Assignment 2
Backpropagation and Support Vector Machines
Neural Networks and Learning Systems

January 23, 2013

In this assignment you will try out supervised learning in the form of classification using neural networks, error backpropagation and support vector machines (SVMs). You will work with Matlab. Some of the necessary code are provided and some have to be implemented by you.

Written Report

To pass this assignment you have to make a written report. The aim of this report is to convince us that you have understood the finer points of the backprop algorithm and SVM classifiers. Only include the plots you need to make your points.

Mail the report and the code (in .m format) to Thobias Romu and Anette Karlsson: *thobias.romu@liu.se, anette.k.karlsson@liu.se*

Max two persons per report. Include your names and email-addresses in the report! Good Luck!

Backpropagation Networks

The first part of this assignment deals with backpropagation networks. We have supplied you with a code shell to make it easier to get started with the learning algorithm. First of you should familiarize yourself with the datasets you will be using. Then you will have to implement the update rules of the network. Once done you will be able to train you network and analyse the results!

The Data Sets

Download the code and the data for the lab from:
`http://www.imt.liu.se/edu/courses/TBMI26/pdfs/assignment2.zip`
Then extract the contents and change the current path of matlab using the GUI or the cd() command.

Load the datasets and look at the variables using:

```
>> load lab_data
>> whos
```

The data sets are (X1,X2,X3), (Xt1,Xt2,Xt3) and (Xe1,Xe2,Xe3). The correct classifications for X* and Xt* are in the sets (D1,D2,D3) and (Dt1,Dt2,Dt3) respectively. The X*

data set is the training data. The Xt* data set is the test (or validation) data. The data set Xe* contains the complete domain and is good for visualization of the classification boundaries. The data matrices have the dimensions $dim_{in} \times N$, where dim_{in} is the number of data dimensions and N is the number of samples. The classification label matrices have the dimensions $dim_{out} \times N$, where dim_{out} is the number of possible classification classes for the respective case.

You can visualize the 3 datasets using plot() and/or plotCase().

Questions

1. What is the reason for having the test/validation data?
2. Consider the classification problem for each case. What are the difficulties?
3. Regarding the label matrices. How are they constructed? What are the max and min values? Why not +1 and -1? (Hint: Consider the output range of the sigmoid activation function tanh() ...)

The Implementation

Load, read through and understand the provided matlab code skeleton for neural network classification using back propagation.

```
>> edit backprop_singlelayer.m
```

and

```
>> edit backprop_network.m
```

Tasks

1. Comment every line in evalNet().
2. Implement the function normData().
3. Implement the missing code for a network with a single linear layer. Describe and comment your code!
4. Implement the missing code for a network with one hidden non-linear layer and one linear output layer. Describe and comment your code!

Questions

1. Why can it be important to equalize the variance of the different dimensions of the data? Hint: How will scale differences in the data affect the error and equalization.
2. Why should we normalize the test data and the evaluation data with the mean and the variance of the training data?

3. Describe the purpose of the Learning Rate (variable eta in the code). Name two problems with having only one constant learning rate. How does the gradient length interact with the learning rate in the weight update step?

4. Why do we allocate extra dimensions to the weight matrices during the initialization?

The Training

1. For every classification case (X1,X2,X3) train the two implemented networks. Try different learning rates and different number of epochs.

   Questions

   What is the best performance?

   How do you define "Best Performance"?

   Is this expected? Why (not)?

2. For case X3, try to train a non-linear network with 40 hidden nodes with just 10 random training samples (be generous with the number of iterations). Try varying the size of the training data.

   Questions

   What is the training performance?

   What is the generalization performance?

   What do you call this kind of phenomenon?

   Include relevant training plots. Example code:

```
>> [dummy,i]=sort(rand(1000,1),1,'ascend'); %From 1000 samples ...
>> i=i(1:10); %... we choose 10 at random
>> [a, b, c] = backprop_network(X3(:,i), D3(:,i), Xt3, Dt3, Xe3);
```

Optical Recognition of Handwritten Digits

We are now going to test the complete network on an OCR experiment. Download the files 'optdigits.tra' and 'optdigits.tes' from:
`http://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits`
Import the data using the function:

```
>> [X, Xt, D, Dt] = importOptDigits();
```

The two files have to be in the same folder as the function. To test the network, call the network function like this:

```
>> backprop_network(X,D,Xt,Dt,[]);
```

Questions
Describe the data.
What is the training performance? Relate to the previous results.
The data has already been preprocessed, in what way?
The preprocessing has several benefits both computational and classification wise. Why do we get these effects from this type of preprocessing?
You might also have noticed that the data normalization is turned off for this task, why? Hint: Look at the standard deviation and means of the training data.
Try running the same settings several times, how important is the initialization?

Support Vector Machines

Now we will shift our focus to Support Vector Machines (SVMs). We have provided you with two implementations of the SVM classifier, these are found in 'SVM1.m' and 'SVM2.m', you do not need to look at or understand the code of these two implementations. You can try these out using the script found in 'SVMopt.m'.

In the report we want you to explain what a large margin classifier is and what a support vector is. We also want you to explain what the main weakness of the classic SVM is, as defined during lecture 2. This definition is used in 'SVM1.m'. Then we want you to go into the literature and try to explain how we can make the SVM classifier useful by adding a parameter and modifying the definition slightly. To prove your point, create a scenario where SVM1 will fail but SVM2 will not.

Also, are there any tricks we can use to make SVM non-linear? And, state one advantage with SVM over backprop. Backprop over SVM?