DEEP THREE MATCH: SOLVING MATCH THREE GAMES WITH DEEP LEARNING TECHNIQUES

by

THOMAS BRERETON & ELLIOTT DAVIES

A thesis submitted to
The University of Birmingham
for the degree of
MASTER OF SCIENCE

School of Computer Science
College of Engineering and Physical Sciences
The University of Birmingham
August 2017

CHAPTER 1

PROPOSAL

1.1 Introduction

In this thesis an artificial intelligence (A.I.) agent will be built which will play and solve the game of three match (games such as Bejeweled or Candy Crush Saga). The agent will be composed of a Monte Carlo tree search (MCTS) program, a neural network for the evaluation function, and a neural network for the policy. For this the authors Tom Brereton and Elliott Davies will be collaborating. Tom will be focusing on the evaluation function and aims to learn the features of the function using an auto-encoder, then train a neural network which uses these features that can evaluate the game state, Elliott will design the policy, and the authors will work together on designing the game, implementing the Monte Carlo tree search and integrating their work.

1.2 Motivation

The authors are interested in producing a similar program to that of AlphaGo, however, this will be applied to a match three game rather than Go. In match three games the main game play component is to match three pieces of the same type, in the authors game the aim is to use this to break 'ice' underneath and uncover all the 'medals' (refer Figure 1.1). Match three games comes with their own challenges, namely the highly random nature of the pieces being added to the board. If three cells on the board need to be filled and there are six gem types then the program must already consider 216 different potential states (6^3). This in combination with chain reactions of matches, leads to a tremendous branching factor and an enormous search space.

AlphaGo uses a value network which takes in a raw pixel representation of the board and is sufficiently trained to evaluate this pixel representation. The network uses feature planes where the features are hand-selected and represent details like the number of liberties (empty adjacent points) and ladder capture (whether a move at this point is a successful ladder capture). In this thesis Tom will investigate using a neural network which takes in a integer-valued vector representing the game state and evaluates it. This is essentially a raw representation of the game. He will also investigate using an auto-encoder to extract features from the integer-valued vector, as compared to using pre-determined 'feature planes' used in AlphaGo. The output of the auto-encoder will then be fed into another network to evaluate the state. This will result in a evaluation function that is learned end-to-end by the neural network. Learning the entire evaluation function and the stochastic nature of the game make for a challenging project and it will be interesting to see if the agent can make successful short and long term tactical decisions. Elliott plans to build two policies, a simple one that can be evaluated quickly can used frequently and another is more accurate but slower.

1.3 Plan

In this section the work the authors will undertake is outlined. Each table represents a task, which is broken down into sub-tasks in the description section. A Gantt chart is also included at the end for a graphical view of the project time line.

Figure 1.1: The authors match three game: Gem Island

Task	1. Complete game
Due	2017/06/13
Leader	TB
Collaborator	ED
Objectives	To build a challenging game for an AI agent to solve.
Description	1.1 Make a simple game for proof of concept 1.2 Implement matches for gems 1.3 Implement removable ice 1.4 Implement 'gravity' to pull down gems 1.5 Implement animations 1.6 Implement scoring system 1.7 Implement bonus gems 1.8 Implement combination scoring
Milestones	1.1 Making a simple game 1.6 A fully working game without bonuses
Deliverable	A complete game for the AI to solve

Task	2. Set up game for AI
Due	19/6/2017
Leader	TB
Collaborator	ED
Objectives	To get the game in a state for the AI to control.
Description	2.1 Design the game state representation 2.2 Implement methods to get game state 2.3 Implement methods for AI to call
Milestones	-
Deliverable	A game designed so that an AI can control it.

Task	3. Build naïve AI version 1
Due	22/6/2017
Leader	ED
Collaborator	TB
Objectives	Proof of concept for getting an AI to control the game.
Description	3.1 Implement random policy/move selection 3.2 Connect AI to game 3.3 Collate training data from version 1
Milestones	-
Deliverable	A working AI which can control the game.

