ASEE'S VIRTUAL CONFERENCE
At Home with Engineering Education
JUNE 22 - 26, 2020 #ASEEVC
Paper ID #29911

Mathematics Content of an Undergraduate Course on Deep Learning

Prof. Yosi Shibberu, Rose-Hulman Institute of Technology

Dr. Yosi Shibberu is professor of mathematics at Rose-Hulman Institute of Technology. He has taught undergraduate courses on data mining, machine learning, bioinformatics and computational biology. Dr. Shibberu spent a year at Jimma University, Ethiopia, as a Fulbright Scholar and formerly held the endowed chair for innovation in science, engineering and mathematics education at Rose-Hulman Institute of Technology.

Mathematics Content of an Undergraduate Course on Deep Learning

Abstract

Deep learning, also known as neural networks, is a key AI technology that is increasingly important in science and engineering applications. Most of the undergraduate curriculum in science and engineering places emphasize on analytical models, often derived from first principles. In contrast, deep learning is based on empirical models learned from a large number of training examples. These empirical models are currently being used to solve challenging problems such as autonomous driving in unstructured environments or enabling robots to grasp arbitrary objects.

With the invention and widespread use of computers, the applied mathematics curriculum has evolved to include courses like numerical analysis that supplement analytical modeling with simulation and numerical solutions to complex problems. Even introductory STEM courses such as calculus, differential equations, matrix algebra and statistics are turning to computer-based simulation and solutions. The bootstrap method, for example, has become a main topic in some introductory statistics courses. Widespread adoption of deep learning will likely usher in another analogous evolution in the undergraduate mathematics curriculum.

Engineering students are familiar with core mathematical concepts used in deep learning such as linear functions, vector spaces, matrix arithmetic and the gradient vector. In this paper, the author summarizes his experience teaching an undergraduate course in deep learning in a mathematics department. Also presented are data that suggest that weak programming skills may not be as significant an obstacle for STEM majors as the author originally feared.

Introduction

Deep learning—sometime referred to colloquially as AI—is at the center of a wave of innovation that is changing the way consumers interact with products. It is being used to solve challenging technical problems such as autonomous driving in unstructured environments or enabling robots to grasp arbitrary objects. Deep learning is a special type of machine learning that automates the generation of useful data features. An introduction to deep learning for mathematicians is provided by Higham et at[1].

Deep learning is being used in an increasing number of undergraduate courses and projects[2 3 4 5 6 7]. A brief survey of earlier courses on neural networks is given by Shibberu[8]. In this paper we describe a course on deep learning taught four times in a mathematics department to a total of over 100 students, 93% of whom were undergraduates. The goals of the course are to:

- use the mathematics background of STEM majors to develop, from first principles, the key concepts used in deep learning.
- expose students to empirical modeling.
- expose students to the advantages of machine learning over machine programming.
- inspire students to use deep learning in their future work.

Organizing Principles

Courses can be organized either from a top-down perspective or from a bottom-up perspective. Ergezer et al[2] describe top-down approaches as providing students a better understanding of the big picture but providing less incentive to learn details. In bottom-up approaches, learning occurs in smaller increments and students are in a better position to understand the advantages of more advanced methods. Mathematics courses are typically taught bottom-up as they tend to focus on developing fundamentals.

We primarily follow a bottom-up approach, developing deep learning concepts from first principles. In addition to theoretical understanding, we want students to understand current best practices in deep learning, a goal also stated by Johnson[9]. We also want to inspire students to do further work in deep learning on their own. Deep learning is a rapidly evolving field and students must be prepared to adapt to changes, a concern also expressed by Johnson[9].

Inspiring students and incorporating best practices is best done using a top-down organization. We incorporate an efficient top-down component in the deep learning course that focusing on a widely used technique called transfer learning. We describe transfer learning in more detail in the next section.

