Paper ID #20509

Introduction to Deep Learning: A First Course in Machine 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 and bioinformatics and computational biology. Dr. Shibberu recently spent a year at Jimma University, Ethiopia, as a Fulbright Scholar and is the current endowed chair for innovation in science, engineering and mathematics education at Rose-Hulman Institute of Technology.

Introduction to Deep Learning
A First Course in Machine Learning

Abstract

A new undergraduate course on deep learning is described. Most of the students who took the course were junior and senior computer science majors. Nearly all the students in the course had not had a previous course in machine learning. The course builds on basic concepts students learn in calculus, statistics and probability courses. Key concepts from machine learning, e.g. the cardinal sin of over-fitting, are introduced in the context of deep learning, in a problem driven manner, so that students discover and observe these concepts for themselves. A concept map as well as useful online resources are described in the appendix.

Outline of Article

The article begins with an explanation of what deep learning is followed by a brief review of previously taught courses related to deep learning. Then, several principles used to develop and teach the new course are described. A discussion of what worked and did not work in the new course is given in the Results section. Based on this discussion, several best practices for teaching deep learning at the undergraduate level are proposed. The article concludes with an exploration of how a deep learning course might be coordinated with a course on robotics.

What is deep learning?

Deep learning is the key enabling technology behind the current generation of self-driving cars and is responsible for the dramatic improvements seen in recent years in speech and image recognition. A growing number of companies are investing heavily in deep learning. The number of successful applications of deep learning is growing rapidly.

Deep learning is a special type of machine learning that can take advantage of the growing availability of big data and the increasing computing power of GPUs (Graphical Processing Units). The potential of deep learning in engineering applications is currently under investigation. For example, Cheon et al.[1] replace a PID motor controller with a deep learning based controller and Sanchez et al.[2] use deep learning to learn optimal feedback controllers. The automobile industry is investing heavily in applications of deep learning to autonomous vehicles. Consumer products that depend on deep learning, products like Amazon's Alexa and Google Home, are the tip of a new wave of innovation that will change the way consumers interact with products. Scientists and engineers with expertise in deep learning are already in short supply.

Deep learning is based on an old technology — neural networks. In fact, some argue that deep learning is simply a re-branding of neural networks, a technology that has been through several "hype-cycles[3]." Neural networks date to as far back as the 1940's and 1950's. In 1957, Frank Rosenblatt introduced the Perceptron, the forerunner of today's modern neural networks. What is new today is the availability of big data sets to train large and deep neural networks and the availability of the computing resources needed to run the training algorithms.

A key advantage of deep learning is that deep learning automates feature engineering. Good data features can take decades to discover and develop by hand. The automated feature engineering of deep learning makes it easier for users who are not domain experts in a given domain to use deep learning in that domain[4].

Teaching Deep Learning

Lavesson[5] and Chowdhury[6] each cite the weak mathematics backgrounds and programming skills of students as obstacles that must be overcome in graduate machine learning and neural network courses. Chowdhury also describes the challenging breadth vs depth trade-off that must be negotiated. Segee and Amos[7] taught students how to use Excel to program simple neural networks and Shamoon et al.[8] used a neural network character recognition system to teach matrix algebra to students. Imberman[9] had students train a simple two node, two hidden layer neural network to steer a LEGO mobile robot along a duct tape road. Students were exposed to the advantages of learning vs programming.

Inspiring students should be an important objective of an undergraduate deep learning course. Deep learning is a rapidly evolving field. Students need to be curious and motivated to learn about the latest developments on their own.

At the undergraduate level, depth is far more important than breadth. Undergraduate students need to develop a solid foundation on which to build future learning. For example, nearly half of the deep learning course is focused on linear and logistic regression. The course is also exclusively focused on supervised deep learning as unsupervised and reinforcement deep learning are currently not widely used in industry. (A concept map for the course is given in the Appendix. See Fig 1.)

Deep learning can be taught using elementary concepts from calculus, statistics and probability theory. Most science and engineering students have covered these topics in required courses by the time they are juniors and seniors.

Limited programming experience is likely to be the biggest obstacle an engineering student will face in a deep learning course. As the field matures, deep learning software packages are becoming easier to use. Software packages that require limited amounts of programming are under active development making deep learning increasingly accessible to a broader group of students.

At the graduate level, machine learning is normally considered a prerequisite for deep learning. However, a machine learning prerequisite for an undergraduate deep learning course would significantly reduce the number of students who can take the deep learning course. The engineering curriculum in particular is already filled with a large number of required courses. A more practical approach is to combine machine and deep learning in a single course that emphasises areas of machine learning relevant to deep learning, but does not neglect fundamental machine learning concepts.

A class project should be an important component of an undergraduate deep learning course. Undergraduate students should be exposed first-hand to both the challenges of training a deep neural network and the surprising capabilities of the network once it has been properly trained. A group project counted for 30% of a student's grade in the deep learning course.

