Applications of Machine Learning to Virtual Reality

Jiri Kapralek and Nezamoddin N. Kachouie

Florida Institute of Technology
150 W. University Blvd.
Melbourne, Florida

jkapralek2013@my.fit.edu, nezamoddin@fit.edu

ABSTRACT

Delivering a high quality laboratory experience in the physical sciences outside of a real lab facility is difficult. Immersive virtual reality can provide such an experience. This will provide students in non-STEM majors a deeper understanding of and appreciation for the physical sciences than they would receive without performing any experiment. Moreover, realistic, immersive virtual reality experiments can be considered as a substitute for some real world lab experiments to provide long-term cost savings. The ultimate goal of this project is to create a virtual reality classroom or laboratory in which student-student, instructor-student, and student-instructor interactions are maintained. To achieve this, we develop a virtual lab that will observe how students practice the lab experiments in order to help them avoiding the previous mistakes and in turn improve their performance.

In this paper we develop a simple virtual reality experiment (Figure 1.) as proof of concept.

Figure 1. Virtual Reality Environment

Keywords

Virtual Reality, Machine Learning, Unity, Artificial Neural Networks.

1. INTRODUCTION

In this experiment, students are provided with eggs of different ages. The goal of this experiment is to determine the egg's age by submerging it in the water. To determine the density of an egg, students measure the egg's weight and the volume of displaced water when the egg is submerged in the water. Students will then calculate the density using a simple equation. We will run several simulations of this virtual experiment and data will be collected during each simulation. Collected data include questions asked by students, students' mistakes, unsafe approaches, etc. The collected data during these experiments will be used by a machine-learning algorithm to modify the simulation to better address the students' questions, help them to avoid unsafe approaches, etc. This experiment is proof of concept to demonstrate the value of virtual reality technologies in conjunction with machine learning algorithms in physical education.

This experiment suggests that we can take full advantage of the affordances offered by immersive VR, including the use of more innovative virtual experiences rather than normal lab experiments. Our future goal is to practice and extend the applications of VR and machine learning to other physical educations.

Preparing training datasets for a machine-learning algorithm is often time consuming or cost prohibitive. Therefore, a major advantage of merging a virtual reality engine such as Unity with a machine-learning algorithm is producing virtually unlimited training datasets. When using the physics engine in Unity, we can determine whether the simulation is a success or not. The physics engine's output will feed the machine learning algorithm to modify the simulation. This will be repeated until the desired outcome.

2. Machine Learning Approach: Artificial Neural Network

A machine learning algorithm is developed to train a model in virtual reality environment. Such trained model can be used individually or in conjunction with an electromechanical assembly for training purposes to perform a desired task. To train the model, some measurable criteria for success must be defined.

For the proposed task, i.e. a laboratory experiment, we have some predefined features to evaluate the experiments' success. There are several criteria to evaluate experiment's success such as how the procedure is performed by the student, whether the experiment's instructions have been followed, whether the student has safely performed all steps, and the accuracy of the outcomes. The algorithm will collect data for training purposes regarding successful trials, different approaches that are taken by students to perform the experiment, and potential patterns in different approaches. The algorithm will identify and use the correct procedures for activations of the neurons of an Artificial Neural Network (ANN). For example, when a student grabs a ruler, some neurons will be activated in a neural network layer based on how the student will use the ruler such as whether the ruler is in close proximity to the water surface, whether the ruler is perpendicular to the water surface, and whether the ruler is touching the surface of the table in order to measure the water displacement. In the second phase, after the algorithm is trained how to perform a successful experiment, new data sets will be used to evaluate the model. In the validation (test) phase, incomplete data sets are provided to the model. The goal of validation is whether the model can correctly predict the outcome of an experiment. It means whether the experiment is a success or failure. More specifically, the model monitors the student's performance and predicts whether the student will have correct outcomes (within some acceptable range). To determine how accurate, the prediction of egg's age is, we have defined a cost function that will be evaluated for each egg separately. With enough data collected through the backpropagation network, the algorithm will be optimized by minimizing the cost function. The third and last step in the machine learning algorithm training is to determine where exactly did the student deviated from a successful path in his/her approach and recommend one or more possible solutions in order to produce a successful result. For instance, in the collected data, there will be an indicator to show whether the student took the ruler, whether he/she touched the surface of the table while measuring, and how close the ruler was to a perpendicular position to the water surface. In this way, we can evaluate how accurate the measurements and in turn associated results (calculated density) will be. The algorithm can detect the wrong procedures and in turn the software would be able to generate a warning message and send to the student in real time. As a result, a customized model for each student can be developed to provide feedback to the students in real time. Depending on the experiment and its educational goals, the student might be notified either shortly after the mistake or after a lag-time which will be set to give the student a chance to correct his/her mistake. Once an adequate amount of data is obtained, an optimal time slot can be determined to provide the student with the feedback in order to minimize the interruption in the learning process. Although, the focus of this work is mainly to improve the success and productivity of education, it has a broad range of applications in the real world.

