

Real-Time Visualization of Neural Network Training to Supplement Machine Learning Education

Michael You and Jessica Yin
Carnegie Mellon University, myou, jessicay@andrew.cmu.edu

Abstract - In machine learning, neural networks have excelled at performing tasks at a high level with a simple and flexible implementation. Neural networks are particularly well-suited for novice programmers due to the availability of open-source libraries like TensorFlow and Caffe. However, novice programmers often neglect to learn beyond the black-box behaviors that these libraries provide. Introductory college students often lack the understanding of neural network internals, such as hidden layers and activation functions, and their interactions during training, which are crucial to efficiently solving more complex problems. Here, we present Omega³, a device that opens up the black-box of neural networks by visually representing how hidden layers behave during training in real-time. In addition, Omega³ provides an engaging tactile and visual educational experience to students, and waives the requirement for a strong programming background in order to learn about neural networks. In this paper, we will discuss the fabrication and set-up of Omega³ as well as evaluate and compare Omega³ to traditional lecture-based learning.

Index Terms – Hidden layer, modular, neural networks, visualization

INTRODUCTION

In the past decade, machine learning has emerged as one of the dominant techniques in software for building solutions in autonomous driving, natural language processing, image processing, and many other domains [1, 2, 3]. One leading technique for machine learning models is the neural network (NN), which is usually the best performing model compared to others [4, 5, 6, 7]. However, despite their success, the inner workings of NNs are seldom understood due to the sophisticated math associated with how they work [8]. As a result, people often treat NNs as black-boxes, which can lead to blindly changing parameters and features that describe the NN. As NNs become more complex, understanding how they work is vital for them to be used properly and effectively in future applications.

Currently, NN visualizations are limited. They are primarily software based, and require some degree of machine learning, data science, and/or programming experience [9, 10, 11].

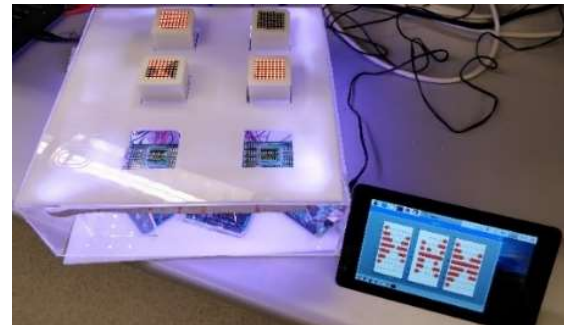


FIGURE I
OMEGA³ PERFORMING REAL-TIME TRAINING.

As machine learning gains popularity, an increasing number of people will need to learn about models such as neural networks. However, the programming and math can be inaccessible, out of scope, and present a high overhead to both students and teachers alike to learn.

There are numerous educational tools that target an analogous challenge with teaching children the fundamentals of programming. For example, toys such as Fisher-Price Think & Learn Code-a-pillar, littleBits, and LEGO BOOST make abstract programming concepts accessible to preschool and elementary school aged children [12, 13, 14]. These toys rely on hands-on and modular features to be engaging and effective, appealing to both visual and physical learning pathways.

TensorFlow Playground, a web-based interactive NN visualization, has been developed to address the growing need for accessible NN education. However, it still requires an extensive background in programming and math for basic use and may be less advantageous for tactile learners.

Building upon TensorFlow Playground, we present a simple, modular, physical representation of a NN that is easily accessible and requires minimal technical experience. This device, which we call Omega³, has customizable data and training settings, and displays a real-time visual of how a NN works as it runs. The user can move modular neuron blocks on the device that change the structure of the NN in real-time, which gives interactive feedback of the relationship between NN structure and model accuracy. This paper will first describe the materials of Omega³, the construction, the code, a few sample runs demonstrating features of Omega³, and finally, a method for evaluating its effectiveness.

FABRICATION

I. Materials and Overview

The component names and the current (as of December 2018) prices of the required materials are listed in Table I, and the product links are provided in Appendix A. The equipment required for the assembly of the device is listed in Table II. All items are initial investments and the device can be reused numerous times once assembled.

Because the fabrication process mostly consists of assembling components, the device can be put together by either the supervisors of the workshop or the students.

