Andrew Garrett

Dr. Richard Di Rocco

PSYC-549: A Neuroscience Perspective of Artificial Intelligence

May 11th, 2021

A Brief Introduction to Machine Learning and Neural Networks

I. Introduction

Machine Learning - The application of Artificial Intelligence (AI) which focuses on giving computer systems the ability to automatically learn and improve their ability to complete some task without explicit instructions (*Expert.ai*). Machine Learning (ML) is a beautiful and fascinating area of Computer and Data Sciences which can be used in a variety of fields and industries. The general goal of ML is essentially to automate and generalize processes which have traditionally been accomplished using explicit instructions infused with user interpretation and analysis. In a world that is constantly changing and progressing, ML offers solutions not only to problems that have historically been considered intractable, but also to problems that are emerging on a continuous basis. As data becomes increasingly available and prevalent, ML will continue to grow in significance, and as such it is imperative to have a fundamental understanding of what it is, how it works, and what it does for the society around us.

II. Historical Context

It is perhaps most natural to first give some historical background and contextualization of the field of Machine Learning. The origins of ML come much earlier than most might expect, which gives some credence to the fact that it remains a fairly elusive concept to the general public. The first ever computing device was conceived by Charles Babbage in 1834. A far cry from the modern computer, his theoretical device used punch-card transcribed mathematical expressions to perform arithmetic operations. Although his concept never came to fruition, the

underlying logic for his machine remains the foundation for which computers are designed today (*Google Cloud*). Nearly 100 years later, Alan Turing, who is considered to be the father of computer science, developed the first mathematical model of computation known as a "Turing Machine," (*Google Cloud*). This "Turing Machine" served as a platform and framework for which all fundamental problems of computer science could be modeled and solved. Not long after, Turing formulated one of the most important questions that Machine Learning is based upon, which is dubbed "The Turing Test," (*Forbes*). This test proposes the following idea: If a machine can convince a human that it is also a human, then is the machine a true intelligence? The near-philosophical Turing Test is often considered to be the birth of Artificial Intelligence. In 1952, scientists at IBM developed the first algorithm which would learn and improve its performance in the game of checkers (*Google Cloud*). And thus, within the same two years, the fields of AI and ML were realized from both theoretical and application perspectives.

Throughout the mid to late twentieth century, computers and machine learning developed from obscurity into hot topics in not only the academic community but also pop culture. In 1968, one of Stanley Kubrick's cornerstone works of film "*2001: A Space Odyssey*" captured the attention of audiences around the world with its dark yet logical take on the interaction between humans and an Artificial Intelligence known as "HAL" (*Google Cloud*). In fact, after the film was released, Kubrick consulted computer scientists at MIT who believed that the depiction of AI in the film was a very real possibility by the year 2001. With the increasing depiction of artificial intelligences in films such as "*The Terminator*" (1984), the research community became just as enthralled by this mysterious field. One such early advancement came from a 1986 paper from Johns Hopkins University, where scientists developed a computational architecture called NETtalk, which could learn to pronounce words in a manner similar to a human child; in the breakthrough discovery, though their learning model began with no knowledge of pronunciation,

after of a single week of being trained on spoken English words it was able to understandably articulate over 20,000 words (*Sejnowski and Rosenberg*). Another notable event which caught the attention of worldwide audiences was the 1997 chess match between IBM's *Deep Blue* and the current World Chess Champion, Grandmaster Garry Kasparov. By defeating Kasparov, *Deep Blue* became the first computer to hold the title of the best chess player in the world, despite much upheaval from the likes of Kasparov, who was very dissatisfied by the result (*Forbes*). As one of the last great accomplishments of the century, in 1998 AT&T Bell Labs developed the first neural network which was used to interpret and classify handwritten digits with a relatively high degree of accuracy (*Google Cloud*). One scientist in this group, Yann Lecun, would go on to pioneer the field of computer vision and perception in Deep Learning.

