Xi'an Jiaotong-Liverpool University

西交利物浦大学

CSE 301: Bio-Computation Assignment 3

Name: Kai-Yu Lu
ID: 1614649
Module Leader: Dr. Rui Yang

1. Introduction

Artificial neural network is the abbreviation of ANN, which commonly consists of input layers, hidden layers and output layers. Neural network is a computational model or a biological neural network used to estimate or approximate functions. After decades of development, the neural networks have been applied in numerous area such as pattern recognition, signal processing, assisted and automatic control.

This assignment requires students to implement MLP with BP neural network and RBF neural network with the RBF centers initialized from k-mean clustering for vehicle logo classification by using MATLAB. 80% of the data set will be used for training and the rest 20% of the data will be used for testing, which would be discussed during the report. With respect to MLP, the effects of different learning rate, momentum and number of hidden units should be explored. Regarding to RBF neural network, the effects of different number of RBF centers should be discussed based on the testing results. Eventually, confusion matrixes would be used for comparing the best MLP model and the best RBF model. his report intends to present a brief task description and certain backgrounds, testing results and thorough discussions based on the results and experience in finishing the assignment.

2. Methodology

2.1 MLP with back-propagation training algorithm

Multilayer perception is assumed as a logistic regression classifier and is able to learn non-linear functions by applying non-linear activation functions such as sigmoid function and hyperbolic tangent function. The error backpropagation learning algorithm is a supervised training algorithm based on MLP and its basic principle is gradient descent method. By adopting the gradient search mechanism, the mean square errors between the actual and the expected outputs could be minimized. Figure 1 the structure of the BP neural network.

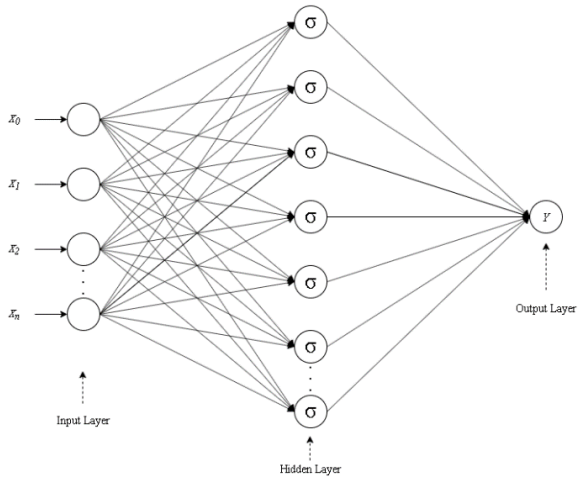

Figure 1: Stricture of BP neural network

The hidden layers indicated in Figure 1 is considered that more meaningful and useful information could be extracting from the input of complex tasks. In addition, the connecting topology of this structure is fully-connected, because every input unit is connected to all the units in the hidden layer and every hidden unit is connected to all the units in the output layer. In the meanwhile, every unit is not connected to the unit in the same layer, which guarantees the independence of the connection and the input data would be processed critically.

The error backpropagation algorithm contains two significant parts, which are forward-propagation and back-propagation. At the beginning, forward-propagation would be executed, input information

would be processed from the input layer to the output layer. Before transforming the data to the output layer, the output of the input data from input layer will be propagated to the hidden layer. As mentioned previously, the fully-connected topology of this neural network results in the highly independent for each unit, which means the states of units in next layer are only influenced by the sates of the previous layers. The formula for the forward-propagation is provided in followings.

The output of the hidden layer:

$$o_j = \sigma(s_j) = \frac{1}{1+e^{-s_j}}, \qquad s_j = \sum_{i=0}^{d} w_{ij} x_i$$

where x_i is the input and $\sigma(s_j)$ is the non-linear Sigmoid activation function.

The output of the output layer:

$$o_k = \sigma(s_k) = \frac{1}{1+e^{-s_k}}, \qquad s_k = \sum_{i=0}^{d} w_{jk} x_j$$

After the forward-propagation, back-propagation would be executed if the outputs from the output layer cannot satisfy the desired outputs. During forward-propagation, the errors between the expected output and the actual outputs would be calculated reversely according to the connected paths. The errors are desired to decrease via applying gradient descent rule to adjust the weights between each layer. The formulas for the back-propagation are provided in followings and could be referred to the lecture notes [1].

