Hindawi Publishing Corporation
Advances in Artificial Neural Systems
Volume 2011, Article ID 374816, 8 pages
doi:10.1155/2011/374816

Research Article

Multilayer Perceptron for Prediction of 2006 World Cup Football Game

Kou-Yuan Huang and Kai-Ju Chen

Department of Computer Science, National Chiao Tung University, 1001 University Road, Hsinchu 30010, Taiwan

Correspondence should be addressed to Kou-Yuan Huang, kyhuang@cs.nctu.edu.tw

Received 9 May 2011; Revised 9 September 2011; Accepted 23 September 2011

Academic Editor: Mohamed A. Zohdy

Multilayer perceptron (MLP) with back-propagation learning rule is adopted to predict the winning rates of two teams according to their official statistical data of 2006 World Cup Football Game at the previous stages. There are training samples from three classes: win, draw, and loss. At the new stage, new training samples are selected from the previous stages and are added to the training samples, then we retrain the neural network. It is a type of on-line learning. The 8 features are selected with ad hoc choice. We use the theorem of Mirchandani and Cao to determine the number of hidden nodes. And after the testing in the learning convergence, the MLP is determined as 8-2-3 model. The learning rate and momentum coefficient are determined in the cross-learning. The prediction accuracy achieves 75% if the draw games are excluded.

1. Introduction

Neural network methods had been used in the analysis of sport data and had good performance. Purucker employed the supervised and unsupervised neural networks to analyze and predict the winning rate of National Football League (NFL), and found that multilayer perceptron neural network (MLP) with supervised learning performed better than Kohonen's self-organizing map network with unsupervised learning [1]. Condon et al. used MPL to predict the score of a country which participated in the 1996 Summer Olympic Games [2]. And the result outperformed that of regression model. Rotshtein et al. used the fuzzy model with genetic algorithm and neural network to predict the football game of Finland [3]. Silva et al. used MLP to build the non-linear relationship between factors and swimming performance to estimate the performance of swimmers, and the difference between the true and the estimated result was low [4]. However, there are few discussions on the parameter determination of MLP in different applications. Here, we adopt the supervised multilayer perceptron neural network with error back-propagation learning rule (BP) to predict the winning rate of 2006 World Cup Football Game (WCFG). We use the theorem to determine the number of hidden nodes. Also we determine the learning rate and momentum coefficient by the less average time and deviation time in the cross-learning.

According to the schedule of 2006 WCFG, shown in Figure 1, there are 32 teams in this competition and overall 64 matches at 5 stages in this tournament from the beginning to the end. The competition rules in each stage are explained as follows.

(1) Stage 1 is the group match, also known as round robin tournament. There is no extending time after 90 minutes regular time. In this stage, there are 32 teams in 8 groups (Group A–H), each group has 4 teams, and each team plays 3 matches. There are 6 matches in each group and there are 8 groups, so totally it has 48 matches (Match 1–48) in stage 1. The criterion of gaining points is that winning one game has 3 points, drawing one game has 1 point, and losing one game has 0 point. After stage 1, two teams that have the higher points in each group enter the next stage. Table 1 lists the score table for 32 teams in 8 groups after 48 matches finished at stage 1.

(2) The competition rule of stages 2–5 is single elimination tournament. It is necessary to have penalty

