

NEURAL NETWORKS FOR SELF LOCALISATION

AN APPLICATION TO ROBOCUP DATA

Western Sydney University: 200045 Quantitative Project

James Monks, Nathan Villalobos

09. 09. 2019

Supervisor: Oliver Obst

Contents

1	Intro	1
1.1	RoboCup	1
1.2	History and Goals	2
1.3	Data	3
1.4	The Problem	4
2	Geometric Baseline	4
2.1	Geometric Localisation	4
2.2	Geometric Positioning of Other Players	6
3	Neural Networks for Localisation and Positioning	7
3.1	Neural Networks	7
3.2	Neural Networks for Self-Localisation - James Monks	8
3.3	Neural Networks for Player Positioning - Nathan Villalobos	8
4	Evaluation	8
4.1	Comparison of Models	8
4.2	Error Metrics	9
5	Considerations	10
5.1	Data Structuring	10
5.2	Error Compounding	10
	References	11

1 Intro

RoboCup, short for Robot World Cup, is an international competition held yearly that simulates a match of soccer using robots. This project will be using data from the RoboCup 2D soccer simulation league, which is a 2D virtual simulation of teams of 11 agents competing against each other in a match of soccer. There are two main objectives this project hopes to solve. The first is a problem of self-localisation, which involves creating a method for an agent to accurately measure and recall their location on the field. The second objective is to create a method to accurately locate the position of other agents on the field. Each match has two separate types of data. The “groundtruth” data displays the exact information of the players and the ball at every step of the game. The second type of data is player specific, and displays what each player perceives at any specific time with some level of perception error. Using player specific data, geometrical mathematics are used to attain self-localisation and other agent locations. Later in the project, neural networks will be utilized in tandem with 180 hours of game time data in an attempt to decrease prediction error.

1.1 RoboCup

RoboCup, short for Robot World Cup, is an international competition held yearly that simulates a match of soccer using robots. There are five major competition areas in RoboCup, with each having their own different leagues. This project will be using data from the RoboCup 2D soccer simulation league.

The 2D soccer simulation league is a virtual soccer competition between two teams of 11 virtual agents. Each match is 10 minutes long and played on a 2D virtual field that represents a soccer stadium. Even though it has no physical robots, viewers can watch on a screen as the programs compete. It uses the official soccer rules of FIFA with the exception of the time limit, which is 10 minutes long per match (Michael et al. 2017).

The agents play on a field that has two separate boundaries, each having different uses. The inner boundary in figure 1, which will henceforth be referred to as the “play boundary”, is the perimeter the ball must be within to be considered in play. The outer boundary is called the physical boundary, which is the area the robots’ movements are confined by, as well as the area they may sense objects within.

The world soccer association FIFA rules state that a standard soccer field is $105\text{m} \times 68\text{m}$.

