Learning from Recorded Games: A Scoring Policy for Simulated Soccer Agents

Achim Rettinger[1]

Abstract.

This paper outlines the implementation of a new scoring policy for the agents of the Simulated Robot Soccer team from the University of Koblenz, called RoboLog. The applied technique is capable of acting in real time in the dynamic environment of the RoboCup Simulation League and uses data obtained from prerecorded soccer games for supervised neural network learning. The benchmark used for testing this approach is the Optimal Scoring Problem stated as finding the point in the goal where the probability of scoring is the highest when the ball is shot to this point in a given situation. Goalshot situations from numerous logfiles are extracted and employed for the training of two independent multi layered perceptrons. Beside the usage as training patterns the gained data is evaluated statistically and provides interesting general insights into goalshots carried out lately in Simulated Robot Soccer.

The results obtained after extensive testing of the new policy are presented. Furthermore, general issues of learning from observed logfile data and starting points for future work are discussed.

1 INTRODUCTION

Scoring goals is essential for winning games not only in real soccer but also in the RoboCup Simulated Soccer League. The purpose of the RoboCup Simulated Soccer League is to provide a standardized problem domain for Artificial Intelligence research based on a soccer simulation called the RoboCup Soccer Server [2]. Teams of soccer agents programmed by researchers from all over the world can compete with each other by using this simulator.

This paper outlines the implementation of a new scoring policy for the RoboLog team from the University of Koblenz. Searching for a scoring policy is a comparably simple task. Although the properties of the environment provided by the RoboCup Soccer Server are inaccessible, non-deterministic, dynamic and continuous (see [6]), the success of a goalshot can directly be evaluated. In most other problems within the RoboCup domain the outcome of actions cannot be estimated as simple because the actions only result in intermediate and therefore not easily evaluable states. Contrary to that, a goal is definitely a success for the attacking team and final reward can be assigned. This makes the Optimal Scoring Problem a well suited benchmark for various techniques.

Accordingly, we chose the Optimal Scoring Problem for evaluating the innovative use of supervised learning from existing data. The data needed for this kind of inductive learning was obtained by analyzing relevant situations in prerecorded games. The automatically learned heuristic was intended to replace the analytical algorithm applied so far in the RoboLog team which based its decision whether to shoot and where solely on human consideration. In the old approach, manually adjusted thresholds gave the striker the positions, relative to goal and opponents, in which he was supposed to shoot.

[1] University of Koblenz, Koblenz, Germany email: achim@uni-koblenz.de

1.1 Problem statement

The Optimal Scoring Problem is stated as follows (see [4]): *"Find the point in the goal where the probability of scoring is the highest when the ball is shot to this point in a given situation.*"

When observed in more detail another side of this problem appears to be essential for finding an optimal scoring policy: *Given the point to shoot, determine the probability of scoring if the ball is shot to this point in a given situation.* Although, this heuristic is especially interesting for deciding whether to shoot or not, it is not mandatory for finding the point to shoot in our approach.

Both problems can be correlated to each other. If you can solve the first problem you know which point to test for the second problem. But if you can solve the second problem you can also find a good solution to the first problem by comparing numerous different points and taking the one with the highest probability of scoring. Thus, the second problem seems to be an intermediate step to solve the first statement of the Optimal Scoring Problem.

In this paper solutions to both problems will be presented which are not dependent on each other.

1.2 Related work

As the Optimal Scoring Problem is well suited for Machine Learning techniques previous work has been carried out in this area.

A detailed implementation of the scoring policy used by the UvA Trilearn 2001 team is described in [4]. Here, data is generated from repeated experiments where a striker is placed somewhere in front of the goal, the opponent goalie somewhere in the goal. Then the ball is shot to some position in the goal. The outcome of this shot is evaluated statistically. In the end, a function is presented that can calculate the probability of scoring if shot to a given point in the goal. Finally, the best point to shoot at is determined by computing the probability for some discretized scoring points on the goal line and by choosing the global maximum of the results.

In comparions to [4] our approach differs in three major points. First, the training data is not generated by simulating situations but by extracting already existing data from prerecorded soccer games (logfiles). Second, far more influencing factors of a goalshot situation, not just one forward and one goalie are taken into account. Third, two separate modules are developed to solve both in section 1.1 mentioned problems independently from each other. Thus, there is no need for testing discretized shooting points.