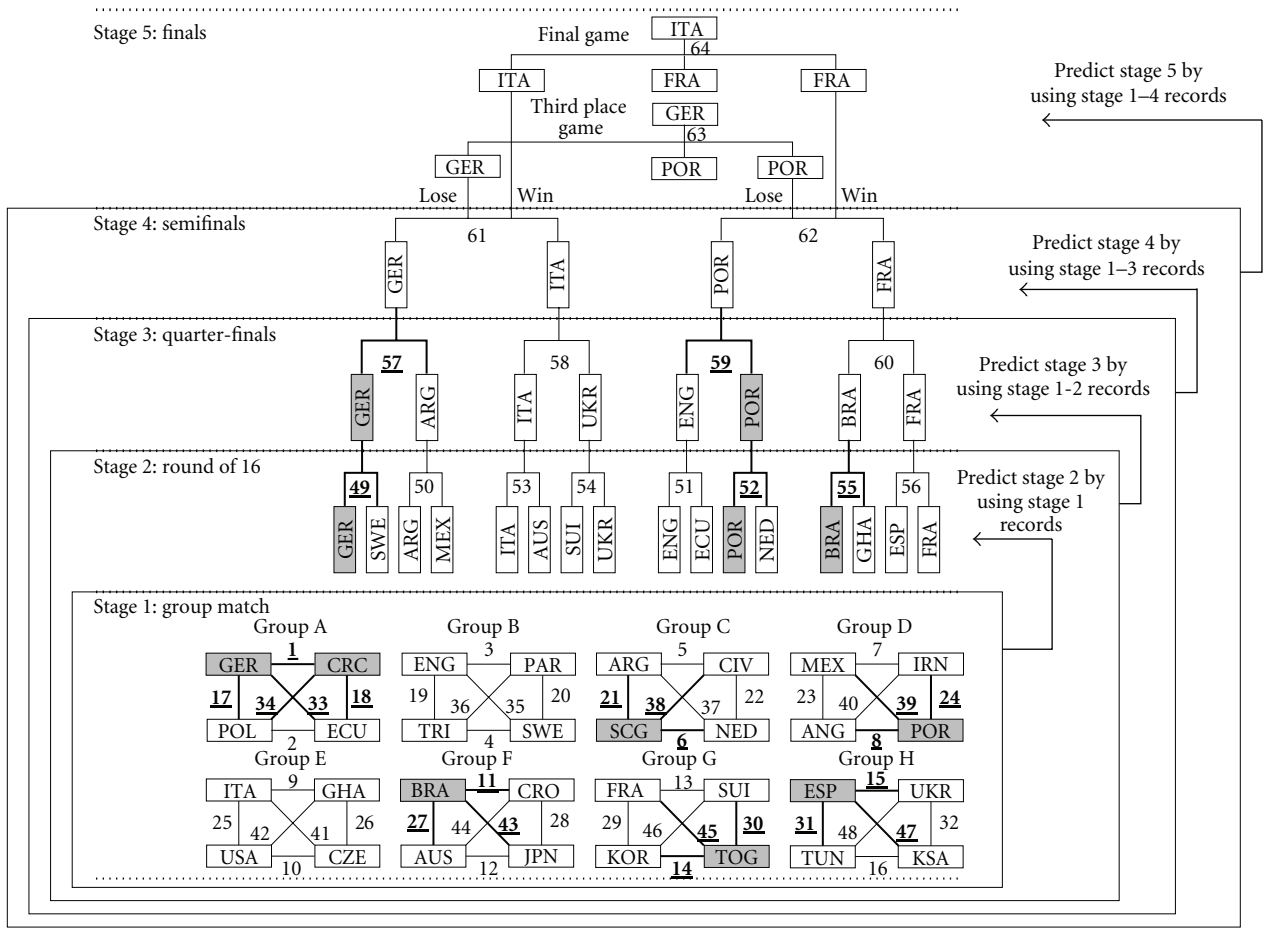

Figure 1: Total 64 matches at 5 stages for 32 teams in the schedule of 2006 WCFG.

kick if two teams tie after regular time (90 minutes) and additional time (30 minutes). The winner enters the next stage and the loser is eliminated from the competition. Stage 2 is the round of 16, and there are 8 matches (Match 49–56) for 16 teams. Stage 3 is the quarter-finals, and there are 4 matches (Match 57–60) for 8 teams. Stage 4 is the semifinals, and there are 2 matches (Match 61-62) for 4 teams. Stage 5 is the final-game, and there are 4 teams (the same teams as in stage 4) for 2 games. One is the third place game (Match 63), and the other is the final game (Match 64).

From the website of 2006 WCFG held in Germany [5], we can obtain the official 64 matches' statistical records provided by FIFA [6]. From the report of each match, there are 17 statistical items: goals for, goal against, shots, shot on goal, penalty kicks, fouls suffered, yellow cards, red cards, corner kicks, direct free kicks to goal, indirect free kicks to goal, offside, own goals, cautions, expulsions, ball possession, and foul committed, which represent the ability index to win the game. From these statistical data, we apply an MLP neural network to predict the winning rate of two teams at the next stage games (stage 2 to 5) by means of their statistic data from previous games. Figure 2 shows the supervised

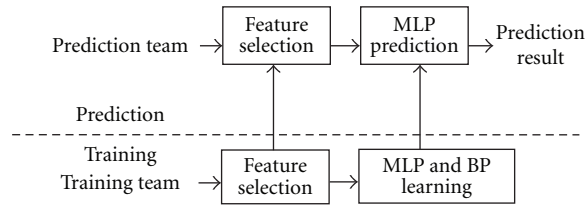

Figure 2: Supervised prediction system.

prediction system, which is composed of two parts: training part and prediction part. There are training samples from three classes: win, draw, and loss.

