THE PAST, PRESENT AND FUTURE OF ARTIFICIAL NEURAL NETWORKS IN DIGITAL GAMES

Darryl Charles[1] and **Stephen McGlinchey**[2]
[1]School of Computing & Information Engineering, University of Ulster, N. Ireland.
dk.charles@ulster.ac.uk
[2]School of Computing, University of Paisley, High Street, Paisley, Scotland.
stephen.mcglinchey@paisley.ac.uk

KEYWORDS

Neural Networks, learning, adaptive.

ABSTRACT

Over the past 20 years, the topic of artificial neural networks has been a vibrant area of AI research, leading to new algorithms that have been used in a variety of disciplines including engineering, finance, artificial perception and control & simulation. Despite this, there has been a limited impact on the commercial games industry. This paper reviews some of the successful uses of neural networks in games and identifies the positive elements of their use, and discusses some of the factors that have deterred their use amongst game developers. Addressing these weaknesses, we outline ideas for future research that may aid game developers in producing more convincing AI, and may supplement or replace more traditional techniques.

INTRODUCTION

There have been attempts to use artificial neural networks in digital games for quite a number of years now and the reason for this is quite straightforward; artificial neural networks are about learning, and the effective use of learning technology in games has been something that many in the game design development industry have desired for a number of years now. It also helps that neural networks are relatively well known and understood – particularly by computer science graduates – and their use is also popular because they (loosely) model biological neural networks such as those in our own brains, and so the link to learning and human-level intelligence is therefore very tangible.

Learning mechanisms in digital games may be offline or online. With offline learning we train the AI during the development process only. Once the product is released, the AI is unable to continue learning as a game is played. For example, the AI could observe and model player behaviour using learning algorithms such as artificial neural networks (McGlinchey, 2003). This may be used to create believable characters by imitation of a typical (or perhaps expert) player or a combination of features from a variety of players, or perhaps to model players or groups of players in order to respond appropriately to a player in-game. Online learning means that the AI learns (or continues to learn) whilst the end product is being used, and the AI in games is able to adapt to the style of play of the user. Online learning is a much more difficult prospect because it is a real-time process and many of the commonly used algorithms for learning are therefore not suitable. Instead these algorithms must be adapted for real-time dynamic processes 0. Real-time strategy (RTS) games are a particular candidate for online learning algorithms and some interesting approaches are being developed (Fyfe, 2004). In some situations a combination of both offline learning and online adaptation is the most appropriate approach (Livingstone & McDowell, 2003). These aspects of the implementation of learning technologies into games are inherent to the use of neural networks in games and we will revisit them often throughout the paper.

NEURAL NETWORK LEARNING

Most attendees of this conference will be familiar with the different categories of learning for neural networks: supervised, unsupervised and reinforcement learning. So we only provide a brief overview. With supervised learning, we provide the network with input data and the correct answer i.e. what output we wish to receive given that input data. The input data is typically propagated forward through the network until activation reaches the output neurons. We can then compare the answer, which the network has calculated with that which we wished to get. If the answers agree, we need make no change to the network; if, however, the answer which the network is giving is different from that which we wished then we adjust the weights to ensure that the network is more likely to give the correct answer in future if it is again presented with the same (or similar) input data. This weight adjustment scheme is known as supervised learning or learning with a teacher.

With unsupervised learning there is no external teacher and learning is generally based only on information that is local to each neuron. This is also often referred to as self-organisation, in the sense that the network self-organises in response to data presented to the network and detects the emergent collective properties within the data. Unsupervised neural methods are often used in an exploratory manner; we use statistical relationships between data variables in order to establish an understanding of the nature of the data. Unlike supervised learning, we do not know the answers before we begin training.

A third less commonly used from of neural learning is reinforcement learning. This learning relates to maximizing a numerical reward signal through a sort of trial-and-error search. In order to learn the network is not told which actions to take but instead must discover which actions yield the most reward by trying them – if an action has been successful then the weights are altered to reinforce that behaviour otherwise that action is discouraged in the modification of the weights. Reinforcement learning is different from supervised learning in that with supervised methods, learning is from examples provided by some knowledgeable external supervisor. With interactive sorts of problems it is quite often unrealistic to expect to be able to provide examples of desired behaviour that are both correct and representative for all scenarios which an agent may encounter. Yet this is perhaps where we would expect learning to be most beneficial, particularly with agent technology where an agent can learn from experience.