Task	4. Build naïve AI version 2
Due	28/6/2017
Leader	TB
Collaborator	ED
Objectives	Proof of concept for using MCTS, evaluation function, and a policy
Description	4.1 Design search, policy, and evaluation function (s, p, e) 4.2 Implement s, p, e with 1-step look-ahead 4.3 Collate training data from version 2
Milestones	-
Deliverable	A naïve version of the final design of the AI.

Task	5. Gather and collate training data
Due	27/6/2017
Leader	TB
Collaborator	ED
Objectives	To obtain the required training data for the neural networks.
Description	5.1 Set up game to output state to file 5.2 Set up game to distribute to users 5.3 Distribute game to users
Milestones	-
Deliverable	Game to distribute - 22/06/17 Collated training data 27/07/17

Task	6. Build neural network 1 (NN1) with hand selected features.
Due	4/7/2017
Leader	TB
Collaborator	-
Objectives	To build a NN which evaluates the game state
Description	6.1 Design and build NN1 6.2 Train NN1 6.3 Connect NN1 to game
Milestones	-
Deliverable	AI agent with working NN

Task	7. Build neural network 2 (NN2) with learned evaluation function.
Due	14/7/2017
Leader	TB
Collaborator	-
Objectives	To build a NN which learns the function features and evaluates the game state.
Description	7.1 Design and build auto-encoder 7.2 Build NN2 with learned features 7.3 Train NN2 7.4 Connect NN1 to game
Milestones	-
Deliverable	AI agent with fully learned evaluation function.

Task	8. Build Monte Carlo tree search program
Due	21/7/2017
Leader	ED
Collaborator	TB
Objectives	To build a tree search program to solve the game.
Description	8.1 Design and implement MCTS program with multi-step look-ahead 8.2 Connect MCTS program to AI
Milestones	-
Deliverable	A working AI similar in design to AlphaGo

Task	9. Write dissertation
Due	13/8/2017
Leader	TB
Collaborator	-
Objectives	To write a dissertation.
Description	9.1 Outline of dissertation 26/06/17 9.2 Literature review 21/06/17 9.3 Definition of problem 03/07/17 9.4 Solution to problem 03/07/17 9.5 Why is it novel 03/07/17 9.6 Methodology for NN1 06/07/17 9.7 Half draft 10/07/17 9.8 Full methodology including NN2, search, policy 01/08/17 9.9 Discussion of results 05/08/17 9.10 Full draft 10/08/17 9.11 Final copy 30/08/17
Milestones	-
Deliverable	Half draft 10/07/17 Full draft 20/08/17

Task	10. Build initial policy heuristic using hand chosen features
Due	28/06/2017
Leader	ED
Collaborator	-
Objectives	Build a simple policy that assigns weights the features of different moves
Description	10.1 Select features to be used 10.2 Determine weights of features from training data 10.3 Connect policy to game
Milestones	
Deliverable	A basic policy to evaluate actions

Task	11. Build policy neural network from whole game state
Due	4/7/2017
Leader	ED
Collaborator	-
Objectives	Build a more complex policy to choose actions
Description	11.1 Build neural network 11.2 Train neural network with training data 11.3 Connect policy network to game
Milestones	
Deliverable	A more complex policy to evaluate actions

Task	12. Build policy neural network from auto-encoder features
Due	14/07/2017
Leader	ED
Collaborator	-
Objectives	Build a more intelligent policy which will be given the features of interest
Description	12.1 Build neural network 12.2 Train neural network with training data 12.3 Connect policy network to game 12.4 Compare results with those of previous NN
Milestones	
Deliverable	A faster complex policy to evaluate actions

Task	13. Write dissertation
Due	31/08/2017
Leader	ED
Collaborator	-
Objectives	Write the dissertation for the whole project
Description	13.1 Complete literature review, 31/06/2017 13.1 Draft outline, 06/07/2017 13.3 Problem to be addressed, 10/07/2017 13.4 Work to be undertaken, 14/07/2017 13.5 Methodology for simple policy, 25/07/2017 13.6 Methodology for policy network one, 01/08/2017 13.7 Methodology for policy network two, 08/08/2017 13.8 Technical Implementation, 15/08/2017 13.9 Discussion of results, 22/08/2017 13.10 Conclusion, 24/08/2017 13.11 Completing Dissertation, 31/08/2017
Milestones	-
Deliverable	Complete dissertation