In order to activate a neuron, the weighted sum of inputs is:

$$weighted\ sum = w_1 a_1 + w_2 a_2 + \cdots + w_n a_n \tag{1}$$

where a_i is input i, w_i is the weight of input i, and n is the number of inputs. In the context of our experiment, the first layer of the neural network will receive multiple inputs associate with student's performance such as whether student picks up the ruler and how accurately he/she will measure the water volume displacement. Hence, a_i's & $w_i{}'s$ are input signals to the neuron layer and their associated weights respectively. This layer of neurons will assess the accuracy of water displacement measurement. For example, the angle of the ruler with perpendicular line to the water surface is an input associated with the accuracy of the measurement the water volume displacement. It means, if the angle to the water's surface is close to the perpendicular line, the weighted sum will obtain a large value and in turn will result in a high activation value for the neuron.

A sigmoid function

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{2}$$

is used in order to control the range of the weighted sum of input signals between 0 and 1. The sigmoid function will assign zero or small positive values to the negative values of the weighted sum. It assigns values close to one for large positive values of the weighted sum. The values monotonically increase between zero and one.

To activate the neuron based on the Hebbian theory, a bias is introduced in the weighted sum (Fig. 2):

$$a_o^{(1)} = \sigma(w_{0,0}\, a_o^{(0)} + w_{0,1} a_1^{(0)} + w_{0,2} a_2^{(0)} + w_{0,n} a_n^{(0)} + \ b_0) \tag{3}$$

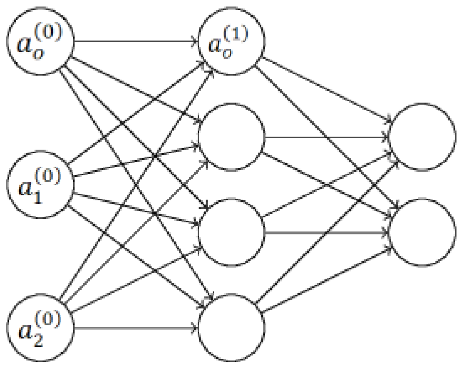

Figure 2. A typical neural networks with three layers

where b_0 is the bias. Bias determines how positive the relative weighted sum must be in order to activate the neuron. Therefore, the bias function acts as a threshold for the activation of the neuron. The neuron will not be activated until its activation weighted sum value reaches predefined positive value $|b_n|$ in Eq. 3. Eq. 3 can be written as

$$a^{(1)} = \sigma\left(\begin{bmatrix} w_{0,0} & \cdots & w_{0,n} \\ \vdots & \ddots & \vdots \\ w_{k,0} & \cdots & w_{k,n} \end{bmatrix} \begin{bmatrix} a_0^{(0)} \\ \vdots \\ a_n^{(0)} \end{bmatrix} + \begin{bmatrix} b_0 \\ \vdots \\ b_n \end{bmatrix}\right) \tag{4}$$

It can be simplified using vector notation to

$$a^{(1)} = \sigma(W a^{(0)} + b) \tag{5}$$

This simplified expression makes the computations faster by using matrix multiplication. An error term is calculated as the squared difference between desired value and the activation value. This error term will be evaluated for a back-propagation network in each iteration.

A cost function C is used to evaluate the accuracy of the machine-learning model. To maintain a time efficient method, in place of global minimum, a local minimum of the gradient of the cost function ∇C is identified. In each iteration, the error term will be computed and the average error for the whole training set (in virtual environment) will be calculated $(-\eta\nabla C(\vec{W}))$. The initial values are randomly generated, and in turn multiple local minima are identified. An absolute minimum among all minima will be determined. Since the optimization is performed in virtual reality, the input signals to neurons are coming from the physics engine. Hence, practically there is not any restriction regarding the number of iterations.

Our future work will be conducted to improve the machine learning algorithm and customize it further for the specific requirements of virtual reality environment. The eventual goal is to develop a sophisticated model, train it in a virtual reality environment, and use it for a real world application.

3. Design Implementation

In order to train the ANN algorithm, for each parameter we generate independent random values within a specific range to simulate student's procedure. In the training phase, the ANN will be provided with the outcome of experiment based on the selected random set of values. It means whether or not the selected random set will produce correct results and if so, how accurate are the results. In this way, the output from the ANN will be compared with the ground truth values either using predefined values or provided values by the virtual reality environment. Through backpropagation, the cost function is going to be minimized and after convergence, ANN's activation values will be used in the test phase.

