Experiences from Teaching Actionable Machine Learning at the University Level through a Small Practicum Approach

Natalie LAO[1*], Irene LEE[2], Hal ABELSON[3]
[1,2,3] Massachusetts Institute of Technology, United States of America
natalie@mit.edu, ialee@mit.edu, hal@mit.edu

ABSTRACT

Machine learning (ML) courses have traditionally been taught through a math-first approach. They generally begin by establishing mathematical theories behind ML, such as the perceptron algorithm, logistic regression, and backpropagation, and then use these building blocks to motivate more complex structures such as neural networks. Such educational resources may not be sufficient or preferable for audiences who wish to use ML to build useful artifacts but do not have a strong mathematical or programming background. In this paper, we introduce a new framework for teaching actionable ML that combines three components in a Use-Modify-Create progression: (1) technical modules taught through hands-on labs, (2) a capstone project, and (3) supplemental lectures for new areas of research. This framework was applied in two iterations of a semester-long practicum at Massachusetts Institute of Technology (MIT) as a beginner-accessible course aimed at helping a broad range of students gain the ability to ideate and implement independent ML projects. We present the curriculum, student projects, pre and post-course survey responses, assignment grades, reflective discussions, and learnings from both iterations of the course. Our results indicate that the proposed actionable pedagogical framework for ML along with the content and practices of the course were effective for increasing students' practical self-efficacy in ML and computational identity as developers of ML applications. The findings of this study illuminate patterns of interaction with ML systems that support a practical approach to teaching ML in order to increase knowledge acquisition, future learning ability, and motivation in beginner students.

KEYWORDS

machine learning, deep learning, actionable pedagogical framework, experiential learning, small practicum

1. INTRODUCTION

As artificial intelligence (AI) and machine learning (ML) have gained prevalence in public education over the past decade, many interpretations of the two terms have been presented. We define ML as models trained on large amounts of data to inductively find patterns while AI also includes algorithms crafted from general deductive principles to solve specific problems (e.g. alpha-beta pruning and minimax); and deep learning as a subtopic of ML that uses neural networks with more than one hidden layer (Lao, Lee, & Abelson, 2019). In order for an ML system to "work", it is dependent on the availability of high quality data, scientific insights on features, appropriate model architectures, and computational processing power. Instead of being generally deterministic programs, ML results in powerful statistical algorithms that can be hard to debug and understand on a detailed level, such as when analyzing a single error in a large neural network (Shapiro, Fiebrink, & Norvig, 2018). Therefore, it may not be the most effective to teach ML in the style of other algorithms courses if we want to educate critical thinkers from a wide variety of backgrounds. In designing our teaching framework, our questions were:

- Can students with no/minimal ML or CS experience quickly apply ML to interesting and suitable problems without being explicitly taught the underlying mathematical theories?
- What human and computational resources are needed for an introductory, projects-based ML course?

This paper serves as an experience report that describes the pedagogical learnings from designing and implementing a small-scale, project-focused practicum that was successful at helping students of various technical backgrounds develop self-efficacy as machine learning project creators (Lao, Lee, & Abelson, 2019).

2. BACKGROUND

2.1. Theoretical vs. Practical Approaches

Most current ML courses teach the mathematics of ML during lessons (e.g. the perceptron algorithm or linear regression), and ask students to work on proofs or math-heavy problems for homework, which may involve translating the relevant math into code (Dror & Ng, 2018; see also Mohri, 2018). However, such methodologies may not work well for students who do not yet have a strong foundation in probability, calculus, or linear algebra. In contrast, practicums are often run as laboratory classes where students work on assignments and projects during class time with the support of mentors and/or teaching assistants. This Deep Learning Practicum course is an example of an ML *practicum* targeted towards university students of a broad range of backgrounds that takes a *practical* approach—its focus is on the "doing and use of ML" and the creation of personal projects and applications.

