**Homework for chapter 1 (Introduction)**

**Paper sheet homework**

**1.4** Learning is one of the most significant ability of an artificial neural network. Supervised learning and unsupervised learning are the two typical learning types of learning. Please try your best to state supervised learning and unsupervised learning in precision respectively.

Supervised learning usually means we teach the computer to learn, and every input data (features) have a certain output (label). Supervised learning can learn the mapping between features and label, so we can predict the label of an unknown data.

Unsupervised learning usually means we need computer to learn by itself, all of data haven’t a certain label, and computer need directly predict the label of an unknown data. We also call this method clustering.

Generally speaking, supervised learning is better than unsupervised learning in precision, because supervised learning can find the relation between input and output though training set, and unsupervised learning need predict an unknown data without any information. However, it’s difficult to make sure correct labels of mass data, such as for NLP, we cannot sure every word’s label. So, unsupervised learning maybe is a good choice.

**1.5** Learning is one of the most significant ability of an artificial neural network. Hence the questions are learning from where? Learning what? And learning to get what?

Our algorithm needs learn from data, we can transport our problem to a math problem and use a serial of data to define problem. Algorithm learns the mapping between input and output, and uses output to modify the mapping. Finally, we can use the mapping to predict unknown data.

**1.6** Classification and regression are the two typical tasks that artificial neural networks can fulfill. Please define the two problems in precision respectively.

When the target variable that we’re trying to predict is continuous, we call the learning problem a regression problem. When  can take on only a small number of discrete values (such as if, given the living area, we wanted to predict if a dwelling is a house or an apartment, say), we call it a classification problem.

**1.7** State a classification problem in precision which is a linearly separable problem.

Spam email filtering is a linearly separable problem, and also is a classification problem. There are two classification: spam email or normal email. We can assume 1 as spam email’s label and 0 as normal email’s label, and use some keywords as features. If an email contains these keywords, the algorithm would classify it to label 1 or label 0.

**1.8** List as many as possible the application areas of the artificial neural networks.

* ***Character Recognition***: Neural networks can be used to recognize handwritten characters.
* ***Image Compression***: Neural networks can receive and process vast amounts of information at once, making them useful in image compression. With the Internet explosion and more sites using more images on their sites, using neural networks for image compression is worth a look.
* ***Stock Market Prediction***: The day-to-day business of the stock market is extremely complicated. Many factors weigh in whether a given stock will go up or down on any given day. Since neural networks can examine a lot of information quickly and sort it all out, they can be used to predict stock prices.
* ***Traveling Saleman's Problem***: Interestingly enough, neural networks can solve the traveling salesman problem, but only to a certain degree of approximation.
* ***Medicine, Electronic Nose, Security, and Loan Applications***: These are some applications that are in their proof-of-concept stage, with the acception of a neural network that will decide whether or not to grant a loan, something that has already been used more successfully than many humans.
* ***Miscellaneous Applications***: These are some very interesting (albeit at times a little absurd) applications of neural networks.

**Homework for chapter 2 (tutorial)**

**Paper sheet Homework**

**1.1** Given two vectors X and Y, please give as many as possible ways for measuring the distance and/or similarity between X and Y.

* Euclidean Distance: 
* Manhattan Distance: 
* Chebyshev Distance: 
* Minkowski Distance:  ( is a variable parameter)
* Cosine: 
* Jaccard similarity coefficient:
* Correlation coefficient and Correlation distance:

**1.2** Formulate the probability density function (pdf) of a uniformly distributed random variable ranged in [0,1], and derive from it the cdf; formulate the pdf of a uniformly distributed random variable ranged in [5,10], and derive from it the cdf.

We assume:, so  .

And , so 

**1.3** Formulate the pdf f(x) of a Gaussian distributed random variable x with the means of 5 and standard deviation of 10. Calculate the derivative of the pdf with respect to the variable x.



**1.5** Calculate the eigenvalues and eigenvectors of a symmetric 2x2 matrix C, where C=[1 2;2 1].

Eigenvalues= -1 eigenvectors=

Eigenvalues= 3 eigenvectors=

**Homework for chapter 3 (perceptron)**

**Paper sheet homework**

**3.1** Consider a linear machine with discriminant functions. Show that the decision regions are convex by showing that if ****and****then  if .

