Machine Learning:
An Undergraduate Engineering Course

Sami Khorbotly*
Valparaiso University
Sami.khorbotly@valpo.edu

Introduction

In today's quickly changing world, staying up-to-date is a recipe for success. This is particularly true in the fields of Electrical and Computer Engineering (ECE). While the main concepts of linear circuits and programming remain unchanged, the tools and the applications are changing at a very fast pace. As a result, curriculum committees within ECE programs across the nation are continuously striving to update their curricula to prepare their graduates for success in both the industrial and the research worlds.

Machine learning (ML) is an emerging field that started as a topic in computing sciences but evolved to gain an increasing popularity in the industries related to Mechanical Engineering, ECE, as well as other fields in the humanities and social sciences. As a result, it is becoming a staple on the list of "highly-demanded skills" for many of the ECE employers. In order to fulfill this workforce need and help increase the employability of our graduates, an undergraduate machine learning course has been developed and is currently being taught for the first time in Spring 2022.

Several challenges were encountered when developing this course. First, considering the myriad of topics that can fit under the ML umbrella, it was necessary to find an adequate breadth vs depth balance to design a course that fits within the time constraints of an academic calendar. Moreover, considering the undergraduate nature of the course audience, we had to simplify some of the high-complexity mathematical concepts. The challenge was to simplify the Mathematics in a way that does not compromise the rigor of the course. Another challenge was finding the right balance between covering the conceptual background behind the various ML methods as opposed to superficially teaching the CAD tools that allow the students to find solutions without looking "under the hood". Finally, as numerous computing platforms are available to implement ML systems, a thorough investigation necessary to identify the most adequate platform to teach and use in the course.

This paper shares the "nuts and bolts" of the newly developed course. It lists the topics that we thought are appropriate to be included in this undergraduate course and the adequate amount of depth for the targeted audience. It also discusses the selected computing platform with the rationale for all the made decisions. Our hope is that the paper will help reducing the learning curve and the amount of preparative work for colleagues who may be interested in teaching a similar course at their own institutions.

Historical Background

It is historically common to see the terms "Machine Learning" and "Artificial Intelligence" used interchangeably to describe the same set of methods and techniques. The Merriam-Webster dictionary defines intelligence to be "The ability to learn/understand/deal with new or trying situations." In this context, it is reasonable to consider ML a form of intelligence. The reality, however, is that there is nothing intelligent about computing machines and that their capabilities are limited to implementing a pre-determined algorithm without the slightest deviation. We therefore feel that Machine Learning is a more accurate title for the course.

According to our research, the first notion of ML goes back to 1943 when McCollough and Pitts introduced their Perceptron[1]. In 1945, Alan Turing[2] made his game-changing statement regarding ML most fundamental requirement: "*In my opinion, this problem of making a large memory available at reasonably short notice is much more important than doing operations such as multiplication at high speed. Speed is necessary if the machine is to work fast enough for [it] to be commercially valuable, but a large storage is necessary if it is to be capable of anything more than rather trivial operations. The storage capability is therefore the more fundamental requirement."*

The term "Artificial Intelligence" (AI) was first used at the Dartmouth conference[3] in 1956. At this conference, four individuals who are considered the founders of the field: Claude Shannon, Marvin Minsky, John McCarthy, and Nathan Rochester, defined AI as follows: *"It is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable."*

In the few decades after the AI conference, very little progress happened in the field. Researchers in both academia and the industry mostly focused on the hardware side of computing. The outcome was, as predicted by Gordon Moore and his famous Moore's law, increasingly smaller and faster computers. That era was also characterized by the invention of many new technologies in the areas of networking (the internet), geo-location (GPS systems), and mobile communications (cell phones). Most of the computer science researchers dedicated their efforts to the service of these new technologies.

