Assignment Report for NN7207

Hu Minghui

September 2020

1 Introduction

The Self-organizing map(SOM) signifies a class of neural-network algorithms in the unsupervised-learning category. Its original form is proposed in 1980s. The main idea of SOM is that it forms a nonlinear projection of a high-dimensional data manifold on a regular, low-dimensional (usually 2D) grid[3]. In the display,the clustering of the data space as well as the metric-topological relations of the data items are clearly visible. f the data items are vectors, the components of which are variables with a definite meaning such as the descriptors of statistical data, or measurements that describe a process, the SOM grid can be used as a groundwork on which each of the variables can be displayed separately using grey-level or pseudocolor coding[2].

Radial basis function (RBF) networks are a commonly used type of artificial neural network for function approximation problems. Radial basis function networks are distinguished from other neural networks due to their universal approximation and faster learning speed. An RBF network is a type of feed forward neural network composed of three layers, namely the input layer, the hidden layer and the output layer[1]. The output of hidden layers are linear combination and fed to the output layer, therefore, the weight between the hidden layer and the output layer can be solved by the close-form function.

2 Methods

In this work, we use Self-organizing Map to generate certain number of center vectors, which are fed to the Radial Basis Function Network as the center of each neuron.

2.1 Self-organizing Map

In the training of the SOM network, data points are sequentially introduced to the SOM. In each iteration, the SOM neuron which is closest to the input unit is selected by Eq.1 . This unit is the Best Matching Unit (BMU) or winner.

$$||x - c_c|| = \underset{i}{min}\{||x - c_i||\} \tag{1}$$

where x is input vector,c_c is the selected center, Best Matching Unit (BMU), and c_i is the current center in the evaluation. The weight vectors are updated using Eq.2. Only the weight vectors which are inside the neighborhood radius h_{ci}, are updated.

$$c_i(n+1) = c_i(n) + \eta(n)h_{ci}(n)[x(n) - c_i(n)] \tag{2}$$

where η is the learning rate, which is adaptive during the training phase. The neighborhood function h_{ci} is the Gaussian function as shown in Eq.3.

$$h_{ci} = \exp(-\frac{d_{ci}^2}{2\sigma(n)^2}) \tag{3}$$

where d_{ci} is the Euclidean distance between to vectors. The whole training phase contains two part, self-organizing part and convergence part. In the self-organizing part, the learning rate is updated by Eq.4

$$\eta(n) = \eta_0 \exp(-\frac{n}{\tau_1}) \qquad \tau_1 \in \mathbb{R} \tag{4}$$

and the parameter $\sigma(n)$ in neighborhood function is updated as following:

$$\begin{aligned} \sigma(n) &= \sigma_0 \exp(-\frac{n}{\tau_2}) \\ \tau_2 &= \frac{1000}{\ln \sigma_0} \end{aligned} \tag{5}$$

In the convergence phase, the learning rate is set as a small number and only the BMU get updated.

2.2 Radial Basis Function Network

The first layer corresponds to the inputs of the network, the second is a hidden layer consisting of a number of RBF non-linear activation units, and the last one corresponds to the final output of the network. Activation functions in RBFNs are conventionally implemented as Gaussian functions. To illustrate the working flow of the RBFN, suppose we have a data set D which has N patterns.

The output of the ith activation function ϕ_i in the hidden layer of the network can be calculated using Eq.6 based on the distance between the input pattern x and the center c_i

$$\phi_i(||x - c_i||) = \exp(-\frac{||x - c_i||}{2\sigma_j^2}) \tag{6}$$

Here, $|| \bullet ||$ is the Euclidean norm, c_i and σ_j are the center and width of the hidden neuron j, respectively. Then, the output of the node k of the output layer of the network can be calculated using the Eq.7

$$y_k = \sum_{j=1}^{n} w_{jk}\phi_j(x) \tag{7}$$

The centers c are determined using SOM described before, the connection weights between the hidden layer and the output layer are found in a way like the common Mean Squared Error (MSE). The solution of $\mathbf{w}$ could be Eq.8.

$$\mathbf{w} = (\Phi^T \Phi)^{-1} \Phi^T y \tag{8}$$

or by pseudo-inverse:

$$\mathbf{w} = \Phi^{\dagger} y \tag{9}$$

3 Experiments

The experiments are carried out in my own laptop with GPU. *PyTorch* is used as the framework to construct the SOM and RBFN's architectures. The codes are available at `https://github.com/mhh0318/NN7207`.

