Chapter IV

Case Study: Supervised Neural Networks in Digital Games

Introduction

In this short chapter we present a case study of the use of ANN in a video game type situation. The example is one of duelling robots, a problem which, as we will see, lends itself to a range of different solutions and where we can demonstrate the efficacy of a biologically inspired AI approach.

Robocode

For this case study we discuss the use of **Robocode** (Nelson, 2001) to explore the use of neural networks as an AI state selector within a game (Stewart, 2004). Robocode may be thought of a special case of a computer game in which the player is responsible for creating programs that operate their own derivation of Java 'robots'

to compete with other stock robots or against other player-created robots. The competitions occur in real-time in a 2D plain square arena with four walls containing the competing robots. Gameplay is relatively straightforward; the **robots** move autonomously around the arena shooting bullets attempting to hit other robots while also trying to avoid being shot. Competitions may be one-on-one or against multiple opponents, and team play is also supported, though we do not use this aspect in our case study. The autonomous nature of the robots, the ease with which the robots may be created, and the restriction of resources—particularly with the time restrictions for the AI of each robot—make Robocode a very interesting and suitable medium for testing a variety of AI mechanisms for real-time computer games.

The Robocode Web community has been very active and enthusiastic since the release of the product in 2001, but most of the focus in the development of new bots has been placed on bullet dodging algorithms and/or geometric or statistical approaches for predicting opponent movement. A few attempts have been made to utilise neural networks but most have been relatively unsuccessful against the best bot examples on the Robocode repository. This illustrates one of the difficulties of using neural networks in a real-time, constrained, and dynamic game environment. It is difficult to set up and train a neural network based AI bot that generalises well to new situations and that is successful in performing in a dynamic environment.

Let us start with a simple example that uses a more traditional form of game AI: the **state machine**. The finite state machine has been one of the most common and effective means for realising an AI within computer games, particularly for the implementation of a behavioural AI. A large part of the reason for its use being that the underlying principles are easy to understand, that it is a straightforward tech-

Figure 1. An example of a state machine that may be used to define the behaviour for a computer controlled opponent in a game