Machine learning is normally a prerequisite for deep learning. For example, Tirkes[10] have created a two course undergraduate curriculum for deep learning that has machine learning as the first course. Since we are interested in making deep learning accessible to engineering students who have limited opportunities to take elective courses, we assume no previous exposure to machine learning. Like Hoover et al[11] and Johnson[9], we must introduce students in the course to the universal work flow of machine learning tasks. In particular, students must learn how to avoid the cardinal sin of machine learning— over-fitting data.

Listed below are the student outcomes for our deep learning course.

Students will be able to:

1. use linear and logistic regression to establish a baseline performance level for machine learning tasks.

2. use transfer learning to repurpose pre-trained neural networks to new machine learning tasks.
3. apply drop-out, early stopping and L^2 regularization to prevent over-fitting.
4. determine, by hand, the number of trainable parameters in fully-connected and convolutional layers of a neural network.
5. determine, by hand, the activation values of fully-connected and convolutional layers for given input values.
6. apply the principle of least squares estimation and the principle of maximum likelihood estimation to derive the optimal parameter value of a simple bias regressor and simple bias classifier respectively.

We try to assess student learning by asking students to perform tasks (i) using by hand computations, (ii) using computer programming and (iii) using a deep learning framework like Keras[12] that has a user-friendly programming interface.

Mathematics Content

A neural network is a parameterized, nonlinear, differentiable, function that is constructed by composing "layers" of a linear function[1] with a simple nonlinear function. When a large number of such layers are used, the neural network is called "deep". The main challenge in deep learning is solving the optimization problem that fits the parameters of a deep neural network to a (typically) large data set. Although neural networks were introduced in the 1960s, it is only recently that it has been possible though improvements both in algorithms and hardware to solve this optimization problem to sufficient accuracy.

The main optimization algorithm used in deep learning is called stochastic gradient descent. Gradient descent is often used as an example of an application in multivariate calculus courses. Stochastic gradient descent differs from gradient descent in that at each iteration, a random sample of the data (without replacement) is used to compute the gradient vector.

Transfer learning is a widely used method in deep learning in which pre-trained neural networks are re-purposed to new tasks. Most of the pre-trained weights of the neural network are reused. Only the weights of the last few layers are re-trained to the new task. Since the last layer of a neural network is a logistic regressor, we use a bottom-up approach to first teach students the fundamentals of logistic regression. Then, we introduce transfer learning using a top-down approach, by treating pre-trained neural networks as fixed nonlinear transformations that transform raw input data features into more useful features for logistic regression.

Convolutional neural networks are an important class of neural networks that contain one or more convolutional layers. Convolution is a linear operation and thus has a special (somewhat complex) matrix representation. By focusing on linear combinations rather than matrices, we provide students with a unified treatment of both the more general linear operation used in logistic

[1] affine with respect to the input, but linear in the parameters

regression and the more specialized convolution operation used in convolutional neural networks. In particular, students are asked to compute convolutions of 4×4 pixel images by hand using linear combinations.

The multivariate chain rule plays a crucial role in the back-propagation algorithm that computes the gradient vector (with respect to the weights) of a neural network. Instead of introducing students to back-propagation for actual neural networks, we instead believe it is more productive to ask students to apply the back-propagation algorithm, by hand, to simpler multivariate functions of no more than three variables. We provide students with a computational graph to guide their computations. Students only need to fill in the nodes of the computational graph by applying the chain rule using an intuitive procedure in discrete optimization called dynamic programming.

Results

In this section we present data on student performance that suggests computer programming skills may not be a major obstacle for STEM majors in a deep learning course.