The fact:





For any  and , the above imply



So we find that the decision regions are convex.

**3.2** Let the d components of x be either 0 or 1. Suppose we assign x to  if the number of non-zero components of x is odd, and to  otherwise. (This is called the d-bit parity problem.) Show that this dichotomy is not linearly separable if d > 1.

Suppose that there exists a linear discriminant function:



Such that



Now, we consider b=0, we suppose. This pattern is in  and hence. Then we consider b =1, we suppose



and thus

 for any

Next consider b=2, for instance



this pattern is in , so .

Now, we can find:

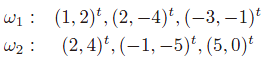


The first two of these imply



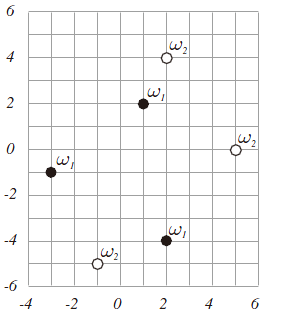
Which contradicts the third equation. Thus our premise is false, and we have proven that this dichotomy is not linearly separable if d > 1.

**3.3** Consider the following six data points:



Are they linearly separable? If yes, try to learn a perceptron for discrimination of the two classes.

The six points are shown in the figure, labeled by their category membership.



By inspecting the plot of the points, we see that ω1 and ω2 are not linearly separable, that is, there is no line that can separate all the ω1 points from all the ω2 points.

**3.4** State the following two generally used schemes for using multiple binary classifiers to solve a multicategory problem: one-against-rest and one-against-one.

One-against-rest: reduce the problem to c-1 two-class problem, where theth problem is solved bya linear discriminant function that separates points assigned to  from those not assigned to.

One-against-one: use c(c-1)/2 linear discriminant function, one for every pair of classes.

**Computer homework**

**3.5** the data is in the following table：



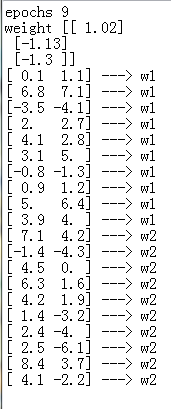
Write a program to implement the Perceptron algorithm.

(a) Starting with a = 0, apply our program to the training data fromand. Note the number of iterations required for convergence.

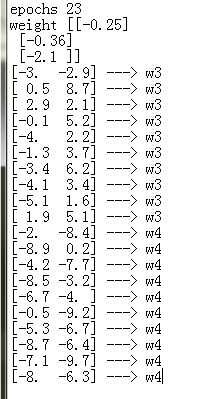
(b) Apply our program toand . Again, note the number of iterations required for convergence.

(c) Explain the difference between the iterations required in the two cases.

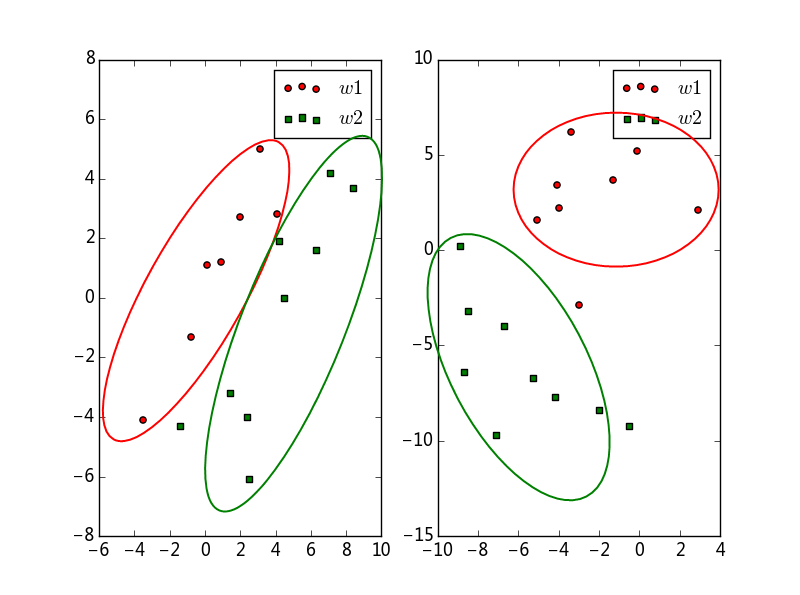
1. When we start with a=0 forand, we finally can get a = [1.02,-1.03], b = -1.3. We also can find our algorithm would be convergent after about 9 iterations. The result as shown down.



