SCHOOL *of* ELECTRICAL ENGINEERING & COMPUTER SCIENCE
FACULTY *of* ENGINEERING & BUILT ENVIRONMENT
The UNIVERSITY *of* NEWCASTLE

Comp3330/Comp6380 Machine Intelligence, Semester 1, 2014

Homework Assignment 1: ANNs and SVMs

v14.3

Deadline: Week 7 (Monday 14.4.2014, 23:59)

Maximum possible marks: 10

Description

The main part for marking this assignment is a report and the quality of the experimental results. The recommended length of the report is: about 4-10 pages for Comp3330 students, and about 6-12 pages for Comp6380 students. Include all files in your submission that are required for verifying your results. Aim at providing quality results and describe and discuss them clearly and concisely in your report following instruction of the individual questions below.

Be prepared that depending on your architecture training the ANNs might require some time. We recommend using pyBrain or a similar simulator. Alternatively you could also implement your own neural network e.g. in Java/C++. Although you were briefly introduced to pyBrain in the lab sessions it is expected that you are able to acquire the necessary details how to use the software or programming language of your choice from relevant on-line help or literature. Plot error curves that indicate convergence times (how many iterations did it take?). For demonstrating how well your trained ANN generalises you can visualise the results of your tests (you can submit several plots from different networks or different training schemes) or you may consider suitable basic statistical measures. Always discuss your results and highlight the most important outcomes.

This assignment can be done in teamwork with other students from this class (1-3 people per team) and we encourage you to do this. Best you include a statement agreed by all team members about who contributed what. Any additional help that you use also has to be explicitly acknowledged in your submission.

Please submit your assignment electronically via the assignment section in blackboard. Include all relevant software, results, and data. Please let us know any questions or if anything requires further clarification.

Q1 Hyperplane Geometry [1 mark]

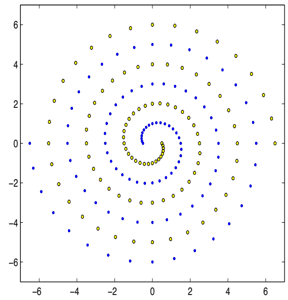
a) Give the equation for a 2-dimensional hyperplane in $\mathbb{R}^3$ which is orthogonal to the vector $w = (6, 3, 2)^T$ and has distance $d = 8$ from the origin.

b) Calculate the distance of the point $p = (10, 1, 5)$ from the above hyperplane.

Q2 Variations of the Two-Spiral Task [3 marks]

Perform an experimental study on the following variations of the two-spiral task:

a) (ANN training): Start with the "original dataset" of Lang and Witbrock (1988) with 194 training points (see Figure below). How fast and how well can you solve this task using a feed-forward NN ? (dataset will be supplied in blackboard) [10% of Q2 mark]



b) (ANN vs. SVM): Generate your own variation of the 2-spiral task. Then solve the associated classification task using ANNs and discuss your approach and solution in comparison to a). [20% of Q2 mark]

c) (ANN vs. SVM): Compare ANNs and SVMs on solving the two classification tasks in a) and b). [30% of Q2 mark]

d) (5-spiral task): Generate a 5-spiral data set. Show and discuss how well your task can be solved with a suitable ANN or SVM. [40% of Q2 mark]

For each subquestion try out different architectures, parameters, and methods. Compare and discuss their performance (speed, generalisation). It is recommended that you focus for each part of your experiments on *about two* different aspects that you investigate in more detail (this could be e.g. variation of the step size, number of hidden layers/units, use of momentum, different kernel parameters in SVMs, ...). The performance of the solution can be evaluated by visual inspection of a generalisation test applied to all pixels of a section of the (x, y)-plane (that for the 2-spiral data should result in two intertwined spiral shaped regions).

A background paper with literature links, description of the data and some hints about successful network architectures is (Chalup and Wiklendt, 2007).

Q3 Autoencoder [3 marks]

Generate a dataset that uses sparse coding to encode the 26 letters of the alphabet:

A = (1,0,0,...,0)
B = (0,1,0,...,0)
...
Z = (0,0,0,..., 1)

Train a 26-H-26 multilayer perceptron (i.e. 26 input units, a hidden layer of H units and an output layer of 26 units) on the identity function that maps 0 to 0, 1 to 1, ..., 26 to 26.

Part I: Determine experimentally what is the minimal number of hidden units, H, required for training the network successfully (Hint: Check chapter 4 of the book (Mitchell, 1997)). Try different training algorithms.

