CMS 2: 155–161 (2005)

DOI: 10.1007/s10287-004-0032-7

Hybrid artificial neural networks for efficient valuation of real options and financial derivatives

Chris Charalambous, Spiros H. Martzoukos

University of Cyprus, Department of Public and Business Administration, 75 Kallipoleos Str., P.O.Box 20537, CY 1678 Nicosia, Cyprus (e-mail: bachris@ucy.ac.cy)

Abstract. A hybrid valuation methodology is proposed and tested for improving the efficiency of contingent claims pricing by combining Artificial Neural Networks (ANN) and conventional parametric option pricing techniques. With one application on financial derivatives and one on real options the method's superiority is demonstrated. The resulting efficiency is instrumental for real time applications.

Keywords: Artificial neural networks, financial derivatives, real options, path-dependency, switching costs

MSC code: 90-08

1 Introduction and the Hybrid ANN approach

Recently, Artificial Neural Networks (ANN) have been used for valuation of traded derivatives (i.e., Hutchinson and Poggio, 1994; see also the review in Lajbcygier, 1999). In this paper we use a hybrid approach that combines parametric with ANN methods, in order to improve the efficiency in real time option pricing. We demonstrate the results with two applications, one in financial derivatives and one in real options. We call the method Hybrid Numerical Option Pricing and ANN (NOP-ANN) and we compare it with the traditional ANN methods and demonstrate the method's superiority.

Neural networks are universal approximations for continuous functions over a compact set (see Cybenko, 1989; Hornik et al., 1989). A three-layer feedforward

Acknowledgements: Both authors are thankful for partial financial support to the HERMES European Center of Excellence on Computational Finance and Economics of the University of Cyprus and a University of Cyprus grant for research in ANNs and Derivatives, and to the anonymous referees for their helpful comments and discussions.

network is commonly used. It consists of an input and an output layer, corresponding to model input and output variables x and y, respectively, as well as a hidden layer. In the first stage, samples of data (x, y) called training data, are generated from simulation or measurement. The neural network is then trained by adjusting its weights so that the network predicted output "best" matches that of training data, the target output. This is done by minimizing some norm of the error function between the predicted output of the neural network and the targeted outputs. In this paper instead of the backpropagation (BP) learning algorithm proposed by Rumelhart et al. (1986), we use instead conjugate gradient algorithms (see for example, Fletcher, 1987) in conjunction with Charalambous (1992) line search. Given the input feature vector x, the output can be computed by,

$$y(x) = v_0 + \sum_{i=1}^{H} v_i z_i, \quad z_i = f(\psi_i), \quad \psi_i = w_{io} + \sum_{j=1}^{N} w_{ij} x_j,$$

where $v_i, i = 1, 2, \ldots, H$, is the weight on the branch connecting the hidden neuron i to the output neuron, v_0 is the threshold weight of output neuron, and z_i is the output of the ith hidden neuron computed and f is the activation function. Similarly, $w_{ij}, i = 1, 2, \ldots, H, j = 0, 1, \ldots, N$, is the weight on branch connecting input j to neuron i (input 0 corresponds to threshold weight). The two most widely used activation functions are the logistic that is bounded in the range (0, 1) and the tansigmoid bounded in $(-1, 1)$. For the testing stage finally, a new set of input-output samples is used.

An innovative approach, the hybrid approach, has been proposed by Watson and Gupta (1996) to reduce the training data needed and to improve the generalization capabilities of a neural network. Now, the target value for a given input vector x, is the difference in the response between that of a *coarse* model and that of a *fine* model. As the terminology might suggest, the "fine" calculations are computationally far more expensive than the "coarse" calculations. The "coarse" alone cannot provide the desired accuracy, and the "fine" would be overly expensive for real time applications. This leads into a smoother input-output relationship, reducing the number of fine model simulations and enhancing effectiveness. The principal object of this paper is to examine the extent to which we can exploit a combination of neural network methodology and parametric option pricing to reduce the computational requirements and price efficiently options in real time situations.