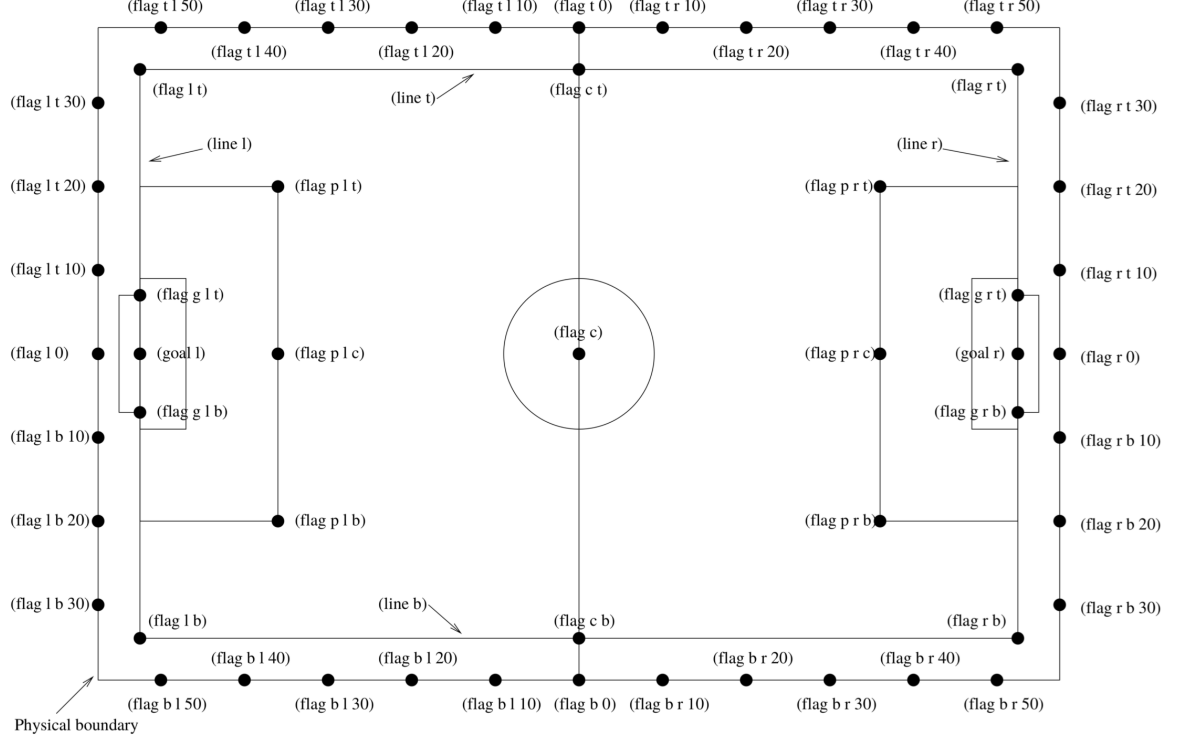


Figure 1: RoboCup Simulation League Field (Michael et al. 2017).

RoboCup has also added these parameters as the play boundary, while the physical boundary has measurements of $120\text{m} \times 80\text{m}$.

To self-locate, various landmarks are placed at set intervals which the “players” can see. The field is divided by left (l) and right(r), as well as top(t) and bottom (b), based on a horizontal and vertical axis cutting the centre (c). Each landmark is denoted by this division of the field, along with a select few on the goal (g), and the penalty box (p). Additionally, flags are marked along the physical boundary lines with numbers (10, 20, 30, 40, 50) that correlates the distance from the centre axis that connects to that specific line (Michael et al. 2017).

1.2 History and Goals

The first official RoboCup tournament, was announced in 1995 at Montreal, Canada, during the International Joint Conference on Artificial Intelligence (IJCAI-95). It was scheduled for 1997, giving researches two years of resource procurement, such as securing funding, as well as creating AI simulation teams. To reduce problems with the first official launch of RoboCup in 1997, it was decided a smaller scale competition consisting of eight teams competing in the simulation league, along with a demonstration of the middle league, would be held in 1996,

named Pre-RoboCup-96 (“A Brief History of Robocup,” n.d.).

RoboCup intends to propel society’s technology forward. To achieve this, they have set an end goal of “winning a soccer game against the winner of the most recent World Cup, complying with the official rules of FIFA, by the middle of the 21st century” (“Objective,” n.d.). While this goal may not impact society directly, the technology breakthroughs needed to achieve this goal will have a great impact on future technology, similar to IBM Deep Blue. In 1997, IBM Deep Blue, an AI program, won a six-game match of chess against the human world champion, Garry Kasparov. This sparked a revolution in computer science by advancing complex computer models and use of “big data,” which led to breakthroughs in many areas of civilization, including bank financial systems, creating new medical drugs, and targeted advertising (“Deep Blue,” n.d.). RoboCup hopes to achieve similar results in the field of AI robotics research.

1.3 Data

There are three separate packages of data this project works with. A package with one match, a package with 10 matches, and a package containing 10 teams playing 25 matches against each other. This creates a total of 45 unique pairings, resulting in a total of 1125 matches. All the teams used are from the top 10 teams in the world in 2016 and 2017 (Michael et al. 2017).

There are two types of data used per match. The first type, the “groundtruth” data, displays the exact information of the players and the ball at every step of the game. Steps are recorded every 100 milliseconds, meaning there are 10 steps per second, for a total of 6000 steps per match. This recording includes the exact position of the ball, the velocity of the ball for both x and y-axis, the score, player positions, the velocity of players for both x and y-axis, the direction a player is facing, and the angle a player can see. Additionally, it displays what the current state of play is (free-kick; kick in, which is the replacement of a throw-in from standard soccer etc.), as well as the current score.

The second type of data is player specific, and displays what each player perceives at any specific time. This data is split into two separate sheets. The first, named “landmarks,” displays the distance (meters), as well as the angle (degrees) to each landmark based on the relative bearing the robot is facing. The second, named “moving,” displays the distance and the angle to each movable object on the field in the specific players’ perspective. To further simulate humans, all player perspective data has some level of perception error based on distance away from the target.