2.2. Experiential Learning: Use-Modify-Create

In experiential learning, or "learning through reflection and doing" (Kolb, 2014), learning can be elicited through direct manipulation of objects or systems as "objects to think with" (Papert, 1980). In our course design, experiential learning exercises are combined with a capstone project through the Use-Modify-Create Progression (Lee et al., 2011). We posit that Use-Modify-Create can help students deepen understanding of ML concepts and master practical skills: (1) students use ready-made ML models within fastresponse and user-friendly interfaces to develop high-level intuitions about training, testing, and the importance of data, (2) students manipulate the models directly to understand how architectural modules, hyperparameters, and datasets impact results for different problems, (3) students scope a problem suitable for ML and create their own application.

This methodology can now be applied to ML due to the creation of libraries that support in-browser ML experiments such as TensorFlow.js (Smilkov et al., 2019), and user-friendly applications such as Teachable Machine (Stoj.io et al, 2018) and ModelBuilder (Google, 2018). These tools allow students to modify powerful ML models (either through a user interface or code) and test the results in real-time, which enable novices to quickly gain experience through direct manipulation of ML systems. Students can quickly iterate through building a model, inputting data, training the model, and analyzing results. Furthermore, these "laboratory" experiences with ML systems provide students with experiences that directly relate to the future of work at the human-machine frontier.

2.3. Self-Efficacy and Engagement

This practicum's framework for teaching actionable ML incorporates several mechanisms for engagement: empowering students, creating meaningful experiences through scaffolded, inquiry-based learning, and authentic learning opportunities (Wu & Huang, 2007). The framework also emphasizes self-efficacy, a belief in one's chances of successfully accomplishing a task and producing a favorable outcome (Bandura, 1977). Students with high self-efficacy develop deeper interests in the tasks at hand and are more motivated to learn challenging material (Bandura et al., 2001). Recent work shows that self-efficacy is developed and strengthened through seeing others like themselves succeed, being persuaded by respected friends and advisors, and reflecting on one's own capabilities (Bandura, 2004). As such, our framework was designed to emphasize collaborative work, work with near-peer mentors, and exposure to ML professionals of diverse demographics (gender, age, and race/ethnicity).

3. INSTRUMENTS AND ASSESSMENT

The data sources used to analyze the course's impact included anonymous responses to pre and post surveys and analysis of capstone projects. At the time of the study, there were no validated instruments for measuring self-efficacy in ML. We created our post survey instrument based on validated instruments for measuring self-efficacy in general sciences, including Children's Science Curiosity Scale (Harty & Beall, 1984) and Modified Attitudes Towards Science Inventory (Weinburgh & Steele, 2000).

4. DEEP LEARNING PRACTICUM V1

The first version of the course ran for 1.5 hours 2x a week over a 15-week semester in spring 2018 at Massachusetts Institute of Technology (MIT). The course did not count towards core undergraduate requirements and was an elective course. In pre-registration, the instructors emphasized that the course was meant for students who did not feel comfortable working with ML and not experts hoping to gain advanced techniques. Class size was restricted due to the personalized, project-based nature of instruction. Twelve students completed the course.

During the course, instructors aimed to ground theoretical constructs of ML in hands-on applications that spanned different topics. Six genres were covered in the pilot that included predictive and generative applications of ML. The order of genres followed the historical development of ML, and naturally presented a progression in the sophistication of ML models. There were 3 starter topics followed by 3 advanced topics. The instructors gave short explanatory technical lectures (<15 min.) with in-class exercises in TensorFlow.js that students ran on their own laptops. The activities often leveraged existing datasets, pre-built models, and web-based tools for ML. For each set of exercises, students were asked to discuss their findings with a partner or with the class. Student teaching staff provided technical and instructional support. Weekly take-home assignments provided an extension to the environment and the exercises introduced during class.

The last 9 weeks of class focused on capstone projects and guest lectures (GLs) from ML professionals and researchers. Students chose a problem that personally interested them and was suitable for an ML application. Mentors were paired to each project. A week-by-week map of the version 1 curriculum is presented in Table 1.

