分类号 F830.9　　　　密级 公开

U D C 303　　　　编号 86387

厦 门 大 学

博 士 后 研 究 工 作 报 告

基于人工智能在期权定价方面的研究

张鸿彦

工作完成日期 2013-8

报告提交日期 2013-9

厦门大学

2013 年 9 月

基于人工智能在期权定价方面的研究

Study on option pricing based on artificial intelligence

博　士　后　姓　名　张鸿彦

流动站（一级学科）名称　应用经济学

专　业（二级学科）名称　数量经济学

研究工作起始时间　2010 年　9 月

研究工作期满时间　2013 年 9 月

厦 门 大 学

2013 年　9 月

厦门大学博士后研究工作报告著作权使用声明

本人完全了解厦门大学有关保留、使用博士后研究工作报告的规定。厦门大学有权保留并向国家主管部门或其指定机构送交该报告的纸质版和电子版，有权将该报告用于非赢利目的的少量复制并允许该报告进入学校图书馆被查阅，有权将该报告的内容编入有关数据库进行检索，有权将博士后研究工作报告的标题和摘要汇编出版。保密的博士后研究工作报告在解密后适用本规定。

本研究报告属于：　　1、保密（　），　2、不保密（√）

纸本在　　　　年解密后适用本授权书；

电子版在　　　年解密后适用本授权书。

（请在以上相应括号内打“√”）

作者签名:　　　　　　　　　日期：　　年　月　日

导师签名:　　　　　　　　　日期：　　年　月　日

摘要

内 容 摘 要

期权理论是20世纪世界经济学领域最伟大的发现之一。由于期权具有良好的规避风险、风险投资和价值发现等功能，且表现出灵活性和多样性特点，故近30年来，特别是上个世纪90年代以来，期权成为最有活力的衍生金融产品，得到了迅速发展和广泛的应用。对于期权价格的正确定价不仅对于学术界而且对于金融市场的实际操作者来说都是十分重要的。目前已经有许多对于欧式期权定价的参数化模型，包括著名的Black-Scholes模型。但是由于有着一些不真实，与真实市场不协调矛盾的隐含参数，所以它们的定价效果并不如我们所期望的那么好。为了避免这些参数化模型的缺陷，基于人工智能的欧式期权定价模型越来越受到关注。同时，如我们所知，美式卖权没有参数化模型。如何正确地对于美式卖权进行定价毫无疑问是十分重要的。在学术界基于人工智能对于美式卖权进行定价的讨论较少。本文通过应用BP神经网络对于可能影响美式卖权的因素进行了一些研究。

本文内容安排如下：

第一章：

对于基于人工智能对于期权定价的文献进行了综述。

第二章：

总未平仓量反映了真实期权市场的活跃程度。非参数化方法例如神经网络不需要涉及任何的金融理论。为了研究真实市场的活跃程度是否与欧式期权的价格相关，本章将几类不同的平均总未平仓量作为神经网络的输入变量进行了研究。这些神经网络包含“纯”神经网络和混合神经网络。

第三章：

隐含波动率是隐含在可以观察的期权的市场价格中的波动率。由于波动率微笑的存在，不同种类同时到期的期权的隐含波动率不同，如何衡量不同种类期权的隐含波动率的最优权重一直是期权定价领域中的重要问题。本章建立了 BP 神经网络和遗传算法相结合的模型，将期权按钱性进行分类，提出了加权的隐含波动率作为神经网络的输入变量，通过遗传算法来求取不同种类期权的隐含波动率的最优权重。在香港衍生品市场的实证中表明，本文所提出的这些混合模型要优于传统的Black-Scholes模型。