Part II: Conduct training experiments using 26-H1-H2-26 ANNs with two equally sized hidden layers. Determine experimentally what is the minimal number of hidden units in H1 and H2 required for training the network successfully. How does it compare to the above experiments with your 26-H-26 multilayer perceptron?

In your report describe what you did in the experiments and what was the outcome. Finally discuss what role the hidden layers plays in this experiment and what role this type of network could possibly play in real applications.

Q4 Time series classification [3 marks]

The Synthetic Control Chart Time Series data set comprises 600 time series. Each of them is an example of a control charts with 60 values over time. There are six classes of time series:

1. 1-100 Normal
2. 101-200 Cyclic
3. 201-300 Increasing trend
4. 301-400 Decreasing trend
5. 401-500 Upward shift
6. 501-600 Downward shift.

The data is available at the UCI KDD Archive

http://kdd.ics.uci.edu/databases/synthetic_control/synthetic_control.html

Your task is to submit the most successful classifier you can create, and document the process of researching and creating this classifier. I.e. the aim is to have a function where the input is one of the sequences or a part of it and the output is the correct class number (1-6). For solving this you can train either a SVM or a Neural Network, or some combination (in this case please provide code to load and run it on another dataset).

Q4+ Time series prediction [additional challenge question]

A time series can be described as a series of values $a_1, a_2, ..., a_n$. Training for a time series prediction task can be set-up by using a time-window of values as inputs and the next value as output. E.g. if the window size is $k = 3$ then the inputs would be a_i, a_{i+1}, a_{i+2} with associated output a_{i+3} for $i = 1, ..., n-3$. ANNs and SVMs can be trained on this task. The input layer would have k units and the output would be 1 unit. Recurrent Neural Networks (RNNs) can also be used. With RNNs it is also possible to do a one-step look-ahead prediction task where the input is just one value (i.e. $k = 1$) to predict the next value in the series.

Please investigate how you can use RNNs, time window ANNs and SVMs to solve the prediction task using the Synthetic Control Chart Time Series data . Explain and critically discuss your experimental results.

Note: This is an additional challenge question that is not compulsory. It is for those who like an additional challenge. You can obtain a maximum of 3 bonus marks for this question. However, in total you can only obtain 10 marks for HWA 1. The bonus marks would be used to replace missing marks in Q1-Q4 of your submission of HWA 1.

Q4++ Time series clustering [some additional information]

Another paradigm that can be addressed with timeseries, apart from classification and prediction, is *clustering*, which is a form of unsupervised learning. One of the students from the COMP3330 course a couple of years ago has been working on this topic for his Bachelors/Honours thesis that he did in our group. He used the above data set and ran some clustering experiments using support vector clustering with some special kernels. A preprint has just been released in the journal *Pattern Recognition Letters* by Elsevier:

Benedikt Boecking, Stephan K. Chalup, Detlef Seese, Aaron S.W. Wong. Support Vector Clustering of Time Series Data with Alignment Kernels. *Pattern Recognition Letters*, 2014.

You can find and download the paper from the Science Direct Database by searching e.g. for one of the author names. Science Direct you can access via Newcat or directly at

`http://library.newcastle.edu.au/record=e1000005~S16`

The details of this clustering algorithm are beyond the scope of our course but it may be useful for you to know that there are different paradigms that can be applied to the same dataset.

Note

To save SVMs and Neural Networks for submission, use the following code:

Listing 1: Saving A Trained Neural Network

```
import pickle
pickle.dump(neural_net, open('myneuralnet','w'))
```

Listing 2: Saving A Support Vector Machine Model

```
from svm_util import svm_save_model
svm_save_model('mysvm',trained_svm)
```

Marks will be awarded for the performance of the classifier, evidence of researching better solutions for the classifier, and evidence of understanding the training process and the effects of the various training parameters.

Literature

S. K. Chalup, and L. Wiklendt. Variations of the Two-Spiral Task. *Connection Science 19*(2), pp. 183-199, June 2007.
Available at http://hdl.handle.net/1959.13/808886

K. J. Lang and M. J. Witbrock. Learning to tell two spirals apart. In: Touretzky, D., Hinton, G., Sejnowski, T. (Eds.), *Proceedings 1988 Connectionist Models Summer School*. Morgan Kaufmann, Los Altos, CA, pp. 52–59, 1988.

T. Mitchell. Machine Learning, McGraw Hill, 1997.