Behavior Analysis Through Games Using Artificial Neural Networks

Didier PUZENAT
GRIMAAG, Université Antilles-Guyane
Pointe-à-Pitre, France
didier.puzenat@univ-ag.fr

Isabelle VERLUT
Centre Hospitalier Universitaire
Pointe-à-Pitre, France
isabelle.verlut@chu-guadeloupe.fr

***Abstract*—This paper demonstrates that a human being using an interface can be efficiently evaluated – in real time – by embedding basic measurements in the interface and using a suitable trained artificial neural network. The approach is introduced through video games but is suitable for any machine capable of valuable measurements on user actions. Of course, the quality of the "diagnostic" depends of the learnability of the task and of the size and quality of the learning base. Typical applications include the detection of fatigue, stress, emotions, the influence of a drug or of medical treatments ; screening a deficit or adequateness to a task, etc. Two successful prototypes are presented, one to predict the mental age of children through a set of simple basic games, and the other to detect if a subject is right-handed of left-handed through a racing car simulation.**

***Keywords*-User interfaces; Games; Neural network applications; Cognitive science; Psychology; Human factors.**

I. INTRODUCTION

The present article introduces and explores the possibilities of analyzing a player by instrumenting video games. By "instrumenting" we mean taking measurements of player actions on input devices such as a keyboard or a joystick. Such an analysis could, for example, detect fatigue, stress, emotion, or the influence of drugs or medical treatments. An instrumented game could also be used to conduct screening, for example to detect a deficit or an inadequateness to a given task. In fact, the potential applications are extremely diverse. Of course, such evaluations are conducted every days without video games and even without computers, for example to detect a deficit, the typical procedure would be for a psychologist to conduct interviews and classical psychometric tests. However using video games provides additional dimensions: attractiveness, reduced dependence regarding culture and language, reduced costs, more accurate measures, and so on. Furthermore, the analysis can be performed as a secondary task while playing (for a game) is the main task. In addition, the evaluation can be conducted in real time, offering a wide range of new possibilities. As an example, a video game could estimate the age (or maturity) of the player and adapts the difficulty of the game, the complexity of the narration, or the level of violence.

The key of the success of our approach is the use of artificial neural networks to perform the behavioral analysis, as presented in section II. This approach has been validated through two prototypes. A first application, presented in sections III to V, estimates the mental age of a subject via a small set of games. The second application, presented section VI, determines if a subject playing a racing car simulation is right-handed of left-handed after a few laps.

II. ARTIFICIAL NEURAL NETWORKS

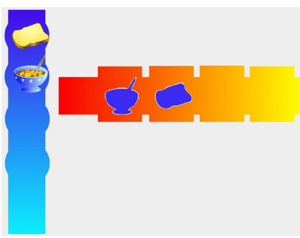
Playing a video game implies mobilizing many skills. Depending on the game, it may be more or less necessary to think, to understand (the rules, the environment, enigmas, etc.), to develop a strategy, to store information, to act quickly, to act with precision, to manage stress, to manage fatigue, etc. Thus, instrumenting a game should give valuable informations on the player. However, the point is to know what to measure and how to go from the measurements to a conclusion (for example, a diagnostic). The originality of this paper is to use artificial neural networks to select and combine relevant measures to assess the player. Once trained on a representative population, a dedicated neural network will be able to evaluate a new player. Our approach eliminates the need to explicit the significance of measures for a given goal, therefore it becomes possible to make "low level measurements", e.g. the frequency of corrections on the steering wheel during a racing car simulation. Beyond games (and "serious games"), the method developed could be used in human-machine interfaces or incorporated into non-computer equipments. As an example, analyzing car drivers fatigue should easily be embedded into a real car, and at low cost since (i) sensors are already in place, and (ii) an artificial neural network involves a quite small amount of computational power (excepted for the learning phase).

Among all artificial neural network models, for all our present developments, we have chosen a classical Multi-Layer Perceptron (MLP) with a backpropagation learning algorithm [1]. The neural network has been implemented with the free open source *Fast Artificial Neural Network Library* (FANN) [4]. FANN is a C library but binding is provided for most languages, facilitating the use of the developed neural network within the instrumented video game (or interface, or computer embedded in a machine, etc.), e.g. to perform a real-time evaluation of the subjects.