第四章：

本章我们将研究通过 BP 神经网络来如何提高美式卖权的定价性能。如我们所知，标的资产价格，行使价格，到期日是可能影响美式卖权价格的主要因素。是否还有其他因素能够影响美式卖权的价格？如麦克米伦所说，金融市场的实际操作者往往用交易量作为股票期权的指标。金融市场的活跃程度能够影响美式卖权的价格吗？正如我们所知，隐含波动率包含了未来金融市场波动的信息。如果我们将隐含波动率作为神经网络的输入变量，神经网络的预测性能能否得到提高？金融市场的反向操作者经常将看跌看涨比率作为他们买入或卖出的指标。如果将一些看跌看涨比率作为变量输入到神经网络中，神经网络的性能能够得到提高吗？考虑及此，本章将使用多种不同的BP神经网络来研究是否有其他因素可以影响到美式卖权的价格。

关键词：期权定价；人工智能；Black-Scholes模型；钱性；隐含波动率

英文摘要

Abstract

The option theory is one of the most great discovery in the world economic field in 20th century . Owing to the function of the risk of elusion , venture investment , value discovery and the characteristic of agility and multiplicity , option has been the most great-hearted derivative product and has gained rapid development and broad application since 90th of the last century . It is important to price an option correctly not only to Academy but also to practitioners in financial markets. There are many parametric models to price European option including the famous Black-Scholes model. But owing to some unreal, not harmonize implied parameters with the real markets, the forecasting results is not as good as what we have expected. In order to avoid these deficiencies of parametric pricing models, the European option pricing models based on artificial intelligence is receiving more and more attention. At the same time, as we know, there are no analytical models in pricing American put options. How to determine the American put option price correctly is very important nodoubtedly. There is little discussion in pricing American put option based on artificial intelligence in academy. This paper will study the factors that may influence American put options by using BP neural networks.

The particular content of this paper is arranged as follows :

In chapter I:

Option pricing literatures based on artificial intelligence are reviewed in this chapter.

In chapter II:

Gross open interest reflects the activity of option markets. Nonparametric techniques such as artificial neural networks don't necessarily involve directly any financial theory. In order to study whether the activity of a real market is relative to a European option price, several kinds of average gross open interest are applied to the input variables of the neural networks including pure neural networks and hybrid neural networks in this chapter.

In chapter III:

Implied volatility is the volatility implied by an option price observed in the market. The implied volatilities among varied kinds of option expiring on the same date are different because of volatility smile effects. How to determine the optimal weight of the implied volatility among varied kinds of option is an important issue in option pricing fields. In this chapter, Hybrid forecasting models combining BP neural network with genetic algorithm are built. In such an approach, option partition according to moneyness is applied and weighted implied volatility measures are regarded as input of the neural network. The genetic algorithm is used to determine the optimal weight of the implied volatility among different kinds of option. Case study on Hong Kong derivative market shows that these hybrid models perform better than the conventional Black-Scholes model.

In chapter IV:

We will investigate how to improve American put option pricing by using BP neural networks in this chapter. As we know, Asset price (S), Strike(K), Maturity(T) are the key factors that can influence American put option price. Are there any other factors that can influence American put option price? As Mcmillan says, practitioners in financial markets often use trading volume as the index of stock option. Can trading activities of financial markets influence the American put option price? As we know, implied volatility may include information of fluctuation of financial market in the future. If we adopt implied volatility as an input variable of neural network, Can the performance of the neural network model be improved? Contrary operating traders in option markets often use ratio between put to call as the index of their buying or selling. If some kinds of ratio between put to call are input into neural networks, can the performance of NN models be improved? In view of this, several kinds of BP neural networks are used to explore whether other factors can influence American put option price in this chapter.

Keywords: Option Pricing； Artificial Intelligence；Black-Scholes Model；Moneyness；Implied Volatility

目 录

目　　次

Chapter I

Literature review in option pricing based on Artificial Intelligence

1.Sunisa, A., David, E. and Cihan H.D 's model[1].(2007)

The general regression neural network(GRNN) is used. A classical GRNN has four layers which are input layer, pattern layer, summation layer and output layer respectively.

A new hybrid neural network architecture is proposed in this work. Two GRNNs are used to predict option price. The input of the first GRNN are *S*(stock price);*X*(strike price);*T*(time to maturity); *r*(risk-free interest rate). The target output of this GRNN is the implied volatility(σ).Then the volatility obtained from the first network and the first four parameters are used as inputs of the second GRNN. The target output of the second GRNN is the difference between the BS model and actual price. Finally we can obtain the call or put option prices(*C,P*) from the combination of the BS model and the second GRNN.

