Evolving Artificial Neural Networks Applied to Generate Virtual Characters

Joan Marc Llargues Asensio
Universidad Autnoma de Barcelona
Edifici O Campus UAB
08193 Bellaterra (Barcelona), Spain
Email: fireball787b@gmail.com

Juan Peralta Donate
Universidad Autnoma de Barcelona
Edifici O, Campus UAB
08193 Bellaterra (Barcelona), Spain
Email: jperalta@cvc.uab.es

Paulo Cortez
ALGORITMI research centre
University of Minho, Guimaraes, Portugal
Email: pcortez@dsi.uminho.pt

***Abstract*—Computer game industry is one of the most profitable nowadays. Although this industry has evolved fast in the last years in different fields, Artificial Intelligence (AI) seems to be stuck. Many games still make use of simple state machines to simulate AI. New models can be designed and proposed to replace this jurassic technique. In this paper we propose the use of Artificial Neural Networks (ANN) as a new model. ANN will be then in charge of receiving information from the game (sensors) and carry out actions (actuators) according to the information received. The search for the best ANN is a complex task that strongly affects the task performance while often requiring a high computational time. In this work, we present ADANN, a system for the automatic evolution and adaptation of artificial neural networks based on evolutionary ANN (EANN). This approach use Genetic Algorithm (GA) that evolves fully connected Artificial Neural Network. The testing game is called Unreal Tournament 2004. The new agent obtained has been put to the test jointly with CCBot3, the winner of BotPrize 2010 competition [1], and have showed a significant improvement in the humanness ratio. Additionally, we have confronted our approach and CCBot3 (winner of BotPrize competition in 2010) to First-person believability assessment (BotPrize original judging protocol), demonstrating that the active involvement of the judge has a great impact in the recognition of human-like behaviour.**

I. Introduction

Developing video game characters that can behave like a human is a current challenge. In fact, the ultimate goal is to to create characters who can provide an engaging and entertaining experience, with believable behavior and try to make them indistinguishable from human players. In other words, agents able to pass the Turing test [2] (or more specifically, an adapted version of the Turing test designed for assessing the believability of video game characters [3]).

Classical AI Non Player Characters (NPC), like *Mario* turtles or *Space Invaders* alien spacecrafts, were relatively simple to program, and little or none AI techniques had to be used. However, video games have evolved into more complex virtual environments, new games like *Max Payne* or *Call of Duty* require much more realistic behaviors for their characters. State machine programmed behaviors are valid in some old and even new scenarios, but in realistic games consumers expect to find artificial opponents behaving like humans. When compared with other human players the behavior of artificial characters is usually considered disappointing. Current AI characters can be intelligent in some sense; however, they cannot match the behavior produced by a human player (see for instance the results of the last 2K BotPrize contest [4] a competition based on the Turing test). Currently, playing with other humans is generally more realistic and engaging than playing againts NPC.

To create new models that can behave like humans it is propossed, in this paper, the use of ANN [5]. Computational Intelligence models such as ANNs are natural solutions, since they are more flexible (i.e. no a priori knowledge is required) when compared with classical models, presenting nonlinear learning capabilities and robustness to noisy data. This paper is focused on the use of ANN [5], and more specifically multilayer perceptron. When using multilayer perceptrons, a key issue is the design of the best model. Often, such design is set manually, using trial and error heuristics. A better alternative is to use an automatic design procedure. In particular, by adopting Evolutionary Computation to search for the best ANN, in what is known as Evolutionary ANNs (EANNs). Several authors indeed have proposed EANNs to solve different general domain problems, such as [6], [7]. In particular, Jacob et al. [8], presented a ANN approach for AI in games. In this paper, we present a different approach and the test in an adapted version of the Turing test based in a video game [4]. Additionally, we assess the believability (or ”humanness”) of these NPCs (i.e. bots) using two different assessing methods: First-person and Third-person judges.

The remainder of this paper is structured as follows. In the next section we present the state of the art, followed in Section III by a description of the implementation that we have developed for the believability experiments. Finally, experimental results are presented in section IV and discussed in Section V.

