Usable Machine Learning

Preliminary Project Report

Supervised by Rebecca Fiebrink

NIKOLAY NIKOLOV
BSc Creative Computing
Goldsmiths College, University of London
February 19, 2016

1 Introduction

Machine Learning is not a new field inside Computer Science. It has evolved from the study of pattern recognition, statistics and AI and has been studied for over 50 years. Its popularity, however, has soared in the last few years thanks to the large amounts of data accumulated by Google and other big tech giants, increase in computing power and algorithmic innovations such as Deep Learning. As any other new (or rediscovered) technology, it is starting to become useful outside its initial development niche and is being used in a majority of interesting contexts - from speech recognition to cat recognition. At this stage it is primarily used by experts of the field or experienced computer scientists who can translate research papers into code. But to become a more useful or even groundbreaking consumer tool, it needs to be made available to a less technical crowd or to experts in other fields (designers, artists, medical practitioners etc.). In fact, we see some forms of Machine Learning deployed to users in the form of "Siri", "Cortana" or other consumer-facing products but they rarely have any control over the algorithms or training processes - such implicit learning is never understood by users which makes it a rather one-sided relationship. Furthermore, in such closed models of Machine Learning, it is next to impossible to use a tool outside its initial context which forbids any kind of creative use.

Only recently Autodesk announced their future software products will include intelligence that learns about you and your workflow as much as you learn to use their software. This embodies the spirit of future digital tools as the symbiosis of predictive algorithms and human creativity can achieve much more than one or the other by itself. The next step is bringing such tools to ubiquitous computing devices such as smartphones, smartwatches, IoT devices, as well as on available platforms such as the Web itself. In order for them to be usable, they will need to include the appropriate user experience and interaction design. Users need to be able to understand the process of training their own data and get precise intuitive feedback to follow up on errors or provide better data.

Taking inspiration from Rebecca Fiebrink's Wekinator [1] - a piece of software that allows artists to train data coming from multiple sources to control their music instruments or other interactive software, I have decided to apply my past experience and skills in developing web and server applications to prototype an interactive Machine Learning environment where, similarly to Wekinator, people can plug in different data sources and route them to a number of outputs controlling different things. The beauty of the modern web is that the variety of outputs and inputs is enormous and ever growing. From device data such as smartphone motion and orientation data, to the webcam and microphone. The number of APIs that can be fed in and controlled with it is limitless. To create such an environment that makes sense to users and is not too intimidating, I have to provide a gradual learning experience, appropriate UX design and visual, intuitive feedback. While I will strive to provide as much ready-made data sources as possible, I envision an additional component system where people can develop wrappers for the inputs and outputs they want to use but are not provided by default.

This project topic was initially suggested by my supervisor, Rebecca Fiebrink, from which I developed my current vision. My background knowledge required to successfully complete this project (topics such as Audio-visual programming, Machine Learning, Web Development, User Experience Design and User Testing) is based on both courses I took during my time at Goldsmiths as well as outside experience such as industry work and personal projects. Some of the relevant courses I took are Data Mining, Perception And Multimedia Computing, Databases, Networks and the Web, Creative Projects, Advanced Audio-Visual Processing and Advanced Graphics and Animation while my outside experience includes web development, creating data visualisations, 3D graphics on web using WebGL and server scripting using Node.js. Furthermore, my group project from last year involved creating a mobile IDE using web technologies from which I will reuse a lot of skills and experience.

2 Aims and objectives

I aim to create an interactive Machine Learning web environment. Computational creativity has had a long and successful history and after every milestone technological breakthrough, tools, processes, and the way people work has changed radically. I aim to explore how (if at all) Machine Learning methods can have a similar effect when presented in a usable, well-designed form. In order to achieve that I will have to deliver a developed and working web application. Additionally, it would have been tested with potential users and iterated upon based on their feedback. In my report I will outline which processes and workflows were successful and which not. The application should include a working set if inputs and outputs as described in the introduction.

Additionally, there will be interactive data visualisations and explanations to guide the process of training data. They will be refined based on what users find helpful and what increases their understanding of the process.

I will hope to discover some of the challenges when teaching people how and why they would want to build their own classifiers and regressions systems. Hopefully, with each iteration I will have a clearer and better understanding of the needs and desires of my testers which would influence my prototype in the direction of satisfactory usability and generally would result in a useful product, albeit I will not regard the factor of commercial polish to be of great importance, at least not in the scope of this project. If the project is successful and potential users state their intention to use it on a regular basis, I will pursue commercialisation or most likely maintain it as an open-source software package, similar to Wekinator's model.

How I evaluate my project will depend on the quality of the answers to the questions and challenges posed above. The most important part of prototype will be the unique features of interaction between users and Machine Learning algorithms. In order to achieve satisfactory usability at the end of the project, the users must have interacted, understood and practiced a two-way relationship with the tool. Ideally, my piece of software would have been influenced by the user as much as the user by the software.

