Shell *et al.*: Bare Demo of IEEEtran.cls for IEEE Journals

Training a Gaming Agent on Brainwaves

Bartolomé Francisco, Moreno Juan, Navas Natalia, Vitali José,
Ramele Rodrigo, *Member, IEEE,* Santos Juan Miguel

***Abstract*—~~The field of Brain Computer Interfaces (BCI) has seen a bloom in the past few years. One of its applications is the training of systems using signals originated from the human brain. Out of all the methods used to acquire information from the central nervous system, the one that has gained most popularity is Electroencephalography (EEG) , mainly because of its effectiveness, low cost and its noninvasive characteristic. Event-related potential (ERP) are brain signals found on EEG that are time-locked to a particular event.~~ In this study, we propose a simple game scenario that can be used to trigger a feedback response embedded in Electroencephalographic (EEG) signals of an observation human critic that observes an agent playing a game. Based on a Reinforcement Learning (RL) model, the gaming agent receives rewards for their actions on the game and learns its optimal policy. These rewards are obtained by implementing a Brain-Computer Interface (BCI) system that identifies signal components called Error-related ~~potential~~ Potentials (ErrP)~~are a particular type of ERP that can be elicited by a person who attends~~ , which occur when a person witnesses a recognizable error. ~~When subjects observe an agent who makes a mistake this potential can be discerned by analyzing the signals which are directly related with the cognitive interpretation of an erroneous outcome or decision. Reinforcement Learning (RL) is an Artificial Intelligence approach where an agent tries to maximize rewards obtained while interacting with an unknown environment. In this work, a gaming agent is trained using RL, and the feedback obtained from brainwaves of human critic observers. Results show that there is an effective transfer of information and that the agent learns successfully to solve the game efficiently~~We perform an experiment where rewards are obtained from the ErrP signals from observational human critics watching an agent playing randomly a game. The agent is iteratively trained by receiving these rewards and updating its policy, improving its overall performance. Our results are expressed in threefold: (i) the structure of a simple grid-based game that can elicit the ErrP signal component; (ii) the verification that low classification accuracy of just above chance level that produces noisy rewards is enough to allow an agent to learn the optimal policy; (iii) collaborative rewards from multiple observational human critics can compensate the lack of accuracy or the limited scope of transfer learning schemes.**

***Index Terms*—ErrP, BCI, EEG, RL, Agent, AI**

I. Introduction

THE effectiveness of today's human–machine interaction and artificial intelligence is limited by a communication bottleneck, as humans are required to translate high-level concepts into a machine-mandated sequence of instructions ~~[?] . This work tackles this problem by exploring the use of brain signals as an interface between human and computer, trying to provide information to the system without explicit communication from the user. This information is used to make a gaming agent improves its operational performance using electroencephalography (EEG) signals as feedback of the performed task,~~ [?], [?]. Hence, new interaction methods are required to increase the communication bandwidth between computers and humans or to produce alternative communications systems to increase the efficiency of this channel. In this respect, video games have been widely used as test tools to assess new means of interactions [?], [?]. Video gaming agents are computer programs that can sense the computer game environment, process information, and react accordingly within the environment. They are used in the context of testing and evaluating artificial intelligence algorithms that aim to win the game or to behave like a real user player [?]. In this work, the feedback obtained from an observational human critic ~~. The idea is that~~ (OHC) in the form of electroencephalographic (EEG) signals is used to evaluate the operational performance of a gaming agent. Observational human critics are silent subjects observing a computer ~~game playing , can train the gaming agent using only signals~~ gaming agent playing the game.

The feasibility of a distinct non-biological communication channel between the Central Nervous System (CNS) and a computer device has been previously proven with Brain Computer Interfaces (BCI) or Brain Machine Interfaces (BMI). [?]. BCI systems provide a new input modality that can be used in the context of a computer game [?], [?]. This advancement is relevant in the context of the accessibility for video games [?] and the growing area of e-sports [?].

In this study, gaming agents are trained using only signal components called Error-related Potentials (ErrP) that can be identified ~~from their~~ in the observer's brain signals. These types of signals can be found on EEG traces and are elicited when ~~a subject is~~ subjects are aware of the presence of an unexpected outcome, which ~~she/he identifies~~ they identify as an error. ~~It~~ The analysis of ErrP signals is currently an extensive area of research in the neuroscience community [?]. Error-related Potentials can be detected by signal processing and machine learning techniques [?] and ~~they~~ are also used in Brain-Computer Interfaces ~~(BCI)~~ to implement or enhance artificial communication channels [?].

~~In order to train the agent, RL [?] comes~~ Given the scenario, Reinforcement Learning (RL) [?] stands out as a natural ~~solution for this scenario from the field of Artificial Intelligence. RL is~~ method to train the agent. Reinforcement Learning refers to an algorithmic learning strategy ~~that can be established by mimicking~~ inspired on how biological agents learn ~~from its environment by exploring it and getting feedback rewards, either~~ by exploring their environment while getting negative or positive ~~. This strategy~~ feedback rewards. The method aims to maximize ~~the amount of~~ positive rewards

R. Ramele and J.M.Santos are with the Department of Computer Engineering, Instituto Tecnológico de Buenos Aires(ITBA), Argentina, e-mail: rramele@itba.edu.ar.