Table 1. V1 of the curriculum annotated with the ITEST Use-Modify-Create progression per week.

wk.	Topics	Progression
1	K-Nearest Neighbors	**Use:** Teachable Machine webapp (Stoj.io et al, 2018). **Modify:** Confidence algorithms in source code.
2	Multilayer Networks	**Use:** Model Builder webapp (Google, 2018). **Modify:** Starter TensorFlow.js and HTML code for programming multilayer networks.
3	Convolutional Neural Networks (CNNs)	**Use:** Model Builder webapp, filter visualization webapp (Harley, 2015), Fast Style Transfer webapp (Nakano, 2018). **Modify:** Starter code for programming CNNs.
4	Generative Models and Embeddings	**Use:** Embedding Projector webapp (TensorFlow, 2018), Latent Space Explorer (deeplearn.js., 2018). **Modify:** Feature projection functions in Latent Space Explorer source code.
5	Generative Adversarial Networks (GANs)	**Use:** GAN Playground webapp (Nakano, 2017). **Modify:** Starter TensorFlow.js and HTML code for programming GANs.
6	Recurrent Neural Networks (RNNs) and	**Use:** RNN text generation webapp (Karpathy, 2015), SketchRNN webapp (Ha, Jongejan, & Johnson, 2019).

	Long Short-Term Memory (LSTMs)	**Modify:** Architecture and parameters in webapp source code. **Create:** Music generation RNN application through TensorFlow Python notebook.
7	Project Overview	**Create:** Students scoped and presented 3 project ideas.
8	Spring Break	**Create:** Teams/individuals worked on the project proposal writeup.
9	Project Mentor Matching, GL: Healthcare	**Create:** Teams submitted proposals and were matched with industry mentors.
10	GL: Fairness, GL: Testing and Training Tools	**Create:** Teams work on projects.
11	GL: Interpretability	**Create:** Teams work on projects.
12	Project Midpoint Checkpoint Presentations, GL: Art & Music	**Create:** Teams presented 5-minute project progress reports in class, received feedback.
13	GL: People + AI Research	**Create:** Work on projects.
14	Final Presentation Dress Rehearsal, GL: Adversarial Examples	**Create:** Teams presented a practice run of their final 10-minute project presentations in class, received feedback.
15	Final Presentation Showcase, Project Writeup Due	**Create:** Present projects in front of industry professionals and submit project writeups in the form of instructional blog posts.

4.1. Student Demographics

Of the 12 students, there were 2 (17%) second-years, 4 (33%) third-years, 5 (42%) fourth-years, and 1 (8%) graduate student. Nine (75%) majored in EE/CS, 1 in Math, 1 in Math & Physics, and 1 in Humanities. There were 3 black women, 3 Asian men, 2 Asian women, 2 white women, and 1 white man. Ten students (83%) had basic exposure to AI or ML, but wrote in the pre- survey that they wanted to take another introductory course because they did not feel that they could build practical applications. All students had at least some coding experience, but only 8 (67%) had experience in JavaScript.

4.2. Teaching Staff and Industry Mentors

There were 6 student staffers who helped debug in-class exercises for each topic, answer questions, and lead reflective discussions that directed towards learning goals for the exercises. For the 9 projects in the class, 3 of the staff mentored 1 project each and 3 mentored 2 projects each. There were 9 industry mentors. We reached out to companies and researchers in the area to ask for volunteers who have experience with ML projects. We invited all volunteers to a mixer with the students after project teams had formed. At the beginning of the mixer, each mentor gave a brief overview of their expertise and each student team summarized their project goals. After the mixer, student teams submitted their preferences for mentors and were matched. Mentors met with teams during the beginning and middle of their project cycles to help with high level ideas, resources, and project scoping.

4.3. Capstone Projects

Within this “Create” stage of the course, students marshaled the tools and techniques at their disposal along with mentorship to create capstone projects. Students were instructed to choose a project that they were personally interested in, but were also cautioned that *a realistic project implemented well and evaluated thoroughly is better than a half-implemented ambitious project with no result*. Projects could be a real-world **Application** of ML, an **Exploration** of properties of neural networks, or a **Replication** of an ML paper. To scaffold project scoping, students were given a “3 Ideas” assignment in which they presented 3 project ideas in class. For each idea, students defined a “Safe” goal that they were confident they could achieve by the end of the semester, a “Target” goal that they hoped to achieve, and a “Stretch” goal that would be good to achieve if extra time was available.

