



## AI Education Matters: Teaching with Deep Learning Frameworks in Introductory Machine Learning Courses

**Michael Guerzhoy** (Princeton University, University of Toronto, and the Li Ka Shing Knowledge Institute, St. Michael's Hospital; [guerzhoy@princeton.edu](mailto:guerzhoy@princeton.edu))

DOI: [10.1145/3284751.3284756](https://doi.org/10.1145/3284751.3284756)

### Introduction

In this article, we demonstrate an assignment<sup>1</sup> in which students use TensorFlow to build a face recognition system. Students build shallow and deep neural networks for face recognition in TensorFlow and use transfer learning to obtain near-perfect performance on a simple face recognition task. Visualizing neural networks in order to explain how they work is central to the assignment.

The emergence of frameworks such as TensorFlow (Abadi et al., 2016) and PyTorch<sup>2</sup> and the explosive growth of ready-made open source machine learning systems on the Web changed the practice of machine learning. The pedagogy of machine learning faces challenges that are analogous to the challenges introductory computer science pedagogy started facing with the advent of high-level programming languages and GUI libraries (Astrachan, Bruce, Koffman, Kölling, & Reges, 2005): we would like for students to use the tools practitioners use to build real-world systems, but at the same time we want students to have a thorough understanding of how systems can be built “from scratch.”

We resolve this dilemma by advocating for an assignment style that approximates building a new machine learning system prototype as closely as possible. We avoid giving out incomplete starter code and asking students to “fill in the blanks.” Instead, we give students working code that is relevant to the task they are working on and that is similar to code that they might find on the Web. Students use as much or as little of the code as they want to complete the assignment.

Copyright © 2018 by the author(s).

<sup>1</sup><http://modelai.gettysburg.edu/2018/nnfaces/>

<sup>2</sup>Excellent introductory tutorials are available at <http://www.tensorflow.org> and <http://www.pytorch.org>



Figure 1: Demonstrating overfitting by visualizing the weights of a one-hidden-layer neural network

We ask students to visualize the models that they are working with, building on recent research work on visualizing and understanding neural networks (Zeiler & Fergus, 2014) (Springenberg, Dosovitskiy, Brox, & Riedmiller, 2014). Visualizing ML models is a nontrivial technical task which allows students to practice developing machine learning code, it is helpful when debugging ML models, and, finally, it allows students to gain a better intuition for why the models work and when they can fail. Pedagogically, this task is both nontrivial and feasible for students to accomplish.

### Model Assignment: Neural Networks for Face Recognition with TensorFlow

In the assignment, students train both shallow and deep networks to classify faces of six famous actors. Students adapt handout TensorFlow code in order to build and train the networks.

Students train a one-hidden-layer neural network for face classification. Students then improve the performance of their system by train-

ing a convolutional neural network to perform face classification; as a first step, students use the activations of a pre-trained AlexNet network (Krizhevsky, Sutskever, & Hinton, 2012) as the features for their face classifier.

Students visualize the shallow and the deep networks that they train.

### Introducing TensorFlow

TensorFlow is introduced by giving students working code that trains a one-hidden-layer neural network on the MNIST dataset. We also provide a beginner-friendly implementation of AlexNet in TensorFlow, along with weights pretrained on ImageNet<sup>3</sup>. The handout is designed to be very easily readable and only uses the most elementary TensorFlow features. This makes it possible for students to modify the code to suit their needs without having to immediately understand the highly-engineered neural network implementations that are included with TensorFlow. Students can (and do) reuse the generic handout code later on in their career.

### Visualizing Feedforward Networks

Visualizing the weights of shallow neural networks allows students to understand one-hidden-layer neural networks in terms of template matching, and allows students to explore overfitting. Students can demonstrate that, with enough hidden units, a one-hidden-layer neural network will “memorize” faces (see Fig. 1).

Students start exploring the visualization of deep neural networks by computing the gradient of a network’s output with respect to its input using TensorFlow and visualizing it. For the bonus component of the assignment, students are asked to explore a visualization strategy of their choice for understanding the convolutional network they train. Guided Backpropagation (Springenberg et al., 2014) is a common technique for students to attempt to implement.

<sup>3</sup>[http://github.com/guerzh/tf\\_weights](http://github.com/guerzh/tf_weights)

### Conclusion

Deep Learning frameworks should be introduced early in machine learning courses. Handout code that trains simple networks can be given out. Performing transfer learning and visualizing the neural networks that students train are two interesting tasks that can be assigned to students.

### References

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... Zheng, X. (2016). TensorFlow: A System for Large-Scale Machine Learning. In *USENIX Symposium on Operating Systems Design and Implementation* (Vol. 16, pp. 265–283).
- Astrachan, O., Bruce, K., Koffman, E., Kölling, M., & Reges, S. (2005). Resolved: objects early has failed. In *ACM SIGCSE Bulletin* (Vol. 37, pp. 451–452).
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems* (pp. 1097–1105).
- Springenberg, J. T., Dosovitskiy, A., Brox, T., & Riedmiller, M. (2014). Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*.
- Zeiler, M. D., & Fergus, R. (2014). Visualizing and Understanding Convolutional Networks. In *European Conference on Computer Vision* (pp. 818–833).



**Michael Guerzhoy** is a Lecturer in the Center for Statistics and Machine Learning at Princeton University, an Assistant Professor (Status Only) in the Dept. of Statistical Sciences at the University of Toronto, and a Scientist at the Li Ka

Shing Knowledge Institute, St. Michael’s Hospital. His professional interests are in computer science and data science education and in applications of machine learning to health-care.