The transition into the modern era of ML can mostly be attributed to the work of Geoffrey Hinton, a professor and researcher in AI at the University of Toronto. In a 2006 paper of his, Hinton effectively rebranded Neural Network research as "Deep Learning research," with the goal of attracting interest and funding (*Hinton, Oscindero, Teh*). This change seemed to trigger an explosion of research and fascination with the groundbreaking field of Deep Learning. Since then, technological giants such as Google and IBM have funded and directed massive amounts of research and integration into their businesses. Notably in 2011, IBM *Watson* was trained and prepared for a game of *Jeopardy* against some of the best champions in the game show's history, where *Watson* emerged victorious without the aid of any external input (*Google Cloud*). Similarly in 2016, Google's flagship research project *AlphaGo* defeated reigning World Go champion Lee Sedol four to one in a five game series. This was incredibly significant not only due to the revolutionary nature of *AlphaGo*'s architecture, but also due to the fact that Go is widely regarded as the most complex and difficult board game in the world (*Medium*). Since this historic win, the *AlphaGo* architecture has spawned more advanced and even stronger algorithms

such as *AlphaZero* and *MuZero*, both of which learn faster than their predecessor and have been generalized to perform at the same level across almost any "simple objective" game in the world. This brings us to the modern-day, where Machine Learning and Deep Learning are both becoming dominant technologies in fields such as finance, consumer products, and marketing.

III. Classes, Tasks, and Practicality of Machine Learning

Now that we have developed a historical, societal, and developmental scope of Machine Learning, it is reasonable to examine more closely its nuances and applications. As an overview, there are three main classes of Machine Learning: Supervised Learning, Unsupervised Learning, and Reinforcement Learning. Each of these have unique and encompassing definitions, as well as their own specialized algorithmic tasks. On top of this, these algorithmic tasks can be optimized for practical applications and use cases. Organizing Machine Learning makes it easier for both experts and novices alike to determine how best to solve a data centric problem.

A. Supervised Learning

Supervised Learning is a class of Machine Learning which is defined by the relevant data being labelled. This means that in Supervised Learning, there is a designated set of input data which has some label or output, and once a model has been trained using these labelled inputs, it can predict what the label of new, unseen data will be. In the process of training through Supervised Learning, the model predicts the appropriate output for some input data, and then evaluates its own performance by comparing predicted labels to the given correct labels. Using methods of approximation which are too complex for the scope of this discussion, theory has proven that an optimal model, which can perfectly predict the labels of the training data, can be achieved. Although this may seem like a powerful result, there can be unintended consequences, most notably in overfitting to training data, which results in poor performance on new data.

Under the class of Supervised Learning, there are two primary algorithmic tasks: Classification and Regression. Both of these are fairly self-descriptive, but to properly enunciate their difference, classification tasks focus on assigning an ordinal or nominal label to new data while regression analysis tries to develop some trend or relationship between input features and a discrete or continuous label. In the context of classification, an illustrative example is a manufacturing line which is used to sort lego bricks by color. A classifier could be trained using a database of pictures of lego bricks which have been labelled according to their color. The trained classifier could then be deployed with a camera above the manufacturing line, where the classifier uses the incoming camera data of colored lego bricks for automated sorting. Some of most powerful algorithms which have been designed for this task include k-Nearest Neighbors, Decision Trees, Support Vector Machines, Adaboost, and Random Forest. The first three of these algorithms are relatively simple, baseline classification methods whose goal is to find some linear or non-linear separation to the data.