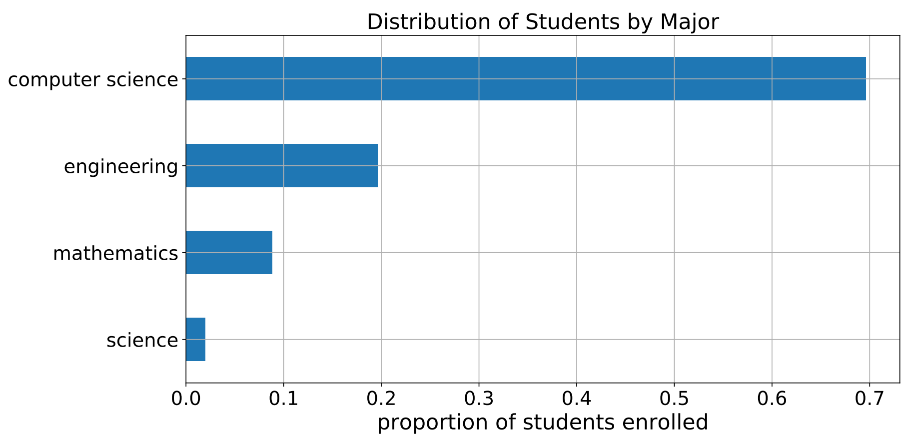

Figure 1: Distribution of students by major who were enrolled in deep learning the last four times the course was taught.

Our deep learning course has been taught four times over four years to over 100 students, 93% of whom were undergraduates and approximately 20% of whom were engineering majors. See Figure 1 for the distribution of majors taking the deep learning course.

Figure 2 contains box plots of standardized final class averages grouped by computer science and non-computer science majors. From Figure 2 we see that graduate students, all of whom were engineering majors, had a median class average one standard deviation higher than computer science majors. Possibly the additional year or two of experience of graduate engineering students was more significant to class performance than the computer programming experience of undergraduate students majoring in computer science. At the junior level, computer science majors had a slightly higher median average and at the senior level, computer science students had a median average more than one half a standard deviation higher than their non-computer science counterparts.

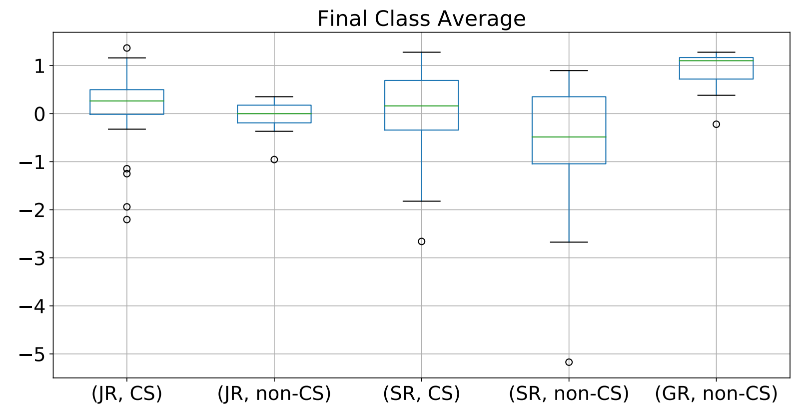

Figure 2: Distribution of standardized final class averages. JR represents juniors, SR represents seniors and GR represents graduate students. Software engineering is listed under computer science (CS) and computer engineering is listed as not computer science (non-CS). Double majors are listed under their primary major.

As a control, in Figure 3, we plotted the standardized final quiz averages of students. The quizzes were online quizzes graded automatically and required no programming. Again, the graduate students scored a full standard deviation higher. At the senior level there is less difference between computer science and non-computer science majors, but at the junior level, non-CS students scored half a standard deviation below their junior computer science counterparts. The quiz averages suggest that there may be factors other than programming skills that may be just as important in determine student performance.

