A survey on Self-Organizing Maps applied to games

Artur Henrique Gonçalves Coutinho Alves

January 8, 2016

Introduction

Games have, for the last years, started to get recognition as a vast research area, as seen in the IEEE panorama of artificial and computational intelligence in games [1]. Video games can be both rich and dynamic tools for research and development of computational intelligence techniques and the focus of research itself. Neural networks in special have seen various applications based on games, ranging from game development and automatic playing to player analysis; it's in the latter that Self-Organizing Maps (SOMs) have most applications.

SOMs offer one basic functionality for researchers and developers when talking about games: extraction of player behavior. Collecting statistical data from players and processing it in order to reduce its complexity and facilitate visual analyses comprise a crucial process in various applications, such as social analysis of virtual societies, tactical analysis of sports and e-sports teams and, expanding from player analysis into automatic playing, non-player character behavior learning based on real players. This short survey presents six works where SOMs have been successfully applied to these scenarios.

NPC behavior learning

Combining Self Organizing Maps and Multilayer Perceptrons to Learn Bot-Behavior for a Commercial Game

Throughout the decades of game development, bot behavior has been usually determined by scripts or finite state machines. In [2], the authors argue that this can lead to an obsolete gameplay, since players are able to perceive such repetitive actions and react accordingly, removing the thinking and planning aspects and replacing them with experience. To improve on this, their proposal is to apply various computational techniques on human gameplay data in order to create adaptive bots. Utilizing id Software's Quake II, a famous and currently open source game, they extract gameplay recordings through network packets and input them into MATLAB, where a classifier is trained; the trained bot is then connected to the game server in order to actually play.

Technically, this approach uses a SOM to reduce problem complexity, i.e., it maps a high dimensional space into a lower dimensional manifold by clustering training samples; the system then associates two LM-backprop-trained MLP networks with six hidden neurons to each cluster, one for adjusting viewangle and another for adjusting velocity, the two components of the reaction vector. Therefore, the bot is able to map state vectors from the server into player reactions according to the sample data clustering. It's worth noting that the state vector is composed by the player's three-dimensional position, his scalar distance to the nearest enemy and his horizontal and vertical angles relative to such enemy. The online evaluation of the approach with the best offline results was arbitrary, made by observing its behavior ingame. The researchers say that it had good aiming - not perfect, but the training data wasn't either.

Player modeling

Player Modeling using Self-Organization in Tomb Raider: Underworld

One of the most important questions in game development is: how are players playing the game? Hundreds or even thousands of hours go into playtesting a game during creation in order to guide development. With increasing popularity, nonlinear games are specially challenging to evaluate, since players have multiple choices at any given moment; therefore, better testing approaches are needed. New game metrics (numerical data extracted from gameplay sessions) and machine learning techniques have made possible analysis based on player modeling, such as presented in [3], where researchers investigate dissimilar behaviors among EIDOS' Tomb Raider: Underworld players.

Data was collected using an in-house tool, made available by EIDOS for this research, and comprises 1365 players which had completed the game. From this data set, six features were extracted: game completion time, number of deaths, causes of death (three values) and help-on-demand requests. Initial analyses, using k-means and Ward's hierarchical clustering method, show that the expected number of clusters lies between 3 and 4. A 5000-neuron toroid-shaped SOM was then applied to the problem and, as expected, four clusters were found, displayed in Figure 1.