The call option data from five different primitive stock assets which include Coca-Cola,McDonald's Corporation,Boeing, Citigroup, and IBM are used to train and test the model. There were totally 27496 records from July 1, 2002 to October 15, 2002. The data were divided into two sets: the first data including 23319 records were used as the training set. The second data including 4177 records were used as testing set. Since the hybrid neural networks are trained under a 3-month period, The 90 day HV(historical volatility) is applied in this paper.

z-score method which is used to normalize the data set is shown as the next :

$$v' = \frac{v - \overline{A}}{\sigma_A} \tag{1.1}$$

Where:

$\overline{A}$: means of input *A*; v : original value; σ_A : standard deviation of input *A*; v' : normalized value.

The mean squared error(*MSE*) and mean absolute error(*MAE*) are used to compare the performance:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \overline{y}_i)^2 \tag{1.2}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|y_i - \overline{y}_i\right| \tag{1.3}$$

Where:

y_i : the actual values;

$\overline{y}_i$: the estimated values;

Three models including two kinds of BS models are tested in this research finally. The hybrid NN model performs the best in the models.

2. Henrik Amilon's model. (2003)

An MLP neural network with one hidden layer of neurons is used in this work. The transfer functions of hidden layer is $g(\cdot)$=tanh($\cdot$). The transfer functions of the output neurons are logistic function. That is :

$$g(\cdot) = \frac{1+\tanh(\cdot)}{2} \tag{1.4}$$

The MLP networks are trained in batch mode. The training algorithm for adjusting weights is conjugate gradient algorithm. All inputs are normalized. The continuously compounded return on a 90-day treasury bill is used as an approximation of a risk-free interest rate.

The data used in this work are from the time periods June-March 1998, and June 1998-March 1999. The dataset is divided into three parts: a training set, a validation set, and a test set.

The input variables of NN are:

$I_t / K; I_{t-1} / K; I_{t-2} / K; I_{t-3} / K; I_{t-4} / K$ ($I_t, I_{t-1}, I_{t-2}, I_{t-3}, I_{t-4}$: the index value; *K*: the strike price); *T-t* (time-to-maturity in trading days(252 per year)); $r_t(T^{cal} - t)$ (r_t : the risk-free interest rate; $T^{cal} - t$: time-to-maturity in calendar days(360 days per year)); σ_{30} (the 30 most recent continuously compounded daily returns of the OMX index); σ_{10} (the 10 most recent continuously compounded daily returns of the OMX index).

The output variables of NN are C_t^b / K (the bid price of the call option at time *t*), C_t^a / K (the ask price of the call option at time *t*).

Finally, three models including the Black-Scholes model with historically estimated volatility and the Black-Scholes model with implied volatility estimates were used to compare the performance.

3. Shinn-Wen Wang's model[3] (2005)

Since the market price of warrants and the stock prices are not assumed to change continuously as in the BSM. This minimum price tick-jump system will result in a space for arbitrage. A genetic based neural network is used to arbitrage successfully in Taiwan option market.

4. Chih-Hsiung T, Sheng-Tzong C, Yi-Hsien W and Jin-Tang P' model[4]. (2008)

A Grey-EGARCH model which combines GARCH model and GM(1,1) model is built to estimate the volatility in Taiwan stock market. Then the estimated volatility is used as the input of neural network to forecast option price.

A three-layer back propagation neural network is used in this paper. The data which comprise 21120 call options are from January 3, 2005 through December 29, 2006. The 70% of the data are used as the training set and the other 30% of the data are used as the testing set. The input variables of NN are moneyness, the time-to-maturity, risk-free rate and the above estimated volatility. The output variable of NN is option price.

Finally different volatility approaches are compared.

5. Christopher A Z's model[5]. (2003)

A binomial trees model which is linked to an innovative stochastic volatility model based on wavelets and artificial neural networks is used to price call option in this paper. On the other hand, neural networks trained with genetic algorithms reverse-engineer the Black-Shcoles formula.

