Applying Evolutionary Neural Networks to Robotic Soccer

By Oliver Chamberlain

Robotic Soccer

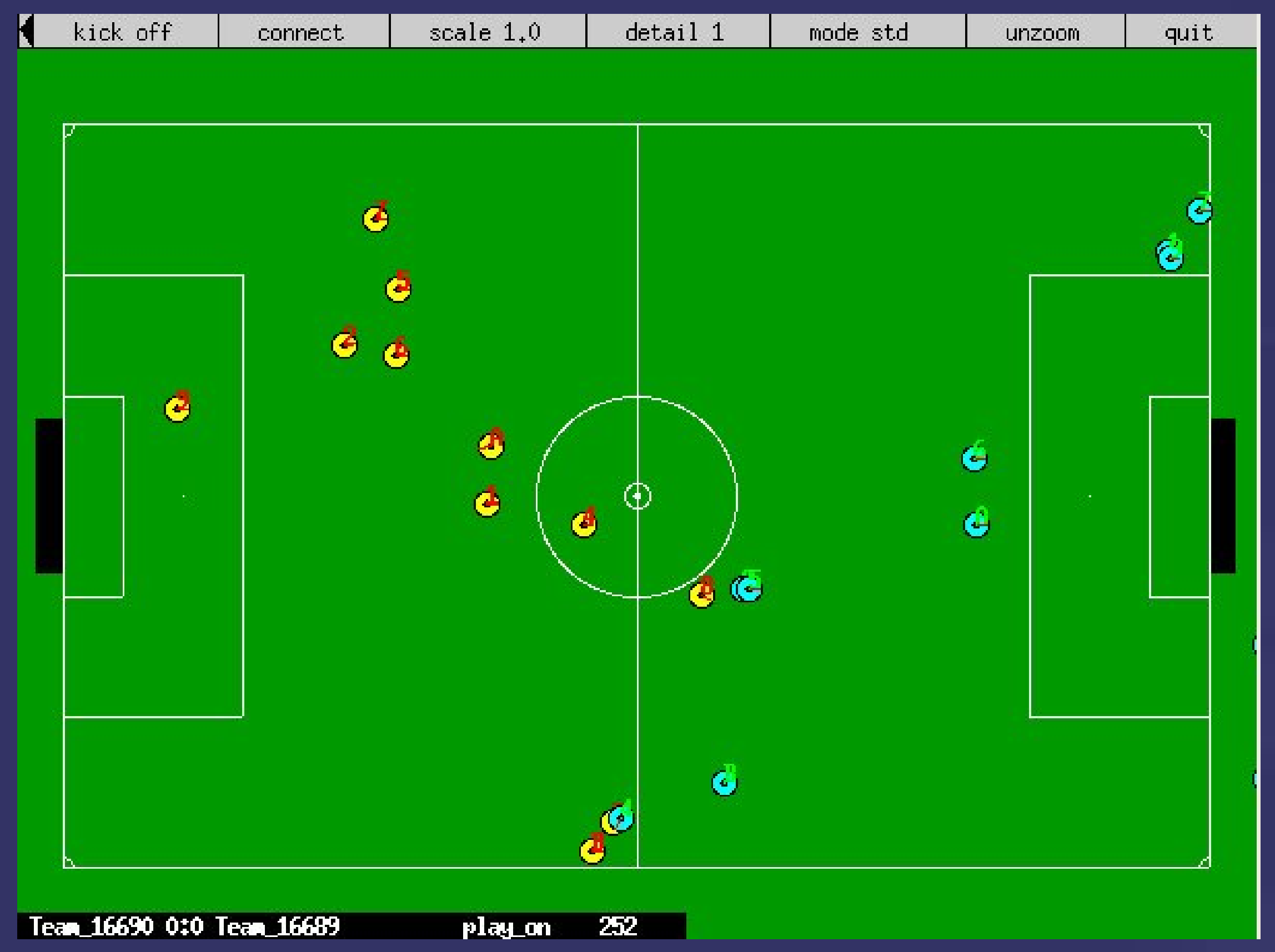
Robotic soccer is an active field of research within Artificial Intelligence. It is a test bed for developing multi-agent systems, with agents requiring to learn behaviours which allow them to work as a team and simultaneously defeat the opposition.

