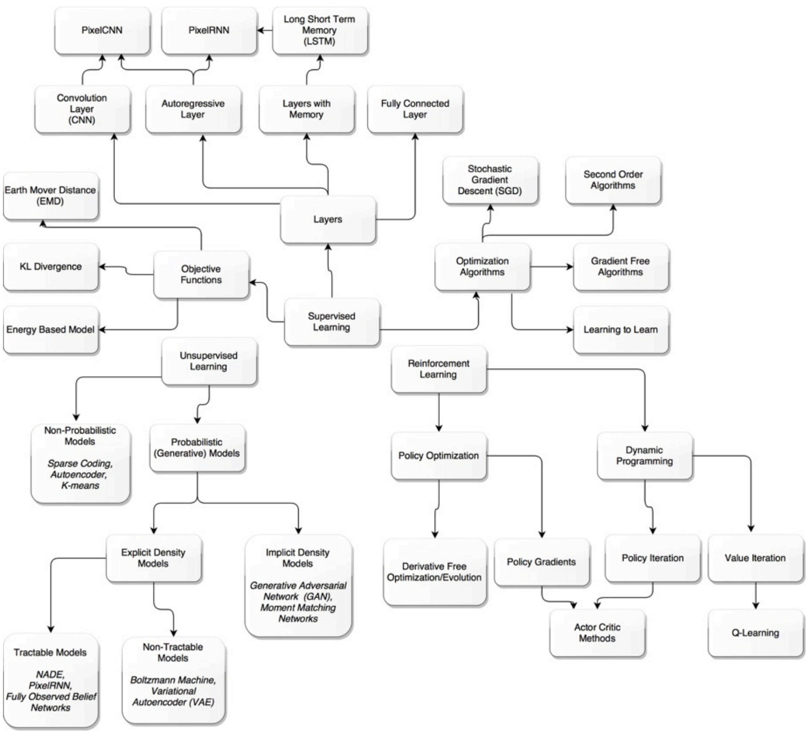
Artificial Intuition

The Deep Learning Revolution



Carlos E. Perez

For information about special quantity discounts, please email info@intuitionmachine.com

Printed and bound in the United States of America.

ISBN:

ISBN-13:

DEDICATION

To Lidith, my wife, who's dedication to make a difference in other people's lives has always been an inspiration. To Consuelo, my mother, who has incomparable empathy and patience. To Matthew, Danika and Ava, who I will always be extremely proud of.

CONTENTS

ACKNOWLEDGMENTS

I also wish to thank the many supporters who took a chance on reading the book prior to its completion.

Images on cover generated by the following Deep Learning projects:
TBD

Preface

I challenge you to find a field as interesting and exciting as Deep Learning. For some background, I attained my early academic training doing Physics, but I've always had a natural inclination towards Computer Science. In my high school years I was introduced to Apple][+ and have been caught with the bug ever since. Physics is one of those winner-take-all professions; if you aren't a genius then you might as well look elsewhere. So instead I studied Computer Science for my graduate degree. However, as a young man with a yearning to achieve more, to explore the technological frontiers of our day and age, I eventually pursued visualization and artificial neural networks.

1 Introduction

"The revolution in deep nets has been very profound, it definitely surprised me, even though I was sitting right there."

- Sergey Brin

Coursera co-founder and Stanford adjunct professor Andrew Ng spoke recently about AI transforming industry:

> *AI is the new electricity. Electricity transformed industries: agriculture, transportation, communication, and manufacturing. I think we are now in that phase where AI technology has advanced to the point where we see a clear path for it to transform multiple industries. Just as electricity transformed industry after industry 100 years ago, I think AI will do the same.*

There are many reasons why businesses are hesitant to jump into AI. Although AI has been around since the 1950s, it has gone through several boom and bust cycles. To add further to that confusion, there are many different approaches to AI. In the last five years however, a new kind of AI called Deep Learning has emerged. Since 2012 it has been delivering spectacular and surprising results. Previous AI failures, the ambiguity regarding what AI means and the newness of this emerging technology are all making it difficult for businesses to comprehend what is occurring.

This problem is further exacerbated by mass media that continues to dole out sensational articles that exhibit boom and gloom. The press either writes about capabilities that are outside the realm of what is feasible or a far-in-the-future scenario that is in the realm of science fiction. This is compounded with the constant drumbeat of spectacular research results coming from academe and industry. Indeed, a lot of excitement is coupled with mass confusion.

Businesses are conservative by nature. The emerging AI economy is however disruptive enough to demand that business take notice. Businesses do not have the luxury of waiting until technology has matured. Deep Learning developments are happening at breakneck speed and are accelerating. I believe that this will have consequences on not only enterprises but also to the future of work.

This book introduces the field, advises on enterprise best practices and surveys the latest developments. The sooner a company gets involved in Deep Learning, the better it will be positioned to reap the benefits of this disruptive new technology.

A New Kind of Artificial Intelligence

Peter Thiel uses a phrase, "The Last Company Advantage" [COM]. Although you don't necessarily need to have the "First Mover Advantage", but you absolutely want to be the last company standing in your business. So Google may be the last Search company, Amazon may be the last E-Commerce company, and Facebook will hopefully not be the last Social Networking company. What keeps me awake at night though is that Deep Learning could in fact be the "Last Invention of Man"!

However, let's ratchet it down a little bit. Kurzweil's Singularity (estimate is 2045) is after all still 3 decades away. That's still plenty of time for us humans to scheme on our little monopolies. Your mission in the next 30 years, if you wish to accept it, is to figure out if you are going to be living in Elysium or in some unnamed decaying backwater.

I quote from a recent interview of Ray Kurzweil which reveals that in "A.D. 2035: Rich people will be thousands of times smarter than poor people" [KOE]:

> *Right now, let's say that you can personally afford to spin up 1,000 Amazon instances, for argument's sake. Elon Musk can spin up one million, perhaps. And maybe Bill Gates can spin up 100 million.*

Deep Learning is a technology that is as revolutionary as the Internet and mobile computing that came before it.

The current revival of interest in all things "Artificial Intelligence" (AI) is primarily due to the spectacular results achieved with Deep Learning research. I must however emphasize that this revival is not due to other classical AI technologies like expert systems, semantic knowledge bases, logic programming, or Bayesian systems. Most of classical AI has not changed much, if any, in the last 5 years. The recent quantum leap has solely been driven by Deep Learning successes. Many venture capitalists like to cling to the term AI, but these days, the term has become meaningless [BOG].

For some perspective on the extent of Deep Learning development, look at this graph from Google that shows the adoption of Deep Learning technology in their applications:

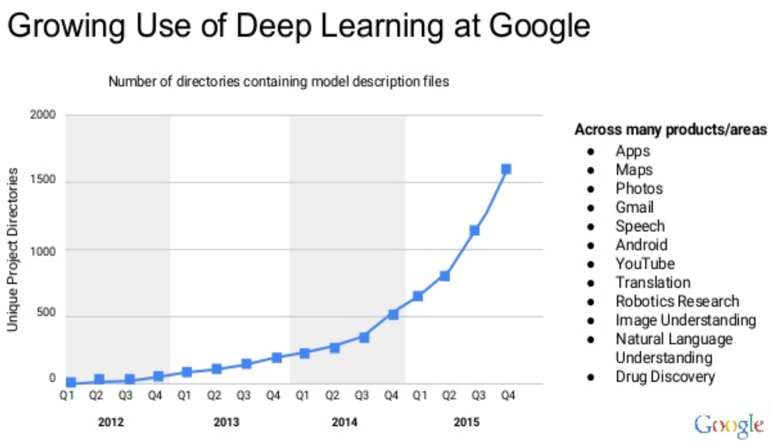


Figure 1.0 Growth of Deep Learning use at Google. Source: Google

As you can see, the adoption at Google has been exponential and the statistics are likely similar for many of the other big Internet firms like Facebook and Microsoft.

When Google embarked on converting their natural language translation software into using Deep Learning (DL), they were surprised to discover major gains. This was best described in a recent article published in the NY Times, "The Great AI Awakening" [LEW]:

> *The neural system, on the English-French language pair, showed an improvement over the old system of seven points. Hughes told Schuster's team they hadn't had even half as strong an improvement in their own system in the last four years.*
>
> *To be sure this wasn't some fluke in the metric, they also turned to their pool of human contractors to do a side-by-side comparison. The user-perception scores, in which sample sentences were graded from zero to six, showed an average improvement of 0.4—roughly equivalent to the aggregate gains of the old system over its entire lifetime of development.*
>
> *In mid-March, Hughes sent his team an email. All projects on the old system were to be suspended immediately.*

Let's pause to recognize what happened at Google.

Since its inception, Google has used every type of AI or machine learning technology imaginable. In spite of this, their average gain for improvement per year was only 0.4%. In Google's first implementation, the improvement due to DL was 7 percentage points better.

This translates to more improvement than the entire lifetime of improvements!

Google likely has the most talented AI and algorithm developers on the planet. Several years of handcrafted classical AI and algorithm development could not hold a candle against a single initial Deep Learning implementation.

Deep Learning is unexpectedly, and disruptively, taking over the world

Google's founder Sergey Brin, an extremely talented computer scientist himself, stated in a recent World Economic Forum [CHA] discussion that he did not foresee deep learning:

> *The revolution in deep nets has been very profound, it definitely surprised me, even though I was sitting right there.*

Sundar Pichai in a recent quarterly financial call said:

> *Machine learning is a core, transformative way by which we're re-thinking how we're doing everything.*

The more precise term is "Deep Learning". He just described it in terms that the less educated financial press could perhaps comprehend.

Deep Learning progress has been taking the academic community by storm. Two articles by practitioners of classical machine learning have summarized why they think DL is taking over the world. Chris Manning, a renowned expert in NLP, writes about the "Deep Learning Tsunami" [MAN]:

> *Deep Learning waves have lapped at the shores of computational linguistics for several years now, but 2015 seems like the year when the full force of the tsunami hit the major Natural Language Processing (NLP) conferences. However, some pundits are predicting that the final damage will be even worse.*

The same sentiment is expressed by Nicholas Paragios, who works in the field of computer vision. Paragios writes in "Computer Vision Research: the Deep Depression" [PARG]:

> *It might be simply because Deep Learning on highly complex, hugely determined in terms of degrees of freedom graphs once endowed with massive amount of annotated data and unthinkable—until very recently—computing power can solve all computer vision problems. If this is the case, well it is simply a matter of time that industry (which seems to be already the case) takes over, research in computer vision becomes a marginal academic objective and the field follows the path of computer graphics (in terms of activity and volume of academic research).*

Make no mistake - Deep Learning is a "Disruptive" technology that is taking over the operations of the most advanced technology companies in the world.

Disruption happens when some new way of doing things entirely changes the way we think, act, learn and go about our daily activities. Clayton Cristensen expresses this more specifically that disruption displaces an existing market, industry or technology. It replaces what exists today with something that is simply better.

The Internet gave civilization a more efficient way to communicate. The World Wide Web introduced a better way to exchange knowledge. Mobile computing gave people ubiquitous access to communication and computation. Deep Learning enhances our cognitive capabilities by making it accessible through the technologies that came before it. Deep Learning is an enabling technology that we improve how we do things across every human activity. Mark Cuban remarked recently [CLI]:

> *"Whatever you are studying right now if you are not getting up to speed on deep learning, neural networks, etc., you lose," says Cuban. "We are going through the process where software will automate software, automation will automate automation."*

What is Deep Learning?

Deep Learning, two simple words that we can all understand, but yet in the context of Artificial Intelligence, when these words are combined they become inscrutable to the uninformed. In fact, even for the informed it is inscrutable. That's because decades worth of statistical training have become a liability in understanding what it means.

Deep Learning is a Connectionist approach to Artificial Intelligence. A Connectionist system is composed of a large number of simple components that collectively exhibit complex behavior. Deep Learning specifically employs multiple layers of components arranged in an acyclic graph (i.e. without loops). There are many kinds of layers but the common property is that the layer is differentiable. Another way of saying this is, the gradient can be calculated for any layer. We will explore this structure in more detail in a later chapter.

Here are 9 points that address some misconceptions of Deep Learning:

1. **Deep Learning is not Good Old Fashion AI (GOFAI)**

Expert systems, semantic web and deductive logic systems are examples of systems that are based on symbolic logic. These systems are traditionally associated with AI. They all do work, but they do have one shortcoming: They are unable to learn effectively from data. GOFAI needs developers to hand craft symbolic rules.

2. **Deep Learning is Different from Machine Learning**

Machine Learning in its most basic distillation is "curve fitting". That is, if you have an algorithm that is able to find the best fit of your mathematical model with observed data, then that's Machine Learning. Although both ML and DL can learn from data there are many capabilities that exist in DL that doesn't exist in ML. Examples of this difference are support for transfer learning and incremental training.

3. **Deep Learning does not mimic Biological Brains**

The architecture DL has is nowhere close to a biological neuron in structure. Even in behavior they are different. Biological neurons work on spiking behavior, DL system work as a continuous dynamical system. Some DL systems use Artificial Neural Networks, but this is just historic terminology that still exists to this day. Anyone explaining DL in terms of biological neurons really doesn't know what he/she is talking about. DL isn't designed to 'mimic' biology, DL just happens to be a computational architecture that learns surprisingly well.

4. **Deep Learning is not Artificial General Intelligence**

DL can do some fantastic things like cross translate between different human languages and read out captions from images. The intelligence is however really specialized and narrow. Sure DL can drive cars, but that's nowhere near the capability of AGI.

5. **Deep Learning is not "Just Math"**

There was a Wired article titled "Deep Learning isn't a Dangerous Genie, it is Just Math" [ETZ]. This is really the most vacuous statement I've heard! It is like saying that computers are just boolean circuits, or brains are just made up of neurons, or DL is made up of layers that are described using mathematical functions. It doesn't explain the emergent complex behavior you find in computers, brains and DL systems.

6. **Deep Learning is not Statistics**

Classical statistics is about analyzing data using aggregate measures. DL systems however work in a domain where statistical methods do not apply. That is high-dimensional data with high mutual information among the variables. Simplifying i.i.d. (Independent and identically distributed [IID]) assumptions are simply not applicable to many realistic scenarios.

7. **Deep Learning is not Big Data**

Big Data is a technology that is based on the idea that if you are able to store and compute through a massive amount of data, typically hosted in hundreds or thousands of off-the-shelf computers, then you can gain insight. DL is an algorithm that can sit on a single machine and can incrementally, special emphasis on incrementally; process your data to learn from it. Big Data can crunch massive amounts of data, but just because you can process a lot of data doesn't mean you can derive insight or learn from the data. One last point, unlike Big Data, DL doesn't need a lot of data to be useful.

8. **Deep Learning is not understood by Data Scientists**

Data Scientists are trained to do modeling of data, feature engineering, and data analysis. DL does what a Data Scientist does, but without a human in the loop. This is actually a bit of an exaggeration. The reality is that most Data Scientists trained in other methods have not come up to speed with DL techniques.

9. **Deep Learning is not just Artificial Neural Networks or Multi-Level Perceptrons**

ANN or MLPs were developed way back in the 1950s. There is a common misconception that Deep Learning today is no different from the older approach. However in recent years new kinds of layer models such as Convolution layers, Autoregressive layers, Long Short Term Memory and Residual layers have extended the original approach. The field has a much richer collection of concepts now than when you first studied it in graduate school.

One way to look at the correspondence of AI, GOFAI, Machine Learning and Deep Learning is through an evolutionary tree. Different approaches emerged at different times and over time new approaches have emerged. So in 1955, the term Artificial Intelligence was coined. The first approach that dominated the space was the GOFAI (i.e. Symbolist), let's make the analogy that this is like the emergence of plants. The development of statistical methods like Bayesian based graph models and Machine Learning techniques, a new kind approach arrived that was able to adapt to data. Consider these as the emergence of reptiles. In 2012 Deep Learning came into prominence, the ideas had been around since the late 1950's. Let's make the analogy that these are like mammals, that is, an evolution from reptiles with distinctly new capabilities. So when many writers make the statement that Deep Learning is Machine Learning, it is as nonsensical as saying that Mammals are Reptiles. Similarly, GOFAI is not Deep Learning, but rather like plants can be turned into tools for mammals, the correspondence of GOFAI with Deep Learning is analogous with paper and humans. Paper is a way for humans to write down knowledge. Symbolist approaches are ways to encode knowledge.

Explaining to a Five Year Old

Richard Feynman had a method of learning complex subjects. The method is simple, try to explain the complex subject to a five year old. If you can't do it, go back, refine you language and try again.

Deep Learning is one of those complex subjects that continues to perplex. Here I am going to attempt an explanation that hopefully could be understood by a five year old.

Just read the following:

> **DE3p Larenn1g mhica3ns wrok smliair to hOw biarns wrok.**
>
> **Tehse mahcnies wrok by s33nig f22Uy pa773rns and cnonc3t1ng t3Hm t0 fU22y cnoc3tps. T3hy wRok l4y3r by ly43r, j5ut lK1e A f1l73r, t4k1NG cmopl3x sc3n3s aNd br3k41ng tH3m dwon itno s1pmLe iD34s.**

I hope this explanation was simple and intuitive enough for you to understand Deep Learning.

Tribes of Artificial Intelligence

One of the biggest confusions about "Artificial Intelligence" is that it is a very vague term. That's because Artificial Intelligence or AI is a term that was coined way back in 1955 with extreme hubris:

> *We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire.*
>
> *The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions, and concepts, solve kinds of problems now reserved for humans, and improve themselves.*
>
> *Dartmouth AI Project Proposal; J. McCarthy et al.; Aug. 31, 1955.*

AI is over half a century old and carries too much baggage with it. For a very long time, Symbolists dominated AI, a rule-based system that had "Zero Learning". In the 1980's a new kind of AI began to emerge, which was called Machine Learning. Finally, we at least had "Simple Learning". The big disruption however, occurred in this decade, when we stumbled upon "Deep Learning", and it has not been taking prisoners ever since.

This is of course a grossly simplified history of AI. There are actually many different approaches or tribes in AI. In his book, the Master Algorithm, Pedro Domingo talks about five different "tribes". Not to be outdone, A YCombinator user "solidrocketfuel" posts about at least "21 different cultures".

It is important for anyone that plans on doing AI to understand that there are differences in the approaches of the different of AI. AI is not a homogenous field, but rather a field in constant tribal warfare. Here's an overview:

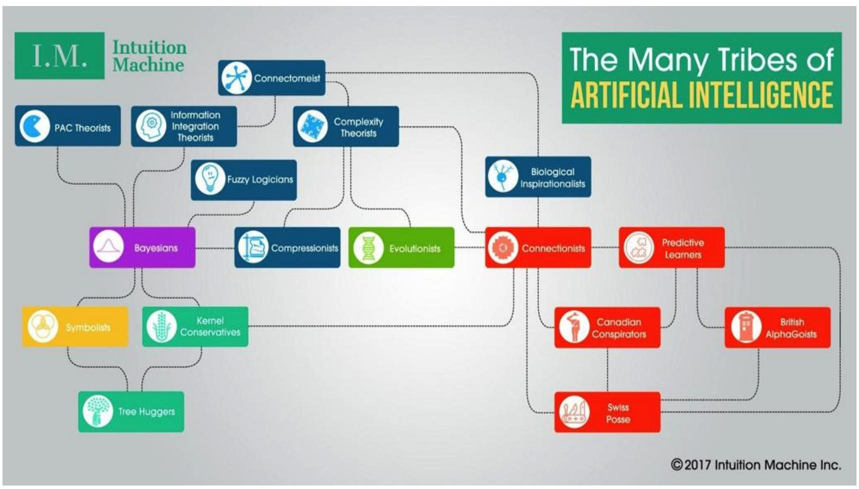


Figure 1.1 Many Tribes of Artificial Intelligence

Here is a breakdown of the many tribes:

Symbolists—These are researchers who leverage symbolic rule-based systems to make inferences. Most of AI has revolved around this approach. The approaches that used Lisp and Prolog are in this group, as well as the SemanticWeb, RDF, and OWL. One of the most ambitious attempts at this is Doug Lenat's Cyc that he started back in the 80's, where he has attempted to encode in logic rules all that we understand about this world. The major flaw is the brittleness of this approach - one always seems to find edge cases where one's rigid knowledge base doesn't seem to apply. Reality just seems to have this kind of fuzziness and uncertainty that is inescapable. It's like playing an endless game of Whack-a-mole.

Evolutionists- Folk who apply evolutionary processes like crossover and mutation to arrive at emergent intelligent behavior. This approach is typically known as Genetic Algorithms. I do see GA techniques used in replacement of a gradient descent approach in Deep Learning, so this approach does not live in isolation. Folk in this tribe also study cellular automata such as Conway's Game of Life [CON] and Complex Adaptive Systems (CAS) [NEW].

Bayesians—Folk who use probabilistic rules and their dependencies to make inferences. Probabilistic Graph Models (PGMs) are a generalization of this approach and the primary computational mechanism is the Monte-Carlo method for sampling distributions. The approach has some similarity with the Symbolist approach in that there is a way to arrive at an explanation of the results. One other advantage of this approach is that there is a measure of uncertainty that can be expressed in results. Edward is one library that mixes this approach with Deep Learning.

Kernel Conservatives (Analogizers) One of the most successful methods prior to the dominance of Deep Learning was SVM. Yann LeCun calls this "glorified template matching". There is what is called a kernel trick that makes an otherwise non-linear separation problem into one that is linear. Practitioners in this field live in delight over the mathematical elegance of their approach. They believe the Deep Learners are nothing but alchemists conjuring up spells without the vaguest understanding of the consequences.

