Algorithm structure

The Artificial Neural Network consists of 1 input layer, 3 hidden layers and 1 output layer. It is able to take in a file that can be read without the need to know the attribute names and labels (thus can be unsupervised and be trained with any set of data that suits binary classification.

The first layer uses ReLu and the rest of the layers use Sigmoid to regulate the weights so they can be flexible for categorical data that isn't binary. For the final layer, the value is computed using sigmoid and then compared to be higher or less than 0.5, which leaves a binary output. This cannot be done using an activation function as they are only used to activate signals based on the weights to predict this output.

For my tests, I am using 5 neurons in hidden layer 1, 5 neurons in hidden layer 2 and 4 neurons in hidden layer 3. This is at a learning rate of 0.5 as this is a perfect ratio for determining change in weights without too much influence. Moreover, the learning rate will not affect the bias and will allow for the data to pass normally due to this being a binary classification algorithm, so no extra steps are taken unlike multi-classification algorithms.

When the data is trained for both algorithms, I use a seed number of 100 and train them by a ratio of 0.8.

The GP calculates the fitness using the accuracy of the tree as it is pruned and crossed over. It uses the crossover methods that swap the branches between two possible solutions and trims the tree for any duplicates. As it uses ID3, it uses that to grow the tree but only through the mutation operator and only a random attribute is used (unlike normal ID3 which chooses the best attribute based on information gain). This allows for the trees to be diverse every generation and if the tree is too big, it is taken out of the generation. Trimming also helps to bounce off the local optima so that the trees can take on new forms and reach a better and more accurate decision tree.

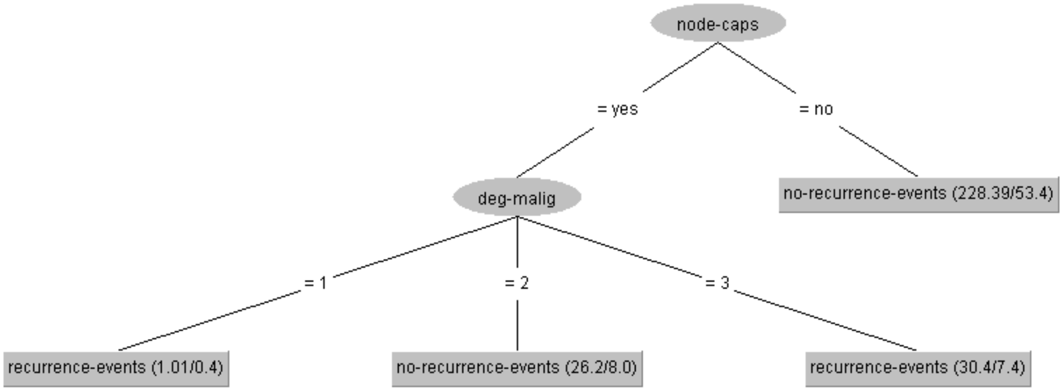
	Accuracy	F-Measure	TimeTaken
ANN	0.4616	0.3801	
GP	0.5128	0.32558	13045
Weka	0.755	0.755	--

As shown, GP takes a considerable longer time than ANN given the circumstances, but will still perform with a relatively better accuracy and F-measure (higher precision). Weka uses the C4.5 algorithm which supports pruning, and hence enhances its accuracy as well compared to the ID3 algorithm used in GP.

Weka Model Run model information

```
=== Run information ===


Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
```

```
Relation:     breast-cancer
Instances:    286
Attributes:   10
              age
              menopause
              tumor-size
              inv-nodes
              node-caps
              deg-malig
              breast
              breast-quad
              irradiat
              Class
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===


J48 pruned tree
------------------


node-caps = yes
|   deg-malig = 1: recurrence-events (1.01/0.4)
|   deg-malig = 2: no-recurrence-events (26.2/8.0)
|   deg-malig = 3: recurrence-events (30.4/7.4)
node-caps = no: no-recurrence-events (228.39/53.4)

Number of Leaves  :    4

Size of the tree :     6



Time taken to build model: 0.04 seconds

=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances       216            75.5245 %

Incorrectly Classified Instances       70            24.4755 %

Kappa statistic                    0.2826

Mean absolute error                  0.3676

Root mean squared error              0.4324

Relative absolute error             87.8635 %

Root relative squared error          94.6093 %

Total Number of Instances            286


=== Detailed Accuracy By Class ===


             TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class

             0.960    0.729    0.757      0.960    0.846      0.339    0.584     0.736     no-recurrence-events

             0.271    0.040    0.742      0.271    0.397      0.339    0.584     0.436     recurrence-events

Weighted Avg.    0.755    0.524    0.752      0.755    0.713      0.339    0.584     0.647


=== Confusion Matrix ===


   a   b   <-- classified as

 193   8 |   a = no-recurrence-events

  62  23 |   b = recurrence-events
```


node-caps
= yes
= no
deg-malig
no-recurrence-events (228.39/53.4)
= 1
= 2
= 3
recurrence-events (1.01/0.4)
no-recurrence-events (26.2/8.0)
recurrence-events (30.4/7.4)