In this work the feedforward neural network structure with one-hidden layer will be used (two-hidden layers did not offer any noteworthy improvement). All input variables are scaled to zero mean and unit variance. We use tansigmoid activation functions for the neurons in the hidden layer. The simple ANN tries to capture the functional relationship between the option price, V_f (fine model) obtained by the numerical option pricing (NOP) techniques and the input variables, while the hybrid neural network tries to capture the functional relationship between the deviation of the option price, V_f (fine model), from that given by the coarse model, V_c, and the input variables.

To be more specific for this hybrid approach we want to find the optimal set of weights and biases, such that combined response of the ANN and the coarse model is as close as possible to the fine model response, by solving the optimization problem:

$$\min_{w,v}\left\{F\left(w,v\right)=\frac{1}{2}\sum_{j=1}^{l}e_j^2\right\}, \tag{1}$$

with l the number of learning samples and e_j the error due to the jth input sample point:

$$e_j=\left(y_{N_j}+V_{c_j}\right)-V_{f_j}. \tag{2}$$

The starting point for the above optimization problem is set to: $v_0=v_1=\ldots=v_H=0$ and w_{ij} to small random numbers, in which case $y_N=0$ and $y=V_c$. As a consequence, it follows that at the optimal solution obtained, (w^*,v^*),

$$F\left(w^*,v^*\right)\leq F\left(w^{(1)},v^{(1)}\right)=\frac{1}{2}\sum_{j=1}^{l}\left(V_{c_j}-V_{f_j}\right)^2. \tag{3}$$

This shows that once the ANN is trained, the overall response in the least squares sense, will always be better than that of the coarse model. In the next section we discuss the two contingent claims examples we will use in pricing.

2 The two examples: Financial and real options

We adopt the extended (with a dividend yield) Black and Scholes (1973) assumptions that a traded asset follows in the risk-neutral probability measure the stochastic process

$$\frac{dS}{S}=(r-\delta_S)dt+\sigma_S dz_S. \tag{4}$$

It has an instantaneous standard deviation $\sigma_S S$, it pays a dividend yield δ_S, r is the continuous riskless interest rate, and dz_S is the increment to the standard Wiener process. For parametric option pricing we implement the multi-dimensional lattice-schemes recommended by Boyle et al. (1989) (adjusted for dividend yields).

In the first application we price a European put option on the average of three assets. Such an option is often embedded in the payoffs of investment accounts like Guaranteed Investment Contracts (GICs), in order to improve payoff patterns. We use the 3-dimensional lattice, and we derive (for the parametric *fine* model) the average price of those provided by a 40-step and a 41-step lattice. For the parametric *coarse* model the price is derived from a 10-step lattice. The difference in computational intensity between the two is by a factor of 357 (option node evaluations). We also use a simple ANN and the NOP-ANN to price these financial options. We allow six variables to take five different values each, thus creating a *training* set of 15625 for each of the *fine* and the *coarse* values. Input variables are the three underlying assets S_1, S_2 and S_3, each with values equally spaced between 85 and 115, the riskless rate r with values between 0.02 and 0.06, all correlations of the assets' continuous rates of return $\rho_{1,2}$, $\rho_{1,3}$ and $\rho_{2,3}$ (equal to each other in order to keep the sampling space to manageable size) between -0.20 and 0.20, and all standard deviations of the assets' continuous rates of return σ_1, σ_2 and σ_3 (again equal to each other) between 0.10 and 0.50. The exercise price is $X = 100$, the two dividend yields δ_1 and δ_2 are fixed at 0.03, and the time to maturity $T = 3.00$. We also create a *test* set of 200 by drawing randomly and independently for the six input variables from uniform distributions in the range of the values used in the *training* set.

