# Neural Network Using for Prediction Spinal Diseases

Yue Wang[1], Vera Verba, Danil Bezumnov, Elena Tarasenko[2]
Moscow Technical University of Communications and Informatics
National Research University Higher School of Economics[2]
Moscow, Russian Federation
Urumqi, China[1]
wynfnone@gmail[1].com, verba@list.ru, danbez@yandex.ru, elenatar05@mail.ru[2]

***Abstract*—Spine disease is a universal health issue. More than half of the world's people are troubled by all kinds of back pain problems throughout their lives. Therefore, it is particularly vital to predicting related diseases. The identification and prediction of disease is an important and time-consuming task in medical diagnostic systems. With proper initial data, we can use machine learning to develop more effective prediction and diagnosis plans for certain diseases. This paper describes the implementation of a neural network based on a back propagation algorithm that predicts certain spinal diseases based on some biomechanical properties of the patient to aid in medical diagnosis.**

## I. Introduction

According to the World Health Organization, more than 2 million people die every year because they sit still for a long time; by 2020, 70% of the world's diseases are expected to be caused by sitting for too long and lack of exercise. According to the data, "more than half of the world's people are currently plagued by various back pain problems." Besides, spinal diseases have also shown a trend of becoming younger and younger. The development of spine disease is fierce and will be irrefutable as the number one enemy of human health in the future [1].

Since the middle of the 20th century, disease prediction has gradually evolved into an emerging discipline combining information technology and medical technology. Disease prediction refers to the use of personal symptoms and physical signs, and some tests to reflect whether there is a potential disease. In the field of information science, conventional methods for predicting diseases include regression prediction, time series prediction, gray prediction, Markov prediction, and artificial neural network. The most widely used is the artificial neural network algorithm.

Artificial neural networks refer to a series of mathematical models inspired by biology and neurology. These models mainly simulate the biological neural network by abstracting the neural network of the human brain, constructing artificial neurons, and establishing connections between artificial neurons according to a certain topological structure. In the field of artificial intelligence, artificial neural networks are also often referred to as neural networks or neural models. Unlike traditional computing, which uses a series of logical operators to perform tasks, neural networks use a network of nodes that act as neurons and synaptic analogs (edges) for data processing. The input data passes through the system and generates the outputs. The outputs are then compared with the known data. In 1943, McCulloch and Pitts modeled an artificial neuron, like a switch that receives information from other neurons and, depending on the total weighted input, is either activated or remains inactive. In the ANN node, the incoming signals are multiplied by the corresponding synapse weights and summed. These coefficients can be both positive and negative. Artificial neural network theory began to revive worldwide in the 1980s. In 1987, the first international neural network conference was held in California. Since then, various international conferences related to neural networks have emerged. Among them, the International Joint Conference on Neural Networks (IJCNN) has become an important communication platform for neural network researchers. Besides, more than a dozen international journals related to neural networks have appeared, including "IEEE Transactions on Neural Networks", "Journal of Artificial Neural Networks", and "Machine Learning". The role of artificial neural network theory research in the academic field is also becoming more and more critical.

In the field of disease prediction:

- Palaniappan et al. [2] proposed the use of decision trees, naive Bayes and neural networks for the prediction of heart disease, which promoted the establishment of significant patterns in heart disease.
- S. M. K. Chaitanya et al. [3] proposed using artificial neural networks and the gravitational search algorithm to detect chronic kidney disease.
- Mrs. S. Kalaiarasi et al. [4] proposed using a mobile phone camera to collect human facial data and create an application that helps diagnose and predict skin diseases through the device's image processing functions and machine learning.
- Tsipouras [5] gave a decision support system based on fuzzy rules for the diagnosis of coronary artery disease. The fuzzy model is used to optimize the parameters.
- Himaja Gadi et al. [6] proposed the implementation of the back-propagation algorithm to compute and compare the percentage of the output accuracy, which was used for medical diagnosis on various chest diseases (i.e., asthma, tuberculosis, lung cancer ).
- Patil et al. [7] introduced the K-means clustering algorithm to extract data suitable for heart attacks from the data warehouse.
- Srinivas et al. [8] proposed data mining methods to predict heart disease and use the patient's medical data

such as age, gender, blood pressure and blood glucose to predict the patient's risk of heart disease.

- Resul et al. [9] proposed predicting disease through neural network sets. This integrated approach is to create new models for disease prediction by combining existing methods.