A separate multilayer perceptron (MLP) neural network is adopted in the first neural network model. The input variables of first neural network are wavelets coefficients and the output variable of it is also wavelet coefficient.

Two separate neural networks are adopted in the second model. The one forecast option price and the other one forecast corresponding deltas.

The input variables of the above neural network are T(the time to maturity); S-X(the current stock price S divided by the option strike price X); r(the current risk-free rate); σ(standard deviation measured over a sliding window whose length are the number of days to the expiry). In order to lie between zero and one, the inputs are normalized.

A single MLP-type neural network is adopted to price the delta and a committee of expert MLP networks is adopted to price option prices. A traditional conjugate-gradients method is used to fit current market option prices. After that, genetic algorithms are used in order to minimize the risk of option mispricing.

Option prices for the stocks of three US companies: AOL Time Warner, IBM and Motorola are used in this paper.

6. Lev B, Alex F's mode[6] (2003)

There are six European option pricing models in this work. Four of them are neural network models and the other two are Black-Scholes models.

MLP (Multilayer perceptron) neural network with one hidden layer is used in this paper. The input variables of the first neural network are $S/X,T$. The output variable of the first NN is C/X. The input variables of the second NN are $S/X,r,T,\sigma_{30}$(Historical volatility of 30 days). The output variable of the second NN is C/X. The input variables of the third NN are $S/X, r, T, \sigma_{imp}$(implied volatility). The output variable of the third NN is C/X. The last neural network is a hybrid neural network. The input variables of the fourth NN are $S/X, r, T, \sigma_{30}$. The output variable of the fourth NN is C/X-BSF/X.

There are two other Black-Scholes models whose input variables are S, X, r, T, σ_{30} and S, X, r, T, σ_{imp} respectively.

Three kinds of data sets including training dataset, validation dataset and testing dataset are used in this paper. 60 percent of data are for the training set, 20 percent of data are for the cross-validation set and the other 20 percent of data are for the testing set. The data are from January 1986 to June 1993. The activation function of the neurons is the hyperbolic tangent function.

Finally the performances of six models are compared by using OEX 100 index call options.

7. Chin-Tsai Lin and Hsin-Yi Yeh's model[7]. (2009)

A three-layer back propagation neural network is used in this paper. The input variables of NN are *S, X, t, r, σ*. There are four methods to estimate the volatility *σ* including historical volatility approach, implied volatility approach, Garch approach and Grey prediction approach in this work. The output variable of NN is *C*.

The method to estimate historical volatility follows as the next:

$$R_t = \ln(S_t) - \ln(S_{t-1}) \tag{1.5}$$

$$\sigma_t = \sqrt{\frac{1}{n-1}\sum_{t=1}^{n}(R_t - \bar{R})} \tag{1.6}$$

$$\sigma^H = \sigma_t \sqrt{N} \tag{1.7}$$

Where:

R_t : the return of the stock at time t;

$\bar{R}$: the mean of the R_t;

S_t : the stock closing price;

N: the number of trading days;

The method to estimate implied volatility follows as the next:

$$\sigma^I = \frac{1}{n}\sum_{j=1}^{n}\sigma_j \tag{1.8}$$

σ_j : the implied volatility on day j.

The Garch(1,1) approach is used in this work:

$$\sigma_i^2 = \omega + \alpha \cdot u_{i-1}^2 + \beta \cdot \sigma_{i-1}^2 \tag{1.9}$$

The data consist of Taiwan stock index option prices from 2 January 2003 to 31 December 2004. Data partition according to moneyness is adopted before training neural networks. 70% of the data are used as the training set and the remaining 30% are used as the testing set.

8. Chris Charalambous and Spiros H.M's model[8]. (2005)

Conjugate gradient algorithm joined with Charalambous line search algorithms is used in this paper. A feed forward neural network which has one-hidden layer is used. The activation function of the neurons in the hidden layer is tansigmoid. All input variables are normalized.

Neural network approach is used to price an European put option and a real option respectively.