Tree Huggers—Folk who use tree-based models such as Random Forests and Gradient Boosted Decision Trees. These are essentially tree of logic rules that slice up the domain recursively to build a classifier. This approach has actually been pretty effective in many

Kaggle competitions. Microsoft has an approach that melds the tree-based models with Deep Learning.

Connectionists—Folk who believe that intelligent behavior arises from simple mechanisms that are highly interconnected. The first manifestation of this was Perceptrons back in 1959. This approach has died and was resurrected a few times since then. The latest incarnation is of this is Deep Learning.

There are many sub-approaches under Deep Learning. These approaches are all very complementary and often used in combination. The following groups introduced these approaches:

The Canadian Conspirators—Hinton, LeCun, Bengio et al. End-to-end Deep Learning without manual feature engineering.

Swiss Posse—Basically LSTM and that consciousness has been solved by two cooperating RNNs. Jurgen Schmidhuber's group in Switzerland did a lot of pioneering work prior to the Deep Learning boom.

British AlphaGoist—Emphasizes a hybrid approach. AI = Deep Learning + Reinforcement Learning (RL). Their most successful creation to date is the Go playing system AlphaGo.

Predictive Learners— I use the term that Yann LeCun conjured up to describe unsupervised learning. This is the cake of AI or the dark matter of AI. This is a major unsolved area of AI.

In addition to the above mainstream approaches, we also have:

Compressionists—Cognition and learning are compression (actually an idea that is shared by other tribes). The Information Theory derives from an argument about compression. This is a universal concept that it is more powerful than the all too often abused tool of aggregate statistics.

Complexity Theorists- This approach employs methods coming from physics, energy-based models, complexity theory, chaos theory and statistical mechanics. Swarm AI likely fits into this category. If there's any group that has a chance at coming up with a good explanation as to why Deep Learning works, it will likely come from this group.

Fuzzy Logicians—This approach was once quite popular, but for some reason, I haven't heard much about it as of late. One would think that there would be a little more interest here considering the success of the similar 'fuzzy' approach of Deep Learning. A recently published result showed the use of Fuzzy rules defeating a fighter pilot in a mock dogfight.

Biological Inspirationalists—Folk who create models that are closer to what neurons look like in biology. Examples are the Numenta folk and the Spike-and-Integrate or Neuromorphic folks like IBM's TrueNorth chip.

Connectomeist—Folk who believe that the interconnection of the brain (i.e. Connectome) is where intelligence comes from. There's a project that is trying to replicate a virtual worm and there is some ambitious heavily funded research [CNC] that is trying to map the brain in this way [HUM].

Information Integration Theorists—These folks make the argument that consciousness emerges from some internal imagination of machines that mirrors the causality of reality. The motivation of this group is that if we are ever to understand consciousness then we have to at least start thinking about it! However, I cannot see the relationship of learning and consciousness in their approach. It is possible that they aren't related at all! That's maybe why we need sleep.

PAC Theorists— This is proposed by Leslie Valiant in his book "Probably Approximate Correct". Valiant doesn't really want to discuss Artificial Intelligence, but rather prefers studying solely intelligence because at least he knows that it exists! His whole idea is that adaptive systems perform computation (coined as ecorithms) expediently such that they are all probably approximately correct. In short, intelligence does not have the luxury of massive computation.

In summary, there really is a bewildering array of alternative approaches to AI. I am certain that there are other approaches that I have missed. Some approaches are in opposition to each other, while others can be used together synergistically. I do however want to point out is that a bit of understanding of what is out there can help an investor navigate this space.

There are many companies that all claim to be doing AI (companies only need to slap on a .ai domain). As an investor though, you need to ask a more pointed and precise question. What sort of AI is a firm employing? The stark reality here is that not all AI are the same. Said differently, “Some AI are more equal than other AI”.

My opinion is that Deep Learning (i.e. Artificial Intuition) related approaches have a disproportionate monopoly on the upside. The simple reason is: “It is the learning, stupid!” If your AI approach does not have a strong mechanism for learning, then you will forever be doomed to Doug Lenat’s fate (see: Cyc). That is, having to write all the rules by hand (for 30 years)! The other approaches tend to be pretty much dead end approaches. It is critical that the AI approach has a way to learn or alternatively, mechanically bootstrap internal rules.

One of the most effective approaches has been to use Deep Learning in combination with other algorithms. I have seen this in the AlphaGo implementation that used a Monte-Carlo Tree Search technique in combination with Deep Learning. The integration of a symbolic approach with Deep Learning is extremely promising considering that they have complementary strengths and weaknesses. Looking forward into the future, it’s all going to be Deep Learning, one AI to rule them all. Deep Learning combined with some other AI approach will however more likely be just as promising. Ignoring this reality is a surefire way to ensure one’s own viability.

The distinction between AI, ML and DL is very clear to practitioners in these fields. AI is the all-encompassing umbrella that covers everything from Good Old Fashion AI (GOFAI) all the way to connectionist architectures like Deep Learning. ML is a subfield of AI that covers anything that has to do with the study of learning algorithms by training with data. There are whole swaths (not swatches) of techniques that have been developed over the years like

Linear Regression, K-means, Decision Trees, Random Forest, PCA, SVM and finally Artificial Neural Networks (ANN). Artificial Neural Networks is where the field of Deep Learning had its genesis.

Some ML practitioners who have had previous exposure to Neural Networks (ANN) (it was after all invented in the early 60's), would have the first impression that Deep Learning is nothing more than ANN with multiple layers. Furthermore, the success of DL is more due to the availability of more data and the availability of more powerful computational engines like Graphic Processing Units (GPU). This is of course true - the emergence of DL is essentially due to the advantages of having more data and more powerful computational systems.

To coin Andreesen who said, "Software is eating the world", "Deep Learning is eating ML".

The current DL hype tends to be that we have algorithms that if given enough data and enough training time, are able to learn on its own. This is of course either an exaggeration of what the state-of-the-art is capable of, or an over simplification of the actual practice of DL. DL has over the past few years given rise to a massive collection of ideas and techniques that were previously either unknown, or known to be untenable.

Deep Learning today goes beyond just multi-level perceptrons, but is instead a collection of techniques and methods that are used to build composable differentiable architectures. These are extremely capable machine learning systems and we are now only seeing the tip of the iceberg. The key takeaway from this is that Deep Learning may look like alchemy today, but we will eventually learn to practice it like chemistry. That is, we will have a more solid foundation to enable us to build our learning machines with greater predictability of its capabilities.

Armed with this knowledge of the different tribes of AI, we can now get a better understanding of the developments in AI. As an example, we can explore IBM's strategy.

In 2011 IBM was on top of the world in the AI field. The company had demonstrated their DeepQA system that had bested former "Jeopardy!" champions. The feat was so impressive that the Jeopardy champion Ken Jennings declared prior to his imminent loss: "I, for one, welcome our new computer overlords."

IBM's DeepQA leveraged many AI technologies of its time, including NLP, document search, knowledge representations, inference engines and machine learning. IBM then leveraged the visibility they received to begin a new brand: Watson. Watson is a collection of a lot of many of IBM's older products and some newer one all bundled together under the notion of "Cognitive computing". Concurrent with these developments, Deep Learning began to emerge in 2012. Overtime, IBM began incorporating Deep Learning in their newer Watson Developer Cloud platform.

Like almost every technology company, IBM found itself underinvested in this new emerging AI approach. IBM however was well aware of the value of data, so the company went about acquiring data rich companies such as Merge Healthcare ($1B), the Weather Company and Truven Health Analytics ($2.6B). In contrast, Intel, which found itself in 2016 to be behind the curve in Deep Learning, rushed out and acquired several Deep Learning startups (i.e. Nervana, Movidius) and in 2017 acquired Mobileye for $15.3B.

The Sputnik Moment in Asia

Many younger readers may be unfamiliar with the history of Sputnik. The Soviet Union's achievement in launching the first man made satellite (i.e. Sputnik) in 1957 had an outsized effect on the American psyche. Sputnik created the urgency for America to upgrade its science and technology infrastructure:

Sputnik also contributed directly to a new emphasis on science and technology in American schools. With a sense of urgency, Congress enacted the 1958 National Defense Education Act, [NDEA] which provided low-interest loans for college tuition to students majoring in math and science. After the launch of *Sputnik*, a poll conducted and published by the University of Michigan showed that 26% of

Americans surveyed thought that Russian sciences and engineering were superior to that of the United States.

In March 2016, DeepMind's AlphaGo bested Go's world champion Lee Sedol. This was viewed by a shocked audience of over 200 million people. A vast majority of that audience was from countries where the game of Go is popularly played (i.e. China, Japan, Korean). The game of Go has a special reverence in China, it's a 2,500 year old game that is traditionally considered the four arts [ART] that aristocrats considered as essential accomplishments:

They are *qin* (the guqin [GQN], a stringed instrument. 琴 [QIN]), *qi* (the strategy game of Go [GGO], 棋 [CHR1]), *shu* (Chinese calligraphy [CHI] 書 [CHR2]) and *hua* (Chinese painting [PNT] 畫 [CHR3]).

To have a Western developed automation arrive and vanquish a legendary player like Lee Sedol certainly shocked many Asian populations to its core. Chinese authorities were concerned enough about the social disruption that they hastily imposed a country-wide ban on the live-streaming [BAN] of the game. This kind of shock to one's core view of the world will likely galvanize an nation into serious action.

The Koreans promptly created an 860 million fund [ROK] right after the game:

> *Korea announced on 17 March that it would invest $863 million (1 trillion won) in artificial-intelligence (AI) research over the next five years. The commitment includes an already-budgeted 138.8 billion won for 2016; if the rest is spread evenly over the following four years, it represents a 55% increase in annual funding for AI.*

Not to be surpassed, in a July 20th, 2017 article from [CHN] NY Times reported on China's heavy investment on A.I.:

> *Many are spending hundreds of millions of dollars, but some have earmarked even more. In June, the government of Tianjin, an eastern city near Beijing, said it planned to set up a $5 billion fund to support the A.I. industry.*

In addition, just look at the heavy investment money flowing into Chinese AI startups [AIS]. The funding appears to be 10 times more than what you find for US and European AI start-ups.

A Nikkei report August 2017 [NIK] titled "Japan to pump funding into AI chip development" from writes about heavy investment by the Japanese of A.I. hardware:

> *To fund the program, the ministry plans to seek more than 10 billion yen from the fiscal 2018 budget, and will also finance basic studies of next-generation semiconductors.*

All the above announcements indicate substantial government funding.

The Russians (who don't usually play Go) don't really have their own Sputnik event to galvanize more heavy investment. However, Vladimir Putin took into matters into his own hands by imploring the need to educate his population through a broadcasted speech to the students [PUT1] of Russia. He tells Russian students that "the one who becomes the leader in this sphere will be the ruler of the world." In classic Russian fashion, if they can't supply their people with arms, then they might as well have them rely on their grit and perseverance. The Russian objectives [RUSS] however are quite alarming:

> *The government's Military Industrial Committee has set a target of making 30 percent of military equipment robotic by 2025.*

In stark contrast to these panicking nations, the EU and US government investments in Artificial Intelligence and more specifically in Deep Learning have been inconspicuously absent from announcing any major investments. I won't hold my breath waiting for this to change. For most Westerners, few have ever played or much less seen a game of Go. DeepMind's accomplishment is seen as some obscure esoteric achievement that requires very little urgency in response. There is zero appreciation of the magnitude of this achievement.

I surmise that A.I. isn't consider to be a public good that should be shared by its citizenry. The West appears to be all perfectly fine surrendering their own privacy to a few private monopolists in exchange for an occasional dopamine fix. A recent article [SPU], tells you about the sad state of affairs:

> *To date, the Trump administration has paid little attention to how AI is likely to affect Americans—or the world writ large. Treasury Secretary Mnuchin has cavalierly dismissed [MNUC] concerns that automation will displace U.S. workers; the Office of Science and Technology Policy lies in shambles [TRP]; and the State Department's science envoy recently resigned [SCI] while calling for the president's impeachment.*

Well, at least the Canadians are proactively throwing in some spare change [CND] to address the rising competition:

> *In Budget 2017, the Government of Canada announced $125 million in funding for a Pan-Canadian Artificial Intelligence (AI) Strategy to be led by the Canadian Institute for Advanced Research (CIFAR).*

The severe lack of government subsidy in the U.S. is forcing academic institutions into selling their souls to private corporations. Corporations want to lock-in the intellectual property as fast as possible, the best way to do that is to lock-up the Deep Learning researchers. Meanwhile academic institutions with the smarts are starved of government research funding and are forced into indentured servitude. The latest MIT-IBM announcement [MIB] is simply a reflection of this dire predicament.

In conclusion, there indeed has been a Sputnik moment for East Asian countries. The consequence of this is an urgency to upgrade their competitiveness in A.I. and Deep Learning. Meanwhile, the rest of the world is mostly complacent, neglecting vital research funding and unawares of the disruptive potential of this new technology.

In my opinion, I don't think the West can rest on the idea that firms like Google, Microsoft and Facebook are performing the majority of the research in Deep Learning. Certainly we see them sharing some of their research. However, but I highly doubt that they will be quick to share their biggest breakthroughs. These firms are way ahead of anything the government is doing. A recent report shows the current problems the defense industry [DEF] is having:

> *"That's a hard problem to solve because … a lot of the best minds are moving to these large commercial ventures," he said. "You can't even find them in the universities anymore. They're being stripped out of universities to go work and develop algorithms" in places like Silicon Valley.*

To conclude, current A.I. research and funding will benefit only a few firms and a few nations. For the rest of us, we'll have to beg that others will be kind enough to share their discoveries.

2 Applications in the Wild

There was a time when people felt the internet was another world, but now people realize it's a tool that we use in this world. - Tim Berners-Lee

In this chapter I will discuss applications of Deep Learning in a more general sense. I will cover the ideas of self-driving enterprises, assistive and generative forms of automation, behavior prediction, and a recent advance revelation about how Deep Learning can teach us about complexity. We end the chapter with a list of applications that have been developed using this disruptive technology.

Self Driving Anything

Self Driving Cars are all the rage these days. When Udacity announced a "self driving car engineering degree" a few months ago, they were swamped with 11,000 hopefuls:

> *The high number of applicants—for 250 spots in the course—underscores the pressing need for talent by technology leaders such as Alphabet's Google and Apple, traditional car companies and automotive start-ups, as they race to develop production-ready autonomous-driving vehicles within the next decade.*

Self-Driving Automation is an entirely new field. It is usually described as Self-Driving Cars. Knowledge in this space is in its infancy. It is a complex field that involves the integration of many different technologies and the real-time orchestration of these integrations. A glimpse of this idea of an automation that employs

Deep Learning, Vision, Sensor Fusion and many other technologies can be found in Amazon Go. Amazon Go isn't a car - it is a self-service retail store!

> *As for how its "Just Walk Out Shopping" experience works, Amazon seems emphatically not to want to share details. It steeps its description of how the system works in buzzwords: computer vision, sensor fusion, and deep learning. It uses sensors throughout the store and artificial intelligence to tell which direction customers are looking, even in a crowd, and can identify partially blocked labels. Beyond that, details are hazy.*

The software development field has contributed hugely to the way we do work and the way we run businesses. In the early 2000's the "Agile Manifesto"*[BEC]* was created in response to our collective inability to effectively manage complex software development. The tenets of the manifesto are as follows:

> *Individuals and interactions over processes and tools*
>
> *Working software over comprehensive documentation*
>
> *Customer collaboration over contract negotiation*
>
> *Responding to change over following a plan*
>
> *That is, while there is value in the items on the right, we value the items on the left more.*

Software development is fundamentally a knowledge creation activity. The factory floor processes that are so effective in the physical world simply don't apply in the virtual world. The agile manifesto was created to challenge our conventional wisdom of processes that were prevalent at that time.

Agile development has evolved since then and what we have seen is the practice of employing automation wherever needed. Programmers are extremely lazy folk - rather than perform any kind of repetitive and monotonous work, their preference would be to code out automation. It is therefore unsurprising to find so much automation in agile practices. Examples of this include intelligent editors, automated testing, continuous integration and deployment; tracking dashboards, issue trackers that automate tracking and version control that automate change control. The most sophisticated uses of automation can be found in agile software development, and this is indeed a rich source of ideas of how to integrate advanced DL technology into one's own processes.

The key advantage of agility is its ability to respond quickly to change. If you can automate sections of that feedback loop without compromising, then you have a tighter loop and therefore a more nimble process.

The concepts of "continuous integration" and DevOps are enabled because of automation. In other words, agile processes are effective because there is an unrelenting motivation to insert automation to reduce human work. This kind of automation, however, works in synergy with human processes. The correct prescription for introducing automation in one's processes is to begin with a process that is centered around humans and then to amplify that with automation.

The agile approach is in contrast with the waterfall approach; That is - beginning with a factory floor model and then automating the parts. This leaves you with even more inflexible processes and many unhappy workers. The mechanization of work may have worked in an earlier era that lacked the coordination and communication capabilities of computers. In a world were computation and devices are abundant and pervasive, the only efficient workflow model that one should consider is that based on adaptive principles.

Ideas coming from the agile development have led to further usage in the world of startups. The startup world is a world where agility is a critical capability for survival. Startups are buying into the agile philosophy of prioritizing feedback with potential customers. The purpose is to create a learning organization by effectively iterating through many business plans. For startups with a lack of vision, this kind of "The Lean Startup" process is better than having no process at all [LEA]. How can we learn more quickly what works, and discard what doesn't? That's the essence of the approach.

Generate Anything

We can classify DL capabilities into 2 kinds: Assistive and Generative. These kinds are differentiated by how a human interacts with the DL system. In both kinds, the human is rarely out of the loop; the human is rather in constant collaboration with the AI in their work.

We are very familiar with automation with assistive capabilities. We experience it every day when we text on our cell phones. The phone is able to spell correctly and provides suggestion to words that we type. We also find this capability in the cameras we use. We now take for granted the auto-focus capabilities of our cameras. Newer DL driven cameras like that found in the Google's Pixel are even able to blur out image backgrounds. This is a capability only available on DSLRs with much larger lenses. Assistive capabilities allow us to work at a quicker pace, enhancing our content creation on the fly towards even greater quality.

We see many of these assistive technologies deployed in the software development space. IDEs are enabled with many capabilities that assist and correct developers on the fly. We have style checkers, bug checkers, security vulnerability checkers, code metrics, code coverage etc. applied to source code as we write code. This kind of automated feedback via continuous integration is a best practice in software development. There are tools that not only correct, but also perform tedious tasks that require precision like refactoring.

As we can see, assistive capabilities can happen in real time as well as in the backend. DL allows us to perform repetitive and time-consuming tasks such as sorting and categorizing our photo collections. We are also continually overloaded with information. There are certain professions where the ability to curate and analyze information is extremely valuable. We can enhance these curation and analysis capabilities by reducing the deluge of information into smaller chunks that are more quickly digestible.

Generative capabilities are a new kind of capability that is just becoming more pervasive. By now, we've all experienced the capabilities of the mobile app Prisma that is able to re-render photographs into the style of different artists:

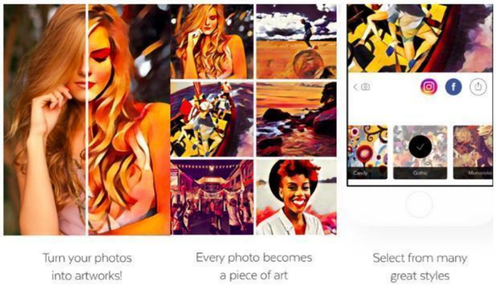


Figure 2.0 Prisma mobile application. Source: http://prisma-ai.com/ [TMA]

Progress in "visual attribute" transfer has been extremely impressive that not only style be transferred but other attributes such as color, texture or tone. A new paper "**Visual Attribute Transfer through Deep Image Analogy**" [LIA] reveals some astonishing capabilities:

Figure 2.1 Attribute style transfer examples. Source: https://arxiv.org/pdf/1705.01088v1.pdf [LIA]

Wherein the style transfer is performed in a manner that is less haphazard and more deliberate fashion. The automation appears to have a comprehension of the underlying semantics to be able to apply a different style to apply to each specific segment of an object in the scene. Observe how the automation was able to discern that it preserves only the hair color to be red or that it paints only within the shape of the horses. It really is mind boggling how automation is able to discover how to do this through training alone. [LIA]

This is an example of a generative application. We can use DL to explore a design space exhaustively to serve as a "brainstorm" of ideas. This mode also works in the realm of planning and execution. DL can quickly explore many scenarios and present workers with a Chinese menu of promising options. We've previously written about some interesting examples of generative design: "The Alien Style of Deep Learning Generative Design" [PER13]. We have only seen the tip of the iceberg here, as there is so much more to explore in this space.

One application of DL in the space of "Game Theoretic" design is enough to scare the bejeezus out of anyone.

Predicting Any Behavior

One of the most promising applications of Deep Learning (DL) is in its use to enhance interaction with computers. DL is particularly suited for this since it has an intuitive capability that is similar to biological brains. It is able to handle the inherent unpredictability and fuzziness of the natural world.

This does however imply that improving AI to human interaction will require some level of capability of an AI to predict human behavior. This is a requisite capability and interestingly enough, there has been some recent research on this topic.

The conventional behavioral model approach is to have social scientists build analytic models (statistical models in most cases) of human behavior and to employ a distribution fitting exercise to verify validity of the model. The DL approach is however different since a prior model is not required to begin with. Rather, the DL system learns to predict by observing the behavior of human participants. We can get a glimpse of this groundbreaking technique by examining some recent research work that uses this approach.