Results

Twenty-four students enrolled in the deep learning course: 1 sophomore, 9 juniors, 12 seniors and 2 graduate students. Twenty of the student were either computer science or software engineering majors and the rest were engineering majors.

Student comments on the course evaluations at the end of the course indicated that students felt that the deep learning course was well structured and the topics were well ordered. They also appreciated the time spend during class working on problems and the opportunity that the class project gave them to exploring topics on their own. However, some felt the need for more lectures, particularly on convolutional neural networks. Several students appreciated the fact that key concepts were covered in depth. One student mentioned cross-entropy as a topic that they learned about in depth and would not have had they studied deep learning on their own.

Data preprocessing is a time consuming step and requires some skill. The author partially preprocessed the data sets used for the course to spare the students some of this effort. Students, however, preprocessed the data for their own projects.

While it is not too difficult to program a simple neural network from scratch, convolutional neural networks are best implemented using a deep learning framework. TensorFlow was used for the course. It was fairly easy to install TensorFlow using the Anaconda package, but only 3 students successfully installed TensorFlow on their personal GPUs. These students found the process to be challenging. However, these student were able to train much more sophisticated neural networks than the rest of the class. One student group was able to train a generative adversarial network for their class project. Programming TensorFlow was fundamentally different from the programming the students had done in their previous classes and required a period of learning and adjustment.

Linear and logistic regression networks (no hidden layers) with more than 1,000 nodes can easily be trained using the current generation of laptop CPUs. Introducing three or more hidden layers, however, significantly increases the computational requirements and training times from minutes to hours or more. For a fixed data set size, larger networks are easier to train than smaller ones, provided that sufficient computing power is available to train the larger networks.

Several excellent online resources exist that can be used to support a deep learning course. One of the best is the neural network playground. Many videos of keynote conference presentations and panel discussions on deep learning are available for free on YouTube. These presentations are quite accessible to STEM undergraduates and can be inspirational.

Best Practices

Based on the author's experiences teaching the undergraduate deep learning course for the first time, several best practices are proposed below:

- Make full use of online resources. Consider sharing any resources created to ease the burden for others.
- Provide students with partially preprocessed data sets and plenty of skeleton code to ease their programming burden. Preparing the data sets and creating and maintaining the skeleton code, however, is a significant burden on the instructor that should not be underestimated.
- Select a few core concepts to cover in-depth and let students explore other concepts on their own.
- If possible, use a centralized server or cloud computing. Students will have a common installation making it easier for the instructor to help students debug their code. Installation problems for students will also be eliminated. Large data sets can be placed on the server so that students don't need to download these data sets to their laptops. Students can also train their neural networks overnight.
- Introduce students to a user friendly interface to TensorFlow such as Keras. TensorFlow's native interface was challenging for most students in the course to master.

Deep Learning and Robotics

Kumar[10] describe the advantages of using robotics to teach AI concepts: Robots are more engaging than software. They also promote hands-on learning. Testing is in a physical rather than virtual environment. Berry[11] describes a mobile robot course in which students program basic behaviors, e.g. wall following or obstacle avoidance. Berry also discusses possibilities for including higher level AI concepts and computer vision in future iterations of the course.

Team teaching a coordinated deep learning course with a robotics course would require significant planning and student cooperation between courses, but would likely have many of the advantages described in the previous paragraph. Additional benefits to faculty could include enhanced (multidisciplinary) collaboration and junior faculty mentorship[12].

A new generation of hardware like NVIDIA's Jetson mobile deep learning card, combined with transfer learning, could ease the burden of incorporating deep learning into an undergraduate robotics course. Transfer learning involves taking a fully trained neural network and re-purposing it by using a limited amount of data to partially retraining the network. Fully trained networks can also be combined like building blocks in interesting ways to create new networks. For example, neural networks work surprisingly well at combining multi-modal sensor data.

Conclusions

Deep learning is creating a wave of innovation that will fundamentally change the way that consumers interact with products. Engineering students need to be exposed to the advantages of machine learning over machine programming. By focusing on depth instead of breadth, deep learning can be taught as a first course in machine learning. Weak programming skills is likely to be the biggest challenge engineering students will face in a deep learning course. To address this challenge, instructors should provide students with partially preprocessed data sets, skelton code to help students get started and should use a user friendly deep learning framework like Keras.

Appendix: Online Resources for Teaching Deep Learning

- Neural Network Playground: Train a small neural network in your browser! Highly recommended for illustrating concepts such as under and over training neural networks.
- Homework problem used to inspire students:

  Watch the video *Why you can't afford to ignore Deep Learning* (43 min) by Jeremy Howard and answer the following questions. (Read the questions before watching the video.) (a) Describe two recent "amazing" applications of Deep Learning presented by Jeremy Howard. (b) Describe the key differences between Machine Learning and Deep Learning. (c) According to Jeremy Howard, what impact will Deep Learning have on society in the future.
- JupyterHub: A free package that makes it possible for students to seamlessly run Jupyter notebooks on a centralized server from within their browser. (The author was not able to get this package properly installed in time for the course.)