In [1] high level actions are based on Neural Networks which are trained to learn success rates. In this case the ”shoot2goal” action will compute the probability of scoring which is later on used for decision support by ranking the success rates of all actions in a priority list. This paper does not mention how to find the best point to shoot at. Again, training data was obtained by repeated generations of situations.

In contrast to that, a tool for the analysis of games played by a certain soccer team is presented in [5]. Special game situations (like goalshots) are identfied in logfiles on this selected team only. The patterns obtained are fed into a decision tree induction algorithm resulting in a set of rules which describe classes of successful scoring attempts and classes of unsuccessful attempts, respectively. Afterwards, those rules are used for a perturbation analysis that can give recommendations for changes in the goalshot heuristic used in this certain team.

Although logfiles were used for obtaining data in [5] and, among others, goalshot situations were extracted the crucial difference to the method outlined here is that the knowledge obtained was used for recommending changes to an already existing behavior (like the scoring policy) of a certain team. In contrast to that, we intended to find a universal and optimal scoring policy from scratch.

By combining data acquisition from logfiles with neural network learning two promising techniques are combined in the approach described in this paper. In addition to that, not only are success rates learned, but the best point in the goal to aim at is determined directly by a module independent from the success rate module. This redundantizes the test of several different scoring points as done in previous work.

2 APPLICATION

The application of our approach can be separated in three phases. First obtain the training data by extraction from logfiles, second analyze this data by supervised neural network learning and last evaluate the performance of the heuristic, in this case the feed forward networks.

2.1 Extraction of data

To obtain training samples, goalshot situations must be identified in logfiles. It is not enough to find successful scoring attempts because positive and negative training samples are required for classifying the success rate. The characteristics of a potential goalshot, identifiable from logfile data, are:

- A forward has kicked the ball.
- The forward is in a reasonable distance to the opponent goal.
- The shot has the potential to reach the opponent goal (reasonable power and direction).

Even if all those conditions apply, further tests need to be done to make sure that it is a valid goalshot and to obtain information about the outcome of this scoring attempt. To determine that, the successive cycles are scanned and checked individually:

- Can the situation be classified as goal, out, goalie catch or offsite? In this case it is a valid shot and the outcome is known.
- But, if the ball was kicked by another player it could also be classified as passing (if kicked by a player from the own team) or dribbling (if kicked by the same striker again) and thus not as a scoring attempt. If kicked by an opponent defender or opponent goalie though, it is interpreted as a valid but unsuccessful goalshot.

total		ratio	
games analyzed	996		
goalshots extracted	9315	shots/game	9.352
successful			
successful goalshots	3745	goals/game	3.760
unsuccessful			
intercepted by goalie	4305	goalie/game	4.322
intercepted by defender	993	defender/game	0.997
out	203	out/game	0.204
other reasons for miss	69	others/game	0.069

Table 1. Statistical evaluation of analyzed goalshots

All those heuristics can be no guarantee for identifying and classifying all scoring attempts correctly, as the internal state of the striker cannot be reconstructed from logfiles precisely. It is impossible to restore the intentions of a player in a specific situation only by observing the visual outcome of its actions. Nevertheless evaluation by hand showed that most of the shots a human observer would classify as scoring attempts were equally categorized by the automatic extractor. Besides that, the classification accuracy is, in that case, not essential for the purpose of neural network learning as long as the action holds valuable information. On this account, successfull shots, never intended to be scoring attempts (but e.g. passes), are important as well.

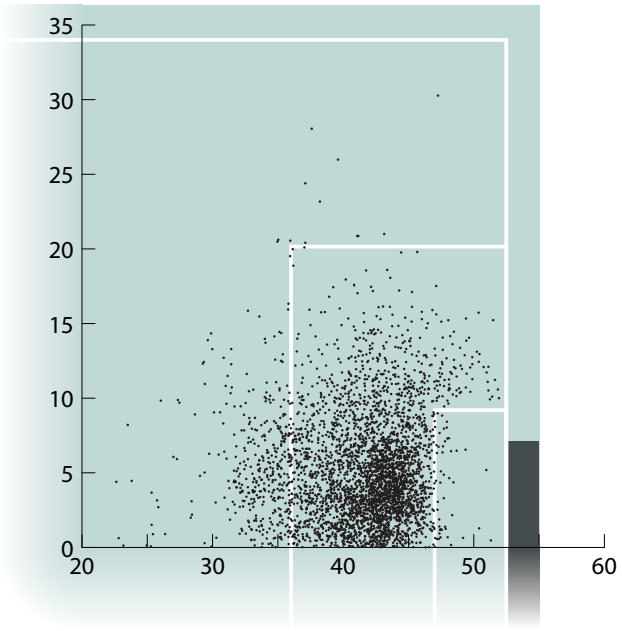

Figure 1. Position of striker while kicking; successful shots

Those heuristics were finally applied to all recorded games from the last RoboCup in Padua (2003) and games from the Simulated Soccer Internet League. Plenty of interesting information can be