MIT trained a Predictive Vision [VIS] system on YouTube videos from shows like "The Office" and "Desperate Housewives" to predict whether two persons will hug, kiss, shake hands or slap a five [CSI]. The trained the Deep Learning system on 600 hours of video. The system was able to predict an action 43 percent of the time. This compares to previous algorithms that could only predict 36 percent of the time.

One pragmatic use of predicting human behavior is to do so in the confines of a car. Brains4Cars uses a sensor fusion Deep Learning system based on LSTMs to anticipate driver behavior 3.5 seconds before it happens [BRA]. It uses a collection of sensors such as cameras, tactile sensors and wearable devices to make its predictions.

A recent NIPS 2016 paper uses Deep Learning to predict strategic behavior [HWL]. In most systems the assumption is that the participants perform in a perfectly rational manner and are based on insights from cognitive psychology and experimental economics. In this system however, one that is based on Deep Learning, the system learns a cognitive model without the need for expert knowledge. This system is able to outperform systems that are built from expertly constructed features.

Extending beyond just making predictions, Deep Learning systems have been used to assist in contexts with human negotiation. In a paper, "Reinforcement Learning of Multi-Issue Negotiation Dialogue Policies" [NEG] the authors used Reinforcement Learning and a handcrafted agenda based policy and evaluated them by having each negotiate against the other in different settings [PAP]. It was discovered that the RL model consistently outperformed the hand crafted agenda based model. In addition, humans were asked to rate both systems and the result was that the RL based approach was rated to be more "rational".

Finally, in an impressive act of engineering, a team from the Czech Republic and Canada created a Poker playing system that played 33 professional pokers players from 17 countries in Heads Up No Limit (HUNL) poker. The system gained a win rate that was an order of magnitude better than a good player rating. The team coined their creation DeepStack [MMN].

DeepStack takes a fundamentally different approach. It continues to use the recursive reasoning of Counter Factual Reasoning (CFR) to handle information asymmetry. However it does not compute and store a complete strategy prior to playing and thus has no need for explicit abstraction. Instead it considers each particular situation as it arises during play, but not in isolation. It avoids reasoning about the entire remainder of the game by substituting the computation beyond a certain depth with a fast approximate estimate. This estimate can be thought of as DeepStack's intuition:

> *a gut feeling of the value of holding any possible private cards in any possible poker situation. Finally, DeepStack's intuition, much like human intuition, needs to be trained. We train it with Deep Learning using examples generated from random poker situations. We show that DeepStack is theoretically sound, produces substantially less exploitable strategies than abstraction-based techniques, and is the first program to beat professional poker players at HUNL with a remarkable average win rate of over 450 mbb/g.*

The DeepStack algorithm is composed of three ingredients: a sound local strategy computation for the current public state, depth-limited look-ahead using a learned value function over arbitrary poker situations, and a restricted set of look-ahead actions.

There's additional commentary about DeepStack in Technology Review [KNI2]:

> *The researchers compare DeepStack's approximation technique to a human player's instinct for when an opponent is bluffing or holding a winning hand, although the machine has to base its assessment on the opponent's betting patterns rather than his or her body language. "This estimate can be thought of as DeepStack's intuition," they write. "A gut feeling of the value of holding any possible private cards in any possible poker situation ."*

In this context, not only is DeepStack able to perform accurate predictions of the behaviors of its opponents, it does so in a way that its own behavior is not predictable!

To summarize, Deep Learning is able to make prediction on tacit behavior of humans as well as rational behavior. The ability to anticipate behavior, predict behavior, or win in games of bluffing are extremely advantageous tools to have in one's business arsenal. However, if you still aren't convinced that Deep Learning can predict human behavior, then you might wonder why Facebook has a job opening in this kind of work: Facebook's Mysterious Job Listing Sounds Like It's Working on How to Read Your Mind"[JRB].

Teaching Machines

The western world is unaware of the profound effect of AlphaGo's mastery of the game of Go has had on the psyche of the population of China, Japan and Korea. We are unfamiliar with the game of Go, so DeepMind's accomplishment isn't as relevant to us. These countries have now made a national priority to sprint ahead in the development of Deep Learning technology.

Meanwhile, in the United States, there is a backlash against science and there is zero conversation on a public initiative to accelerate development in this space. The U.S. is fortunate however to have deep pocketed internet giants like Google, Facebook, Amazon and Microsoft that have essentially carved out there own monopolies for Deep Learning talent.

AlphaGo has profound implications that go beyond just achieving a breakthrough in strategic game playing technology. Rather, it is the realization by its creators (i.e. DeepMind), that AlphaGo has not only bested the most skilled human but is inventing new strategies that goes beyond what human's can conceive of. In short, it is transcendental technology. This is not hyperbole, but rather it is quite real indeed.

A bit of explanation is needed here to explain the mechanics of AlphaGo. AlphaGo [AGO] is architected using many different techniques. It combines Deep Learning, Reinforcement Learning and Monte-Carlo Tree Search (MCTS) in an innovative way to solve a narrow AI problem of playing expert level Go. AlphaGo uses supervised Deep Learning to bootstrap a Reinforcement Learning policy function. The developers begin with a training set of previously played games. The system then is trained to play against itself, learning a value function that is leveraged in MCTS to discover innovative new game play.

The Go communities findings are that AlphaGo has revealed novel new strategies that human thinkers have not discovered. In fact, expert Go players who have played against AlphaGo have improved their own game play. In fact today, there are groups that study AlphaGo's game play to gain greater insight on the game of Go. Gu Li, Go's world champion, remarked that "AlphaGo's self play games are incredible - we can learn many things from them."

DeepMind's Demis Hassabis has been pushing this narrative in several talks. DeepMind has subsequently made available details of the games that AlphaGo has played in the hopes that players can learn from them. It is definitely a paradigm shift [HAS]. That is, we have machines now teaching humans to improve their game. When you leap from assistive to generative technology, you transition to an entirely new paradigm.

Summary

To summarize, Artificial Intuition (or Deep Learning) is a technology that is demonstrably superior to previous AI methods. The current revival of interest in all things "Artificial Intelligence" (AI) is primarily due to the spectacular results achieved with Deep Learning research. The recent quantum leap has solely been driven by Deep Learning successes.

The smartest companies in the world are migrating their infrastructure to support this new paradigm. Artificial Intuition is a departure from the traditional reductionist way of thinking and is an entirely new way of "automating automation." On a daily basis, the press continues to report the amazing progress of AI. Furthermore, you hear about firms like Google and Microsoft changing their entire software DNA to move into AI. The reason for this massive migration is Deep Learning.

Deep Learning is enabling self-driving cars and other self-driving automation. Self-Driving Automation is an entirely new field. It is usually described as Self-Driving Cars. Knowledge in this space is in its infancy. A glimpse of this idea of an automation that employs Deep Learning, Vision, Sensor Fusion and many other technologies can be found in Amazon Go. Amazon Go isn't a car - it is a self-service retail store!

Deep Learning is supporting work by not only providing assistive capabilities, but also more creative generative capabilities. Assistive capabilities can happen in real time as well as in the backend. There are certain professions where the ability to curate and analyze information is extremely valuable. We can enhance these curation and analysis capabilities by reducing the deluge of information into smaller chunks that are more quickly digestible.

Generative capabilities are a new kind of capability that is becoming more pervasive. By now, we've all experienced the capabilities of mobile app Prisma that is able to re-render photographs into the style of different artists.

Deep Learning is also enabling the prediction of human behavior. The utilization of AI to manipulate human behavior has begun to emerge. Deep Learning is able to make prediction on tacit behavior of humans as well as rational behavior. The ability to anticipate behavior, predict behavior, or win in games of bluffing are extremely advantageous tools to have in one's business arsenal.

The major companies are acquiring Deep Learning talent like there's no tomorrow. Deep Learning is unexpectedly, and disruptively, taking over the world.

I will spare some trees by not having to re-document existing Deep Learning applications, However, here is a short table of the applications that have been built:

Use Case	Category
Real-time Conversational Assistance [GAS]	Classification
Brain Tumor Detection [HAV]	Classification
3D Object Classification [3DS]	Classification

Gesture Recognition [DHR]	Classification
Sorting Cucumbers [FAR]	Classification
Detecting Spam Email [MET]	Classification
Automated Email Replies [MET2]	Classification
Identification of Common Objects [CED]	Classification
Organizing Photographs [GOO]	Classification
Face Identification [CMU]	Classification
Real-time Speech Translation [SKY]	Translation
Self-Driving Cars [MAB]	Translation, Planning, Prediction
Championship Go Play [KOC]	Planning, Prediction
No-Limit Texas Hold'em Poker Play	Prediction
Predictive Keyboard [BUR]	Classification
Infrared Colorization [LIM]	Generation
Blur Out Background in Images [STA]	Generation
Translation of Images of Text [GOOD]	Translation
Sketch to Generate Realistic Photos [GUC]	Translation
Sketch to Search [USI]	Translation
Speech Synthesis [WAV]	Translation
Face Tracking for Augmented Reality	Optimization
Warehouse Optimization [OPT]	Optimization, Planning
Reducing Electric Bill [CLJ]	Optimization
Product Recommendations [CHU]	Prediction
Predicting Clinical Events [CHO]	Prediction
Drug Design [PAS]	Generation, Optimization
Reducing Traffic [ALM]	Optimization
Reverse Engineering Biological Processes [PLA]	Generation
Fluid Simulation	Generation
Conversational Interfaces [KHA]	Translation
Stocking Shelves [VIN]	Optimization

Be sure to navigate the link to see more details.

If this is all new to you, or you have yet to jumpstart your Deep Learning investment, then this quote should be applicable: "If you are not at the table, then you're on the menu."

3 Intuition and Cognitive Bias

"So recognizing things is difficult for the machines at the present time and some of those things are done in a snap by a person. So there are things that humans can do, that we don't know how to do in a filing system." - Richard Feynman

Richard Feynman spoke the above words in a lecture explaining how computers worked. He used a 'filing system' as an analogy for his explanations. What Feynman was describing was the limitation of machines in relation to the capabilities of humans. What he was not able to foresee was that this intuitive capability would be achieved by Deep Learning in 2012.

This chapter explores human thinking, its limits and its nuisances. The revelation here is that Deep Learning systems are intuition machines. Understanding this idea will give an innovator a solid footing in understanding the capabilities and limitations of this emerging technology. This is a necessary first step in being able to apply any new technology.

Limits of Rationalilty

There are three essential ingredients needed to understand intelligence that, unfortunately, present day mathematics has trouble tackling. Mathematics are tools that enhance our reasoning processes. Mathematics is a human language that we employ to derive understanding of reality. However, mathematical languages are not all-powerful and do have limitations. We explore some of these limitations here with respect to areas important to AI.

Although mathematics tends to get developed way ahead of its time, there are many times that the application of a different kind of mathematics to a new domain leads to breakthroughs. Richard Feynman, for example, employed century-old path integrals mathematics to gain new insight on developing Quantum Electrodynamics. There are however plenty of limitations in mathematics, and this article addresses those limitations with respect to our ability to comprehend essential ingredients of cognition.

The "Quasi-empiricism" of math is not a new idea [QUA]. Mathematics is a human language that we employ to describe our reality. Quoted from the Wikipedia article [WIKI]:

Eugene Wigner (1960) noted that this culture need not be restricted to mathematics, physics, or even humans [EUG]. He stated further that "The miracle of the appropriateness of the language of mathematics for the formulation of the laws of physics is a wonderful gift which we neither understand nor deserve. We should be grateful for it and hope that it will remain valid in future research and that it will extend, for better or for worse, to our pleasure.

Reasoning about Time

The first ingredient is the notion of time and memory.

Time is a difficult concept to grasp. I guess the easiest way to handle it is to do what Einstein did: Just treat it as another dimension.

Most physics is invariant in time. Meaning, you can move forward or backward in time and the physics are identical. However, at the macro-world we don't see it that way, time exists because entropy exists. The arrow of time follows that of increasing entropy.

In fact, there's really no notion of memory without having to consider the existence of time.

Most mathematics has no concept of memory. Memory is the equivalent of having state, and almost all mathematics involves functional constructs that are stateless. Functional programming follows a single assignment rule where once any variable is set, it remains set to that state, never changing. It is this constraint that makes the use of functional programming something that is easily parallelizable. It is also a convenient constraint that allows our mathematics to be analyzable and able to make leaps in prediction.

We cannot, however, avoid time, because that's where real dynamics come from. The only context that mathematics is helpful in dynamics is in the context where memory is not present. Introduce memory or introduce state, then all bets are off! The best that mathematics can do is to quantify the boundaries of computation without predict its final behavior [WIKI2].

Dynamics that are analyzable by math are in equilibrium states. We can only make statements about states that are in equilibrium, that is only after the interesting dynamics has settled. Computation, what happens in between, can only be, at best, simulated. Equilibrium is the state where we assume that time approximates infinity. An unrealistic assumption, but this assumption is brought about because of convenience.

There is also the notion of asynchrony that is such a beast in complexity. This is when multiple events happen without apparent correlation. That is, when different parallel processes are not in lockstep synchrony. All our digital circuitry requires lockstep synchrony in the form of a common clock. The biological brain does not have a common clock, it works in a regime of asynchrony.

Reasoning about Emergent Behavior

The second ingredient is the notion of collective emergent behavior.

Robert Sapolsky has a short lecture on Youtube [SAP] ("Thinking about emergence and chaos") that brings about the point about bottom-up behavior. He says that "most of the stuff that he and his peers do is reductive stuff that is very limited."

Intelligence comes from the emergent behavior that arises from the collective behavior of millions or billions of interacting components. This is the very essence of the concept of Connectionist AI. The components themselves do not have to be constructed in a complex manner. They can be very simple and in fact be all uniform or homogeneous. Artificial Neural Networks and Deep Learning spring from this very idea of deriving intelligence from simple components called 'neurons'. It is important to remind oneself that the neurons in ANN are a cartoonish version of a biological neuron. However, it is not the precise construction of the neuron that is important, but rather it is the collective behavior that is important.

That is why the reasoning that ANN and DL should be rejected because they are not biologically plausible is a very bad argument. It is entirely conceivable that intelligence can be arrived with very different kinds of 'neurons'. That's because, there's some fundamental capability that a neuron performs (i.e. information dynamics, meaning computation, memory, and signaling) all that is needed, however, the connectivity is where intelligence emerges.

Reasoning about Meta-Level Reasoning

The third ingredient is the notion of meta-level reasoning.

This is the most difficult to grasp idea and it may, in fact, be the reason why 'consciousness' exists. We can understand the idea of building up ideas by the composition of more primitive ideas. We can understand this because that is how language is constructed. That is, from letters to syllables to words to sentences to paragraphs etc.

We also know of meta-level reasoning. It's one of those ideas that's hard to explain to novice programmers, but it exists in many programming languages. That is, you have programs that operate on the building blocks of the language itself. It leads to very expressive and short source code. Experienced programmers have no difficulty working at the meta-level. However, these kinds of system are extremely difficult to debug.

However, it doesn't stop with just one level of meta-reasoning. You could have meta-meta level constructs ad infinitum. I've encountered this idea in the wild in the modeling language UML. There's a concept of meta-metamodels, here's the definition:

A **metamodel** or surrogate model is a model of a model, and **metamodeling** is the process of generating such metamodels.

Which is the most universal definition of Generalization.

Artificial Intuition

How is intuition developed? How does this lead to innovation? What does this have to do with Deep Learning? Intuition like consciousness is something that we are aware of its existence, but likely have not investigated in enough detail to have a grounded understanding of its nature. In fact, we would say that there's more research on the nature of consciousness than research on intuition. There exists a few research groups that have explored consciousness with respect to an artificial general intelligence, however I don't think this has been equivalently the same effort compared to the study of intuition.

The specific interest of mine is in the study not of human intuition but of artificial intuition. In summary, artificial intuition, applicable equally to intuition, has attributes that differ from the more studied form of cognition (i.e. logical, rational). It is my conviction that Deep Learning systems are artificial intuition systems in contrast to GOFAI, which is based on rational cognition. This judgment call is based on the commonality of attributes of intuition and Deep Learning. [PERR]

In this chapter however, I explore deeper an understanding of intuition and reveal the existence of blind spots in developing more capable deep learning systems.

Bruce Kasanoff has a very short article on Forbes, "Intuition Is The Highest Form Of Intelligence" which spurred me to explore this idea further [KAS]. The study of intuition is not entirely new. Dual Process Theory introduces the idea that the mind works using two different kinds of cognitive systems. Kahneman, (see "Thinking, Fast and Slow") employs Dual Process theory to explore how humans can fall into biased and incorrect thinking as a consequence of the interplay of intuition and rationality [HOL].

There are however other investigators that have studied intuition in greater detail. One of the most popular books written about intuition is arguably Malcolm Gladwell's book "**Blink: The Power of Thinking Without Thinking**". Gladwell's analogy of intuition is this idea that the mind "perfected the art of filtering the very few factors that matter from an overwhelming number of variables". I would argue that this is an imprecise analogy. Inferences or predictions made through intuition are in many cases extremely difficult to explain. The mind does not do "filtering"; rather, it performs predictions in a massive parallel manner that appears to be a process of reduction, but is entirely different. Interestingly enough, Gladwell writes "In the act of tearing something apart, you lose its meaning."

Gladwell popularized though the idea of intuition, enough to compel Gerd Gigerenzer, director and researcher at Max Planck Institute for Human Development, to write his own book "Gut Feelings: The Intelligence of the Unconscious" explaining his research in much greater detail [ELR]. Gigerenzer defines intuitions as having the following attributes:

> *"Appear quickly in consciousness",*
>
> *"Their underlying reasons we are not fully aware of"*
>
> *"Are strong enough to act upon."*

The open question though is, how does the brain improve intuitive thinking? Theo Humphries explores this in "Considering intuition in the context of Design and Psychology". Humphries challenges the ideas of Psychologists that have tagged the notion of intuition "to the 'fringes' of the field of psychology, within the realms of parapsychology, telepathy and premonition" [HUMP]. From a design perspective this is surprising to him, and he provides an enumeration of his surprise:

Intuition, as understood from a design perspective, appears to be so important as one interacts with "the designed world". As an example of a poor and good intuitive design:

Figure 3.0 Unintuitive design (left) and Intuitive design (right). Source: http://theohumphries.com/papers/HUMPHRIES_intuition_psychology_design.pdf [HUM]

Intuitive designs that are labeled as 'intuitive' is an indicator of a mark of praise and intuition is a foundational concern for usability design. Humphries explores the disconnect between the design and psychology community:

> *for psychologists 'action that is not planned or premeditated, answers that come without reasons, understandings that cannot clearly and quickly be put into words, are stigmatized as essentially second rate." However, for designers, and artists (designers closely associated kin) spontaneous action and tacit intuition are valued as vitally important.*

Designers however do not always work purely from their gut. Expert designers apparently have better intuition than novice designers. Experts have gained their knowledge through experience. In many complex fields, this kind of experience is captured in explicit form through the use of Design Patterns (alternatively "Pattern Languages") [DES]:

> *The elements of this language are entities called patterns. Each pattern describes a problem that occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice.—Christopher Alexander*

Brains are pattern recognition machines and with sufficient experience we begin to recognize not only more patterns, but we increasingly discover more complex patterns with experience. This is what separates experts from novices. Bruce Kasanoff has this insight:

> *If all you do is sit in a chair and trust your intuition, you are not exercising much intelligence. But if you take a deep dive into a subject*

> *and study numerous possibilities, you are exercising intelligence when your gut instinct tells you what is—and isn't—important [KAS].*

The practice of Design Patterns is that it captures these patterns (i.e. tacit knowledge or intuition) in a form that is collectively curated and communicable to a much wider audience. Rather than have each individual learn from experience on their own, collective wisdom is captured through Design Patterns. In some sense, Design Patterns are "intuition that is industrialized".

Now that we have these collections of patterns (codified or tacit), how then does this arrive at insight or innovation? What are the ways that we can process patterns? Technology Review reviews a recent research paper "Mathematical Model Reveals the Patterns of How Innovations Arises" [MAT].

In the cited paper, that innovation is enabled by "the adjacent possible" [BRO]. That is those patterns that are one step away from existing learned patterns. So rather than developing patterns that have no connection, new patterns are realized through existing patterns and thus new areas of unexplored patterns are discovered:

> *by providing the first quantitative characterization of the dynamics of correlated novelties, could be a starting point for a deeper understanding of the different nature of triggering events (timeliness, scales, spreading, individual vs. collective properties) along with the signatures of the adjacent possible at the individual and collective level, its structure and its restructuring under individual innovative events.*

Unexplored patterns include either:

Novelties—Patterns that are easily imagined and expected.

Innovations—Patterns that are entirely unexpected and hard to imagine.

Surprisingly enough, the statistical occurrence of innovations shows striking regularities that can be researched in greater depth.

Deep learning is creating a renaissance in Artificial Intelligence. For many long-term practitioners in the field, the sea change is not too obvious. Deep Learning is a departure from classical machine learning methods. One researcher who recognized that our classical approaches to Artificial General Intelligence (AGI) were all but broken is Monica Anderson [GK]

Anderson is one of the early few researchers that recognized that the scientific approach of reductionism was fatally flawed. Reductionism was leading research astray in its quest for AGI. One very good analogy that highlights the difference between Deep Learning and classical AI approaches is the difference between intuition and logic. Dual Process Theory theorizes that there are two kinds of cognition, unremarkably labeled as "System 1" and "System 2".