II. State of the Art

In the last years many studies treating human-like bot behaviour have appeared [9], [10], [8] We discuss some of these projects and the differences with our approach.

Most projects that connect agents to UT2004 are built on top of Gamebots [11] or Pogamut [12]. Gamebots is a platform that acts as a UT2004 server and thus facilitates the transfer of information from UT2004 to the client (agent platform). The GameBots platform comes with a variety of predefined tasks and environments. It provides an architecture for connecting agents to bots in the UT2004 game while also allowing human players to connect to the UT2004 server to participate in a

978-1-4799-3547-5/14/1.00 ©2014 IEEE

game. Pogamut is a framework that extends GameBots in various ways, and provides a.o. an IDE for developing agents and a parser that maps Gamebots string output to Java objects. We have built on top of Pogamut because it provides additional functionality related to, for example, obtaining information about navigation points, ray tracing, and commands that allow controlling the UT2004 gaming environment, e.g. to replay recordings.

In [13], in order to design interesting opponents, the authors propose to teach a computer opponent to play like a human using machine learning techniques. Similarly, in [14], the authors introduced Dynamic Scripting, an online learning technique based on the use of reinforcement learning to adjust a mechanism for selection to choose between various scripted behaviours. The stochastic selection mechanism and the online learning provided the play with a degree of unpredictability so to make it more human-likeness. In addition, in [15], the authors studied a Bayesian-based approach to the derivation and imitation of human strategic behaviour and motion patterns in commercial computer games. They demonstrated the effectiveness of their approach in producing convincingly human-like game agents in conjunction with the believability-testing system. Finally, in [16], the authors propose a method for generating natural-looking behaviours for virtual characters using a data-driven method called behaviour capture.

Although we recognize the strengths of cognitive previous implementations like those based on cognitive architectures [9], [17] and others based on Long-Term Memory Database implementation [10], we focus in the use of ANN and in particular the use of EANN [8] working in a more low level that cognitive studies.

III. ADANN Model

A. Naturally occuring Neural networks

Neural networks are collections of interacting brain cells called neurons. Each neuron is a fantastically complex information processor. A Neuron can hold a certain electric charge before it 'fires' a signal to the other neurons it is connected to. This signal charges up the neurons it is connected to. Neurons connect themselves with other neurons with long growths off the nucleus of the cell called axons. These split up into many sub-branches which terminate in little nubs called synapses. These synapses can latch on to parts of other neurons, establishing an electrical/chemical link between the the neurons. In this way each neuron may be connected to up to around 10000 other neurons. Some Neurons can also communicate by other means such as emitting gas that can effect other neurons they are not connected to by synapses. So each neuron is influenced by many other neurons, meaning there is no central seat of power or center of influence in the brain. The complex thought processes of a brain result from the many interactions of these cells. The human brain has about 100 billion neurons with about 100 - 1000 trillion synapses(connections). The nematode worm Caenorhabditis elegans has only 302 neurons but can move, react to stimuli and even cary out simple pattern recognition to find food.(ta New Scientist).

B. Evolving Artificial Neural Networks

The goal of the artificial intelligence has been around for a long time. And what a better way to create the infrastructure of intelligence than to emulate a design that has been proven to work and be versitile by evolution. Artifical neural networks are (usually greatly simplified) models of naturally occuring neural networks that seek to utilise the power and capacity for learning of neural networks. An ANN implemented on a computer usually consists of a grid of stylised neurons and synapses (Fig. 1).

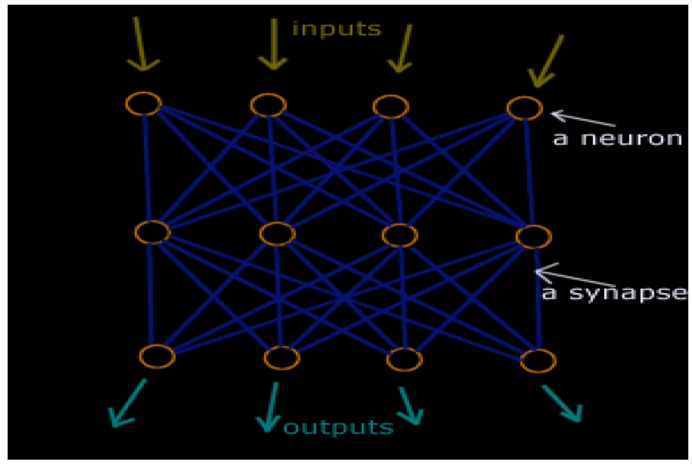

Fig. 1. ANN schema.

