A Case Study Using Neural Network Algorithms: Horse Racing Predictions in Jamaica

Janett Williams[1, 2] and Yan Li[2]

[1]School of Computing & IT, University of Technology, Jamaica, 237 Old Hope Road, Kingston 6, Jamaica

[2]Department of Mathematics and Computing, University of Southern Queensland, Toowoomba, Australia

Abstract *–Neural networks (NNs) have been applied to predict many complex problems, such as horse racing prediction. This study investigates the performances of four supervised neural network algorithms in horse racing. We employ Back-propagation, Quasi_Newton, Levenberg-Marquardt and Conjugate Gradient Descent learning algorithms to real horse racing data collected from Caymans Race Track, Jamaica. The training and testing data come from 143 actual races collected from 1 January to 16 June 2007. The experimental results demonstrate that all algorithms can provide acceptable predictions with an accuracy of 74%. BP algorithm slightly performs better than other three algorithms but needs a longer training time and more parameter selections.*

Keywords: Neural Network, Back Propagation Algorithm, Machine Learning

1 Introduction

Neural Networks (NNs) since their introduction in the 1940s have been applied in many areas including predictions. They have been employed to predict weather forecasting, traveling time and the stock market, *etc*. The gaming industry has also seen its share of neural network predictions, covering from predicting the results of fighting, racing to team sports, such as soccer and basketball. The most popular training algorithm for those predictions has often been Back-Propagation (BP) due to its simplicity. In this study, we employ four supervised neural network training algorithms, namely: BP, Quasi_Newton (QN) algorithm, Levenberg-Marquardt (LM) algorithm and Conjugate Gradient Descent (CGD) algorithm, to predict the horse racing in Caymans Race Track, Jamaica. The training and testing data come from 143 actual races collected from 1 January to 16 June 2007. The experimental results demonstrate that all algorithms can produce acceptable predictions with an accuracy of 74% by average.

The rest of the paper is organized as follows: The structure of the neural network and its training algorithms are presented in Section 2. In Section 3 we briefly introduce the horse racing in Jamaica and the data collection for the experiments. The experimental results and their analysis are presented in Section 4. Finally the paper is concluded in Section 5.

2 Neural Networks

2.1 The NN Structure

A NN is normally configured for a specific application and learning involves adjustments of the synaptic weights that connect neurons. There are various kinds of network structures, each of which results in different computational properties. The main distinctions lie between two categories – recurrent neural networks (RNNs), which have extra feedback paths from outputs of some neurons to inputs, and feed forward neural networks (FNNs) which use directed acyclic graph and data in a training set to induce learning. No matter what the structure is, all NNs, once trained, are able to recognize similarities when presented with new input patterns and resulting in predicted output patterns. In this study, the network consists of three layers of neurons: a layer of input neurons, each corresponding to one input signal; a layer of output neurons, each corresponding to an output value; and a hidden layer of neurons that adjust in order to represent a relationship. For a supervised learning algorithm, a network is structured by designating the number of neurons in each layer, and then provides a set of training patterns, each consisting of input values and correct output values. The network is then trained to recognize these input values and the correct outputs. Once the network is fully trained by an algorithm, it can be provided by new input values and produces a set of desirable outputs.

2.2 Learning Algorithms

Neural networks acquire knowledge through a learning algorithm and by storing this knowledge in their weights. The learning of a network is a process of adjusting its interconnected weights recursively. For a supervised learning algorithm, in order for a NN to perform any task it must be trained with an established set of input and output values. The weight updating is generally described as:

$$w_{ij}(n+1) = w_{ij}(n) + \Delta w_{ij}(n). \tag{1}$$

Here $\Delta w_{ij}(n)$ is determined by a learning algorithm and $w_{ij}(n)$ is initialized randomly. In this study, we employ the following four supervised algorithms to test their prediction power and compare their performances.