Manuscript received December 9, 2019; revised August 25, 2020.

while ~~keeping the number of negative feedbacklow~~minimizing negative feedback. Thus, the learning problem is posed as a ~~mere~~ stochastic optimization strategy ~~[?]~~[?]. Recently, this technique has ~~seen a come-back. Nonneglected is the~~ been used extensively in the context of advances in artificial intelligence [?]. The influence of DeepBrain's AlphaGo project ~~, which~~ cannot be neglected, since it was the first to reach a very high proficiency when it won the complex game Go against several world champions [?].

Previous research has explored the usage of RL with reward signals based on brain activity, recorded by an EEG-based BCI system during task execution. The papers [?], [?], [?] have successfully demonstrated that a robot can be controlled ~~by obtaining a reward signal~~ with brain signals from a person ~~'s brain activity which is observing the~~ that is observing a robot solve a task. Moreover, a growing number of studies have demonstrated the feasibility of using ErrPs as rewards for RL schemes such as to enhance robotic behaviour [?], to assess air traffic controller's decisions [?] or to categorize actions as errors [?]. Other approaches have used ~~the signal as an~~ these signals as important feedback for human-robot interaction or to implement shared-control strategies [?]. Additionally, ErrPs have also been used in the context of games as an additional feedback channel that can be explored to ~~enhance~~ improve gaming experience [?], [?]. ~~In this same line, the objective of this work is~~

Therefore, we aim to use the information extracted from brainwaves to enhance the performance of a gaming agent~~, replicating and extending the usage of this signal~~. The three contributions are (1) a simple game that can elicit the ErrP potential, (2) results that confirm that even when ErrP classification accuracy is low and produces a noisy reward signal, enough information is generated for an agent to learn the optimal policy and solve a simple game and (3) collaborative rewards from multiple observational human observers can compensate the lack of classification accuracy or the inefficacy of transfer learning procedures for brainwaves signals.

In Section II the general layout of the cognitive game is described. ~~Section II-A outlines the processing pipeline used to detect~~ Sections II-A and II-B outline the cognitive game procedure used to obtain rewards in the form of ErrP components. ~~The following Section II-A describes the agent learning phase. Results and Conclusions~~ Section II-C describes the gaming agent learning procedure. Lastly, results and conclusions are exposed in Sections III and IV.

II. Materials and Methods

The experimental procedure is summarized in Figure 1. The proposed system has two distinct parts. This first part consists ~~in collecting~~ of the collection of brainwave signals from a person ~~'s brainwaves while they are watching a game where the~~ that is watching an agent play a game. The agent knows the game rules but not how to win it. ~~Hence, the agent~~ The second part, the gaming agent learning phase, is where the agent can learn the winning strategy using the person's feedback to improve its own performance. ~~This section describes the process of obtaining the person's brain signals that are used to make the agent more efficient.~~

~~The experimental procedure is summarized in Figure 1. The central part is the~~

A. *Brainwave Session*

The retrieval of the ~~subject~~OHC's brain activity~~. This process is called brainwave session~~, called the brainwave session, is one of the most critical parts of the study. Subjects are recruited voluntarily ~~. They are given a consent~~ and given a form with questions regarding their health (previous health issues and particular visual sensitivity), habits (sleeping hours, caffeine and alcohol consumption), ~~as well as an approval~~ and a consent petition to collect the required data. The brainwave sessions are performed with 8 subjects, 5 males and 3 females, ~~average age 25.125 years, standard deviation~~ with an average age of 25.12 years, a standard deviation of 1.54 years, and a range of 22-28 years. All subjects have normal vision, are right-handed and ~~have~~ no history of neurological disorders.

After the form is filled out, a short description of the procedure is given to each subject. They are only told that the objective of the agent is to reach the goal and the four movements that the agent can make. When this concludes, the subject is introduced to the wireless digital EEG device (g.Nautilus, g.Tec, Austria) that she/he has to wear during the brainwave session. It has eight electrodes (g.LADYbird, g.Tec, Austria) on the positions Fz, Cz, Pz, Oz, P3, P4, PO7, and PO8, identified according to the 10-20 International System, with a reference set to the right ear lobe and ground set as the AFz position. The electrode contact points are adjusted applying conductive gel until the impedance values displayed by the program g.NeedAccess (g.Tec, Austria) are within the desired range. This process takes between 10 to 15 minutes. After this step, the subject is instructed to close their eyes~~and the same program is used~~, make eye movements and muscle chew in order to check the program and guarantee that the live channel values ~~so that there are no dead ones and the expected values are displayed for eye movements or muscle chewing~~are accurate.

Once the headset is correctly applied, the OpenVibe Acquisition Server program, from the OpenVibe platform [?], is launched and configured with a sampling rate of 250 Hz. A 50 Hz notch filter is applied to filter out power line noise. An additional bandpass filter between 0.5 Hz and 60 Hz is applied~~as well~~. Data is handled and processed with the OpenVibe Designer, from the same platform, using 8 channels for the brain data (one channel per electrode) and an additional channel ~~for~~ to record the stimulus, which corresponds to a game movement performed by the agent. After everything is connected, the subject ~~seats~~ is seated in a comfortable chair in front of a computer screen. The brightness of the screen is set to the maximum setting to avoid any visual inconvenience in which the subject can not distinguish the components of the game that appear on the screen. This dataset has been published on the IEEE DataPort initiative [?].