The information a player receives, however, is limited by its field of view, meaning each player cannot see every action and interaction happening on the field at once. The field of view for each player can freely be changed between 60° , 120° , and 180° during a match at any given time, and players receive information corresponding to their current field of view angle. At 60° , the robot receives new visual information every step, with this increasing an extra step for each larger angle respectively. Additionally, any information the players cannot receive at any given step displays “NAN” in the sheet of data.

1.4 The Problem

This project aims to solve two major problems in the creation of simulated RoboCup agents. The first problem is self-localisation, which involves creating a method for the agent identify its current coordinates on a given playing field based on the surroundings. The second problem is to accurately determine the position of other players based on the same perception data. These problems can be solved geometrically using relatively simple mathematical concepts, however, there are problems with these methods that will be discussed. To improve on this, the project aims to solve the research question: “To what extent can neural networks be used to improve the positioning calculation problems presented in RoboCup data?”.

2 Geometric Baseline

2.1 Geometric Localisation

It is possible to obtain the coordinates of a player on the field through using the angle and distance from the player to known points. This method utilises cosine and sine rule to obtain the unknown dimensions of the triangle formed by the player and each flag. It then uses vector algebra to obtain the coordinates of the player.

Specifically, cosine rule is used to obtain the length of the vector $F_1\vec{F}_2$. This is then used as in input for sine rule to obtain the angle $\angle PF_1F_2$ and the angle $\angle PF_2F_1$, which results in all dimensions of the constructed triangle being formed. The vector $F_1\vec{F}_2$ is then rotated anti-clockwise around the point F_1 such that it is pointing in the direction of P . The unit vector in this direction is obtained and then multiplied by the scalar value of the length of the vector $\|P\vec{F}_1\|$.

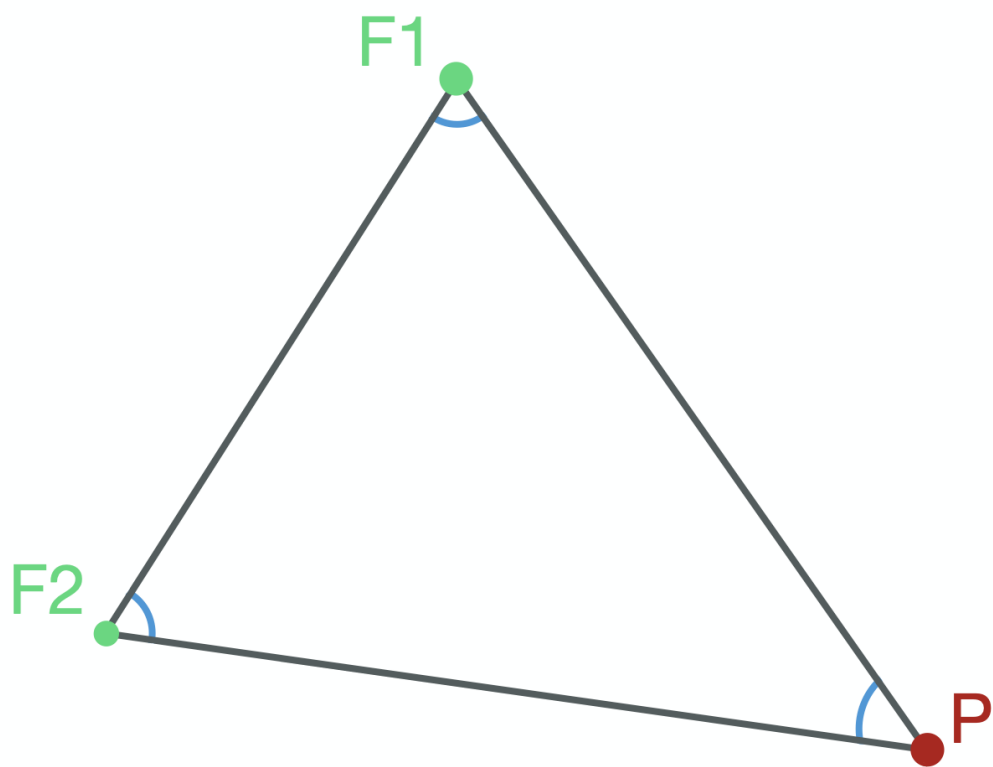


Figure 2: Illustration of the triangle formed by the player and two flags.