The RoboCup initiative has a goal of creating a team of autonomous humanoid robots that can beat the winners of the the FIFA World Cup by the middle of the 21st Century.

RoboCup Soccer Simulator

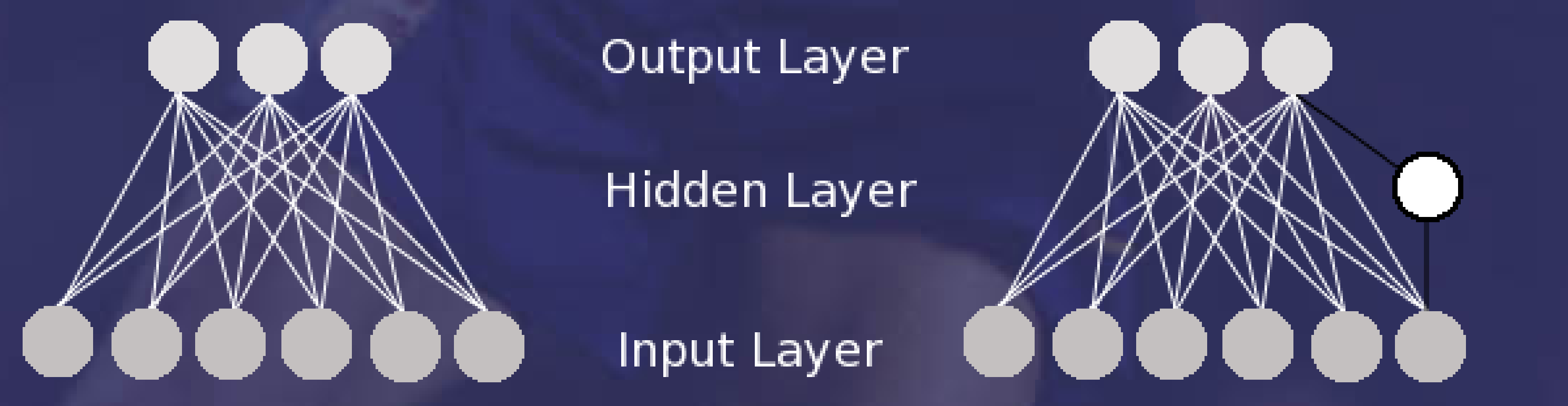
Due to the time taken to implement learning with physical robots a lot of work is done, at least initially, using simulators. The RoboCup competition has a simulation category in which games are played using their simulation server.

The RoboCup Soccer Server is written in C++ and uses a client / server architecture. The server simulates the field and the movements of the ball and players based on instructions received from the clients, which control the players. The server and clients communicate using UPD/IP sockets. Each client controls one player by sending commands such as dash, turn and kick to the server. The server responds with information about the world.



Evolutionary Complexification

In traditional evolutionary computing systems a genome of a fixed size is used to represent solutions for the problem trying to be solved. However for complex open ended problems such as robotic soccer it is not clear how large the genome should be to allow for appropriate representation of the problem. Open ended problems also pose another problem due to the fact that the idea of an optimal solution does not exist, as it is a fair assumption that a marginally better agent could always be found. A good solution in a fixed length genome system will use all of the genome to encode the solution. The only way to improve this solution is to alter it, which will result in changing the behaviours the agent has already learned.



Initial fully connected network

A new hidden node and connections are added. Allowing for new behaviours rather than adapted behaviours

Evolutionary complexification is an approach that attempts to solve this problem. It works by starting with small simple genomes which over the generations have more genes added to them. This allows more complex solutions to be generated over time. By initially keeping the search space relatively small, it should be easier to find optimal solutions. New dimensions can then be added to the search space as required to increase the complexity of potential solutions. It seems likely that this approach is more likely to discover complex solutions than an approach which initially starts with a massive multi dimensional search space.

