

# Using Artificial Neural Networks for Pass Evaluation in Soccer Simulation

Sadegh Aliakbary, Seyed Jalal Kazemitabar, Hossein Hodjat, Hamidreza Amouzegar,

Mahdi Dashtbozorgi, Mahdy Khayyamian, Hamid Bagheri

**Kimia** Soccer Simulation Team

Department of Electrical and Computer Engineering

University of Tehran, Iran

{s.aliakbary, j.kazemitabar, h.hojjat, h.amouzegar, m.dashtbozorgi, m.khayyamian, h.bagheri}  
@ece.ut.ac.ir

**Abstract.** One of the most important characteristics of a Multi-agent system (MAS) is the agent collaboration. Soccer simulation is a multi-agent system in which the agents are "simulated soccer players". A soccer simulator is a research and educational tool for multi-agent systems and artificial intelligence. In this simulator, like the real soccer game, a soccer player collaborates with other teammates by passing to them. If a soccer player can evaluate the confidence of his passes before passing, he can always choose the best teammates for passing and so the overall performance of the team will improve. In this paper, we consider Artificial Neural Networks as a learning algorithm to estimate whether a particular teammate will receive a passed ball or not.

## 1 Introduction

In the process of passing the ball, the agent should be able to guess whether the pass to a particular teammate will succeed or not. In other words, according to the target condition and the situation of other opponents it should estimate the possibility of receiving the pass by the target teammate. In this paper we explain the method of **Kimia** soccer simulation team to construct a function which does this estimation.

## 2 The training procedure

Here we introduce a method to calculate the possibility of a pass success. We first calculate the danger of each opponent for the pass (By means of Neural Network Learning). Danger of each opponent is the possibility of his intercepting the ball before the target of the pass. Let  $(di)$  be the value in the range  $[0, 1]$  that represents this danger for opponent number  $i$ . So  $D = \text{Max}(di)$  is the maximum danger caused by opponents. The possibility of receiving the pass by the desired teammate will be  $1-D$ .

Now we use a Neural Network to learn a function to estimate the danger of one opponent  $(di)$  for a special pass and use this function to construct the final pass evaluation function as mentioned above. To train the Neural Network we need a training set. So we define a training scenario and use it to gather the training data.

The training procedure is as follow:

- 1 - The agents of each team come to the game field like an official match.
- 2 - Whenever the ball is in the kickable area of a teammate, he just passes the ball to a teammate. This teammate is selected randomly from teammates that the agent has a good knowledge of their status in the field.

3 - Passer announces his intention for the pass via a message. In this message the target of the pass is specified. The process of generating a training instance starts here.

4 - The specified teammate will try to intercept the ball, and other teammates will not affect the ball, even if the ball is in their kickable margin.

5 - The process of gathering a training instance will be finished whenever the target of the pass or an opponent has possessed the ball. (The passes which go out of field are ignored)

After finishing the process of gathering a training instance, the features and results are stored in a file and the process will be repeated to gather more training data. In this process we need an omniscient to see the world and collect the training set. We used the Offline Coach as the omniscient. He receives the information about the world without noise, and can collect the training set.

In the next chapter we explain the features which are considered in a training instance.

### 3 Inputs of the NN and its output

Assume  $d$  to be the vector that starts from the passer's position and ends to the receiver's position,  $opp$  to be the closest opponent to the ball **when the process of passing is finished**, and  $e$  to be the vector from passer to  $opp$ . (Usually only one opponent tries to intercept the pass, so for each pass we notice only this opponent. We assumed  $opp$  to be the opponent). These parameters are illustrated in Fig. 1.

The inputs of NN are as follows (All the features are calculated when the process of a pass is started):

1 - The distance between the passer and the receiver.

***dist (passer, receiver)***

2 - The distance between  $opp$  and the passer.

***dist (passer, opp)***

3 - The angle between  $d$  and  $e$ .

***angle (d,e)***

4 - The relative angle between body direction of receiver and  $d$ .

***bodyAngleDiff(receiver)***

5 - The velocity of receiver in the direction of  $d$ .

***velX(receiver)***

6 - The velocity of receiver vertical to  $d$ .

***velY(receiver)***

7 - The relative angle between body direction of  $opp$  and  $d$ .

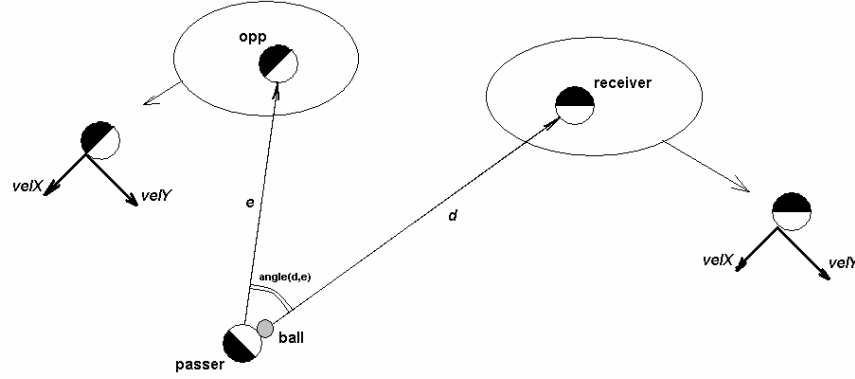
***bodyAngleDiff(opp)***

8 - The velocity of  $opp$  in the direction of  $d$ .