$$\begin{aligned}
& \text{let : } \alpha = \angle PF_1F_2 \\
& \text{and let : } \vec{v} = F_1\vec{F}_2 \cdot \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix} \\
& \text{then : } F_1\vec{P} = \frac{\vec{v}}{\|\vec{v}\|} \cdot \|F_1\vec{P}\|
\end{aligned}$$

This method is flawed in that it cannot predict where a player is if there are fewer than two flags that are in sight. It also assumes that the measurements in the data have no error associated with them, which is not true as discussed in section 2. This means that flags that are further away will produce worse results than those that are close. The method is also limited to utilising two and only two flags, which means there are flags that could offer some additional information being left out of the localisation prediction.

2.2 Geometric Positioning of Other Players

When referring to the positioning of other players when using the perception data from another, terminology needs to be defined to prevent ambiguity. The term “subject agent” will refer to the player of which the perception data is being used, and the term “target player” will refer to the player that is being located.

Determining the position of other players on the field utilises similar vector algebra to the method of geometric self-localisation. If the subject agent’s orientation and position are known, an absolute bearing from the subject agent to the target player can be determined using the relative angle and the subject agent orientation. Then the vector distance and the absolute bearing can be used to determine the position of the target player.

The problem of perception error and distance that was discussed in the context of geometric localisation are also applicable to this for similar reasons. There are additional limitations that arise in this case however, as the method can only make a prediction of target player coordinates if they are within the field of view. This means that it is not possible for a subject agent to know where most of the players on the field are.

3 Neural Networks for Localisation and Positioning

3.1 Neural Networks

Neural networks are a collection of methods for solving regression or classification problems (Tu 1996). They are a branch of machine learning that take inspiration from human brain neurons. As with most machine learning methods, data is used to train parameters that will be used to create predictions for new and unseen data.

The structure behind neural networks consists of a series of layers, with their corresponding “neuron”. The neurons’ function is to learn how to appropriately weigh inputs based on previous inputs and outputs. First is the input layer, which corresponds to input variables. Several factors, based on the information presented, are imported into the neural network via the input layer. The input layer has a neuron for every input variable. This then connects to an arbitrary number of hidden layers, which are fixed with weights and bias. The weightings of these hidden layers are equivalent to the parameters, like alpha and beta, in more traditional machine learning models. These weightings and bias influence the final prediction by determining whether to discard or operate the factors. Finally, there’s an output layer, with one output neuron per desired outputs, which the last hidden neurons feed into. This output neuron then gives a prediction (Titterton 2009).

Neural networks are beneficial for many reasons. They can learn complex functions intrinsic to data which would otherwise be difficult to implement, as well as they can include an arbitrary number of variables to inform the prediction. Another advantage of neural networks is their capability of detecting interactions between two or more variables in how they influence the prediction (Tu 1996).

Structural properties of neural networks can be manipulated in order to dictate how effectively they can learn higher order information encoded in the input variables. The depth of a neural network refers to the number of hidden layers between the input variables and the output number or class. The width of the neural network refers to the number of neurons in the hidden layers and effects the potential for the network to identify higher order variables embedded in the data.

3.2 Neural Networks for Self-Localisation - James Monks

Neural networks provide many advantages to the process of self-localisation over the previous trigonometry and vector based solution. This is because they allow for more than two flags being included in the prediction of the coordinates, which results in no useful information being discarded. This enables better predictions to be made, because allowances may be made for the error in perception. There is also the potential to include additional lagged variables for each of the flags in view, introducing information regarding the movement of the player through time. The optimal number of time lags to introduce while preventing overfitting problems may be investigated in the process of developing this model.

3.3 Neural Networks for Player Positioning - Nathan Villalobos

The advantage of using neural networks for identifying the positions of other players on the field is that there is the potential to utilise lagged values in the prediction of current and future positions. This method of prediction which can be trained on the over 180 hours of data from previous RoboCup matches opens up the possibility of predicting the location of a player that is outside of the field of view of the subject agent. There is also the possibility of calculating confidence intervals in the predictions, which can be used to identify the region that a target player is in with a high level of confidence. This represents a new capability that is not present in the simple geometric positioning techniques and may allowed for a more predictive method of play for the subject agent.