Students had the option of finding a project partner after the presentations. There were 9 projects consisting of 3 pair projects and 6 solo projects. 7 projects were in the Application category, 1 in Replication, and 1 in both Application and Exploration. All teams achieved their Safe goals. One team continued working on their project after the class ended and was able to publish a paper.

4.4. Learnings for V2

Feedback was obtained through surveys and a discussion-style post-mortem on the last day of class. Due to the small class size, quantitative analyses are not presented to preserve anonymity. Overall, students loved the interactive lab style of the modules in the class. Two students with no prior JavaScript experience felt that the course was surprisingly JavaScript-independent, although some coding experience was helpful. Students felt that the small class size was beneficial in creating an environment that made them feel comfortable speaking during the open reflective discussions that accompanied in-class exercises. Nearly every student felt that there was not enough time for project implementation, but students also said that it was the most valuable and enjoyable part of the course. Students suggested that the course should cover data collection, data processing, and using external computational resources to better scaffold the projects. Students enjoyed the guest lectures and thought that they helped “*put what we learned into a much bigger picture.*” Students noted that some guest lectures may have been useful before starting their final projects and would have provided additional context for project choices.

Several students said that the course demystified ML and made it more approachable. Two students mentioned their increased concern over bias in ML algorithms as well as a deeper understanding of how to resolve some of these issues: "*Before this course, I thought of computer programs more linearly – as if [programmers] were mostly in control of a program's results. Now I have a much greater understanding of how ML programs can be biased and unfair... I learned the importance of providing good, varied input data and how this data can have significant impact on a network and ultimately the world.*"

5. DEEP LEARNING PRACTICUM V2

5.1. Changes from V1

The second version of Deep Learning Practicum was offered at MIT in fall 2018, the semester following the pilot. There were 6 main changes from version 1: (1) the final project was introduced at the beginning of the semester and ran in parallel to the modules portion of the course, (2) there were two additional scaffolding workshops for the final project (data mining and using computing clusters), (3) students were required to have a partner for their project unless given permission, (4) guest lectures were more interspersed throughout the course instead of all at the end, (5) the staff-to-student ratio decreased from 6:12 to 6:26, and (6) an additional unit on reinforcement learning was added. The full set of version 2 curricula, lectures, assignments, and final projects can be found online at http://mit.edu/6.s198 (Lao & Abelson, 2018). A weekly summary of the version 2 curriculum is presented in Table 2 (Lao, Lee, & Abelson, 2019).

Table 2. V2 of the curriculum annotated with the ITEST Use-Modify-Create progression per week.

wk.	Topics	Progression
1	K-Nearest Neighbors, Transfer Learning	Module from version 1 wk. 1 with more emphasis on transfer learning techniques.
2	Multilayer Networks, Project Overview	Module from v1 wk2. **Create:** Scope 3 ideas for capstone final project.
3	CNNs, GL: Adversarial Attacks	Module from v1 wk3. **Modify:** Starter adversarial attack TensorFlow code on CNNs.
4	3 Ideas Project Workshop, Data Mining Workshop	**Use:** Kaggle to find datasets (Kaggle Inc., 2019). **Modify:** 3 project ideas based on feedback. **Create:** Web scraping scripts using Beautiful Soup (Python Software Foundation, 2019).
5	Generative Models and Embeddings, Computational Resources workshop, Project Mentor Matching	Module from v1 wk4. **Use:** Holyoke Computing Cluster tutorial (MGHPCC, 2018).
6	GANs	Module from v1 wk5.
7	Project Data Review, Reinforcement Learning	**Use:** Metacar webapp (thibo73800, 2019), OpenAI Gym webapps (OpenAI, 2019). **Modify:** TensorFlow starter code for RL. **Create:** Data review document to describe project dataset details.
8	RNNs and LSTMs	Module from v1 wk6.
9	Informal Project Checkpoint	**Create:** Work on projects and discuss progress with staff.
10	GL: Art & Music, GL: People + AI Research	**Create:** Finish project proposal.
11	Formal Project Checkpoint	**Create:** Work on projects and show basic working demo to staff.
12	GL: Healthcare, Project Practice Lightning Talks	**Create:** Present 2-minute project lightning talks, receive feedback for final showcase.
13	Office Hours	**Create:** Work on projects.
14	Final Presentation Showcase	**Create:** Presented capstone projects to an audience of varying ML experience with lightning talks, then individual booths.
15	Project Writeup Due	**Create:** Submit project writeups in the form of instructional blog posts.

