Using Artificial Neural Networks for Pass Evaluation

Sadegh Aliakbary, Hossein Hojjat, Hamid Bagheri, Hamidreza Amouzegar,
Mahdi Khayyamian, Mahdi DashtBozorgi, Seyed Jalal Kazemitabar

Kimia Soccer Simulation Team
Department of Electrical and Computer Engineering
University of Tehran, Iran

{s.aliakbary, h.hojjat, h.bagheri, h.amouzegar, m.khayyam, m.dashtbozorgi, j.kazemitabar}
@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 "soccer players". 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 would improve.
In this paper , we consider Artificial Neural Networks as a learning algorithm to estimate whether a particular teammate will get a pass 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. Let (*di*) be a variable in the range [0, 1] that represents the danger of each opponent for passing, and so **D=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 for estimating the success of a pass between two teammates and this function will be used to construct the final pass evaluation function.

To gather the training data we first define a training scenario. The training procedure is as follow:

1 - The agents of each team come to the game pitch as an official match.

2 –Whenever the ball is in the kickable area of an agent, it 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 its 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.

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 and the process will be repeated to gather more training sets.

In this process we need an omniscient to see the world and collect the training set. We used an Offline Coach as the omniscient. Offline coach receives the information about the world without noise, and can save the training set.

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

3 - Inputs of the NN and it's output

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

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 relative angle between body direction of receiver and **d** .
bodyAngleDiff(receiver)

3-   The velocity of receiver in the direction of **d**.
velX(receiver)

4-   The velocity of receiver vertical to **d**.
velY(receiver)

5-   The distance between **opp** and the passer.
dist(passer,opp)

6-   The angle between **d** and **e**.
angle(d,e)

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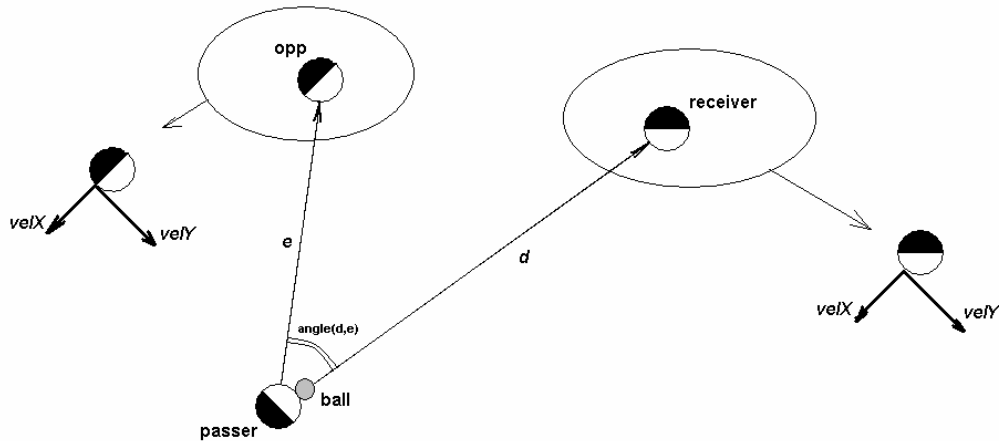
Surely we could add other features as the input of NN, but they are eliminated in order to simplify the network and improve it's efficiency during the game.

The output of the network is:

Output=exp(-dist)

where ***dist*** is the distance of ***opp*** to the ball when the process of passing is finished.
As you see 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.

3 - Training

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

4 - 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.

5 - Advantages of this 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 could note two points:

a)   Each opponent which is not seen recently, and it's confidence in the passer's world-model is not good, will be ignored and assumed not to be dangerous for the success of the pass. (Because if it was dangerous, it should be around the receiver of pass. On the other hand, passer has watched the receiver and it has seen the receiver's surroundings. So the opponent is not around of receiver and can be safely ignored). In this case we don't use the NN to estimate the likelihood of success of the pass, 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 40), 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 **o**f using NN to estimate the result.

These two points will make the function ***dangerOfOpponent( )*** so efficient and will improve it's 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 it's input is not very large. So the training precision will be nice, and its good speed makes it practical in official games.

6 – 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.

7 - Acknowledgement

8 – References

1   - Peter Stone . *Layered Learning in Multi-Agent Systems.* PHD Thesis , Carnegie Mellon University

2   - Remco de Boer,Jelle Kok . *The Incremental Development of a Synthetic Multi-Agent System : The UVA-TriLearn 2001 Robotic Soccer Simulation Team* . Master Thesis, University of Amsterdam

3   - S.Russell & P.Norvig . *Artificial Intelligence A Modern Approach*. Prentice Hall

4   - Principe . *Neural and Adaptive Systems: Fundamentals Through Simulations*. John Wiley and Sons