System 1	System 2
Unconscious Reasoning	Conscious Reasoning
Implicit	Explicit
Automatic	Controlled
Low Effort	High Effort
Large Capacity	Small Capacity
Rapid	Slow
Default Process	Inhibitory
Associative	Rule-Based
Contextualized	Abstract
Domain Specific	Domain General
Evolutionary Old	Evolutionary Recent
Non-verbal	Linked to Language
Includes recognition, perception, orientation.	Includes rule following, comparisons, weighting of options
Modular cognition	Fluid intelligence
Non-logical	Logical
Parallel	Serial

Table 3.0 Dual Process Theory characteristics. Source: https://en.wikipedia.org/wiki/Dual_process_theory [DPT]

Classical AI techniques has focused mostly on the logical basis of cognition, Deep Learning by contrast operates in the area of cognitive intuition. Deep Learning systems exhibit behavior that appears to be biological. This is despite not being based on biological material or being biologically inspired. It so happens that humanity may have discovered Artificial Intuition by pure luck. Deep Learning is Artificial Intuition, and in this book I will use the terms interchangeably.

The argument that Anderson brings forward is that to build systems with capabilities of intuition, one cannot be dependent on constructions that are based on 'Reductionist Methods' (Note: Anderson also acknowledges that Deep Learning is Artificial Intuition). Anderson characterizes Reductionist methods as having the following characteristics [AND]:

Optimality: We strive to get the best possible answer.

Completeness: We strive to get all answers.

Repeatability: We expect to get the same result every time we repeat an experiment under the same conditions.

Timeliness: We expect to get the result in bounded time.

Parsimony: We strive to discover the simplest theory that fully explains the available data.

Transparency: We want to understand how we arrived at the result.

Scrutability: We want to understand the result.

Anderson argues that the logic based approach needs to be abandoned in favor of an alternative 'model-free' approach. That is, intuition based cognition cannot arise from reduction based principles. What Anderson describes a 'model-free' are 'unintelligent components", that is she writes:

> *If you are attempting to build an intelligent system from intelligent components then you are just pushing the problems down one level.*

Anderson proposes several 'model-free' mechanisms can lead to emergent behavior that we see in intuition. I also agree that there are computational mechanisms, that are simple (or unintelligent) in their underlying mechanism, can certainly lead to complex (or intelligent) emergent behavior. This is evident in that even universal computation can arise through the presence of just 3 fundamental mechanisms (i.e. computation, signaling and memory) [PER9].

There are limitations to our present day mathematics that serves as an insurmountable barrier towards progress. This however does not imply that we cannot make progress because of our collective weakness in our mathematical tools. On the contrary, Deep Learning researchers have made considerable progress via experimental computational means. Engineering and practice has well outpaced theory, and this will continue to be the trend.

Variety of Human Intelligences

I've written earlier about the limits of rational thought, which is the notion of time, emergent collective behavior and meta-level reasoning. We thus are in a fork in the road: we can either wait for our mathematical language to make a quantum leap, or we can soldier on with the expectation that there comes a point where analytic techniques have limited capabilities in the realm of universal computation.

My personal bias is that a method must not require global knowledge for its agents to be successful. In other words, all agents must decide on their actions based solely on local knowledge and that emergent global behavior arises through the individual local interactions of its constituents.

In many alternative theoretical models that have been proposed, there usually is an underlying assumption that some global knowledge is required by its agents to function. We are also biased towards methods that have been known to work. As far as I know, Deep Learning is the only Artificial Intuition that has been shown to work very well. The one conceptual problem though of DL is that learning by gradient descent requires global knowledge that gets back propagated to its constituents. Therefore I prefer the term 'localized models' and not 'model free models', even though both mean a measure of lack of intelligence of constituent parts.

In general though, this idea of local only knowledge is related to the concept of 'parsimony', in that we seek the simplest of mechanisms that give rise to emergent properties. The simplest mechanism is the mechanism that knows the least about its context.

This is where I make a segue (or is it quantum leap?) into Deep Learning and that cumbersome concept of generalization. Deep Learning networks are pattern recognition machines. In fact, they are constructed in a self-similar manner where pattern recognition exists at different scales. These networks are self-similar fractal recognition machines. That is, Deep Learning systems consist of collections of pattern recognition machines that are also composed of collection of pattern recognition machines. A recursive structure that terminates at each neuron which themselves are recognition machines.

They are designed to recognize patterns based on previously learned patterns. Generalization, in a general sense (or should I have used "broad" instead?) is the recognition of new unseen patterns. My conjecture here however is that the class of unseen patterns are either of the class that is "easily imagined and expected" or even better, "an entirely unexpected and hard to imagine" class.

That is these are novelties versus innovations. The research paper "**Dynamics on expanding spaces: modeling the emergence of novelties**" indicates that the mechanisms to discover the latter is the same mechanism as that of the former. The same model accounts for both phenomenon [LOR] It seems that the pattern behind the way we discover novelties—new songs, books, etc.—is the same as the pattern behind the way innovations emerge from the adjacent possible.

This is thus something extremely intriguing, if we assume that novelty discovery an intrinsic capability (i.e. generalizability) of DL systems, then perhaps so is innovative discovery. That is, a capability that goes well beyond what I had expected!

To summarize, intuition is a cognitive mechanism that performs massive parallel pattern recognition to arrive at predictions.

Generalization is an emergent behavior that arises through the combination of recognized adjacent patterns. These patterns may be of the novel kind (i.e. previously unseen) or the unexpected kind (i.e. unexpected). I conjecture that we can leverage the concept of the 'adjacent possible' as an abstract explanation for generalization.

One other curious characteristic of intuition is that is has a timelessness quality to it. What I mean is that, when we put our intuition to work, we don't have predictability as to when it reaches a good insight. It's like some backend parallel thinking machine that goes off on its own. How often do we experience discovering new insights by just sleeping on a problem? Our intuition seems to work overtime when our consciousness is not awake. On the flip side, it also works extremely quickly in a manner that is not observable by the conscious mind. This indicates to me that the parallel nature of intuition leaves it unable to accurately make sense of time.

The psychologist Howard Gardner has his Theory of Multiple Intelligence where he describes 8 kinds that humans are theorized to have [THE]. It's the same idea as Marvin Minsky's Society of Mind. Conventional computers that perform mathematical calculations or store memory can be considered as a different kind of intelligence [SOC]. This kind of machine intelligence without a doubt has long surpassed human capabilities in these areas. Nobody will doubt that machines can perform numerical calculations better than humans.

Howard Gardner's list of human intelligences (refer to Wikipedia article for more detail [MUL]):

Musical-rhythmic and harmonic, aka Musicality

This area has to do with sensitivity to sounds, rhythms, tones, and music.

Visual-spatial, aka spatial intelligence

This area deals with spatial judgment and the ability to visualize with the mind's eye.

Verbal-linguistic, aka Linguistic intelligence

People with high verbal-linguistic intelligence display a facility with words and languages.

Logical-mathematical, aka Reason

This area has to do with logic, abstractions, reasoning, numbers and critical thinking.

Bodily-kinesthetic, aka Gross and Fine motor skills

Control of one's bodily motions and the capacity to handle objects skillfully.

Interpersonal, aka Social skills

This refers to sensitivity to others' moods, feelings, temperaments, motivations, and their ability to cooperate in order to work as part of a group.

Intrapersonal, aka Introspection

This area has to do with introspective and self-reflective capacities.

Naturalistic, aka Ecological receptiveness

This sort of ecological receptiveness is deeply rooted in a "sensitive, ethical, and holistic understanding" of the world and its complexities—including the role of humanity within the greater ecosphere.

Gardner adds a few more such as spiritual intelligence and teaching-pedagogical intelligence [THE].

Somewhat related to this is a paper by Brendan Lake et al. "Building Machines that Learn and Think like Humans" [LAK] that focuses on subset of Gardner's 8 intelligences as where research should focus on to build more capable machines. The researchers focus on psychological and physical intuition as skills that need to be achieved by machines.

A good graphic may be instructive that shows how multiple intelligences and various cognitive mechanisms are related in humans:

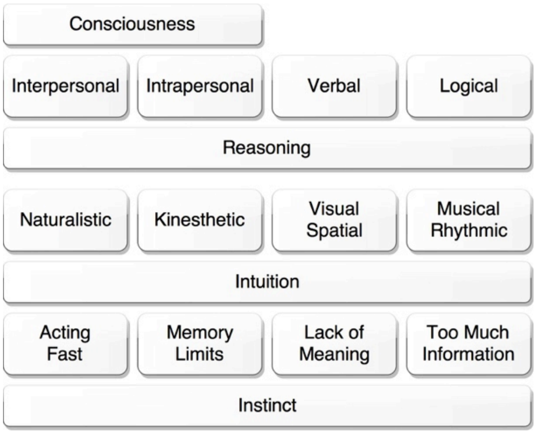


Figure 3.1 Cognitive stack

Automation is incapable of many of the intelligences that are listed above. However, as you can see from this "stack", the relationship between the capabilities of an intuition machine and that of higher order intelligence capabilities. Human brains are not built from the ground up like machines that have precise numerical capabilities. Human brains use an entirely different substrate that is built from instinct and intuition. It is the conjecture of this book that artificial intuition was discovered in the form of Deep Learning and that the road to more capable machines is through this route.

Cormac McCarthy proposed one interesting conjecture, that relates our own language capabilities, in his paper "The Kerkule Problem – Where does language come from?" [MCC] McCarthy argues that human language is not an intrinsic capability of the human brain and that humans did not evolve to develop the specialized infrastructure to support abstract thought. Rather that the human brain actually is forever compensating to enable itself with the skills of language. McCarty writes:

> *The unconscious is just not used to giving verbal instructions and is not happy doing so. Habits of two million years duration are hard to break.*

So what we have here is something that is counter-intuitive but seems to support the recent experimental evidence in Deep Learning. That is, a system that is built for intuition, more specifically visual intuition, can be quite capable in performing sequential reasoning or perhaps logical thought.

Learning using Intuition

There is a learning method that is attributed to Richard Feynman (aka "The Great Explainer") coined the "Feynman technique" [FEYN]:

- Pick and study topic
- Pretend to teach your topic to a student
- Go back to the literature when you get stuck
- Simplify and use analogies

I don't think that Feynman had explicitly described a "Feynman Technique", but there are some hints that he had preferences to many aspects of this learning approach. Biographer James Gleick in his book "Genius: The Life and Science of Richard Feynman." [GLE] Gleick writes:

> *"[He] opened a fresh notebook. On the title page he wrote: NOTEBOOK OF THINGS I DON'T KNOW ABOUT. For the first but not last time he reorganized his knowledge. He worked for weeks at disassembling each branch of physics, oiling the parts, and putting them back together, looking all the while for the raw edges and inconsistencies. He tried to find the essential kernels of each subject."*

Which describe a deconstruction and then construction approach to understanding a topic. This method of course is not unusual. Feynman himself (an excerpt of the book published in a 1996 issue of Caltech's Engineering & Science magazine) remarked though about the need to explain complex ideas using simpler concepts:

> *Feynman was a truly great teacher. He prided himself on being able to devise ways to explain even the most profound ideas to beginning*

> *students. Once, I said to him, "Dick, explain to me, so that I can understand it, why spin one-half particles obey Fermi-Dirac statistics." Sizing up his audience perfectly, Feynman said, "I'll prepare a freshman lecture on it." But he came back a few days later to say, "I couldn't do it. I couldn't reduce it to the freshman level. That means we don't really understand it."*

Finally, written on his blackboard when he passed away in 1988:

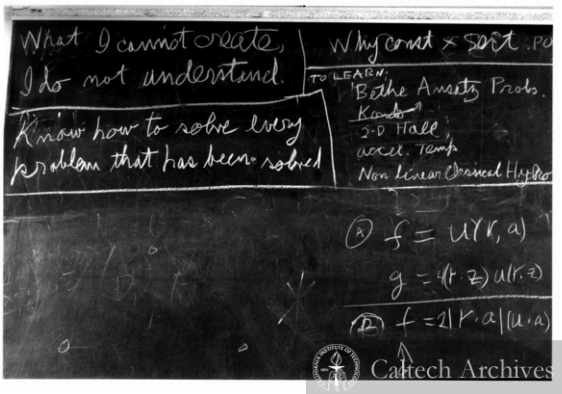


Figure 3.2 Richards Feynman's blackboard Source: http://archives-dc.library.caltech.edu/islandora/object/ct1%3A483

It reads "What I cannot create, I do not understand." Again emphasizing the requirement of generating an explanation from likely first principles.

To be subsequently quoted by OpenAI researchers to describe motivation of their Deep Learning [generative model approach]() [OAI]. The point though that they are making is that there seems to be a connection with understanding and the ability of re-generating a concept or idea. Although, GANs can generate realistic images, I highly doubt that GANs can understand any of it!

However, let's go back to the method, since it indeed is illuminating in how the human mind works in learning as well as potentially how a Deep Learning system (a machine) may also improve its learning.

There are three key elements of the method that is worth focusing on:

- The activity of explaining some complex topic. (To generate)
- The disassembly of a complex topic into simple terms. (Analogies perhaps?) (To compose)
- The reconstruction of the topic looking seeking any inconsistencies. (To compose and validate)
- The iterative nature of the method. (To iterate and refine)

I have this thesis that humans and deep leaning machines are both intuition machines. What I mean by this is that evolution has bequeathed humans with a cognitive machine that reasons through induction and analogy. Our evolutionary machinery does not have specialized symbolic and deductive reasoning capabilities. Humans simply have not had sufficient evolutionary time to evolve this capability. Rather, our intuition machinery is forever compensating in an inefficient manner to perform rational thought.

Unlike computers that can natively understand symbolic forms and can perform deductive reasoning at blinding speeds, our brains don't have these specialized capabilities. This is one reason for the failure of the GOFAI approach to AI, where there was an assumption that human thought can be built up from formal reasoning.

However, having evolved to be social beings, humans are equipped with machinery that enables us to effectively function within social groups. Our brains allow us to (1) understand the behavior of members of our social group and (2) communicate and share our thoughts with members of our social group. Over time, humans have developed language that has persisted and evolved through many human generations. Our brains have learned to understand and communicate in the languages that we have been taught.

Language is basically information compression. Actually, the ability to generalize can be framed in terms also of information compression. Our ability to express our thoughts through language can also be a measure of our intelligence.

Our brains have different kinds of intelligence. The higher levels of this stack (i.e. logical, interpersonal, verbal and intra-personal) are supported by more primitive cognitive capabilities (i.e. visual-spatial, rhythmic, sensory and motion). Our brains have built only approximations of these higher-level cognitive capabilities. We pretend that we are indeed rational, but it actually takes us a lot of mental energy to work our way through a rational thought process. Our natural tendencies are to employ our intuition to make fast and sometimes error prone (or biased) judgments.

The point I want to make here is that, to learn something well, humans and intuition machines have to learn it at a basic visual-spatial, rhythmic, sensory level. Knowledge just doesn't stick when conveyed as a symbolic and logical level. Our brains fundamentally can't understand abstractions. Rather, our brains have metaphors of abstractions that are captured by very primitive constructs. According to cognitive linguist, George Lakoff:

> *You can only have meaningful thought through connections to the body.*

If you have ever listened to one of Feynman's many lectures, you will find how he takes a great effort of using analogies to explain complex topics. He engages our intuition to its fullest. He made popular the use of Feynman diagrams; a visual notation that represented complex particle physics interactions (see tensor networks).

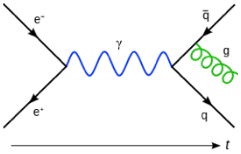


Figure 3.3 Feynman Diagram. Source: https://en.wikipedia.org/wiki/Feynman_diagram

In addition, his approach to explaining dynamics employed the use of the idea of "path integrals", also a highly intuitive representation of dynamics. Something that in fact can also be expressed visually:

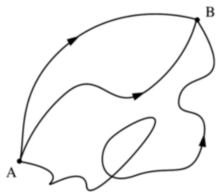


Figure 3.4 Feynman Path Integral https://en.wikipedia.org/wiki/Path_integral_formulation

Simply explained the dynamics of a system can expressed as the aggregation of the alternative paths from point A to point B. Clearly Feynman understood the value of engaging our intuition in the study of advanced abstract concepts.

The Feynman technique takes advantage of the cognitive tools that we humans are good at. That is, exercising our ability to understand simple conceptual ideas coupled with our learned ability to explain things through language. That is why, it is very important in this method to engage the communication ability of the brain. You actively change your perspective by trying to explain it to someone else (even if it is pretend). This engages the mind to the think in a certain way.

The act of explaining something is a natural way of model building for the mind. It places oneself from the perspective of another person. When we explain something we simultaneously attempt to understand the ideas while also sensing if the other person understand what we say. The engagement is a more intense mental activity than say, simply highlighting sections of text in a book. Our minds are simply chaotic systems where our consciousness takes great pains to manage. We can't learn thing by just reading, we have to involve ourselves in more active engagement by the process of actually re-creating what we studied. There is no learning without effort and there is no effort without engagement.

However, we can very easily fall into the trap of the "Cargo Cult." This is another idea that Richard Feynman came up with. This is why "First Principle" thinking is extremely important. The above technique helps you understand complex topics, however it does not mean that your understanding is correct! Feynman said:

The first principle is that you must not fool yourself—and you are the easiest person to fool.

That is, one needs to examine the concepts that are used in one's explanation and determine if you can break down these concepts to irreducible concepts. One should verify the validity of each irreducible concept (usually applicable only in a few domains!). The validation part, is why Elon Musk remarked:

> *"I think it's important to reason from first principles rather than by analogy"*

It is easy to think in analogy because that is what we are wired to do. However, in many advanced scientific and technology areas, many concepts are counterintuitive. We simply cannot assume that reality operates in the same way that our primitive minds are comfortable with. Thinking by analogy serves us very well in surviving in the jungles, but can be a problem with complex subjects.

The first principle, or rather "Feynman's First Principle" is what gives grounding to our intuitive thinking. Intuition allows us to explore multiple alternative paths simultaneously to arrive at new thought patterns. However, it may contain errors and thus re-validation of our thoughts through higher rigor and rational thinking is a must.

The Indian mathematician Ramanujan, had an immense mathematical intuition that gave him the uncanny ability find mathematical generalizations of infinite series without rigorously deriving the details. Ramanujan's intuition was unparalleled, he had no formal advance mathematics training and he was self-taught. Ramanujan had a exceptional intuition with numbers, however there were times where his intuition could only "see" so far. That is, certain infinite series will work out for the first hundred instances, but eventually break down. Ramanujan had developed a mind with an unimaginably advanced intuition with respect to abstract algebra. However, advanced, that intuition had its limitations. Those limits could be verified more rigorous rational thought (i.e. rigorous hand calculation).

This is the nature of intuition, it can be creative and fast, but at the same time fallible. The most gifted "computing machines" of our species are still unable to perform computations like computers do, rather they've developed their own intuition to perform unexplainable acts of mental gymnastics.

Neil Lawrence argues [LAW] that the difference between human and machine intelligence comes down to embodiment factors. That is the ratio of the ability compute power over the communication bandwidth. Humans are not blessed with telepathy or the ability to mind-meld. Rather, we are restricted to language to communicate with other. In contrast, machines can interface through high-bandwidth channels and have massive computational capabilities. The embodiment factor that Lawrence ascribes to human is 10^{16} and for machines it is a mere 10.

So as a measure of progress of human intelligence of a machine, a machine needs to be trained with the constraints of a high embodiment factor. That is, a machine needs to be able to explain its thought processes. Alternative way of saying this is, a machine must be able to perform sufficient generalization that it can explain its conclusion in a constrained sequential language.

Natural Stupidity

Do you know what's more dangerous than artificial intelligence? Natural stupidity. In this article, I will explore natural stupidity in more detail and show how our current technology (driven by narrow artificial intelligence) is making us collectively dumber.

We've all had this experience of using a GPS to guide us around an unfamiliar place only to realize later that we have no recollection or ability to get to that place again without the aid of a GPS. Not only is our directional instinct diminished because of lack of use, but so is our own memories. We've all experienced losing our ability to recall due to our over use of Google. We now recall more as to how we can search for something rather than the details of that something.

The framework that I often use to explore intuition is the [Cognitive Bias Codex](https://en.wikipedia.org/wiki/List_of_cognitive_biases) [LIS] found at Wikipedia. It's a massive list of biases, however to get an overview of it, there are four high level categories that are the drivers of theses biases. These are "Too Much Information", "Not Enough Meaning", "Need to Act Fast" and "What Should we Remember?".

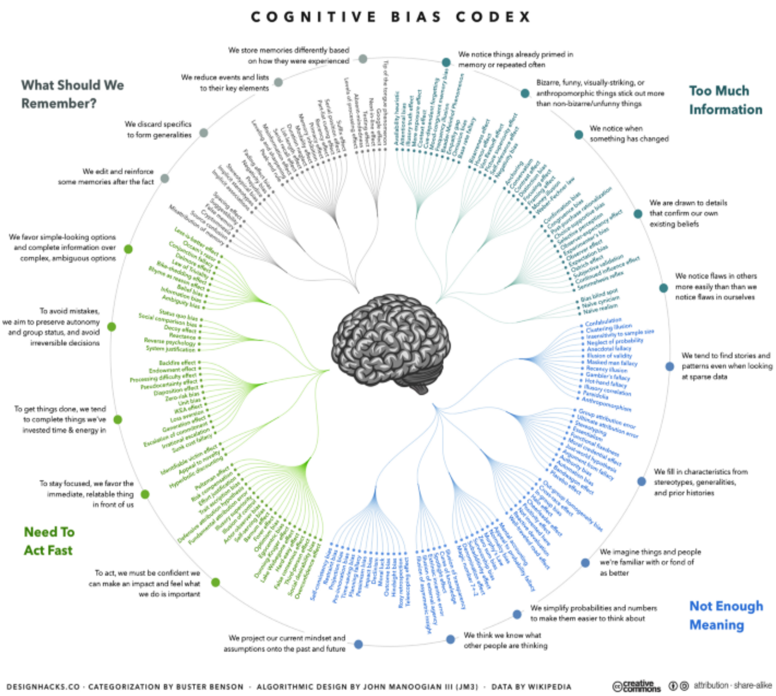


Figure 3.5 Cognitive Bias Codex https://en.wikipedia.org/wiki/List_of_cognitive_biases

Our world requires more automation to run efficiently and sustainably. The products and services that will be in demand are the products that compensate for our inadequacies. The clear downside of this is that with every assist, the less we exercise our already weak facilities.

