Artificial Neural Network

Part 1 Introduction of Artificial Neural Network

BPNN(short for Back Propagation Neural Network) is a variety of ANN, whose attribute is forword transmission of signal and reversed shift of error. In the process of forward transmission, input signals are operated through a few middle layers and exported from output layer ultimately. The state of every nerve cell could only influence the state of the next layer’s nerve cell.

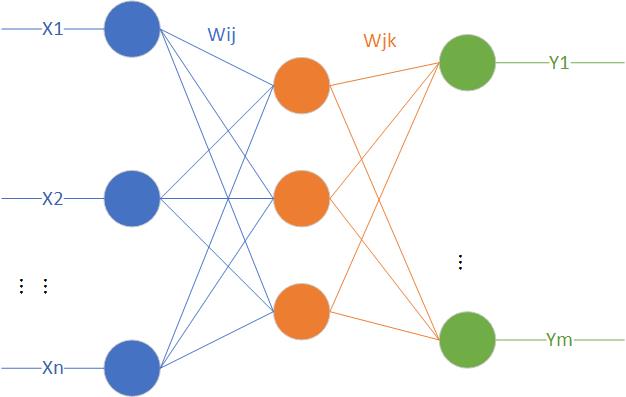


Figure1 BPNN Topological Structure Diagram

The training of BPNN includes following steps.

Step 1 Initialization of Network

According to the system’s input-output sequence, decide the number of input layer’s nodes, middle layer’s nodes and output layer’s nodes. Initialize weights among input layer, middle layer and output layer. Initialize thresholds from hidden layer and output layer. And preset other constants just like learning rate and nerve cells’ driving function.

Step 2 Calculate the Output of Hidden Layer

Calculate the output of hidden layer’s nodes on the basis of input variables, connection weights and thresholds from hidden layer. And we can have

Where represents the output of hidden layer, denotes the number of hidden layer’s nodes and f is the driving function of hidden layer. There are many different kinds of expression for , and we select

as its expression.

Step 3 Calculate the Output of Output Layer

Refer to values of the output of hidden layer , connection weights and thresholds , and then calculate the output of output layer. The basic formula is shown as:

Where represents the output of output layer.

Step 4 Calculate Errors

Calculate errors of the predictive values of the network. And here is the formula:

Step 5 Update Weights

We should use errors calculated from the previous step to update weights of network connections. We can have

Where is learning rate.

Step 6 Update Thresholds

Step 7 Estimate whether the iteration of algorithm is finished. If it is not finished, go back to the step 2.

Part 2 Implementation to Weight Calculation on ANNs

Step 1 Initialize parameters

We choose 16 indicators involving serval different aspects including economy, environment, education, social equality and so on. So we regard these 16 indicators as input nodes of ANN(short for Artificial Neural Network), which can be marked as . Of course, we should normalize the value of every indicator.

Then we are supposed to look for an index, which can express the comprehensive meaning of smart growth, to be regarded as the only one output node for the ANN model. After the assessment of serval indexes, we select HDI(short for Human Development Index) as this critical index. The Human Development Index (HDI) is a composite statistic (composite index) of [life expectancy](https://en.wikipedia.org/wiki/Life_expectancy), [education](https://en.wikipedia.org/wiki/Education), and [per capita income](https://en.wikipedia.org/wiki/Per_capita_income) indicators, which are used to rank countries into four tiers of [human development](https://en.wikipedia.org/wiki/Human_development_(humanity)).

Step 2 Select the Number of Middle Nodes

According to the reference, we can get the inequality for the number of middle layer’s nodes:

Where is the number of input layer’s nodes, denotes the number of hidden(middle) layer’s nodes and represents the number of output layer’s nodes.

So we preset the number of middle layer’s nodes range from 4 to 15. And we build a neural network with variable numbers of middle layer’s nodes. At the same time, we use this trained network to get output results with test input data. And we can determine the best number of hidden layer’s nodes through errors of these new results.

Finally we use MATLAB to find the best results in Figure 2:

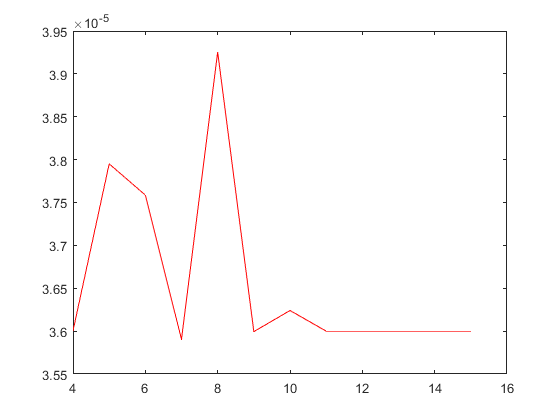


Figure2 Relationship Between Errors and Numbers of Hidden Layer’s Nodes

From the above figure, we can learn errors could be the smallest when the number of hidden layer’s nodes is 7.

Step 3 Get Weights

Finally, we can get the weights of different indexes with the best number of hidden layer’s nodes.

Part 3 Results Analysis

We have used mathematics modeling of ANNs and programmed with MATLAB to analyze weights of different indexes. And here are the Results.

Table1 Calculating Weights by Different Indexes

|  |  |
| --- | --- |
| Index | Weight |
| PM2.5 | 0.0519757415056530 |
| GDP per capita | 0.0752161603005759 |
| proportion of the third industry | 0.0592011234908896 |
| housing per capita | 0.0578843019579407 |
| green area per capita | 0.0578843019579407 |
| proportion of high school education | 0.0731738234110257 |
| population density | 0.0533150833476775 |
| road area per capita | 0.0576160186342898 |
| number of armarium per capita | 0.0520758132636086 |
| number of bus per capita | 0.0568970735423348 |
| disposable income per capita | 0.0593535084572225 |
| rate of employment | 0.0641632079857798 |
| investment of infrastructure per capita | 0.0735650647423245 |
| number of cars per capita | 0.0735650647423245 |
| proportion of cultivated land area | 0.0642756409502941 |
| number of scenic spots | 0.0671083786528661 |