

A computational model for physics learning

Florin Bocaneala and Lei Bao
Physics Department, Ohio State University
174 W 18th Avenue, Columbus, OH 43210

Abstract

In this paper we test the versatility of the neural network approach to modeling the dynamics of student learning. We choose a problem from an introductory physics class and we construct a neural network model for it. Based on this simulation, we argue for the future use of neural network models in Physics Education Research and what this approach can teach us about how physics learning takes place.

1 Introduction

Due to the nature of the model we are advocating in this paper, we look at the learning process from an unorthodox perspective for Physics Education Research. Whether some of these definitions are satisfactory does not concern us here. In the beginning, we just need to make sure that our assumptions are clear in order that a dialog may take place.

In this paper, we consider that the ability to perform correctly to all possible questions related to a concept or a situation is the final goal of the learning task associated to that concept or situation. Taking as an example the problem modeled here, we think for the purpose of this paper that a student has mastered the situation represented by the problem when the student is able to answer correctly to all possible combinations of external charges and ground connections.

The model, which we propose here, might not make any prediction about one particular student, but we believe that it may be useful in understanding the behavior of large students populations. Therefore, when we talk about student learning, we envision the learning of a statistical entity rather than a particular student.

2 The problem and the model

We have chosen an Electricity and Magnetism problem given to students in algebra and calculus based classes during the Spring Quarter 2003 (this includes both students majoring in Engineering and non-technical fields). We reproduce here the text of the problem from [1].

Q-22 A negative charge is brought and kept fixed in location close to a neutral conducting rod. The end closer to the charge is then connected to the ground by a conducting wire as shown in figure 1. What is the charge on the conducting rod

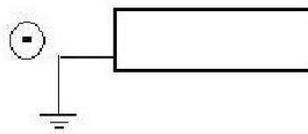


Figure 1: **Q-22**

after the ground connection is removed?

1. Positive charge.
2. Negative charge.
3. No charge (Neutral).

This question (1) was preceded by another version of it, which had the ground connection and the external charge on opposite sides of the conducting rod. Having answered the two questions, the participating students were asked to provide argumentations for their answers. Based on these argumentations, their answers have been categorized in 4 classes [1].

We have built a neural network model for this problem, using feed-forward networks (see figure 2) with 2 and 3 layers. The network receives the input on the first layer and produces an output on the last layer. During training, the output of the network is compared against the expected output and an error is computed. This error is back-propagated from the output to the input layer adjusting the link weights according to their contribution to the error. The elegance of the algorithm

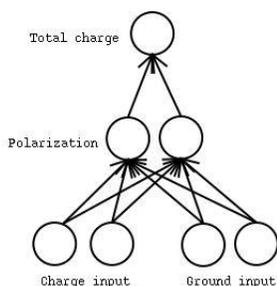


Figure 2: Feed-forward neural network.

and previous successes [2] have motivated choosing back-propagation over other neurologically more plausible algorithms [3].

3 Simulations and results

In order to train or test a network, one needs samples (i.e. a tuple of input and output patterns) grouped together into environments (sets of samples). Considering the way our problem is presented in major textbooks [4], we have constructed three main environments: a polarization environment (including samples for a ungrounded conducting rod near an electric charge), a textbook environment (including samples for grounded conducting rod near an electric charge such that the ground connection and the external charge are on opposite sides of the conducting rod) and a general environment (including all the possible samples).

A complete loop through all the samples contained within the training environment is called an epoch. Each other 30 epochs, we checked how far from the correct answers its answers were. Whenever the sum of errors during an epoch became less than a critical error, the network fulfilled its internal learning criterion and the simulation halted.

3.1 Cognitive performance

Because the standard curriculum teaches the students about polarization before potential, we have used the polarization environment to train the network made up by the first two layers. Then we started training the entire network using the textbook environment and testing the network during this training using the general environment.

Figure 3 shows the sum of errors as a function of epoch number during this simulation. You may

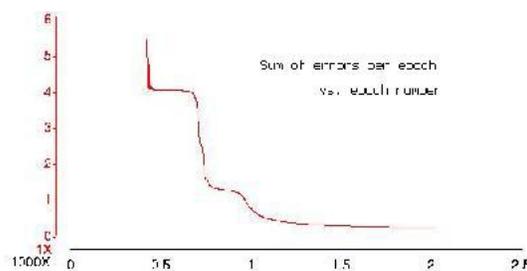


Figure 3: The sum of errors vs. epoch number.

notice three plateaus on the graph where the sum of errors function remains the same for some number of epochs: between 400 and 650 epochs, between 700 and 900 epochs and after 1300 epochs. Since the training algorithm looks for minima of the sum of errors function in the space of the connection weights, we recognize these plateaus as local minima. Because the function moves abruptly between these plateaus, when we test the network during training, we find it in one of the states corresponding to one of these plateaus.

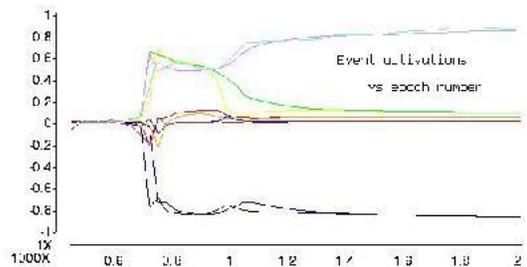


Figure 4: Activation patterns vs. epoch number.