For those just starting to work with neural networks in digital games there are now many introductions to the topic available (for example (Champandard, 2002, Sweetser, 2004) and gameai.com is also well worth a visit. To delve more deeply you may wish to refer to one of the books in the area such as the introductory text by Gurney (Gurney, 1996) or the more advanced book by Haykin (Haykin, 1998).

CURRENT APPROACHES

It may be observed even from a brief literature review that the use of neural networks is still quite rare in mainstream commercial games and that the range of neural networks used is very limited – the error back-propagation algorithm is the most widely used neural network because is the most well known. However, the use of neural networks in digital versions of classic games such as Mastermind, Othello, Checkers (Draughts), and Backgammon is not unusual and has been successful in many situations as with Big Blue (see gameai.com). However, the use of neural networks in this type of game mostly focuses on strategy and the games are often more slowly paced. Modern digital games generally have more dynamic environments and the CPU has to deal with much more than just the AI. Current commercial digital games are varied and strategy is only one aspect of these games that we may apply neural networks to. Having said that, there are surprisingly few examples of the use of neural networks in commercial games, a couple of the best examples including “Colin McRae Rally 2” which uses neural networks to train the non-player vehicles to drive realistically on the track, and “Creatures” which uses neural networks along with evolutionary algorithms to dynamically evolve unique behaviours for game creatures. Black & White is the most high profile example of a recent game that utilises in-game learning – neurons are incorporated into an AI module for the game avatar, and these neurons are iteratively re-trained based on game feedback. The game uses a form of Perceptron learning within modules, for example, to model an avatar’s desire (Evans, 2002). The output of the neuron providing a measure of desire based on inputs which represent levels of “desire sources” for avatar attributes, such as: hunger, tastiness (of food), and unhappiness. The agent architecture is loosely modelled in the first place from psychological/philosophical ideas.

Social simulation games such as The Sims (Electronic Arts, 2001) naturally lend themselves to dynamic learning; these games are based on interaction between characters and objects due to environmental and social input. A character makes decisions within the game based on their current state and the state of the environment, for example if a character is hungry and they are close to a fridge containing food then they will prepare some food and eat it. A character may change their preferences or reactions over the period of the game based on “experience”. Recent academic research has demonstrated the use of neural networks (MacNamee & Cunningham, 2003) to create intelligent social controllers for agents that represent non-player characters. Other interesting recent examples of the use of neural networks within games include an approach for strategic decision making (Sweetser, 2004a), use of a self-organising map for modelling player behaviour (McGlinchey, 2003), and modelling player behaviour in first person shooters (FPS) using a method involving a multi-layer perceptron network (Geisler, 2004).

POTENTIAL FUTURE APPLICATIONS FOR NEURAL NETWORKS IN DIGITAL GAMES

There are a wide range of neural networks that have not even been attempted to be used in games applications, particularly unsupervised and reinforcement learning methods. Here we discuss a few potential techniques within the context of a number of key application areas for neural networks in games that have either not been addressed yet or have only been tackled recently.

Online Learning

Learning technologies for digital games have become increasingly important (Rabin, 2002). Yet, while there a number of examples of games that use “off-line” learning – for example, Quake III Bots may be trained using artificial neural networks or genetic algorithms – there are only a few examples of games that explicitly use “on-line” dynamic learning within a game, e.g. Black & White as discussed earlier.

The most significant issue with on-line learning is that it may produce unpredictable results; sometimes these effects serve to enhance but more often it leads to erratic game behaviour that reduces the quality of gameplay, and in worse scenarios will introduce dynamic game bugs. Testing, debugging and balancing games that incorporate learning is a challenging task (Barnes Hutchens, 2002). With the use of neural networks we have the added problem that, although they are very good for learning purposes, most neural algorithms can not easily be adapted incrementally but would generally require complete retraining online. Retraining is often very slow, and in many cases a small quantity of new data examples will not be enough to significantly impact the training of the algorithm, and of course, retraining the network completely online may lead to unsatisfactory results. An element of control is lost because tuning of the neural network by the developer will not be possible, as it is with the offline training of the network. These factors are presented not to discourage the reader from using neural networks for online learning but to encourage the development of new techniques and the use of suitable existing methods to approach this problem. For example, it is likely that dynamic online neural networks may need to be constrained to operate within predefined boundaries – i.e. the outputs of the networks are restricted to pre-tested values.

