

Controllers for Ms. PacMan

Jakob Melnyk

I. APPROACHES

In this paper, I discuss three approaches for making a Ms PacMan controller: a Neural Network trained with Backpropagation, a Monte Carlo Tree Search and a Behaviour Tree evolved with an evolutionary algorithm.

A. Neural Network with Backpropagation

An artificial neural network (ANN) consists of layers of neurons connected through synapses. The two outer layers of the neural network are the input and the output layers with a number of "hidden" layers in between. The synapses connect each neuron on a layer to all neurons on the previous layer. Each synapse contains a weight, which defines how much the neuron in the previous layer influences the neuron the synapse is connected to. The weights of the synapses influence the output value of each neuron in the network; eventually the input will have been fed forward through the network and the neurons in the output layer will have a value that can be used for classification[1].

Backpropagation is an algorithm for training a neural network using training data to improve the classification of the neural network. It works by using the training data to feed forward a specific input and comparing the actual output of neural network to that of the expected output. This continues until a terminating condition is reached; either the neural network is able to classify an acceptable amount of training tuples correctly or a predetermined maximum number of iterations over the training data (epochs) have been done[2, 3, 1].

The Ms. PacMan controller was built using a neural network trained with the backpropagation algorithm. Every time the game requests a move, the controller feeds the current game state to the neural network. The neural network makes a decision regarding which direction Ms. PacMan should move based on a classification from the game state to the chosen move.

1) Algorithm Parameters: A number of parameters exist in both the neural network and the backpropagation algorithm. A neural network can have a variable amount of inputs, outputs, hidden layers and neurons in each hidden layer; the weight of each synapse and whether a layer has a bias neuron or not are also variables.

Experimenting with different topologies led to no significant differences in the performance of the PacMan controller; the main influencing factor is likely in the way the network is used.

The input values of the network were normalised between 0.0 and 1.0 as is typical[3, p.400]. The weights of the network were initialised to be a random number between -0.5 and 0.5; this could also have been between -1.0 and 1.0 [3, p.400].

Every neuron in the hidden layers and the output layer was also given a bias.

During experimentation, a number of combinations of different variables in the game state were used as input to the neural network. No particular combination of inputs changed the performance of the controller.

The backpropagation algorithm consist of five elements that can be changed: learning rate, samples to average over, how often learning rate is reduced, the terminating conditions and whether weights are updated in batch at the end of an epoch or during training of each tuple. The terminating conditions consist of the maximum amount of epochs, maximum change in a single weight and the percentage of misclassified tuples[3].

Over the course of developing the controller, only the terminating conditions and how often learning rate should be reduced were experimented with. While it made no difference to the controller's performance, testing with the "AND" and "XOR" truth tables showed that a higher number of epochs were preferable, but that the descent of the learning rate should follow the number of maximum epochs. Due to being unable to classify any tuple correctly during training of the neural network for the PacMan controller, the maximum percentage of misclassified tuples was arbitrarily set to 5%. For the same reason, the same was done for the maximum change in a single weight.

B. Monte Carlo Tree Search

1) Algorithm Parameters:

C. Evolutionary Algorithm for Behaviour Tree

1) Algorithm Parameters:

II. PERFORMANCE MEASURE

The controllers in this paper are evaluated by the scores that they achieve in 1000 trials against four different ghost controllers; "Starter Ghosts", "Random Ghosts", "Legacy2TheReckoning", and "Aggressive Ghosts". The controllers will be ranked by their average score over the 1000 trials; additionally, the paper will discuss of the standard deviation of each controller. The results of each of the three controllers will be compared to a Behaviour Tree controller submitted to the Initial Ms. PacMan competition of the Modern AI in Games course at IT-University of Copenhagen, Fall 2014[4].

REFERENCES

- [1] Sebastian Risi. Lecture 6: Neural networks and backpropagation. Course on Modern AI in Games at IT-University of Copenhagen, September 2014.

- [2] Peter Ross. Neural networks: an introduction. 1999.
- [3] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2011. ISBN 0123814790, 9780123814791.
- [4] Viktor Risi, Sebastian; Chudinov. "the hunger game". Course on Modern AI in Games at IT-University of Copenhagen, September 2014.