The MTUCI department «Intellectual Systems in Management and Automation» conducts scientific work on the development and application of data mining methods for predicting situations in different subject areas, including solving the problem of analysis and classification of situations in the medical field using neural network modeling [10].

In this paper, the authors attempt to establish a neural network to assess the condition of the patient's spine based on the patient's biomechanical attributes and to predict the likelihood of developing a spinal disease.

## II. DATA SET ANALYSIS

The biomedical dataset used in this study was created by Dr. Henrique da Mota [16]. 15 records from the data set are shown in Table .

The data sets are divided into three categories: normal (100 patients), Disk Hernia (60 patients) or Spondylolisthesis (150 patients). Each patient is represented in the data set by six biomechanical attributes derived from the shape and orientation of the pelvis and lumbar spine (in this order): pelvic incidence, pelvic tilt, lumbar lordosis angle, sacral slope, pelvic radius and grade of spondylolisthesis.

Pelvic incidence is defined as the angle between a line perpendicular to the sacral plate at its midpoint and a line connecting this point to the femoral head axis. Pelvic incidence is easily measured in daily practice using radiographs. Pelvic tilt is the orientation of the pelvis concerning the thighbones and the rest of the body. The pelvis can tilt towards the front, back, or either side of the body. Lordosis is an increased inward curving of the lumbar spine (just above the buttocks). Small degree of both kyphotic and lordotic curvature is normal. Too much lordotic curving is called swayback (lordosis). The sacral slope denotes the spatial orientation of the pelvis, which varies according to the position, with a greater or lesser degree of tilt forwards or backwards in relation to a transverse axis passing through the two femoral heads. The spondylolisthesis is graded by measuring how much of a vertebral body has slipped forward over the body beneath it.

The visualization of changes occurring in the spine in certain types of diseases is shown in Fig. 1.(Disk Hernia) and Fig. 2.(Spondylolisthesis).

TABLE I. PARTIAL SOURCE DATA FROM THE SET [16]

| pelvic incidence (%) | pelvic tilt (°) | lumbar lordosis angle(°) | sacral slope (°) | pelvic radius (mm) | grade of SPONDYLOLISTHESIS | classification |
|---|---|---|---|---|---|---|
| 63.02 | 22.55 | 39.60 | 40.47 | 98.67 | -0.25 | Disk Hernia |
| 39.05 | 10.0 | 25.01 | 28.99 | 114.40 | 4.56 | Disk Hernia |
| 68.83 | 22.21 | 50.09 | 46.61 | 105.98 | -3.53 | Disk Hernia |
| 69.29 | 24.65 | 44.31 | 44.64 | 101.86 | 11.21 | Disk Hernia |
| 49.71 | 9.65 | 28.31 | 40.06 | 108.16 | 7.91 | Disk Hernia |
| 74.37 | 32.05 | 78.77 | 42.32 | 143.56 | 56.12 | Spondylolisthesis |
| 89.68 | 32.70 | 83.13 | 56.97 | 129.95 | 92.02 | Spondylolisthesis |
| 44.52 | 9.43 | 52 | 35.09 | 134.71 | 29.10 | Spondylolisthesis |
| 77.69 | 21.38 | 64.42 | 56.30 | 114.81 | 26.93 | Spondylolisthesis |
| 76.14 | 21.93 | 82.96 | 54.21 | 123.93 | 10.43 | Spondylolisthesis |
| 38.50 | 16.96 | 35.11 | 21.54 | 127.63 | 7.98 | Normal |
| 54.92 | 18.96 | 51.60 | 35.95 | 125.84 | 2.00 | Normal |
| 44.36 | 8.94 | 46.90 | 35.41 | 129.22 | 4.99 | Normal |
| 48.31 | 17.45 | 48 | 30.86 | 128.98 | -0.91 | Normal |
| 45.70 | 10.65 | 42.57 | 35.04 | 130.17 | -3.38 | Normal |

Fig. 1. Disk Hernia

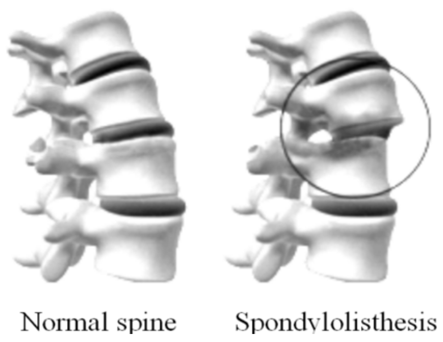



Fig. 2. Spondylolisthesis

Fig. 3 shows the relationship between every two properties.