There are significant obstacles in the way of developing generic, robust and effective dynamic learning algorithms and architectures for digital games but the potential rewards are great (Charles, 2003). Perhaps the greatest potential gain with on-line learning is with the dynamic adaptation to player behaviour, play patterns and skill levels. In particular, a worthy pursuit is to develop technologies that may learn where a human player is being challenged too much or too little and modify the player’s character attributes, AI opponent behaviour or game environment accordingly. These alterations may be temporary, just to finish a particularly challenging section or the changes may be implemented for a longer time and player’s progress monitored. The flexibility afforded by dynamic learning mechanisms may also be used to counter a player benefiting unduly from – or being hindered by – unforeseen player behaviour or minor bugs in the game design. The capability of a game to self-adapt in these situations to prevent a significant deterioration in gameplay due to minor design oversights and player behaviour is certainly a laudable goal.

Player Centred Approaches: Player Modelling and Learning about the Player

It is perhaps not an obvious or much discussed issue relating to digital game AI but an important one nonetheless – that of attaining a more wide-spread appeal to entertainment of playing digital games. We need to keep the state of the games industry in perspective, the games industry continues to grow rapidly but it still represents only a small proportion of the entire entertainment and media industry. Even though there are a wide range of age groups playing games now, thanks in part to the release and marketing of the PlayStation and the more mature content of PC games, there is still a wide range of people who never even try to play a game, or simply give up after a short attempt.

All game players are different; each has a different preference for the pace and style of gameplay within a game, and the range of game playing capabilities between players can vary widely. Even players with a similar level of game playing ability will often find separate aspects of a game to be more difficult to them individually and the techniques that each player focuses on to complete separate challenges can also be very different. For these reasons and others it can be very difficult to design a game that caters for a wide range of player capability and preference. Game developers have traditionally dealt with the range of player abilities in a very straightforward manner, for example, by allowing the player to select a difficulty level at the beginning of the game, as with the classic first person shooter “Doom”. Once a player selects their level of difficulty for a game designed in this way, then there is usually no attempt within the game to monitor how a player is performing in order to adjust the level of challenge or gameplay experience. While the concept of an adaptive game is a controversial topic among some gamers and developers, there are clear benefits to tailoring the game experience to particular player types – especially for educational games (Beal et al, 2002). Catering for the individual more effectively could help attract a wider participation, if for no other reason than making it easier for players to get started, progress and complete a game, and therefore widening the accessibility of games.

The use of neural networks for the player modelling process is quite an obvious approach but the authors are not aware of them having been used for this purpose in games yet and so we provide an overview to a few possible supervised and unsupervised approaches below. Neural networks are good at detecting patterns and clustering data (depending on the method) and so we can use a variety of neural network techniques in different ways to identify or understand different players. For example, we can use player reaction times, choices made, styles of play, accuracy of shots/hits, how often a stage needs to be repeated before completing, average health, number of deaths per level, kills per level per possible kills to build models of player's offline through the training of a neural network such as the multilayer perceptron network trained with the error back-propagation algorithm. These variables may be observed online used directly to decide how to change the parameters of the game environment, attributes of the player character or non-player character behaviour dynamically.

Another neural network approach to player modelling is to use a clustering algorithm. In this way we use the neural networks to cluster player types according to out-of-game and in-game data, grouping player with a similar profile into the same group type. There is a wide range of ways in which this may be done, for example we could use a radial basis network with fixed cluster centres to classify the players, with the centres fixed on different areas of the data space that we believe to provide a good "centre" for our player classification. By monitoring and adapting the player profile throughout the game then the player may achieve a new classification, and thus the game would respond differently. Radial basis networks may also have moving cluster centres and so the centres can be moved automatically during training to fit the data more appropriately.

We may use unsupervised neural networks in particular so as to form a statistical understanding of player data, to explore or investigate structure or patterns in data on the basis of statistics or information theory (or similar). Using projection methods such as Principal Components Analysis and Factor Analysis we typically want to explore the relationship between the input variables which is a different approach to clustering methods. Factor Analysis can identify relationships between sub-sets of the data variables that may be used to identify more refined aspects of player behaviour, e.g. output one could identify the overall capability of the player and output two may identify whether the player is cautious or just dashes in etc. Being able to identify more subtle or complex aspects of player behaviour could be very valuable in tailoring the game experience to the player, and it also potentially opens up new possibilities for dynamic gameplay. For example, if we are able to discover patterns that relate more to player emotion or motivation then this may be used with other sensory devices to discern the needs or desires of the player and the game can be adapted to account for this.