In order to see what these plateaus correspond to, we have plotted the activation pattern for each of the samples belonging to the textbook environment during the same training (figure 4). As you can see, during the first plateau of the sum of errors function, the network's response to any of the samples is 0, which corresponds to a neutral conducting rod. Let's compare this answer to a student's answer [1] (34% of the student answers fell in this category):

Q-22 3. the negative charge may cause charges within the rod to realign, but the overall charge on the rod remains neutral.

During the second plateau the network answered correctly to all the samples from the textbook environment except samples corresponding to a rod which was not grounded. Specifically, when we asked the network what the net charge was on a rod **without** a ground connection near a positive (negative) charge, the network answered “negative” (“positive”). We include here a student’s view [1] (14% have given similar answers):

Q-21 1. Positive charge because without the ground the conducting rod becomes negative but with the ground it becomes positive.

The third plateau, corroborating figure 3 and figure 4, corresponds to the final stage in the training when the network has learned all the samples contained in the textbook environment. But what happens when we test the network on samples not included in the training environment? Well, the network gives incorrect answers. We just remind here that neural networks are very robust in dealing with noisy data and as such they are quite good to generalize beyond their training [5]. However, in our case it failed.

We think that it is interesting how it failed: when we asked for the net charge of a conducting rod near a positive (negative) charge placed on **the same side** of the rod as the ground connection, the network answered “positive” (“negative”). Here is a student’s answer to the same question [1] (29% of the tested students have answered similarly):

Q-22 2. Negative charge. The negative fixed charge that is closer to the end that is connected to the ground attracts the protons in the rod and pulls them toward it and into the ground leaving the rod with an induced negative charge when the ground is removed.

3.2 Previous experience and training environment

It is often suspected that certain student difficulties are the results of particular ways of instruction. Let’s see what happens if we don’t train the network with the polarization environment before

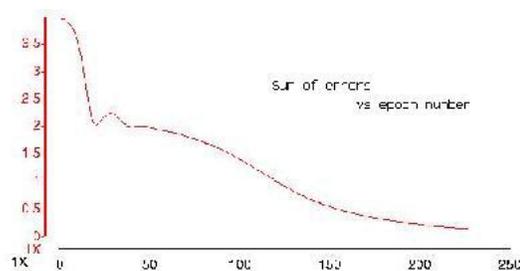


Figure 5: Sum of errors vs. epoch number : tabula rasa.

running the simulation. The figure 5 shows the sum of errors if we start training the network from a *tabula rasa* (i.e. random configuration of link weights). The fact that the first plateau has disappeared proves a strong relation between cognitive behavior and pre-training experience.

As seen above, when we tested the network using the general environment after we had trained it using the textbook environment the network answered incorrectly to the questions not included in the textbook environment. However, when we trained the network using the general environment, which includes even the samples on which the previous simulations had failed, the network answers correctly to all the questions.

4 Summary and conclusions

Let’s review what we have given to the network:

- The network architecture: the number and types of layers, units and connections between units.
- How to represent the questions and interpret the answers.
- We have provided the network with feedback regarding how far its answers were from the correct answers.
- We did not code rules into the network by which the network would associate the input directly with the output: it was its task to find these “rules.”

After training the network was able to answer correctly to our questions, but during training it

committed the same mistakes students make. This point needs further explanations. The model's performance has been checked throughout training against all the 15 questions. Because each question had 3 possible answers, the number of patterns the network could produce was 3^{15} . However, both the student population and the network seemed to select a limited number of patterns (in our case, 4 categories of argumentation). Moreover, including the correct answers, that accounts for $83 \pm 4\%$ of the tested student population ([1]). This feature of neural network models has been noted previously in other settings ([2], [6]) but, since we find it particularly interesting for PER community, we emphasise it here too.

The neural network model has predictive power. For example, we had not known all the students' answers when we designed and ran the simulations. For this reason, we couldn't explain the first plateau (neutral rod). Later, having all the answers the explanation became obvious.

One may use these models in lecture, laboratory and testing design. In general, when it comes to structuring the presented material, a neural network simulation might help. Our network failed to generalize in a case we didn't have any reason to believe it would. Hence, the network model may provide a reliability check for our assumptions of similarity. Due to time and resource constraints, we find ourselves often in the position of selecting the material to be presented to the students based on our internal criteria of similarity. Alas, it turns frequently out that our choices were not the best even though they logically seemed so when we made them. Therefore, we find this feature of the model helpful for some education decisions.

The process of training (teaching, in the real world) can significantly affect how the system (students) develops its knowledge. To this end, previously acquired knowledge about polarization has influenced the system's cognitive behavior. Moreover, our decision to leave out certain samples from the textbook environment has biased the training and the answers of the network. We believe this to be a new feature of the model that has direct applications in Physics Education Research.

References

- [1] Warnakulasooriya, R. "Students' Models In Some Topics Of Electricity & Magnetism" PhD. Thesis 2003
- [2] McClelland, J.L. & Jenkins, E. "Nature, nurture and connectionism: Implications for connectionist models of development" in van Lehn, K (Ed.), *Architectures for Intelligence - The twenty second (1998) Carnegie Symposium on Cognition*. Hillsdale NJ: Erlbaum 1999
- [3] O'Reilly, R.C. & Munakata, Y. "Computational Explorations in Cognitive Neuroscience" MIT Press 2000
- [4] Giancoli, D.C. "Physics", Prentice Hall 1998
- [5] Reed, R.D. & Marks, R.J. "Neural Smithing" MIT Press 1999
- [6] Bocaneala, F. & Bao, L. work in progress