4 Evaluation

4.1 Comparison of Models

The neural network based methods will be compared to the original geometric implementations of both self-localisation and player positioning. In order to do this, a common error metric needs to be defined which is discussed in detail in section 4.2. This error metric may also be used as a loss function for the neural networks to optimise for, which will produce the best possible results.

The neural network based solutions to these problems will go through several iterations in order to improve performance. There are a variety of structural properties that can be changed that

will influence the way in which the neural network trains and its capacity to learn information. These properties include the number of hidden layers (or the depth of the network) and the number of neurons per layer (or the width of the network). These iterations will need to be evaluated against one another, as well as against the base line geometric predictions.

4.2 Error Metrics

If the neural networks are going to be compared to the baseline geometric measurements, a common error metric needs to be decided upon. The form of this error metric is important, as it will dictate what is seen as important in the data and will determine what the neural networks will optimise for. The sum of absolute errors is one form of this error metric, which will penalise all errors evenly. The sum of squared methods is another form that will penalise worse predictions more harshly. The latter also allows for gradient descent methods of optimisation to be used as it is a differentiable function, whereas the absolute error does not allow for this as it is not differentiable.

The quantitative comparison of the neural network based methods and the baseline geometric methods presents a problem in how we measure error. This is due to one of the benefits mentioned in the neural networks approaches of being able to make a prediction when it was previously impossible (in the case where there are less than two flags in view or the target player is outside the field of view). In these cases, there is no attempt at prediction in the geometric cases resulting in no additional error, however, in the neural network implementations a prediction will be made and the resultant error will be likely higher than average as there is less information. This is not a desirable quality in the localisation system.

A solution to this problem would be to give the errors generated by each method and then use as a secondary metric giving the number of failed predictions. Another way would be to only assess the predictions in instances where it is possible for both implementations to make a prediction. The problem with this method is that it discounts the value brought from the prediction of coordinates where it was previously impossible. This could be rectified through stating the errors obtained by the neural network in both situation. Finally, it is also possible to assign some form of penalty in the error calculation for failed predictions. This would require domain knowledge based justification as to what this penalty should be, along with evidence based optimisation.

5 Considerations

5.1 Data Structuring

The perception data is structured in such a way that it may prove difficult to include all of the relevant data in the models. This is due to the nature of the data, with flags dropping in and out of the field of view it is hard to encode relationships to specific flags, and may throw off the prediction if it is not correctly allowed for. This is particularly a problem when attempting to include the lagged value of flags. This is something that requires further investigation. One method of addressing this may be to use variables representing the flags in order of distance, and using additional variables to track whether the true id of these flags change.

5.2 Error Compounding

An important note is that the source of the error in the player positioning stage may not be coming from only the model being used and the perception errors. This is because player positioning relies on the current coordinates of the subject agent. If self-localisation coordinate predictions being used as inputs for player positioning, the error of the initial self-localisation will be compounded with the error in the player positioning model. To mitigate this issue and assess the error of each method independently, the ground truth will be used as the input coordinates for the player through training and evaluation. This can be thought of as assuming the self-localisation has been done perfectly in order to assess the error of the positioning model independently.

References

- “A Brief History of Robocup.” n.d. *RoboCup Federation Official Website*. https://www.robocup.org/a_brief_history_of_robocup.
- “Deep Blue.” n.d. *IBM100 - Deep Blue*. <https://www.ibm.com/ibm/history/ibm100/us/en/icons/deepblue/>.
- Michael, Obst, Oliver, Schmidberger, Falk, and Frieder. 2017. “RoboCupSimData: A Robocup Soccer Research Dataset.” *arXiv.org*. <https://arxiv.org/abs/1711.01703>.
- “Objective.” n.d. *RoboCup Federation Official Website*. <https://www.robocup.org/objective>.
- Titterington, Michael. 2009. “Neural Networks - Titterington - 2010 - Wiley Interdisciplinary Reviews: Computational Statistics - Wiley Online Library.” *Wiley Interdisciplinary Reviews: Computational Statistics*. John Wiley & Sons, Ltd. <https://onlinelibrary.wiley.com/doi/epdf/10.1002/wics.50>.
- Tu, Jack V. 1996. “Advantages and Disadvantages of Using Artificial Neural Networks Versus Logistic Regression for Predicting Medical Outcomes.” *Journal of Clinical Epidemiology* 49 (11): 1225–31. [https://doi.org/10.1016/s0895-4356\(96\)00002-9](https://doi.org/10.1016/s0895-4356(96)00002-9).