**Machine Learning Research Proposal**

Christopher Rodriguez and Edwin Gomez

WOS Capstone Project

Dr. Edwards

April 14, 2017

**Significance**

The initial seed for this project arose from the two participants’ desire to learn the principles of machine learning. As such, the project proposes to answer a vital question: how can we best design a project that is structured, rigorous, and effective that will introduce us to the fundamentals of machine learning and provide the foundations for practical application? Being that we are undertaking this project outside a traditional academic setting, the project design must necessarily be innovative and largely self-directed. Furthermore, the obvious tangential question that naturally arises, and is perhaps even more significant from an educational standpoint, is: can a complex field of study such as machine learning be learned successfully by students of an accelerated programming course with little to no prior, formal instruction in Computer Science and Mathematics?

**Background**

Machine learning (ML) applications have become an integral part of our daily lives. Large companies such as Google, Amazon, and Apple have developed tools that utilize ML techniques to solve complex problems. For example, the speech recognition technology behind Siri and Amazon’s Alexa are great examples of ML in everyday life. In addition, Amazon is beta testing a new kind of store that uses ML named AmazonGo which will require no checkout. Customers can grab the items they need and simply walk out of the store. Although machine learning has fairly recently become a popular interest the term has been around for more than fifty years. In 1959, the computer scientist Arthur Samuel defined machine learning as “The field of study that gives computers the ability to learn without being explicitly programmed.” Samuel’s research was important because the central principles of machine learning were verified by his early experiments. A more recent formal definition came in 1997 by Tom Mitchell in which he stated “A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.” Mitchell’s definition gives us a more detailed description of what it means for a computer to learn. Depending on the complexity of the problem ML can be separated into two areas: Supervised and Unsupervised learning.

In supervised learning, a data set is provided and we have a notion of what the correct output should appear like and we are aware that there is a relationship between the input and the output. Supervised learning investigations can be divided into regression and classification problems. In a regression problem, results are being predicted from a continuous output. In essence, input variables are being mapped to a continuous function. In contrast, for a regression problem the input variables are being mapped into discrete categories. In the past, supervised learning problems have included topics such as housing prices and breast cancer prediction.

In unsupervised learning, a data set is given and some structure is found within the given data. The difference here is that we do not have a notion of what the results should look like. This approach can derive structure from the data where the effect of the variables is unknown. The structure is derived by clustering the data based on relationships among the variables.

A supervised learning problem can often be thought of as a regression problem. An important aspect in linear regression is to generate a good output function traditionally called a hypothesis. A hypothesis function will map input variables to predict some output variable. A general form of the hypothesis function is shown below in Figure 1.

Figure 1.

The theta ( components are the parameters for the function that will dictate the shape of it. For every input variable (x) we are trying to map the predicted value. The goal is to generate a straight line through the training data which is represented by .

The accuracy of the hypothesis function can be measured by using a cost function. The cost function takes an average difference of every result from the hypothesis, input variables (x), and the true output values (y). The goal is to get the smallest difference between the predicted value and the true value. The smaller the difference the more confident we can be in our hypothesis function.

A research problem that could benefit from machine learning is the task of classifying atomic force microscopy (AFM) images. AFM is a high resolution imaging technique which allows researchers to gain insight into the external properties of materials. This type of microscopy can be applied to synthetic polymers such as high and low density polyethylene or it could also be applied to biological units such as DNA. It is not uncommon for experts in the field of AFM to debate on the interpretation of the images. There can also be uncertainty in what the images are actually showing, such as a residual complex that might be a result of the scanning probe. By carefully training a ML algorithm to classify AFM images perhaps some insight can be gained from a non-human interpreter.

Machine learning has become a large field with a variety of disciplines for solving complex problems. The ability to process images and to get meaningful predictions from those models will be an area of interest for this project. There are several online resources for implementing machine learning techniques such as the classification of images. Google’s Tensorflow and SciKit are good examples of some open source software libraries that are freely available to use. Also, there is an 11-week online machine learning course by Dr. Andrew Ng at Stanford University.