1. When we start with a=0 forand, we finally can get a = [-0.25.-0.36], b = -2.1. We also can find our algorithm would be convergent after about 23 iterations. The result as shown down.



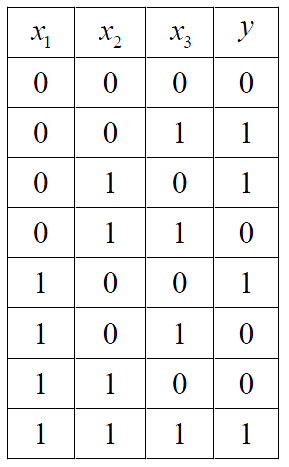
1. According to the scatter figure, we can find the distribution ofandis more tight, and the distribution of andis more scattering. So we need less iteration can achieve convergence for and, and 、 needs more.



**Homework for chapter 4 (MLP)**

**Paper sheet homework**

**4.1** Implementation of a 3D XOR problem. Here we have a three-dimensional XOR problem, in which the logic output y is an XOR operation on the three logic inputs x1, x2, x3. The relation between the inputs and the output can be stated in the following table:



Now the questions are:

1. is it available to implement such an logic operation with a perceptron?
2. is it available to implement such a logic operation with an MLP with only one hidden layer?
3. design an MLP with only one hidden layer to implement such a logic operation, including the structure of the MLP, and the parameters of the MLP.
4. No. This problem is not linearly separable, and a perceptron only can solve a linearly separable problem.
5. An MLP with only one hidden layer can solve this problem.
6. the MLP is shown down:



**4.2** Show that if the activation function of the hidden units is linear, a three-layer network is equivalent to a two-layer one. Explain why, therefore, that a three-layer network with linear hidden units cannot solve a non-linearly separable problem such as XOR or n-bit parity.

Now, we suppose -th neurons’ status value is , the -th neurons’ activation value is . If f(\*) is linear function, ,  and  is linear, as well as  and . So the whole net would become a linear classifier.

**4.3** Consider a standard three-layer backpropagation net with d input units, H n hidden units, c output units. (a) How many weights are in the net? (b) draw the relation between the number of weights and the number of hidden neurons.

(a) 

(b)



**4.4** Express the derivative of a sigmoid in terms of the sigmoid itself in the following two cases (for positive constants a and b):

(a) A sigmoid that is purely positive: 

(b) An anti-symmetric sigmoid: 

(a)

(b)****

**Computer homework**

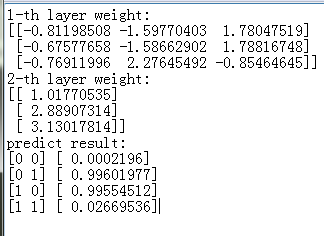
**4.6** Write a backpropagation program for a 2-2-1 network with bias to solve the XOR problem. Show the input-to-hidden weights and analyze the function of each hidden unit.

According the table:

|  |  |
| --- | --- |
| Input\_data | Output\_target |
| [0,0] | 0 |
| [0,1] | 1 |
| [1,0] | 1 |
| [1,1] | 0 |

We design the 2-2-1 MLP, and we add a bias in input layer and hidden layer. And we set the learning rate is 0.2, 5000 iterations. Finally we output the weights and test the network. As the figure shown, the 1th-layer’s weight is  (*note: we add a bias in input layer and hidden layer, so the matrix is* 3x3), the 2th-layer’weight is .

In our test, we find the real output is  when our input is  , the result is very close expected output . So we think our backpropagation algorithm is effective.