5.2. Student Demographics

Of the 26 students, there were 5 (19%) second-years, 6 (23%) third-years, 11 (42%) fourth-years, 3 (12%) graduate students, and 1 (4%) post-doc. Twenty students (77%) majored in EE/CS, 2 in Architecture, 2 in Physics & EECS, 1 in Materials Science & Engineering, and 1 in Biological Engineering & Math. There were 9 Asian women, 6 Asian men, 5 white men, 3 Hispanic men, 1 black woman, 1 white woman, and 1 black man. Similar to V1, 21 students (81%) had basic exposure to AI or ML, but commented that they wanted to participate in the course due to self-perceived lack of ability to apply theory and math in building practical applications.

5.3. Capstone Projects

For V2 of the course, students were asked to work in groups of two for the final project due to the decrease in the staff-student ratio. Students started work on the projects in wk. 2 of the course, so they had not been exposed to all of the topic modules. The instructors were concerned that students may avoid later topics and tried to mediate this by giving lightning talks and sample use cases for the topics that would be presented later. There were 14 projects, all of which completed their Safe goal. All three project categories were represented with the majority being Application projects. More projects bridged multiple project categories than in V1, likely due to students having more time.

The first project workshop was the 3 Ideas Workshop during wk. 4, which changed in format from the pilot due to the increased number of students: The staff ran two 30-

minute sessions of guided group presentations. For each session, the class was divided into four groups of 5-6 students based on shared project topic interests. One to two staffers led each group, where students took turns presenting their 3 ideas. During the final 30 minutes of the class, students were encouraged to talk to others they had met and form groups. After the 3 Ideas Workshop, there were two workshops given on project skills: data collection and how to connect to computing resources. There was also a data review checkpoint assignment due in wk. 7 to confirm that students had completed data collection and processing in a timely manner.

5.4. Post Survey Responses

The post survey was emailed out after the course ended and received 17 responses (65%). Demographic results indicated that the participants were representative of the class in terms of grade level, major, gender, and ethnicity. Table 4 presents the responses to all linear scale questions, where 5 = "Strongly agree" or "Completely confident" and 1 = "Strongly disagree" or "No confidence."

Table 4. Means and standard deviations of post survey linear scale question responses.

	Item	Mean	S.D.
1	I felt that I was successful in this class.	4.4	0.6
2	I am proud of what I was able to accomplish in my final project.	4.2	0.8
3	I will be able to complete an ML project (of a similar level and scale to my final project) on my own.	4.6	0.5
4	In this class, I saw people like me succeed at learning ML	4.2	0.6
5	When I saw people like me succeed in ML, it made me feel that I could succeed as well.	4.3	0.7
6	How confident do you feel about describing your project to a non-technical person?	4.6	0.5
7	The project work made me feel uncomfortable	1.6	0.9

As a follow-up to Question 1, we asked "What did you use to determine your sense of success in the class?". The majority of responders attributed their sense of success to *work on the final project* (94%) and *understanding of the concepts presented in class* (88%). Responses to "Which of the following elements from the course did you use in your project work?" also indicated that the modules and workshops were helpful. More than half of responders said they *used concepts/architectures from the units* (82%), *used independent researching skills [developed] through the assignments* (59%), or *used the 3 Ideas Workshop to [help] improve or refine [their] project idea* (59%).