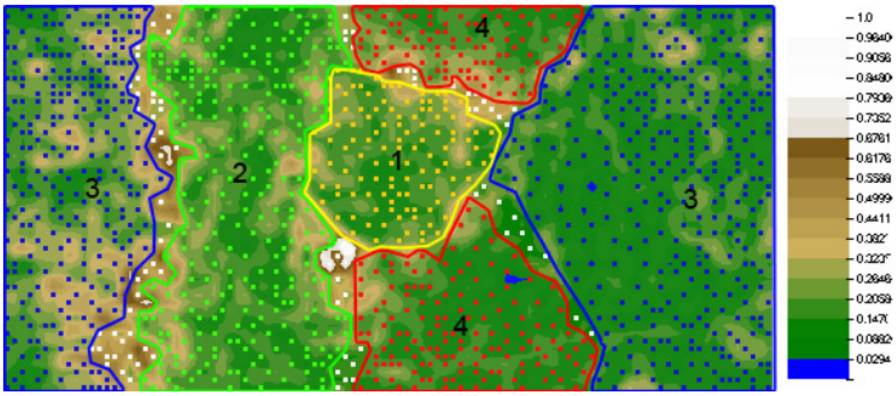
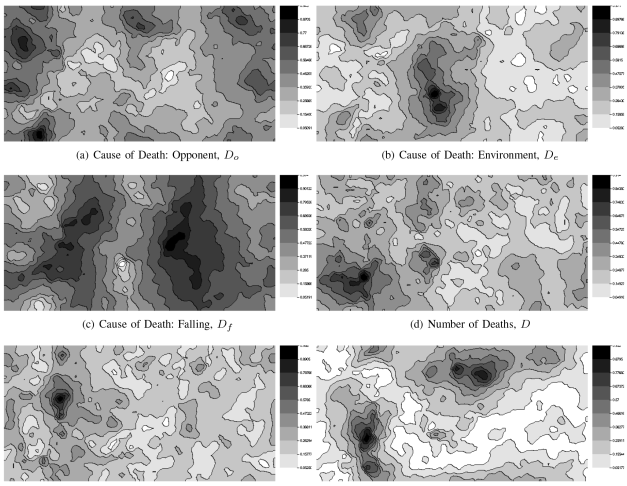


Figure 1: U-matrix from Tomb Raider: Underworld player data

Overlapping the U-matrix and the six component planes representing each feature, presented in Figure 2, the researchers were able to classify each cluster as:

1. Veterans: fast, have few help requests and die few times, mainly by environment
2. Solvers: slow-paced, have few help requests and die more by falling
3. Pacifists: fairly fast, have average help requests and die more by enemies
4. Runners: fast, have varying help requests and die often by enemies and environment



(a) Cause of Death: Opponent, D_o

(b) Cause of Death: Environment, D_e

(c) Cause of Death: Falling, D_f

(d) Number of Deaths, D

(e) Completion Time, C

(f) Help on Demand, H

Figure 2: Component planes of each extracted feature

Social analysis

Clusterization of an Online Game Community through Self-Organizing Maps and an Evolved Fuzzy System

Massive Multiplayer Online Role Playing Games (MMORPGs) have their origins on the old school table RPGs and are among the most played games in the last years. These games present a virtual world with adventures, professions, character customization and social structures that appeal to millions of players. These virtual societies, such as their real life counterparts, can be analysed by collecting statistical characteristics, clustering these data points and profiling such clusters. The work presented in [4] proposes a SOM-based approach to this social analysis.

Data was collected by interviewing brazilian Ragnarök players and expanded with generated random samples based on the original distribution. Each data point presents age, gender, state, occupation, character level, class, hours per week, period of the day, game events frequency, locale of play, player versus player (PvP) mode frequency and motivational factor. Not every data point is used in the clustering phase, in order to use only the most relevant information; this classification is made by a fuzzy system that takes level and usage time as inputs and outputs the respective relevance (players with higher level and higher usage time are more relevant). The fuzzy rule operates on Gaussian functions with parameters defined by a genetic algorithm - the GA aims for the lowest SOM error by changing the fuzzy system parameters, filtering the data points and training the network.

Two errors can be used to determine the quality of the map: the quantization error, representing map resolution, i.e., the average distance between each data point and its best matching neuron, and the topographic error, representing map topology preservation, i.e., the proportion of data vectors whose two best matching neurons are not adjacent. Tests were ran with different weights for each error type and the researchers conclude that the topographic error is a better fitness factor, presenting better defined clusters and lower noise levels.