2. Feature Selection and Normalization

2.1. Feature Selection. We get 64 match reports from [5], and there are 17 statistical items in each match report. We select 8 items by an ad hoc choice, and they effectively represent the significant capability to win the game as the input features. Ad hoc choice is a common process to the real application of an algorithm. These 8 features are marked as x_1 = goals for (GF), x_2 = shots (S), x_3 = shots on goal (SOG), x_4 = corner

Table 1: Score table after finished 48 matches at stage 1.

Group	(team)	Win	Draw	Keys Loss	Play	Point
A	*Germany*	*3*	0	0	3	**9**
	Ecuador	2	0	1	3	6
	Poland	1	0	2	3	3
	Costa Rica	0	0	*3*	3	**0**
B	England	2	1	0	3	7
	Sweden	1	2	0	3	5
	Paraguay	1	0	2	3	3
	Trinidad and Tobago	0	1	2	3	1
C	Argentina	2	1	0	3	7
	Netherlands	2	1	0	3	7
	Côte d'Ivoire	1	0	2	3	3
	Serbia and Montenegro	0	0	*3*	3	**0**
D	*Portugal*	*3*	0	0	3	**9**
	Mexico	1	1	1	3	4
	Angola	0	2	1	3	2
	Iran	0	1	2	3	1
E	Italy	2	1	0	3	7
	Ghana	2	0	1	3	6
	Czech Republic	1	0	2	3	3
	USA	0	1	2	3	1
F	*Brazil*	*3*	0	0	3	**9**
	Australia	1	1	1	3	4
	Croatia	0	2	1	3	2
	Japan	0	1	2	3	1
G	Switzerland	2	1	0	3	7
	France	1	2	0	3	5
	Korea Republic	1	1	1	3	4
	Togo	0	0	*3*	3	**0**
H	*Spain*	*3*	0	0	3	**9**
	Ukraine	2	0	1	3	6
	Tunisia	0	1	2	3	1
	Saudi Arabia	0	1	2	3	1

kicks (CK), x_5 = direct free kicks to goal (DFKG), x_6 = indirect free kicks to goal (IDFKG), x_7 = ball possession (BP), and x_8 = fouls suffered (FS).

2.2. Normalization: Relative Ratio As Input Feature. We consider relative ratio as input feature. Training samples and prediction samples are normalized by relative ratio as follows:

$$\begin{aligned} &\text{If } x_{i\mathrm{A}} = x_{i\mathrm{B}}, \quad \text{then } y_{i\mathrm{A}} = y_{i\mathrm{B}} = 0.5 \\ &\text{else } y_{i\mathrm{A}} = \frac{x_{i\mathrm{A}}}{x_{i\mathrm{A}} + x_{i\mathrm{B}}}, \quad y_{i\mathrm{B}} = \frac{x_{i\mathrm{B}}}{x_{i\mathrm{A}} + x_{i\mathrm{B}}}, \quad i = 1, \ldots, 8. \end{aligned} \tag{1}$$

The input features of $x_1 - x_8$ are converted into $y_1 - y_8$ by (1), and then the $y_1 - y_8$ are fed into the neural network model for training or prediction. In (1), the symbol "A" indicates team A and the symbol "B" indicates team B. The symbol "i" is the index of 8 features. The input values of $y_1 - y_8$ are between 0–1 after normalization. We set "If $x_{i\mathrm{A}} = x_{i\mathrm{B}}$, then $y_{i\mathrm{A}} = y_{i\mathrm{B}} = 0.5$" that includes "if $x_{i\mathrm{A}} = x_{i\mathrm{B}} = 0$, then $y_{i\mathrm{A}} = y_{i\mathrm{B}} = 0.5$." The example of normalization result of Germany (GER) versus Costa Rica (CRC) is listed in Table 2.

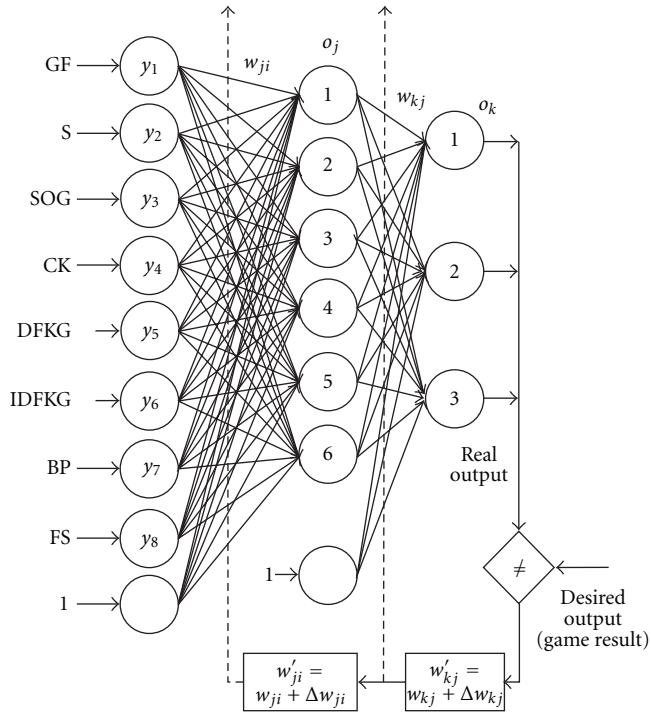

Figure 3: MLP network used in predicting the winning rate of 2006 WCFG.