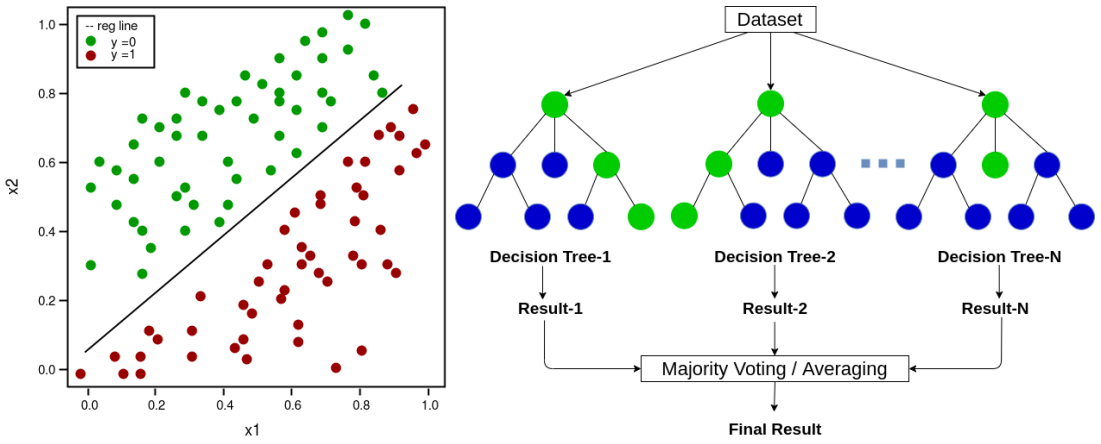

Figure 1: (left) The resulting linear separation of binary labeled data using an algorithm like a Support Vector Machine; (right) An outline of the general structure of the Random Forest classification algorithm

At the risk of oversimplification, regression analysis can be boiled down to the simple idea of finding the line or curve of best fit between a feature or variable and the target label. This is why regression analysis is sometimes called "fitting". Regression has found repeated and heavy use in finance and marketing to find relationships or trends between variables such as annual cost of operation and daily sales, with target indicators like stock price and profits. Some of the most important and common regression algorithms include Linear Regression, Polynomial Regression, and Regularized Regressions. Different regression models can be distinguished by the type of fit that is made; for instance, polynomial regression attempts to find some polynomial (squared, cubed, etc.) relationship between the variables and a label, while Linear Regression attempts to find a linear or straight line trend between the variables and a label. It should also be noted that, thanks to the utility and versatility of ML algorithms, all classification algorithms can be slightly modified into regression counterparts.

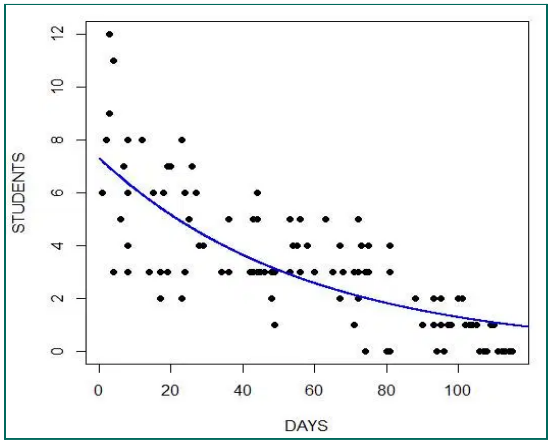

Figure 2: A typical result from running Polynomial Regression to find a trend in volume of student absences

B. Unsupervised Learning

The next class of ML, Unsupervised Learning, focuses on identifying pattern and structure in unlabeled data. This differs innately from Supervised Learning in that there is no output or target label to which input data is being related. As such, most Unsupervised Learning

tasks are often automatic, meaning models don't require explicit training towards optimality; rather, upon receiving input data, these methods find the exact solution to a task. Unsupervised Learning may not fall victim to the same optimization perils of Supervised Learning; however, the utility and correctness of the results in Unsupervised Learning can be difficult to interpret.