In summary, my application can be logically divided in two parts: a training/running section and a feedback/data visualisation guidance section. Each of their UX and UI design will be closely tight to the user feedback after each iteration.

3 Methods

I will work on the initial prototype until there is enough content to test with users. After this phase is completed, I will strive to repeat many small iterations of prototyping and testing. This will provide for a tight feedback loop and minimise wasted effort common in "Waterfall"-like methodologies. I will spend most of my working time implementing my application and some additional time reading up on Machine Learning theory or creating mock designs. If in the future my list of tasks and issues grows large and complex, I will resort to using the Github issue system where issues can have tags (bug, feature etc), comments, internal to do lists and so on. A given issue can be closed using a git commit.

I tend to be a disciplined user of revision control, namely git, which will provide for an atomic historical breakdown of my code, logs and designs. I plan on (and have already started) using contemporary methods and frameworks for implementing my web applications. I will most likely make use of D3.js - the most popular data visualisations JavaScript library for the past few years. Furthermore, my whole project will most likely be a React app. React is a modern JavaScript UI library released three years ago. React is currently being used on the homepages of Feedly, Airbnb, Netflix, Imgur, and many others. It's strength lies in the declarative model, separation of components and it's Virtual Dom innovative rendering.

All these things provide for very fast and responsive interactive web applications which is precisely what my project is going to become. While I will spend additional time studying such third-party libraries, they will nonetheless save me huge amounts of effort which would equate to me reinventing the wheel in terms of data visualisations graphics and interactive interfaces.

Server-side scripting and real-time interactions will be handled by Node.js and socket.io, the first being JavaScript in the server and the latter a popular framework for dealing with web sockets. My prototype uses Wekinator to train Machine Learning models but this will be moved to the browser. I will most likely leverage open-source implementations of the same algorithms but this issue requires some more research on my part.

All chosen methods are standard practice in the industry and open-source development.

4 Project Plan

22 - 29 February
- Develop initial prototype to a testable state with a one or two working inputs and outputs each.
- Research on JavaScript Machine Learning implementations I can use.

1 - 13 March
- Work on moving from Wekinator to a browser-only implementation.
- Include another input.
- Write the Introduction chapter and start writing Literature review chapter.
- Organize between 1 and 2 testing sessions.

14 - 27 March
- By the end of this period, I shouldn't be dependant on Wekinator.
- Include another output.
- Write Methods chapter and finish Literature review.
- Draft the rest of the report for submission.
- Organize between 1 and 2 testing sessions.

28 - 10 March/April
- Include another input.
- Write Methods Chapter.
- Organize between 1 and 2 testing sessions.

11 - 24 April
- Include another output.
- Write on the rest of the report.
- Organize between 1 and 2 testing sessions.

25 - 9 April/May

- Include another data input.
- Finish report.
- Organize a final testing session.

9 - 16 May

- Work on remaining bugs and issues on the Software side.
- Read and redact report.

5 Progress To Date

Up until now I developed a few connected but independent working prototypes. As part of my prototyping phase, I leveraged Wekinator. To use make use of that from a web client, I developed a server script that acts as a router from the web page to Wekinator where it receives data from the first source and translates it into an OSC message which it can then send along to the latter. At this point I could successfully train my data using Wekinator so I developed two prototypes in the form JavaScript programs.

The first one was the most basic proof of concept I could start with - send mouse x and y coordinates to Wekinator and teach it a few simple shapes. At first, sending raw data was not working very well so I had to implement some preprocessing techniques, namely moving all shapes to a uniform origin and improve my downsampling method. Once that was working, I moved on to extracting motion and orientation data from my smartphone using Web APIs. Since the implementation of this was largely based on the mouse example, a working version was relatively easy to develop.

Having two working prototypes, I spent some time training and testing. It took some effort to get a kind of intuitive feel for what works and what does not. Lastly, I started adding basic UI features to the mobile interface of the prototype. I included a three graphs each showing the X, y and Z of the device motion data. While not visually intuitive, they provide for a rudimentary way to compare gestures. I have also separated the process of recording a gesture and sending the data over the wire into two buttons. What this means is that you can record a sophisticated gesture by holding the record button, then letting go and picking up from another place. For example, you could train Wekinator to recognize the "X" shape.

6 Planned Work

These past few weeks were spent on prototyping and exploring what is possible with web technologies and Wekinator. I am happy with the fact that I built a working proof of concept. As the Project Plan mentions, I will try to finish up a testable version of my prototype that users that try out and collect feedback for the first iteration. Shortly after that, I will start writing the final report.

References

[1] Rebecca Fiebrink "Real-time Interaction with Supervised Learning" (ACM CHI, 2010)
`http://www.cs.princeton.edu/~fiebrink/publications/Fiebrink_CHI2010.pdf`