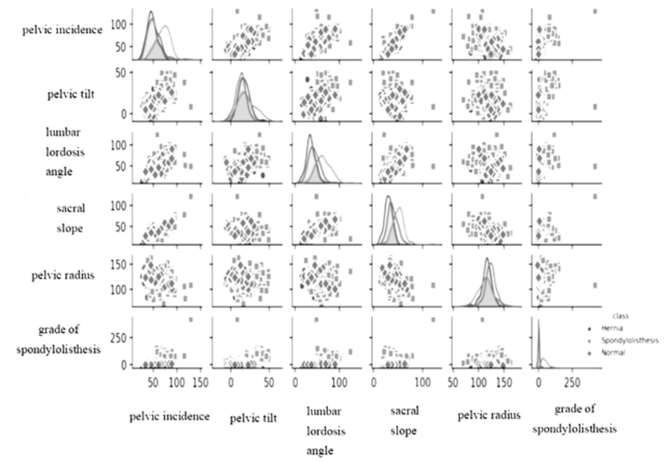



Fig. 3. The relationship between every two properties

As Fig. 4, 5 shown, the pelvic incidence is proportional to the pelvic tilt, the lumbar lordosis angle and sacral slope. In addition, most patients with spinal disorders have a higher grade of spondylolisthesis.

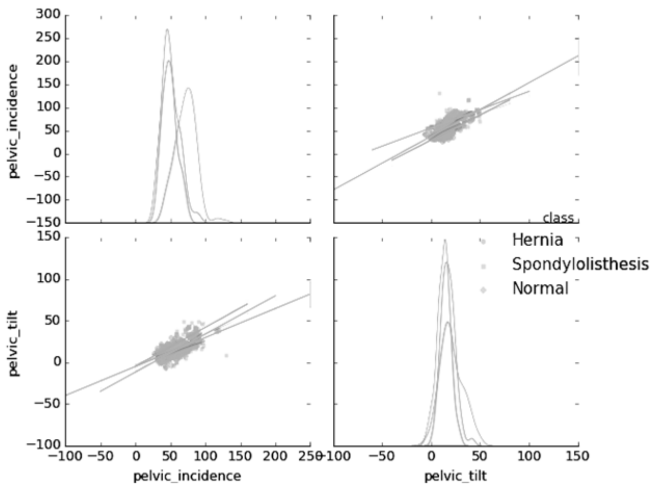



Fig. 4. The relationship between pelvic incidence and pelvic tilt

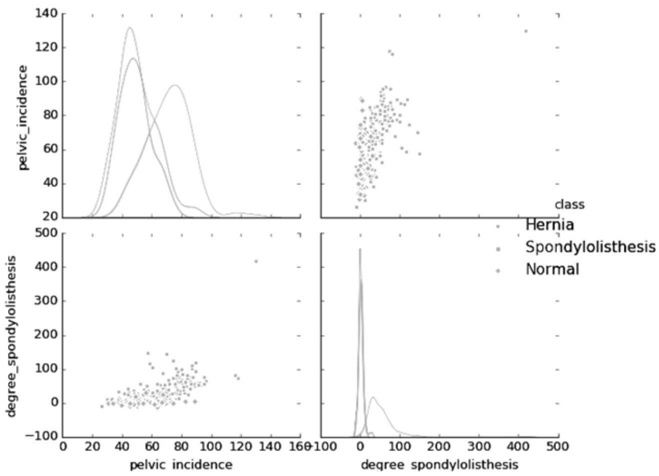



Fig. 5. The relationship between pelvic incidence and grade of spondylolisthesis

## III. Preprocessing Data

Different variables tend to be different in dimension, and normalization can eliminate the influence of dimension on the final result, making different variables comparable. In addition, normalization can also make the feature quantity within the same size ratio, which helps the gradient descent algorithm to converge faster.

In this study, because the data set has significant differences in dimensions and values, for example, the grade of spondylolisthesis is significantly different from other signs. Therefore, we need to normalize the data. Here we use the min-max normalization [17].

$$X_{norm} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \qquad (1)$$

where $X$ is an example of data, *Xmin* is the minimum value of the sample data, *Xmax* is the maximum value of the sample data.