3. Multilayer Perception with Back-Propagation Learning Algorithm

MLP model with BP learning algorithm is important since 1986 [7, 8]. The weighting coefficient adjustment can be referred to in [7–9]. Figure 3 shows the 8-6-3 MLP with one hidden layer used in this study to predict the winning rate of the football games. There are 8 inputs, 6 hidden nodes, and 3 outputs. Each symbol is explained as below: $\mathbf{y}$ is the input data vector with 8 features that have been normalized, w is the connection weights between nodes of two layers, net is the value which is the sum of the product of inputs and weighting coefficients, $f(\text{net})$ is the transfer function and the value is in $0 \sim 1$, o is the output value, d is the desired output, and e is the error value. The transfer function used in hidden layer and output layer is a log-sigmoid function, shown in (2). Using the least-squared error and with gradient descent method, we can get (3) and (4) for weighting coefficient adjustment. Considering the momentum term in the inertia effect of the previous step adjustment, the final adjustment equations are modified as (5) and (6), where η is the learning rate, t is the index of iteration, and β is the momentum coefficient:

$$f(\text{net}) = \frac{1}{1 + e^{-\text{net}}} \qquad (f \text{ is in } 0 \sim 1), \tag{2}$$

$$\Delta w_{kj} = \eta (d_k - o_k) f'_k(\text{net}_k) o_j, \tag{3}$$

Table 2: Data normalization of GER versus CRC.

Feature name	Features	Before Normalization		After Normalization	
		Team A (GER)	Team B (CRC)	Team A (GER)	Team B (CRC)
GF	x_1	4	2	0.6666	0.3333
S	x_2	21	4	0.84	0.16
SOG	x_3	10	2	0.8333	0.1666
CK	x_4	7	3	0.7	0.3
DFKG	x_5	1	0	1	0
IDFKG	x_6	0	0	0.5	0.5
BP	x_7	63%	37%	0.63	0.37
FS	x_8	12	11	0.5217	0.4782

$$\Delta w_{ji} = \eta \left[\sum_{k=1}^{K} (d_k - o_k) f_k'(\text{net}_k) w_{kj} \right] f_j'\left(\text{net}_j\right) o_i, \quad (4)$$

$$\Delta w_{kj}(t) = \eta (d_k - o_k) f_k'(\text{net}_k) o_j + \beta \Delta w_{kj}(t-1), \quad (5)$$

$$\Delta w_{ji}(t) = \eta \left[\sum_{k=1}^{K} (d_k - o_k) f_k'(\text{net}_k) w_{kj} \right] f_j'\left(\text{net}_j\right) o_i + \beta \Delta w_{ji}(t-1). \quad (6)$$

4. Training, MLP Model, and Prediction

4.1. Training Samples. At stage 1, we select the teams which win or lose all the three games as the training samples. The data are representative samples of the win or loss. But the selection is ad hoc also. They are italicized in Table 1. Also we select the teams which have the draw games. We set the desired output to 1-0-0 for winning the game, 0-1-0 is for the draw game, and set to 0-0-1 for losing the game.

From stage 2, the winning team's record will be added to training samples for all subsequent stages' training process only if the team had won three games at stage 1.

The selected training teams (background color is gray in Figure 1) and training matches (the bold line and bold number with under line in Figure 1) from stage 1 to stage 4 are as follows. Also, the draw games are selected as training samples (Match 4, 13, 16, 23, 25, 28, 29, 35, 37, 40, and 44). Games that end in penalty shoot-out after stage 2 are considered as draw games. Only the record of the regular 90 minutes and extension 30 minutes is calculated.

(1) The selected training samples for predicting the games in round of 16 (stage 2)—we select the teams whose score is either 9 (win 3 games) or 0 (lose 3 games) at stage 1 as the representatives of winning team or losing teams. Also, teams in draw games are selected as training samples. Then those game's records are normalized to relative ratios as the training data. Consequently, we can find 21 samples from the 20 matches of 7 teams as the training data. We have 4 teams with 3 wins: GER, POR, BRA, and ESP, and 3 teams with 3 losses: CRC, SCG, and TOG. GER and CRC have one match and their records are selected as two training samples. Besides, there are 11 draw games (Match 4, 13, 16, 23, 25, 28, 29, 35, 37, 40, and 44) in group matches. So there are 22 training samples for draw games. Totally, there are 43 (21 + 22 = 43) training samples.