Uso de redes de Kohonen para identificação de perfis de jogadores no World of Warcraft

Following the work presented in [4], [5] once again uses a SOM to analyse the social behaviors of MMORPG players, this time in World of Warcraft. Here, the concepts of cooperative and competitive interactions are considered and ingame player groups, called guilds, are used as data filtering. Data points were collected using a game plugin and represent 17697 players; only players from the eight largest guilds were used in training, since the largest guilds theoretically present the best typical class behaviors. The 591 remaining data points yielded the clusters presented in Figure 3, where the U-matrix shows the total clustering and the other maps represent classes and two variables. These results were analysed based on current game state (such as class balancing, which varies greatly throughout the years) and relations between clusters and specific class characteristics are found, e.g. classes with better solo capabilities aren't as commonly seen in groups as those most dependent on others.

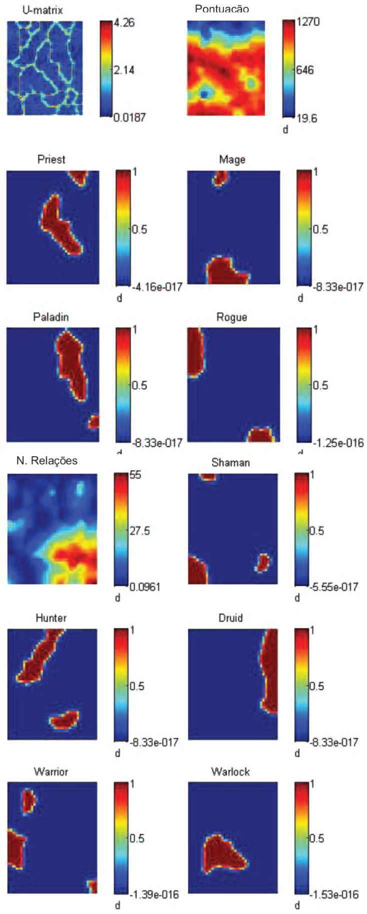


Figure 3: World of Warcraft player analysis through clustering

Tactical analysis

Analysis of players' configurations by means of artificial neural networks

Not only video games can benefit from the analytical power of SOMs. Sports have, for a long time, used tactical analysis in order to study opponents and improve strategies. On the other hand, technology has found various applications in sports, ranging from performance prediction to whole activities, such as robot soccer. One application of SOMs in sports is presented in [6], where the authors propose a method to detect player configuration patterns in volleyball. Such configurations represent the *when* and *where* of players during a match and are heavily influenced by employed tactics. Since tactical flexibility, that is, the use of variable attack and defense configurations, is supposedly necessary for success, this application may predict team performance and be used to study opponents and develop better tactics and strategies.

A SOM is used to map the time series derived from telemetry into a two-dimensional neuron grid, where each cluster represents a specific configuration, as seen in Figure 4. Since the data is time continuous, a Dynamically Controlled Network (DyCoN) is used, because it allows continuous training. Data was collected during the first round of the German team in the 2002 World Cup, yielding five match data sets, then split into rallies, of which one per match was randomly chosen. The data contains the x and y coordinates of play1.0ers at 25Hz. Each rally is expanded using a Monte Carlo data generator and the generated samples are used for initial training, since they are virtually unlimited. In the second phase the original data is used, and, after training, the network can be tested with other original samples.

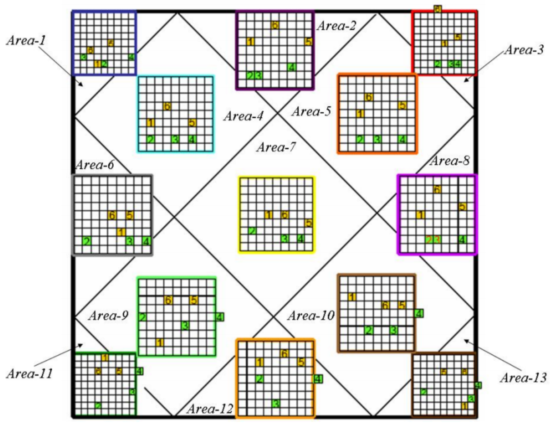


Figure 4: An example of SOM mapping team configurations

Initial experiments show that this approach is not successful in detecting trajectories of configurations, but is able to detect individual configurations. Therefore, the final analysis, focused on the World Cup final (Germany vs. Italy), presents only target configurations for each trajectory. The results are demonstrated in Figure 5. It is possible to see that, while the German team is more rigid, employing fewer configurations more frequently, the Italian team is more flexible, with more configurations less repeated. Italy won the match; tactical flexibility seems to be a deciding factor, as expected from the literature.