gained from statistical analysis of the obtained goalshot situations. An overview is given in Tab 1.

It is interesting to know that 40.2% of the identified scoring attempts were successful and 77.3% of the unsuccessful ones were caught by the goalie. As expected, the goalie is the main factor in intercepting goalshots but it also becomes apparent that the opponent defenders should not be neglected. After all, they are responsible for 17.8% of the inhibited attempts.

The extracted goalshot data can give even more interesting insights. Fig 1 shows the upper right quarter of a soccer field when looked at in top view and landscape format. One half of the opponent goal is drawn as a filled black rectangle in portrait format at the lower right part of the figure. Accordingly, one of the corners is presumed in the top right. The white lines denote parts of the goal line, the side line, the goal area and the penalty area, respectively. The axis refer to the coordinates used in the Soccer Server. The scattered black dots indicate the position of the forward at that point in time when the successful goal kick was carried out.

In contrast to that, Fig 2 marks the position of the forward at the moment of a goal kick that turned out to be unsuccessful. Obviously the dots are spread more widely as expected.

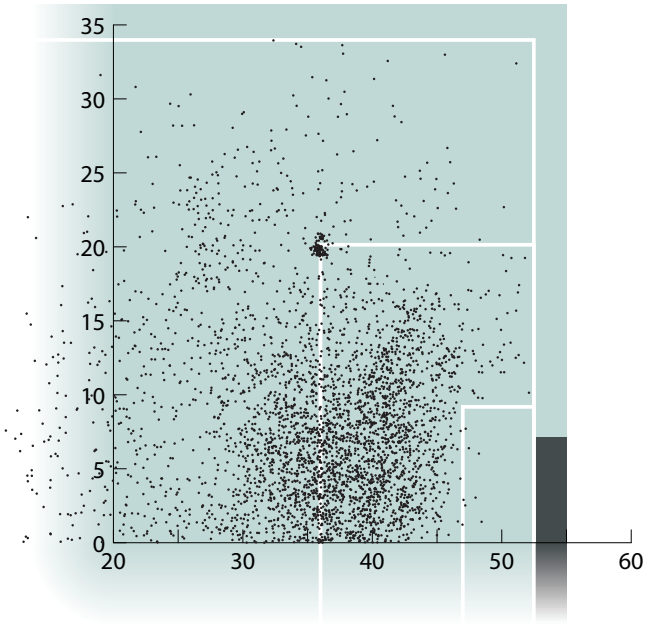

Figure 2. Position of striker while kicking; unsuccessful shots

Note that all goalshot situations are mirrored to this upper right quarter of the soccer field not only for visualization reasons, but mirroring is also essential for avoiding the aliasing problem. While training, the network could get confused if apparently different patterns have the same outcome, if mirrored.

Fig 3 shows where successful goalshots crossed the goal line. Darker areas denote more crossings. This time the goal is drawn in landscape format as a white rectangular boundary; scaling and mirroring is applied accordingly. As it can be easily seen, most of the shots were aimed at the corners of the goal, especially to the goal pole which was closer to the attacker.

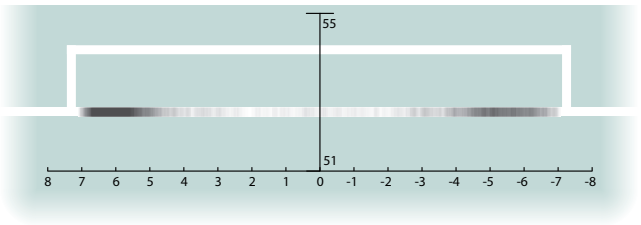

Figure 3. Goal line crossings: dark areas denote more crossings

2.2 Learning

As mentioned before, two basic 3-layered backpropagation neural networks were trained to solve the two tasks. The first network is required for predicting the point to shoot at that maximizes the likelihood of scoring in a specific situation. The second network should be able to classify the success rate of scoring, given a specific situation and the point to shoot. Diverse issues need to be addressed concerning the pragmatics of neural learning.