As with Supervised Learning, there are two main algorithmic tasks which make up the majority of Unsupervised Learning: clustering and dimensionality reduction. Clustering attempts to find patterns, structure, and groupings in unlabeled data by proximity or density. Clustering techniques are suitable for organizing data when it is far too complex for the human eye to do so. Clustering comes in handy as a first step in analysis of massive and relatively unknown datasets, where it can not only be used to develop initial intuitions about the structure of data, but also help to guide the direction of further analysis. The two most popular algorithmic implementations of clustering are K-means Clustering and Spectral Clustering. The aim of K-means Clustering is to find some number (K) of centroids which effectively and reasonably capture groupings in the data. Spectral Clustering performs essentially the same task, but is much better at estimating structure than simple centroidal proximity.

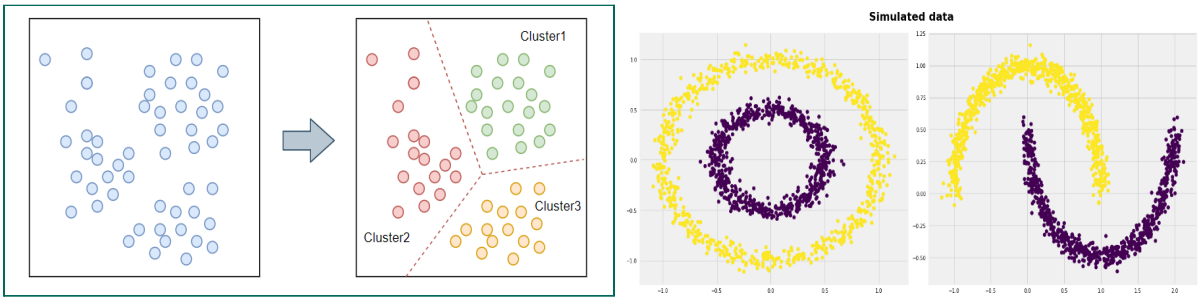

Figure 3: (left) A typical result of the K-means Clustering algorithm where data is organized into K=3 groups; (right) Typical results of the Spectral Clustering algorithm with two groupings of data. Notice that a simple centroid proximate analysis would not yield the same results.

Dimensionality reduction represents the other primary type of Unsupervised Learning. Dimensionality reduction is a group of methods which serve to simplify the number of attributes required to represent a dataset. Sometimes, a dataset might have several thousand samples, each with several thousand attributes, making it unwieldy for many powerful ML techniques. When this is the case, there is a strong chance that some of the attributes are extremely correlated with one another; using dimensionality reduction techniques, these highly correlated attributes can be replaced with fewer attributes which might be a fusion of original attributes. In doing so, the variance of each of the original attributes can be accounted for, while also reducing the amount of information required to describe the dataset. With a more compressed dataset and only important, modified attributes, the use of other ML techniques generally becomes faster and more efficient. That being said, in the realistic application of dimensionality reduction techniques such as Principal Component Analysis (PCA) and t-distributed Stochastic Neighbor Embedding (t-SNE), although the resulting dataset still captures a large majority of the same information in the original dataset, the modified attributes posses almost no individual meaning because modified attributes represent some complex combination of the original attributes.

C. Reinforcement Learning

The last main class of ML is Reinforcement Learning, which takes a significantly different approach than both Supervised and Unsupervised Learning. Reinforcement Learning makes use of an agent which learns through trial and error how to make optimal, sequential actions to accomplish a task. An agent has some set or space of states which describe its current status. This could be the agent's current position and speed, but could also be more abstract and quantitative descriptors. An agent can also interact with its environment by making some action which will always result in an updated state, and will sometimes result in a reward to the agent. As an agent attempts to accomplish a task, the goal is to develop some policy which directs its

actions according to its current state. This policy is a probabilistic modeling of the optimal action to take given the current state in order to maximize the potential reward to the agent. The agent policy is developed by the agent making actions and receiving appropriate rewards from such actions. Thus, Reinforcement Learning is unique because rather than training a model using external, existing data as is done in Supervised and Unsupervised Learning, data is continuously generated by the agent being in different states and performing different actions.