Intelligent Character Animation

The Artificial Intelligence in a game is perhaps one of the most influential ingredients for enabling a game player to suspend disbelief long enough to become properly immersed into the gameplay. If characters or objects behave in an obviously unexpected – or unintelligent – way, then the game experience is very much diminished. The quality of graphics in digital games has reached an incredible degree of realism, as witnessed by games like Doom III (ID Software, 2004), and realism of visuals is important, of course, because many of us enjoy the "wow" factor afforded by the visual impact of the newest and most graphically advanced game – this facet clearly sells games. Visual realism is only a part of what makes a game world and the characters in it believable, if any aspect of the game shatters our immersive gameplay experience and we are less able to suspend disbelief within the game world. In other words we may have a beautifully created wall using the latest vertex and pixel shader programs to enhance the illusion of the game world existence, but the illusion is shattered when our supposedly intelligent character continually bangs his head off the wall in an attempt to get round it or walks in mid-air!

Intelligent character animation is one approach to improve this aspect of player immersion. For example, a Neural Network may be used as the "decision maker" for an animating character and when paired to a fuzzy controller system this particular agent architecture can be useful (Wen et al, 2002). Neural networks may also have broader uses in character animation; for example, it should be possible to train a neural network to act as a transformation matrix in order to interpolate in the mesh blending technique described above. This will provide benefits during game development, but also opens new possibilities for run-time generation of animation data, allowing game characters to be truly responsive to game events and user interactions.

Let us first consider savings that can be made during development. Motion captured data can be time consuming and expensive to post-process due to several factors. Firstly, data from optical motion capture systems is normally incomplete due to optical occlusions, and this data must be completed by artists. Magnetic motion capture systems also have problems since the sensors produce noisy data. Mechanical systems have neither of these problems, however, actors and artists tend to dislike using these systems since they constrain the actor, and they can only be applied to simple bone structures. Neural networks have been used for dealing with noisy and incomplete data in other disciplines, and if new research in intelligent character animation can tackle these problems, the cost of using motion capture systems would be significantly reduced. Neural networks have also been trained on motion data and later used to synthesise key-framed motion data and this is the basis of at least one commercial tool for automatic generation of animation.

Current methods of animating characters can produce very impressive results; however, this comes at a significant cost, requiring skilled animators to work at a low level, specifying limb and joint positions and orientations, and restricting games to replaying fixed animation sequences. The idea of a "virtual actor" is to allow a director (or game developer) use a high-level set of instructions (e.g. creep, walk, run, read, say etc.) to direct the actions of avatars, and this may include adverbs describing style and emotional state (e.g., fearfully, excitedly). Convincing virtual actors will allow game developers to be less concerned with low-level details, and focus their efforts on drama and emotion, which can add significantly to the immersive qualities of games. Real-time generation of animation for virtual actors is an area where neural networks may be useful.

Prediction

Neural networks have been successfully used for prediction in several application areas including finance, weather forecasting, power consumption, sales forecasting etc. 0. It is not uncommon for a variety of different types of neural networks to be used for prediction generally in the computing and engineering world but they have not been used much (or at all) for this purpose in digital games. Prediction can be useful in quite a few ways in games, especially in strategic aspects of the games. For example in a real-time strategy game it is interesting to explore predictive approaches for a computer opponent in building its strategy, and in a 1st person action/adventure game non-player character prediction of player movement or strategy would be interesting in countering their movement. An added bonus of the use of a predictive approach is that behaviour can be non-deterministic and thus potentially more believable and can provide a more varied and interesting computer opponent.

Human-level Intelligence Studies

In one of the well known early papers (Laird, 2001) of the recent surge of interest into digital games research it was suggested that digital games provide an excellent platform to explore human level intelligence – which is after all one of the key original reasons that researchers began to work on Artificial Intelligence. Part of the reasoning behind this argument being that the virtual worlds and characters of commercial games are so rich in detail and that they provide an opportunity for a player to become immersed in realistic environments and interact with believable characters. An individual computer game or videogame may be played by millions of people and so this offers significant opportunity for study, and many games also offer high quality and easy to use game content development kits, e.g. "Neverwinter Nights", that provide an opportunity for the creation of suitable tailored experimental test-beds.