The only people maintaining their smarts are the few people willing to constantly exercise their smarts. Meanwhile, we have a population that is becoming more out of shape and lazy with their own mental faculties. We imagine ourselves to being smarter because we can multi-task more. Yet, our brains have not evolved to do multi-tasking well. In fact, recent research have shown that pigeons have greater multi-tasking capabilities than humans [PGN]. It is just ironic that we've taken pride in our new found multi-tasking skills only to discover that we are dumber at it than pigeons!

However, there is a far worse problem than automation making us dumber. The bigger problem is that other humans are aware that it can make us dumber and they are opportunistically exploiting our natural stupidity to influence our behaviour. Over the decades, the industry of advertising has spend trillions of dollars inventing new ways to "motivate" us to do new things without us being aware of its influence. The techniques to do this neatly falls under the exploitation of our cognitive biases. After all, if we were indeed all perfectly logical, then we'll likely spending our money in the most efficient way possible and very few companies will like us to do that. If we reduce our spending, our economies would stall and there would be an economic depression! (BTW, something is really wrong when we must accelerate our consumption so as to avoid economic stagnation)

So, "Natural Stupidity" is basically our lack of meaning, lack of memory, inability to think fast and inability to process too much information. The current systems that we have in place provide products and services to substitute these inabilities. It is the natural tendency to seek out the method of least action. That is, the method that requires lest effort or the laziest thing that we can do. Let's explore each of the four in greater detail.

Humans from the beginning of the their life are driven to seek meaning. The simplest explanations to this are going to be the most natural appealing ones. Civilization will naturally create religion to not only create a necessary shared understanding of acceptable behaviour but one that is driven by our need for meaning.

The written word (i.e. books) and its more advanced form, the world wide web are devices that address our limited memories. Memories require not only storage but also the capability of recall. Throughout history, religion and law has been transcribed in scrolls, books and now in automation (see: blockchain). Money is a form of memory, that is, once possession of it is a measure of one's ability to acquire goods and services.

Mankind created computers to automate the math that we invented. Computers not only store memories but are able to perform laborious and error-free computations. We find it an inconvenience to use cash in that we have to calculate in our minds the amount of change so as to guard against error or outright fraud. We have time keeping devices so that we don't need to look out into the heavens to determine the time of day. We have GPS devices to help us avoid reading a map and calculating a path to our destination.

Finally, we have the problem of information overload. Our knowledge driven economies have accelerated our consumption of information. However, our brains have not magically evolved to process this fire-hose of information. The device that we use to process more information are services that curate information and exhaust it out in more easily digestible forms. Today, social networks such as Twitter and Facebook have become our primary tools for curating and receiving new information about the world. It is dumfounding that the leaders of these two companies believe it is not in their charter to 'police' the contents that they help propagate. With great power comes great responsibility, unless it I guess if it affects the bottom line!

We collectively become dumber when we relinquish responsibility and accountability to the automation (or A.I.) that furnishes us with cognitive assistance.

When we avoid questioning the positions of our religious leaders and ignore obviously repugnant behaviour in defence of our own beliefs.

We avoid verifying our history and cling to untrue historical information to justify our beliefs. This is the case for Neo-Nazis and Confederates who would like to imagine a more benevolent and just past.

We ignore common sense by following algorithms in enforcement of procedures. Like the United Airlines where a 70 year old doctor was assaulted and removed from an airline just because the crew blindly followed protocol instead of their own common sense.

Finally, we don't hold accountable organizations that employ information overload in the form of massive disinformation to mould public opinion.

Weaponized AI

On July 28th, 2015 a group of AI scientists, lead by Max Tegmark, published an open letter concerning the creation of weaponized AI.

Open Letter on Autonomous Weapons - Future of Life Institute [OPE]

They write the following:

> *The key question for humanity today is whether to start a global AI arms race or to prevent it from starting. If any major military power pushes ahead with AI weapon development, a global arms race is virtually inevitable, and the endpoint of this technological trajectory is obvious: autonomous weapons will become the Kalashnikovs of tomorrow. Unlike nuclear weapons, they require no costly or hard-to-obtain raw materials, so they will become ubiquitous and cheap for all significant military powers to mass-produce. It will only be a matter of time until they appear on the black market and in the hands of terrorists, dictators wishing to better control their populace, warlords wishing to perpetrate ethnic cleansing, etc. Autonomous weapons are ideal for tasks such as assassinations, destabilizing nations, subduing populations and selectively killing a particular ethnic group.*

Which lead to some research on "Concrete Problems in AI Safety" [AMO]. Unfortunately, by November 2016 a different kind of weaponized AI had arrived with devastating effects: "How We Broke Democracy (But Not in the Way You Think)" [ROS]:

> *This contact is important in the context of our social channels. They are designed to let us insulate ourselves from the people and opinions we would prefer not to see.*

In "Blue Feed, Red Feed" *[KEE]* the author writes:

> *Facebook's role in providing Americans with political news has never been stronger—or more controversial. Scholars worry that the social network can create "echo chambers," where users see posts only from like-minded friends and media sources.*

These "filter bubbles" or "echo chambers" are primed staging areas for further exploitation. Persuasive Machines are AI driven automation, leveraging knowledge of our cognitive biases to hack into our thought processes, leading to persuasion (or alternatively mind control) [LIS]. An example of this exploit is the rampant rise of fake news and events on Facebook [PIC]. Mike Caulfied writes, "Despite Zuckerberg's Protests, Fake News Does Better on Facebook Than Real News. Here's Data to Prove It." [CAU]:

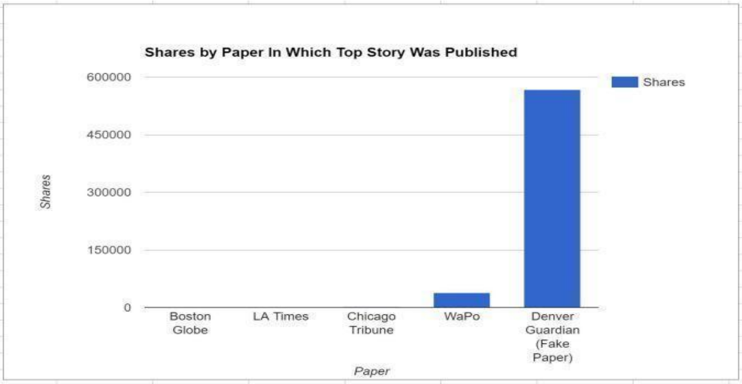


Figure 3.6 Fake news sources (source: https://hapgood.us/2016/11/13/fake-news-does-better-on-facebook-than-real-news)

Fake news is orders of magnitude more popular than real news. It is not just fake news that's a problem, security breaches are also another. Maria Korolov writes [KOR][CHA]:

> *"It's a combination of a lot of things that we've seen for a lot of years coming together," said Ric Messier, head of the cybersecurity program at Burlington, Vt.,-based Champlain College. "The fact that it's so easy to do this leaking and be able to manipulate people in this way certainly suggests that we're probably just starting to see the beginning of these sorts of activities or attacks".*

Automation in the form of Twitter bots have amplified messaging to bring about a 'bandwagon effect' to influence the masses. In "**The Algorithmic Democracy" [CAM]:**

> *Analyzing Twitter during three televised debates, they discovered that 20% of all political tweets were made by bots. … These bots, they wrote, can make online conversations more polarized. They make it easier to spread factually incorrect news stories. And they are easy to make: Nearly anyone "could obtain the operational capabilities and technical tools to deploy armies of social bots and affect the directions of online political conversation" .*

These automatons, in combination with massive hacking and phishing attacks by a nation state were part of a coordinate effort to undermine the decision making of the population [SZO]. Esquire magazine has an even more detailed account of this in "How Russian Pulled the Greatest Election Hack in History"[RID].

Jonathan Albright has an even more detail analysis of what he calls: "Micro Propaganda Machines" *[ALBR].* Albright writes:

> *There's a vast network of dubious "news" sites. Most are simple in design, and many appear to be made from the same web templates. These sites have created an ecosystem of real-time propaganda: they include "viral" hoax engines that can instantly shape public opinion through mass "reaction" to serious political topics and news events. This network is triggered on-demand to spread false, hyper-biased, and politically-loaded information.*

Scott Adams (of Dilbert fame) understands the power of persuasion quite well in "No One Understands Donald Trump Like the Horny Narcissist Who Created Dilbert" [DOL]:

> *Adams, like Trump, recognized that the election would play out at the limbic level of primal furies and genital anxieties .*

Dylan Love writes about solutions to solving the AI arms race in "The Next Global Arms Race Aims to Perfect AI" [LOV]:

> *Make AI development illegal, but history suggests that prohibition*

> *doesn't work.*
>
> *Win the AI race.*
>
> *Assemble an international AI consortium, with many nations pooling their resources.*
>
> *Every nation unites under one flag and one leadership.*

Not a very promising list of options. The threat of AI is not some futuristic idea; it is indeed here today and will become ever more sophisticated. In a latter chapter, we will further discuss this dimension of artificial social intelligence.

Human Centricity Bias

Humans are very good at talking themselves out of investing heavily on new and disruptive technologies. Here are some of the reasons that experts around you will use to talk you out of investing in Deep Learning.

It's just Machine Learning

Practitioners' introduction to neural networks is usually via the introduction of linear regression which then moves to logistic regression. That's because the mathematical equations for an artificial neural network (ANN) are identical. There is therefore immediately a bias that the characteristics of these classical ML methods would also convey into the world of DL. After all, DL is, in its most naive explanation, nothing more than multiple layers of ANN.

There are also other kinds of ML methods that have equations that are different from DL. The basic objective for all ML methods is, however, a general notion of curve fitting. That is, if the data is a good fit with a model, then that is perhaps a good solution. Unfortunately, with DL systems, these systems will by default over-fit any data, because the number of parameters in the model is so large. This is enough of an indication that a DL is an entirely different kind of animal from an ML system.

It is just Optimization

DL systems have a loss function that is a measure of how well its predictions match its input data. Classic optimization problems also have loss functions (also known as objective functions). In both systems, different kinds of heuristics are used to discover an optimal point in a large configuration space. It was once thought that the solution surface of a DL system was sufficiently complex so that it would be impossible to arrive at a solution. Curiously enough however, one of the simplest methods of optimization, the Stochastic Gradient Descent algorithm, is all that is needed to arrive at surprising results.

What this tells you is that there is something else going on here that is actually very different from what optimization folks are used to.

It's a Black Box

Many Data Scientists have an aversion for DL because of the lack of interpretability of its predictions. This is a characteristic of not only DL methods, but classical ML methods as well. Data Scientists would rather use Probabilistic methods where they can have better control of the models or priors. As a result, they have systems that are able to make predictions with the least number of parameters. All are driven by the belief that parsimony, or Occam's razor, is the optimal explanation for everything.

Unfortunately, probabilistic methods are not competitive in classifying images, speech, or even text. That's because DL methods are superior to human beings in discovering models. Brute force just happens to trump wetware. There's no experimental evidence in the DL space that parsimonious models work any better than entangled models. For those cases where it is an absolute requirement to have some kind of explanation, there are now newer methods in DL that provide aid to interpretability, as well as uncertainty. If a DL system can generate the captions in an image, then there is a good chance that it can be trained to generate an explanation of a prediction.

It's too early and too soon

There is a natural bias that something that is around 5 years old and rapidly evolving is too new and volatile a technology to trust. I think we all said the same thing when the microprocessor, internet, web and mobile technologies came along. Wait and see was the safe approach for most everyone. This is certainly a reasonable approach for anyone who has not really spent the time investigating the details. However, it is a very risky strategy, ignorance may be bliss, but another company eating your lunch can mean extinction.

There is too much hype

There are a lot things that DL can do that were deemed inconceivable just a few years ago. Nobody expected a computer to beat the best human player in Go. Nobody expected self-driving cars to exist today. Nobody expected to see Star Trek universal translator-like capabilities. It is so unbelievable that it must likely be an exaggeration, rather than something that may be real. I hate to burst your bubble of ignorance, but DL is in fact very real, and you experience it yourself when using any smartphone.

AI winter will likely come again

We've had so many times where the promise of AI has led to disappointing results. The argument goes further that because it has happened so often before, it is bound to happen again. The problem with this argument is that despite disappointment, AI research has led to many software capabilities that we take for granted today and thus never notice its existence. Good, old-fashioned AI (GOFAI) is embedded in many systems today.

The current pace of DL development is accelerating and there are certainly big problems that need to be solved. The need for more training data and the lack of unsupervised training are two problems. This however doesn't mean that what we have today has no value. DL can already drive cars. That in it tells you that even if another AI winter arrives, we would have achieved a state of development that is still quite useful.

There is not enough theory of how it works

The research community does not have a solid theoretical understanding as to why DL works so effectively. We have some idea as to why a multi-layer neural network is more efficient in fitting functions than one with fewer layers. However, we don't have an understanding as to why convergence even occurs, or why good generalization happens. At this time, DL is very experimental and we are just learning to characterize these kinds of systems. Meanwhile, despite not having a good theoretical understanding, the engineering barrels forward. Researchers, using their intuition and educated guesses, are able to build exceedingly better models. In other words, nobody is stopping his or her work to wait for a better theory. It is almost analogous with what happens in biotechnology research. People are experimenting with many different combinations and arriving at new discoveries that they have yet to explain. Scientific and technological progress is very messy and one shouldn't shy away from the benefits because of the chaos.

It is not biologically inspired

DL systems are very unlike the neurons in our brain. The mechanism of how DL learns (i.e. SGD) is not something we can explain using what happens in our brain. The argument here though, is that if it doesn't resemble the brain then it is unlikely to be able to perform the kind of inference and learning of a brain. This is of course an extremely weak argument. After all, planes don't look like birds, but they can certainly fly.

I'm not an expert in it

Not having in-house expertise shouldn't be an excuse for avoiding finding expertise outside. It also should not prevent you from having your experts learn this new technology. If these experts are however have the dogmatic persuasion, then that should be an indication that you should get a second and unbiased opinion.

It does not apply to my problems

Businesses are composed of many business processes. Unless you have gone through the exercise of examining the processes that can be automated with current DL technologies, you are not in a position to make the statement that DL does not apply to you. Furthermore, you may discover new processes and business opportunities that may not exist today, but are possible with the exploitation of DL technology. You cannot really answer this question until you have invested in some due diligence work.

I don't have the resources

The large Internet companies like Google and Facebook have gobbled up a lot of the Deep Learning talent out there. These companies have very little interest in working with a small business to identify their specific needs and opportunities. Fortunately, these big companies have been gracious enough to allow their researchers to publish their work. We do therefore have a view into their latest developments and are thus able to take what they've learned and apply it to our context.

Summary

Perhaps you noticed that his chapter is indeed quite odd in that the book is supposedly about a new kind of computer technology, yet this chapter discusses the nature of intelligence and thus deals with how humans think. Indeed, we are forced here to examine intuition and as a consequence we cannot avoid the existence of cognitive bias. Awareness of cognitive bias tells us that it can actually be exploited by automation. Furthermore, in examination of any new technology, we have to introspect into our own thinking to identify if biases are hindering our own progress.

4 Learning the Unknowable

"The intellect has little to do on the road to discovery. There comes a leap in consciousness, call it intuition or what you will, and the solution comes to you and you do not know how or why. All great discoveries are made in this way"

- Albert Einstein

In this chapter we explore in more detail the structure and composition of Deep Learning systems. As previously defined, Deep Learning is a Connectionist approach to Artificial Intelligence. Deep Learning specifically employs multiple layers of components arranged in a graph without loops. There can be many kinds of layers but the common property on any layer is that it is differentiable. This is a consequence of the learning mechanism (i.e. gradient descent).

Unknowable Knowns

One good way to frame the question of the limits of Deep Learning is in the context of the Principle of Computational Equivalence by Stephen Wolfram. Wolfram showed that simple cellular automations are able to exhibit complex behavior that cannot be predicted from initial conditions or the simple rules that specify its incremental behavior. Certain kinds of cellular automata can exhibit complex behavior that cannot be reduced to a mathematical model that captures its behavior in closed form. Wolfram examples of an 'irreducible' system that exhibits this complex behavior are the brain and weather systems. Wolfram classifies these kinds of systems as exhibiting "Universality" [WEW].

A Deep Learning system that has memory belongs to this class of universal machines, however this does not imply that these systems can replicate the behavior of another universal machine. Bernhard Scholkopf reveals this conclusion in a paper "Towards a Learning Theory of Cause and Effect" [LPD]. That is, a learning system is able only to derive the cause while observing the effect. This tells you that a Learning system can't learn the mechanisms of say, how a DNA manufactures specific proteins. A fundamental limitation of any system that learns from data is that it cannot predict effect from cause. This theory is analogous to the "Halting Problem" in Computability Theory.

One criticism that I hear often about Deep Learning is that it doesn't capture the biological mechanisms of the brain. This is a fair criticism. However, Wolfram's Universality explains why it should not be a major issue toward achieving AGI. Deep Learning systems can possibly have equivalent capabilities as a biological brain albeit by using different computational mechanisms. At the fundamental level, all these systems are computational systems that exhibit three building blocks. That is, computation, memory and signaling. Complex behavior is an emergent behavior that, like cellular automata, arises from very simple rules.

A useful schema in understanding the capabilities of a system to learn or discovery unknowns can be stated as follows:

Knowable knowns, meaning models that will converge on training data. **Knowable unknowns**, are trainable models that are able to make accurate predictions on non-training data. **Unknowable knowns**, reflects an inability to learn a known system, this covers the area of performing predictions of other irreducible universal machines. **Unknowable unknowns** is the inability to discover what a machine does not know. We break this down in more detail:

(1) Knowable Knowns - Given a large enough set of knowns (i.e. training data) we can get good convergence of our prediction errors.

(2) Knowable Unknowns - This is an expression of the concept of generalization. With good generalization, we can know about test data that a machine has never encountered in its training set.

(3) Unknowable Knowns - However, there are certain classes of system that deep learning can never learn. These are in the class of computational irreducible systems. However a Deep Learning system may detect the direction of causality and be able to know that is able to learn from the system. What it will not be able to know (unknowable) is if a system is in the class of computational irreducible systems.

(4) Unknowable Unknowns - Finally, there is a class of total ignorance. This is really a metaphysical statement. The class of unknowables that cannot be known is unknowable. Think of it as the "Great Firewall" of knowledge.

What is Knowable?	*Knowns*	*Unknowns*
Knowable	Trainability	Generalization
Unknowable	Universality	"The Great Firewall"

The reader may have encountered a similar classification before by an infamous defense secretary. That is, "Known knowns", "Known unknowns", "Unknown knowns" and "Unknown unknowns". "Known knowns" are what we currently know. "Known unknowns" are what we know that we don't know. "Unknown known" is a reflection of willful ignorance or what politicians may call "alternative facts". "Unknown unknowns" is simply ignorance of what one does not know. A good cliché is the "Black Swan." That is the belief that black swans don't exist when they in fact do. This schema differs in that the "knowable" schema in that this expresses the current state of understanding rather than an ability to learn.

What is the Current State of Knowledge?	*Knowns*	*Unknowns*
Known	Current Facts	Identified Unknown Facts
Unknown	Willful Ignorance	"Black Swan"

For the study of learning machines it is more important to understand what is knowable rather than the current state of knowledge.

Even though the above schema of ignorance looks complicated, it is but the tip of the iceberg of understanding ignorance. Consider more complexities such as misinformation, detecting model bias, ambiguity, disagreement, and accommodating for change.

Characteristics of Artificial Intuition

Let's explore further the characteristics of Artificial Intuition with the goal of describing a set of patterns that can aid us in formulating novel architectures in Deep Learning. Previously, I introduced the idea that there are two distinct cognitive mechanisms, one based on logical inference and another based on intuition. At least 6 decades have been spent exploring cognitive mechanisms based on logical inference without making much progress towards AGI. Deep Learning, a breakthrough discovered in 2012, revealed an alternative promising research approach based on the different cognitive paradigm.

In the field of Psychology, Kahneman and Tversky researches the interplay of these two kinds of cognitive function in a book "Thinking, Fast and Slow" [HOL]. The book has been highly praised:

> *New York Times columnist David Brooks recently declared that Kahneman and Tversky's work "will be remembered hundreds of years from now," and that it is "a crucial pivot point in the way we see ourselves." They are, Brooks said, "like the Lewis and Clark of the mind".*

Kahneman's book explores human cognitive biases and employs the dual cognitive processes as a root cause of these biases. In this section however, I will be exploring system 1 (i.e. intuition), more specifically artificial intuition and the mechanisms that give rise to it.

Deep Learning has a long history. The approach originates from the Connectionist approach and derives much of its philosophy from ideas found in the Complexity sciences. In a nutshell, the idea is that emergent complex behavior can arise from simple mechanisms. Chaos and complexity are the two driving forces that exist in complex systems.

Our goal then is to either explain or better understand how emergent features arise through chaos and complexity. Here are some key features, and some questions that require good answers:

Self-Organization: How does a system self-organize itself so that behavior required for survivability is encouraged and destructive behavior discouraged? How does complex organizational structure arise from simple structures?