TABLE I
MATERIALS

Qty	Item	Price
1	Raspberry Pi	\$35.00
1	Elegoo Mega 2560 Microcontroller	\$13.99
3	Acrylic Sheets, 24" x 12" x 1/8"	\$8.99 each
6	LED Matrix, 8x8, with HT16K33 Module	\$13.99, 3-pack
6	Protoboard	\$6.95, 10-pack
42	Pogo pins	\$7.99, 100-pack
66	Socket pins	\$8.30, 50-pack
3	IRF644 MOSFET (optional)	\$1.50 each
1	LED Strip (optional)	\$8.49
Total:		\$127.18

TABLE II
EQUIPMENT REQUIRED FOR FABRICATION

Item
Laser Cutter
HDMI Display (e.g. computer monitor or Raspberry Pi display)
Soldering Iron with Solder
Hot Glue Gun

II. Step-by-Step Fabrication Process of Neuron Blocks

The main display components of Omega³ are the neuron blocks, which display the decision boundary of each hidden layer. Each block consists of:

- An 8x8 LED matrix display with the HT16K33 module
- An acrylic enclosure
- Protoboard
- Socket pins (11)

The assembly process for the neuron blocks is illustrated in Figure II. Source files for laser-cut parts can be found in Appendix B.

1. Laser cut the acrylic components with patterns provided in Appendix B.
2. Cut protoboard to size and solder socket pins to protoboard in configuration shown in Fig. II-A.
3. Wire LED matrix to socket pins and protoboard according to Fig. II-B. The completed step is shown in Fig. II-C.
4. Apply adhesive and press together the acrylic components, protoboard, and LED matrix to complete the assembly of the neuron block. An exploded view is provided in Fig. II-D (wires not shown) [15]. Fig. II-E shows the completed neuron blocks.

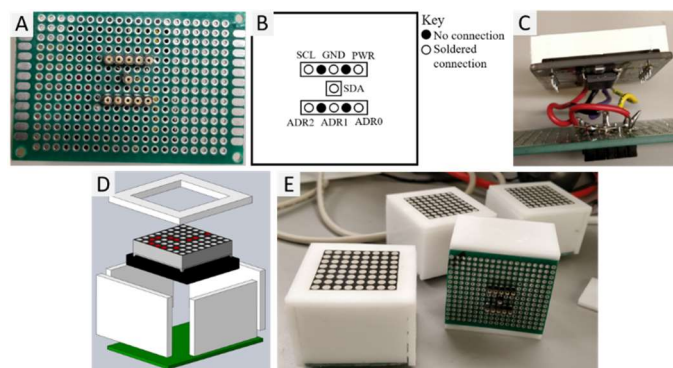


FIGURE II

ILLUSTRATED STEP-BY-STEP FABRICATION PROCESS OF NEURON BLOCKS.

III. Step-by-Step Fabrication Process of Base

The base of the device, where users place neuron blocks, hosts landing pads and socket pins that allow the neuron blocks to be connected to the microcontroller and the Raspberry Pi. An LED strip was glued around the perimeter of the inside of the base, purely for decoration, and thus is optional. The base consists of:

- An acrylic enclosure
- Protoboards (6)
- Pogo pins (42)
- Microcontroller
- Raspberry Pi
- LED strip & IRF644 MOSFETs (optional)

The assembly process for the base is illustrated in Figure III. Source files for laser-cut parts can be found in Appendix B.

1. Laser cut acrylic components with patterns provided in Appendix B.
2. Solder the pogo pins to the protoboard as shown in Fig. III-A.
3. Solder a wire to one end of each pogo pin and connect the wire to the appropriate pin of the microcontroller. The completed step is shown in Fig. III-B. Fig. III-C shows how the LED matrix display and the socket pins of the neuron block fit onto the pogo pins.
4. Wire the microcontroller and Raspberry Pi according to Fig. III-D.
5. Assemble the base by applying adhesive to the acrylic components and the protoboards. Acrylic supports may be placed between the layers for additional stability if desired. Place the Raspberry Pi and microcontroller inside the base before completing the assembly. An exploded view of the assembly can be found in Fig. III-E (wires not shown) [16, 17].
6. Connect the HDMI display to the Raspberry Pi.
7. Adhere the LED strip to the perimeter of the base. (Optional) Fig. III-F shows the completed base after this step.

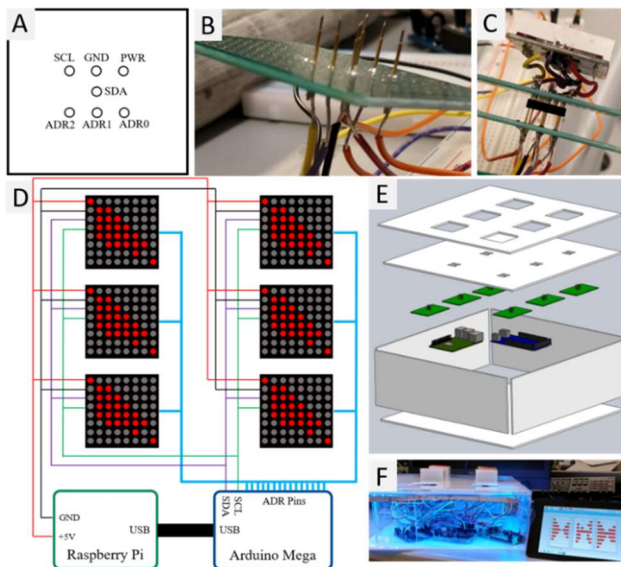


FIGURE III
ILLUSTRATED STEP-BY-STEP FABRICATION PROCESS OF BASE.