In the following, some considerations of the decisions that needed to be made shall be presented. One main issue is whether to use objective world data taken from the logfiles directly (accessible environment) instead of trying to simulate the subjective world model of a specific soccer agent (inaccessible environment). In the later case, the objective world data from logfiles like the exact ball position would have to be reduced und altered according to the limited subjective world model of an agent. On the one hand, it seems reasonable to use incomplete and noisy data for training because in a real simulated soccer game an agent would only get incomplete data as well. There is already previous work providing a method for estimating the internal state of RoboLog agents in a specific situation from logfile data only. Thus, it would be easy to use this data as input to the machine learning technique, every agent could be prepared with a specific decision module for its specific procedure of constructing its world knowledge. Unfortunately, it is still impossible to make sure that the reconstructed subjective world model precisely matches the original model from the recorded situation. Thus it is likely that the recorded action is not appropriate to the interpreted world model. Additionally, there is another fundamental shortcoming of using subjective data. As soon as the way a player constructs his world knowledge is changed, all the training needs to be redone. Therefore objective world data was used for learning to take advantage of this more general approach.

Another issue is the question which format of the input data would be the most suited one for this kind of problem. A polar representation of the positions was favored over a Cartesian representation because polar coordinates implicitly express relations between objects which could be more useful for the networks to generalize over the seen examples.

Besides that, the search for the most significant relations in the data remains a challenge, independent from the representation. As most design decisions involved in neural learning are still considered an empirical art (see [3]), the final selection and representation of inputs was found by comparing the results of numerous trained networks using three set cross-validation. A visualization of the final inputs is given in Fig 4. The indices refer to Fig 6. The attacker is drawn in yellow, the goalie in dark grey, the ball is a white circle. Defender 1 to Defender 3 (marked blue) are the three opponents which

can reach the ball first[2]. All variables were scaled to range between 0 and 1 and assigned to one input node each.

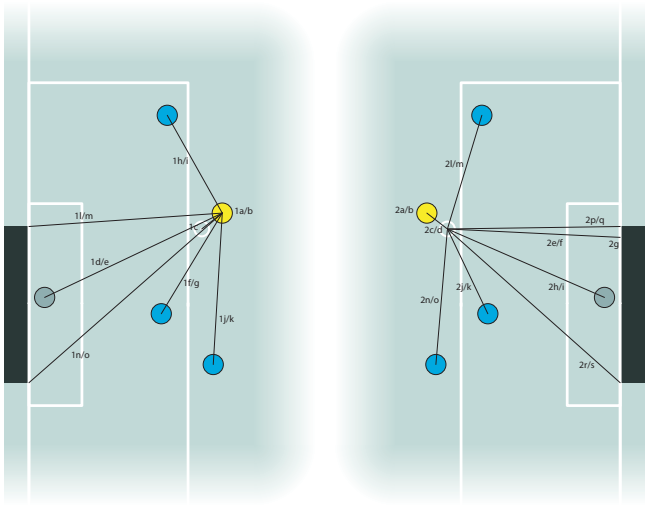

Figure 4. Visualization of input variables; left: best-scoring-point net, right: success-rate net