There are again two essential types of Reinforcement Learning, which are model-free RL and model-based RL. Model-free RL describes learning where the agent does not have a prior understanding of its environment and thus has to learn the interactive dynamics between itself and its environment. Much of the literature in Model-free RL breaks down RL problems into three stages: Planning, Trajectory Optimization, and Dynamic Programming. Importantly Dynamic Programming is a method of optimization which attempts to approximate the underlying dynamics of a system. There are many popular Model-free techniques including Value and Policy Iteration, Q-Learning, and Actor-Critic methods.

Model-based RL differs from Model-free RL because the agent develops an understanding of the environmental dynamics before learning and optimizing its policy. Looking back at the general outline for Model-free approaches, Model-based approaches take the step of Dynamic Programming out of the workflow. This is due to the fact that once the agent has achieved Trajectory Optimization, the optimal policy will also be determined without the need to determine the agent's interactions with its environment. Some common techniques in Model-based RL include Sampling-based Planning and Data-Generation. Model-based RL presents huge promise in the future of ML by its generality and utility across a wide variety of tasks.

Now that we are familiar with the intricacies of Reinforcement Learning, it is natural to classify some of the scenarios and tasks it is best suited to. Tasks involving sequential decision making are often the best choice for RL applications. This can include Robotics, Autonomous Vehicles, Language and Dialogue, and Finance. For instance, current state of the art self-driving car technologies, such as route-planning and speed-control are spearheaded by the efforts of Model-based RL. Additionally, Google's *AlphaGo*, *AlphaZero*, and *MuZero* are all examples of Model-based RL algorithms trained to play games (*Medium*). Although these implementations have proven to be highly successful, success in RL often comes at the cost of potentially high variability in performance, long and finicky training processes, and extreme model complexity.

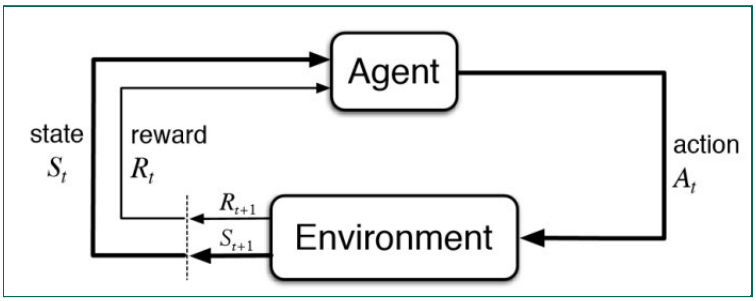

Figure 4: The general outline for how an RL agent interacts with its environment where actions generate states and rewards.

IV. Deep Learning

Deep Learning, as previously mentioned, describes the field of Machine Learning focusing on the implementation of Neural Networks. Neural Networks are a broad family of ML architectures which are identified by highly complex parallel and sequential connected structures called layers, which take in many inputs, perform linear and non-linear transformations, and yield outputs or intermediate results. Neural network architectures are somewhat analogous to biological neuronal systems, and in fact the nomenclature in biological and ML settings overlaps significantly. Deep Learning differs importantly from simpler Machine Learning techniques because it reduces the required human input. In cases of Machine Learning, raw data is often not

suitable for many of the aforementioned techniques because of extraneous factors such as noise, outliers, and impurities. As such, Machine Learning engineers take the role of wrangling and cleaning data, as well as manually extracting important information. This can be broken down into the processes of data curation and feature extraction. These manual processes are oftentimes the most time consuming and tedious stage of work for the analyst or engineer. Deep Learning offers a simplification to this process by incorporating processes such as noise reduction and feature extraction into the Neural Network architecture. This allows for analysts and engineers to spend less time on data preparation for a simpler ML algorithm, and more time tuning and perfecting the performance of a DL model. Deep Learning is thus incredibly attractive and advantageous for many different ML problems, especially across Supervised, Unsupervised, and Reinforcement Learning tasks.

