An Artificial Neural Network for the Optimization of Stock Portfolios, with a Genetic Learning Algorithm

Luís Felipe Melo de Miranda

December 03, 2010

Chapter 1

Background and Context

It is natural research should be fueled by potential applications and that, sometimes, the most potent of methods spawn from emulating natural processes. In this project, the objective is a most profitable one and the method is a hybrid algorithm, combining two of the most often emulated natural phenomena.
It is hard to imagine a research topic with a more practical application than predicting stock market fluctuations. Knowing when to buy and when to sell is one of the highest paid jobs out there, and getting a computer to tell you that would yield obvious phenomenal benefits. What truly makes this problem interesting, however, is its endogenous nature–meaning: the predicted solution affects the outcome. The answer is not truth, but the perceived truth. This results in a lot of noise in the time-series.
Such properties lend a very human-like quality to the subject, meaning simple laws and rules should hardly suffice. We then turn to two optimization methods inspired by natural processes.

1.1 Artificial Neural Networks

Details of the brain's functionality are, to this day, shrouded in mystery. However, it is understood that individual nodes (neurons) communicate with each other through edges (synapses); meaning a pulse reaching a node will generate a response which will be sent through the node's edges. That is the basis of the Artificial Neural Network. Since it is not readily understood how the phenomenon at hand works, we create a mesh of nodes which respond to certain inputs in a way we do not readily understand. We then take these nodes through a learning process to calibrate them, and use the learned ANN to predict future results.
In layman's terms, each edge contains a weight, which will magnify or diminish the pulse sent through it. The node will then evaluate a function with all the pulses it received and send it through all its out-going edges. What we optimize are the edges, and since we work with a layered ANN (which is basically a lay-

ered, forward moving graph), and since the function evaluated in each node is a single argument function, which takes the sum of all the pulses it receives as the parameter, the weights between each layer can be represented as matrices.

1.1.1 Learning Method

What occurs then is the optimization of the edges' weights, so as to maximize profit, given past information. Though the methods are many, we have opted for the genetic algorithm. Admitedly, this is a procedure which in no way fits the problem, seen as it is a form of random search in a case where we have a function (albeit a complicated one). Nevertheless, it does allow us to optimize the ANN using the same program, regardless of how its structure may change; i.e. the process does not depend on the structure.

1.2 Genetic Algorithm

The genetic algorithm first generates a multitude of solutions, pseudo-randomly, and then evaluates them. Meiosis occurs next, generating an offspring population which is then evaluated. The best from both survive to go on and mate again. After a set number of iterations, the best-evaluated solution is chosen. In this case, our solutions are lists of matrices, one for each set of edges. Our evaluation hinges on past information, economic indices which form the input of our ANN and the growth of the stock options being followed in the next few days (our output).

Chapter 2

The Program

The program, though its first draft was written in C, was completed in its entirety in Python, for the complexity of the three-star necessities of the algorithm represented a lot of potential bugs in the former. The solutions were indeed more than satisfactory, but in no way unique.

2.1 Tests and Results

Of the many tests conducted, I list the first four here. They followed six different indexes: Microsoft, Oracle, FTSE, NYSE, Petrobras, and Bovespa. There was a learning period of five weeks, and the "enlightened" ANN was applied to the next five weeks, for a total of 40 weeks when taking in account all four tests. They began in January 2004 and ended september of that year. In all cases there were significant profits:

Table 2.1: Percent gains

Jan - Mar	3.9
Mar - May	1.3
May - Jul	3.8
Jul - Sep	2.9

Considering these gains all happened in the course of five weeks, on average this would represent, roughly, a 34 percent profit in the course of a year, far more than the market average.

However, unlike the application of the genetic algorithm to the cutting stock problem, these problem's answers never converged. If the routine was run through twice, the two answers were very likely completely different (NOTE: the answer is the ANN; not how much should be invested in each stock option, but a list of matrices that, given the input information, would tell you how much should be invested). Meaning, though great answers were found, they were most assuredly not optimal, the reason being that the genetic algorithm is unsuited for such a largely-dimensioned problem (the one at hand involved two 30 X 10 matrices and one 10 X 10, meaning each potential answer was composed of 700 integers, a space the genetic algorithm was unsuited to search, given the available machine and time), specially when there is a differentiable function involved, which means smarter descent routines could have been followed.

Chapter 3

Conclusion

The effectiveness of the ANN in optmizing stock portfolios was indeed corroborated by this project, even though the learning procedure was far from optimal. The genetic algorithm is not the best way to calibrate problems of this magnitude, specially when the "profit function" is differentiable. Nonetheless, it is uncanny how in the many tests carried out, all of them produced satisfactory returns, above the market average. The basic principle behind the ANN is to simulate a small brain, to create "thought patterns" able to recognize and inuitively predict growth patterns of a not entirely understood system. In that sense, the ANN taught by the genetic algorithm was a complete success.

Bibliography

[1] Site on ANN http://en.wikipedia.org/wiki/Artificial_neural_network.

[2] Site on ANN http://www.learnartificialneuralnetworks.com/.

[3] Site on ANN http://www.dkriesel.com/en/science/neural_networks.