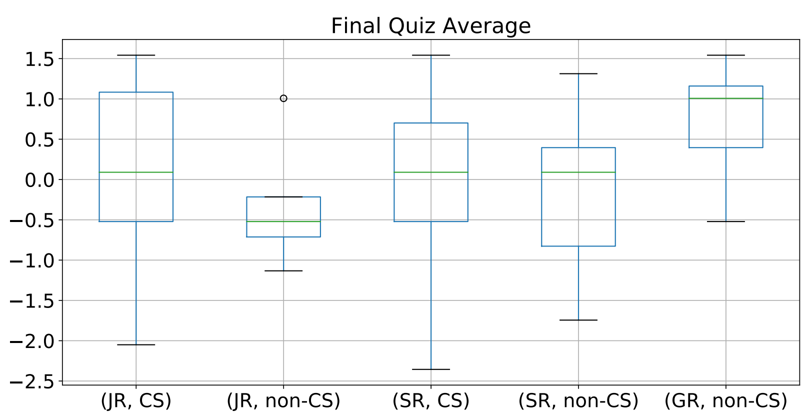

Figure 3: Distribution of standardized quiz averages group by major. The quizzes were taken online, graded automatically and required no computer programming.

The last time the deep learning course was taught, we asked students to rate each homework assignment as easy, average or hard. The homework assignments required significant amounts of computer programming using Python and the Keras deep learning framework implemented in

Tensorflow 2.0. Figure 4 does not indicate any major differences in how computer science and non-computer science students perceived the level of difficulty of the homework assignments.

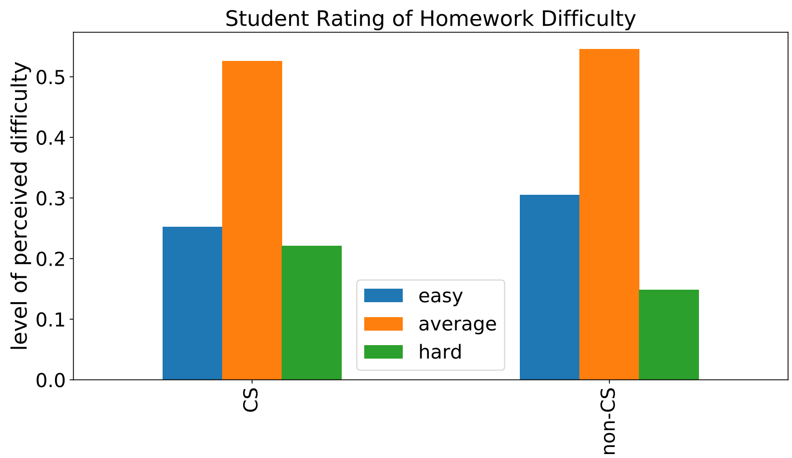

Figure 4: Student rated each of 13 homework assignments as easy, average or hard. All but one homework assignment required significant programming. Shown are the distribution of responses grouped by computer science majors (CS) and non-computer science majors (non-CS).

It is likely that the non-computer science majors who enrolled in deep learning have better programming skills than a typical STEM major's programming skills. Most were familiar with Python programming which is not required for science and engineering majors at our institution. However, computer science juniors and seniors have taken far more computer programming courses than these undergraduate STEM majors and even graduate engineering majors. Deep learning packages have become easier to program over the four years the deep learning course was taught. A larger sample size using more recent data is needed before stronger conclusions can be reached.

Discussion

In this section we compare our deep learning course to three other deep learning courses described in recent publications. Tirkes et al.[10] have developed a two course sequence for deep learning named *Fundamentals of Machine Learning* and *Applications of Deep Learning*. Johnson[9] has developed a 15 week deep learning course that is similar to our 10 week course. No previous exposure to machine learning is assumed. However, students are expected to have several years of programming experience.

Johnson's course covers recurrent neural networks, a major topic not covered in our own deep learning course. Unlike our course, students are expected to read original research publications. This enables students to stay abreast of the latest developments in a rapidly evolving field. Our approach has been to focus instead on mathematical fundamentals that are not likely to change as the field evolves.