## IV. Neural Network Design and Training

The BP neural network was developed in 1986 by a team of scientists led by Rumelhart and Mc Celland. It is already the most widely used neural network so far. BP neural network not only inherits the advantages of artificial neural network, but also has its own characteristics. Therefore, it has begun to emerge in various fields, such as data compression, pattern recognition, and perceptual prediction (including disease prediction). The characteristics of BP neural network are as follows:

- It has advantages in multi-dimensional input data and output data processing;
- Ability to handle arbitrarily complex nonlinear problems, so nonlinear mapping is more capable;
- Ability to process information or data in parallel, improving data processing power and performance;
- Has strong adaptability;

- Better performance in data fusion.

Based on these advantages, here we are proposing the algorithm using a feed-forward neural network. Backpropagation algorithm is used for learning procedure and for training the multilayer feed-forward network. The algorithm flow chart we proposed is shown in Fig. 6,

The BP artificial neural network uses a gradient search method to minimize the mean square error between the actual output value and the expected output value of the neural network, thus avoiding the occurrence of statistical averages. The training process of the BP algorithm is to correct the weighting coefficients while the error propagates backward. In the forward propagation phase, input information is processed from the input layer to the hidden layer and passed to the output layer. The neuron condition of each layer affects only the next layer of neurons. If the output layer cannot obtain the desired output, it will return to the backpropagation phase, return the error signal according to the original connection route, and then modify the weighting coefficient of each layer of neurons to minimize the error.

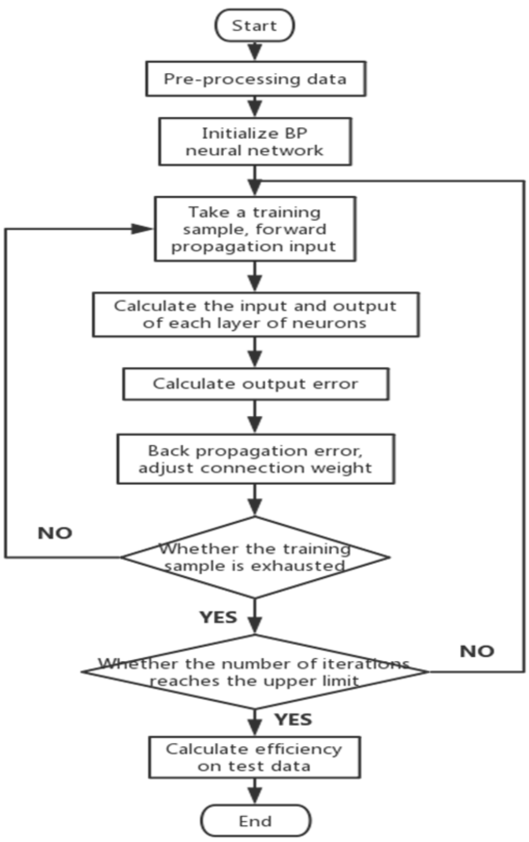



Fig. 6. Our proposed algorithm flow chart

The topological of the neural network is critical to estimating the accuracy of the training and testing data set. The choice of neural network architecture is usually based on experience. It has been found through experiments that the three-layer neural network has the highest prediction accuracy [18]. The input layer contains six neural units (corresponding to 6 features), the output layer includes three neural units (corresponding to 3 classification results). The neural unit contained in the hidden layer, we initially determined by the empirical formula:

$$m = \sqrt{n+l} + \alpha \tag{2}$$

where m - hidden layer neural unit number, l - output layer neural unit number, a - constant between 1 and 10.

After several training comparisons, when the number of hidden layer neural units is 10, the neural network has the fastest training speed and the best performance. Because of this, in the hidden layer, we choose ten neural units. Finally, the topology of the neural network we built is shown in Fig. 7.

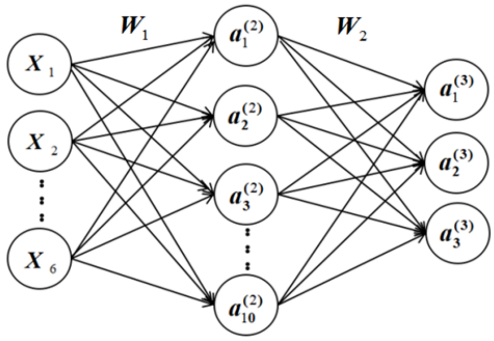



Fig. 7. Neural network architecture

The principle of BP neural network algorithm is as follows:

Hidden layer input vector: $hi$
Hidden layer output vector: $ho$
Output layer input vector: $yi$
Output layer output vector: $yo$
Expected output vector： $d_o$
The weight between the input layer and the hidden layer: $w_{ih}$
The weight between the hidden layer and the output layer: $w_{ho}$
Threshold value of each neuron in the hidden layer: $b_h$
Threshold value of each neuron in the output layer: $b_o$