(2) The selected training samples for predicting the games in quarter-finals (stage 3)—besides the 43 training data at the stage 1, we add the match data of stage 2 from those teams, which are not only the winner at stage 2, but also have all 3 wins at stage 1, as the training samples. We can find 3 teams' records (GER, POR, and BRA) as the training samples at stage 2. Also, there is a game (SUI versus UKR), which ends in penalty kick, and we consider it as a draw game. Therefore, currently we have 48 (43 + 3 + 2 = 48) training samples for the training to predict at stage 3. Although ITA is the winner at stage 2, it is not selected as the training sample. It is because ITA did not win 3 games at stage 1.

(3) The selected training samples for predicting the games in the semifinals (stage 4)—besides the above 48 training samples, we add 4 samples (GER, ARG, ENG, and POR) from 2 draw games which need penalty kick at stage 3 as the training samples. Therefore, we totally have 52 (48 + 4 = 52) training samples. ITA is the winner at stage 3, but it is also not selected as the training sample. It is because ITA did not win 3 games at stage 1.

(4) The training samples for predicting the games in the finals (stage 5)—because the two teams (ITA and FRA) do not have all 3 wins at stage 1, they are not selected as the training samples. The training samples at stage 5 are the same as that at stage 4 (52 training samples).

4.2. Input Team Data for Predicting. We do not have to predict the game result at stage 1, but records at stage 1 are extracted in order to predict the game results at the next stages. Therefore, the input data used to predict game results at stages 2–5 are described as follows:

(1) The input data for predicting round of 16 (stage 2)—we respectively take the *average* value from the 3 match records of each winning team that enters stage 2 as the input data. Totally, we get 16 input team data for the 8 games that we want to predict at stage 2. For

example, the input team data to predict the winner of GER versus SWE are listed in Table 3.

(2) The input data for predicting the quarter-finals (stage 3)—the input data are taken from the records of stages 1-2. We respectively take the average value from the 4 match records (3 games from stage 1 and 1 game from stage 2) of each team as the input data. Totally, there are 8 input team data for the 4 games that we want to predict in quarter-finals.

(3) The input data for predicting the semifinals (stage 4)—the input data are got from stage 1∼3. We respectively take the average value from the 5 match records of each team as the input data. Therefore, we totally get 4 input team data for the two games to predict result of semifinals.

(4) The input data for predicting the finals (stage 5)—the input data are got from stage 1∼4. We respectively take the average value from the 6 match records of each team, which have entered the final stage, as the input data. Therefore, totally 4 input team data are ready for the last two final games we want to predict.

4.3. Determination of the Number of Hidden Nodes by Theorem. Mirchandani and Cao [10] proposed a theorem that maximum number of separable regions (M) is a function of the number of hidden nodes (H), and input space dimension (d).

$$M(H,d) = \sum_{k=0}^{d} C(H,k), \quad \text{where } C(H,k) = 0,\ H < k. \quad (7)$$

There are total 52 training samples at stage 4 and the input dimension d is 8. Based on formula (7), when the network has 6 hidden nodes, it makes the maximum 64 separable regions:

$$M(6,8) = C(6,0) + C(6,1) + \cdots + C(6,8) = 2^6 = 64. \quad (8)$$

Therefore, 6 hidden nodes are sufficient. So from the theorem we adopt the 8-6-3 MLP model.

4.4. Cross Determination of Parameters for the Back-Propagation Learning. Kecman ever recommended the ranges of learning rate η and momentum coefficient β for BP learning [11]. Here we use cross-learning. We use the 43 training team samples selected from stage 1 in the training set. The cross procedures of determining the parameter settings for BP are listed in Table 4.

To determine the momentum coefficient β, we set hidden nodes = 6, mean square error (MSE) = 0.01, and fixed learning rate $\eta = 0.1$, then test five different momentum coefficients β ($\beta = 0.5, 0.6, 0.7, 0.8$, and 0.9). Each β setting is tested for 40 tests. Testing results are listed in Table 5. From Table 5, it shows that the standard deviation of convergent iterations at $\beta = 0.8$ is the smallest, which means the training process is more stable. Therefore, we decide to set momentum coefficient β to be 0.8 in the MLP model.