9. Ulrich A, Olaf K and Christian S's model[9]. (1998).

Statistical inference for neural network is used in this paper. A model-selection strategy based on significance tests is adopted. With the help of statistical inference we can decide which input variable contribute more significantly than the other input variables to the explanation of option prices. MLP type of neural networks with a single hidden-layer is adopted exclusively in this work. The transfer function of the hidden neurons of neural network is tanh-function.

After some preprocessing, the dataset comprised 13676 call options issued on the leading German stock index DAX. The call options in the first nine months of 1994 including 10848 data were used as training sample. On the other hand, the remaining 2828 call options were used as test sample.

The first estimate of volatility is a historical volatility which is as the next:

$$\sigma_{30} = s\sqrt{252} \tag{1.10}$$

Where:

s: the standard deviation of the return for the close-to-close DAX levels of the most recent 30 days.

The second estimate of volatility is the DAX volatility Index(*VDAX*) shown as the next:

$$\sigma_{VDAX} = VDAX \tag{1.11}$$

Where:

VDAX: weighted average of volatilities implied by different DAX options traded at the DTB on the previous day.

The input variables of NN are $S/X, T-t$, r; σ(σ_{30} or σ_{VDAX}).The output variable of NN is C/X. Two network architectures including pure network and hybrid network are adopted in this work.

Finally the performances of six models are compared and hedged parameters are calculated.

10. Yi-Hsien Wang's model[10]. (2009)

Since ARCH or GARCH approaches can't capture asymmetric features of returns behavior, GJR-GARCH approach is proposed as the next:

$$\sigma_t = \tau_0 + \sum_{j=1}^{q} \beta_j h_{t-j} + \sum_{i=1}^{p} \alpha_{1i} \varepsilon_{t-i}^2 + \alpha_2 S_{t-1}^- \varepsilon_{t-1}^2 \tag{1.12}$$

Here, $S_{t-1}^- = 1$ if $\varepsilon_{t-1} < 0$ and $S_{t-1}^- = 0$ if $\varepsilon_{t-1} \geq 0$.

In order to modify the error terms, Grey-GJR-GARCH volatility approach is proposed in this paper.

A back propagation neural network with one hidden layer is employed. The activation function is the sigmoid function.

The dataset consists of 21120 Taiwan stock index call options from January 3, 2005 through December 29, 2006. Data partition according to moneyness is adopted. 70% of the data are used as the training set and the remaining 30% of the data are used as the testing set.

The input variables of NN are $S, X, T-t$, r; σ(GARCH volatility approach, GJR-GARCH approach, Grey-GJR-GARCH volatility approach). The output variable of NN is C.

Finally the performance of three nonlinear neural network forecast models with different volatility approaches are compared.

11. Po-Chang Ko's model[11].(2009)

A neural regression model is proposed in this article. The training algorithm of this model includes three steps:

Step 1:

The traditional linear regression method $C' = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$ is modified to produce C' .

Where:

C' : the predicted normalized value of call option;

$\beta = [\beta_0\ \beta_1\ \beta_2 \cdots \beta_p]^T$: the weighted coefficients; represented by distinct neural networks;

$x = [1\ x_1\ x_2 \cdots x_p]^T$: the considered variables vector;

P: the number of parameters estimated

Step 2:

$$C' = \phi^{-1}(C') \tag{1.13}$$

$\phi(\square)$: the normalized function.

Step 3:

The total error energy ξ_m shown as the next is estimated from the tracking error based on the no-arbitrage phenomenon. ξ_m should be zero at the expiration date.

$$\xi_m = e^{-r_f T}(|V(T)|) \tag{1.14}$$

$\xi_m(q)$: the instantaneous error energy at iteration *q* by model *m*;

$V(T) = V_S(t) + V_B(t) + V_C(t)$: the value of portfolio at time *t*;

$V_S(t)$: the value of stocks at time *t*;

$V_B(t)$: the value of bonds at time *t*;

$V_C(t)$: the value of held call options at time *t*;

The dataset issued on Taiwan Futures Exchange(TAIEX) includes 34403 European call options from January 3,2005 to December 31, 2006. 80% of the data are used as training set and the other 20% of the data are used as testing set.