When asked "How can you see yourself using the tools, techniques, and methods presented in the class?", all responders gave multiple use cases. The most prevalent were: *Applying ML to new domains* (82%); *Be(ing) able to talk about it with experts* (77%); *Being able to talk about it with non-experts* (77%); *Using it for fun* (65%); *Developing my final project further* (65%); *Using it [for] another class* (65%); and *Using it as part of a job* (65%).

When asked "How did your views on ML change through taking this course?", 53% mentioned a "*personal realization of the easy application potential of ML*"; 18% had "*increased enjoyment of the field*"; 18% wrote "*realizing limitations of ML*"; and 12% were "*excited…the field is rapidly evolving*".

6. DISCUSSION

The course aimed to help students with some coding background and none to novice AI or ML knowledge gain self-efficacy in ML. In general, students highly enjoyed the course, felt that it helped demystify ML, and were encouraged to pursue independent, personal ML projects in the future. We felt that both iterations of the course were successful in our goals, with V2 allowing students more time for projects. Survey responses from V2 indicate that successful completion of capstone projects most heavily influenced development of self-efficacy in ML, and that the modules portion of the course was successful at preparing students for the projects. While our results are promising, we recognize limitations to replication: there was a high staff-to-student ratio and many students had exposure to ML/AI before the course (although we found no significant difference in performance between students of varying levels of ML and coding backgrounds).

We believe that the following 4 components of the course best contributed to its success: First, while the modules did not teach all of the skills and concepts students needed for every type of ML project, we hypothesize that the hands-on, exploratory lab work for each application helped students feel more comfortable playing with new architectures. This encouraged students to conduct research and learn on their own – 3 teams from V2 (21%) even applied methods not taught in the class to their projects. Second, TensorFlow.js allowed beginners to dive directly into exploring complex and visually appealing ML applications – modifying ML models in the browser allowed for near-instantaneous feedback and reduced infrastructure problems. Third, mentors for each project greatly assisted students in properly scoping problems and finding resources. Fourth, the blog post style for the final project writeup helped students learn disciplinary sharing norms, situate their work in the community, and identify with other ML developers, enthusiasts, and researchers. Thirteen of the 14 projects from V2 shared their project blog posts publicly on the web (Lao & Abelson, 2018).

In replicating this course, the advanced modules (wk. 4+) can be replaced based on the types of projects instructors want to encourage (e.g. more types of RNNs, LSTMs, and GANs for an arts-focused ML class). Additionally, we found that transfer learning was extremely useful – students were able to adjust and retrain high quality pre-built models with great results for repurposed use instead of spending a long time trying to create (often ineffective) models from scratch. We recommend encouraging students to research and experiment with different architectures as often as possible. Many of the students without coding experience also suggested that a version of the course could be adapted for high school students.

7. REFERENCES

Bandura, A. (1977). Self-Efficacy: Toward a Unifying Theory of Behavioral Change. *Psychological Review*, *84*(2), 191–215.

Bandura, A. (2004). Health Promotion by Social Cognitive Means. *Health Education Behavior: The Official Publication of the Society for Public Health Education*, *31*(2), 143–164.

Bandura, A., Barbaranelli, C., Caprara, G. V., & Pastorelli, C. (2001). Self-Efficacy Beliefs as Shapers of Children’s Aspirations and Career Trajectories. *Child Development*, *72*(1), 187-206.

Deeplearn.JS. (2018). *Font-Explorer*. Retrieved November 18, 2019, from https://github.com/mintingle/font-explorer

Dror, R., & Ng, A. (2018). *CS229: Machine Learning*. Retrieved November 18, 2019, from http://cs229.stanford.edu/syllabus.html

Ha, D., Jongejan, J., & Johnson, I. (2019). *Draw Together with a Neural Network*. Retrieved November 18, 2019, from https://magenta.tensorflow.org/sketch-rnn-demo

Harley, A. (2015). *3D Visualization of a Convolutional Neural Network*. Retrieved November 18, 2019, from http://scs.ryerson.ca/~aharley/vis/conv/

Harty, H., & Beall, D. (1984). Toward the Development of a Children’s Science Curiosity Measure. *Journal of Research in Science Teaching*, *21*(4). 425-436.