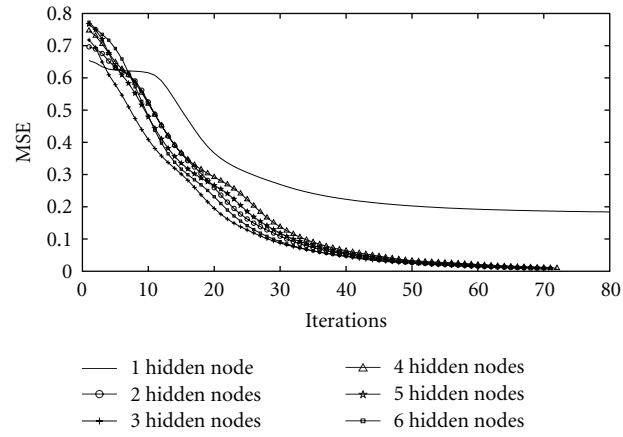

Figure 4: Plots of MSE versus iterations under 6 different hidden node MLP models. Set $\beta = 0.8$ and $\eta = 0.9$.

After setting the $\beta = 0.8$, we find the learning rate η next in this kind of cross-learning. We set hidden nodes = 6, MSE = 0.01, fixed $\beta = 0.8$, and then test five different learning rate η ($\eta = 0.1, 0.3, 0.5, 0.7$, and 0.9). Each η setting is tested for 40 tests. The testing result is listed in Table 6. From Table 6, we find out that when η is set as 0.9, the learning can have a less average convergent time than other η settings. Finally, using this systematic analysis, we decide to set the learning rate $\eta = 0.9$ and momentum coefficient $\beta = 0.8$ in the MLP model.

4.5. Refine the Number of Hidden Nodes. From a previous theorem, it needs 6 hidden nodes to converge for 52 training samples with 8 input dimensions. However, in practice, the number of required hidden nodes may be less than that in theory due to data distributions. It is worth checking the number of hidden nodes for MLP to converge in this application. Tests are made from 6 hidden nodes to 1 hidden node with total 52 training samples at stage 4. We decrease one hidden node at each time for learning convergent test. Figure 4 shows the plots of MSE versus iterations for 1–6 hidden nodes. The result shows that MLP converges in 80 iterations when 6, 5, 4, 3, and 2 hidden nodes are given. But it cannot converge with only one hidden node even after 10,000 iterations. For clear view, Figure 4 only shows first 80 iterations. According to this test, we conclude that it needs only two hidden nodes for MLP to train samples. We can infer that some training samples are grouped together, and we do not need the 6 hidden nodes.

5. Prediction Results

From previous analysis, the final prediction model used in this study is 8-2-3 MLP with BP learning. The parameter settings for BP learning are $\eta = 0.9$, $\beta = 0.8$, and MSE = 0.01. The prediction method is explained as follows.

We input the average data of each team into the well-trained MLP. Then we compare the output values of the first output node if two teams in a game to determine the win or

Table 3: Input data used to predict the winning rate of GER versus SWE at stage 2.

Features	Team A (GRE)				Team B (SWE)			
	Rec. 1	Rec. 2	Rec. 3	Average	Rec. 1	Rec. 2	Rec. 3	Average
GF	0.6664	1	1	0.8889	0.5	1	0.5	0.6667
S	0.84	0.7619	0.6818	0.7929	0.75	0.7692	0.4286	0.6493
SOG	0.8333	0.7619	0.8182	0.7612	0.75	0.5152	0.3913	0.5522
CK	0.7	0.7143	0.45	0.6214	0.4737	0.6667	0.6667	0.6023
DFKG	1	0.5	0	0.5	1	0.5	0.5	0.6667
IDFKG	0.5	0.5	0	0.3333	0.5	0.5	0.5	0.5
BP	0.63	0.58	0.43	0.5467	0.6	0.57	0.37	0.54
FS	0.5217	0.5526	0.538	0.5374	0	0.4412	0.4194	0.2868

Table 4: Procedures of determining the parameters for BP learning rule.

To determine parameters	Fixed conditions	Variable conditions	Observation items
Momentum coefficient β	(1) Hidden nodes = 6 (2) MSE = 0.01 (3) $\eta = 0.1$	(1) β = 0.5,0.6,0.7,0.8,0.9 (2) Testing times = 40	Average convergent iterations and standard deviation of convergent iterations
Learning rate η	(1) Hidden nodes = 6 (2) MSE = 0.01 (3) $\beta = 0.8$	(1) η = 0.1,0.3,0.5,0.7,0.9 (2) Testing times = 40	Average convergent iterations

Table 5: Average iterations and standard deviation of 40 tests under different settings of β. Set MSE = 0.01, hidden nodes = 6, $\eta = 0.1$.

