UNIVERSITY OF PRETORIA
DEPARTMENT OF COMPUTER SCIENCE

COS314: Project 3 (Neural Networks)

Due: Thursday 28 May 2015, 20h00

For this project you will implement a feedforward neural network to predict whether whether a chunk of text is English or Afrikaans. The neural network should be trained with the backpropagation learning algorithm using gradient descent. You must program the network and training algorithm yourself in either Java or C++. You will need to report on your implementation details, run experiments and report on your findings. This is an individual assignment.

1 The Language Recognition Problem

The classification problem to solve using a neural network is to identify a chunk of text as either English or Afrikaans. For each chunk of input data (a file in plain text format), you should determine the frequencies of each of the 26 letters of the alphabet in the text (i.e. what proportion of the characters in the document are 'a' or 'A', what proportion are 'b' or 'B', etc.). These 26 values will be the input of your neural network.

Your network should consist of an input layer, one hidden layer, and an output layer. For this problem there will be 26 inputs and 2 outputs: the first output for English (1/0), the second output for Afrikaans (1/0). You must decide on the number of hidden nodes (see Experimentation section). Use the sigmoid activation function for hidden and output nodes.

2 Dataset

You will need to create your own training and generalisation datasets. The UP website has plenty of text in both languages for you to use. Make sure you have enough data (and that each chunk of text is big enough) to properly train your network. Perform the necessary pre-processing on the data to be in a format acceptable for neural network training. Your pre-processing should be such that the training process is optimized. Describe the details of your dataset creation in your report:

- Specify the number of chunks of text used, the size of each chunk of text and the language.
- Describe how the frequencies of characters were calculated.
- Describe how you handled any special characters.
- Describe the format used to store your data patterns in the data file.

- Describe exactly how you have pre-processed the data, and provide motivations.

You may choose to extract other language features. If you do, then describe these in your report.

3 Backpropagation Learning Algorithm

Find below a pseudocode algorithm to train a feedforward neural network that consists of an input layer, one hidden layer, and an output layer, using backpropagation and gradient descent. Note that this algorithm implements stochastic learning, and not batch learning.

1. Initialize all the weights (including the threshold values) to random values in the range $[-\frac{1}{\sqrt{fanin}}, \frac{1}{\sqrt{fanin}}]$, where $fanin$ is the number of weights leading to the neuron.
2. Initialize values for η (the learning rate), α (the momentum), $\xi = 0$ (the epoch counter) and ξ_{max} (the maximum number of epochs).
3. Repeat until the maximum number of epochs ($\xi = \xi_{max}$)
   (a) Set the training accuracy to zero ($A_T = 0$)
   (b) Increment the epoch counter (ξ++)
   (c) For each pattern in the training set D_T,
      i. Compute the net input, net_{y_j}, to each hidden unit.
      ii. Compute the activation, y_j, of each hidden unit (using the sigmoid activation function).
      iii. Compute the net input, net_{o_k}, to each output unit.
      iv. Compute the activation, o_k, of each output unit (using the sigmoid activation function).
      v. Determine if the actual output, a_k, should be 0 or 1, as follows: If $o_k \geq 0.7$, then let $a_k = 1$, meaning that the language recognized is that represented by the k-th output unit. If $o_k \leq 0.3$, then $a_k = 0$, meaning the language is not the one represented by the k-th output unit. For outputs between 0.3 and 0.7, the network is uncertain about the classification, and you record a classification error for that pattern.
      vi. Determine if the target output has been correctly predicted. Let $accuracy = 1$ if the target is correctly predicted (if $t_k = a_k$ for all output units, then the target is correctly predicted); otherwise, $accuracy = 0$.
      vii. $A_T += accuracy$
      viii. Calculate the error signal for each output

$$\delta_{o_k} = -(t_k - o_k)(1 - o_k)o_k$$

      ix. Calculate the error signal for each hidden unit:

$$\delta_{y_j} = \sum_{k=1}^{K} \delta_{o_k} w_{kj}(1 - y_j)y_j$$

      where w_{kj} is the weight between output unit k and hidden unit j.
      x. Calculate the new weight values for the hidden-to-output weights

$$\Delta w_{kj} = -\eta \delta_{o_k} y_j + \alpha \Delta w_{kj}$$

$$w_{kj} += \Delta w_{kj}$$

xi. Calculate the new weight values for the weights between hidden neuron j and input neuron i

$$\Delta v_{ji} = -\eta \delta_{y_j} z_i + \alpha \Delta v_{ji}$$

$$v_{ji} += \Delta v_{ji}$$

(d) Calculate the percentage correctly classified training patterns as $A_T = A_T / P_T * 100$, where P_T is the total number of patterns in the training set.

(e) Set the generalization accuracy to zero ($A_G = 0$)

(f) For each pattern in the generalization set D_G, and using the neural network as adjusted in the epoch of training above:

i. Compute the activation (o_k) of each output unit.

ii. Determine if the target has been correctly predicted. Let $accuracy = 1$ if the target is correctly predicted (if $t_k = a_k$ for all output units, then the target is correctly predicted); otherwise, $accuracy = 0$.

iii. A_G += $accuracy$

(g) Calculate the generalization accuracy: the percentage of correctly classified patterns in the generalization set, D_G as $A_G = A_G / P_G * 100$, where P_G is the total number of patterns in the generalization set.

(h) Write the epoch number, A_T and A_G to the results file.

4 Experimentation

Part of this project involves experimenting with different neural network settings (the number of hidden units, learning rate and momentum). Since the initial weights are random, conclusions on whether one setting is better than another cannot be made based on only one training session. To make valid conclusions, one should train a neural network using the same settings a number of times (at least 5, but ideally 30). The average performance for these n trials is then used to draw conclusions. The process you should use is as follows:

1. Repeat for n trials:

   (a) Shuffle the pre-processed dataset, and then divide it into a training set (80% of the data) and a generalization set (20%) of the data.

   (b) Train the network using the training set until the stopping condition has been satisfied. This should result in an output file listing the percentages of correctly classified patterns in the training set (A_T) and generalisation set (A_G) for each epoch.

2. Using the output data files, calculate the average training accuracy and average generalisation accuracy of each epoch/generation for all n trials.

Using the average training and generalisation error values over time, convergence graphs can be drawn showing how the percentage of correctly classified patterns hopefully increases over time. Plotting the generalisation convergence graph with the training convergence graph will illustrate where overfitting might be occuring. Be aware that the training and experimentation will take long, so allocate enough time for this. If time becomes a problem, you may use fewer numbers of trials to calculate your averages, but do not use less than 5.

5 Report

You are required to write a report on your project with the content as specified below. Please provide the items to be included in the report in the same order as specified, and under appropriate section headings. You may provide additional sections if necessary. Please note that this document MUST BE in pdf.

Report sections (minimum required):

1. Title page: State your name, student number, and "COS314: Project 3".
2. Dataset: describe the dataset as outlined in Section 2.
3. Experimentation: Through experimentation, determine the best number of hidden units, and the best values for the learning rate and the momentum. Motivate for why you say these values are best by providing tables and graphs showing the average training and generalization accuracy values against epoch number. You must provide a discussion of your results, not just tables of values.
4. Conclusion: Summarize your main findings.

6 Marking guide

The following will be used as a guide when marking your project:

Aspect of project	**Allocation**
Report	40
Dataset and feature extraction code	10
Neural Network code with backpropagation	40
Training and Experimentation code	10

7 Submission procedure

You are required to submit all of your source code. This would include any code used to process your data, neural network code, training algorithm and experimentation code. Submit a compressed archive with all the necessary files to successfully compile and execute your program, as well as your input data files, result files and report. You have to provide a Readme file (in plain text) to explain how to compile and run your code. Name your report `u????????.pdf`, where the question marks are replaced with your student number. If your archive expands to subfolders, then your report and readme file must be in the root folder, and not in any subfolder.