Step1: Initialization weight.
Step2: Randomly select the Kth $x(k)$ input sample and its corresponding expected output $d_0(k)$.
Step3: Calculate the inputs and outputs of each neuron in the hidden layer.

$$ho_h(k) = f(hi_h(k)) \tag{3}$$

$$ho_h(k) = f(hi_h(k)) \tag{4}$$

$$yi_o(k) = \sum_{i=1}^{n} w_{ho} ho_h(k) - b_o \tag{5}$$

$$yo_o(k) = f(yi_o(k)) \tag{6}$$

Step4:Compare the expected output to the real output and get the partial derivative $\delta_o(k)$ of the error function for each neuron in the output layer.

$$\frac{\partial e}{\partial w_{ho}} = \frac{\partial e}{\partial yi_o} \frac{\partial yi_o}{w_{ho}} \tag{7}$$

$$\frac{\partial yi_o(k)}{\partial w_{ho}} = \frac{\partial(\sum_h^p w_{ho} ho_h(k) - b_o)}{\partial w_{ho}} \tag{8}$$

Step5: Use the connection weight of the hidden layer to the output layer, $\delta_o(k)$ and the output of the hidden layer, to calculate the partial derivative $\delta_h(k)$ of the error function for each neuron in the hidden layer.

$$\frac{\partial e}{\partial w_{ho}} = \frac{\partial e}{\partial yi_o} \frac{\partial yi_o}{w_{ho}} = -\delta_h(k) ho_h(k) \tag{9}$$

$$\frac{\partial e}{\partial w_{ih}} = \frac{\partial e}{\partial hi_h(k)} \frac{\partial hi_h(k)}{\partial w_{ih}} \tag{10}$$

$$\frac{\partial e}{\partial h_{ih(k)}} = -(\sum_{o=1}^{q} \delta_o(k) w_{ho}) f'(hi_h(k)) - \delta_h(k) \tag{11}$$

Step6: Adjust the weight $w_{ho}(k)$ by comparing the $\delta_o(k)$ of each neuron in the output layer with the output of each neuron in the hidden layer.

$$\Delta w_{ho}(k) = -\mu \delta_o(k) ho_{(h)}(k) \tag{12}$$

$$w_{ho}^{N+1} = w_{ho}^{N} + \eta \delta_o(k) ho_h(k) \tag{13}$$

Step7: Adjust the weight by comparing the $\delta_h(k)$ of each neuron in the hidden layer with the input of each neuron in the input layer.

Step8: Calculate global error.

Step9: Determine whether the error meets the preset condition. If the condition is met or the maximum number of trainings is reached, the algorithm ends; otherwise, Step3 is repeated to perform the next round of training.

For neural network training, we randomly selected 80% (250) of the data set (310), and the remaining 20% (60) will be used for testing.

Before starting the training of the neural network, the weights of its weights were randomly generated. The weight matrix is determined based on the number of neurons in the input layer (n_x), the hidden layer (n_h) and the output layer (n_y).

$$W_1 = np.random.randn(n\_h, n\_x + 1) * 0.1 \tag{14}$$

$$W_2 = np.random.randn(n\_y, n\_h + 1) * 0.1 \tag{15}$$

where *np.random.randn* - the function that generates several samples corresponding to the normal distribution.

Calculate the inputs and outputs of each neuron in the hidden layer:

$$\begin{aligned} &\mathbf{A1 = np.column\_stack((np.ones((X.shape[0],1)), X))} \\ &\mathbf{Z2 = np.dot(A1, W1.T)} \\ &\mathbf{A2 = sigmoid(Z2)} \end{aligned} \tag{16}$$

where *A1* - input value of the first layer, *A2* - input value of the hidden layer.

For the activation function, we choose the sigmoid function:

$$s = 1.0/(1.0 + np.\exp(-np.asarry(x))) \tag{17}$$

The learning process can be thought of as minimizing the cost function. Cost function [19]:

$$J = -\frac{1}{m} \sum_{i=0}^{m} (y^{(i)} \cdot \log(a^{(i)}) + (1 - y^{(i)}) \cdot \log(1 - a^{(i)})) \tag{18}$$

where $m$ - the number of training examples, $a$ – the output variables, $y$ - the output in the data set.

In order to minimize the cost function, we need to adjust the size of the weight by the back propagation algorithm [20].