2.2.1 BP Algorithm

The inputs from the input units are propagated to the output units through the network. Each link from unit i to unit j has a numeric weight w_{ij} which determines the strength and sign of the connection. A weighted sum, a_j, for the given inputs, x_i, and weights, w_{ij}, is computed by the following function [1]:

$$a_j = \sum_{i=1}^{n} w_{ij} x_i \tag{2}$$

Here n is the number of inputs to one neuron. The standard sigmoid activation function with values between 0 and 1 is used to determine the output at j:

$$y_j = f(a_j) = \frac{1}{1+e^{-a_j}} \tag{3}$$

The error, $E_j(n)$, between the actual output $y_j(n)$ and the target output $d_j(n)$ for the neuron j is then calculated as below.

$$E_j(n)=d_j(n) - y_j(n). \tag{4}$$

Then the learning rule for the traditional BP algorithm is

$$\Delta w_{ij} = \eta x_i + \alpha \Delta w_{ij} = \eta x_i - \alpha \frac{\partial E_j}{\partial w_{ij}} \tag{5}$$

Where η is the learning rate and α is the momentum factor, which determines the influence of the past parameter changes on the current direction of movement in the parameter space. Using the momentum method, the network tends to follow the bottom of narrow gullies in the error surface (if exist) rather than crossing rapidly from one side to the other.

2.2.2 Quasi-Newton BFGS Algorithm

Quasi-Newton algorithm [2] is an enhancement of the BP. Instead of updating after each training set case, Quasi-Newton updates only after all training sets have been presented. There is also no need for parameter selections which removes human errors and makes it easier to use. Calculation of the error gradient is done through the sum of the error gradient on each training case and maintaining an approximation to the inverse Hessian matrix. On each epoch a back tracking line search is performed in the direction of

$$d = -Hg \tag{6}$$

Where H is the estimation of Hessian matrix using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) update and g is the direction of the steepest descent.

2.2.3 Levenberg-Marquardt Algorithm

A Levenberg-Marquardt training algorithm [3] can only be used on networks with a single output unit. The weights are updated by the formula:

$$\Delta w = -\left(Z^T Z + \lambda I\right)^{-1} Z^T \mathrm{E} \tag{7}$$

where E is the vector of case error and Z is the matrix of partial derivatives of these errors with respect to the weights.

2.2.4 Conjugate Gradient Descent Algorithm

Like the Quasi-Newton, Conjugate Gradient Descent algorithm [4] is also a batch updating algorithm. It calculates the average gradient of the error surface across all cases before updating the weights at the end of the epoch. Conjugate Gradient Descent algorithm firstly determines the direction of the steepest descent, then projects a straight line in that direction and locates a minimum gradient value along the line. Further line searches (conjugate directions) are done at each epoch. The initial search direction is given by:

$$d_k = g_i \tag{8}$$

The search directions are updated thereafter using the Polak-Rebiere formula:

$$w_{i+1} = w_i + \alpha p_i \tag{9}$$

where $p_i = -g_i + \beta_i p_{i-1}$, $\beta_i = \frac{\Delta g_{i-1}^t g_i}{g_{i-1}^t g_{i-1}}$ and $\Delta g_{i-1}^t = g_i^t - g_{i-1}^t$

The implementation of the above neural network algorithms are public available through several software packages. In this paper, we use Trajan 6.0 neural network simulation package available via the website: http://www.trajan-software.demon.co.uk/. The classic Back Propagation, Quasi-Newton, Levenberg-Marquardt and the fast second-order Conjugate Gradient Descent algorithms are supported in the package.

3. Horse Racing Prediction and Data Collection in Jamaica

3.1 Horse racing prediction using Neural Networks

The most popular form of modern horse racing in Jamaica is the racing of mounted Thoroughbred horses over flat courses at distances from 1000m ($^1/_2$ of a mile) to 3200m (2 miles). Using ANNs to predict horse racing is not new. B. Halford [5] in his report analysed different methods used to train ANNs. He eventually created a network that predicted the correct winner the most of times, which outperformed the racing experts' predictions [5]. Salzberg [6] even before Halford tested an algorithmic solution to horse racing creating Handicapper. After several trials his algorithm was found to outperform experts at predicting the winners.

