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

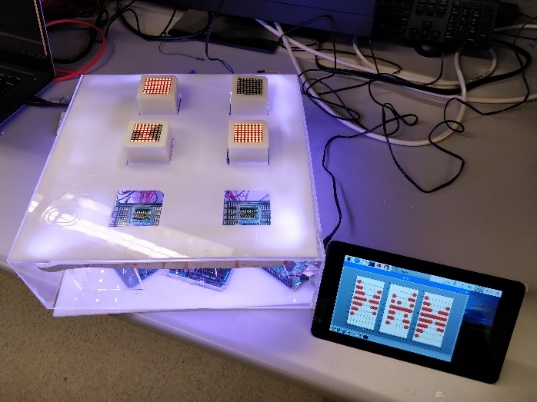


Figure 1

omega3 performing live-time training.

*Abstract* - Despite the popularity and success of neural networks in machine learning applications, gaining an understanding of their inner workings, especially the hidden layer, is often neglected by novice learners because of the complexity associated with these parts. Here, we present Omega3, a device designed to increase students' high-level understanding of how neural networks operate. We will discuss the device's fabrication and a method to measure how effectively Omega3 can be used as a learning tool compared to traditional teaching methods. Omega3 aims to open up the black-box of neural networks by visually representing how hidden layers behave during training in live-time. In addition, because of its hands-on and interactive nature, Omega3 provides an engaging educational experience to students and waives the requirement for a programming background to understand how neural networks function.

*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 technologies for building solutions such as autonomous driving, natural language processing, image processing, and many other applications [1, 2, 3]. One of the most successful 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 properly, due to the complicated math that is associated with how they work [8]. As a result, people often treat NNs as black-boxes, by only paying attention to inputs and outputs when tuning internal parameters to get the results they need for their specific application. Given that NNs are becoming more complex, understanding how they work rather than blindly tuning parameters 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]. Because machine learning is gaining popularity, more and more people will need to learn about models such as neural networks, including younger children and students from non-technical fields. However, for children and non-technical majors, programming and math can be inaccessible, out of scope, and present a high overhead to both the students and teachers alike to teach and learn.

There are numerous educational tools that currently target an analogous challenge with teaching children the fundamentals of programming. Specifically, several toys have become available in the past decade with the goal of taking abstract concepts from programming and making them more accessible by lowering the threshold of previous background knowledge required to receive an educational experience. Toys such as the Fisher-Price Think & Learn Code-a-Pillar, littleBits, and Lego Boost are marketed towards preschool and elementary school aged children and rely on hands-on and modular features to be engaging and effective [12].

Therefore, we present a simple, modular, physical representation of a neural network that is easily accessible and requires minimal technical experience. This device, which we call Omega3, can be used not only as an educational tool, but also as a useful tool for people who are designing NN architectures to more deeply understand the inner workings of NNs. Omega3 is easily modifiable due to its modular nature, has customizable data and training settings, and displays a live-time visual of how a NN works as it runs. 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 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 assembly pre-designed components, the device can be put together by either the supervisors of the workshop or the students.

TABLE I

Materials

|  |  |  |
| --- | --- | --- |
| Qty | Item | Price |
| 1  1  3  6  6  42  42  3  1 | Raspberry Pi  Elegoo Mega 2560 Microcontroller  Acrylic Sheets, 24” x 12” x 1/8”  LED Matrix, 8x8, with HT16K33 Module  Protoboard  Pogo pins  Socket pins  IRF644 MOSFET (optional)  LED Strip (optional)  **Total:** | $35.00  $13.99  $8.99 each  $13.99, 3-pack  $6.95, 10-pack  $7.99, 100-pack  $8.30, 50-pack  $1.50 each  $8.49  $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 (7)

The assembly process for the neuron blocks is illustrated in Figure 2. 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. 2A.

3. Wire LED matrix to socket pins and protoboard according to Fig. 2B. The completed step is show in Fig. 2C.

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. 2D (wires not shown) [13]. Fig. 2E shows the completed neuron blocks.

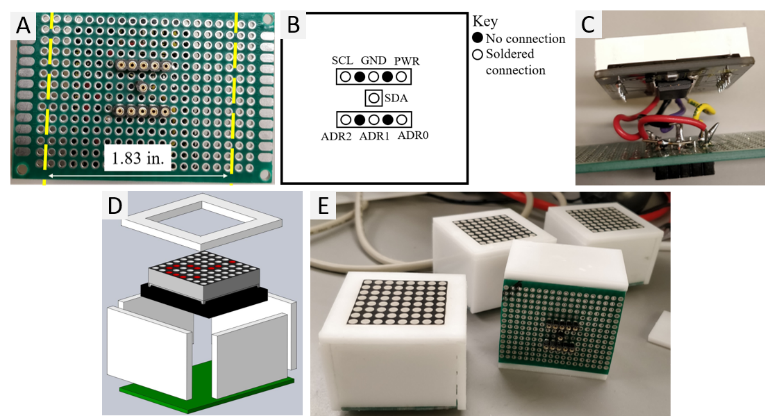


Figure 2

Illustrated step-by-step fabrication process of neuron blocks.

*III. Step-by-Step Fabrication Process of Base*

The base 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 3. 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 pogo pins to protoboard as shown in Fig. 3A after confirming the locations of the pogo pins allow the neuron blocks’ socket pins to be inserted.

