

## Assignment 2

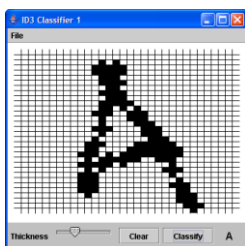
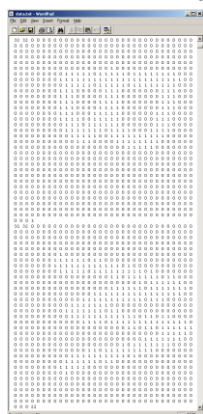
## Assignment 2

Handwritten letter and number recognition

Due 5pm Friday 21 October 2011

- 26 letters: A-Z, encoded 0-25 and 9 numbers: 1-9, encoded 26-34.
- 5605 letter examples in cleaned dataset (~200 of each).
- 400 number examples in number dataset.
- Assume that each letter and number is equally likely (not true in English or other natural languages).
- Notation in code is older form, e.g. *d* for desired output, *y* for network output.
- Encoded using 1 network output per letter – correct output =1, rest = 0.
- That's the desired outputs (targets).

data.txt



bitmap

Class Summary	
<b>Bitmap</b>	A Bitmap holds a matrix of bits (true or false, on or off, 1 or 0).
<b>ClassifiedBitmap</b>	A subclass of Bitmap and extends it by adding a classification (a target class).
<b>Classifier</b>	A null model for classifiers acting on a bitmap.
<b>LetterClassifier</b>	This class extends Classifier and provides some functionality specific to letter recognition.
<b>ID3Classifier</b>	An implementation of a classifier based on ID3/decision trees.
<b>NNClassifier</b>	A neural network handwritten letter recognizer.
<b>TrainClassifier</b>	This program trains a classifier and saves it in a file to be read when used.
<b>EvalClassifier</b>	This program tests a classifier after loading it and the bitmaps.
<b>UseClassifier</b>	This program allows the user to draw on a bitmap.

machl

Class Summary	
<b>BinID3</b>	BinID3 contains methods for inducing a "binary" decision tree using Shannon's information theory.
<b>BinTree</b>	BinTree is a class for storing and generating binary trees (of nodes).
<b>NN1</b>	A basic implementation of a single-layered feedforward neural network and backpropagation learning.

## Applications

- GenerateLetters
- TrainClassifier
- UseClassifier
- EvalClassifier

**1: Comparison of Optimised Single-layer Neural Network and ID3 Decision Tree**

You should complete all of the following tasks (1A-1D). Divide the data set into at least two sets (training and test), adding extra data as necessary. Investigate and optimise the performance of the already implemented single layer neural network. Investigate and optimise the performance of the already implemented ID3 decision tree. Discuss the differences in performance obtained for the optimised Single-layer neural network and the optimised ID3 Decision tree.

**2: Advanced Classifier**

You should complete one of the following tasks (2A-2C). Improve on one of the provided machine learning techniques or use a different technique to design and train an advanced classifier

**3: Refine study**

Any number (0 to 4) of the following tasks (3A-3D) may be completed. All are assigned the same marks. You may attempt any part(s) of this section. Describe the method, your implementation, and the performance of your implementation. Compare the performance of the classifier with and without the task implemented.

**4: Competition**

Enter your best classifier into the class competition to be tested on a new set of data.

**1: Comparison of Optimised Single-layer Neural Network and ID3 Decision Tree (6 marks)**

You should complete all of the following tasks (1A-1D). Divide the data set into at least two sets (training and test), adding extra data as necessary. Investigate and optimise the performance of the already implemented single layer neural network. Investigate and optimise the performance of the already implemented ID3 decision tree. Discuss the differences in performance obtained for the optimised Single-layer neural network and the optimised ID3 Decision tree.

**1A: Data Sets (1 mark)**

Build a series of training and test data sets that can be used for parts 1B and 1C (and for later parts as needed). The data sets should contain both letters and numbers.

**1B: Single-layer Neural Network (2 marks)**

Investigate and optimise the performance of the already implemented single layer neural network. Optimisations include the size of the test/training sets, the learning rate, and the number of iterations of learning.

**1C: ID3 Decision Tree (2 marks)**

Investigate and optimise the performance of the already implemented ID3 decision tree. Optimisations include the size of the test/training sets, and the two thresholds used for deciding when to create a leaf node (proportion and number of samples)

**1D: Comparison (1 mark)**

Discuss the differences in performance obtained for the optimised Single-layer neural network and the optimised ID3 Decision tree.

## What does optimisation mean?

- For a particular technique, there will be parameters that need to be set
- When optimising the technique, you should find the performance of the technique with the parameters set to several different values
- The optimised classifier is the one that has the best performance of those that you have tested
- E.g. Neural network learning rate set to values between 0 and 1.0
- E.g. Decision tree proportion of partition threshold set to values between 0 and 1.0

## What are the Decision Tree parameters?

- The ID3 Decision tree algorithm has been adapted to allow for noise in the data set, with the use of 2 thresholds:
- Proportion of the partition that should match for a leaf node to be created (a value between 0 and 1)
  - 1.0 or 100% in the standard algorithm
- Smallest number of examples for which a branch will be considered (an integer)
  - 1 in the standard algorithm, that is all examples can have their own leaf node

### 2A: Multi-layer neural network (5 marks)

(See Russell and Norvig, p. 731-736.)