The Acquisition Server ~~has the responsibility of receiving and synchronizing~~ receives and synchronizes the signal

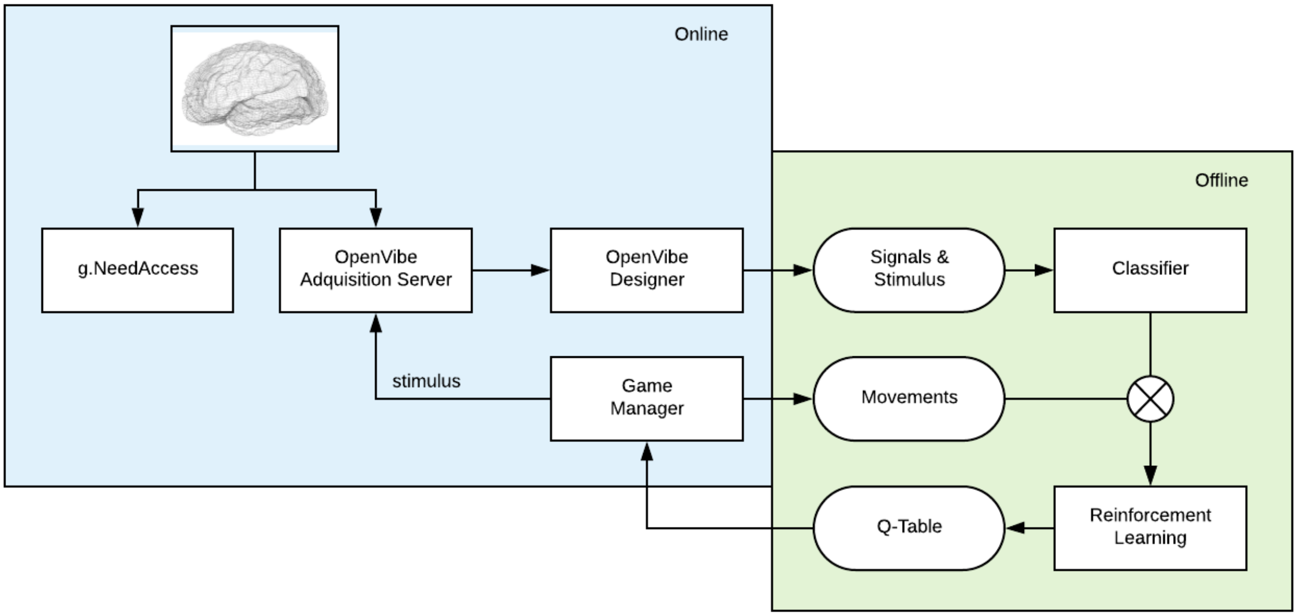

Fig. 1: Overview of the experimental procedure. Brainwaves are obtained by the OpenVibe Acquisition Server. The Game Manager is responsible for generating the game screen, the game mechanics, and the game movements performed by the gaming agent. It is also connected to the Acquisition Server to send stimulus information. The captured information is stored by the OpenVibe Designer. Offline, EEG signals are classified and they are linked to each game movement calculated by the Game Manager to determine proper rewards for each action. This information is used by a Reinforcement Learning algorithm to learn iteratively a Q-Table to improve the performance of the agent that plays the game.

data from the headset and any event information from the game, and transfers it to the OpenVibe Designer application. When the subject is ready, the Game Manager and the OpenVibe Designer programs are launched and configured to communicate with the previously mentioned Acquisition Server. A brainwave session consists of several ~~experiences~~matches, each one being a ~~game run. At~~ gameplay. In the end, the ~~game state information~~ sequence of game movements and the signal data ~~with the game event information of each run~~ generated for each match are saved for offline processing.

B. *Cognitive Game Procedure*

The game parsimoniously consists of a $5x5$ grid of grey circular spots with a black background. A blue spot indicates the current position of the agent ~~whereas~~ and a green spot represents the goal, as shown in Figure 2. The agent's objective is to reach the goal. The circular spot representing the goal remains static at the bottom-right position of the grid, while the one representing the position of the agent ~~always~~ starts at the upper-left position of the grid and moves in each iteration. When the agent reaches the goal, the position where the agent and the goal are located turns red, showing that the ~~experience~~ match has ended. There are four possible movements that the agent can perform: it can go upwards, downwards, towards the left and ~~towards~~ the right, and those movements are bounded to avoid the agent from leaving the grid. The movement direction is selected randomly and is executed once every 2 seconds. ~~At the end~~After each gameplay, there is a pause of 5 seconds ~~before the next experience~~ until the next match starts. Each time an agent moves, the Game Manager program sends an event marker to the Acquisition Server. This is considered ~~as~~ a stimulus to the observational human critic. The ~~experience is designed for it~~ game is designed as to be evident whenever there is an error (i.e. the agent moves away from the objective) so the subject can notice it immediately after the stimulus is presented, possibly triggering the expected cognitive response, which can be imprinted as an ErrP component within the EEG stream.

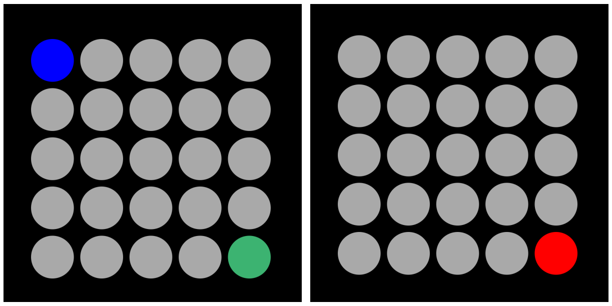

Fig. 2: Grid system representation used in the ~~Cognitive Game Experience~~cognitive game. The blue spot represents the initial location while the green spot represents the target location. Once the agent reaches the target spot, its color turns red to indicate the end of the play.