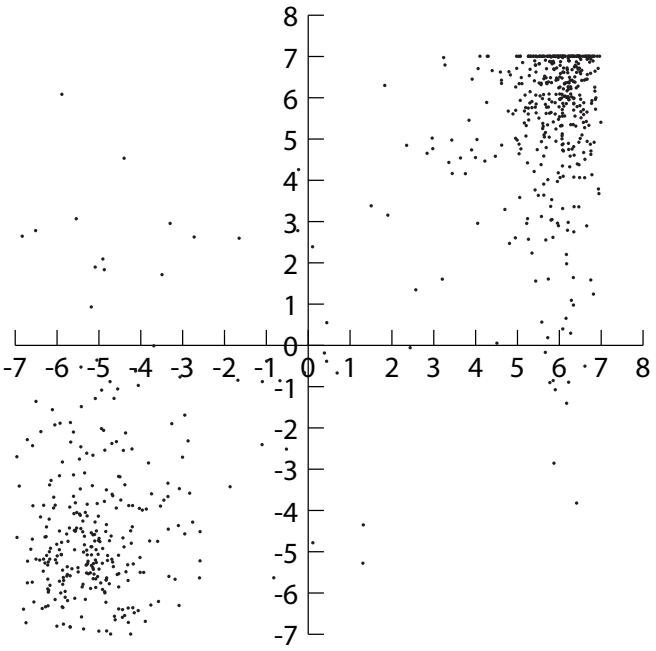

Figure 5. performance of the best-scoring-point net

The target value of the best-scoring-point network is the y-coordinate on the goal interval. The output, goal or no-goal, of the success-rate network is a binary class variable. As the classes are separable, two output nodes - one for goal and the other one for no goal - are used. In the end, both values are combined again to get a success rate between 0 and 1 by using the simple formula: $(((\text{output for no goal}) - (\text{output for goal}))/2) + 0.5$.

The final topology of the best-scoring-point network and the success-rate network derived from cross validation, is 15-53-1 and 19-80-2, respectively. For the first network only positive samples were used partitioned into three data sets for cross validation purposes (sample size: 2060, 936 and 748). The stratified sets of the second network contained equal proportions of positive and negative samples (sample size: 4494, 1693 and 1302). The stopping criteria is based on a calibration interval of 200 (total of training patterns processed per event), where training stops when the last minimum on the testing set (second data set) has occurred 50000 events ago.

After tweaking the various parameters involved in neural network learning, the prediction accuracy of both networks on the evaluation set (third data set) showed promising results.

The best-scoring-point network gave a mean average error of 1.4 units, which is reasonable if taken into account that the goal is more than 14 units wide. So the deviation on average is 10%. Fig 5 visualizes the performance using a scattered graph. The horizontal axis denotes the actual value and the vertical axis the interpreted output. Optimal predictions would result in a line from bottom left to top right. It can be observed that there is no bias towards a certain point in the predictions. The success-rate network achieved a remarkable 85.4% classification accuracy of the goal/no-goal patterns.

Calculating the contribution factor for each input variable is another way to get information about the networks' performance[3]. Fig 6 exemplarily shows the contribution factors for the success-rate network. "X" denotes a Cartesian x-coordinate and "Y" denotes a Cartesian y-coordinate, respectively. A "D" denotes distance in polar coordinates and "A" angle in polar coordinates. This nomenclature also corresponds to Fig 4.

Most of the values correspond to common sense. For instance the most important input is the angle and the distance between the ball and the goalie. Regarding the opponent defenders, the distance is more important than the angle and the closest defender has the biggest influence.

2.3 Evaluation

A feed forward version of both networks, using the learned weights, was finally integrated in the RoboLog framework. Two methods are provided to the soccer agents. If an agent decides that he is in a position where it makes sense to consider a goalshot he makes use of both methods. The first is required to find the best-scoring-point and the second to get the probability of scoring with this shot.

The threshold indicating whether an agent risks a shot or not has to be found by experiment. An observation that thereby needs to be taken into account is the following: The closer the striker gets to the goal the more unlikely it is that he can improve his position for a goalshot. This is because the resistance of the defenders is more concentrated around the goal. This fact cannot be taken into account by a neural network as used here. There is no temporal component that could give feedback about the quality of future states. Consequently, a region model was introduced. From the plot of the coordinates of successful goalshots (see Fig 1) we specified three regions according to the number of goalshots. Region 1 is closest to the goal and most of the goals were scored here. Region 2 is more spacious but still

[2] Those three defenders are determined by using a method from the RoboLog code, based on a Newton iteration.

[3] Contribution factors are a rough measure of the importance of a variable in predicting the network's output, relative to the other input variables in the same network.