The trained ANN will receive input parameters from the virtual reality physics engine, (Fig. 3) such as coordinates, vectors, forces, distance, and Boolean operators.

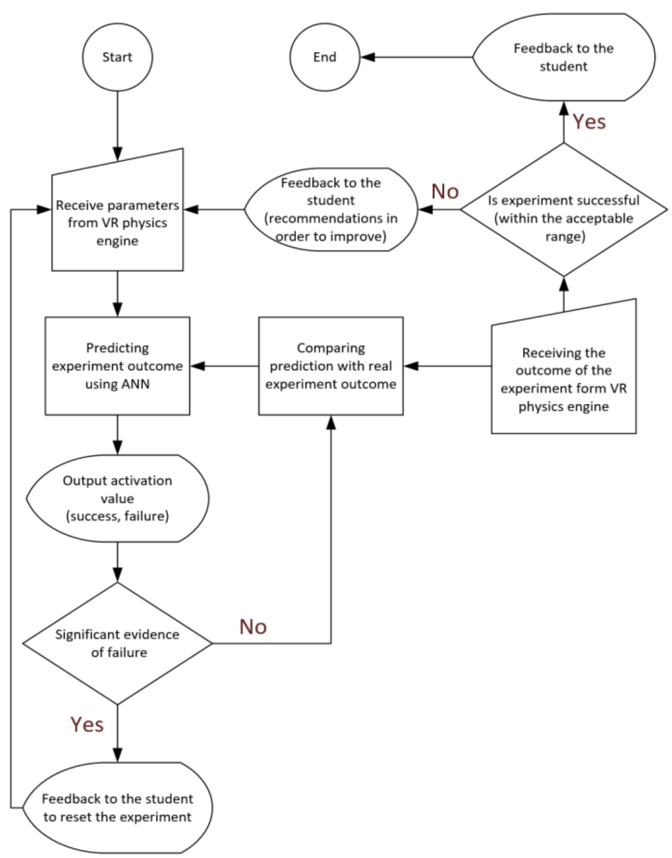

Figure 3. Implementation flow chart

The trained ANN will process the data generated by student's procedure to predict the outcome of the experiment and indicate whether or not the experiment is going to be successful. If there is strong evidence that the experiment is going to fail, the student will be notified by providing feedback in order to correct his/her mistakes. Once the experiment is complete, the accuracy of the prediction is going to be determined by comparing the predicted value (output activation value) with the actual value provided by the virtual reality physics engine. If the experiment outcome falls outside the acceptable range, the student will receive feedback to repeat the experiment along with recommended course of actions to improve his/her performance. If the experiment outcome falls within the acceptable range, the student will be provided with feedback regarding the accuracy of the results and potential ways to improve them.

4. RESULTS

In order to validate the proposed method, the ANN was trained using multiple sets of randomly generated vectors and the success rate was recorded. The validation is based on independently generated random vectors, therefore the validation and training are independent. Figure 4 shows proportion of successful trials. As we can see in the figure, the success rate depends on the tolerance considered to accept the procedure as a success. By increasing the acceptable range for the parameter the success rate increases accordingly. In this way we are able to adaptively set the acceptable accuracy based on the experiment at hand.

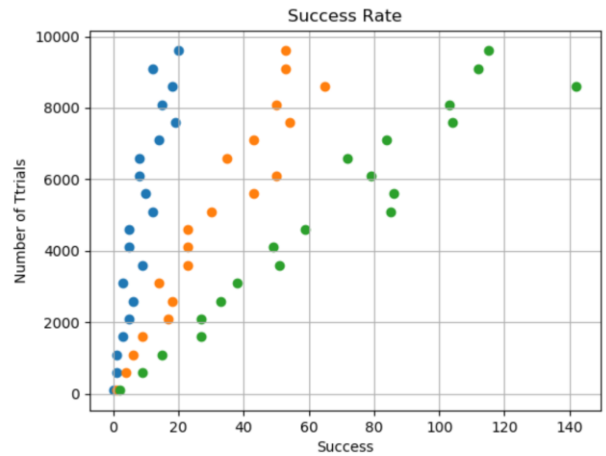

Figure 4. Success Rate of ANN

Future work will be conducted to further explore the impact of having multiple parameters on the training time of the ANN.

5. REFERENCES

[1] Johannes Fürnkranz, Tobias Scheffer, Myra Spiliopouou, "Learning Stochastic Tree Edit Distance" in *Machine Learning: ECML 2006*. Berlin, Germany, September 18-22, 2006

[2] Johannes Fürnkranz, Tobias Scheffer, Myra Spiliopouou, "A Selective Sampling Strategy for Label" in *Machine Learning: ECML 2006*. Berlin, Germany, September 18-22, 2006