At the beginning of the 21st century, AI resurrected as a hot topic of research with the ML term dominating as a new buzzword. The reason for this resurrection can be attributed to three main factors. The first factor is the abundance of computing power because fast computers are nowadays available and affordable everywhere. The second factor is the abundance of computer storage due to the invention of the inexpensive solid-state storage devices. Finally, the abundance of extensive training data that can be collected via the Internet of Things (IOT) sensors, crowdsourcing, or other similar media.

At this day and age, ML applications are everywhere around us. Humans can have conversations with computing machines such as Siri or Alexa. Shopping online, odds are that you will receive ML advice from Amazon or Netflix. Even when you are just surfing the internet, you see targeted advertisements that are custom-made to match your profile. But ML applications are not limited to software applications. Intelligent cyber physical systems are slowly, but surely replacing human driven and controlled machines. Autonomous robots, smart drones, and the autonomous Google cars are just a few examples around us. That is why it is extremely important for our students to be have the skills to design and implement those systems.

Course Logistics

The course is offered as a 3 credit upper level technical elective in the ECE department. It is accessible to Junior-level and Senior-level students in the Electrical Engineering, Computer Engineering, Mechanical Engineering, and Computer Science students. It is expected that a student in the course must have previous exposure to advanced calculus (differentiation and partial derivatives), linear algebra (vector and matrix operations), probability & statistics, and computer programming.

The class meets three times every week for 50 minutes per meeting. While the official course description states that it consists of three weekly lectures with no laboratory components, the instructor lectures twice a week while reserving the third class meeting for an active learning in class project session where students implement the concepts they have learned about in the two lectures in that week.

The student performance will be assessed via three different categories:

1. <u>Weekly assignments:</u> every week, students will be given homework problems pertaining to the lecture contents. In addition, students will have to implement a programming assignment during the third class meeting every week. Students will submit a weekly assignment including the solved problems and the completed programming assignment. The total of all weekly assignments will make up 30% of the total course grade.
2. <u>Examinations:</u> Students will have to sit for two exams during the semester and one final exam at the end of the semester. These exams will be all in-class exams and will cover conceptual questions. The final exam will be worth 20% of the course grade while each of the other two exams will be worth 10% of the grade.
3. <u>Project:</u> Students will be required to complete a course project of their choice. Students can work in small groups of two (self-selection) or individually. The projects must be approved by the instructor midway through the semester. The students must also demonstrate their projects in a class presentation and submit a written report before the end of the semester. The project grade will make up 30% of the overall course grade.

Teaching Schedule

The most critical aspect of this course was the selection of the topics to be covered and the depth at which they are taught. While a whole lot of subjects can be included, the reality is that only 14 weeks of classes are available. With two lectures per week, as discussed above, and a few lecture slots dedicated to in-class exams and project presentations, that leaves a total of 24 lectures dedicated to covering new topics. These lectures will be used as follows:

1. <u>Introduction (4):</u> In the first two weeks, we will spend four lectures to introduce the course logistics and the topic. We will define what ML is and discuss its famous real-world applications. In this section, we will also present a historical perspective of the evolution of

the field, similar to the discussion above. Finally, the students will be introduced to the Gradient Descent (GD) algorithm, which will be later used in many of the ML techniques.

2. Regression (3): Over the span of three lectures, we will introduce the concept of Supervised Learning (SL) and talk in details about one SL technique – linear regression. Naturally, since this will be the students' first exposure to SL, we will spend some time to briefly talk about over-fitting, under-fitting and diagnostic tools.

3. KNN (2): We will invest two lectures to talk about the K-Nearest Neighbor (KNN) technique and its applications. We will also spend some time discussing feature engineering and how the scaling/normalization of features can improve the performance of a ML algorithm. Note that while feature engineering is not exclusive to KNN and can be used with most ML methods, it seemed like this will be a good place to discuss it.

4. Naïve Bayesian (2): A weeklong discussion of the naïve Bayesian method and the probabilistic approach to ML will take place. This is also a good time in the course to introduce Pandas – the Python data analysis library and the dataframe structures.