III. ~~Signal Processing, Segmentation and Classification~~

A. Signal Processing, Segmentation and Classification

To aid ~~in detecting the~~ the detection of the ErrP response, an offline processing pipeline and classifier is constructed to identify whether the action taken by the agent is an error or not, from the human observer's point of view. It is developed in Python using the ~~MNE~~ "MNE" software platform [?], which is a package designed specifically for processing EEG and Magnetoencephalography data, and built upon the machine learning library Scikit-Learn [?].

This pipeline consists of the offline processing of the collected signals ~~in order~~ used to train a classifier that can decide whether an error potential is triggered. Firstly, the output of a brainwave session is read and an additional ~~band pass~~ band-pass filter of 0.1-20.0 Hz is applied to the signal. Samples that correspond to the start of an event are tagged using the data from the stimulus channel.

After the raw data is loaded and tagged, epochs are extracted~~from the raw data~~. Epochs consist of all the sample points that take place ~~between the~~ during the 2 seconds from the start of the event~~and~~, 2 seconds ~~later (time~~ corresponding to the time it takes for each action to take place~~)~~, resulting in 500 samples per channel, as the ~~sample~~ sampling frequency is 250 Hz. Thus, each epoch is composed of a matrix 500 x 8 channels.

Samples that do not correspond to an epoch (located beyond the 2 seconds frame after the onset of the event) are not used. Also, epochs referring to the start or finish of ~~the experience~~ each match are excluded. ~~This is done because the start and the end of the experience doesn't involve the agent taking an action.~~

In this way, the raw data of a brainwave session is processed into an array of ~~experiences~~ matches where each element is an array of epochs tagged with a number specifying the prediction of the classifier, i.e. if the epoch corresponds to an action that made the agent ~~moves~~ move further from the goal (hit) or an action that made the agent moves closer to the goal (no-hit). The ErrP is expected to be found in hits. To get the data ready for classification, the stimulus channel is removed ~~in order~~ to classify the signals using only the EEG data. Each epoch is regularized using a MinMaxScaler, i.e. ~~substracting the minimun~~ subtracting the minimum value in the epoch and dividing by the signal peak-to-peak amplitude [?]. The eight channels are concatenated using the MNE Vectorizer function~~to transform~~, which transforms the data matrix into a single array sample. Lastly, this data is used by the classification module as information to train and test a classifier. ~~Four~~ Five different classification algorithms are used~~.~~: Logistic Regression, ~~Multi-layer~~ Multilayer Perceptron with a hidden layer of 100 neurons (i.e. default values for the Scikit-Learn MLPClassifier), Random Forest, KNeighbours with k=3 and finally a linear kernel Support Vector Classifier (i.e. SVM) [?].

III. ~~Reinforcement Learning~~

A. Reinforcement Learning

Each ~~experience~~ match consists of a list of game movement configurations and the associated epochs obtained from ~~subject~~OHC's brainwaves. The set of ~~experiences of each subject~~ matches of each OHC is split into training and testing. Training ~~experiences~~ matches are used to train the classifier ~~, whereas test experiences are used to test its performance~~to identify the ErrP signal. After a classifier is trained, the epochs ~~of the experience~~ extracted from the test matches are classified as hit or no-hit. A reward for each movement in the ~~game run is produced,~~ match is generated based on the ~~classification of the epoch that correspond to that~~ prediction from the classifier for that movement. The reward can either be -1 when the event is classified as a hit or 0 when it is classified as a no-hit. The accuracy of ~~this~~ these rewards depends on the performance of the classifier. The list of game movements and their associated reward information ~~are~~ is used to train the agent by a variant of Reinforcement Learning called Q-Learning algorithm.

B. Q-Learning

Q-Learning [?] is a form of model-free reinforcement learning~~algorithm~~, where an agent tries an action at a particular state and ~~evaluate~~ evaluates its consequences in terms of the reward or penalty it receives. ~~In order to~~ To represent rewards, a matrix $Q(s, a)$ is used, where rows correspond to all the possible states, ~~whereas~~ and columns represent all possible actions. This matrix is known as ~~a~~ the Q-Table. The algorithm proceeds by randomly choosing what action to do and ~~updating iteratively~~ iteratively updating the Q-Table based on the received reward r by the following equation

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma * \max_{\tilde{a}} Q(\tilde{s}, \tilde{a}) - Q(s, a)] \quad (1)$$

where s is the current state, a ~~is~~ the action, α ~~represents~~ the learning rate and γ ~~represents~~ the discount factor, a value between 0 and 1 that determines the importance of long term results versus immediate rewards. Hence, $Q(s, a)$ is the expected value of the sum of discounted rewards that the agent will receive if in the s state, it takes the action a according to this policy. Once the environment has been extensively explored and the Q-Table has been ~~obtained~~optimized, the action chosen for a given state is the one that maximizes the expected reward according to the Q-Table matrix.

The algorithm is developed in Python and uses the OpenAI Gym toolkit [?]. Gym is a toolkit for developing and comparing reinforcement learning algorithms. It makes no assumptions about the structure of an agent, and is compatible with any numerical computation library, such as TensorFlow or Theano [?].

C. Gaming Agent Learning Procedure

~~The Q-Table is initialized with zeros, unless a preexisting~~ The gaming agent learning procedure uses the testing matches from brainwave sessions produced during the cognitive game procedure phase, and their components are schematized in Figure 1.