The corresponding input set for β_0 are *K,T*; The corresponding input set for β_1 are *K, T* , *S* ,*σ*; The corresponding input set for β_2 are *K,T, S*; The corresponding input set for β_3 are

$K, T_{t-1}, T_{t-2}, T_{t-3}$;The corresponding input set for β_4 are $K, T_{t-1}, T_{t-2}, T_{t-3}$;The corresponding input set for β_5 are K,T, $\sigma_{t-1}, \sigma_{t-2}, \sigma_{t-3}$;

The number of hidden layers of each neural network are set to two; The number of neurons in each hidden layer are set to 6; The activation function of neurons of hidden layer is sigmoid function; The activation function of neurons of output layer is linear function; The learning rate is set to 0.1; The number of epochs is set to 5000.

Then the absolute delta-hedging errors of two models including the Black-Scholes model and the neural regression model are compared respectively.

12. Rene Garcia and Ramazan Gencay's model[12]. (2000)

A feed forward neural network is adopted in this work. The activation function of neurons of hidden layer is logistic function. Two neural network models without hint and with hint are estimated respectively:

$$f^{NN}(S_t / K, \tau; \theta) = \beta_0 + \sum_{j=1}^{d} \beta_j \frac{1}{1+\exp(-\gamma_{j0} - \gamma_{j1}(S_t / K) - \gamma_{j2}\tau)} \tag{1.15}$$

$$f^{WH}(S_t / K, \tau; \theta) = \beta_0 + \frac{S_t}{K} \sum_{j=1}^{d} \beta_j^1 \frac{1}{1+\exp(-\gamma_{j0}^1 - \gamma_{j1}^1(S_t / K) - \gamma_{j2}^1\tau)}$$
$$-e^{-\alpha\tau} \sum_{j=1}^{d} \beta_j^2 \frac{1}{1+\exp(-\gamma_{j0}^2 - \gamma_{j1}^2(S_t / K) - \gamma_{j2}^2\tau)} \tag{1.16}$$

Given from the above models, we know that the input variables of two NNs are *S/K*,τ,1 and the output variable of NNs is *C*.

The data consist of daily S&P 500 Index European options issued on CBOE from January 1987 to October 1994. For each year, the sample are divided into three parts: first half of the year(training set), third quarter(validation set) and fourth quarter(prediction set).

Finally the pricing error and the average hedging errors of the above models and the B-S model are calculated respectively.

13. Guido M, Marco M, Oreste N, Paolo A and Marco F's model[13]. (2003)

First, a novel algorithm based on a path integral approach is proposed to price option. Second, a Radial Basis Function network is used in this work. The activation functions adopted in this paper are Gaussians of variable width. A simulation test is conducted when *r*, *X* ,σ are fixed. The input variables of NN are *S*, τ . The output variable of NN is *C*.

14. Marco J M, Guido M, Oreste N, Michele T, Marco F, Paolo A's model[14]. (2004).

This paper is the context of the last paper. Two kinds of neural networks including a MLP NN and a RBF NN are used in this work. Then a simulation test is conducted when *r*, *X*, σ are fixed similar to the previous paper. Finally Greek letters are also evaluated.

15. A. Carelli, S. Silani and F. Stella's model[15]. (2000)

Feedforward neural networks with one hidden layer or two hidden layers are applied in this work. The activation function of the neurons of the hidden layer is logistic function($f(z)=\dfrac{1}{1+\exp(-z)}$). The activation function of the neurons of the output layer is hyperbolic tangent function($f(z)=\dfrac{\exp(z)-\exp(-z)}{\exp(z)+\exp(-z)}$). The input variables of ANN are time to maturity and strike price. The output variable of ANN is implied volatility.

A novel training algorithm is proposed to select the network's structure.