β	0.5	0.6	0.7	0.8	0.9
Average convergent iterations	821.3	758.4	702.9	667.5	629.7
Standard deviation of convergent iterations	51.27	42.58	35.28	27.24	35.79

Table 6: Set MSE = 0.01, hidden nodes = 6, $\beta = 0.8$, then test average iterations of 40 tests with five different η settings.

η	0.1	0.3	0.5	0.7	0.9
Average convergent iterations	667	219	133.97	97.9	77.9

loss. The team with bigger output value of the first output node, meaning the greater ability to win the game, is the winner. The winning rate prediction results of two teams at each game from stage 2 to stage 5 are listed in Table 7. The symbol "W" means the team is the winner whose real output value of the first output node is bigger. The symbol "L" means that the team is the loser whose real output value of the first output node is smaller.

The prediction results must compare with the real game results. The symbol "Y" means that the prediction result is correct, and the symbol "N" means that the prediction result is wrong. The symbol "N/A" means that the prediction result is not counted because two teams draw.

From Table 7, we can see the percentage of prediction accuracy at stage 2 is 85.7% (6/7), and the percentages at the following stages are 50% (1/2), 50% (1/2), and 100% (1/1). Totally, there are nine correct predictions, three error predictions (N), and four noncounted real draw games (N/A) in 16 games.

The prediction for football games is not easy because the players use feet to control the ball. Too many factors and situations are sometimes changeable, and thus the game results are usually unpredictable. Scoring is not easy in football games, so there are many draw games in the records. Most of the time neural network can only predict the winner and loser. In fact, it is not easy to get an equal winning rate from the output value of the first output node of MLP for two teams, and to predict the draw game. So we exclude the draw games in the calculation of prediction accuracy. If we exclude four draw games (Match 54, 57, 59, 64) and calculate the average prediction accuracy of other 12 games from stage 2 to stage 5, the percentage of the prediction accuracy is 75% (9/12).

The odds can be calculated from the MLP for betting reference and its formula is defined as follows:

$$\text{Odd(Team\_A versus Team\_B)} = \frac{O_B}{O_A}, \tag{9}$$

where O_A and O_B are the real outputs from the first output node of MLP for team A and team B. The odds for team B versus team A are reversed. The results of odds are also shown in Table 7.

6. Conclusions and Discussions

In this study, we adopt multilayer perceptron with back-propagation learning to predict the winning rate of 2006 WCFG. We select 8 significant statistical records from 17 official records of 2006 WCFG. The 8 records of each team are transformed into relative ratio values with another team. Then the average ratio values of each team at previous stages

Table 7: Prediction results of winning rates from stage 2 to stage 5.

Stage	Match	Team	Outputs from MLP			Prediction result	Game result	Prediction correct	Prediction accuracy	Odds
			Node 1 (win)	Node 2 (draw)	Node 3 (lose)					
2	49	GRE	0.9914	0.2010	0	W	W	Y	85.7% (6/7)	0.79
		SWE	0.7866	0.3230	0.0001	L	L			1.26
	50	ARG	0.9604	0.0799	0	W	W	Y		0.04
		MEX	0.0338	0.9631	0.0102	L	L			28.4
	51	ENG	0.8325	0.2620	0.0001	W	W	Y		0.88
		ECU	0.7348	0.3779	0.0001	L	L			1.13
	52	POR	0.9909	0.0221	0	W	W	Y		0.97
		NED	0.9570	0.0864	0	L	L			1.04
	53	ITA	0.9856	0.0330	0	W	W	Y		0.01
		AUS	0.0129	0.9786	0.0329	L	L			76.5
	54	SUI	0.9843	0.0353	0	W	D	N/A		0.79
		UKR	0.7823	0.3176	0.0001	L	D			1.26
	55	BRA	0.9912	0.0212	0	W	W	Y		0.22
		GHA	0.2219	0.8161	0.0013	L	L			4.47
	56	ESP	0.9914	0.0209	0	W	L	N		0.89
		FRA	0.8848	0.1954	0.0001	L	W			1.12
3	57	GER	0.9878	0.0169	0.0001	W	D	N/A	50% (1/2)	0.91
		ARG	0.9017	0.1382	0.0003	L	D			1.10
	58	ITA	0.9841	0.0220	0.0001	W	W	Y		0.33
		UKR	0.3225	0.7406	0.0031	L	L			3.05
	59	ENG	0.9546	0.0639	0.0002	L	D	N/A		1.03
		POR	0.9868	0.0182	0.0001	W	D			0.97
	60	BRA	0.9874	0.0175	0.0001	W	L	N		0.88
		FRA	0.8703	0.1784	0.0005	L	W			1.13
4	61	GER	0.9864	0.0252	0	L	L	Y	50% (1/2)	1.0008
		ITA	0.9872	0.0238	0	W	W			0.9992
	62	POR	0.9843	0.0289	0	W	L	N		0.98
		FRA	0.9653	0.0608	0	L	W			1.02
5	63	GER	0.9372	0.1354	0	W	W	Y	100% (1/1)	0.98
		POR	0.9221	0.1628	0	L	L			1.02
	64	ITA	0.9917	0.0257	0	W	D	N/A		0.99
		FRA	0.9846	0.0433	0	L	D			1.01