The benefit β_k at the units k in the output layer in condition of Sigmoid activation function:

$$\beta_k = o_k(1-o_k)(y_k - o_k)$$

The changed weights between j to k:

$$\Delta w_{jk} = \eta \beta_k o_j$$
$$\Delta w_{0k} = \eta \beta_j$$

The benefit β_j for the hidden units j:

$$\beta_j = o_j(1-o_j)(\sum_k \beta_k w_{jk})$$

The changed weights between i to j:

$$\Delta w_{ij} = \eta \beta_j o_i$$
$$\Delta w_{0j} = \eta \beta_j$$

Eventually, the weights should be updated:

$$w = \Delta w + w$$

The process of forward-propagation will be executed until the outputs from the output layer satisfy the desired outputs.

2.2 RBF neural network based on k-means

It is stated that the curve-fitting or function approximation could be regarded as the problem of finding interpolating surface that fits the training data optimally. The regularization network is mathematic model of the interpolating surface and the formula is:

$$F(x) = \sum_{i=1}^{N} w_i \exp\left(\frac{-||x - x_i||^2}{2\sigma_i^2}\right)$$

where w_i are weights and x_i are centers defined in advance [2].

However, excess computations between the training set and Gaussian function might result in computationally inefficient if the training sets are large. Additionally, in reality, it is required to invert quite large $\Phi^T\Phi$ matrices because of finding the weights of the linear basis function. In consideration of this problem, network topology is desired to be decreased and a solution approximating to the regularization network is promoting to be explored.

Radial-basis function (RBF) neural network is promoted to solve the problem of curve-fitting or function approximation in high dimensions. Similarly, RBF neural network consists 3 layers, which are input layers, hidden layers and output layers. It is significantly mentioned that:

- RBF has only one hidden layer
- Hidden layer is non-linear, while the output layer is linear.

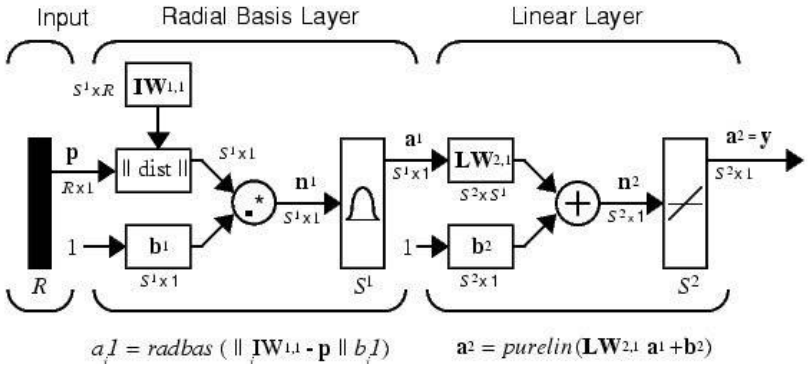

Figure 2: Structure of RBF neural network

The related formulas for the training algorithm are provided orderly in followings:

$$\phi_{ei} = \frac{exp(-||x - t_i||^2)}{2\sigma_i^2}$$

where :

- $i = 1, 2, 3, \ldots, n;\ e = 1, 2, 3, \ldots, N$. i is the number of centers and N is the number of input examples.
- weights:$w = (\Phi^T\Phi)^{-1}\Phi^T d$

3. Experimental Results and Analysis

3.1 MLP with back-propagation training algorithm

3.1.1 Experimental procedure:

- **Data acquisition and data processing:** obtain and divide the data set (logo.m). In the training set, divide first 80% of each total number of each logo as training set; last 20% of each total number

of each logo as testing set. This step guarantees the fixed training sets and fixed testing tests, because different training set or testing set might influence the performance of the trained neural network.

