new approximation W^{k+1}, solving the task (1) for the form of the ANN (2)-(4), with the regularizer (7).

With a limited number of data, the ANN $f(x,W^k)$ with a number of parameters n not exceeding N and having the smallest value of the exponent (8) is selected as the final approximation model.

5. Results of a computational experiment

In the following tasks, the neuron activation function $\varphi(s)=1/1+exp(-s)$ was used. The maximum deviation Δ of the neural network from the test function and the value $S=S(W,D)$ calculated on a set of 1000 data placed by the sensor of uniformly distributed random numbers in the data area were used as the quality criteria for the approximation. The regularization parameter was chosen within $\alpha\in[10^{-6},10^{-10}]$.

In [7], on the data with N = 625, formed by a sensor of uniform random numbers in the area $\Omega=[-3,\ 3]\times[-3,\ 3]$, the function was approximated:

$$f_1(x_1,x_2)=3(1-x_1)^2\,exp(-{x_1}^2-(x_2+1))-$$
$$-10(x_1/5-{x_1}^3-{x_2}^5)\,exp({x_1}^2-{x_2}^2)-exp(-(x_2+1)-{x_2}^2)/3.$$

The maximum deviation obtained in [7] by the ANN on a test sample of 1000 data was Δ = 0.06 [7].

With the help of the Compound Training Algorithm (CTA) on the sample with a smaller number of data is N = 350, the initial number of neurons is m_0 =70 and $\alpha=10^{-9}$ the ANN was obtained (m = 67 n = 269) with the deviation Δ = 0.056 and the value S = 0.0078. When a number of data was N = 600 and the initial number of neurons was m_0 =70 the ANN was obtained (m = 63 n = 253) with deviation Δ = 0.017 and S = 0.0015.

The similar calculation results were obtained by the CTA with γ = 0.7 in (7) and $\alpha=10^{-9}$: 1) N=350, m=66, n=265, Δ=0.058, S= 0.0072; 2) N=600, m=65, n=261 , Δ=0.041, S= 0.0034.

In [7], the function was approximated on the data with N = 500, formed in the area $\Omega=[0,1]\times[0,1]$ by a sensor of uniform random numbers:

$$f_2(x_1,x_2)=\frac{1}{2}sin(\pi{x_1}^2)sin(2\pi x_2).$$

The maximum deviation obtained in [7] by the ANN at m = 41 on a test sample of 1000 data was Δ = 0.15. The ANN (m = 48, n = 193) with deviation Δ = 0.018 and value S = 0.0028 was obtained by CTA on the sample with a smaller number of data N = 150 and with the initial number of neurons m_0 =70 and $\alpha = 10^{-7}$. Here, a number of model parameters is greater than a number of data. However, the regularization helps to get a good approximation. With the further removal of neurons (m = 35, n = 141), the deviation is Δ = 0.027 and S = 0.0048. Here, a number of model parameters is less than a number of data. This number of neurons is not enough for a better description.

Similar calculation results were obtained by the Compound Training Algorithm with $\gamma = 0.7$ в (7) and $\alpha = 10^{-9}$: 1) N=150, m=69, n=277, Δ=0.079, S= 0.0080; 2) N=150, m=35, n=141 , Δ=0.058, S= 0.0061.

The example of the regularized algorithm application without the direct presence of the regularization in a minimized function can be given in [9].

On the data of the function $f_3(x_1, x_2) = x_1^2 + x_2^2$ in the area $\Omega = [-3, 3] \times [-3, 3]$ (on evenly distributed data in the area with their number N = 100), the CTA made several miscalculations, with different numbers (N = 35, 50, 100) and $\alpha = 10^{-10}$. Due to the fact that the training performance is checked on an independent sample, as before, the root mean square error per 1000 new data was calculated. On a sample of N = 100, the initial number of neurons m_0 = 30 was obtained by the ANN (m = 16, n = 65) with the value S^2=5.3*10^{-10}. With N = 50, the initial number of neurons m_0 =30 is obtained by ANN (m = 14, n = 57) with the value S^2=6.5*10^{-9}. With N = 35, the initial number of neurons m_0 =30 is obtained by the ANN (m = 14, n = 57) with the value S^2=4.1*10^{-7}. Here in the last two cases, a number of network parameters exceeds a number of data. Nevertheless, all the obtained approximations turned out to be more qualitative in comparison to the approximation by the Levenberg-Marquardt method.