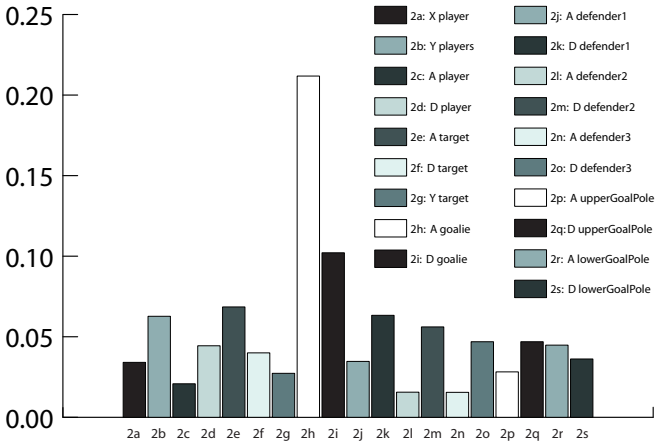

Figure 6. Contribution factors of the success-rate net

covers lots of goalshots. Region 3, finally, is the most wide-spread only containing a few positive data patterns (see Fig 7). From Region 3 down to Region 1 we gradually decreased the threshold that is used for the final decision whether to shoot or not. We observed that these rules could successfully prevented the agent from trying to score a goal when he is far from the goal and there is enough time and space to improve its scoring position.

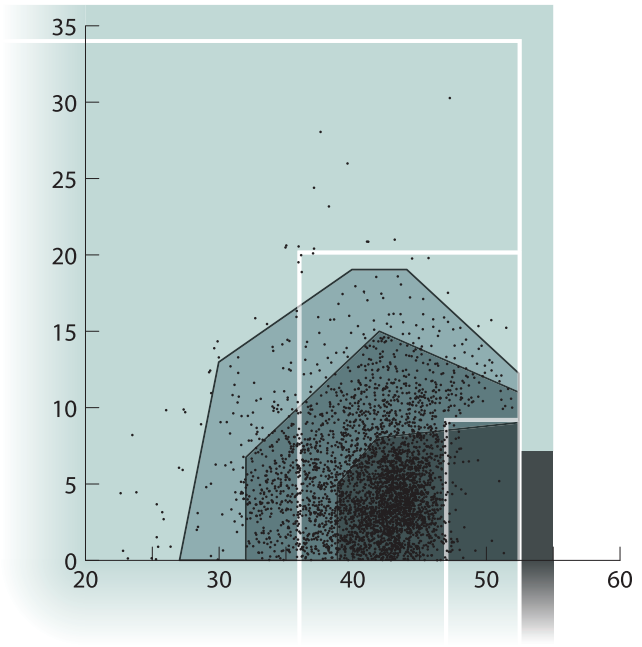

Figure 7. Zone model

For the purpose of finetuning and evaluating the performance, 400 test games were played. To allow for a meaningful comparison this was done by letting the conventional RoboLog team play against a RoboLog team with the new neural modules. As the performance of an agent is dependent on the available computational power the goalshot extraction heuristic described in section 2.1 was once again used for automatically evaluating all games played. Thus, the ratio of scored goals to scoring attempts could also be calculated and so the real performance without the deviation of server related performance losses could be determined.

Tab 2 shows the average performance over the 440 test games played. Statistics of all games played are summarized on the left half and the final 60 games on the right half of the table. This distinction is due to the fact, that we tried different parameter-settings for the networks and the zone model in the first 380 games. This resulted in fluctuations of the performance. The best set of parameters was finally used for the last 60 games.

	average test		average final	
	conventional	ANN	conventional	ANN
games total	440		60	
games won	121	130	11	21
ratio won/total	0.275	0.295	0.183	**0.35**
shots total	578	556	75	83
shots goals	172	216	23	33
ratio goals/shots	0.298	0.388	0.307	**0.398**

Table 2. Results for test phase and final settings

The ratio of successful shots to scoring attempts is significantly better for the team with the neural networks. Although most of the games still were a draw, in the end the new goalshot module could clearly outperform the conventional module by winning twice as often.

Another indication for the potentials of this approach is the performance of the RoboLog team at the RoboCup German Open 2004 where the new module was used in a competition for the first time. Even though, the overall results were not good and therefore don't look promising on the first sight the performance of the module becomes obvious after having a closer look at the logfiles. There were hardly any chances to score for the RoboLog team because the RoboLog strikers rarely got close to the opponent goal. So once more the ratio of goals to scoring attempts was calculated by using the goalshot extractor described in section 2.1. This more signifcant benchmark turned out to be exactly 50% which means that every second shot was successfull. In addition to that, a RoboLog agent for the first time managed to score against the Brainstormers04 team. Brainstormers04 ranked third in the end and conceded only two more goals in the whole competition.

3 CONCLUSION

This paper outlined a technique that uses data obtained from prerecorded soccer games for supervised neural network learning. The benchmark used for testing this approach is the Optimal Scoring Problem. The problem was tackled by decomposing it into two sub problems which where both individually addressed with one multilayered perceptron each, resulting in a variety of applications. The results show that observational learning from logfiles using neural networks delivers a promising performance while providing a series of advantages compared to previous work on this field.