Then a new pricing algorithm is shown as the following five steps:

1) Using a ANN model to compute the implied volatility from the market prices;
2) Using the Black's formula to compute the option price;
3) Using Dupire's formula to compute the local volatility;
4) Using a trinomial model to compute option price;
5) A new iteration begins;

16. Panayiotis, C.A , Chris C and Spiros H.M's model[16]. (2006)

Feedforward neural networks with single hidden layer are applied in this paper. The transfer function of the hidden layer is hyperbolic tangent sigmoid transfer function and the transfer function of the output layer is linear transfer function as well.

A novel artificial neural network optimized with the Huber function is proposed in this work.

The dataset traded on CBOE consists of 64627 S&P 500 Index call options from April 1998 to August 2001. On the other hand, the dataset are divided into training dataset, validation dataset and testing dataset. The Root Mean Square Error(*RMSE*) and the Mean Absolute Error(*MAE*) are calculated as the performance of the pricing models.

There are three types of neural network such as standard ANN , ANN optimized with the Huber function and hybrid neural network architecture.

The input variables of ANNs are $Se^{-\delta T}/X$ (δ: daily dividend yield; *T*: time to maturity computed assuming 252 days in a year); *r*(continuous interest rate calculated by using nonlinear cubic spline interpolation for matching each option contract); $\sigma_{60}\sqrt{T_{252}}$ (historical volatility estimate calculated using all the past 60 log-relative index returns) or $\sigma_{vix}\sqrt{T_{252}}$ (volatility calculated as a weighted average of S&P 100 option with an average time to maturity of 30 days especially on at-the-money options) or $\sigma_{av}\sqrt{T_{252}}$ (daily average implied volatility) or $\sigma_{avT}\sqrt{T_{252}}$)(daily average prematurity volatility obtained by fitting the BS to all options that meet the same maturity date if four different available call options exist)

The output variable of the above two neural networks is C_q^{mrk}/X_q and the output variable of the hybrid neural network is $C_q^{mrk}/X_q-C^{\theta}/X$.

Finally the performances of all the neural network models and the BS models with different volatility measures are compared.

17. Christopher Zapart's model[17]. (2002)

With the help of wavelets and artificial neural networks, a binomial tree-based option pricing model is built in this work.

A MLP neural network with one hidden layer containing 5 neurons is used to produce a one-step ahead forecast of wavelet coefficients. The Haar wavelet transform is chosen. The activation function of the neurons in hidden layers is tanh(x). The activation function of the neurons in output layer is linear function ($f(x)=x$). The input variables and the output variables of ANN are also wavelet coefficients .

Option prices of three US companies: AOL Time Warner, IBM and Motorola are used in this paper. The risks shown as the next are compared between the Black-Scholes and the dynamic volatility approach for the stocks of AOL Time Warner, IBM and Motorola.

$$risk = \sum_{i} \left|(PL)_i\right| \tag{1.17}$$

Where:

i: all strike prices for a given call option;

$(PL)_i$: the profit(loss) made on a particular delta-hedged position(sell a call option, buy the underlying stock).

18. Z. I. Sameur and G.T. Temur's model[18]. (2009).

MLP(multilayer perceptron) neural networks are adopted in this work. The number of neurons in hidden layer and the number of hidden layers are determined by the value of performance measurement of training sets. Mean Square Error (*MSE*) is adopted as a performance measurement.

There are three kinds of MLP network is this paper. The input variables of ANN Model1 are type of options (American or European), strike price, spot price, maturity and Interest rate. The input variables of ANN Model2 are type of options(American or European), strike price, spot price, maturity, Interest rate and variance 1(volatility estimated from the period that begins from 2007 and ends on the day of options price gathered). The input variables of ANN Model3 are type of options(American or European), strike price, spot price, maturity ,Interest rate and variance 2(volatility estimated from data that cover the month that option price gathered). The output variable of all the NN models is market price.

The dataset traded on CBOE consists of 134 data covering S&P 100 European and American index options for both put and call type. The dataset are divided into training set and testing set.

Finally the performances of all the NN models are compared.

19. Yi-Hsien Wang's model[19]. (2009)

This paper is similar to paper [4] in this chapter.

GM(1,1) model and GARCH model are combined to construct a new GM(1,1)-GARCH