**Homework for chapter 5 (k-means)**

**Paper sheet homework**

1. Let  be *n d*-dimensional samples and Σ be any non-singular *d*-by-*d* matrix. Show that the vectorthat minimizes

****

is the sample mean,.

We have:

****

where we used

****

Since Σ is positive definite, we have



with strict inequality holding if and only if . Thus



Is minimized at , that is, at

****

**Homework for chapter 4 (RBF)**

**Paper sheet homework**

**5.1** Consider an exact solution of the XOR problem using an RBF network with four hidden units, with each radial basis function center being determined by each piece of input data. The four possible input patterns are defined by (0,0), (0,1), (1,0), and (1,1), which represent the cyclically ordered corners of a square.

(a) Construct the interpolation matrixfor the resulting RBF network. Hence, compute the inverse matrix.

(b) Calculate the linear weights of the output layer of the network.

(a) 

(b) 



**Homework for chapter 6 (SVM)**

**Paper sheet homework**

**6.1** Now let us revisit Exclusive OR (XOR) problem. Let the kernel function to be  where  and . Please give the kernel matrix for the four logic input samples for this problem, and write down the corresponding objective function of the dual problem.



So 



And 



So the corresponding objective function of the dual problem is



**6.2** The inner-product kernelis evaluated over a training sample set of size N, yielding the *N*-by-*N* matrix



where . The matrix K is positive in that all of its elements have positive values. Using the similarity transformation



Where  is a diagonal matrix of eigenvalues and Q is a matrix made up of the corresponding eigenvectors, formulate an expression for the inner-product kernelin terms of the eigenvalues and eigenvectors of the matrix K. What conclusions can you draw from this representation?

Since the Gram **** is a square matrix, it can be diagonalized using the similarity transformation:

****

where is a diagonal matrix consisting of the eigenvalues of K and Q is an orthogonal matrix whose columns are the associated eigenvectors. With K being a positive matrix, has nonnegative entries. The inner-product (i.e., Mercer) kernel ****is the *ij*th element of matrix K. Hence,

 (1)

Let denote the *i*th row of matrix Q. We may then rewrite (1) as the inner product

 (2)

Where is the square root of .

By definition, we have

 (3)

Comparing (2) and (3), we deduce that the mapping from the input space to the hidden (feature) space of a support vector machine is described by



**6.3**

(a) prove the unitary invariance property of the inner-product kernel; that is,where Q is a unitary matrix defined by 

(b) Demonstrate that the RBF kernel satisfies this property.

(a), so 

****

(b) Consider first the polynomial machine described by

****

Consider next the RBF network described by the Mercer kernel:

****

Finally, consider the multilayer perceptron described by

****

Thus all three types of the support vector machine, namely, the polynomial machine, RBF network, and MLP, satisfy the unitary invariance property in their own individual ways.

**Computer homework**

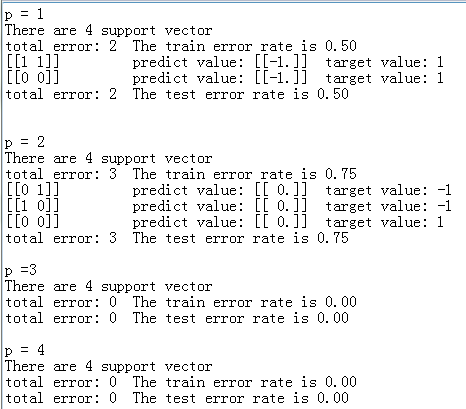
**6.5** The inner-product kernel for a polynomial learning machine used to solve the XOR problem is defined by



What is the minimum value of power *p* for which the XOR problem is solved? Assume that *p* is a positive integer. What is the result of using a value for *p* larger than the minimum?

We apply Platt SMO algorithm to achieve SVM and solve this problem. And we set *p* = 1,2,3,….

Finally, we find the minimum p is 3. According our result as shown down, we can find the error rate is 50%, the [0,0], [1,1] samples is falsely predicted when *p* = 1. And we also can find the error rate is 75% when *p* = 2, [0,0],[1,0],[0,1] have a wrong result. But the predict result will be all right when *p* >=3. For the reason. We think XOR is linear inseparable problem. And quadratic function or linear function only can solve a linear inseparable problem. So we need *p* >=3.