When working with ANN, finding an adequate ANN model is a key issue. Different studies have dealt with the design of an ANN from two different points of view.

- Topology: number of hidden layers, hidden nodes in each layer, etc.; and
- Connection weights: values for each connection in an ANN.

In this specific domain the topology is given by the problem. After a detailed study of the inputs (current health, armour, damage, weapons, etc.) and outputs (movement, rotate, dodge, jump, crouch, etc.) from the game that should be used, carried us to the final decision of using the same architecture for each ANN. This architecture is translated into chromosomes, encoding the information given by the ANN previously commented into chains of information as it can be seen at Fig. 2.

Related to the estimation of the connection weights, it is well known that learning algorithms like backpropagation usually got stuck in a local minimum [18] but also, due to we are working with real time computer games, it was not possible to obtain learning patterns from the game to train the ANN models. Whitley et al. [19] proposed the use of evolutionary computation to search for appropriate connection weights and avoiding the local minimum problem by means of a global search [20]. The process of obtaining a candidate model will be split into two steps. First, several random initialized individuals (ANN) of a first generation are obtained. We make them combat using the input information that comes from the game and let them apply actions (outputs), while we measure how

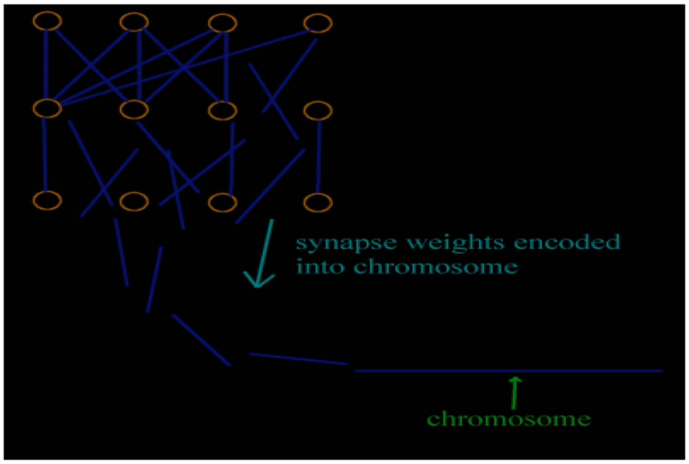

Fig. 2. ANN encoding.

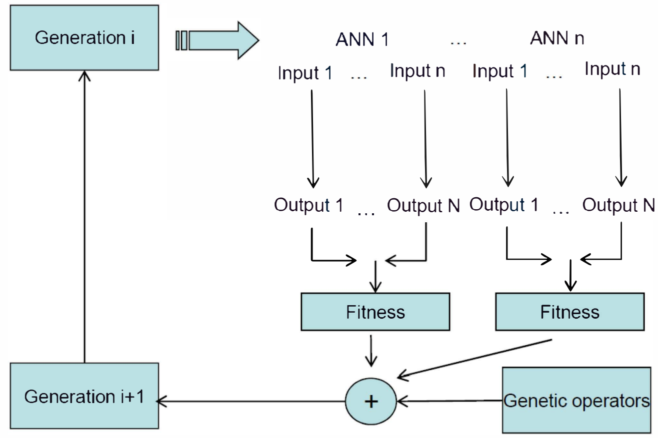

Fig. 4. ADANN schema.

good they are (fitness value). This fitness value is measured as the number of deaths caused by the bot divided by the number of times it has died.

Every six minutes we apply genetic operators (i.e., gaussian mutation) to obtain a new generation as we show in Fig. 3, is repeated until a maximum number of generations (i.e., 100) is reached.