Although Chen *et al* [6] investigated greyhound racing it was a very similar environment to horse racing. Their research investigated the use of BPNN in predicting races. It was found that like Salzberg and Halford the ANN outperformed the human experts. Today the use of ANN in horse racing is becoming popular. There are now horse racing prediction software products available, such as the ThoroBrain 5 [7] and BrainMaker [8, 9]. BrainMaker's successful use was demonstrated by Don Emmon who narrowed his winner search down to 3 horses and of the 22 races observed, he predicted 17 of the correct winners, a 77% prediction accuracy [10].

Caymanas Race Track (CRT) is located in Portmore, St. Catherine, Jamaica W.I. and is designed solely for horse racing. The race track has a 1800m (9F-Furlong)) dirt circuit with a stretch run of just over two furlongs on the left side. There are two chutes on opposite sides. The 1000m (5F) straight course utilizes one chute, while the other caters to distances up to 1800m consisting of one turn. All distances in excess of 1800m have two turns. The maximum number of runners in a race is sixteen (16), excepting the 1820m (9F25 yards) route, which is 12. It is close to the sea, causing a gentle breeze to blow against the runners in the final stretch run on most racing days.

3.3 Data Collection

Thoroughbred racing is done with handicaps, which involves evaluating the demonstrated abilities of a horse in light of the conditions under which it will be racing on a given day. These conditions include the distance of the race, the type of racing surface – dirt or grass, and the level of the competition. Horse racing predictions involve many of these conditions. To make a prediction a person learns everything about the race, ranging from the horses, their riders, bloodline and the type of competition to the track.

Because a NN uses numeric values as input variables certain information cannot be used, such as the names of a horse and its jockey. In this study, the following variables are used: racing distance, type of race, past position, weights of horse and jockey, horse's finish time, equipment used, such as visor and blinker, age of horse and number of the horses in the race.

```
1. SALSA    112 lb (51.0 kg) ( 10 days)   CliveLynch (147:24,22,11)  $240,000   JThomas
(32:7,4,5)
  5,B, Sire: DIXIE DANCER (169:19,18,20     87:12,7,9) (     )     34:9,9 RANGE   : 5R - 6H
  G    Dam: VESTIA (14:3,1,2     4:1,0,2)   by EXOTIC (B)          0:0,0 CONFIRM: 5R - 6
       TRAVELER                                      (V)           1:0,1 CURR:
       Breeder: DR. C. BRADFORD, PHILIP FEANNY & MUD 2:0,1               6:2,1,0,$278400
       IVOR                                CLARKE                        LIFE:
       Owner: JOSEPH THOMAS                                              35:8,6,4,$1222000
       Colours: BLACK BODY WITH WHITE STAR,RED SLEEVES,RED CAP

                          23  47                                                 10          101
29May02 CP 9  M 3uCL180*  3   4   --   1013  5R - 2-3½ 2-8  2-3  2-1   RobReid   8   3  7 2/1  4   ......

          *** Claimed from Trainer - DESMOND C. SHAND, Owner - IVAN E. ROWE ***

                          24  47                                                 11          100
           1              4   4        1003                                      2           3
           4              23  48                                                 10          123
           1                      --   1223                                          4
15May02 CP 5  F 3uCL190   4   2        5S - --   4-4½ 1-1  1-1¼ RobReid   6      1   6 *4/5 2
20Mar02 CP 1  F 3uCL180   23  47  115       6H - 3-2  2-2  2-1½ 5-3½ LMiller   10    0   9 5/1  100
6Mar02  CP 3  F 3uCL180   3   3   4    1004  5S - --   1-1  1-4½ 1-4  RobReid   1    6   7 9/5  4
20Feb02 CP 1  F 3uCL190   23  47  --        5H - 2-2½ 3-5  4-4  4-6½ DMiller3  11    6   7 3/1  110  .......
2Feb02  CP 7  F 3uCL180   3   2   --   1084  5R - 4-2  4-5  4-5  4-3¾ DSThomas 2     4   7 2/1  1
29Dec01 CP    G 3uCL150*  23  47  --        5S - --   3-2  4-1  4-4¼ SEllis   11         7 *4/5 102
```