This phase is divided into a sequence of run sessions and a gaming agent training match. During the run session, the agent plays 200 matches guided by a specific Q-Table with a 5% chance of randomly selecting a movement, to reduce deadlocks and loops. Following the run session, the agent performs a single gaming agent training match. The gaming agent starts first with a Q-Table ~~is passed as a parameter~~initialized with zeros, so the initial policy for the agent is randomized. For the agent ~~, in order~~ to learn from the feedback generated by the ~~subject, the Q-Table is not used to determine the action to take at a given state. Instead, the policy which determines which action to take in a particular state is given by the~~ OHC, movement actions are determined by the reply of the agent's actions ~~taken from the brainwave session results~~that were taken during one brainwave session match, in an offline reinforcement learning scheme [?]. This allows ~~to learn~~ the Q-Table to be built based on the ~~subject~~OHC's feedback from the movements the agent took, which ~~are chosen~~ were executed pseudo-randomly ~~, while executing~~ during the brainwave session. The previously mentioned feedback is not explicit as it comes from the interpreted brain signal data. This implies that the reward is determined by the ~~subject~~OHC's brain activity.

Hence, following the iterative procedure based on Equation 1, the Q-Table is updated in each ~~iteration~~gaming agent training match. After the algorithm finishes ~~iterating through all the training episodes~~replicating all the steps from the brainwave session match, the Q-Table is stored ~~to test the performance of the agent~~ and used by the agent in the next run session.

III. Results

~~Figures 3 and ?? show~~ Figure 3 shows the binary classification accuracy obtained for the eight ~~subjects for the four~~ OHCs using five different classification algorithms and using a 10-fold ~~cross validation~~ cross-validation procedure. The best overall performance is obtained using Logistic Regression. In addition, the classification accuracy obtained averaging 5 epochs is shown as well. Although the signal averaging procedure improves the Signal-To-Noise-Radio (SNR) of the ErrP response, it reduces the number of data samples, producing a clear improvement only for the OHCs that contain more samples (1, 3 and 7).

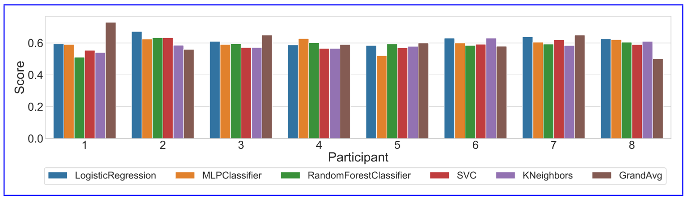

Fig. 3: Binary ~~Classification Score~~ single trial classification score using five different classifiers while recognizing ErrP potentials for the eight ~~subjects~~OHCs. The classification score using an ensemble average of 5 epochs is shown as well (GrandAvg). Chance Level is 0.5.

~~Binary Classification Score using Logistic Regression for the eight subjects.~~

On the other hand, Figure 4 shows ~~for each subject~~ the average amount of steps ~~the agent takes~~ it takes for the agent to reach the goal for each OHC, as the Q-Table is progressively trained using the reward information obtained from ~~their classified experiences~~the prediction of the trained classifier. Each point corresponds to a run session where the average number of steps ~~in 200 repetitions that it takes~~ for the agent to reach the goal ~~for~~ using a specific Q-Table is specified, for 200 repetitions. The first point~~represents the amount~~ , at the x-value 0, represents the number of steps the agent takes to reach the goal ~~for a~~ with an untrained Q-Table~~that hasn't been trained at all~~, where movements are decided randomly. The next point corresponds to the amount of steps it takes to reach the goal using a policy derived from a Q-Table trained ~~with one experience~~after one brainwave session match, and so on.

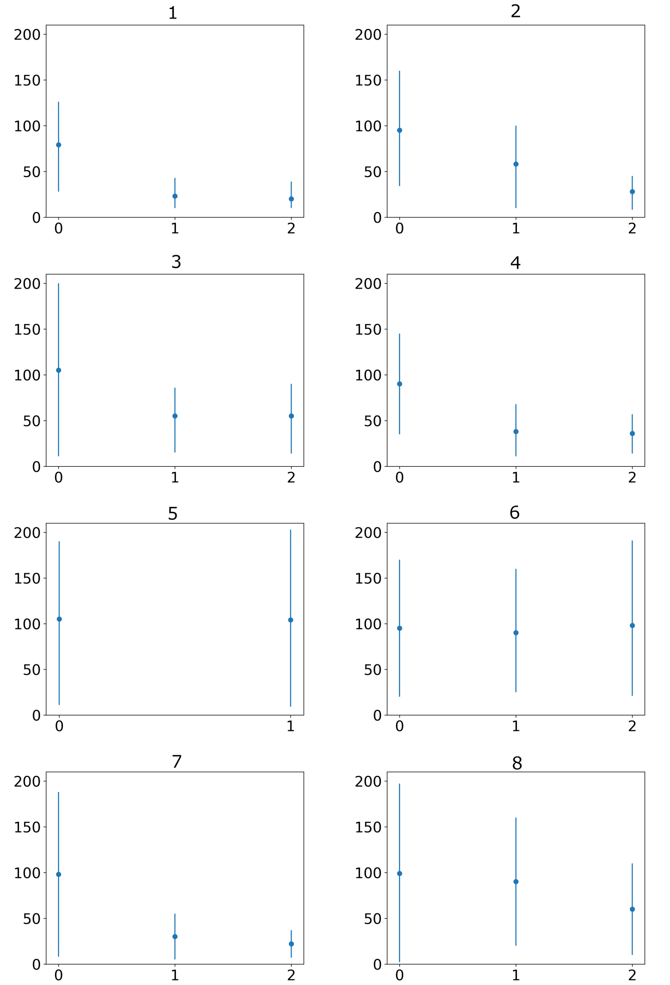

Fig. 4: Average number of steps for the agent to reach the goal when trained with ~~experiences~~ rewards generated from ~~subjects 1(A)-8(H)~~brainwaves from OHCs 1-8. Y axis show the averaged number of steps for a run session, while x axis show the number of ~~experiences~~ game matches used to cumulative train the Q-Table.