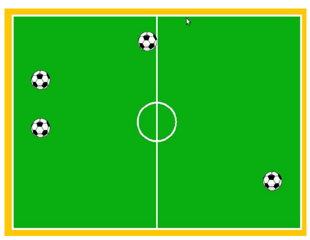
III. Collecting data from young children

A battery of games based on the Gcompris suite [2] has been instrumented in order to collect data about young children way to play, to create a database suitable for multiple experiments. The Gcompris suite is an educational software under General Public License (GPL) comprising of numerous activities for children aged 2 to 10. Using a GPL software avoids the need to write games “from scratch”, gives access to the code to insert the measures, and makes it possible for us to redistribute the whole application. For now, 12 games have been adapted, figure 1 presents some screenshots and the complete list of games is given in table I.

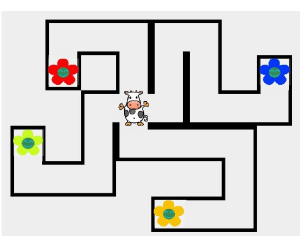
To make the application autonomous, and to avoid experimental conditions variability from a subject to another, instructions are given by an audio message before each game while a video shows a user playing the game about to start. Finally, for each game, the task is repeated several times at increasing speed and/or difficulty, a voice message recalls and eventually details what to do at each iteration, e.g. “*point the blue duck*” for the so called “colors” game (figure 1).



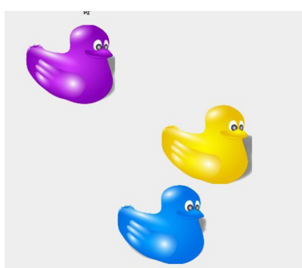
“shadows” (2)



“super balls” (11)



“super maze” (4)



“colors” (0)

Figure 1. Screenshots of 4 of the 12 instrumented games suitable for 2 to 8 years old children. Numbers refer to table I.

Our battery of games runs on a lowcost touchscreen computer (figure 2). A touchscreen is the perfect input device for the youngest subjects. Furthermore using a touchscreen eliminates the bias of children familiar (or not) with more complex input devices such as mouses, trackballs, touchpads, or joysticks. The computer is running under a GNU / Linux operating system (Ubuntu 8.10 desktop version), but all developments have been done in Java and games have been successfully tested on Microsoft Windows XP running on the same PC, with limited changes (filesystem pathways).

Figure 2. Asus “eeeTop” lowcost touchscreen computer used for the all experiments. The computer has been used without any keyboard or mouse.

A database with measures from games and any available data (e.g. age and sex) has been build with 36 children from 2 to 8 years old. Each child has been playing the 12 games, during 20 minutes. Data has been collected in a public school in Guadeloupe, experiments took place in a dedicated but noisy room, with a psychologist. Most children were from first and third year of nursery school (French “*maternelle*” school) witch introduces a small gap in age distribution for 1500 to 2000 days old children (figure 3), one child was older (French “*primaire*” school) with 2998 days since birth.

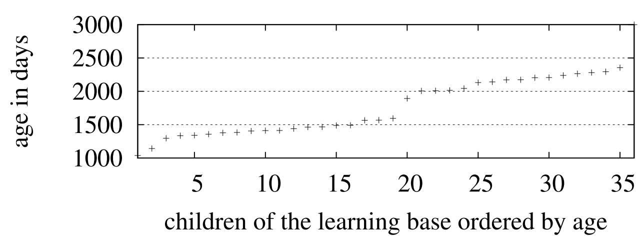


Figure 3. Age of the 36 children of the learning base.

0	A duck has to be finger-pointed according to its color.
1	A cow has to be finger-dragged through a maze to a flower.
2	An object and its shadow have to be bound.
3	An animal appearing on the top of the screen has to be found (& selected) among animals at the bottom of the screen.
4	Same as (1) but the cow has to be reach a precise flower according to its color.
5	Same as (0) but the duck to point depends of its color and of its position (ducks can be upside-down and go left or right).
6	Raccoons appear (and disappear) and must be pointed.
7	Same as (2) but pairs of semantically related objects have to be bound (e.g. a bottle and a glass).
8	A ball, crossing the screen and bouncing on the sides of the screen has to been finger-pointed.
9	Same as (8) with 2 balls.
10	Same as (9) with 3 balls.
11	Same as (10) with 4 balls.

Table I

A short description of the 12 instrumented games.