5. Decision Trees (2): Another week will be invested to cover the use of Decision Trees (DT) as a ML technique. While it is expected that most students have probably seen a DT earlier in their lives, a systematic entropy-based approach to the tree creation will make an interesting discussion. This discussion requires reviewing some of the probability and statistics concepts.

6. Neural Networks (6): This will be the most time-consuming topic in the course. After a brief introduction to Neural Networks (NN), we will dive into the simplest form of NN, namely logistic regression. We will discuss the mathematics of forward propagation and backpropagation and a computer-based implementation of the network. The same process will repeat with a shallow NN and then a deep NN.

7. SL Wrap up (2): We will spend a week discussing the performance evaluation of SL methods. Especially with classification problems, we will introduce confusion matrices and cross-validation methods. We will also talk about tuning hyper-parameters for better results.

8. Unsupervised Learning (3): The last three lectures of the course will cover the less popular, yet important unsupervised learning techniques. This includes K-means clustering, dimensionality reduction, and Principal Component Analysis (PCA).

Computing Platform

While it is theoretically possible to allow students to select their own computing platform of choice, we felt that it would be more practical to adopt one platform with an IDE that is guaranteed to be available on the classroom computers. We also thought that, from a pedagogical perspective, it would be beneficial to be consistent in providing the in-class examples in one programming language. Having said that, we will make it clear for students that we do not limit their options if they choose to use a different language/platform in the course work and course submissions.

The selection of a platform to adopt was no trivial task. The number of available options was very large. Some of these are traditional languages such as C/C++, Java, Python, and R while others are higher level computing packages such as Mathworks' Matlab[4], python-based Tensorflow or PyTorch[5], and Microsoft's Azure[6]. A quick research showed that when it came to ML applications, Python was the most popular among the programming languages as it has the capability to handle and visualize large amounts of data (which is key in ML), and also offers a great set of open source libraries. Matlab is also capable of handling and visualizing large amounts of data with a very nice user interface. The main problem with Matlab is its proprietary nature and the lack of open source libraries. Finally, because of their very high-level nature, we decided to avoid Tensorflow or PyTorch in order to guarantee an in-depth "under the hood" coverage of the discussed material. As a result, we adopted Python as the "official" computing platform of the course.

Textbooks and References

The intention was to use a single textbook that would be required for all the students. Unfortunately, we failed to identify a single textbook that would cover all the topics of interest at the adequate level. Therefore, we used more than one reference[7-10] to cover the various topics. Students are not required to acquire these references but are invited to consider them as resources.

Summary

The curriculum committee in our program recognized the need to introduce a course in machine learning at the undergraduate level. In this paper, we presented the way we are teaching this course with a review of all the encountered difficulties. The suggested course is an application-oriented course that summarizes several ML techniques with an adequate balance of rigor and mathematical depth. The course targets upper level students majoring in electrical engineering, computer engineering, mechanical engineering, and computer science. Python is used as a computational tool in the course but a different language can be easily used if needed.

BIBLIOGRAPHIC INFORMATION

[1] W. S. McCulloch and W. Pitts, Bulletin of Mathematical Biophysics. vol. 5, 1943.
[2] D. J. Norris, *Beginning Artificial Intelligence with the Raspberry Pi*. Springer 2017.
[3] J. McCarthy, M. Minsky, N. Rochester, and C. Shannon, *A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence*. 1955.
[4] https://www.mathworks.com/
[5] https://www.tensorflow.org/
[6] https://azure.microsoft.com/en-us/
[7] E. Alpaydin, Introduction to machine learning — 2nd edition ISBN 978-0-262-01243-0
[8] D. J. Norris, *Beginning Artificial Intelligence with the Raspberry Pi.* ISBN 978-1-4842-2742-8
[9] Burkov, The Hundred-Page Machine Learning Book – Draft
[10] A. Ng, Online courses at https://deeplearning.ai