Our second application is in the context of real options pricing (see Trigeorgis, 1996). We are interested in the hard problems of partial investment reversibility like the ones introduced by Brennan and Schwartz (1985) and Dixit (1989) in a perpetual horizon for natural resource investment decisions with operating flexibility (early exercise feature) and switching costs (path-dependency inducing feature) and one stochastic asset price. Here though, we assume the harder case of a finite horizon and of two assets. We assume that each underlying asset is perfectly correlated with a traded one, and follows a Geometric Brownian motion process. We seek to value a risky venture that offers the option to operate on the best of two outputs, with the additional option of costly switching between operating and idle states. For our parametric solution we implement a lattice scheme in two-dimensions. This state-space is used to optimize the firm's operations for every possible path of the state-variable and for every possible past operating decision. Our solution is equivalent to solving a multi-stage optimization problem by implementing a forward-backward looking algorithm of exhaustive search. Here the operation can be in an *on* (j) or an *off* (i) state, with costly switching between the two. In the *on* state operations provide the best of two underlying assets (S_1, S_2) for an operating cost X. Switching from state i to j occurs at a switching cost $I^{i\rightarrow j}$ and there is also a similar cost $I^{j\rightarrow i}$. When the state remains the same, cost I equals zero. The claim value is a function of the net cash flows in each state plus the discounted expected value of the claim at the next decision point. The model allows decisions at time zero, at the maturity of the option, and twice in-between, thus the optimization problem with exhaustive search is similar to a 3-stage stochastic programming model. We implement a 24-step 2-D lattice for the *fine* and an 8-step 2-D lattice for the *coarse* model. The difference in computational intensity between the two is by a factor of 1567 (option node evaluations). We allow five variables to take five different values each creating a *training* set of 3125 for each of the *fine* and the *coarse* values. The input variables are the two underlying assets S_1 and S_2, each with values equally spaced between 85 and 115, the standard deviations of the assets' continuous rates of return σ_1 and σ_2 (equal to each other) between 0.10 and 0.50, the switching cost I to get to active mode (from idle) between 5.00 and 25.00, and the switching cost I to get to idle mode (from active) similarly between 5.00 and 25.00. The exercise price is $X = 100$, the riskless rate r is fixed at 0.05, the two dividend yields δ_1 and

Table 1. Comparison between the hybrid NOP-ANN, the simple ANN, and the coarse NOP model

Error Measure	Hybrid NOP-ANN					Simple ANN	Coarse NOP
H	1	2	5	10	20	20	
Example 1							
$mse(e)$x10^4	(1.1) 1.0	(0.79) 0.85	(0.47) 0.45	(0.24) 0.21	(0.16) 0.14	(1.5) 1.0	(1.3) 1.3
$mae(e)$x10^2	(0.84) 0.84	(0.72) 0.75	(0.55) 0.54	(0.38) 0.33	(0.30) 0.27	(0.97) 0.79	(0.90) 0.95
$max(\|e\|)$x10^2	(4.4) 3.2	(3.3) 2.3	(2.4) 2.2	(2.1) 1.9	(2.1) 1.4	(6.1) 2.8	(4.7) 2.7
Example 2							
$mse(e)$x10^4	(3.2) 2.5	(1.1) 0.9	(0.82) 0.74	(0.39) 0.60	(0.23) 0.21	(2.7) 1.8	(6.0) 4.1
$mae(e)$x10^2	(1.4) 1.3	(0.85) 0.79	(0.75) 0.72	(0.47) 0.59	(0.36) 0.35	(1.2) 1.1	(1.8) 1.6
max($\|e\|$)x10^2	(8.1) 6.0	(8.3) 3.4	(7.4) 2.9	(5.0) 2.1	(2.7) 1.6	(2.3) 4.4	(11.4) 8.6

Note: The 1st and 2nd rows in each measure correspond to training and testing sets respectively.

δ_2 are fixed at 0.05, the correlation of the assets' continuous rates of return $\rho_{1,2}$ is fixed at 0.35, and the time to maturity $T = 3.00$. A *test* set of 200 is created with random drawing similarly to the first example. In the next section we discuss the results for both examples, and we suggest future extensions.