However, we must be careful in acknowledging the difference between human level intelligence imitation and the form of AI common in games which has more of a relationship to the Turing Test (Turing, 1950) and creating believable behaviour (or fooling the player). The inspiration for developing AI opponents for games may be traced back the to the Turing Test, since the original Turing Test may be thought of as a kind of game in which a computer must be programmed to fool an interrogator into believing that it is real woman as often as a man can fool the interrogator that he is a woman. The original question posed by Turing Test has evolved over the years to, "can a machine play a game of skill as well as a human being?" or "can the program compete with people?" (Fogel, 2002). In a way this is a distortion of the original goal of early AI research in that a central objective has been to understand human-level intelligence and replicate this functionality holistically. Writing a program that competes with a human opponent in a computer game is often as much to do with having enough raw computing power to process a large set of rules and creating an illusion of intelligence than it is about developing convincing human-level intelligence models.

Neural networks and models of the brain that include neural networks can prove very useful when exploring more human aspects of AI in games – because of their learning capabilities and resemblance of brain function. An interesting potential future technology related to human-level intelligence research involves the development of character AI architectures which allow us to "grow" or evolve game characters off-line – outside the game – and then insert this character into the game so that it will continue to learn. Could such a character be retrained and used in future games – a bit like a game actor? Would a player be able to extract an intelligent character from one game for use, with retraining, in a future game release? – sort of like an extended, intelligent, version of the character game save. Some work has been performed in this area already, where one well-known AI researcher believes that he can grow a conscious character on his computer (Cohen, 2002) and that characters such as these may be sold to game development companies.

With more lifelike characters and realistic emotional representation in our games we may have to consider the moral and ethical implications of decisions made by gamers even more than we do now and deliberately design-in effective consequences for actions. These issues become more significant as game characters approach some form of realistic consciousness, however, utilising AI to construct well-designed moral dilemmas and emotionally effective set pieces with games opens a range of new and interesting gameplay scenarios.

DISCUSSION

Most games allow only limited processing resources for AI, and this can often prohibit many advanced AI techniques. It is reported that 50% of the processing resources were allocated to AI in the game "Creatures" (see http://www.gameai.com/cgdc97notes.html). However, few commercial games have this rich allocation of processing resources to AI; 1% - 5% is a more typical allocation. With such limited resources available, it is often perceived that neural networks are too computationally expensive to be used in the majority of commercial games – particularly when the AI is to be trained online. This is a fair criticism of some of the well-known neural network methods such as the error back-propagation algorithm. However, there are many other neural algorithms that have a comparatively low computational cost, such as some Hebbian learning methods, topology-preserving maps, radial basis networks and Learning Vector Quantisation (LVQ). Moreover, some older neural network training methods can be implemented using optimisations that vastly reduce their computational cost. (Kohonen, 1996) For offline applications, the argument of prohibitive computational cost is rarely valid, since it is the weight update (training) procedures that normally require the majority of the computation – not the feed-forward, or sensory phase, which would be used at runtime.

Inexperienced users of neural networks often encounter problems with parameter selection. Let us consider the radial basis function network as a typical example. For this network, the user must choose a suitable number of centres, and each of these must be initialised to a point in the data space. There are several other parameters that need to be "tweaked" to ensure that the network converges to a stable state without over-fitting the data, including the centre variances, the initial weight values, the learning rate, and any learning rate annealing strategy. The set of parameters that works best will vary between different data sets and applications, and it can be time-consuming for a developer to find an acceptable set. The problem is further exemplified in online training, where there is no expert to hand-pick parameters. The problem of parameter selection has been tackled in recent years by several probabilistic methods, which have created a great deal of interest amongst the computational intelligence community. There are now many probabilistic neural algorithms (e.g. (Bishop, 1998, Hinton et al, 1995, and Yin & Allinson, 2002) that work using objective functions to train the network. These methods tend to have fewer user-selected parameters, and where parameters must be chosen they tend to be less sensitive to picking critical values. To our knowledge, this exciting area of neural network research has yet to be applied to games-specific applications, and this promises to be a worthwhile area for future research.