Robustness: How does a system organize itself to become more tolerant to failure? How does a system gain the adaptability required to survive in unexpected environments?

Diversity: Adaptability and survivability requires diversity that may be less optimal than a homogeneous solution. Mixture of experts or ensemble methods emphasizes the value of diversity in improving predictability.

Abstraction: How does a system learn the abstractions required to perform accurate predictions in a hostile complex environment? How does generalization arise from the learning of abstractions?

Adaptation: What mechanisms of adaptation are necessary to compensate for incorrect predictions? How can a system forget learned behavior that may be detrimental to its survival?

Bounded Prediction: How can computational resources required for predictions are bounded? Predictions must be made in a timely manner important for survivability. How can a system learn to optimize its predictions to fit within fixed bounds?

Coordination: How can a system learn to coordinate its actions with other participating actors? An environment not only includes inanimate objects, but also other systems that have learning capabilities. How can a system not only learn its environment but also learn how to interact with other learning systems?

These features of complex adaptive systems all relate to a previous discussion on "Ilties". That is Expressibility, Trainability and Generalizability. One of the clear traps that exist among practitioners is that we can inadvertently bring in detrimental methods that originate from our mathematical or engineering training. That is, we take ideas such as the need for optimality, the requirement for sparse solutions, the need for interpretability and understandable solutions, the need for completeness and repeatable guaranteed behavior. These needs are of course desirable, however we should not optimize for these as a starting point. This leads to premature optimization, an idea that we are all familiar with in computer science. Rather, we should all embrace first complexity and chaos and work out solutions that holistically incorporate these as a given. Research in Deep Learning is a major paradigm shift and thus requires a different kind of thinking.

The first big conceptual leap that we have to make is to understand that learning systems evolve in non-equilibrium settings. Stated in a different way, researchers should be very cautious about employing statistical or alternatively bulk thermodynamic metrics in their analysis of these systems. It is our belief that one of the most glaring inappropriate tools in the study of AI is the use of Bayesian methods. I can understand its utility in the domain of logical inference, however I doubt its effectiveness in a domain of intuitive systems.

The second conceptual leap is that understanding of "Generalization" is quite grossly inadequate. The use of the term in Machine Learning is extremely liberal. Furthermore, the Machine Learning approach of 'curve fitting' and thus interpolation and therefore generalization between adjacent points in the fitted curve, breaks down under the recently discovered notion of rote-memorization of Deep Learning. How rote memorization can lead to generalization is a fuzzy idea at best. In fact, Kahneman's research points out that human cognitive biases exists because of flawed reasoning in our intuitive system 1 inference. Said in other words, very poor and flawed generalization. To conclude, rote memorization leads to a kind of generalization that is inherently flawed!

A third conceptual leap is to accept that Deep Learning systems may be computational systems just like Von Neumann computers. The primary difference is that there is a discovered mechanism (SGD) for these systems to learn from data as opposed to computers that require programmers. Neural Networks are usually treated as continuous dynamical systems. Deep Learning systems have one common requirement, in that the computational layers must be differentiable. Computers by contrast do not have differentiable subcomponents. Cellular Automata (used in Evolutionary paradigms) also do not have differentiable components. Cellular Automata and Genetic Algorithms aren't as successful in learning from data as DL. Yet, if all DL does is rote memorization, and then they aren't very different from Von Neumann computers. At the core though, DL consists mostly of threshold units that are not that remotely distinct from NAND/NOR gates that we find in logic circuitry. Why then is differentiability such an important requirement for trainability? Are continuous dynamical systems a real requirement or are we overlooking a more general principle? Why does SGD lead to learning? As we track the latest research in DL, we are beginning to discover that DL looks more and more like Von-Neumann computers (see: "Conditional Logic" [COND]) and less like the simple dynamical systems we find in Physics. I think we can draw some inspiration in the complexity of 'random boolean networks' that describe biological processes [BOO].

There remain plenty of open questions on the true nature of artificial intuition systems. Mankind has stumbled on a kind of artificial intuition in the form of Deep Learning, however is it possible to discover other kinds of architectures that exhibit similar capabilities? As of this writing, I have not found alternatives. However, one should realize though that I know of at least two kinds of architectures that lead to intuition. That is Deep Learning and biological brains. Although these two systems are functionally similar, the computational mechanisms are like night and day.

A Roadmap for Deep Learning

DARPA has an excellent video that I encourage everyone to watch (note: viewing just the slides will give one a wrong impression of the content) [LAU]. The video distills the current state of AI into 3 waves:

Handcrafted Knowledge—Where programmers craft sets of rules to represent knowledge in well-defined domains.

Statistical Learning—Where programmers create statistical models for specific problem domains and train them on big data.

Contextual Adaptation—Where systems construct contextual explanatory models for classes of real world phenomena.

It's a bit of a simplified presentation because it lumps all of machine learning, Bayesian methods and Deep Learning into a single category. There are many more approaches to AI that don't fit within DARPA's 3 waves [LAU2].

Pedro Domingos author of the "The Master Algorithm" describes the five tribes of AI: Connectionists, Symbolists, Evolutionaries, Bayesians and Analogizers (I discuss 17 tribes of AI above). But let's give DARPA the luxury of simplifying their presentation of the current state of the field. They do cover some of the known problems of Deep Learning such as adversarial features and the need for supervised training data.

DARPA's third wave model takes a lot of inspiration from some of their previously announced research initiatives such as Explanatory interfaces and Meta-Learning. DARPA's presentation nails it, by highlighting what's going on in current state-of-the-art research. If anyone is seeking out a short explanation of what's going on in the field, then this is the video to watch.

The main problem that we face today is bridging the semantic gap between what I would call intuition and rational (symbolic) machines. Deep Learning systems have flaws analogous to our own intuitions having flaws. When you have cognitive processes that have limits on memory and time, and within the context information overload and lack of meaning, then you are bound to have flaws. These flaws however can be caught by logical systems. That's why bridging the gap can have some profound synergistic effects.

One reason that the semantic gap wasn't addressed as vigorously as before was that Connectionist systems (i.e. Artificial Neural Networks) did not historically work well. With the advent of Deep Learning, there's a new emphasis in finding a solution that melds Symbolic and Connectionist systems. That's where a lot of research is chipping away at the problem. The excitement here though is that it appears that the researchers are making outstanding progress!

Arend Hintze has a good short article on "Understanding the four types of AI, from reactive robots to self-aware beings" where he outlines the following types [HIN]:

> **Reactive Machine**—The most basic type that is unable to form memories and use past experiences to inform decisions. They can't function outside the specific tasks that they were designed for.
>
> **Limited Memory**—Are able to look into the past to inform current decisions. The memory however is transient and isn't used for future experiences.
>
> **Theory of Mind**—These systems are able to form representations of the world as well as other agents that it interacts with.

> **Self-Awareness**- This is a highly speculative description hence I will avoid further discussion.

I like his classification much better than the "Narrow AI" and "General AI" dichotomy. This classification makes an attempt to break down Narrow AI into 3 categories. This gives us more concepts to differentiate different AI implementations. Our reservation though of the definition is that they appear to come from a GOFAI mindset. Furthermore, the leap from limited memory able to employ the past to theory of mind seems to be an extremely vast leap.

I however would like to take this opportunity to come up with our own roadmap, more targeted towards the field of Deep Learning. I hope this roadmap is a bit more concrete and helpful for practitioners. This roadmap gives us a sense of where we currently are and where we might be heading.

We are inundated all the time with AI hype that we fail to construct a good conceptual framework for making a precise assessment of the current situation. This may simply be due to the fact that many writers have trouble keeping up with the latest development in Deep Learning research. There's too much to read to keep up and the latest discoveries continue to change our current understanding.

Here we introduce a pragmatic roadmap of Artificial Intuition (i.e. Deep Learning) capabilities:

1. Basic Inference (BI)

Deep Network

This level includes the fully connected neural network (FCN) and the convolution network (CNN) and various combinations of them. These systems take a high dimensional vector as input and arrive at a single result, typically a classification of the input vector. You can consider these systems as being stateless functions, meaning that their behavior is only a function of the current input. Generative models are one of those hotly researched areas and these also belong to this category. In short, these systems are quite capable by themselves.

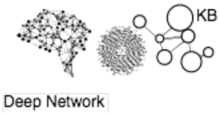
2. Inference with Memory (IM)

This level includes memory elements incorporated with the C level networks. LSTMs are example of these with the memory units are embedded inside the LSTM node. Other variants of these are the Neural Turing Machine (NMT) and the Differentiable Neural Computer (DNC) from DeepMind. These systems maintain state as they compute their behavior. These kinds of system lead to a more generalized translation system that can handle sequences and other structured input and output pairs.

3. Inference with Knowledge (IK)

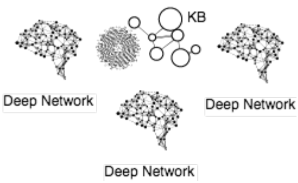
This level is somewhat similar to the IM level, however rather than raw memory, the information that the BI-level network is able to access is a symbolic knowledge base. There are actually three kinds of symbolic integration that I have found, a transfer learning approach, a top-down approach, a bottom up approach. The first approach uses a symbolic system that acts as a regularizer. The second approach has the symbolic elements at the top of the hierarchy that is composed at the bottom by neural representations. The last approach has it reversed, where a BI-level network is actually attached to a symbolic knowledge base.

4. Inference with Imperfect Knowledge (IIK)



At this level, we have a system that is built on top of IK, however is able to reason with imperfect information. An example of this kind of system would be AlphaGo and Poker playing systems. AlphaGo however does not employ IK but rather IM level capability. Like AlphaGo, these kinds of systems can train themselves by running simulations of self-play. This means the system learns new strategies by playing against itself.

5. Collaborative Inference with Imperfect Knowledge (CIIK)



This level is very similar to the "theory of mind" where we actually have multiple agent neural networks combining to solve problems. Theses systems are designed to solve multiple objectives. We actually do see primitive versions of this in adversarial networks that learn to perform generalization with competing discriminator and generative networks. Expand that concept further into game-theoretic driven networks that are able to perform strategically and tactically solving multiple objectives, and you have the making of these kinds of extremely adaptive systems. We aren't at this level yet and there's still plenty of research to be done in the previous levels.

Different levels bring about capabilities that don't exist in the previous level. BI-level systems for example are only capable of predicting anti-causal relationships [ANTI]. IM level systems are capable of very good translation. IIK level systems are capable of strategic game play.

DARPA's research hints at the third wave happening at the IIK level.

We can see how this roadmap somewhat aligns with Hinzte classification, with the exception of course of self-awareness. That's a capability that we really have not explored and don't intend to until the pre-requisite capabilities have already been addressed. We cannot know at this time that when we arrive at the final level we've achieved awareness.

Exploitation, Exploration and Representation

Back-propagation is the bread and butter mechanism for Deep Learning. Researchers had discovered that one could employ any computation layer in a solution with the only requirement being that the layer must be differentiable. Said differently, that one is able to calculate the gradient of layer. In more plain speak, that in the game of 'hotter' and 'colder', that the verbal hints that are made accurately reflect the distance between the blindfolded player and his objective.

There are several questions about back-propagation. The first is whether the gradient that is calculated is always the correct direction towards learning. This intuitively is questionable. One can always find problems wherein the moving towards the most obvious direction does not always lead to a solution. So it should not be unexpected that ignoring a gradient may also lead to a solution. (I don't think though you can ignore the gradient forever).

Let's step back a little bit and try to understand historically where this back-propagation idea comes from. Historically, machine learning originates from the general idea of curve fitting. In the specific case of linear regression (i.e. fitting a prediction to a line), calculating the gradient is a solving the least squares problem. In the field of optimization, there are many alternative ways other than using gradient to find an optimal solution. As a matter of fact, stochastic gradient descent is likely one of the most rudimentary approaches towards optimization. So it is just outstanding that one of the simplest algorithms one can think of actually works outstandingly well.

Most optimization experts had long believed that the high dimensional space that deep learn occupied would demand a non-convex solution and therefore be extremely difficult to optimize. However, for some unexplained reason, Deep Learning has worked extremely well using Stochastic Gradient Descent (SGD). Many researchers have later come up with different explanations as to why deep learning optimization is surprisingly easy with SGD. One of the more compelling arguments it that in a high-dimensional space, one is more likely to find a saddle point rather than a local valley. There will always be sufficient dimensions with gradients that point to an escape route.

There is also the question regarding the typically objective function that is employed. Back-propagation is calculated with respect to an objective function. Typically, the objective function is a measure of the difference between the predicted distribution and the actual distribution. Usually, something derived off the Kullback-Liebler divergence or some other similarity distribution measure like Wassertsein. However, it is in these similarity calculations that the "label" in a supervised training exists.

You can't do back-propagation if you don't have an objective function. You can't have an objective function if you don't have a measure between a predicted value and a labeled (actual or training data) value. This is why Deep Learning has yet to be successful in the realm of "unsupervised learning".

The algorithms of learning can be coarsely abstracted as being a balance of exploration and exploitation. A balanced strategy is followed in the pursuit of a fitter representation. This representation can either be one that improves a model that is being learned or can be at the meta-level where it improves the algorithm that learns better models.

In exploitation, automation greedily pursues a path of learning that provides immediate rewards. In exploration however, automation must decide to forego an immediate reward and select instead a directionless exploration with the intent of discovering a greater reward elsewhere. The strategy to select one over the other is sometimes referred to as "regret minimization".

It is also related to the idea of Counterfactual Regret Minimization (CFR). This method is used by Libratus[LIB] a poker playing machine that has bested professional players. CFR is applicable in domains with imperfect-information [IMP]. In short, the strategy of selecting exploration over exploitation is relevant to domains with imperfect information.

Stochastic Gradient Descent (SGD), the workhorse learning algorithm of Deep Learning, are algorithms that employ exploitation as its fundamental motivation. SGD works only for networks that are composed of differentiable layers. Convergence happens because there will be regimes in the parameter space that guarantee convergence of iterative affine transformations. This is well known in other fields such as Control Theory [CTR] (known as Method of Adjoints) as well as in Chaos theory (Iterated Function Systems).

However, exploration features are shoehorned into classic gradient descent through different kinds of randomness. Examples of these are, the randomness in how training examples are presented, noise terms in the gradient, dropout and batch normalization. When we examine the two phases of gradient decent [CEP], we realize that the first phase is dominated by exploitation behavior. This is where we see a high signal to noise ratio, and the convergence is rapid. In this phase, second order methods that exploit the Natural Gradient (see: Fisher Information Matrix) will converge much faster. A recent method known as the Kronecker Factorization [K-FAC] that approximates the FIM has shown to exhibit 20–30 times less iterations than traditional first order methods.

In the compressive phase, exploration will dominate and randomization methods facilitate these explorations. In this regime, the gradients carry negligible information [GBDL] and thus the convergence is extremely slow. This is where representation compression occurs. The elusive goal of Generalization is achieved through the compression of representation. We can explore many interpretations as to what Generalization actually means, but ultimately, it boils down to the shortest expression that can accurately capture the behavior of an observed environment.

Evolutionary algorithms (aka Genetic algorithms) occupy the space of exploration approaches. In Deep Learning, evolution algorithms are usually been employed in Meta-learning searching for architectures. It is a more sophisticated version of hyper-parameter optimization in that instead of juggling constants like learning rates; the search algorithm juggles the composition of each layer of a network. It is used as the outer loop of the learning algorithm. The thing though about evolutionary algorithms is that serendipitous discovery is fundamental. In short, it works only when you are lucky.

Either method or a combination of both can lead to a fitter Representation. Let's deconstruct the idea of Representations. In this book, I discuss 3 different dimensions of intelligences that are being developed (computational, adaptive and social). The claim is that these are different kinds of intelligences. What is apparently obvious is that the domains in which they operate are different from each other. So form will have to follow function. The methods and architectures that are developed for each kind of intelligence are going to be different from each other. One dimension of Representation is obviously the domain in which it is applicable.

There is of course a question whether we should explore different kinds of Representations. I mean this at a more general level. In Deep Learning, there are all kinds of different neural embeddings. These embeddings are vector representations of semantics. These vector spaces are learned over the course of training. These vectors are supposed to represent an invariant form of the actual concept. One major difficulty of Deep Learning systems is that these representations are extremely entangled. The fact that they are entangled can explain why Deep Learning systems have zero conceptual understanding of what they predict. Understanding requires the creation of concepts, if concepts cannot be factored out, then what does that imply for understanding? It is important to realize, that there are many cases where understanding is not needed for competence.

I will argue that AlphaGo doesn't understand Go in the same way as humans. Humans understand Go by creating their own concepts behind the strategies they employ. AlphaGo doesn't have an understanding of these concepts. Rather, it has memory and the statistics of billions of moves and their consequences.

The concepts that exist as part of a Representation may exist in 3 ways. As a generalization, a prototype or an exemplar. Deep Learning focuses on creating generalizations through the capture of an invariant representation. This is why; data augmentation is a best-practice approach. So when working with images, images are rotated, cropped, de-saturated etc.. This trains the network to ignore these variations. In addition, Convolution Networks are designed to ignore image translations (i.e. difference in locations). The reason DL systems require many training sets is that it needs to "see" enough variations so that it can learn what to ignore and what to continue to keep relevant. Perhaps however that the requirement for invariances is too high and we should seek something less demanding in the form of equivariances.

In the realm of few-shot or zero-shot learning, where an automation must learn something by seeing it only once or a few times, then there is zero opportunity to discover invariances. Automation only has a few examples to create a prototype that is representative of the entire class. So there needs to be some prior model that is capable of performing the appropriate similarity calculation. The system must know how to determine if an example is similar to a prototype.

Even worse, if its just one example, then there isn't really a class and the system has to deduce a generalization. The implication of the latter is that, automation requires that an internal model existing prior to any deduction. So we have here three kinds of models: a model-free representation (learned through induction), a representation for a similarity algorithm and a rich representation that can drive deduction.

We explored how model-free and model-based cognition can be interleaved in the process of learning. Exploration and exploitation can also be interleaved in learning. However, both sets are orthogonal. As in SGD, you can have a model-free algorithm that uses both exploration and exploitation. You can also have model-based algorithms that explore or exploit. That is, there are at least three dimensions that are described here. One dimension is on the axis of exploration to exploitation. The second dimension is if an explicit model drives the learning process or not. The third dimension is the nature of the learned representation itself.

Meta Model

Deep Learning can be conceptually thought of as consisting of a computational graph that is constructed by composing layers with other layers. Most introductory texts emphasize the individual neuron, but in practice it is the collective behavior of a layer of neurons that is more important. So from an abstraction perspective, layers of computational units, rather than individual neurons, are the correct abstract on how to understand Deep Learning. Google's Tensor Processing Unit (TPU) provides supporting evidence of this point of view. The TPU, unlike conventional CPUs and GPUs that treat scalar and vectors as primitives, treats matrices as a primitive instead.

These layers are built on a computational graph; its main purpose is to orchestrate the computation of the forward and backward phases of the network. From the perspective of optimizing the performance, this is an important abstraction to have. However, it is not at the ideal level to reason how it all should work. It is like plumbing, very important, but to aid our understanding, we can ignore that it exists.

Deep Learning frameworks have evolved to develop models that ease construction of Deep Learning architectures. Theano has Blocks, Lasagne and Keras. Tensorflow has Keras and TF-Slim. Keras is based on Torch, so by default has a high-level modular API. Many other less popular frameworks like Nervana, CNTK, MXNet and Chainer do have high-level model APIs. All these APIs however describe models. What then is a Deep Learning meta-model? Is there even a meta meta-model?

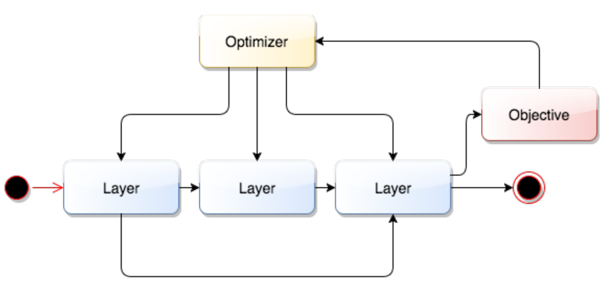


Figure 4.1: This is what a Deep Learning model looks like.

Let's explore first what a meta-model looks like. A good example is in the UML domain of Object Oriented Design. This is the UML metal model:

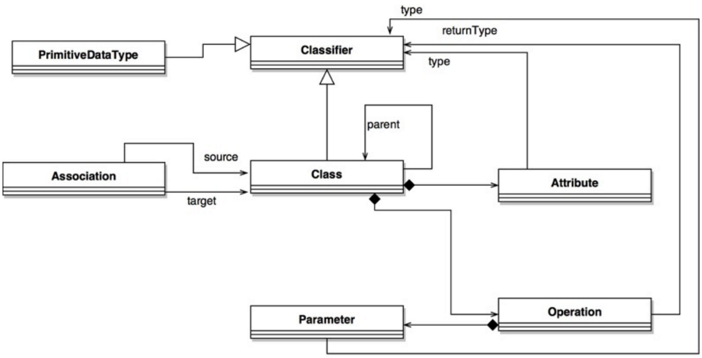


Figure 4.2 UML Metal model.

This makes it clear that Layers, Objectives, Activations, Optimizers, Metrics in the Keras APIs are the meta-models for Deep Learning. This is not too difficult of a concept to understand.