References

[1] Kangbeom Cheon, Jaehoon Kim, Moussa Hamadache, and Dongik Lee. On replacing pid controller with deep learning controller for dc motor system. *Journal of Automation and Control Engineering Vol*, 3(6), 2015.

[2] Carlos Sánchez-Sánchez, Dario Izzo, and Daniel Hennes. Optimal real-time landing using deep networks. In *Proceedings of the Sixth International Conference on Astrodynamics Tools and Techniques, ICATT*, volume 12, pages 2493–2537, 2016.

[3] Gary Marcus. Hyping artificial intelligence, yet again. *The New Yorker*, 2014.

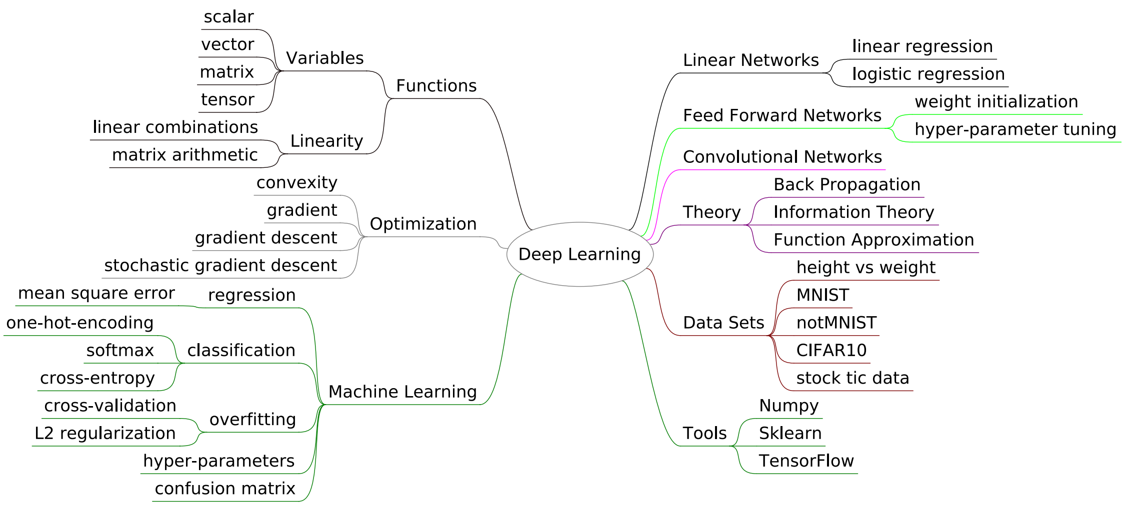

Figure 1: Concept Map for an Undergraduate Deep Learning Course

[4] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[5] Niklas Lavesson. Learning machine learning: a case study. *IEEE Transactions on Education*, 53(4):672–676, 2010.

[6] Fahmida N. Chowdhury. Interdisciplinary course on neural networks at the graduate level. In *1997 Annual ASEE Conference*, Milwaukee, Wisconsin, June 1997. ASEE Conferences.

[7] Michael D. Amos and Bruce E. Segee. Artificial neural networks using microsoft excel for windows 95. In *1997 Annual ASEE Conference*, Milwaukee, Wisconsin, June 1997. ASEE Conferences. https://peer.asee.org/6424.

[8] Sami Shamoon Sharlene Katz, Bella Klass-Tsirulnikov. Using neural networks to motivate the teaching of matrix algebra for k-12 and college engineering students. In *2007 Annual ASEE Conference and Exposition*, Honolulu, Hawaii, June 2007. ASEE Conferences.

[9] Susan P Imberman. An intelligent agent approach for teaching neural networks using lego® handy board robots. *Journal on Educational Resources in Computing (JERIC)*, 4(3):4, 2004.

[10] Amruth N Kumar. Three years of using robots in an artificial intelligence course: lessons learned. *Journal on Educational Resources in Computing (JERIC)*, 4(3):2, 2004.

[11] Carlotta Berry. Mobile robotics: A tool for application based integration of multidisciplinary undergraduate concepts and research. In *2010 Annual ASEE Conference & Exposition*, Louisville, Kentucky, June 2010. ASEE Conferences. https://peer.asee.org/15642.

[12] Michael Georgiopoulos, Ronald F DeMara, Avelino J Gonzalez, Annie S Wu, Mansooreh Mollaghasemi, Erol Gelenbe, Marcella Kysilka, Jimmy Secretan, Carthik A Sharma, and Ayman J Alnsour. A sustainable model for integrating current topics in machine learning research into the undergraduate curriculum. *IEEE Transactions on Education*, 52(4):503–512, 2009.