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. 3B. Fig. 3C 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. 3D.

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. 3E (wires not shown) [14, 15].

6. Connect the HDMI display to the Raspberry Pi.

7. Adhere the LED strip to the perimeter of the base. (Optional) Fig. 3F shows the completed base after this step.

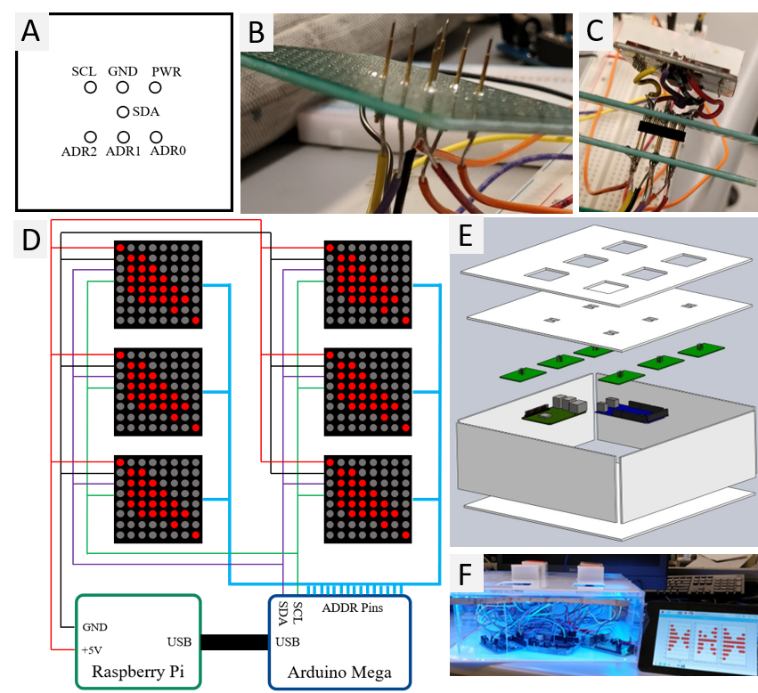


Figure 3

Illustrated step-by-step fabrication process of base.

Software

The Raspberry Pi and the microcontroller are the components that utilize custom software for Omega3. The Raspberry Pi was programmed to simultaneously train the neural network and relay the hidden layer information 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 both the microcontroller and the Raspberry Pi can be found 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 4 shows an example of what would be shown on the HDMI display.

The Raspberry Pi uses Raspbian, an open source operating system that was installed via New Out Of the Box Software (NOOBS). The software for the Raspberry Pi uses PyBrain, an open source machine learning library for Python, which will need to be installed before using the custom software for Omega3. The weblinks and installation instructions for NOOBS and PyBrain can be found in Appendix B.

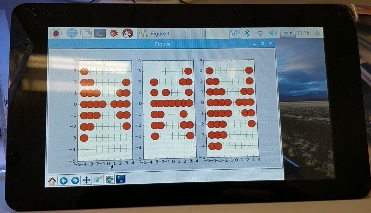


Figure 4

an example of the hdmi output from the raspberry pi. the leftmost graph shows the original data set, the middle graph shows the sampled data set, and the rightmost graph shows the 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:
   1. Read from serial port
   2. Interpret message
   3. 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 a survey about NN knowledge (pre-activity)
2. Participate in one 10-minute activity of:
   1. Learn about neural networks through presentation slides and an oral presentation, or
   2. Learn about neural networks through interaction with Omega3 and an oral presentation
3. Take a survey about NN knowledge (post-activity)

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

Once the experiment is completed, the results will be reported.

Conclusion

Omega3 is a novel visual and interactive approach to neural network education. The device’s modular nature and hands-on interface allows it to be useful to a wide range of students with varying levels of experience with math, programming, and machine learning. As neural networks become more widely utilized in everyday applications, developing an understanding of how they operate becomes more appealing.

For future work, we have considered expanding the 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. The development of more devices that emphasize a hands-on and interactive nature for abstract or complex STEM concepts could be a future research direction.

Acknowledgement

We would like to thank HackPrinceton for giving us the resources to build Omega3. We would also like to thank 1517 Fund for providing us financial support that has been invaluable to the continuation and development of Omega3.

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] Wieczorek, 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. “Understand-ing neural networks through deep visualization.” http://yosinski.com/deepvis. Accessed: 2018-12-24.

[11] Bolton, Kris. “A practical introduction to artificial neural networkswithpython.”http://krisbolton.com/a-practical-introduction-to-artificial-neural-networks-with-python/. Accessed: 2018-12-24.

[12] Schmidt, Gregory. “The best toys that teach kids how to code.”https://www.nytimes.com/2017/12/19/smarter-living/kids-toys-for-coding.html. Accessed 2018-12-25.

[13] Sami, Mohamed. "LED Matrix 8\*8." https://grabcad.com/library/led-matrix-8-8, Accessed: 2018-12-25.

[14] Stoianovici, Dan. "Raspberry Pi 3B+ Mock." https://grabcad.com/library/raspberry-pi-3b-mock-1, Accessed: 2018-12-25.

[15] hannah\_mii. "Arduino Mega." https://grabcad.com/library/arduino-mega-6, Accessed: 2018-12-25.

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