**Project Design/Methods**

The approach that this project will take will be to survey the Coursera Machine Learning course by Dr. Ng at Stanford. The ML course is 11 weeks long and will cover basic topics in machine learning and will also cover some more advanced topics such as neural networks. The course will also use the scientific programming language Octave. Octave is comparable to MatLab and is free to use. Currently, we have completed the first week’s lecture material and we are currently in the second week. A general course outline is shown below:

Course Outline

Week 1 – Estimated time: 3h 37m

* Introduction
* Linear Regression with One Variable
* Linear Algebra Review

Week 2 – Estimated time: 7h 28m

* Linear Regression with Multiple Valuables
* Octave/Matlab Tutorial

Week 3 – Estimated Time: 6h 2m

* Logistic Regression
* Regularization

Week 4 – Estimated time: 4h 42m

* Neural Networks: Representation

Week 5 – Estimated time: 5h 8m

* Neural Networks Learning

Week 6 – Estimated Time: 6h 7m

* Advice for Applying Machine Learning
* Machine Learning System Design

Week 7 – Estimated time: 4h 57m

* Support Vector Machines

Week 8 – Estimated time: 5h 26m

* Unsupervised Learning
* Dimensionality Reduction

Week 9 – Estimated time: 6h 9m

* Anomaly Detection
* Recommender Systems

Week 10 – Estimated time: 1h 23m

* Large Scale Machine Learning

Week 11 – Estimated time: 1h 16m

* Application Example: Photo OCR

This course can act as a foundation for our capstone to build upon. In addition, the programming language that is primary used in most machine learning tutorials is Python. There are several online tutorials and free books available. It would be prudent to learn some basic Python conditionals, control flow and function syntax. One free resource available is the CodeAcademy Python course. In addition, there is an online book called Scipy Lecture. This online book was written to supply beginners with the information that is needed to use Python for science. SciPy has a chapter that goes through the basics of Python all the way through object-oriented programming in Python. A general outline of the chapter is listed below:

1.2 The Python Language

* First steps
* Basic types
* Control Flow
* Defining functions
* Reusing code: scripts and modules
* Input and Output
* Standard Library
* Exception handling in Python
* Object-oriented programming (OOP)

**Feasibility**

There are several advantages to working with the typical ML technology stack. The development tools and libraries needed for this project are generally open source and readily available. Python, Scikit and its various libraries, TensorFlow, MatLab, and Octave all fall within this realm. An added bonus that this project will provide is a working knowledge of many of the most popular programming languages and tools that are in wide use today in the field of machine learning and beyond.

Of course, any good ML project depends heavily on the quality, and sheer size, of its data. In this respect, the project is well-equipped for optimal results. John Wiley, PhD, the director of UNO’s Advanced Materials Research Lab, has authorized use of all image data from the AFM research currently being conducted at UNO. This data will make up the crux of a possible AFM image processing Machine learning project. Other possible image processing projects include a similar solution analyzing veterinary pathology image data for the presence of image artifacting versus valid biological indicators in conjunction with researchers at the Louisiana State University School of Veterinary Medicine.

As the subject of ML is inherently a complex one, we will undertake a formal introduction to the subject matter through the Coursera course outlined above. This course provides many resources including video instruction, tutorials, lecture notes, test cases, and discussion forums. In addition to these features, we will work closely with Dr. Edwards, our research mentor, to ensure that the project is proceeding on-time and track our progress using the Pivotal Tracker tool.

**Bibliography**

1. *Programming computers to learn from experience should eventually eliminate the need for much of this detailed programming effort"* Samuel, A. L. (1959), “Some Studies in Machine Learning Using the Game of Checkers” in IBM Journal of Research and Development (Volume:3, Issue: 3), p. 210
2. Mitchell, T. (1997). *Machine Learning*. McGraw Hill. p. 2. [ISBN](https://en.wikipedia.org/wiki/International_Standard_Book_Number) [0-07-042807-7](https://en.wikipedia.org/wiki/Special:BookSources/0-07-042807-7).
3. Methods Mol Biol. 2011;783:197-212. doi: 10.1007/978-1-61779-282-3\_11.
4. The Journal of Chemical Physics 112, 7307 (2000); doi: <http://dx.doi.org/10.1063/1.481330>
5. Lyubchenko, Yuri L. “Preparation of DNA and Nucleoprotein Samples for AFM Imaging.” Micron (Oxford, England : 1993) 42.2 (2011): 196–206. PMC. Web. 13 Apr. 2017.
6. Sensors (Basel). 2015 Mar 10;15(3):5865-83. doi: 10.3390/s150305865.
7. "Codecademy - About the Company". Codecademy. Retrieved August 5, 2016
8. SciPy-Lectures. Christopher Burns, et al. 2017. http://scipy-lectures.org