There are three primary Neural Network architectures which make up the majority of the state of the art in supervised deep learning. The first of these is called the Multi-Layer Perceptron (MLP), which is commonly referred to as a Universal Function Approximator. This architecture is characterized by the flow of information being fed forward through layers. MLPs are able to take in an input and perform a large number of parallel and series transformations in order to produce an output. MLPs can be used for both regression and classification; in regression tasks, they are advantageous over traditional ML regression models because they are not limited to a single class of fit. In other words, MLPs can model any relationship between two variables, be it linear or very nonlinear. This is analogous to their use in classification because they can be trained to recognize very complex separations between multiple classes. The difficulty with MLPs, which translates to deep learning in general, is in tuning their performance. There are many adjustable parameters to a neural network, such as how fast it trains, the number of layers (depth), the number of neurons per layer, etc. All of these are

changeable and can have very significant effects on the performance and behavior of the network. The process finding of the settings and architecture which yields the best and most consistent results is known as hyperparameter tuning.

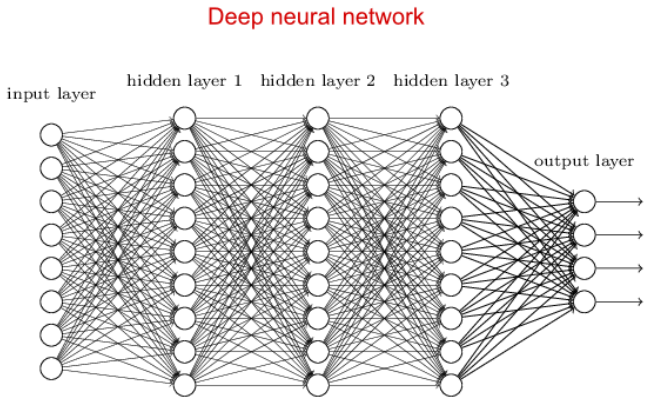

Figure 5: A deep MLP architecture which demonstrates layers whose neurons are fully connected to all neurons in adjacent layers. Information is fed as inputs on the leftmost layer and is propagated through each transformative layer until it reaches the output layer.

The next important supervised deep learning architecture is the Recurrent Neural Network (RNN). These networks make up the class of deep learning models which are specialized for sequential or time-series data. In fact, most of the recent innovations in Natural Language Processing (NLP), which is the field of interpreting language and speech, have come at the hands of RNNs. RNNs differ from simple feed forward MLPs by the fact that their architecture reflects their tailored use case. Essentially, the way these networks function is by taking in some starting input data, whether it be the first 100 days of growth for a particular stock or the first 5 words of a sentence in an email, and can use the sequential nature of this data to make one step ahead predictions about the near future behavior. In the aforementioned examples, this might mean using the first 100 days of growth for a particular stock price to predict the next 25 days of its behavior, or it might mean predicting the next 5 words in the sentence. This task is accomplished by using the more recent data as a highly influential factor

in the near term prediction of behavior, and thus, the most recent output(s) of the RNN are fed back into the architecture and used as additional inputs for the next sequential output. Unlike the MLP architecture, there have been several very powerful advancements and improvements to the baseline RNN architecture, such as the Long Short-Term Memory (LSTM) Network and Transformer Model. LSTMs offer great benefit to tasks such as NLP and predictive modeling by using several unique functions which allow for both recent and distant behavior to affect one-step ahead predictions. Transformers perform a similar task with a revolutionary concept known as self-attention. RNNs have shown great promise in the field NLP but generally, these networks are very difficult to train and are susceptible to fundamental problems such as the Vanishing and Exploding Gradient phenomena.