are fed into 8-2-3 MLP for predicting the win and loss. The teams of 3 wins, 3 losses, and draws at stage 1 are selected as the training samples. The 8 records and training samples are selected by ad hoc choice. It is a common process to the real application of an algorithm. New training samples are added to the training set of the previous stages, and then we retrain the neural network. It is a type of on-line learning. The learning rate and the momentum coefficient are determined by the less average and deviation time in the cross-learning.

We use the theorem of Mirchandani and Cao to determine the number of hidden nodes. It is 6, and the MLP model is 8-6-3. After the testing in the learning convergence, the MLP is determined as 8-2-3 model. We can infer that some training samples are grouped together and we do not need the 6 hidden nodes.

If the draw games are excluded, the prediction accuracy can achieve 75% (9/12).

The 8 features are selected in ad hoc choice. But if we want to select 8 best features, we must work on C(17,8) combinations in analysis. We can select the feature set such that the error is the smallest or the distance measure is the maximum. Usually we may use divergence computation, Bhattacharyya distance, Matusita distance, and Kolmogorov distance in the use of distance measures for feature selection [12]. Also we can use entropy in the feature selection [12].

There are other methods: conjugate gradient method, Levenberg-Mardquardt method, simulated annealing, and genetic algorithm that can be used in the learning [13–17]. But MLP with BP learning is simpler in the determination of hidden node number, parameter setting, and the observation

of learning convergence in this application. Other pattern classification methods may be used for comparison in prediction accuracy [12, 18].

From pattern recognition point of view, compared with the two class training sets (win and loss), the three class training sets (win, draw, and loss) can have the more reliable prediction accuracy, because the decision regions or boundaries after training can be more precise.

Acknowledgments

The authors would like to thank the reviewer for his suggestion of selecting draw teams into the training sets to improve the prediction accuracy. The authors also thank Mr. Wen-Lung Chang for his collection of data.

References

[1] M. C. Purucker, "Neural network quarterbacking," *IEEE Potentials*, vol. 15, no. 3, pp. 9–15, 1996.

[2] E. M. Condon, B. L. Golden, and E. A. Wasil, "Predicting the success of nations at the Summer Olympics using neural networks," *Computers and Operations Research*, vol. 26, no. 13, pp. 1243–1265, 1999.

[3] A. P. Rotshtein, M. Posner, and A. B. Rakityanskaya, "Football predictions based on a fuzzy model with genetic and neural tuning," *Cybernetics and Systems Analysis*, vol. 41, no. 4, pp. 619–630, 2005.

[4] A. J. Silva, A. M. Costa, P. M. Oliveira et al., "The use of neural network technology to model swimming performance," *Journal of Sports Science and Medicine*, vol. 6, no. 1, pp. 117–125, 2007.

[5] "FIFA World Cup Germany—match schedule, matches and results, and statistics reports," 2006, http://fifaworldcup.yahoo.com.

[6] Official website of FIFA, http://www.fifa.com .

[7] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[8] D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1, MIT Press, Cambridge, Mass, USA, 1986.

[9] K.-Y. Huang, *Neural Networks and Pattern Recognition*, Weikeg Publishing Co., Taipei, Taiwan, 2003.

[10] G. Mirchandani and W. Cao, "On hidden nodes for neural nets," *IEEE Transactions on Circuits and Systems*, vol. 36, no. 5, pp. 661–664, 1989.

[11] V. Kecman, *Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models*, MIT Press, Cambridge, Mass, USA, 2001.

[12] K. Fukunaga, *Statistical Pattern Recognition*, Academic Press, New York, NY, USA, 2nd edition, 1990.

[13] F. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Networks*, vol. 6, no. 4, pp. 525–533, 1993.

[14] M. T. Hagan, H. B. Demuth, and M. Beale, *Neural Network Design*, PWS Publishing Company, Boston, Mass, USA, 1996.

[15] S. Haykin, *Neural Networks and Learning Machine*, Pearson Education, 3rd edition, 2009.

[16] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[17] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.

[18] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Elsevier, New York, NY, USA, 4th edition, 2009.