3 The numerical results and concluding comments

Table 1 shows the results obtained for both problems. For the hybrid network we considered values for the number H of neurons in the hidden layer ranging from 1 to 20, while for the simple network we considered only the case $H = 20$. The comparison measures are the mse (e): mean square error, the mae (e): mean absolute error, and the max (| e |): maximum absolute error. For all cases, error is defined as the difference between the option price obtained by the fine model, and that obtained by the coarse or any of the two neural networks. The elements in brackets correspond to the results obtained on the training data, and the rest correspond to those obtained on the testing data. The testing error may in some cases be less than the training error, due to the fact that the testing set is created with parameter values restricted to be within the extreme parameter values used for training.

It is clear by observing the testing sets statistics that the simple ANN is slightly better than the coarse model, and that the hybrid neural network is superior to both the coarse model, and the simple ANN. Examining the tables we can conclude the following: A) The simple neural network, even with 20 neurons in the hidden layer, did not give much better results that the ones obtained by the coarse model. B) The results obtained by the hybrid network with only two neurons in the hidden layer are slightly better than those obtained by the coarse model; this should be expected, because in a way in the hybrid approach, we are taking the coarse model as our base model. C) The results obtained by the hybrid neural network with 20 neurons in the hidden layer are superior to those obtained by the coarse model, and to those obtained by the simple ANN. In real time applications the hybrid approach is certainly justified.

The neural network approach proposed more recently by Bandler et.al. (1999) based on space-mapping technology can also be applied in this work, as an alternative to the hybrid ANN. In this case, the partial derivatives of the coarse model response with respect to the input variables will be required. In addition to alternative neural network approaches, for several option pricing problems we could choose among more than one parametric methods beyond the lattice (i.e., numerical solutions to differential equations, numerical integration, simulation). Evans and Jones (2002) recommend a methodology (based on their Gamma Test) that can provide the minimum sample size (of fine and coarse model evaluations) that is adequate to use in the training set, and an estimate of the error; in addition, using that information and after considering the computational burden of training with those alternative parametric methodologies, we can choose the one that is the most efficient to use for training (assuming this is an important consideration). Their approach can also help optimize the neural network in respect to the number of hidden neurons in a way that avoids the overfitting problem.

References

Bandler J W, Ismail M A, Rayas-Sanchez J E, Zhang Q J (1999) Neuromodeling of microwave circuits exploiting space mapping technology. IEEE Transactions Microwave Theory Technique 47: 2417-2427.

Black F, Scholes M (1973) The pricing of options and corporate liabilities. Journal of Political Economy 81: 637–659.

Boyle P P, Evnine J, Gibbs S (1989) Numerical evaluation of multivariate contingent claims. Review of Financial Studies 2: 241-250.

Brennan M J, Schwartz E (1985) Evaluating natural resource investments. Journal of Business 58: 135–157.

Charalambous C (1992) A conjugate gradient algorithm for efficient training of artificial neural networks. IEE Proceedings – G, 139: 301-310.

Cybenko G (1989) Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals and Systems 2: 303-314.

Dixit A (1989) Entry and exit decisions under uncertainty. Journal of Political Economy 97: 620–638.

Evans D, Jones A J (2002) A proof of the Gamma test. Proceedings of the Royal Society of London A 458(2027): 2759-2799.

Fletcher R (1987) Practical methods of optimization, 2nd edn, John Wiley & Sons.

Hornik K, Stinchcombe M, White H (1989) Multilayer feedforward networks are universal approximators. Neural Networks 2: 356-366.

Hutchinson J, Lo A, Poggio T (1994) A nonparametric approach to the pricing and hedging of derivative securities via learning networks. Journal of Finance 49: 851-889.

Lajbcygier P (September/October 1999) Literature review: The problem with modern parametric option pricing. Journal of Computational Intelligence in Finance, 6-23.

Rumelhart D, Hinton G, Williams G (1986) Learning internal representations by error propagation. In: Rumelhart D, McCleland J (eds.) Parallel distributed processing, MJT Press, vol. 1.

Trigeorgis L (1996) *Real Options: Managerial Flexibility and Strategy in Resource Allocation*, MIT Press.

Watson P, Gupta K C (1996) EM-ANN models for microstrip vias and interconnect in multilayer circuits. IEEE Trans. Microwave Theory and Techniques 44: 2495-2503.