Figure 1 Past Performance Sample Information

Each variable is then normalized. The data collected come from the past performance record given by CRT as shown in Figure 1. Each race begins with a race number and a short description about the race. This includes *age*, *type of race*, *purse*, *distance*, *time of race* and *types of bets* offered. For example,

1 3-Y-O & Up Claim Price: $100,000 Purse : $110,000 Dist : 5H Furl. 12:NOON D/E, QUIN, EXACTA, TRI

The first six lines in Figure 1 describe the horse's statistics and connections such as the horse number, name, equipment worn in bracket, weight in lbs, the number of days since last outing, breeder, owner, and date of birth. All statistics are listed in the order of *total: wins,second+third.* For example; *12:5,3* meaning 12 starts with 5 wins and the sum of second and third places is 3. All equipment, mud and striking range statistics are based on actual finishing position and lengths *i.e.* disqualification by stewards of the race does not affect them. The current and lifetime statistics with earnings do reflect these amendments.

A maximum of nine (9) past performance lines are usually printed in addition to aborted starts. Aborted starts are when a horse which has been scheduled to race, but at the end does not race because of sickness, lameness, injury, fractious behavior, bolted *etc.* These are recorded to show behavior patterns of a horse and can be used to highlight risky propositions. This information is not printed beyond the last two past performance lines. The past performance lines contain race date, track, track variant, track condition, split times, final time, distance in furlongs, equipment worn, race description, position/lengths at 4F, position/lengths at 2F, position/lengths at 1F, position/lengths at finish, jockey, weight carried, post position, no. starters, odds, winner, lengths, second, lengths, third, comments.

3.4 Training and Testing Data for the Prediction

The development of a neural network is divided into two phases – training and testing. For a network to learn it must firstly be taught, in this case it is trained to recognize particular patterns. This training set is passed through the network a number of epochs so that the network will be able to pick up the basic patterns and the relationships. After the network has been trained properly the testing stage begins. This is where new pattern sets (testing set) are passed through the network to be identified. This phase is needed to test whether the network has learnt to recognize the desired patterns. The training and test sets are created using past performances of the horses in the record. At least two-thirds of the data collected should be used for the training set [11]. In this paper, eighty percent (80%) of the data are used as the training data and the rest are used for testing.

4 Experimental Results and Analysis

In this section, we apply the four training algorithms to the horse racing data collected from actual races and compare their prediction performances. Jamaica has races every Saturday and Wednesday as well as every public holiday. For example, between January 1, 2007 and February 21 there were 11 racing days, each having at least 10 races and there were three racing days between June 9 and 16. A total of 143 races were used in this study.

Unlike other neural application, in this paper we use one NN to represent one horse. Each network has eight input neurons representing the input variables. At Caymanas Race Track the maximum number of horses per race is 16. The order of the predicted racing time indicates the positions in the racing results. The experiments are carried out in an Intel (R) Pentium D Processor (3GHz) computer by using the BP, LM, QN and CGD algorithms supported in Trajan 6.0 neural network simulation package. Table 1 shows the predicting results by using all four neural algorithms in one actual race (Race 5, June 9) and its actual racing result.

All four algorithms predicted the correct horses, however only Conjugate Gradient Descent predicted the correct winner. The other three all predicted the same order. It can be seen that these three predicted the third place correctly. There were times when only one horse is predicted correctly.

From Figure 2, the four algorithms give different results and it was observed that Quasi-Newton and Levenberg-Marquardt provided many similar predictions with Levenberg-Marquardt predicted slightly higher accuracy over numerous trials. There are times, however, where Quasi-Newton outperforms over other algorithms.