The last supervised deep learning architecture is the cornerstone of modern computer vision and perception tasks: the Convolutional Neural Network (CNN). First developed by Yann Lecun to recognize handwritten digits, these networks have garnered huge amounts of attention for their specialization in image processing and recognition. The CNN architecture in a deep learning setting can be broken into two main parts. The first is the feature extractor, which uses a special function called a convolution to derive the most important low, medium, and high resolution parts of an image. These feature extractors consist of several convolutional filtering layers which can be trained to derive meaningful properties from a set of images which might show things such as household objects, human faces, and street signs. The second part is essentially an MLP classifier whose inputs are the extracted features and whose outputs are the predictions of what the image is most likely to be classified as. Over the past twenty years, the main improvements to CNNs have been in their depth. Early iterations had only two or three layer feature extractors and similarly only two or three layer classifiers, while the most recent architectures have over 100 layer feature extractors with five to ten layer classifiers.

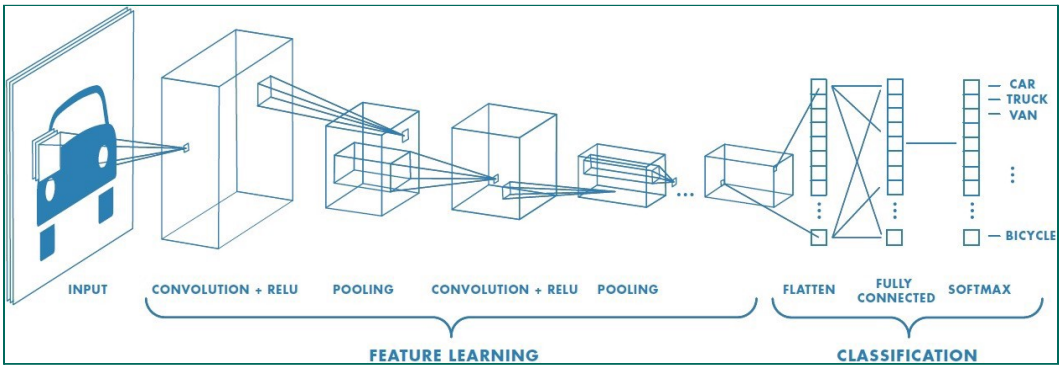

Figure 6: An outline of the CNN architecture, which consists of convolutional and other function layers in the feature extractor and MLP fully connected layers in the classifier.

Not only is deep learning a prevalent field in Supervised Learning, but it has also found use in Unsupervised Learning, mainly through the Autoencoder architecture. The Autoencoder is based heavily on the idea of an MLP, however it is broken into an encoder and a decoder. The encoder serves to reduce an input only to its essential information, while the decoder is used to reconstruct this reduced form back into an un-noisy version of the input. This is commonly used for dimensionality reduction, a common technique in Unsupervised Learning.

Finally, it would be remiss to not mention the application of deep learning techniques to Reinforcement Learning. Deep RL, from both a Model-free and Model-based approach, has garnered a lot of attention for its promising results in comparison to traditional RL methods. All Model-free Deep RL techniques involve using generated agent state, action, and reward data to train an existing neural network or multiple neural network architectures. These architectures have been proven to provide significant performance advantages to more basic, tabular methods such as that used in Q-Learning. Returning to the aforementioned examples in self-driving cars as well as *AlphaGo* and related algorithms, these all make use of Deep Model-based RL. There is huge startup interest in Deep RL being used to improve upon the efficacy and robustness of existing solutions to problems in marketing, manufacturing, and self-driving.