SOFTWARE

The Raspberry Pi and microcontroller were used to interpret user input, train the neural network, and provide real-time feedback of neural network training on the LED matrices. The Raspberry Pi was programmed to simultaneously train the neural network and relay the hidden layer data to the microcontroller. The microcontroller was programmed to interpret the serial communication from the Raspberry Pi and display it on the LED matrixes of connected neuron blocks. The complete code for the microcontroller and the Raspberry Pi can be found in Appendix B. The Raspberry Pi runs Raspbian and PyBrain, for which the weblinks and installation instructions are included in Appendix B.

I. Raspberry Pi Software

The software on the Raspberry Pi has the following workflow:

1. User chooses the type of data for NN to be trained on (e.g., solid, checkerboard, cross, kite, etc.).
2. Program randomly samples from chosen data distribution.
3. Program begins training the NN, sending hidden layer data to the microcontroller after each epoch.

During the entire training process, the Raspberry Pi will display the original data set, the sampled data set, and the predicted data set via the HDMI display. Figure IV shows an example of what would be shown on the HDMI display.



FIGURE IV

AN EXAMPLE OF HDMI OUTPUT FROM THE RASPBERRY PI. GRAPHS LEFT TO RIGHT: ORIGINAL DATA SET, SAMPLED DATA SET, PREDICTED DATA SET.

II. Microcontroller Software

The software on the microcontroller has the following structure:

1. Read LED matrix addresses.
2. Initialize LED matrix objects.
3. While the LED matrix is connected:
 - a. Read from serial port.
 - b. Interpret message.
 - c. Display information to LED matrix.

The open source Arduino IDE was used to program the Elegoo Mega 2560 with the Omega³ software. The LED Matrix library from Adafruit will need to be installed before use of the Omega³ Arduino program. The weblinks and installation instructions for the Arduino IDE and the Adafruit LED Matrix Library can be found in Appendix B.

EVALUATING EFFECTIVENESS

To evaluate the effectiveness of Omega³ as an educational device, the following experiment will be conducted. A pool of 24 students from Carnegie Mellon University, aged 18-22 years old, will be selected to participate based on their backgrounds with machine learning. One third of the students will have no previous exposure to neural networks, another third will have had some exposure to neural networks, and finally the last third will be proficient with neural networks. Half of each student group, categorized by their previous experience with neural networks, will be given a presentation about neural networks, while the other half will be given time to use Omega³.

Each user trial will be conducted with students individually. The trial will consist of the following steps:

1. Take the pre-activity survey about NN knowledge.
2. Participate in one 30-minute activity of:
 - a. Learn about neural networks through presentation slides and an oral presentation, or
 - b. Learn about neural networks through interaction with Omega³ and an oral presentation.
3. Take the first post-activity survey about NN knowledge immediately after the conclusion of the activity.
4. Approximately one week later, take the second post-activity survey.

The pre-activity survey will consist of the following questions:

1. Rate your overall understanding of neural networks on a scale of 1-10.
2. Explain how a neural network operates in your own words.

The first post-activity survey will include questions 1-2 from the pre-activity survey as well as the following:

3. Rate how useful your activity was to learning about neural networks on a scale of 1-10 (post-activity only).

4. Please suggest improvements to the device if any of its features were unclear or difficult to use (post-activity for 2b only).

The second post-activity survey will consist of questions 1-3 from the pre-activity and first post-activity surveys.

Once the experiment is completed, the results will be collected, analyzed, and reported.

CONCLUSION

Omega³ is a novel visual and interactive approach to neural network education. The device's modular nature and hands-on interface may be useful to a wide range of students with varying levels of experience with math, programming, and machine learning. Omega³ will increase accessibility to the understanding of how NNs operate as they become more widely utilized in everyday applications.

For future work, we will first execute the study on effectiveness presented in the paper. After analyzing the collected data, we will be able to identify improvements for future iterations of the device. A prospective feature is an expanded data functionality of Omega³ so that users can select from different types of data sets rather than the current binary sets included in the program, to display more capabilities of neural networks. This device could also be tested for effectiveness and use with a wider audience, such as high school and middle school students. We hope the development of Omega³ will emphasize a hands-on and interactive nature in STEM education.

ACKNOWLEDGEMENT

We would like to thank HackPrinceton for giving us the resources to build Omega³ and the 1517 Fund for providing financial support for the continued development of Omega³. We would also like to thank Professor Carmel Majidi and Tess Hellebrekers for useful suggestions to the manuscript.