The experiments demonstrated that there is no one single "best" training algorithm for all applications. The performance of the algorithms is application specific, depending on the characteristics of the training data set, such as the value distributions of each variable and the correlations between variables. Overall all the four training algorithms provided acceptable predictions as show in Table 2. In many cases they produced very similar predictions as shown in Table 3

Table 1: Actual race result and the predicting results for Race 5 on June 9

Show Placement	Actual	BP	CGD	QN	LM
1st	La Reina	Classy Boy Pete	La Reina	Classy Boy Pete	Classy Boy Pete
2nd	Classy Boy Pete	La Reina	Hidalgo	La Reina	La Reina
3rd	Hidalgo	Hidalgo	Classy Boy Pete	Hidalgo	Hidalgo

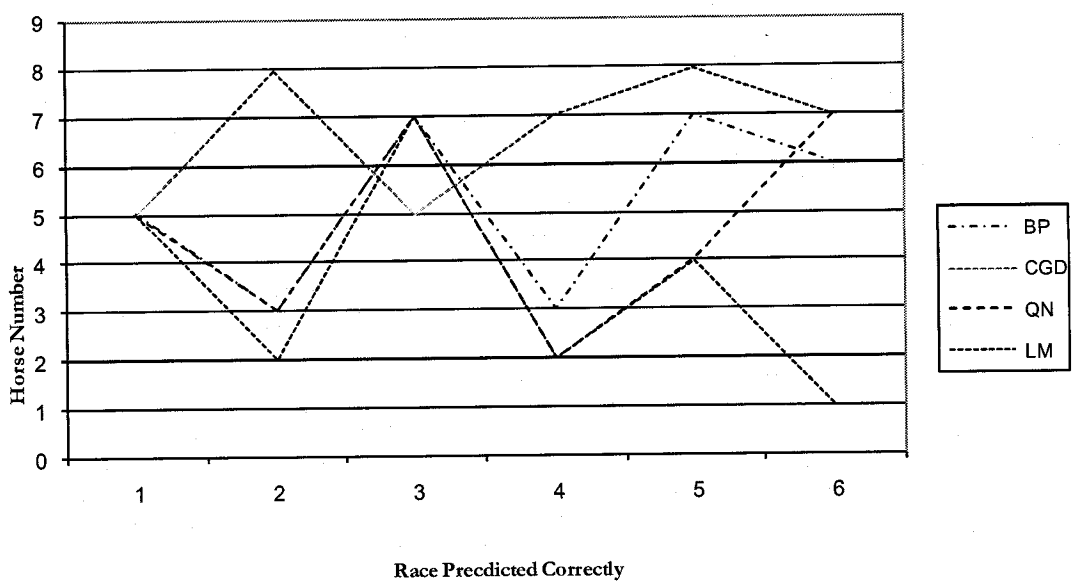

Figure 2: Prediction Performance of Algorithms

In total, 407 (71%) of 572 (143 x 4) races are correctly predicted with at least one of the actual horses ending in a winning position (top three positions) by all four algorithms. In this study, all the races used include different racing distances ranging from 1000m to 1820m. We believe that the variety of the distance influenced the performance of each neural algorithm. The epochs for training the algorithms are set to 100 as it can provide a good result forall. All four algorithms produce very similar results with BP outperforms slightly better but BP suffers a longer training time and more parameter selections. We also investigate the influences of these parameters on the performance of BP.