The results show that as the Q-Table is progressively trained the average amount of steps decreases, meaning that the agent learns. However, the rate at which it learns varies per ~~subject~~OHC, depending on the classification accuracy of the extracted brainwaves. For example results for ~~subject~~ OHC 1 ~~(Figure 4 A)~~ show faster learning than those of ~~subject~~ OHC 8 (Figure 4~~H~~).

In the case for ~~subject~~ OHC 5 and 6, the reward information obtained from the brainwaves is not enough to train the agent effectively. Figures 4 ~~E and 4 F~~ for OHC 5 and 6 show no apparent learning, as the amount of steps to reach the goal doesn't decrease when trained. These results are also consistent with their classification ROC curves, shown on Figures 5 obtained for both ~~subjects~~OHCs, where the area under the curve are close to chance level. Both ~~subjects~~ OHCs have less recorded data from the sessions in comparison to the rest of the ~~subjects~~OHCs. This variation in performance for different ~~subjects~~ OHCs has been studied extensively in BCI [?]. Besides low data samples, there are other reasons affecting the classification accuracy: cognitive reasons ~~, this being the subject~~ (i.e. the OHC not paying extensively attention to the game dynamics), very low ~~Signal-To-Noise Ratio (SNR)~~ SNR of the ErrP component or even the BCI-illiteracy phenomena where the specific ~~subject~~OHC's signals do not contain the expected component response [?].

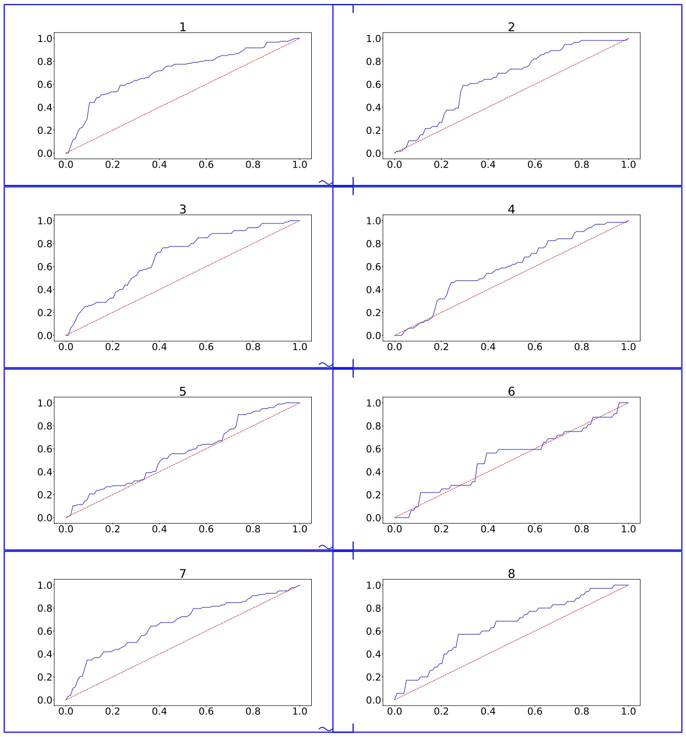

Fig. 5: ROC Curves for ~~subjects 5 (up)~~ OHCs 1-8. True positive rate is on the vertical axis and ~~6 (down)~~false positive rate on the horizontal axis.

Figure 6 shows the result of an agent successively trained with ~~experiences obtained from a brainwave session , generated with sham~~ brainwave session matches where the EEG is generated with random signals. In this case, random EEG signals ~~were~~ are generated using OpenVibe Acquisition Server signal generator for all channels, as if they were ~~generated from a human observer~~ produced by an OHC who doesn't pay attention to the game. ~~As it can be seen, the~~ The agent learns nothing, and regardless of the ~~amount of experiences~~ number of matches that are used to learn the Q-Table, the number of steps required to reach the goal does not decrease. This pattern is also obtained when the ~~experiences from subjects~~ game matches from OHCs 5 and 6 are used, showing that the reward labeling predicted by the trained classifier for those ~~subjects~~ cases worked like a random classifier.

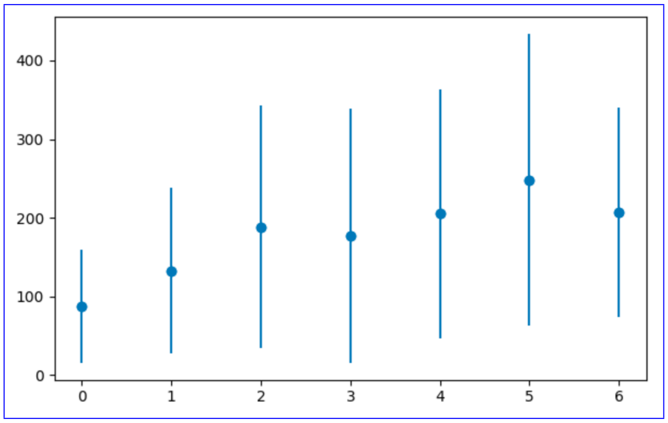

Fig. 6: Average number of steps ~~using Q-Table~~ for the agent to reach the goal when trained with a ~~noisy signal~~classifier produced from sham EEG signals. X axis show the number of gaming agent training matches used to train the Q-Table.