The similar calculation results were obtained by the Compound Training Algorithm with $\gamma = 0.7$ in (7) and $\alpha = 10^{-10}$: 1) N=100, m=27, n=109, S^2= 6.75*10^{-10}; 2) N=50, m=16, n=65 , S^2=5.88*10^{-7}; 3) N=35, m=13, n=53, S^2=1.57*10^{-6}.

The approximation results of three functions f_4, f_5 and f_6 given in table. 1 were obtained by the Compound Training Algorithm and the algorithms of the Surfer 6.0 software package. Characteristics Δ and S were on the test sample with a number of data N = 1000.

Table 1. Approximation results.

Test function	Surfer 6.0	Algorithm (CTA)
$f_4(x) = \sin 8x_1 + \cos 8x_2$ $\Omega = [0,1] \times [0,1]$	Δ=0.005, S=0.0006, N=100	Δ=0.00064, S=0.00013, N=100, m=13
$f_5(x) = \sin^2(5x_1) + \cos^2(5x_2)$ $\Omega = [0,1] \times [0,1]$	Δ=0.0063, S=0.0005, N=100	Δ=0.0009, S=0.00009, N=100, m=11,
$f_6(x) = \sin\left(8/1.4(x_1 + x_2)\right) + \cos\left(8/1.4(x_1 - x_2)\right)$ $\Omega = [0,1] \times [0,1]$	Δ=0.0066, S=0.0006, N=100	Δ=0.0023, S=0.0002, N=100, m=17,

Thus, the algorithm for finding the initial network approximation, the implemented method in CTA, together with the procedure for suppressing redundant neurons, makes it possible to obtain high-quality sigmoid ANN. It is necessary to use fast converging sub-gradient methods to implement the Compound Training Algorithm. We used a sub-gradient method with the stretching – space extension space.

6. Conclusion

The article proposes a compound sigmoid ANN training algorithm. By analogy with radial ANNs [10] where the centers and widths of the working areas of neurons are defined, the concept of the working area of a neuron is introduced for sigmoid ANNs in the form of a band in the characteristic area, its width and location associated with the central hyper plane of the strip to a fixed point. The algorithm for selecting the initial approximation of network parameters is based on the initial estimation of neuron parameters associated with selected centers. At the first stage this ensures the uniform coverage of the data area with neuron working areas. Network training is carried out using a non-smooth regularizer designed to smooth and remove non-informative neurons. Such a technique in the initial approximation makes it possible to specify an excess number of neurons. And, it is possible to obtain a quality-acceptable solution that improves as the less significant neurons move even at the first steps of the algorithm. The results of the computational experiment indicate the effectiveness of the proposed algorithm.

References

[1] Burnaev E V and Prikhodko P V 2013 On one technique for constructing ensembles of regression models. *Automat. and Telemeh*. **10** 36-54
[2] Gorbachenko V I and Zhukov M V 2017. Solution of boundary value problems of mathematical physics by means of networks of radial basis functions. *Journal of Computational Mathematics and Mathematical Physics* **57(1)** 133-43
[3] Gorban A N 1990 *Training of neural networks*. Moscow: ed. USSR-USA JV Paragraph
[4] Vorontsov K V 2016 *Mathematical methods of learning by precedents: course of lectures* // http://www.ccas.ru/voron/, http://www.machinelearning.ru
[5] Marquardt D W 1963 An Algorithm for Least-Squares Estimation of Nonlinear Pa-rameters. *Journal of the Society for Industrial and Applied Mathematics* **11(2)** 431–41
[6] Krutikov V N and Vershinin Ya N 2014 Subgradient method of minimization with correction of descent vectors on the basis of pairs of learning relationships. *Vestnik of the Kemerovo State University* **1-1(57)** 46-54
[7] Osovski S 2016 *Neural networks for information processing* (Moscow: Hot line-Telecom)
[8] Wang L, Zhu J and Zou H 2006 The doubly regularized support vector machine. *Statistica Sinica* **16** 589-615
[9] Tibshirani R J 1996 Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* **58(1)** 267-88
[10] Krutikov V N, Kazakovtsev L A and Kazakovtsev V L 2018 *IOP Conf. Ser.: Mater. Sci. Eng.* **450** 042010