IV. Predicting children ages

As a first experiment to validate our approach, an artificial neural network has been trained to predict the age of the player (in days since birth). Since it is based on the subject behavior, the estimation of the age of the player made by the neural network can be seen as an estimation of the mental age of the subject. Such a challenge is of great interest and does not implies a high experimentation cost since the actual age of the subject is known. The neural network is trained to associate measures recorded with the age of the player.

Only a fraction of the measures from the database presented section III has been used for this specific age prediction application. Indeed only 8 measures from 4 games (2 measures per game) have been used among the hundreds of measures recorded for the 12 games. These 8 measures, feeding the 8 input cells of the network, have been enough to reach a good age prediction. The use of more input cells (i.e. more measures) for this specific task would probably increase the network capability to give accurate answers for examples that are already in the learning base. However, with a constant number of subjects, adding input cells may have reduced the network capability to give accurate answers for examples that were not is the learning base, that is its "generalization capability". Indeed, more examples for the learning base (i.e. more children) are needed to learn valuable weights for the additional synaptic connexions.

As mentioned section II, we will use a MLP artificial neural network with back propagation, the hidden layer will be of the size of the input layer (8 cells), and the output layer will have only one cell giving the age. Hidden and output layer cells use respectively sigmoid and linear activation functions. The classical way to train and test such an artificial neural network is to split a huge set of examples into two sets with an empty intersection: one to train the network (learning set) and one to test the network (generalization set). However, for our present application, 36 examples will not be enough to build two large enough sets, and collecting more data in a school is quite time consuming. Therefore, for all our experiments, we will use all but one examples for the learning set (i.e. 35 children), and we will test the network on the remaining example (1 child). This procedure will be repeated 36 times, extracting each time a different example. As a result, we will virtually have a 35 examples learning set and a 36 examples generalization set.

First successful results have been obtained with an arbitrary selection of 4 games. In order to try to build a better combination of games, the generalization performance of each individual game has been tested using a neural network per game (i.e. with only 2 inputs). The prediction capability of these 12 neural networks (based on only one game) was quite poor, meaning the efficiency of the prediction comes from the combination of games. Furthermore, combining the best individually wining games did not lead to significant improvement regarding our arbitrary selection. Finally, the 495 combinations of 4 games (from the 12 games) have been tested, the 80 best combinations gave valuable predictions. Table II presents the 10 best combinations according to the standard deviation of the error between the predicted age and the real age of the children. Numbers identifying games can be "decoded" with table I, and some screenshots are presented in figure 1. The lowest standard deviation, table II, is of 102 days (i.e. about 3 months), for this combination the median error is of 83 days, the maximum error is of 503 days (about 1 year and 4 months), and the smallest error is null meaning the network found the exact age of one of the child. Such results are comparable to best psychometric tests just by playing a few minutes since only 4 games were needed, thus our approach could be quite valuable to design new tools for (neuro)psychologists.

rank	*combination of games*	*standard deviation*	*median error*	*max. error*	*min. error*
1	0, 2, 3, 8	102	83	503	0
2	0, 2, 9, 10	106	54	465	1
3	1, 3, 8, 11	108	95	639	2
4	2, 5, 6, 8	111	81	501	5
5	0, 2, 5, 10	125	74	517	4
6	0, 2, 4, 6	126	90	540	1
7	0, 2, 3, 10	131	62	532	0
8	2, 4, 5, 6	132	72	502	6
9	0, 8, 9, 11	132	83	634	1
10	1, 2, 4, 7	137	98	687	4

Table II
THE 10 BEST COMBINATIONS (FOR STANDARD DEVIATION) TO PREDICT CHILDREN AGE, WITH THE CORRESPONDING ERROR IN DAYS.

V. Estimating the mental age of patients

The application has been used with 4 children followed by a medical institution for suspicion of mental disorders, in order to try to estimate their "mental age". We will compare the answer of the neural network with the evaluation of the mental age done by classical psychometric tests. Note that none of the 36 children of the learning base are known to have mental disorders. Table III compares estimations done by the network for the 4 patients, with a range of probable mental ages given by psychometric tests (detailed below). Last column gives an indication of the error of the network by presenting the number of days missing to be in the range.

	physical age	*neural network age estimation*	*psychometric tests age range*	*distance to be in range*
A	2420	1405 days	[1731, 2240]	326 days
B	2273	1503 days	[1347, 1452]	51 days
C	2169	2165 days	[1888, 2138]	27 days
D	2478	1405 days	[1504, 2651]	99 days

Table III
RESULTS FOR THE 4 PATIENTS USING GAME COMBINATION (0, 2, 3, 8).

