CSC2515 Assignment 1
Due: Oct 8 2008, 1.00pm at START of class

Geoffrey Hinton

September 30, 2008

Late assignments will have 25% subtracted from the total out of which they are graded for each day or part of a day that they are late. They will be more than one day late if they are not slipped under my office door before 1.00pm the next day.

1 Generalization and Model Complexity (3%)

This question asks you to show your general understanding of underfitting and overfitting as they relate to model complexity and training set size. Consider a continuous domain and a smooth joint distribution over inputs and outputs, so that no test or training case is ever duplicated exactly.

1. For a fixed training set size, sketch a graph of the typical behaviour of training error rate versus model complexity in a learning system. Add to this graph a curve showing the typical behaviour of the corresponding test error rate (for an infinite test set drawn independently from the same joint distribution as the training set) versus model complexity, on the same axes. Mark a vertical line showing where you think the most complex model your data supports is; chose your horizontal range so that this line is neither on the extreme left nor on the extreme right. Mark a horizontal line showing the Bayes error. Indicate on your vertical axes where zero error is and draw your graphs with increasing error upwards and increasing complexity rightwards.

2. For a fixed model complexity, sketch a graph of the typical behaviour of training error rate versus training set size in a learning system. Add to this graph a curve showing the typical behaviour of test error rate (again on an iid infinite test set) versus training set size, on the same axes. Mark a horizontal line showing the Bayes error. Indicate on your vertical axes where zero error is and draw your graphs with increasing error upwards and increasing training set size rightwards.

3. For a fixed range of model complexity (from very simple to very complex), sketch a graph of training set size versus the model complexity which achieves the best test performance (on an iid infinite test set).

2 Mystery Matlab program (4%)

Study the Matlab program shown below. Also, try running it. It is available as assoc.m on the course webpage under assignments.

```
numCases  = 8;   \%must be less than numIn
numIn     = 10;
numOut    = 5;

inputs  = rand(numCases, numIn);  \%each row is a random input vector
targets = rand(numCases, numOut); \%each row is a random target vector
lateral = eye(numIn);     \% lateral connections start at the identity
inOut   = rand(numIn,numOut);  \% initializes input-output connections

for caseNum = 1:numCases,
      input     = inputs(caseNum,:);
      novelPart = input*lateral;
      novelty   = norm(novelPart)^2;
      output    = input*inOut;
      lateral =  lateral - (novelPart' * novelPart)/novelty;
      inOut   =  inOut + novelPart' * (targets(caseNum,:)-output)/novelty;
      eig(lateral) \% prints the eigenvalues of lateral
end

outputs = inputs * inOut;
targets
outputs
```

Write one paragraph describing what the program achieves.

Write another paragraph describing as clearly as you can how the program manages to achieve what it achieves.

3 Digit Classification Using Backpropagation (8%)

Invoke matlab and load ONE of the files
`http://www.cs.toronto.edu/~hinton/csc2515/matlab/assign1v7.mat`
`http://www.cs.toronto.edu/~hinton/csc2515/matlab/assign1v7point3.mat`
`http://www.cs.toronto.edu/~hinton/csc2515/matlab/assign1v6.mat`
The files are for different versions of matlab. assign1v7.mat is for matlab version 7.0 to 7.2.

You will find that you now have four matrices. Each row of **traindata** represents an image of a handwritten 16×16 digit. There are 1000 rows which represent 100 ones followed by 100 twos etc. **testdata** is the same except it has 1000 rows for each digit class. Each row of **traintargets** contains nine 0's and one 1. The 1 is at the position that corresponds to the class of the digit. The class zero is represented by a 1 in the tenth position. Use your generalization abilities to figure out what is in **testtargets**.

Write a matlab program that uses backpropagation to learn a classifier for the digits. Use softmax output units, and a single hidden layer of logistic units. Do not use any regularization. (**Clarification:** Divide the training data into a training set and a validation set.) Train the network by computing the gradient on the whole of your training set for each update of the weights. As you train, print out the error rate on your training set and validation sets. Use these error rates to choose a sensible learning rate by hand. Do not use momentum or adaptive learning rates on each connection.

Using only the training data determine a good value for the number of hidden units.

1. Explain why it is sensible to make the training set small and the test set big when you are evaluating the generalization performance of a learning procedure.

2. Explain precisely how you arrived at a good value for the number of hidden units. Also say what number you chose and what test error rate you achieved.

3. Hand in a copy of the matlab code you wrote for backpropagation and also a copy of the additional code you used for arriving at a good value of the number of hidden units.