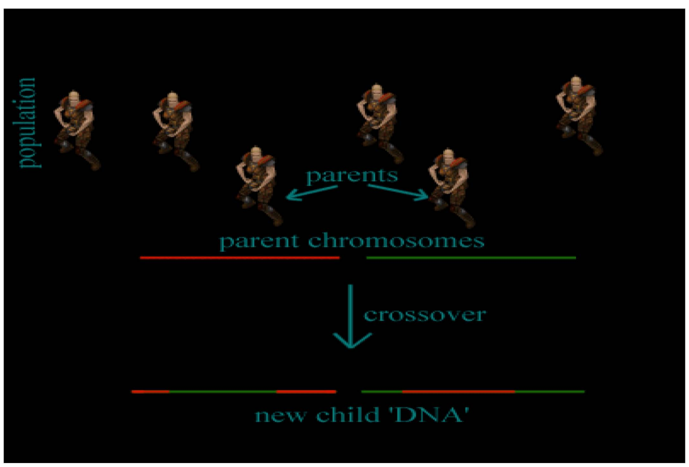

Fig. 3. Genetic Algorithm schema.

A whole graph of the complete process can be seen at Fig. 4.

At the end of the training process we will have a good NPC which will be introduced into the game (test) with other human and non human players as it will be explained below in detail.

IV. Experimentation

A. *Testing Protocol*

The first method that we have used in order to assess the believability of our bots is the international BotPrize competition testing environment [4]. The BotPrize challenge (held yearly since 2008) was originally conceived as a Turing Test for First-Person video game bots (NPCs). In the classical Turing Test accurate verbal report and conversational skills are the key factors, however in the domain of FPS bots these aspects are neglected, focusing the assessment completely in observed non-verbal sensory-motor skills.

BotPrize environment is based in the video game "Unreal Tournament 2004" by Epic Games, a First-person shooter set in a fictional future with futuristic weapons. The objective of the game (deathmatch mode) is to kill as many opponents as possible without being killed by the other players. Both artificial bots and human players connect to the game server by means of a local area network or over the Internet.

Different judging schemes were used in early editions of the BotPrize competition. In this work, we use the latest scheme adopted in 2010 [21]. In this scheme a judging gun (the "Link Gun") is included in the game. All players, humans and NPCs spawn with a Link Gun with infinite ammo. Although the primary and alternate fire modes of the judging gun look and sound the same to all observers, they have completely different meanings and effects: the primary firing mode is meant to issue a vote for a bot (artificial player or NPC) and the alternate firing mode is meant to issue a vote for a human player.

If a player shoots a bot using the primary firing mode (like if the objective was a bot), then the player gains 10 points as this was a good guess and the bot gets a vote as bot. However, if a player shoots a human player using the primary firing mode, then the player that fired loses 10 points (because he/she made a wrong guess) and the human gets a bot vote. Analogously, if a player shoots a bot using the alternate firing mode (like if the objective was a human), then the player loses 10 points (bad guess) and the bot gets a human vote. If the player shoots a human using the alternate firing mode, then the player gets 10 points (goog guess) anf the human gets a human bot.

During our tests we allow the players to judge any other players as many times as desired. Using the judging gun the game play is transformed from a pure deathmatch game into a hybrid game in which both judging and killing/surviving aspects have to be taking into account simultaneously. It is important to remark that all players (humans and artificial) have access to the judging gun. Therefore, the designer of a bot also has to take decisions on how and when the bot will use the Link Gun, as this usage will also be part of the observed behaviour.

Apart from the judging gun, the rest of the weapons function as usual. However, the damaged produced by these weapons is reduced by a 60%, thus giving human players more chances to observe the other players before being riddled under enemy fire.

In order to obtain a significant amount of judging data and reduce the bias that a given map would introduce, different testing sessions in different maps are organized using a centralized game server that runs the BotPrize mod. Each session lasts for 15 minutes and different maps and scenarios are used each time. Anonymity of players is guaranteed using random player names and random player skins (clothes and body appearance) that changes from one session to the next.

The number of human players and bots is balanced, having a similar number of human judges and artificial characters. All human players are meant to be judges, but they also compete for the highest score (that they obtain both from judging and from killing and not being killed).