**Homework for chapter 9 (SOM)**

**Paper sheet homework**

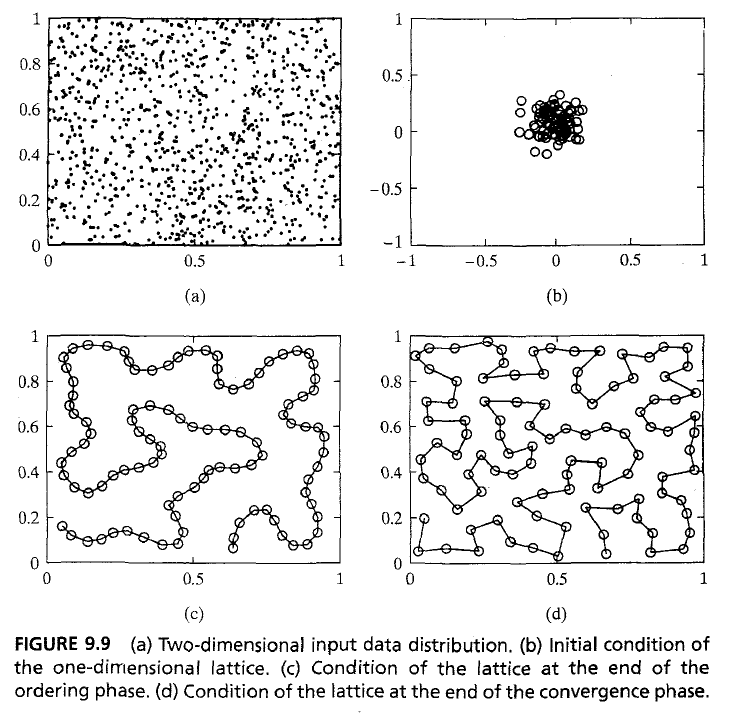
**9.1** It is sometimes said that the SOM algorithm preserves the topological relationships that exist in the input space. Strictly speaking, this property can be guaranteed only for an input space of equal or lower dimensionality than that of the neural lattice. Discuss the validity of this statement.

Consider the Peano curve shown in part (d) of Fig.1 of the text. This particular self-organizing feature map pertains to a one-dimensional lattice fed with a two-dimensional input. We see that (counting from left to right) neuron 14, say, is quite close to neuron 97. It is therefore possible for a large enough input perturbation to make neuron 14 jump into the neighborhood of neuron 97, or vice versa. If this change were to happen, the topological preserving property of the SOM algorithm would no longer hold.

For a more convincing demonstration, consider a higher-dimensional, namely, three-dimensional input structure mapped onto a two-dimensional lattice of 10-by-10 neurons. The network is trained with an input consisting of 8 Gaussian clouds with unit variance but different centers. The centers are located at the points (0,0,0,...,0), (4,0,0,...,0), (4,4,0,...,0), (0,4,0,...,0),(0,0,4,...,0), (4,0,4, ...,0), (4,4,4, ..., 0), and (0,4,4, ...,0). The clouds occupy the 8 corners of a cube as shown in Fig.2a. The resulting labeled feature map computed by the SOM algorithm is shown in Fig.2b. Although each of the classes is grouped together in the map, the planar feature map fails to capture the complete topology of the input space. In particular, we observe that class 6 is adjacent to class 2 in the input space, but is not adjacent to it in the feature map.

The conclusion to be drawn here is that although the SOM algorithm does perform clustering on the input space, it may not always completely preserve the topology of the input space.

**Fig.1**



**Fig.2**



**Homework for chapter 10 (Hopfield Neural Networks)**

**Paper sheet homework**

**10.2** Please try your best to list optimization problems as many as possible which might be solved by applying Hopfield neural networks.

Neuronal network such as Hopfield Network have diverse applications. Generally they are used for:

--**Associative memories**: the network is able to memorize some states, patterns.

--**Combinatorial optimization**: if the problem is correctly modelled , the network can give some minimum and some solution but can rarely find the optimal solution.