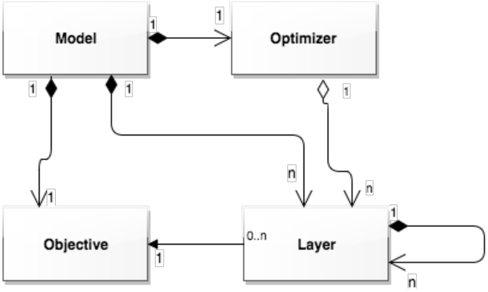


Figure 4.3. Deep Learning Meta Model

The model depicted above captures all the main elements of a Supervised Learning network. The key elements are in the model that consists of many layers of different kinds. The model includes an objective function, sometimes known as a loss function or a fitness function. Finally, there is an optimizer or leaning algorithm that is used to train the layers of the model. The optimizer is usually only needed in the training phase of a network. Depicted below is a partial view of the many kinds of networks, objective functions, and optimizers that are being used:

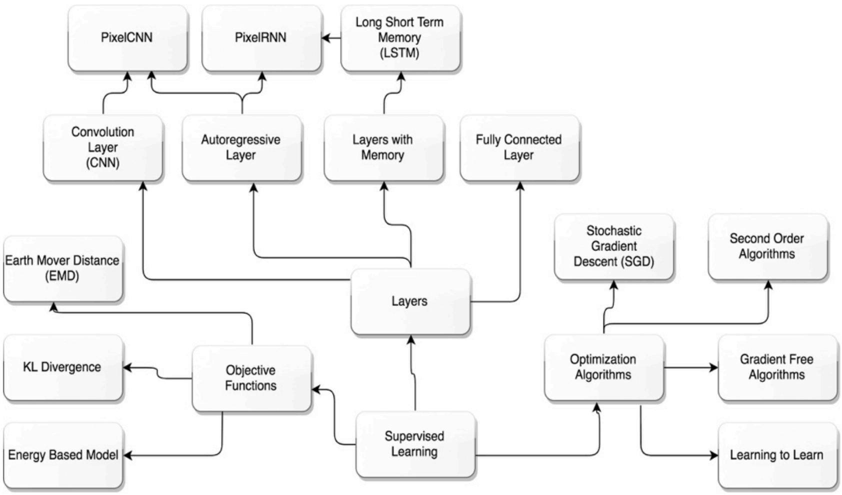


Figure 4.4 Deep Learning Supervised Learning

Many kinds of networks can be combined to form a solution. In the more classical machine learning approaches, a lot of effort is spent wrangling the data to extract features. In contrast however, in Deep Learning, a judicious use of a good combination of layers is what is needed to develop a good solution. Four of the common kinds are depicted above. These include the Fully Connected layer or Multi-Level Perceptron (MLP), the Convolution layer, the Recurrent layer (RNN) and the Autoregressive network. Note that RNN is not depicted in the diagram since a more common variant of it called the Long Short Term Memory (LSTM) is used in practice.

Conventionally, an Objective is a function and an Optimizer is an algorithm. However, what if we think of them instead as also being models. In that case we have the following:

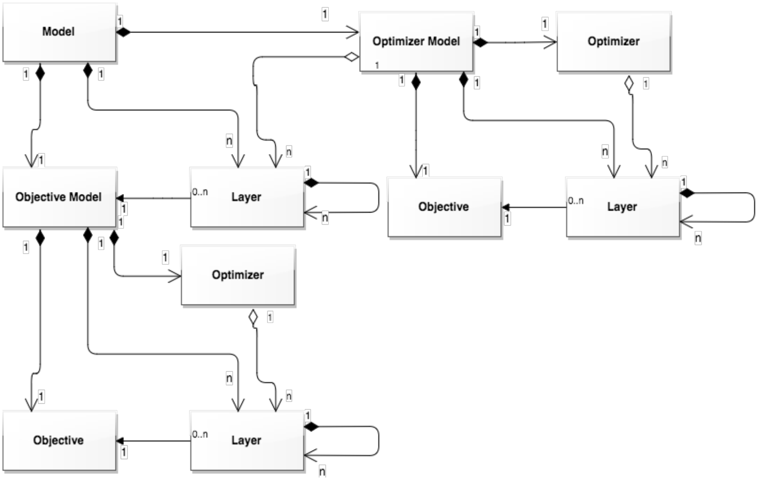


Figure 4.5. Every function is a neural network

This definitely is getting a whole lot more complicated. The objective function has become a neural network and the optimizer has also become a neural network. The first reaction to this is, has this kind of architecture been tested before? It's possible someone is already writing this paper. That's because an objective function that is a neural network is equivalent to the Discriminator in a Generative Adversarial Network (GAN) and an Optimizer being a neural network is precisely what a meta-learner is about. So this idea is not fantastically out of mainstream research.

The second reaction to this is, shouldn't we make everything neural networks and be done? There are still boxes in the diagram that are still functions and algorithms. The Objective's optimizer is one and there are 3 others. Once you do that, there's nothing else left that a designer needs to define! There are no functions and everything is learned from scratch!

So a meta-model where everything is a neural network looks this:

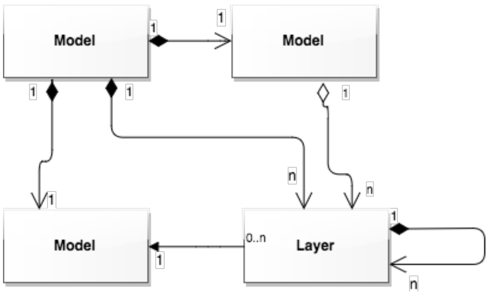


Figure 4.6. Deep Learning Meta-Model

Where the mode is broken apart into 3 parts just for clarity.

What this makes abundantly clear however is that the kinds of layers that are available come from a fixed set (i.e. fully connected, convolution, LSTM etc.). There are in fact research papers that exploit this notion of selecting different kinds of layers to generate DL architectures. A DL meta-model language serves as the Lego blocks of an exploratory RL based system. This can generate multiple DL meta-model instances to optimize for the best architecture. That is a reflection of the importance of Deep Learning Patterns. Before you can generate architectures, you have to know what building blocks are available for exploitation.

A Reality Checklist

Where is Deep Learning applicable? This is one of the more fleeting ideas to understand about Deep Learning and related A.I. technologies. It is all too easy to fall in the trap that an "Artificial Intelligence" application can solve your problem.

The usual coverage of this problem involves the question of "do you have enough data?" Unfortunately, that is too vague in that to answer this you have to at least understand your problem domain. In the academic sense, you want to understand the "boundary conditions." Said differently, you want to understand the intrinsic constraints of your problem. What exactly are "boundary conditions" of Deep Learning or A.I. learning problem?

To understand a problem domain, you have to understand what the current state of knowledge is. Uncertainty is a measure of what is unknown relative to what is knowable. The strange assumption of the word uncertainty is that it assumes that everything can be eventually known. That is we assume determinism, in the real world this is rarely ever the case.

Nevertheless, it is a useful term that we will use to identify to boundaries of a problem. Here is the checklist. It is written in the form of questions and an answer of "no" implies the existence of uncertainty. For some odd reason, I cannot phrase it in the other way that makes it simple! Contrapositive statements always seem complicated to parse! I will also use the terms actors, activity and environment to describe the entire context.

Execution uncertainty—Does the sequence of actions of the actors, from the environment's initial state, always lead to the same final state?

Observational uncertainty—Do actors have complete information of the environment?

Duration uncertainty—Do the actors know how long the activity will last?

Action uncertainty—Are the effects of the actor's actions known exactly?

Evaluation uncertainty—Is there an evaluation criteria to measure the successful completion of the activity?

Training uncertainty—Is there knowledge or data of previous successful solutions that can be used as guidance to learning?

In most real world problems, the answers to these questions are most likely to be in the negative. Automation requires that a majority of the above be answered in the affirmative. Deep Learning based automation however equips the practitioner with a little more wiggle room here. The nature of Deep Learning automation is that they are approximation machines. However, DL can only learn a good approximation of an uncertain situation if there exists information that already removes the uncertainty (i.e. supervised learning).

No machine can know the unknowable (i.e. the unsupervised learning problem). More specifically, systems learn by using knowable knowns to increase certainty of knowable unknowns. The checklist above guides one to identify what is unknown and thus is a list of "known unknowns". Let's be real here, you can't solve a hard problem if you don't know what you don't know! Strive to break that "reality distortion field" by identifying exactly what not only uncertain but also what is unknowable in your problem domain.

Deep Learning cannot do magic even if it is practiced like alchemy. However, understanding the fuzzy boundaries of what is reality and what is science fiction can give you a tremendous leg up from the competition.

Deep Learning Limitations

In addition to the above general limitation of what a Deep Learning system can predict there are other known limitations of Deep Learning systems. These limitations unlike the ones described in the previous section are not fundamental. In other words, there is a confident expectation that these limitations will eventually be solved.

The largest flaw is the discovery that by tweaking an input image in a specific way one can fool a Deep Learning Neural Network to misclassify. This is known as an Adversarial input or feature. This is a problem for not only Deep Learning but for almost every machine learning method that employs linear algebra to approximate its classifications. Restrictive Boltzmann Machines (RBMs) interestingly enough does not exhibit this problem.

Secondary to this flaw is the requirement for a large labeled training size to learn new concepts. There is research to reduce the amount of training data, but we have yet to achieve the kind of learning that we see in humans. Humans are able to learn by seeing a new concept just once.

Third is the lack of interpretability of the models that are learned. We have yet to find a satisfactory explanation on how a Deep Learning system arrives at its predictions. These systems that learn from data tend to be black boxes with respect to their predictions.

Fourth, most Deep Learning systems require supervised learning. There is a lot of progress developing unsupervised learning systems but that is not as widely used outside of research.

Deep Learning progress moves at a brisk pace, but we can get a bearing by consulting the capability levels that we described earlier. We are currently in the early stage of level 2 (i.e. CM). Memory is an essential ingredient for systems that are able to examine sequences and temporal behavior. The most advanced kinds of this are encoder-decoder networks, or alternatively networks that translate sequences to other sequences. The most impressive example is Google**'s** Natural **Machine Translator** (GNMT) [MER].

The interesting thing about GNMT is that Google headlines this as "**Zero-Shot Translation**" *[SCH]*:

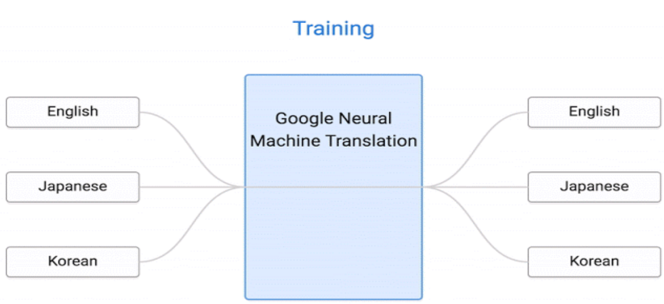


Figure 4.8 Google NMT. Credit: Google

This zero-shot capability refers to the capability of the machine to learn for example a Japanese to English translation, even if it was never trained with this particular translation pair! To quote them:

> *This means the network must be encoding something about the semantics of the sentence rather than simply memorizing phrase-to-phrase translations. We interpret this as a sign of existence of an interlingua in the network.*

Translation systems are examples of state-of-the-art Deep Learning. There is an unfortunate misconception about Deep Learning that they are only capable of classification. This is a Level 1 capability in our classification. Level 2 capabilities enable translation. One good rule of thumb in evaluating opportunities for DL is that if a problem can be framed in terms of a translation problem, there is a good possibility that DL can be an effective solution.

DeepMind, a company that had zero revenue and products in the market, was acquired by Google in 2014 for around half a billion USD. The story goes that Google executives were so impressed by their Atari video game playing program that they could not resist making the acquisition. DeepMind's goal is to "solve intelligence"; they do so by combining Deep Learning and an older AI technology called Reinforcement Learning. This combination is usually referred to as "Deep Reinforcement Learning". The Atari game-playing program combined Convolution Networks and Q-Learning. The following diagram depicts the relationships between the different RL algorithms:

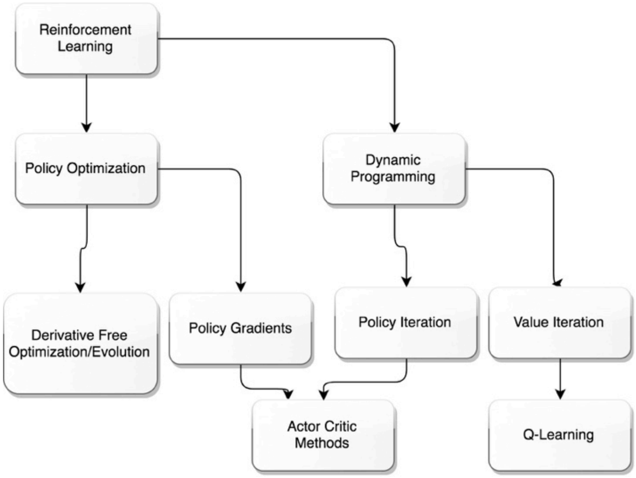


Figure 4.9 Reinforcement Learning

Despite many of the advances shown by DeepMind, Yann LeCun has repeatedly hammered away at this analogy:

> *If intelligence was a cake, unsupervised learning would be the cake,*

> *supervised learning would be the icing on the cake, and reinforcement learning would be the cherry on the cake. We know how to make the icing and the cherry, but we don't know how to make the cake.*

Since NIPS 2016, LeCun started using the phrase "**predictive learning**" instead of "unsupervised learning" [CUN]. LeCun says:

> *A key element we are missing is predictive (or unsupervised) learning: the ability of a machine to model the environment, predict possible futures and understand how the world works by observing it and acting in it.*

This is an interesting change and indicates a subtle change in his perspective as to what he believes is required to build up the "cake". In LeCun's view, the foundation needs to be built before we can make accelerated progress in AI. In other words, building from current supervised learning by adding more capabilities like memory, knowledge bases and cooperating agents will be a slog until we are all able to build that "predictive foundational layer".

Predictive Learning is a formidable problem that is ahead of us. Predictive learning clearly requires machines to be able to learn not only without human supervision, but also to learn a predictive model of the world. It is very important to emphasize this point and understand why LeCun is attempting to change our perspective of the canonical taxonomy of AI (i.e. unsupervised, supervised and reinforcement learning).

Ruslan Salakhudinov, Apple's lead AI researcher, had a good **survey talk on Unsupervised Learning** (i.e. Predictive Learning) where he provides this informative taxonomy:

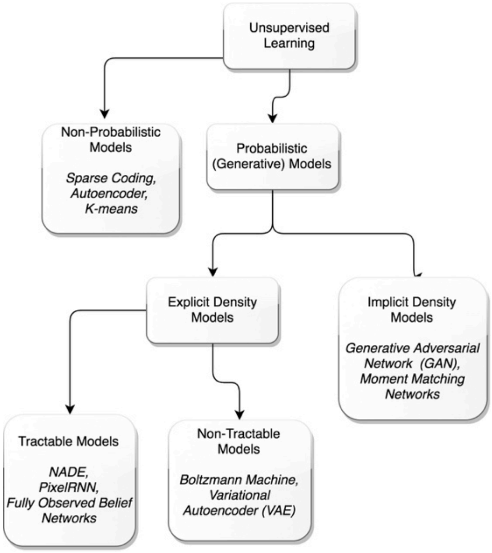


Figure 4.10 Unsupervised Learning, from a talk by Ruslan Salakhudinov [FRI]

At the right corner of the slide he mentions Generative Adversarial Networks (GANs). GANs consist of competing neural networks - a generator and a discriminator. The former tries to generate fake images while the latter tries to identify real images. The interesting feature of these systems is that a closed form loss function is not required. In fact, some systems have the surprising capability of discovering its own loss function! A disadvantage of adversarial networks is that they are difficult to train. Adversarial learning consists of finding Nash equilibrium in a two-player non-cooperative game. In a recent lecture on unsupervised learning, Yann LeCun calls adversarial networks the "**the coolest idea in machine learning in the last twenty years**" [C18].

Elon Musk's OpenAI research has a big focus on **Generative Models** [KAR]. Their motivation can be summarized by Richard Feynman's quote "What I cannot create, I do not understand." Feynman is alluding to his "First Principles" method of thought where he needs to be able to build up understanding by composing proven concepts. The basic idea here is that if a machine is able to generate models with high realism, then it can perhaps develop an understanding of the predictive model.

Here are some images of this state-of-the-art technique:

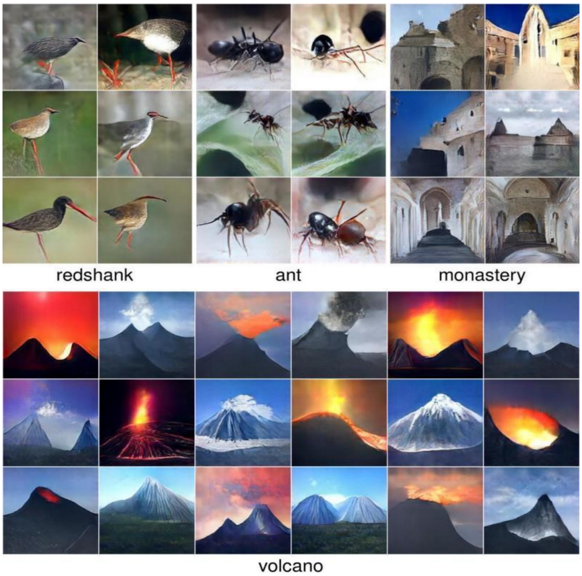


Figure 4.11 Source: http://www.evolvingai.org/ppgn [NGU]

These are images generated by the DL system when it is given the word shown. This is indeed quite impressive. I wouldn't expect many humans to be able to draw this well!

The current consensus is that these generative models aren't able to capture the semantics of the task. They don't understand the meaning of an ant, volcano or redshank. They are however very good at mimicry and in making predictions. These images are not recreations of images that the machine was previously trained on. Rather, the machine has come up with some generalized model that allows it to extrapolate a very realistic result.

This approach of using adversarial networks is different from the more classical approach of machine learning. Here we have two competing neural networks (i.e. discriminator and generator) that seem to work synergistically to accomplish a kind of generalization. In the classical ML world, one would define an objective function that would fire up one's favorite optimization algorithm. In this research area however, the correct objective function is unclear. Even more surprising is that these systems are able to learn their own objective function!

The fascinating realization here is that DL systems are extremely malleable. The classic ML notion that the objective functions, constraints, or optimization algorithms are fixed does not apply to DL. Even more surprising is that a meta-level approach can be used. That is, DL systems can learn how to learn.

Generative models are state-of-the-art techniques and are applicable in problem contexts requiring the exploration of many design alternatives.

What we begin to realize when we see the capabilities in translation and generation is that our notion of what DL can do is continuously shifting. This is why it is difficult to provide a definite answer to the limits of DL. The boundaries of what is possible continues to be pushed outward and our ideas of what is possible for 'narrow intelligence' continues to change.

The applicability of narrow intelligence can be broadly sketched as one where supervised learning is required and the environment is sufficiently controlled. A controlled environment is one that is self-contained, or isolated from outside influences. This definition acknowledges the limitations of systems that need to react to unexpected data. We are at this time unable to quantify accurately how well our systems are able to generalize.

This is the current state-of-the-art. I can however not make a judgment at this time as to when this definition will need to change. Understanding the limitations of the technology is the first step in knowing how to wield it effectively.

Artificial General Intelligence

It is important to have an Artificial General Intelligence (AGI) roadmap. Not because we are going to achieve this anytime soon, but rather because we need a framework to understand the progress that is happening in Deep Learning. Ask any researcher and they will tell you that progress in Deep Learning is at breakneck speed. Russ Salakhutdinov (Apple's lead in AI) in a recent Simon's Institute lecture remarked that the developments were "crazy". So, in this context of "crazy", we got to get our bearings and see exactly where the hell are we at!

Previously I gave a roadmap in terms of capabilities that we may anticipate for Deep Learning. Unfortunately, this does not give enough of a sense of what is achievable at each capability level. Peter Voss points out a paper by Pat Langley in 2012 in his post about "Cognitive Architectures" [LAN] [VOS]. Pat Langley's paper is a good framework for assessing what kind of progress needs to be made in AGI. I have to admit, I have not spent a lot of energy trying to contrast the different AGI approaches that are out there. I do however find Pat Langley's paper to be reasonable, simple, and conservative enough to be a good basis for assessing current AGI development.

I will revisit the points in the paper from the perspective of recent Deep Learning development. I will ignore any development from other A.I. tribes in my analysis. I encourage the reader to read Langley's paper. The paper was written in 2012, prior to the Deep Learning boom, so I am revisiting it today to see if we have made any progress in the AGI fronts that was described in the paper.

1. High-Level Cognition

Hybrid architectures like Deep Reinforcement Learning and AlphaGo reveal an extremely compelling way to fuse the intrinsic intuition based cognition in DL system into higher-level capabilities that require planning and strategy. We are still a ways off in achieving abstract reasoning, comprehension and problem solving

2. Structured Representations

Deep Learning representations are opaque and inscrutable. However, interestingly enough, the many forms of neural embedding (i.e. word2vec, Glove etc.) seem to be able to capture some semantics to be useful as input features to other networks. It also seems that some prior external knowledge about the world can be introduced by these embeddings. Integration tends to require end-to-end training, however this is an extremely promising area to pursue. What I don't see happening soon is the interpretability of the representations.

3. System-Level Approach

I think "Cognitive Synergy" is going to be one of the more powerful developments in Deep Learning. Cognitive Synergy is the notion that many agents can work off the same representation. I think Deep Learning is making considerable strides in this space with regards to "Multi-objective" systems as well as in encoder-decoder networks. What we are seeing today is that multiple neural networks (see: Modular Deep Learning) are working in concert to create impressive results. The main stumbling block is that there is a need for 'late-binding' of representations. That is something that has yet to be developed.