The estimations given by our best neural network are systematically below the physical age of the subjects (table III), which is also the case for psychometric test ranges, meaning the network does detect the deficits. The network almost predicted the exact physical age of the patient referred as patient C which is compatible with the fact that this patient has not definitively been diagnosed to have mental disorder (but a psycho-motor instability syndrome due to social issues is discussed). Our neural network interestingly highlights the normality of patient C, more than classical tests do, possibly because abilities measured in our games differ from school works, showing the interest of varying test tools to observe the potential of a subject.

However, the fact that none of the answers of our neural network are within psychometric test ranges is disappointing. Disappointing but logical. Indeed, by selection, configuration (0, 2, 3, 8) is the best to discriminate between normal subjects, while patients A, B and D present documented physical troubles impacting respectively verbal working memory, acquisitions, and language. Since patients A, B, and D are not homogeneously developed, the notion of mental age should be associated with a precise cognitive function, that is why we used ranges reflecting lowest and highest results of psychometric tests. Consequently, the mental age of a patient depends on the chosen games. For example, combination (1, 2, 4, 7), ranking 10 in table II, finds a mental age within ranges for all patients (table IV) and still performs well for normal subjects with a median error of 98 days and a standard deviation of 137 days, versus a median error of 83 days and a standard deviation of 102 days for the best configuration (table II). Of course, more patients would help to confirm the interest of configuration (1, 2, 4, 7).

	physical age	*neural network age estimation*	*psychometric tests age range*	*distance to be in range*
A	2420	2106 days	[1731, 2240]	0 day
B	2273	1388 days	[1347, 1452]	0 day
C	2169	2107 days	[1888, 2138]	0 day
D	2478	2106 days	[1504, 2651]	0 day

Table IV
RESULTS FOR THE 4 PATIENTS USING GAME COMBINATION (1, 2, 4, 7).

However, for patients with cerebral damage(s), rather than an estimation of a global mental age, psychologists need understanding, cognitive function by cognitive function. Therefore, a strategy could be to build a learning set with normal subjects and patients, and to use the result of function specific psychometric tests to drive the learning phases (desired output). But such an approach implies a significant amount of patients, and the results of the psychometric tests for each example. Hopefully, patients are generally followed by a medical institution so such tests have already been performed, but all normal subjects still have to be tested. We plan to apply this strategy as soon as possible.

VI. CAR RACING SIMULATION

This section presents the instrumentation of an action game, subjects are not anymore young children, but computer science students (from the *Université des Antilles et de la Guyane*). Our hypothesis is that it is not necessary to use educative games. Of course, a memory game is valuable to test memory, a maze game is valuable to test spacial representation, and a reflex game is valuable to test reflexes. However, general purpose games as a car simulation or a First Person Game (FPG) are potentially rich enough for a neural network to find valuable measures (among enough measurements) and to combine them for a given task.

As a prototype, we have developed a racing car game dedicated to a simple task: detecting if the player is right-handed or left-handed, in real time, and after only a few minutes. Once again, a free open source software has been chosen to access the code and to eventually disseminate the instrumented game without restrictions (except those of the GPL). The game is "SuperTuxKart" [3], figure 4 shows a snapshot of the game. This game has been chosen since it is relatively linear, for example the driver cannot decide to drive in the opposite way or to take a short cut, furthermore the car cannot get stuck which would compromise the recordings. SuperTuxKart was relatively easy to instrument because the code is in standard C++ and well structured. For all our experiments, the car is controlled by an analogical joystick, but the implementation supports a steering wheel without any change, however changing the controller would definitively implies to rebuild the learning base.

Figure 4. Snapshot of the "SuperTuxKart" open source game.

For each full lap of the circuit, a vector of 4 components is preserved and will feed the artificial neural network. Those components are (i) the average number of steps per rotation (the joystick is analogical), (ii) the number of accidents (collisions and falls), (iii) the total number of actions of the joystick, and (iv) the number of changes in direction (e.g. the driver was turning left and is now turning right).