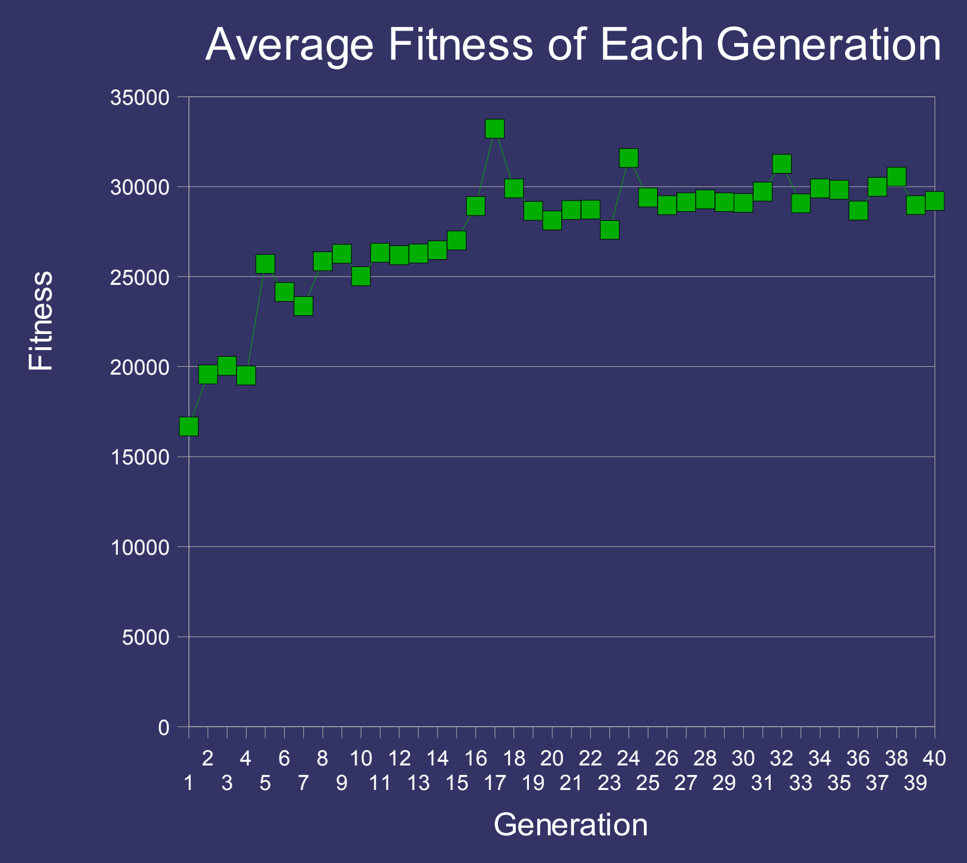
Implementation

An approach called NEAT (NeuroEvolution of Augmenting Topologies) was used for this system. NEAT applies the principles of evolutionary complexification to neural networks. Initially fully connected neural networks with no hidden nodes are used and over the generations nodes and connections are added. A Java implementation of Neat called ANJI (Another NEAT Java Implementation) was used.

The agents were coded in Java using the Dainamite framework, which includes code for communicating with the RoboCup Soccer Server and keeping track of objects in the world. The agents have three actions to choose from which are dashing, turning and kicking. The outputs from the neural network determine which action is chosen and the relevant values required for each action. A fitness function was devised which tried to reward good behaviour, such as facing the ball, moving towards the ball, kicking the ball and scoring goals.

Results

The system ran for 40 generations with a population of 20. As shown in the graph below the average fitness of the population improved suggesting that their may be potential in this approach. However due to the time taken to run the system, in depth testing has not been possible and the solutions so far do not play convincing robotic soccer.


Average Fitness of Each Generation
Fitness
Generation