Electroencephalographic signals have high inter-subject variability [?]. This is evidenced in Figure 7 where the agent training is performed ~~by using~~ with rewards obtained by classifying epochs from one ~~subject, but using a classifier which~~ Tester OHC and a classifier that was trained using the brainwaves from a different ~~subject. Not~~ Trainer OHC. The figure shows the cumulative variation for all run sessions on the average number of steps required to reach the goal after training the agent with all the available matches from the brainwave session. Enhancements are shown as negative values. Only the diagonal of the heatmap matrix shows a clear improvement in terms of the reduction of the required number of steps to reach the goal (averaged per 200 runs) which corresponds to the same information for each OHC shown in Figure 4. For the transfer learning experiment [?] , no performance gain is evidenced, the agents learn nothing ~~which~~ and this implies that the reward information is useless.

Finally, Figure 8 shows the result of training an agent with ~~accumulative experiences from subjects~~ cumulative brainwave session matches from OHCs 1, 2, 3, 4, 7, and 8. It can be seen that the overall performance of the agent improves as long as there ~~are more experiences to be used to train it, regardless if~~ is information to produce rewards, regardless of the fact that they were generated from ~~the brainwaves classification from different subjects~~classifiers trained with different OHC's signals.

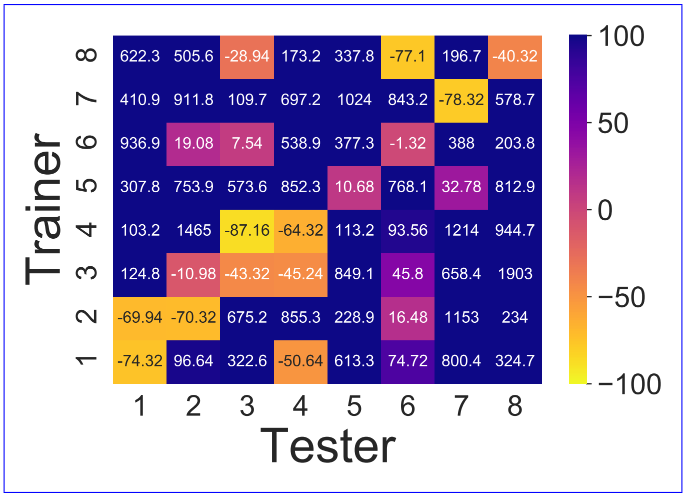

Fig. 7: ~~(A) Average~~ Heatmap for the transfer learning experiment. Values represent the reduction in the average number of steps ~~using Q-table trained with experiences from subject 8 classified with a classifier trained with data from subject 6. (B) Average steps using Q-table trained with experiences from subject 1 classified with a classifier trained with data from subject 3.~~required to reach the goal. Negative values represent net improvements.

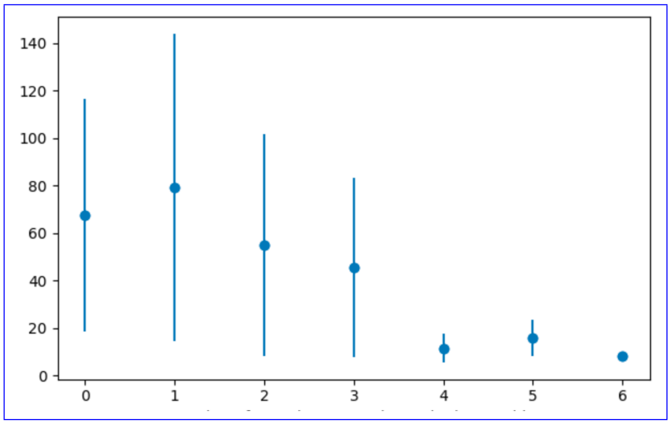

Fig. 8: Average steps using Q-Table trained with ~~6 experiences~~ brainwave session matches from six different ~~subjects~~OHCs. X axis show the progressive number of gaming agent training matches used to train the Q-Table.

IV. Conclusion

This work aims to ~~state whether ErrP signals could be used~~ propose a simple game that can use the ErrP component to train a gaming agent using ~~reinforcement learning~~a RL model. The collected data ~~show~~ shows that ErrP signals can in fact be classified and used to train an agent effectively.

This proposal tries to keep the system as simple as possible, emphasizing information flow from the subjective error perception of the human critic~~, through the reward generation~~. Rewards are generated using the signal processing and classification pipeline, and ~~finally~~ the Q-Table ~~updating to enhance~~ updates, enhancing the performance of the gaming agent.

~~While classifying, the better performing classifier is Logistic Regression. One~~ One additional aspect to remark is the robustness of the learning strategy based on Q-Learning [?], [?]. The obtained accuracy to discriminate ErrPs is low. However, even with such low accuracy values, the RL algorithm was able to extract meaningful information from rewards that were helpful to improve, and often maximize, the agent's performance. Additionally, one important aspect of the classification results is the low percentage of false positives (Figure 9), showing a high specificity. ~~It is not common that the agent learns that an action is wrong when in fact it is an action that takes it closer to the goal.~~ On the other hand, the percentage of false negatives is generally higher. However, even though this implies that the agent misses frequently ~~that an action taken is wrong~~ when a wrong action takes place, this is not hindering the overall performance and the agent is still learning.