Table 2 : Summary of Experimental Results

	BP	CGD	LM	QN
All 3 horses predicted correctly	60	48	58	57
All 3 horses in actual results predicted	12	14	11	11
2 horses in actual results predicted	14	11	10	9
1 horse in actual results predicted	19	27	23	23
No horse in actual results predicted	38	43	41	40

Table 3: Race showing Same Prediction (Race 11, February 10)

Show Placement	Actual	BPNN	CGD	QN	LM
1st	Tuna Saloona	Faithful Robbie	Faithful Robbie	Faithful Robbie	Faithful Robbie
2nd	Faithful Robbie	Tuna Saloona	Tuna Saloona	Tuna Saloona	Tuna Saloona
3rd	Exclusively	Exclusively	Exclusively	Exclusively	Exclusively

Hidden Units: To determine the most effective multilayer architecture the number of hidden units must be determined. Through the experiments, we test the effect of the number of hidden neurons on the performances of the algorithms. It is noted that when the number of hidden units is too small there is a higher training error. When there are too many units in the hidden layer, the training error is low. A satisfactory outcome for BP algorithm is reached when the network structure is 8-13-1. In order to obtain a good convergence a bias node with a constant of 1 is used in both the input and hidden layers.

Learning rate: The learning rate for BP is the parameter which controls the steep size when weights are iteratively adjusted. It determines how far to move in the direction of the gradient of the surface over the weight space defined by the error function. The learning rate, therefore, affects the convergence speed and accuracy of the prediction algorithm. We set the other parameters as a fixed value (such as momentum factor is 0.4) but adjust the learning rate to different values to test its influences on the prediction results. It is observed that the network produces the best prediction when the learning rate is 0.02.

Momentum factor: The momentum factor improves the speed of learning and the best value would avoid the side effect of skipping over the local minima [12]. It is found that the momentum value of 0.4 produces an optimal result.

5. Conclusions

We have investigated the performances of four supervised neural network algorithms, Back-propagation, Quasi_Newton, Levenberg-Marquardt and Conjugate Gradient Descent learning algorithms, and tested their prediction power in horse racing. The experimental results demonstrate that all algorithms can produce acceptable predictions with an accuracy of 74% by average. The performance of each algorithm has been influenced by the data selection as well as the number of epochs. An application, such as horse racing, is environmental dependent. It is partially observable, stochastic, episodic, dynamic, continuous.

All four algorithms produce very similar results with BP outperforms slightly better. But BP also suffers from a longer training time and more parameter selections.

6. References

[1] Rumelhart, D. E. and McClelland, J. L.: Parallel Distributed Processing, The MIT Press, 1986

[2] Byrd, R. H., Nocedal, J.: A tool for the analysis of quasi-Newton methods with application to unconstrained minimization. SIAM J. Numer. Anal, 26 (1989) pp. 727–739

[3] Lera, G.; Pinzolas, M.: Neighborhood based Levenberg-Marquardt algorithm for neural network training. IEEE Transactions on Neural Networks, Volume 13, Issue 5, (2002) 1200 - 1203

[4] Hagan, M.T., Demuth, H.B. and Beale, M.H.: Neural Network Design, Boston, MA: PWS Publishing, 1996.

[5] Baulch, M.: Using Machine Learning to Predict the results of Sporting Matches, School of Information Technology & Electrical Engineering, University of Queensland, 2001

[6] Chen, H., Buntin, P. *et. al*: Expert Prediction, Symbolic Learning and Neural Networks: An Experiment on Greyhound Racing, IEEE, EXPERT Vol. 9 No. 6, (1994) 21-27

[7] Tortello, Rebecca, *Jamaica Horse Racing History: The sport of Kings*, http://www.jamaica-gleaner.com/pages/history/story0078.html, 2003

[8] Predicting Thoroughbreds Finish Time with BrainMaker Neural Networks. http://www.calsci.com/Thoroughbreds.html

[9] BrainMaker Predicts the Order of Finish in Horseracing, http://www.calsci.com/Thoroughbreds1.html

[10] Nilsson, N.J.: Introduction to Machine Learning, Department of Computer Science, Stanford University, USA, 1996

[11] Evans, O.V.D.: Short-Term Currency forecasting using Neural Networks, ICL Systems Journal Vol. II, Iss. (1997) 1-17

[12] Smith, L.: An Introduction to Neural Networks. http://www.cs.stir.ac.uk/~lss/NNIntro/InvSlides.html.