Regarding the choice of the learning rate, if the learning rate is too small, the convergence will be too slow, and it will take a long time. If the learning rate is large, the training process may not converge at all, or even diverge. The amount of change in weight can be very large so that the optimization exceeds the minimum value and the loss function cannot be minimized. Here, we set the learning rate to 0.3, 0.5, and 0.8 respectively.The result is shown in Fig. 8.

From the Fig. 8, we can see that when the learning rate is 0.8, the loss function fluctuates, and the loss function with a learning rate of 0.5 decreases faster than when the learning rate is 0.3. So we chose 0.5 as the learning rate.

To avoid overfitting, we optimize the neural network by changing the value of the λ - regularization parameter [21]. In this case, we will keep the number of iterations constant, equal to 5000. Then take λ equal to 0.01, 0.1, 1, and calculate the loss function separately. The result, is shown in Fig. 9.

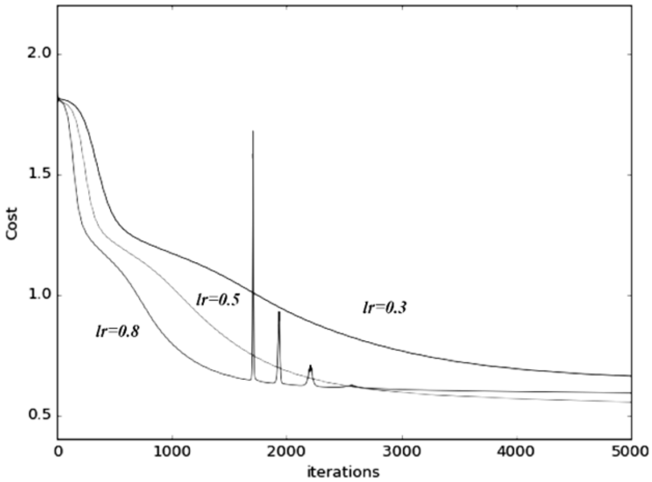



Fig. 8. The loss function at different learning rates (lr)

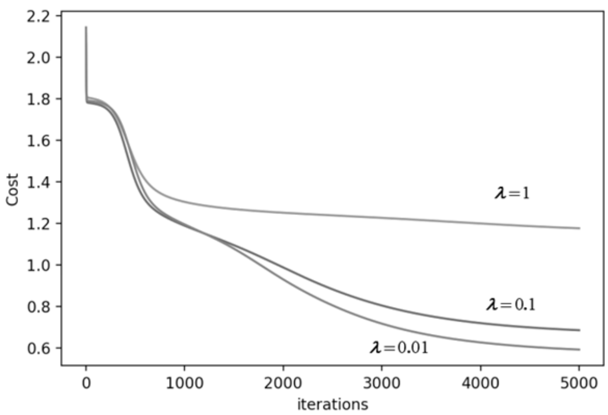
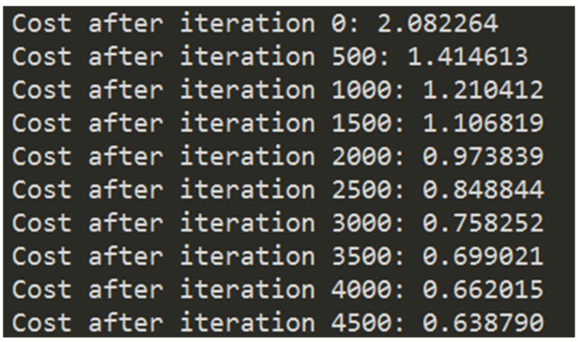



Fig. 9. The loss function for different values of the regularization parameter λ

As can be seen from the figure, for $\lambda = 0.01$, the loss function is relatively small, so we use this value in further calculations.

After 5000 iterations, stopping the training, we will get the best values of the weight parameters and save them in the files *W1*, *W2*, which we will use to work with the test set.