Though scarce, accurate rewards are very useful for the RL algorithm.

~~Once ErrP signals are identified they can be used to train an agent using a reinforcement learning algorithm. Brainwave sessions have a low amount of experiences in order to reduce fatigue from subjects. However data suggests that longer sessions are required in order to reach better classification scores, since more data is available in order to train the classifier. It can be seen that subjects with the largest amounts of data have the best classification. This can also be achieved designing a bigger game system that generates more samples with every session.~~

At the same time, effective agent training depends on the ~~subject's data that was used to train it. Results show that~~ OHC's training data. Results confirm the futility or complexity of using Transfer Learning [?]: training a classifier with data ~~of one subject, but using it to classify the events of experiences of another subject does not lead to an improvement on~~ obtained from one OHC, and using the same classifier to identify ErrPs for another OHC does not increases the performance of the agent. Despite that, the rewards generated from different ~~subjects~~ subject's classifiers can be used to train the same Q-Table to improve its performance, which may lead to strategies where the overall performance is ~~improved~~ enhanced based on the information from different human critics at the same time.

~~One additional aspect to remark is the robustness of the learning strategy based on Q-Learning [?], [?]. The obtained accuracy to discriminate ErrPs is low. However, even with such a lower accuracy values, the RL algorithm was able to extract meaningful information from rewards that were helpful to improve the agent performance~~There seems to be an agreement in terms of the subjective interpretation of what may be an appropriate movement to reach the goal.

The simple setup of the grid-based game allows further experimentation, using the reduction on the number of average steps to reach the goal as a validation of the achieved information transfer. It will be of research interest to verify if the smooth progression towards the end alters the shape of the ErrP response, how the ErrP response is triggered in relation with different shapes and colors of the board markers [?], or if there is a differential ErrP signal component in relation to

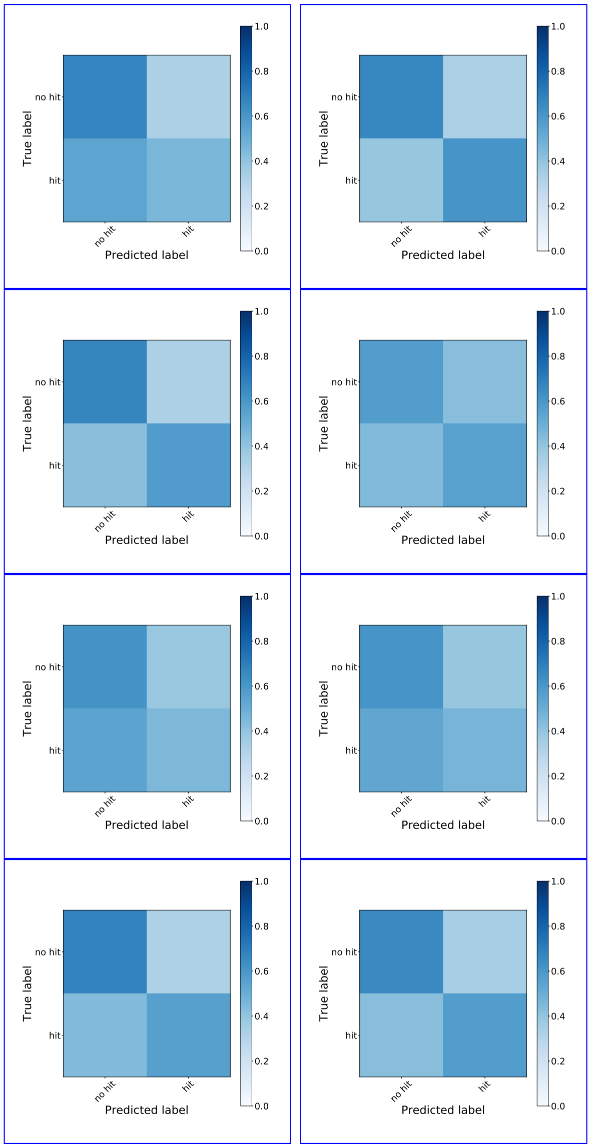

Fig. 9: Confusion Matrix for ~~subjects 1,3,4 and 8.~~ OHCs 1-8. Darker colors show higher values. It can be seen the lower percentage of false positives (upper right corner of each chart).

up, down, left and right movements. In addition, the outcome of manipulating the stimulus could be further studied as well as the influence on the results if incentives are given to participants.

Further work will be conducted in order to increase the complexity of the game to allow the possibility that the target position ~~be~~ is dynamically changed. ~~In that case the agent would start to learn to follow the target, instead of learning to go to a specific static destination spot. Additionally, the~~ Although we found that the best performing classifier is Logistic Regression, there is room for improvement. The classifier could be enhanced to recognize ~~more effectively~~ the Error Potential [?] ~~.~~ more effectively or could be pre-trained to allow higher accuracy [?].

~~Concluding, this research show evidence that brain signals can be used as an interface between human and a gaming computer enabling an alternative communication with the system without explicit input from the user.~~

Acknowledgment

The authors would like to thank the Laboratory Centro de Inteligencia Computacional and to ITBA University.

Funding

This work was supported by the grant ITBACyT-15 issued by ITBA University.