As mentioned in section II, we will use a MLP artificial neural network and back propagation. For this application, the MLP will have 4 cells on the input layer (for the 4 measures already detailed), 8 cells on the hidden layer, and one cell on the output layer. Hidden and output layer cells use sigmoid activation functions. Depending on the value of the output, the answer will be that the subject use his dominant hand or not. Since we know which hand the subject use to play, we can deduce if he is right-handed of left-handed. A strict experimental protocol has been established to minimize experimental conditions variability from a subject to another. The subjects do not know what is the purpose of the experiment, they are just asked to complete a lap as fast as possible. Each subject drives 4 laps of the circuits. The 2 first laps help the subject to be comfortable with the game and the experiment, no measures are recorded. The driver can use his left or right hand but (i) he must not change during the lap and (ii) he must change for the next lap. The 2 last laps are used to collect the measures.

A database has been built with 20 subjects, resulting in 40 examples: 20 for the dominant hand and 20 for the non-dominant hand. The methodology presented section IV is used to compensate the not so large size of the database, thus the network will be trained with a 39 examples learning set and tested with a virtual 40 examples generalization set. The difficulty to find subjects willing to play for 10 minutes has been a surprise. Such an attitude contrasts with the motivation of young subjects playing our modified Gcompris games (section IV), since some children asked to start again at the end of the 20 minutes. A probable cause is that students are used to play commercial games. SuperTuxKart can compete with Super Mario Kart but not with the last car simulations running on last generation consoles. This aspect must be taken into account for future works, a game focusing on young adults must be very attractive.

Nevertheless, even with 20 subjects for learning, results are impressive since in generalization the neural network is able to find which hand is used for 95 % of the subjects. Furthermore, the trained neural network has been embedded in the game and the application is able to reassess frequently the player without slowing the game. This point is useless for dominant hand prediction, but it could be valuable for some applications. For example a warning led to alert the driver of a dangerous behavior (e.g. due to fatigue).

VII. Conclusion

"A ship sails the ocean. It left Boston with a cargo of wool. It grosses 200 tons. It is bound for Le Havre. The mainmast is broken, the cabin boy is on deck, there are 12 passengers aboard, the wind is blowing East-North-East, the clock points to a quarter past three in the afternoon. It is the month of May. How old is the captain?" Gustave Flaubert (1821-1880), in a letter to his sister Caroline (1843).

This article does not solve Flaubert's problem. However, it may be possible to find the age of the captain by instrumenting his boat. For now, we have shown that it is possible to instrument a game and find valuable information about the player by using an artificial neural network. The instrumentation by itself does not depend of an application since the neural network can select which measures to effectively use, however too many inputs implies a large learning base and a long training. The quality of the learning base is a critical point, it implies motivated subjects, representative of the whole population, and the availability of the desired output for those subjects. Then, using artificial neural network is not an issue, and a classic MLP model performs perfectly.

The application presented in section IV is capable of predicting the age of a young subject after about 10 minutes of gaming, with an accuracy comparable to best psychometric tests since median error is about 3 months. The same network, without adapting its training, is able of a valuable estimation of the mental age of a child with mental disorder, for a given task. Finally, the application presented section VI is capable to predict if a subject playing a car simulation game is right or left handed with 95 % of success. All this results have been obtained with small amount of subjects, and in non-optimal conditions (e.g. noisy environments).

The next step is to mount collaborations in order to build large databases, with numerous subjects. Two domains will be soon explored. A tool for psychologists to help detecting deficits at low cost and for large populations, with a battery of games. And a car embedded application, starting with a simulation, to detect a dangerous behavior for a car driver, e.g. due to fatigue, alcohol, or the influence of a medicine.

References

[1] C. M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, 1995.

[2] Bruno Coudoin. *GCompris, educational software suite for children aged 2 to 10*. `http://gcompris.net/-en-`.

[3] Joerg Henrichs, Marianne Gagnon, and Constantin Pelikan. *SuperTuxKart, Free 3D kart racing game*. `http://supertuxkart.sourceforge.net`.

[4] Steffen Nissen. Implementation of a fast artificial neural network library (FANN). Technical report, Department of Computer Science University of Copenhagen, October 2003.

Thanks

This work has been done with the help of Audrey Robinel and Rémy Siouray, IT master students, under the direction of Didier Puzenat (doctor in computer science) and Isabelle Verlut (doctor in neuropsychology).