- **Build BP neural network:** newff() is used during this assignment. The initial weights and biases are set to 0 for guaranteeing identity.
- **Acquire different performances for varied number of hidden units:** plot MSE VS. number of hidden units
- **Acquire different performances for varied learning rate and momentum:** plot MSE VS. learning rate and MSE VS. momentum
- **Acquire the confuse matrix of the best MLP model.**

3.1.2 Result

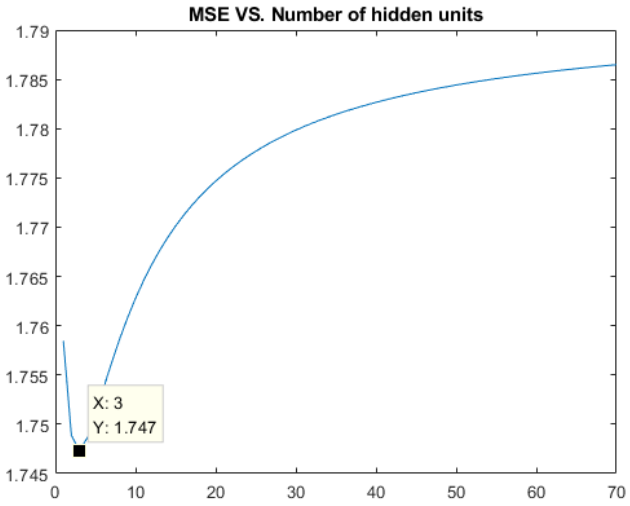

Figure 3: MSE VS. number of hidden units

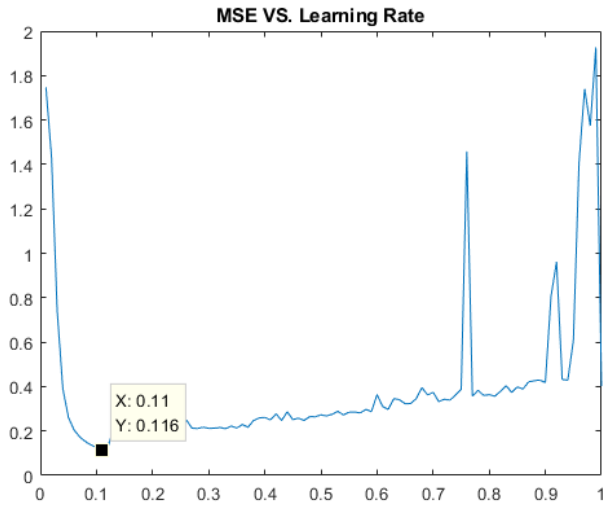

Figure 4: MSE VS. Learning rate for 3 hidden units

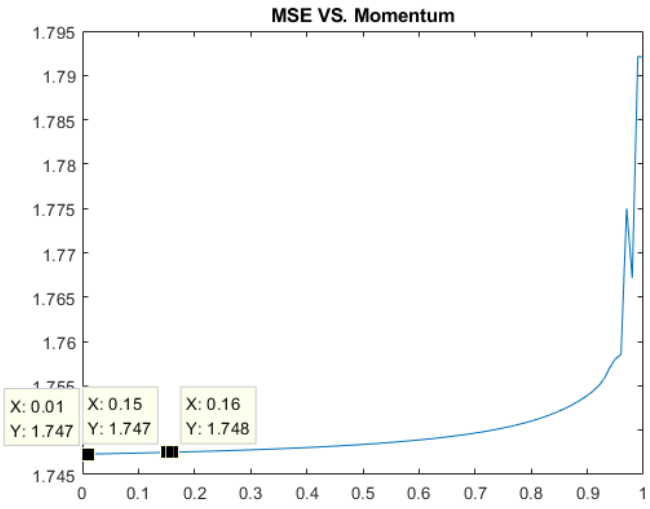

Figure 5: MSE VS. Momentum for 3 hidden units

3.2 RBF neural network based on k-means

3.2.1 Experimental procedure:

- **Data acquisition and data processing:** obtain and divide the data set (logo.m). In the training set, divide first 80% of each total number of each logo as training set; last 20% of each total number of each logo as testing set. This step guarantees the fixed training sets and fixed testing tests, because different training set or testing set might influence the performance of the trained neural network.
- **Initialize centers:** kmeans() is used during this assignment.
- **Acquire different performances for varied number of centers:** plot MSE VS. number of centers (hidden units).
- **Acquire the confuse matrix of the best RBF model.**