Fig. 10 shows the results of the cost function when the number of iterations changes. From this, we can see that when the number of iterations is from 3500 to 5000, the loss function drops from 0.69 to 0.63.



```
Cost after iteration 0: 2.082264
Cost after iteration 500: 1.414613
Cost after iteration 1000: 1.210412
Cost after iteration 1500: 1.106819
Cost after iteration 2000: 0.973839
Cost after iteration 2500: 0.848844
Cost after iteration 3000: 0.758252
Cost after iteration 3500: 0.699021
Cost after iteration 4000: 0.662015
Cost after iteration 4500: 0.638790
```

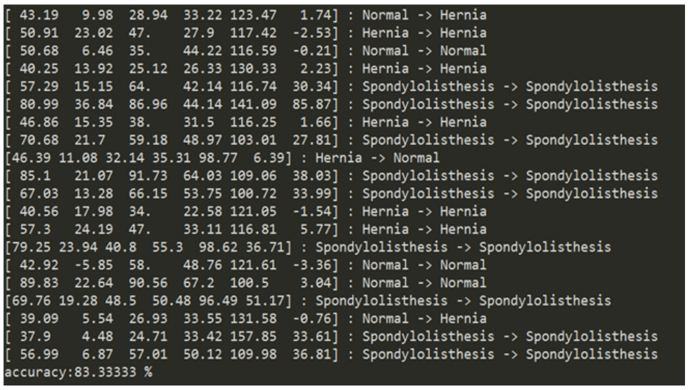
Fig. 10. The change of the loss function with the number of iterations

Thus, the network was trained with the following parameters:

- number of training iterations = 5000,
- learning rate = 0.5,
- regularization parameter $\lambda = 0.01$.

## V. Test Results

For testing, we selected the remaining 20% (60) of the data set (310) as the test set. Fig. 11 shows the test results, from which we can see that the accuracy of testing is 83%. In the same figure, we can also see the discrepancy between the predicted and test results. For example, the first record in the figure, the output is Hernia, but the actual is Normal.



```
[ 43.19   9.98  28.94  33.22 123.47   1.74] : Normal -> Hernia
[ 50.91  23.02  47.    27.9  117.42  -2.53] : Hernia -> Hernia
[ 50.68   6.46  35.    44.22 116.59  -0.21] : Normal -> Normal
[ 40.25  13.92  25.12  26.33 130.33   2.23] : Hernia -> Hernia
[ 57.29  15.15  64.    42.14 116.74  30.34] : Spondylolisthesis -> Spondylolisthesis
[ 80.99  36.84  86.96  44.14 141.09  85.87] : Spondylolisthesis -> Spondylolisthesis
[ 46.86  15.35  38.    31.5  116.25   1.66] : Hernia -> Hernia
[ 70.68  21.7   59.18  48.97 103.01  27.81] : Spondylolisthesis -> Spondylolisthesis
[46.39 11.08 32.14 35.31 98.77  6.39] : Hernia -> Normal
[ 85.1   21.07  91.73  64.03 109.06  38.03] : Spondylolisthesis -> Spondylolisthesis
[ 67.03  13.28  66.15  53.75 100.72  33.99] : Spondylolisthesis -> Spondylolisthesis
[ 40.56  17.98  34.    22.58 121.05  -1.54] : Hernia -> Hernia
[ 57.3   24.19  47.    33.11 116.81   5.77] : Hernia -> Hernia
[79.25 23.94 40.8  55.3  98.62 36.71] : Spondylolisthesis -> Spondylolisthesis
[ 42.92  -5.85  58.    48.76 121.61  -3.36] : Normal -> Normal
[ 89.83  22.64  90.56  67.2  100.5    3.04] : Normal -> Normal
[69.76 19.28 48.5  50.48 96.49 51.17] : Spondylolisthesis -> Spondylolisthesis
[ 39.09   5.54  26.93  33.55 131.58  -0.76] : Normal -> Hernia
[ 37.9    4.48  24.71  33.42 157.85  33.61] : Spondylolisthesis -> Spondylolisthesis
[ 56.99   6.87  57.01  50.12 109.98  36.81] : Spondylolisthesis -> Spondylolisthesis
accuracy:83.33333 %
```

Fig.11. Test result

A three-class classification was used in this study. In order to more specifically describe the type of disease, it is necessary to expand the data set and increase the number of classes.

## VI. Conclusions

The article discusses the use of a simple neural network to determine the condition of the patient's spine based on certain patient's biomechanical features. The authors created an effective neural network architecture and implemented a learning algorithm. The accuracy of the test set is over 80%.The authors believe that in order to improve the prediction accuracy of neural networks, it is recommended to use a more complex neural network, improve the training set, and use accelerated calculation methods.

There may be other methods that are superior to the neural network chosen in this paper. However, in terms of the accuracy of the test results, the neural network based on back propagation algorithm is still satisfactory for the research in this paper. The authors are still in the preliminary stage of research in this field, and will try other methods in future research.

## References

[1] The origin and development of spinal diseases, Web: https://wenku.baidu.com/view/94c25684700abb68a882fb3c.html.