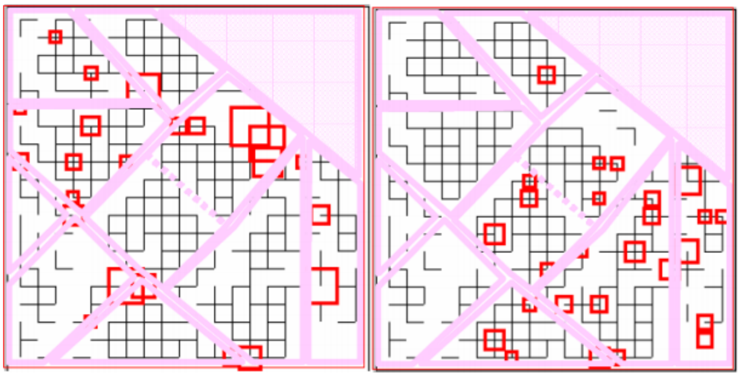


Figure 5: Configurations employed by the German (left) and Italian (right) teams: each red square represents a configuration, with larger squares meaning more frequency.

Tactical pattern recognition in soccer games by means of special self-organizing maps

Advances in technology have made high-level telemetry a reality, resulting in enormous amounts of data to be analysed; better computational techniques are needed in order to process such information. Improving on [6], [7] uses a SOM to extract tactical patterns from soccer games - specifically, to classify short and long game initiations, i.e., time frames between the defense winning the ball and losing the ball. As in the previous work, a DyCoN-based network was used; each neuron maps a constellation of data, such as the one in Figure 6. The goal of the SOM is transforming a time series into a constellation vector of four players, allowing the visual analysis of configuration trajectories, like in Figure 7.

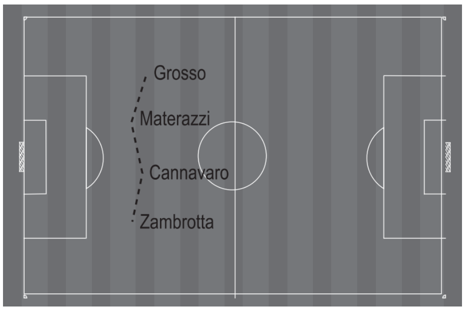


Figure 6: Constellation of data represented by a single neuron

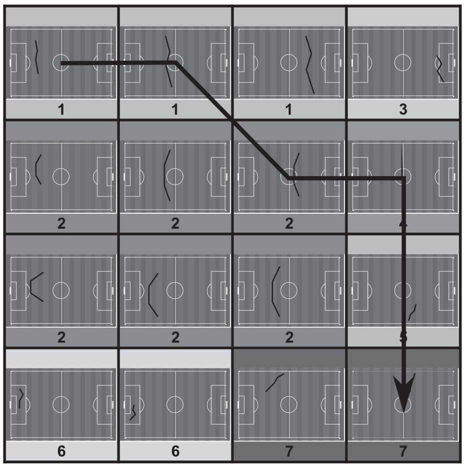


Figure 7: Constellation vector in a SOM

An hierarchical architecture was developed with three phases, where each phase has different SOMs. The first phase treats the x and y coordinates as different constellations for the defense team, the attack team and the ball. The second phase describes the team movement based on the typified constellations returned by the first phase. The third phase describes the game situation as a whole, considering both teams and the ball. This architecture can be seen in Figure 8. Since each time series, comprising one game pattern, can have varying sizes, the weight vectors would be different for each constellation vector. Therefore, instead of using different networks for each length, a sliding window technique was applied with a fixed size equal to the length of the smallest pattern.