***velX(opp)***

9 - The velocity of  $opp$  vertical to  $d$ .

***velY(opp)***



**Fig. 2.** A pass situation and its parameters.

Surely we could add other features as the input of NN, but they are eliminated in order to simplify the network and improve its efficiency during the game.

The output of the network is:

$$\text{Output} = \exp(-\text{dist}) \quad (1)$$

where **dist** is the distance of **opp** to the ball when the process of passing is finished. It is obvious that instead of 0 and 1 (Success and Fail) we chose a continuous number between 0 and 1, which shows the possession of the ball by the opponent when the process of passing is finished. This will help NN to be trained better.

## 4 Training

We gathered 10,000 training instances in our training set. Large training set insures the different situations for the pass to be considered.

## 5 Why not analytic methods?

To construct a function for estimating the confidence of a pass, the analytic methods or use of lookup tables is not suitable. Because in this case, according to large number of affecting parameters, constructing an efficient function is so difficult and also time consuming. Moreover because of the existence of noise in simulation, learning algorithms will work better than the analytic methods.

## 6 Advantages of this method and comparison with other methods

In this section we compare our method with other methods such as *Pass Evaluation using Decision Tree learning* [stone 98], and explain advantages of our method.

1- The NN which is explained estimates the danger of each opponent for the desired pass. So during the use of the NN in official games, we should note two points:

a) Each opponent who has not been seen recently by the passer and has a low confidence in the passer's world-model will be ignored and assumed not to be dangerous for the success of the pass, because if he was dangerous, he should be around vector  $d$ . On the other hand, passer has watched the receiver and he has seen the receiver's surroundings. So the opponent is not around the receiver and can be safely ignored. In this case we don't use the NN to estimate the danger of this opponent, instead we return the result directly i.e. *dangerOfOpponent(opponent)* will return zero.

b) If *angle* ( $d, e$ ) for an opponent is more than a threshold (for example 50), or if *dist* (*passer, opp*)-*dist* (*receiver, opp*) is more than a threshold (for example 20) the opponent is assumed not to be dangerous for the pass. Here again we return the result directly instead of using NN to estimate the result.

These two points will make the function *dangerOfOpponent()* so efficient and will improve its speed in many cases.

2- In this method the training set is collected during a usual game, and the training scenario is not restricted to a specific situation. So the result will be practical and trustworthy during official games. Moreover we can use different teams as the opponent and make different NN weights for different teams. (The binaries of many teams are available in their sites!)

3- In this method the NN is very small, and its input is not very large. So the training precision will be nice, and its good speed makes it practical in official games.

4- Instead of answering "Success" or "Fail" to each training instance, we answer it with a real number between 0 and 1 which shows the possession of the ball by the opponent when the process of passing is finished. For example suppose the situation in which the target of the pass has received the ball. In this situation the distance of nearest opponent to the ball can be less than 0.5 meter or more than 20 meters. Both cases show a successful pass but they are so different in inputs. If we say just "Success", it makes the learning of the function hard.

5- In the pass evaluation method the Neural Network learning is better than other methods such as Decision Tree learning (See *Layered Learning Architecture* [Stone 98]) because of one more reason : Decision Tree learning outputs a source code (for example C++ code) but Neural Network learning outputs a data file (weights of edges of NN). When we use NN to learn our function we can learn it for different teams and in official games we can change this function just by changing the weights file. But if we learn the function by Decision Tree we should change our source code each time we want to change our method.

## 7 Conclusions and future work

This paper describes how neural networks can be used as a useful algorithm for pass evaluation. This algorithm has been developed by **Kimia** soccer simulation team. In the future we want to use Neural Networks for more effective pass algorithms, such as deep passes that have an important role in soccer matches.

## 8 Acknowledgement

This is a part of works done by the Simulation team in Tehran University. The whole work is directed by Professor Fattaneh Taghiyareh and we appreciate her for her untiring works as a leader.

## 9 References

1. Peter Stone, Layered Learning in Multi-Agent Systems, MIT Press , Carnegie Mellon University, 2000.
2. Remco de Boer, Jelle Kok, The Incremental Development of a Synthetic Multi-Agent System: The UVA-TriLearn 2001 Robotic Soccer Simulation Team, Master's thesis, University of Amsterdam, The Netherlands, February 2002.
3. S.Russell & P.Norvig, Artificial Intelligence a Modern Approach, Prentice Hall, 1995.
4. Jose C. Principe, Neural and Adaptive Systems: Fundamentals through Simulations, John Wiley and Sons, 2000.