The BotPrize testing protocol is a First-person observation approach, as judges are not allowed to use the spectator mode of the game, and they are also subject of attacks and votes (Link Gun shoots) from other players.

B. Results

In the following we summarize the results we have obtained confronting our bot controller to the First-person believability assessments againts CCBot3 [22]. As described above, we used the BotPrize competition environment and testing protocol as First-person observer method for assessing believability. We ran a total of 20 matches conducted during 5 sessions of 1 hour each. Matches last for 15 minutes and 4 different maps have been used per session, counterbalancing level maps across the sessions. Therefore, human players (judges) were asked to play (and judge) for 1 hour (4 maps of 15 minutes each) in 5 different days, with a period of one week between consecutive sessions. The whole testing procedure took place during 5 weeks and the selected human players were always the same. There were the same number of NPC and human players, three human and three NPC.

Human judges agreed voluntarily to participate in this study. They neither had previous experience in the design or programming of a NPC, nor any expertise in Artificial Intelligence. However, they were selected because they have intermediate experience with FPS video games (none of the judges was a novice or advanced player). Results shown at table I are in % of humanity.

TABLE I. FIRST-PERSON ASSESSMENT RESULTS.

Bots	S1	S2	S3	S4	S5	Average
CCBot3	**19.59**	19.84	16.81	20.14	28.70	21.02
ADANN	17.34	**37.04**	**31.73**	**30.26**	**47.23**	**32.72**

Comparing the results of the two different methods propossed, we can see that our EANN approach obtains better results when compared to CCBot3 [22] (winner of BotPrize competition in 2010).

While the bot solely based in the CERA-CRANIUM (CCBot3) architecture lacks any learning or long-term adaptation mechanism, the other bot with better results implements mechanism of adaptation. We believe the reason why ADANN outperforms CCBot3 in terms of believability lies in the learning/adaptation mechanisms. ADANN adapts well to the dynamics of the interaction with other players, however, in this case the mechanism used is not an explicit learning algorithm, but a genetic algorithm optimization.

V. CONCLUSION

One new approach was designed and compared using BotPrize measurements method. We can observe that EANN (i.e. ADANN) approach obtains better results in First-person experimentation. The believability assessment method applied in this work is indeed fully behavioural test, as it is inspired by the Turing Test. Although good results have been obtained with this new approach, there is a problem to be solved. The training and testing process of ADANN system are still separated. Due to this, although a good NPC is obtained during the training process, once this bot is taken from the training sandbox to the testing one, it will not learn more while it is playing. Then, a possible future work would be to make use of Non-supervised learning process and ANN so there would be only one sandbox, being it used for training and testing at the same time. On the other hand, everytime the bot is taken to a new game, it would be able to keep on learning from new players.

ACKNOWLEDGMENT

The research reported here has been supported by ”Desafio BotPrize project” [23]. The authors also want to thank Philip Hingston for providing the BotPrize competition testing environment.

REFERENCES

[1] R. Arrabales, J. Muñoz, A. Ledezma, and A. Sanchis, “A machine consciousness approach to the design of human-like bots,” in *Believable Bots*, P. Hingston, Ed. Springer Berlin Heidelberg, 2012, pp. 171–191.

[2] A. M. Turing, “Computers & thought,” E. A. Feigenbaum and J. Feldman, Eds. Cambridge, MA, USA: MIT Press, 1995, ch. Computing Machinery and Intelligence, pp. 11–35.

[3] D. Livingstone, “Turing’s test and believable ai in games,” *Comput. Entertain.*, vol. 4, no. 1, Jan. 2006.

[4] P. Hingston, “A turing test for computer game bots,” *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 1, no. 3, pp. 169–186, 2009.

[5] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998.

[6] J. Peralta, G. Gutierrez, and A. Sanchis, “Adann: automatic design of artificial neural networks.” in *GECCO (Companion)*, C. Ryan and M. Keijzer, Eds. ACM, 2008, pp. 1863–1870.