As implied in this discussion section, progress has been limited in the use of neural networks within digital games largely due to a lack of knowledge or understanding among researchers and game developers of the wide range of methods that may be applied to game AI. This situation can be improved by those of us with a wider and more detailed knowledge of neural methods providing a range of successful, persuasive and meaningful neural network enhanced game AI examples.

REFERENCES

Barnes J and Hutchens J. 2002. "Testing Undefined Behaviour as a Result of Learning", AI Game Programming Wisdom, pp 615-623, Charles River Media.

Beal C, Beck J, Westbrook D, Atkin M & Cohen P, "Intelligent Modeling of the User in Interactive Entertainment", pp 8 – 12, AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment, Standford, 2002.

Bishop, C.K., Svensen, M., Williams, C.K.I., "GTM: The Generative Topographic Mapping" neural Computation 10(1) pp215-234, 1998.

Buckland M, "Advanced AI Techniques for Game Programming", Chapter 10, Real Time Evolution, Premier Press, 2002.

Cander, L. R., "Ionospheric forecasting technique by artificial neural networks" Electronic Letters 34(16) pp1573-1574, 1998.

Champandard A, "The Dark Art of Neural Networks", AI Game Programming Wisdom, pp 640-651, Charles River Media, 2002.

Charles D, "Enhancing Gameplay: Challenges for Artificial Intelligence in Digital Games", Proceedings of the 1st World Conference on Digital Games 2003, University of Utrecht, The Netherlands, 4-6th November 2003.

Cohen D, "Game Over", New Scientist, pp 47-49, 29th June 2002.

Evans R, "Varieties of Learning", pp 567-578, AI Game Programming Wisdom, Charles River Media, 2002.

Fogel, D, "BLONDIE24: Playing at the edge of AI", Chapter 1, Morgan Kaufman, 2002.

Fyfe C, "Dynamic Strategy Creation and Selection Using Artificial Immune Systems", International Journal of Intelligent Games & Simulation, Vol. 3 No. 1, pp 1 – 6, 2004.

Geisler B, "Integrated Machine Learning for Behavior Modeling in Video Games", Challenges in Game Artificial Intelligence, AIII Press, 2004

Gurney K, "Introduction to Neural Networks" Routledge, 1997.

Haykin S, Neural Networks: A Comprehensive Foundation, Prentice Hall, 1998.

Hinton, G.E., Dayan, P., Frey, B.J., Neal, R.M. "The "wake-sleep" algorithm for unsupervised neural networks" Science 268(5214) pp 1158-61, May 1995.

Kohonen, “T. The speedy SOM.” Technical Report A33, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland, 1996

Laird J.E. & Van Lent, M. Interactive Computer Games: Human-Level AI's Killer Application, AI Magazine, 22 (2), 15-25, 2001.

Livingstone D. & McDowell R, “Fast-Marching and Fast Driving: Combining Off-Line Search and Reactive A.I.”, International Conference on Intelligent Games and Simulation (Game-On), London, 19th-21st November, 2003.

MacNamee B and Cunningham P, “Creating Socially Interactive Non Player Characters: The μ-SIC System”, International Journal of Intelligent Games and Simulation, Vol. 2 No. 1 Feb. 2003.

McGlinchey S, “Learning of AI Players from Game Observation Data”, GAME-ON 2003, 4th International Conference on Intelligent Games and Simulation, pp. 106-110, Nov. 2003.

Rabin S (Ed.), AI Game Programming Wisdom, preface, Charles River Media, 2002.

Sweetser P, “How to Build Neural Networks for Games”, AI Game Programming Wisdom II, pp 615-625, Charles River Media, 2004.

Sweetser P, “Strategic Decision-Making with Neural Networks and Influence Maps”, pp 439-446, AI Game Programming Wisdom II, Charles River Media, 2004.

Turing A M, "Computing Machinery and Intelligence", Mind 59 (1950): 433-60.

Wen Z, Mehdi Q & Gough N, “Neural Networks for Animating Variations in Character Behaviours” ”GAME-ON 2002, 3rd International Conference on Intelligent Games and Simulation, pp. 189-196.

Yin, H. and Allinson, N.M., “Bayesian learning for self-organising maps,” *Electronics Letters*, Vol. 33, No. 4, pp. 304-305, 1997.