Project spec:

Project spec has changed

I have a NN and GP,

Whatkind of NN, how many layers,

Design:

I have decided to implemtn two different things

Intention to create two methods to compare them against each other and already made models

Testing -> what kind of testing , validation

How am I comparing the output for the two differnet hing

What kind of crossovers and mutations im doing.

Tree vs infix notations

Data structure and notation

Fitness function

Selection and replacement procedures

Process:

Agile dev

Validation – how do I know it is working

This is not a deterministic process

Testing:

Xvalidation

Since very difficult to test, I tested on small programs that you know it should get the right answer

Test plan- code testing, different functions to you

Product

Is it split into good functions, good clean code?

Testing:

Tuning parameters

What could I have done better

Why testing code is hard because it is not deterministic

Design:

I am planning on making two models as described in the product spec

Before starting project:

Create the financial ratios using the data that I have from the data set

GP’s:

Research the most appropriate GP to use.

Usin gregression trees to produce an optimal function

Research data structures

Chose binary trees

Terminal and functional sets

Understand the full process of a genetic program

Made a flow chart to understand the steps I could take .

Define and research the differnet types of genetic operators

What GO’s will I start with

Why am I using these one

Tournament selection

Selecting best ones

Selecting random ones each time – brute force

types of crossover single point subtree crossover

mutation – single point mutation

fitness function – number of hits technique

terminating conditions – why I have chosen thesw.

Reinstate the final aim as was stated initially – to produce a function tthat will predict CF.

Design of tests:

What kind of testing, valdation

ANN’s:

What libarary to use.

What dataset to use.

Determine what type of ANN to use.

For this, to avoid over complexity, feed forward network used.

Determine how many inputs

Determine how many layers and nodes

What activation function to use

What learning algorithm to use

 A sufficiently wide neural network with just a single hidden layer can approximate any (reasonable) function given enough training data.

 shallow networks are very good at memorization, but not so good at generalization.

advantage of multiple layers is that they can learn features at various levels of abstraction

 Multiple layers are much better at generalizing because they learn all the intermediate features between the raw data and the high-level classification.

the wider your network, the longer it will take to train

How am I comparing the outputs