3.2.2Result

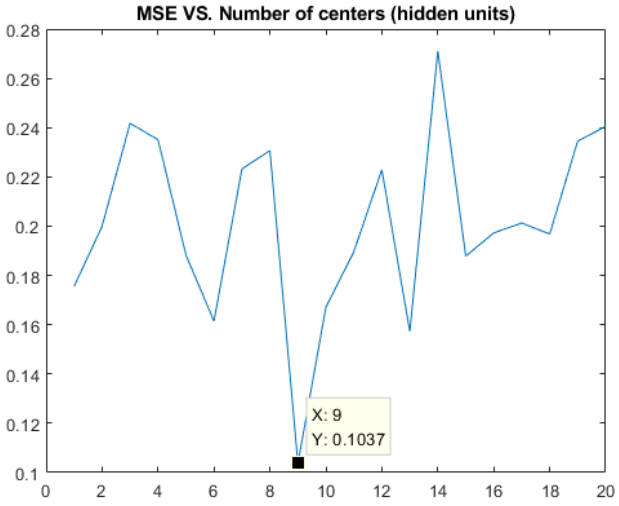

Figure 6: MSE VS. Number of centers

3.3 Best MLP model VS. Best RBF Model using confusion matrix

Figure 7: Confuse matrix of the best MLP model

Figure 8: Confuse matrix of the best RBF model

3.4 Results discussion and Analysis

From Figure 3, the performance of the neural network varies visibly when number of hidden unit changes, where the learning rate and momentum are both set to 0.1. It could be stated that the number of the hidden unit influences the performance of MLP neural network exceedingly, an appropriate number of the hidden unit would optimal the performance of the MLP neural network and vice versa.

From Figure 4: It could be observed that the learning rate affects the performance of MLP neural network. If the learning rate is too high, it will cause the oscillation nearby the local minimum and even failure of converging. If the learning rate is too low, the training process would be slow. Therefore, an appropriate learning rate would optimal the performance of the MLP neural network and vice versa.

Figure 5: After adding momentum, the time for training was reduced. The momentum increased 0.01 every time after each time of training. It could be seen that the low momentum will not affect the

performance of the MLP neural network significantly. However, if the momentum is too large, it will degrade the performance of the MLP neural network because it will cause the problem of oscillation nearby the local minimum.

Regarding to the performance of the RBF neural network indicated in Figure 6, it varies obviously every time. One of the best performances was when the number of centers was 9 and its minimum MSE was 0.1037. Due to the fact that the positions and number of centers are diverse for each training process, the performances were quite varied.

Figure 7 is the best performance of the MLP neural network. It is discovered that the number of hidden units, the learning rate and the momentum would comprehensively influence the performance of the MLP neural network. Selecting appropriate parameters is quite challenged when training the neural network. It is suggested that the value of the momentum should be smaller or equal to the value of the learning rate, which might present a well-behaved neural network.

Although Figure 8 shows the best performance of the RBF neural network, it is considered that the performance could be better. Therefore, it is suggested that more trails for the RBF neural networks should be operated so as to approach better performances. Additionally, an appropriate initialization would be a challenge for RBF neural network.

4. Conclusion

In conclusion, this assignment has satisfied all the requirements and the results are correct by verifying the theories. During this assignment, a fixed training set and fixed testing set are the significant factors, because neural networks might be sensitive to the data in training set. It is worth mentioning that the testing set was not used during training and it was used for testing stage such as finding the best performance of the models. It is believed that both the performance of the MLP neural network and RBF neural network could be optimized if the data set is larger. Likewise, more experiments should be done in future such as adjusting the number of hidden layers of MLP neural network and explore the best performance by changing learning rate and momentum. Regarding to RBF neural network, it is promoted that gradient descent rule could be combined with RBF neural network for updating the weights. After this assignment, students could build neural networks simply and test their performances by adjusting certain parameters. Finally, the knowledge from lectures has been reviewed and understood more deeply.

Reference

[1] *MULTI-LAYER PERCEPTRON* rev. 1 ed., Xi'an Jiaotong-Liverpool University, Suzhou, Jiangsu Province, 2019.

[2] *RADIAL-BASIS FUNCTION NETWORKS* rev. 1 ed., Xi'an Jiaotong-Liverpool University, Suzhou, Jiangsu Province, 2019.