[7] P. Cortez, J. Machado, and J. Neves, “An evolutionary artificial neural network time series forecasting system,” in *In Proceedings of the*

IASTED International Conference on Artificial Intelligence, Expert Systems and Neural Networks, 1996, pp. 278–281.

[8] J. Schrum, I. V. Karpov, and R. Miikkulainen, "Ut2: Human-like behavior via neuroevolution of combat behavior and replay of human traces," in *Proceedings of the IEEE Conference on Computational Intelligence and Games (CIG 2011)*. Seoul, South Korea: IEEE, September 2011, pp. 329–336.

[9] R. Arrabales, A. Ledezma, and A. Sanchis, "Towards conscious-like behavior in computer game characters," in *Computational Intelligence and Games, 2009. CIG 2009. IEEE Symposium on*, Sept 2009, pp. 217–224.

[10] Cothran and J., "Winning the 2k bot prize with a long-term memory database using sqlite," 2009.

[11] G. A. Kaminka, M. M. Veloso, S. Schaffer, C. Sollitto, R. Adobbati, A. N. Marshall, A. Scholer, and S. Tejada, "Gamebots: A flexible test bed for multiagent team research," *Commun. ACM*, vol. 45, no. 1, pp. 43–45, Jan. 2002.

[12] O. Burkert, R. Kadlec, J. Gemrot, M. Bda, J. Havlcek, M. Drfler, and C. Brom, "Towards fast prototyping of ivas behavior: Pogamut 2." in *IVA*, ser. Lecture Notes in Computer Science, C. Pelachaud, J.-C. Martin, E. Andr, G. Chollet, K. Karpouzis, and D. Pel, Eds., vol. 4722. Springer, 2007, pp. 362–363.

[13] G. Acampora, V. Loia, and A. Vitiello, "Improving game bot behaviours through timed emotional intelligence," *Knowledge-Based Systems*, vol. 34, no. 0, pp. 97 – 113, 2012, a Special Issue on Artificial Intelligence in Computer Games: {AICG}.

[14] B. Soni and P. Hingston, "Bots trained to play like a human are more fun," in *Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, June 2008, pp. 363–369.

[15] B. Gorman, C. Thurau, C. Bauckhage, and M. Humphrys, "Believability testing and bayesian imitation in interactive computer games."

[16] R. Zhao and D. Szafron, "Generating believable virtual characters using behavior capture and hidden markov models," in *Advances in Computer Games*, ser. Lecture Notes in Computer Science, H. Herik and A. Plaat, Eds. Springer Berlin Heidelberg, 2012, vol. 7168, pp. 342–353.

[17] K. Hindriks, B. Riemsdijk, T. Behrens, R. Korstanje, N. Kraayenbrink, W. Pasman, and L. Rijk, "Unreal goal bots," in *Agents for Games and Simulations II*, ser. Lecture Notes in Computer Science, F. Dignum, Ed. Springer Berlin Heidelberg, 2011, vol. 6525, pp. 1–18.

[18] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: the rprop algorithm," in *Neural Networks, 1993., IEEE International Conference on*, 1993, pp. 586–591 vol.1.

[19] D. Whitley and T. Hanson, "Optimizing neural networks using faster, more accurate genetic search," in *Proceedings of the Third International Conference on Genetic Algorithms*. Morgan Kaufmann Publishers Inc., 1989, pp. 391–396.

[20] M. Rocha, P. Cortez, and J. Neves, "Evolution of neural networks for classification and regression," *Neurocomputing*, vol. 70, no. 1618, pp. 2809 – 2816, 2007.

[21] P. Hingston, "A new design for a turing test for bots," in *Computational Intelligence and Games (CIG), 2010 IEEE Symposium on*, 2010, pp. 345–350.

[22] R. Arrabales, A. Ledezma, and A. Sanchis, "Characterizing and assessing human-like behavior in cognitive architectures," in *Biologically Inspired Cognitive Architectures 2012*, ser. Advances in Intelligent Systems and Computing, A. Chella, R. Pirrone, R. Sorbello, and K. R. Jhannsdttir, Eds. Springer Berlin Heidelberg, 2013, vol. 196, pp. 7–15.

[23] "http://human-machine.unizar.es/."