- The time consuming step of generating training data from repeated experiments is not required.

- Training patterns obtained from prerecorded games provide universal information about the observed situations. The data is not limited by a specific agent used for data generation.
- The features used for training can be easily extended. There is no fundamental limitation due to complexity if further input variables are added to the networks.

Besides that, the extensive statistical analysis of goalshots provided in this paper should arouse interest of everyone dealing with simulated robot soccer.

3.1 Future work

It is obvious that the presented work is not able to provide a really optimal scoring policy.

First of all, there are starting points to improve the module without making fundamental changes. More sophisticated inputs to the networks, like speed, acceleration and body/view-angle of moving objects on the field, would most likely result in more accuracy. In addition, putting more effort into the learning process and providing more training data would also help the networks to better generalize over the seen samples.

But it becomes apparent, that all those straightforward improvements will never be able to result in an optimal scoring policy. To achieve that all future situations on the soccer field and the strategies of both teams would have to be taken into account. The work introduced in this project would in that case only provide a solution to a sub task in a broad decision support system. For a holistic solution, a prediction of the next actions of the opponent team (opponent modeling) and the own team needs to be made in order to set up an optimal scoring strategy. Only then, would it be possible to decide whether there will be a better position to score if the attacker performs some action, like dribbling first, and then tries to score instead of shooting right away. This cannot be achieved by a simple zone model with thresholds.

Steps in this direction could be to rank success rates of various actions of all team-mates (see [1]) or to use an auction protocol to decide on the next action to be carried out. However, it is very difficult to take all relevant future actions into account in this rapidly changing environment. But a one-step optimization has not the potential of being optimal.

Besides that, there are other interesting aspects to think about. Obtaining data by generation as done in previous work has the capability of finding situations not observed in prerecorded games. This could help to round off areas with sparse logfile training data or test unconventional strategies. Thus, if possible, a combination of both kinds of data acquisition seems to be the most promising approach.

Finally, a policy more specific to an opponent team could be advantageous and achieved in two ways. For one thing, training data from shots against one specific team could be emphasized in the training process to bias the network. This would result in a specific network for each opponent team. For another thing, online learning could most dynamically handle new situations and is as promising as challenging.

ACKNOWLEDGEMENTS

I would like to thank Oliver Obst for all his motivating support.

REFERENCES

[1] Sebastian Buck and Martin Riedmiller. *Learning Situation Dependent Success Rates Of Actions In A RoboCup Scenario*. In R. Mizoguchi and J. Slaney, editors, PRICAI 2000 Topics in Artificial Intelligence, number 1886 in Lecture Notes in Artificial Intelligence, Springer Verlag, page 809, 2000.

[2] Mao Chen, Klaus Dorer, Ehsan Foroughi, Fredrik Heintz, ZhanXiang Huang, Spiros Kapetanakis, Kostas Kostiadis, Johan Kummeneje, Jan Murray, Itsuki Noda, Oliver Obst, Pat Riley, Timo Steffens, Yi Wang and Xiang Yin: *Users Manual, RoboCup Soccer Server, for Soccer Server Version 7.07 and later*; February 11, 2003

[3] Vojislav Kecman. *Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models*, page 267, Bradford Book, MIT Press, Cambridge, Massachusetts, 2001.

[4] Jelle Kok, Remco de Boer and Nikos Vlassis. *Towards an optimal scoring policy for simulated soccer agents*. In M. Gini, W. Shen, C. Torras, and H. Yuasa, editors, Proc. 7th Int. Conf. on Intelligent Autonomous Systems, pages 195-198, IOS Press, California, March 2002. Also in G. Kaminka, P.U. Lima, and R. Rojas, editors, RoboCup 2002: Robot Soccer World Cup VI, Fukuoka, Japan, pages 296-303, Springer-Verlag, 2002.

[5] Tayler Raines, Milind Tambe and Stacy Marsella. *Towards Automated Team Analysis: A Machine Learning Approach*. Third international RoboCup competitions and workshop, 1999.

[6] Michael Wooldridge. *Intelligent Agents*. In Gerhard Weiss, editor, Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, page 30, MIT Press, Cambridge, Massachusetts, 2000.