REFERENCES

- [1] Mitchell, Tom M. Machine Learning. McGraw-Hill, 1997.
- [2] Bridge, James P., Holden, Sean B. and Paulson, Lawrence C. "Machine learning for first-order theorem proving," Journal of Automated Reasoning, vol. 53, pp. 141-172, Aug 2014.
- [3] Sarikaya, Ruhi, Hinton, Geoffrey E. and Deoras, Anoop. "Application of deep belief networks for natural language understanding," IEEE/ACM Trans. Audio, Speech and Lang. Proc., vol. 22, pp. 778-784, Apr. 2014.
- [4] Ciregan, Dan, Meier, Ueli and Schmidhuber, Juergen. "Multi-column deep neural networks for image classification," in 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 3642-3649, June 2012.
- [5] Krizhevsky, Alex, Sutskever, Ilya and Hinton, Geoffrey E. "Imagenet classification with deep convolutional neural networks," Commun. ACM, vol. 60, pp. 84-90, May 2017.
- [6] Wicczorek, Szymon, Filipiak, Dominik and Filipowska, Agata. "Semantic image-based profiling of users' interests with neural networks," 10 2018.
- [7] X. Li and X. Wu, "Constructing long short-term memory based deep re-current neural networks for large vocabulary speech recognition," CoRR, vol. abs/1410.4281, 2014.
- [8] Olah, Christopher. "Neural networks, manifolds, and topology." <https://colah.github.io/posts/2014-03-NN-Manifolds-Topology/>. Accessed: 2018-12-24.

- [9] Tensorflow, "A neural network playground." <http://playground.tensorflow.org/>. Accessed: 2018-12-24.
- [10] Yosinski, Jason, Clune, Jeff, Nguyen, Anh, et al. "Understanding neural networks through deep visualization." <http://yosinski.com/deepvis>. Accessed: 2018-12-24.
- [11] Bolton, Kris. "A practical introduction to artificial neural networks with python." <http://krisbolton.com/a-practical-introduction-to-artificial-neural-networks-with-python/>. Accessed: 2018-12-24.
- [12] "Think & Learn Code-a-pillar". <https://fisher-price.mattel.com/shop/en-us/fp/think-learn/think-learn-code-a-pillar-dkt39>. Web. Accessed: January 14, 2019.
- [13] "littleBits". <https://littlebits.com/>. Web. Accessed: January 14, 2019.
- [14] "LEGO BOOST". <https://www.lego.com/en-us/themes/boost>. Web. Accessed: January 14, 2019.
- [15] Sami, Mohamed. "LED Matrix 8*8." <https://grabcad.com/library/led-matrix-8-8>. Web. Accessed: December 25, 2018.
- [16] Stoianovici, Dan. "Raspberry Pi 3B+ Mock." <https://grabcad.com/library/raspberry-pi-3b-mock-1>. Web. Accessed: December 25, 2018.
- [17] hannah_mii. "Arduino Mega." <https://grabcad.com/library/arduino-mega-6>. Web. Accessed: December 25, 2018.

AUTHOR INFORMATION

Michael You, Undergraduate student, Department of Electrical and Computer Engineering, Carnegie Mellon University.

Jessica Yin, Undergraduate student, Department of Mechanical Engineering, Carnegie Mellon University.

APPENDIX A - MATERIALS

Web links to the products used to fabricate Omega³ are listed below:

- Raspberry Pi: <https://www.adafruit.com/product/3775?src=raspberrypi>
- Elegoo Mega 2560: <http://a.co/d/0fUEueU>
- Acrylic sheets: <http://a.co/d/bpR7mZL>
- LED matrix display: <http://a.co/d/isXfqF9>
- Protoboards: <http://a.co/d/e8fosrh>
- Pogo pins: <http://a.co/d/5AjGkkk>
- Socket pins: <https://www.digikey.com/short/pc5zvq>
- IRF644 MOSFET: <https://goo.gl/NhZn2W>
- LED Strip: <http://a.co/d/isZLFzB>

APPENDIX B - SOURCE FILES

Web links to various source files are listed below:

- Patterns for the laser cutting, code for the Raspberry Pi, and code for the microcontroller can be found here: <https://github.com/mikinty/Omega3>
- Raspberry Pi New Out Of the Box Software Installer (NOOBS) for Raspbian: <https://www.raspberrypi.org/downloads/noobs/>
- PyBrain, machine learning library for Python: <http://pybrain.org/>
- Arduino IDE: <https://www.arduino.cc/en/main/software>
- Adafruit LED Matrix library: https://github.com/adafruit/Adafruit_LED_Backpack