V. Ethics in Machine Learning and Big Data

Now that we have a strong foundation in the principles of Machine Learning and Deep Learning, the ethical and societal implications of such fields must be identified. Ethics in ML and Big Data have become almost as mainstream as the fields themselves. One of the most important ethical questions regards the fair use of data; as the world becomes increasingly digitized, there is also an increase in the amount of data collected and used by companies like Google and Facebook. Although these tech giants aim to maintain transparency with the public about their fair and safe use of user data, there is growing reason for concern. There are several key principles to ethical practice with big data, most of which center around privacy and sharing of data (*Towards Data Science*). Not only should private customer data remain private, but information shared between companies should be limited to only what is shared and to those it is shared with. This illustrates the importance of customer confidentiality as well as limited and fair use of privatized data. Beyond simply the storage, privacy, and sharing of customer data, the use of data must also meet ethical standards. As outlined by the popular documentary *The Social Dilemma*, social media companies like Facebook and Twitter use their access to huge amounts of customer data to monopolize the attention of their users. As in the title of the film, this presents a dilemma because the true product of these social media platforms is the subtle and imperceptible changes in behavior and perception in their users. In other words, using the power of big-data and machine learning to personalize the platform to each individual's preferences, these companies can maximize user interest in their product. The film points out that this can have unintended consequences such as indoctrination in misleading information and more radical changes in behavior (*The Social Dilemma*). These ethical considerations will only become more important as the scale and scope of Machine Learning increases in the future.

VI. Machine Learning in the Scope of Neuroscience

Machine Learning as a broad field draws only weak similarities to the field of Neuroscience, however in state of the art Deep Learning practices, there tends to be much more derivative content. Obviously, the term *Neural Network* can clearly be attributed to the fact that their goal is to perform localized tasks which biological neural systems can also perform. Much of the theoretical nature of methods like MLPs, RNNs, CNNs, and Deep RL draw from equivalent biological systems and tasks such as simple neuronal connectivity, language processing and speech, ocular systems and vision, as well as conditioning and learning how to perform basic tasks. Not only do these similarities exist in the different implementations of ML and DL, but also nomenclature like neurons, signal processing, activation energy, inputs, and firing rate are all shared with neuroscience. This is fairly understandable because the timelines for innovations in neuroscience and ML have been on the same order of magnitude and in the same era of history, most of which has been in the past fifty or so years.

VII. Concluding Thoughts

As we have seen, the field of Machine Learning is both vast and well-developed, despite remaining fairly unknown and obscure. With the advent of more powerful techniques such as Deep Learning and Reinforcement Learning, Machine Learning has also exploded in popularity for both academic research and industrial application. Despite the continued evolution of complex Machine Learning techniques and solutions, it may be wise to keep in mind the idea of Occam's Razor: that oftentimes, the simplest solution to a problem may also be the most effective. Though going forward into the future of ML, there is huge potential for new solutions and applications, we may soon find that some of our most powerful methods are those which have been dismissed as trivial.

Bibliography:

"A History of Machine Learning." *Google*, Google, cloud.withgoogle.com/build/data-analytics/explore-history-machine-learning/.

Coppey, Louis. "What Does AlphaGo vs Lee Sedol Tell Us about the Interaction between Humans and Intelligent Systems?" *Medium*, Point Nine Land, 19 July 2018, medium.com/point-nine-news/what-does-alphago-vs-8dadec65aaf.

Hinton, Geoffrey E., Simon Osindero, and Yee-Whye Teh. "A fast learning algorithm for deep belief nets." *Neural computation* 18.7 (2006): 1527-1554.

Marr, Bernard. "A Short History of Machine Learning -- Every Manager Should Read." *Forbes*, Forbes Magazine, 8 Mar. 2016, www.forbes.com/sites/bernardmarr/2016/02/19/a-short-history-of-machine-learning-every-manager-should-read/?sh=4afe633c15e7.

Sejnowski, Terrence J., and Charles R. Rosenberg. "NETtalk: A parallel network that learns to read aloud." *Neurocomputing: foundations of research*. 1988. 661-672.

Soni, Devin. "Supervised vs. Unsupervised Learning." *Medium*, Towards Data Science, 21 July 2020, towardsdatascience.com/supervised-vs-unsupervised-learning-14f68e32ea8d.

Uria-Recio, Pedro. “5 Principles for Big Data Ethics.” *Medium*, Towards Data Science, 24 Sept. 2018, towardsdatascience.com/5-principles-for-big-data-ethics-b5df1d105cd3.

“What Is Machine Learning? A Definition.” *Expert.ai*, 3 May 2021, www.expert.ai/blog/machine-learning-definition/.