3.1 Using SOM to find the center vectors

Here we set the initial learning rate η_0 as 0.1 in self-organizing phase. The initial σ_0 is calculated by the maximum distance of the map, which is $4\sqrt{2}$ in my experiment. The self-organizing iterations are set as 1000 while the convergence iterations are set as 8000. In the convergence phase, the learing rate is set as 0.01. The training results are shown in Fig.1 Each training point represent a scatter in the figure. Different class of the points are in different colors.

The numerical results are shown in the following tables.

	0	1	2	3
0	34	17	11	18
1	20	39	6	13
2	13	4	44	75
3	9	5	14	8

Table 1: the SOM with all data as points around the neurons

	0	1	2	3
0	1	0	5	1
1	20	32	2	8
2	13	4	3	2
3	9	5	3	8

Table 2: the number of data in class 1 (1) as points around the neurons

The purity of clusters is 0.89 in average.

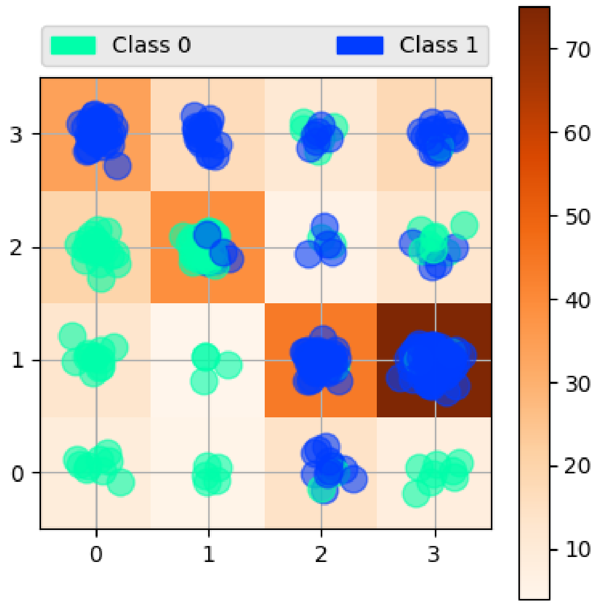

Figure 1: Visualize the SOM with all data as points around the neurons, also plot the density map with winner neuron counts in the background

	0	1	2	3
0	33	17	6	17
1	0	7	4	5
2	0	0	41	73
3	0	0	11	0

Table 3: the number of data in class 0 (-1) as points around the neurons

3.2 Using RBFN to predict labels

After we get the center vectors from SOM, we begin the training process of RBFN. Here are 16 neurons in the hidden layer and only 1 neuron in the output layer. The parameter σ of RBF is set by the following function:

$$\sigma = \frac{d_{max}}{\sqrt{2m}} \tag{10}$$

where d_{max} is the maximum distance in the center vectors and m is the number of vectors, which is 16 in this experiment.

The weight are solved by Eq.8, the result is presented as below:

$$\begin{aligned}\mathbf{w} =&[2.24, -1.58, 2.22, 11.42, -7.01, -23.94, 0.31, 0.48,\\ &-0.96, 0.34, 1.55, 1.35, 3.22, 2.25, 4.13, 1.39, -0.55]^T\end{aligned} \tag{11}$$

The RBFN include a bias term, and I set 0 as threshold in the final output neuron, in another word, I use the bi-polar activation function in the final layer. The accuracy of training data is 90.00%. In addition, Support vector machine performs 96.97% in the task under the training dataset, which is better than RBFN. The predicting labels are attached in the appendix.

4 Discussions

In this assignment, a two-class pattern classification problem is solved by using a Radial Basis Function Network. To obtain the center vector of each neuron in the hidden layer, Self-organizing map is employed, which can cluster the given data into certain number of groups and provide the corresponding center vectors. Support Vector Machine is also used to compare the prediction performance with RBFN.

A Predicting labels for provided test data

$$\begin{aligned}\hat{y} =&[1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, \\ &-1, 1, -1, 1, -1, 1, -1, 1]^T\end{aligned} \tag{12}$$

References

[1] David S Broomhead and David Lowe. Radial basis functions, multi-variable functional interpolation and adaptive networks. Technical report, Royal Signals and Radar Establishment Malvern (United Kingdom), 1988.

[2] Samuel Kaski, Jari Kangas, and Teuvo Kohonen. Bibliography of self-organizing map (SOM) papers: 1981–1997. *Neural computing surveys*, 1(3&4):1–176, 1998.

[3] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.