CS5011: Assignment 1 - Artificial Neural Networks

Assignment: A1 - Assignment 1

Deadline: 10 October 2017

Weighting: 20% of module mark

Please note that MMS is the definitive source for deadline and credit details. You are expected to have read and understood all the information in this specification and any accompanying documents at least a week before the deadline. You must contact the lecturer regarding any queries well in advance of the deadline.

---

1 Objective

This practical aims to construct a neural system capable of playing the 'Guess Who?' game.

2 Competencies

- Design, implement, and document an artificial neural network system for classification tasks
- Understand and use a neural network within a AI system.
- Analyse the features of a neural network system.

3 Practical Requirements

3.1 Introduction

'Guess Who?' is a two-player character guessing game[1], each player starts the game with a board of cartoon images of 24 people and their names with all the images standing up. Each player selects a card from a pile of cards containing the same 24 images. Players in turns ask various yes/no questions to eliminate candidates, to test various features of the characters. The player will then eliminate candidates by flipping those images down until all but one is left. The aim is to be the first to guess which card one's opponent has selected.

3.2 The task

For this practical we consider the role of a single player. The user thinks of a character and an agent should guess the character that the user is thinking of by asking a set of yes/no questions to the user. A neural system is to be constructed to play the agent using Encog

[1] https://en.wikipedia.org/wiki/Guess_Who%3F

for Java[2]. A subset of characters, features and answers is provided in the file *char.csv*. Please see lecture slides for more details on the spec.

3.3 Part 1

The aim of this stage is to design and train a neural network to learn the classification patterns from a subset of characters for the given dataset. The steps for this task are as follows:

1. Code your input and outputs appropriately.
2. Construct a training table with the appropriate coding for input/output of the dataset provided.
3. Construct a multilayer feedforward neural network, with one hidden layer, and sigmoid activation function. The network should have a fixed number of input units to cater for coded bit patterns representing the features. A fixed number of output units is to provide bit patterns for an index to particular characters.
4. A single layer of hidden units is to provide hyperplanes that partition the input space into various regions for character identification. Decide how many hidden units should be included in the hidden layer.
5. Train the network using backpropagation, setting appropriate parameters such as learning rate, momentum, etc
6. Save the trained network
7. Compute the output from the neural network created above with a set of inputs and compare them with the output of the training set
8. Now consider the current neural network, is the number of hidden units appropriate? Try with different numbers of hidden units, and analyse the results, you may use a pruning method to help with this task.

3.4 Part 2

In this part, we will be using the network constructed in the previous step to enable an agent to play a simplified version of the 'Guess who?' game and test with further features and characters. The system should be able to guess the character given the answers to a set of questions corresponding to the different features. Ensure that the system developed in this part is generic enough to be able to accept any number of questions and features. The steps are as follows:

1. Design a system that asks a set of questions corresponding to the features of the dataset. For example: feature: 'blonde', question: 'Is your character blonde?' Each question should be answered by the user with a yes or no answer.

[2] http://www.heatonresearch.com/encog/

2. Use the saved network in Part 1 to compute the output of the given input pattern.
3. Design a method to map the output from the network for a given input in the training set to a character. Consider the case in which the user enters a pattern that does not exist in the training set. What would the guess be?
4. Show the guess to the user at the end of the set of questions. Would the agent be able to make an early guess? Discuss and implement this feature.
5. In addition, add a new character, and a new feature by extending the training table, without altering the previous one. Repeat steps 1–6 in Part 1 to train a new neural network with appropriate fixed parameters. What are the changes in input/hidden/output units? Use the system for the agent to play the extended 'Guess who?' game.

3.5 Part 3

You may wish to consider one or more of the following extensions:

1. Use, compare and discuss different training approaches.
2. Consider the case in which the answers are yes/no/maybe.
3. Modify your system such that the user can teach the agent about a new character or a new feature on the fly. After a game ask the user whether they would like to add a new character or a new feature. Modify the training table as needed, modify the units in the network, retrain the network, as done in Part 2, and let the user play another game.
4. Modify your system such that two characters should be guessed at once, questions should be of the form: "are you characters blonde?" and answers 'Yes (both), only one, No(none)'. Modify the training table as needed, modify the units in the network, retrain the network, as done in Part 2, and let the user play another game.

4 Code Submission and Running

A jar file is to be submitted for each part of the three parts attempted. Name the jar file for Part 1 as "Learning1.jar" and ensure it runs using:

```
java -jar Learning1.jar [any param]
```

Name the jar file for Part 2 as "Learning2.jar" and ensure it runs using:

```
java -jar Learning2.jar [any param]
```

Name the jar file for Part 3 as "Learning3.jar" and ensure it runs using:

```
java -jar Learning3.jar [any param]
```

Your implementation must compile and run on the School lab Machines.

5 Report

You are required to submit a report describing your submission, with a limit of 3600 words. The report should include in this order:

- A list of the parts implemented and any extension attempted
- If any of the functionalities is only partially working, ensure that this is discussed in your report
- Literature review: a short literature survey on neural networks
- Design: A description and justification for the mechanisms you have implemented as part of your solution. Please discuss the choices of each of the numbered steps for each part of the practical that was attempted ([Part 1: points 1–8][Part 2: points 1–5], etc)
- Examples and Testing: Examples of the main functionalities and your approach to testing
- Evaluation: A critical analysis of the functionalities of your system and what can be improved
- Running: Include clear instructions on how to run your system
- Bibliography: List all the references you cite in your literature review and elsewhere in your report and code.

6 Deliverables

A single ZIP file must be submitted electronically via MMS by the deadline. Submissions in any other format will be rejected.

Your ZIP file should contain:

1. A PDF report as discussed in Section 5
2. One, two or three jars depending on the tasks achieved as discussed in Section 4
3. The source code of your implementation containing any non-standard libraries

7 Assessment Criteria

Marking will follow the guidelines given in the school student handbook (see link in the next section).

The following issues will be considered:

- Adherence to the requirements

- Quality of the solution provided
- Code quality
- Examples
- Insights and analysis demonstrated in the report
- Extension activities completed, if any

Some guidelines are as follows. For a mark up to the band 14-16 you must complete Part 1. For a mark up to the band 17-18 you must complete part 1 and part 2. To obtain marks on the band 19-20, there must be a substantial extension such as those suggested in Part 3 in addition to Part 1 and 2. All parts are to be accompanied by an insightful report and good implementation quality.

8 Policies and Guidelines

8.1 Marking

See the standard mark descriptors in the School Student Handbook

`https://info.cs.st-andrews.ac.uk/student-handbook/learning-teaching/feedback.html#Mark_-Descriptors`

8.2 Lateness Penalty

The standard penalty for late submission applies (Scheme B: 1 mark per 8 hour period, or part thereof):

`https://info.cs.st-andrews.ac.uk/student-handbook/learning-teaching/assessment.html#latenesspenalties`

8.3 Good Academic Practice

The University policy on Good Academic Practice applies:

`https://www.st-andrews.ac.uk/students/rules/academicpractice/`

Alice Toniolo
(a.toniolo@st-andrews.ac.uk)
September 21, 2017