Add units to the single-layer neural network to give it two layers of weights.

Optimisations include the number of hidden units.

## What are the Neural Network parameters?

- Standard parameters for neural networks:
- Learning rate (a value between 0 and 1 e.g. 0.1)
- Number of iterations of learning (an integer e.g. 100,000)

### 2: Advanced Classifier (5 marks)

You should complete one of the following tasks (2A-2C).

Improve on one of the provided machine learning techniques or use a different technique to design and train an advanced classifier. You are to modify the appropriate source code provided or write your own code for the technique you have chosen. Evaluate your extension on the same problem. Describe both your implementation and its evaluation.

### 2B: ID3 with pruning (5 marks)

(See Russell and Norvig, p.705-706.)

Use any relevance-based heuristic to remove paths ("prune" decisions) in the decision tree.

Optimisations include the cut-off used for pruning.

**2C: Different Machine Learning Technique (5 marks)**

Implement a classifier with a different machine learning technique and optimise its performance. The technique should not be decision trees or neural networks. One possible technique is a Naive Bayes' Classifier.

Note that you must make relevant use of a machine learning algorithm. It is not sufficient to come up with a smart program.

**3: Refine study (up to 16 marks)**

Any number (0 to 4) of the following tasks (3A-3D) may be completed. All are assigned the same marks. You may attempt any part(s) of this section. Describe the method, your implementation, and the performance of your implementation. Compare the performance of the classifier with and without the task implemented.

**3A: Create an Ensemble of Classifiers (4 marks)**

Make multiple copies of your classifier, and train these copies on random subsets of your training data. Produce an overall prediction in response to a test pattern by combining the results from multiple classifiers via a voting system. See Russell and Norvig, section 18.10 on p. 748-752 for some background.

**3B: Implement a pre-processing technique (4 marks)**

Implement some pre-processing technique that is run on each data instance before it is presented to the classifier for training/testing and which you feel may help the classifier perform better. Ideas in this area may include centring the letter in its 32 \* 32 bitmap square, emphasising curves or straight edges that appear in the instance, or even compressing the letter into a smaller bitmap area (e.g. 8 \* 8). The choice here is up to you.

**3C: Overtraining Prevention (4 marks)**

There are a number of techniques that may be used to ensure the classifier is not overtraining. One option is to partition your training set into two new sets (a training set, and a validation or verification set) and to use this validation set to estimate when to stop training the classifier. Again, the choice is up to you.

**3D: Additional Refinement (4 marks)**

Implement some technical refinement that aims to improve the performance of the classifier and that has not been specified in 3A – 3C.

#### 4: Competition (2 marks)

Enter your best classifier into the class competition to be tested on a new set of data.

- Will be run during the final lecture of the semester (Tuesday 25 October)
- NOTE: make sure that your classifier works as specified; classifiers that do not will not take part in the competition
- Will work as `bitmap.UseClassifier.java`
- Classifiers will be identified by the name you give it
- +1 mark for entering
- +1 mark for top 4 classifiers

## Competition

- All new functionality should be included within the one java file
- Class must belong to the `bitmap` package
- Class must extend the original `bitmap.LetterClassifier`
- You will have access to all java classes and data files that are part of the distributed code
- The competition system will re-construct your classifier instance and test each new letter by calling `Classifier_XX.test(Bitmap map)`
- Make sure that the classifier is generated from the same class definition as you submit

#### REFERENCES (1 mark)

Published literature should be referenced throughout your report, with a reference list provided at the end of your report.

Note that your references should include more than your textbook and that while Wikipedia is a good place to start looking it is not a good reference.

## Competition

- Java file named `Classifier_XX.java`
- Classifier named `Classifier_XX.ser`
- Class should also be named `Classifier_XX.java`
- Name of classifier should be changed from “NN Classifier 1” or “ID3 Classifier 1”
  - Suggested name is `Classifier_XX`, but more inventive names are ok
- `XX` = your student numbers

## Competition

- Due 9am Monday 24 October
- If you submit early and let me know then I can check whether your classifier can be loaded and used

#### INDIVIDUAL WORK (1 mark)

## Marks

- Total available marks is 31
- Maximum mark for assignment is 20
- You should complete all of parts 1A-1D, one of parts 2A-C, any of parts 3A-D, part 4 is optional, a reference list should be provided, and individual work is optional
- Assignment 2 worth 20% of final course mark
- May be beneficial to work in pairs

## Submission

- Submit two files online (<http://submit.itee.uq.edu.au/>)
  - Source code (\*.java into a .jar or .zip)
  - PDF of your report
- Group name should be the student number of the person submitting the assignment
- If you resubmit, please resubmit all relevant files

## Submission

- Extensions will only be granted for documented medical reason or family emergency
- Late submissions will not be marked and will receive 0%

## Submission

- Submit an assignment cover sheet signed by all members of the group in the lecture, at your tutorial, or to 47-308 as soon as possible after submission
  - Provided by the online submission system OR
  - <http://studenthelp.itee.uq.edu.au/assignments/>
- Your mark will not be released until a cover sheet has been received