Hoover et al.[11] have developed a deep learning course called *Creative AI* that does not require a calculus background. The course is taught top-down from a practitioner's perspective and serves as a bridge course that inspires students to take further more traditional data science courses. We too believe inspiring students to do further work in the future is a very important goal.

Conclusions

Deep learning is a rapidly evolving subject that is growing in importance. STEM majors should be familiar with it. Junior and senior engineering majors have already been introduced to many of the core mathematical concepts used in deep learning. We described an efficient bottom-up approach that introduces students to the mathematical fundamentals of deep learning and that does not require a machine learning course as a prerequisite. A top-down component on transfer learning quickly introduces students to deep learning best practices. The data presented on student performance does not indicate that computer programming skills is a major obstacle for STEM majors in the deep learning course. As deep learning frameworks continue to become easier to program, programming skills will likely become less of a concern for STEM majors.

References

[1] Catherine F Higham and Desmond J Higham. Deep learning: An introduction for applied mathematicians. *SIAM Review*, 61(4):860–891, 2019.

[2] Mehmet Ergezer, Bryon Kucharski, and Aaron Carpenter. Work in progress: Designing laboratory work for a novel embedded AI course. In *2018 ASEE Annual Conference & Exposition*, Salt Lake City, Utah, June 2018. ASEE Conferences. https://peer.asee.org/31280.

[3] Jayson Paul Mercurio, Ayesha R Iqbal, Kevin Yamada, Jose L Guzman, Xiaorong Zhang, Wenshen Pong, Amelito G Enriquez, Zhaoshuo Jiang, Cheng Chen, Kwok Siong Teh, et al. Inspiring community college students in electrical and computer engineering research through live digit recognition using Nvidia's Jetson Tx1. In *2018 ASEE Zone IV Conference*, 2018.

[4] Jack Bergquist and Anahid A Behrouzi. Multidisciplinary research efforts in post-earthquake civil infrastructure reconnaissance. In *2018 ASEE Annual Conference & Exposition Proceedings*, 2018.

[5] Deng Cao, Cadance Lowell, and Augustus Morris. Undergraduate research: Introducing deep learning-based image classification to undergraduate students. In *2018 ASEE Annual Conference & Exposition*, Salt Lake City, Utah, June 2018. ASEE Conferences. https://peer.asee.org/31171.

[6] Jennifer Melanie Bastiaan, Diane L. Peters, Juan R. Pimentel, and Mehrdad Zadeh. The autodrive challenge: Autonomous vehicles education and training issues. In *2019 ASEE Annual Conference & Exposition*, Tampa, Florida, June 2019. ASEE Conferences. https://peer.asee.org/33371.

[7] Vivek Sivaraman Narayanaswamy, Uday Shankar Shanthamallu, Abhinav Dixit, Sunil Rao, Raja Ayyanar, Cihan Tepedelenlioglu, Mahesh K Banavar, Sameeksha Katoch, Emma Pedersen, Photini Spanias, et al. Board 162: Online modules to introduce students to solar array control using neural nets. In *2019 ASEE Annual Conference & Exposition*, 2019.

[8] Yosi Shibberu. Introduction to deep learning: A first course in machine learning. In *paper presented at 2017 ASEE Annual Conference & Exposition*, 2017.

[9] Jeremiah W Johnson. Teaching neural networks in the deep learning era. *Journal of Computing Sciences in Colleges*, 34(6):16–25, 2019.

[10] Güzin Tirkeş, Cansu ÇiÇdem Ekin, Atila Bostan, Murat Karakaya, et al. An undergraduate curriculum for deep learning. In *2018 3rd International Conference on Computer Science and Engineering (UBMK)*, pages 604–609. IEEE, 2018.

[11] Amy K Hoover, Adam Spryszynski, and Michael Halper. Deep learning in the IT curriculum. In *Proceedings of the 20th Annual SIG Conference on Information Technology Education*, pages 49–54, 2019.

[12] François Chollet et al. Keras. https://keras.io, 2015.