Kaggle Inc. (2019). *Kaggle: Your Home for Data Science*. Retrieved November 18, 2019, from https://www.kaggle.com/

Google. (2018). *People + AI Research Initiative: Deeplearn.JS Model Builder Demo*. Retrieved November 18, 2019, from http://courses.csail.mit.edu/6.s198/spring-2018/model-builder/src/model-builder/

Karpathy, A. (2015). *RecurrentJS Sentence Memorization Demo*. Retrieved November 18, 2019, from https://cs.stanford.edu/people/karpathy/recurrentjs/

Kolb, D. A. (2014). *Experiential learning: Experience as the source of learning and development*. Pearson FT Press, New Jersey. Print.

Lao, N., & Abelson, H. (2018). *6.S198: Deep Learning Practicum, Fall 2018*. Retrieved November 18, 2019, from http://mit.edu/6.s198

Lao, N., Lee, I., & Abelson, H. (2019). A Deep Learning Practicum: Concepts and Practices for Teaching Actionable Machine Learning. *12th Annual International Conference of Education, Research and Innovation (ICERI’19)*, 10. International Academy of Technology, Education and Development (IATED).

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., & Werner, L. (2011). Computational Thinking for Youth in Practice. *ACM Inroads*, *2*(1), 32-37.

The MGHPCC. (2018). *The Massachusetts Green High Performance Computing Center*. Retrieved November 18, 2019, from http://www.mghpcc.org

Mohri, M. (2018). *Foundations of Machine Learning – CSCI-GA.2566-001*. Retrieved November 18, 2019, from https://cs.nyu.edu/~mohri/ml18/

Python Software Foundation. (2019). *Beautiful Soup, version 4.8.1*. Retrieved November 18, 2019, from https://pypi.org/project/beautifulsoup4/

Nakano, R. (2017). *GAN Playground – Explore Generative Adversarial Nets in your Browser*. Retrieved November 18, 2019, from https://reiinakano.github.io/gan-playground/

Nakano, R. (2018). *Deeplearn.JS Style Transfer Demo*. Retrieved November 18, 2019, from https://reiinakano.github.io/fast-style-transfer-deeplearnjs/

OpenAI. (2019). *Gym*. Retrieved November 18, 2019, from https://gym.openai.com/

Papert, S. (1980). *Mindstorms: Children, computers and powerful ideas*. Basic Books, New York. Print.

Shapiro, R. B., Fiebrink, R., & Norvig, P. (2018). How Machine Learning Impacts the Undergraduate Computing Curriculum. *Communications of the ACM*, *61*(11), 27-29.

Smilkov, D., Thorat, N., Assogba, Y., Yuan, A., Kreeger, N., Yu, P., Zhang, K., Cai, S., Nielsen, E., Soergel, D., Bileschi, S., Terry, M., Nicholson, C., Gupta, S. N., Sirajuddin, S., Sculley, D., Monga, R., Corrado, G., Viégas, F. B., & Wattenberg, M. (2019). TensorFlow.js: Machine Learning for the Web and Beyond. *CoRR*.

Stoj.io, Use All Five, Creative Lab, and PAIR team at Google. (2018). *Teachable Machine*. Retrieved November 18, 2019, from https://teachablemachine.withgoogle.com/v1/

TensorFlow. (2018). *Embedding projector – visualization of high-dimensional data*. Retrieved November 18, 2019, from http://projector.tensorflow.org/

thibo73800. (2019). *Metacar: A reinforcement learning environment for self-driving cars in the browser*. Retrieved November 18, 2019, from https://www.metacar-project.com/

Weinburgh, M. H., & Steele, D. (2000). The Modified Attitudes Toward Science Inventory: Developing an Instrument to Be Used with Fifth Grade Urban Students. *Journal of Women and Minorities in Science and Engineering*, *6*(1). 87-94.

Wu, H. K., & Huang, Y. L. (2007). Ninth-Grade Student Engagement in Teacher-Centered and Student-Centered Technology-Enhanced Learning Environments. *Science Education*, *91*(5), 727-749.