[2] Palaniappan S and Awang R., "Intelligent heart disease prediction system using data mining techniques", International Conference on Computer Systems & Applications, March 30-April 3, 2008, Doha, Qatar. New Jersey: IEEE Press, 2008: 108-115.

[3] S. M. K. Chaitanya and P. Rajesh Kumar, “Innovations in Electronics and Communication Engineering”, January 2019, pp. 441-448.
[4] Mrs. S.Kalaiarasi, Harsh Kumar and Sourav Patra, “Dermatological Disease Detection using Image Processing and Neural Networks”, International Journal of Computer Science and Mobile Applications,Vol.6 Issue. 4, April 2018, pg. 109-118.
[5] Tsipouras M G , Exarchos T P and Fotiadis D I, “Automated diagnosis of coronary artery disease based on data mining and fuzzy modeling”, IEEE Transactions on Information Technology in Biomedicine, 2008, 12(4): 447-458.
[6] Himaja Gadi, G. Lavanya Devi and N. Ramesh, “Diagnosis of Chest Diseases Using Artificial Neural Networks”, Internet of Things and Personalized Healthcare Systems, November 2018, Chapter 11, pp. 113-119.
[7] Patil S B and Kumaraswamy Y S., “Intelligent and effective heart attack prediction system using data mining and artificial neural network”, European Journal of Scientific Research, 2009, 31(4): 642-656.
[8] Srinivas K, Kavitha R B K and Govrdhan A, “Applications of data mining techniques in healthcare and prediction of heart attacks” International Journal on Computer Science and Engineering, 2010, 2(2): 250-255.
[9] Resul D, Turkoglu I and Sengur A, “Effective diagnosis of heart disease through neural networks ensembles”, International Journal of Expert Systems with Applications, 2009(36): 7675-7680.
[10] V. I. Voronov, K. V. Genchel, M. D. Artemov and D. N. Bezumnov, ““Surdotelephone” project with convolutional neural network”, 2018 Systems of Signals Generating and Processing in the Field of on Board Communications, Moscow, 2018, pp. 1-6.
[11] A. Goncharenko, L. I. Voronova, V. I. Voronov, A. A. Ezhov and D. V. Goryachev, “Automated support system designing for people with limited communication”, 2018 Systems of Signals Generating and Processing in the Field of on Board Communications, Moscow, 2018, pp. 1-7.
[12] Voronov Vyacheslav I., Genchel Ksenia V., Voronova Lilia I. and Travina Maria D. “Development of a Software Package Designed to Support Distance Education for Disabled People”, IEEE-International Conference "2018 Quality Management, Transport and Information Security, Information Technologies" (IT&QM&IS-2018), St.Petersburg, pp.746-751, in press.
[13] Mikhail D. Artemov, V. I. Voronov and L. I. Voronova, “Development of the Subsystem for Recognition of Simple Dynamic Gestures Using 3DCNNLSTM”, 24th Conference of Open Innovations Association FRUCT, Moscow, Russia, 8-12 April 2019, unpublished.
[14] Vyacheslav I. Voronov, Vladimir G. Strelnikov, Liliya I. Voronova and Artyom S. Trunov, “Faces 2D-Recognition and Identification Using the HOG Discriptors Method”, 24th Conference of Open Innovations Association FRUCT, Moscow, Russia, 8-12 April 2019, unpublished.
[15] Artem A. Goncharenko and Lilia I. Voronova, “Development of a Sign Language Recognition Information System Using Wireless Technologies for People with Hearing Impairments”, 24th Conference of Open Innovations Association FRUCT, Moscow, Russia, 8-12 April 2019, unpublished.
[16] Henrique da Mota, “Vertebral Column Data Set”, Web: http://archive.ics.uci.edu/ml/datasets/vertebral+column
[17] Normalization / standardization of machine learning, August 2017,Web: https://www.jianshu.com/p/f9bde6a37d75.
[18] L.I. Voronova and V.I. Ravens, Machine Learning: Regression Methods for Data Mining, MTUCI, 2018 - 81 p.
[19] Andrew Ng , Machine Learning Course at Stanford University, 2018, Web: https://www.coursera.org/learn/machinelearning.
[20] Chao Chao , “Getting Started with BP Neural Networks: From Principles to Applications”, October 2017, Web: https://gitbook.cn/books/59e42d6143757a535fe26a70/index.html.
[21] Neural networks: use cases, Web: http://md-it.ru/articles/html/article19.html.