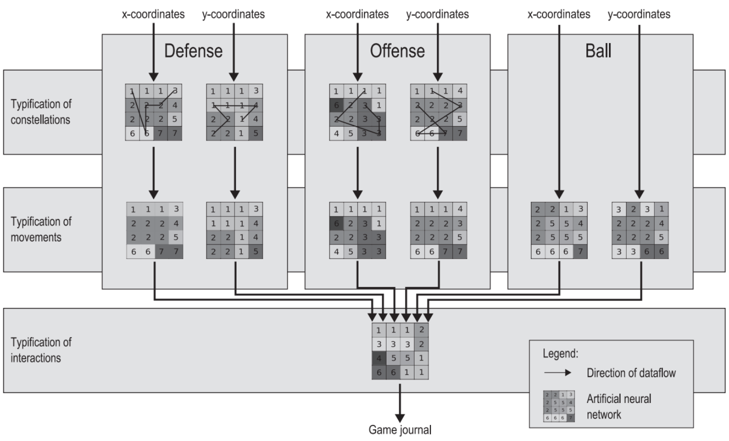


Figure 8: Hierarchical architecture

Data was collected during the 2006 World Cup final, where the match between France and Italy yielded 165235 data sets at 25Hz, where each data set contains the x and y positions of all 22 players and the ball. The database was reduced to one data set per second, effectively returning 6613 data sets. An expert manually classified short and long game initiations in order to color the clusters after the third phase. After training, the system was tested and, of the 131 manually categorized game initiations, it was able to detect 84%. The authors say that the sliding window technique was the main culprit of lower performance, but for the time being it still was the best approach. Improvements in this area, as well as in accepting varying numbers of involved players, are regarded as future work. Nevertheless, the main goal of the work, which was reducing the time needed to analyse match data, was successfully met.

Conclusion

Behavior analysis is an important step both in game development and in sports strategy, and clustering techniques and neural networks can be used to facilitate such analysis. Regarding sports, where tactical analysis has been used for decades, data has become too large and too complex for viable manual analysis, making computational assistance a must. Meanwhile, in virtual games, game developers and researchers are too estranged, missing cooperation opportunities that could benefit both sides, as seen in some works presented in this survey. The industry could, for example, improve player analysis processes and create better games faster, while academic researchers would have a plethora of applications to test new approaches and techniques. Nevertheless, these works have shown that SOMs are capable of solving various problems for both academia and industry.

References

[1] G. N. Yannakakis and J. Togelius, "A panorama of artificial and computational intelligence in games," 2014.

[2] C. Thurau, C. Bauckhage, and G. Sagerer, "Combining self organizing maps and multilayer perceptrons to learn bot-behaviour for a commercial game." in *GAME-ON*, 2003, p. 119.

[3] A. Drachen, A. Canossa, and G. N. Yannakakis, "Player modeling using self-organization in tomb raider: Underworld," in *Computational Intelligence and Games, 2009. CIG 2009. IEEE Symposium on.* IEEE, 2009, pp. 1–8.

[4] L. C. Rodrigues, C. A. Lima, P. Oliveira, and P. N. Mustaro, "Clusterization of an online game community through self-organizing maps and an evolved fuzzy system," in *Natural Computation, 2008. ICNC'08. Fourth International Conference on*, vol. 2. IEEE, 2008, pp. 330–334.

[5] L. C. Lopes and P. N. Mustaro, "Uso de redes de kohonen para identificação de perfis de jogadores no world of warcraft."

[6] J. M. Jäger, J. Perl, and W. I. Schöllhorn, "Analysis of players' configurations by means of artificial neural networks," *International Journal of Performance Analysis in Sport*, vol. 7, no. 3, pp. 90–105, 2007.

[7] A. Grunz, D. Memmert, and J. Perl, "Tactical pattern recognition in soccer games by means of special self-organizing maps," *Human movement science*, vol. 31, no. 2, pp. 334–343, 2012.