4. Heuristics and Satisficing

There is an inherent bias in the research community to demand that the mechanisms that are used by Deep Learning have to conform to more rigid mathematical reasoning. There is a lot of emphasis in our models that demand some probabilistic interpretation. I am however on the camp that complex behavior emerges out of simplistic mechanisms. There is still a lot of work that needs to be done here, however we can see how our current models seem to be cracking under the strain of a lot of unexplainable observations.

5. Links to Human Cognition

Deep Learning developments are nowhere near addressing what Voss' describes as "ambiguity, abstract conceptualization and reasoning, short-term memory and context, as well as meta-cognition."

6. Exploratory Research

Fundamental research on the behavior of Deep Learning takes a back seat to novel techniques that yield impressive results in narrow domains. It will likely stay this way for a while. The 'attention' market today commands a higher premium on novelty and benchmarks rather than fundamental insight.

Note that these are 6 assumptions that "were widely adopted during the AI's first three decades". The beauty of this is that we can contrast what we know today about Deep Learning and see how it fits with decades all perspective of what needs to be invented. As you can see, there are developments with good progress in 5 or 6 fronts.

Experimentation and engineering in Deep Learning far surpasses theory and this trend will not end soon or at all. Fundamental research isn't given as much a priority in this field over more tangible "state-of-the-art" results. However, that does not imply that we should neglect thinking about a much larger AGI roadmap such as this. This post hopes to shed a little more light on how far Deep Learning needs to go to achieve AGI.

The Strange Loop

In the previous section, there are certain kinds of intelligences that are dependent on having a self-referential or reflect capability. Specifically intrapersonal and interpersonal intelligences, require self-modeling and reflection.

Douglas Hofstadter in his book "I am a Strange Loop" [HOF] coined this idea:

> *In the end, we are self-perceiving, self-inventing, locked-in mirages that are little miracles of self-reference.*

Where he describes this self-referential mechanism as what describes the unique property of minds. The strange loop is a cyclic system that traverses several layers in a hierarchy. By moving through this cycle one finds oneself where one originally started.

Coincidentally enough, this 'strange loop' is in fact the fundamental reason for what Yann LeCun describes as "the coolest idea in machine learning in the last twenty years [YAN].

Loops are not typical in Deep Learning systems. These systems have conventionally been composed of acyclic graphs of computation layers. However, as we are all now beginning to discover, the employment of ‘feedback loops’ are creating one of the most mind-boggling new capabilities for automation that is beyond our imagination. This is not hyperbole, this is happening today where researchers are training ‘narrow’ intelligence systems to create very capable specialist automation that surpass human capabilities.

My first recollection of an effective Deep Learning system that used feedback loops were in “Ladder Networks”. Ladder Networks were introduced a very long time ago, way back in July 2015 [RAS]. Here is a depiction of the architecture:

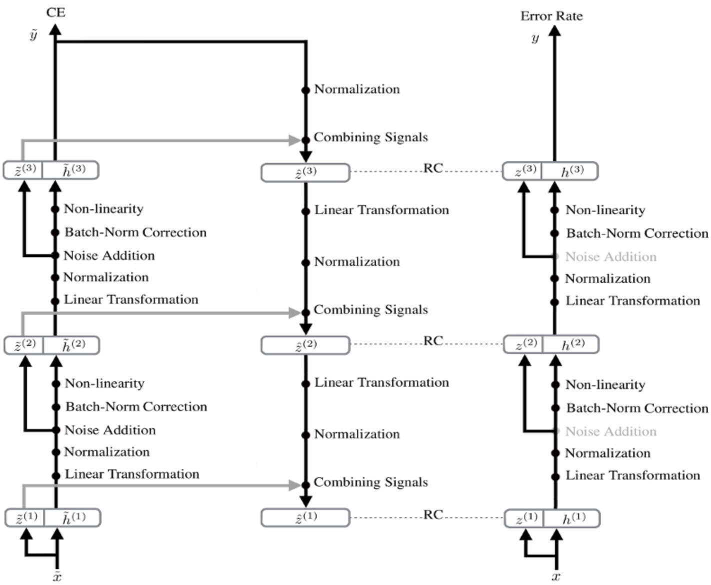


Figure 4.12 Deconstructing the Ladder Network Architecture [RAS]

The Ladder Network is a single loop up and down the layers followed by a final single forward pass. The system gathers information from parts in the loop and uses them as regularization constraint. At the time it was introduced, it exhibited remarkable speed in convergence. The original researchers in a paper further extended this in mid 2016:

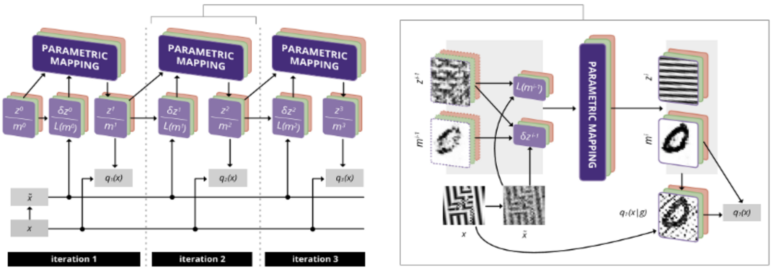


Figure 4.13 Tagger: Deep Unsupervised Perceptual Grouping [GRE]

In the Tagger architecture, you have multiple ladder networks strung together to form a network that performs a better kind of segmentation that is able to group objects in images.

The Generative Adversarial Network (GAN) also has its own loop, but not explicit in its architecture, but rather as part of its training. A GAN involves a training process with cooperative and dueling networks. This involves a generative network and a discriminative network. The discriminator network attempts to perform a classification against data that the generative network is creating. The generative network attempts to find data that tries to fool the discriminative network, and as a final consequence a more robust discriminator and generator is formed. GANS perform a kind of Turing test and are currently the best generative model for images.

There is basically a feedback mechanism that is used in the form of a neural network (the discriminator) that a generator takes advantage of to create more sophisticated results. There are many kind of GANs that are able to generate extremely realistic images.

Systems that leverage learning loops also relate to newer research on 'incremental learning'. One of the drawbacks of Deep Learning systems is that of the problem that 'fine-tuning' the network by training against new data can destroy previously remembered capabilities. This is the problem of the network 'forgetting' its past learning. In an architecture developed by Stanford called **"Feedback Networks**", the researchers explored a different kind of network that feeds back into itself and develops the internal representation the improves incrementally:

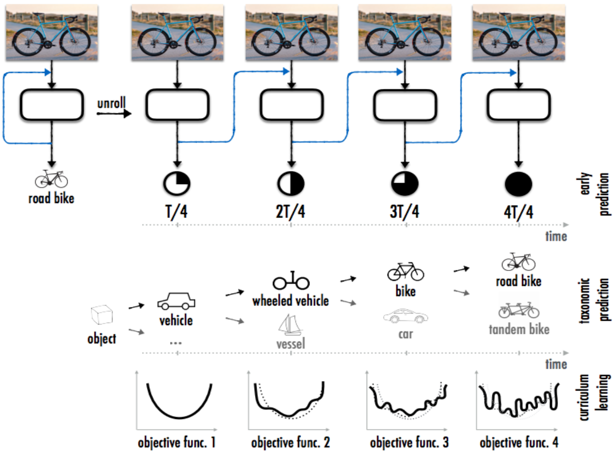


Figure 4.14 FeedbackNet. source: http://feedbacknet.stanford.edu/feedback_networks_2016.pdf

There remains a lot of work to be done in solving the 'forgetting' problem. However, one research paper "**On the Limits of Learning Representations with Labeled Supervision**" constructs a proof that GAN based system have higher representation capacities that comparable straight feed forward networks. So not only do GANs lead to improved generalization, it perhaps can lead to less forgetfulness [SON]

Recently published research (March 2017) from UC Berkeley has created astonishingly capable image-to-image translations using GANs and a novel kind of regularization. They call this system **CycleGAN**, and it has some very impressive results:

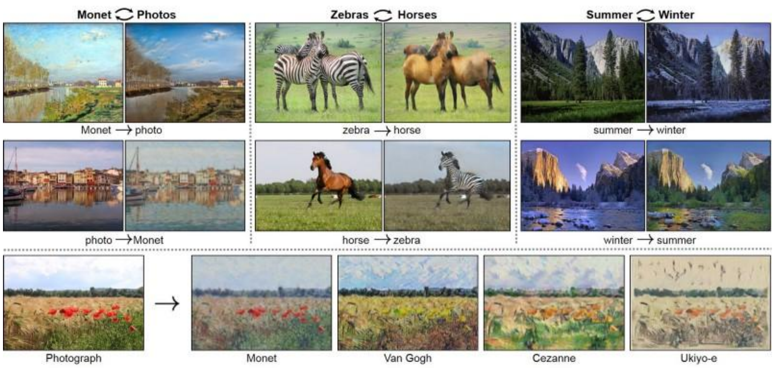


Figure 4.15 CycleGAN examples. Source: [UNP]

CycleGAN is able perform remarkable image translations. As shown above, it takes paintings as input and generates realistic photographs. It can perform what appears to be semantic translation such as converting horses into zebras or converting images taken in one season and making it appear to be taken in another season.

The crux of CycleGAN method is the use of a 'cycle-consistency loss'. This loss ensures that the network can perform the forward translation and the reverse translations with minimal loss. That is, the network must learn how to not only translate the original image, in needs to also learn the inverse (or reverse) translation.

The major difficulty of training Deep Learning systems has been the lack of labeled data. Labeled data is the fuel that drives the accuracy of Deep Learning models. However, these newer kinds of systems that begin to exploit loops are solving this lack of supervision problem. It is like having a perpetual motion machine wherein these automation dream up new variations of labeled data. As a consequence, paradoxically fueling themselves with more data. These automations play simulations games with themselves, and with enough game play, become experts at it.

It is analogous to how AlphaGo was able to develop new Go strategies by doing self-play against itself. When automation is embedded with a feedback loop and is able to simulate (some would call this 'imagination') many different scenarios and self test those scenarios for correctness, then we are at the cusp of some extremely potent technology that can rapidly cascade into capabilities that few in our civilization will be prepared for. So the next time you see some mind boggling Deep Learning results, seek to find the strange loops that are embedded in the method.

Let me point out that Hofstadter's Strange Loop was conceived in the domain of infinite recursion. Deep Learning processes, however, assume finite computation. That is, even if there are loops, it will always be bounded to finite computation. Also there have been previous methods that employed loops or recursion. However, they did not traverse hierarchies consisting of many layers of abstraction.

That is the intuition that Hofstadter had that seems to remarkably make sense in today's research.

I will leave you with the following quotation from an interestingly titled article "How Victory of Google's Go AI is Stroking fear in South Korea" [ZAS]:

> *"AlphaGo actually does have an intuition," Google co-founder Sergey Brin told New Scientist hours after his firm's series-clinching third victory, which he'd flown in to witness. "It makes beautiful moves. It even creates more beautiful* ***moves than most of us could think of".***

Summary

This chapter studies in greater detail the concepts of Deep Learning. An observant reader will notice that the presentation here is unconventional. The more traditional approach is to begin with linear algebra. Gradient descent is explained as an algorithm for performing linear regression. Finally, the Artificial Neural Network is introduced as being a kind of network architecture that does 'curve-fitting' in higher dimensions. The problem I find with the traditional approach is that it tends to focus too much on the algorithms but ignores the big picture.

5 The Universe that Learns

"I believe that, one day, information will come to be viewed as being as fundamental as energy and matter."

- Demis Hassabi – DeepMind Founder

Introduction

TBD

The Unreasonable Simplicity of Universal Machines

Rule 110 cellular automata, or more specifically the one dimensional cellular automata (you can explore those here http://atlas.wolfram.com/01/01/) that has the following rule:

current pattern 111 110 101 100 011 010 001 000 new state for center cell 0 1 1 0 1 1 1 0 is all the complexity that one needs to create a machine that has all the computational capability of a Turing Machine, hence any computer system.

NAND gates (or alternatively NOR gates):

INPUT OUTPUT A B A NAND B 0 0 1 0 1 1 1 0 1 1 1 0

is all the logic one needs to compose any boolean equation.

How does a Rule 110 automata differ from a NAND gate? The NAND gate has 4 rules, the automata has however 8 rules. If we look closely, we see that the Rule 110 automata contains all the rules of the NAND gate. Specifically, 010 -> 1, 011 0 ->, 110 -> 1 and 111 -> 1. In other words, if the center cell is set to 1, then Rule 110 acts just like a NAND gate. However, there are 14 other cellular automata that have the capture the NAND logic but are not universal.

The cellular automata state of 0 for Rule 110 automata apparently has some additional capability that leads to universal behavior. Let's examine these, for when the center cell is 0, the behavior becomes:

1 0 1 -> 1

1 0 0 -> 0

0 0 1 -> 1

0 0 0 -> 0

or if we ignore the center cell:

1 1 -> 1

1 0 -> 0

0 1 -> 1

0 0 -> 0

The middle two rules appear to break symmetry in that there's a clear distinction as to which neighbor cell is on.

Let's examine another automata, Rule 30 that is known to be chaotic:

current pattern 111 110 101 100 011 010 001 000 new state for center cell 0 0 0 1 1 1 1 0

For when the center cell is 0:

101 -> 0

100 -> 1

001 -> 1

000 -> 0

which is a XOR

and when the center cell is 1:

111 -> 0

110 -> 0

011 -> 1

010 -> 1

with that symmetry breaking that we see in Rule 110.

The complement of Rule 110 is Rule 137:

current pattern 111 110 101 100 011 010 001 000 new state for center cell 1 0 0 0 1 0 0 1

Which is the same as 110, but instead with a universal NOR gate.

111 -> 1

110 -> 0

011 -> 1

010 -> 0

Which is the same behavior as Rule 110 but with the center state now 1 instead of 0.

If we replace the rule for 111 and 010 to 111 -> 0 and 010 -> 0 we have

current pattern 111 110 101 100 011 010 001 000 new state for center cell 0 0 0 0 1 1 0 1

which is Rule 13 and not universal.

So its not just the symmetry breaking that's important, but the fact that 11->1 and 00->0 are important. Note: Flipping the rule for 10 and 01 are also universal.

So, what perhaps is the significance of this circuitry…

1 0 1 -> 1

1 0 0 -> 0

0 0 1 -> 1

0 0 0 -> 0

that leads to universality?

What we see with these rules is that the value on the right neighbor cell becomes the center cell. The mirror rule 124 shifts from the left and the complement rule shifts also from the right. So to achieve a universal machine one just needs two rules. A NAND or NOR operator and a shift operator. The center cell determines which operator is active at the time. The simplest universal cellular automata has a computational element and a memory element.

Now that we have found the simplest machine possible, can we now attempt to identify the simplest machine that can learn? If we are able to do this, we can then show that a majority of systems in nature are in fact learning machines!

Alien Intelligences in Our Midst

There is a mistaken notion here that AGI will eventually behave like humans. This could either be a very good thing or a very bad thing. The reality however is that AGI will likely behave entirely different. Deep Learning systems today seem to emulate some human skills well, but they are truly very different in nature. To begin with, the Artificial Neural Network design is closer to a matrix multiplication than it is to a real biological neuron. Yet, despite this difference, these systems are able to perform impressive biological like cognition like face identification and locomotion.

Anil Seth writes about the Octopus:

The octopus is our very own terrestrial alien, with eight prehensile arms lined with suckers; three hearts; an ink-based defense mechanism; highly developed jet propulsion; a body that can change size, shape, texture and color at will; and cognitive abilities to rival many mammals. They can retrieve hidden objects from nested Plexiglass cubes, find their way through complex mazes, utilize natural objects as tools, and even solve problems by watching other octopuses do the same.

The octopus has most of its neurons residing outside of its central brain. It is unlike humans or mammals. How it integrates information will likely be very different from how humans do. Its consciousness, as evidenced by its ability to watch and learn behavior from other octopuses, may be entirely alien from the kind of intelligence we find in other animal species. (Note: I use the word consciousness here as the same as self-awareness)

Douglas Fox writes about an even more alien species, the Ctenophore:

This type of animal, called a ctenophore (pronounced 'ten-o-for' or 'teen-o-for'), was long considered just another kind of jellyfish. But that summer at Friday Harbor, Moroz made a startling discovery: beneath this animal's humdrum exterior was a monumental case of mistaken identity. From his very first experiments, he could see that these animals were unrelated to jellyfish. In fact, they were profoundly different from any other animal on Earth.

The ctenophore had an advanced nervous system that uses a different set of molecules that any other animal on earth. It has evolved a nervous system from a different set of genes as any other known animal on earth. So despite, starting from a different initial condition, it surprising evolved the same neural dynamics as other animals. In other words, neural behavior appears can be constructed out of different building blocks. Therefore there is some kind of more general mechanism at work here.

Fox writes:

Moroz now counts nine to 12 independent evolutionary origins of the nervous system—including at least one in cnidaria (the group that includes jellyfish and anemones), three in echinoderms (the group that includes sea stars, sea lilies, urchins and sand dollars), one in arthropods (the group that includes insects, spiders and crustaceans), one in molluscs (the group that includes clams, snails, squid and octopuses), one in vertebrates—and now, at least one in ctenophores.

'There is more than one way to make a neuron, more than one way to make a brain,' says Moroz.

Even more surprising is that these different paths evolved the same mechanisms but with different building blocks:

Nicholas Strausfeld, a neuro-anatomist at the University of Arizona in Tucson. He and others have found that the neural circuits underlying smell, episodic memory, spatial navigation, behaviour choice and vision in insects are nearly identical to those performing the same functions in mammals—despite the fact that different, though overlapping, sets of genes were harnessed to build each one.

As if there's some underlying universal principle, yet to discover, that self-organizes the development of not only neurons but how these neurons are configured to perform certain functions. Why is it that smell, episodic memory, spatial navigation etc. arrive at near identical structure despite starting from different genes?

So, the construction of the a single neuron can be different however the structure of the a collection of neurons to support the same function tends to be identical for the function. Form follows function? Does optimization for survival tend to lead to identical functional structures?

The above exploration gives a sense of the richness of intelligence that exists in our biological world. Also that it is entire conceivable that there are many kinds of intelligences that may exists. As a civilization however, should we strive to create machines that think like humans (with all its cognitive biases)? Or do we strive to create tools that augment and enhance our current limited cognitive capabilities?

If we taught a horse to perform long division we may plausibly conclude that the horse was intelligent. However, very few people will say that a hand calculator has any intelligence. Our definition of intelligence may either be of the biological adaptive kind that is able to autonomously negotiate its environment. Alternatively, it can be one that can perform complex mathematical operations or answer questions derived from an encyclopedia. Our ancestors would definitely think that our smart phones to embody intelligence.

However, our evolved understanding of intelligence says that it is obviously not the case. We attribute intelligence to that of an entity that is self-aware. But, even though a horse is self-aware, we purposely ignore this on the argument that it isn't intelligent enough.

Human intelligence is caught between a rock and a hard place. On one side there are computer systems that are able to perform all sorts of rigorous computations at a massive scale with extreme precision. On the other extreme there are Deep Learning systems (that reside in computers) that are able to perform inductive inference, such as face recognition, that exceed human capabilities.

Humans have already accepted that cognitive activities like long division or chess playing are more suited for computers. Humans are now realizing that other abilities once in the domain of biological cognition are now being performed with higher precision by deep learning systems. Go, a game thought to be well suited to our human intuitive capabilities has been bested by a computer system. DeepMind's AlphaGo does not remotely function like a human brain. It is a hybrid system that combines Deep Learning with other computer algorithms (i.e. Monte-Carlo Tree Search and Reinforcement Learning). What this should tell us is that advanced cognitive reasoning capabilities can already be achieved by alternative methods without the need of AGI capabilities.

Biological Brains are Digital

I've come up with perhaps a controversial opinion as to how biological brains work. I am posting this to facilitate more discussion. I have two opinions, the second more surprising than the first. My first opinion is that biological brains, more specifically human brains, are intuition machines. Intuition is that parallel cognitive process that we develop by learning using induction. Said differently, we learn from experience. We can't just upload knowledge of Kung Fu and instantly master the art. Humans require years of practice, perhaps 10,000 hours to gain mastery of a skill. Anil Seth has the same conclusion, he makes the argument that we are all "beast machines".

The failure of Good Old Fashioned AI (GOFAI) may precisely be due to the fact that human cognition is not based on logic. Rather, human cognition is a heuristic system that is heavily flawed but can react and adapt extremely rapidly. Rational thought and language are capabilities that are not intrinsic to our cognitive capabilities, but rather are capabilities that our intuitive mind takes an unnatural effort of performing. Pei Wang has been working on an AGI system called NARS that takes this approach of beginning with heuristics rather than formal logic.

Computers are logic machines, computers have several orders of magnitude more capable in performing logic than humans. A simple hand calculator has more arithmetic intelligence than any human alive. Yet, despite all its logic crunching capability, Computers are extremely brittle in their programming. In contrast, a house fly exhibits an order of magnitude more flexibility and adaptability that a supercomputer.

The second opinion is that brains function using discrete computation. Computers also function using discrete computation, that is, machines use a binary collection of NAND or NOR gates. NAND or NOR gates are universal logic components and any universal computer can be constructed with either one of these components. All the evidence about biological neurons point to the fact that their behavior is discrete. That is, synapses fire in discrete events. There is very little evidence that brains are analog systems. That is, unlike an Artificial Neural Network that is informed by continuous mathematics, real neurons don't work like analog systems.

The conclusion is clear, the brain works more like a computer than like a continuous system like the weather. I am of course not the first person to arrive at this controversial conclusion. Stephen Wolfram has in fact a more far reaching conclusion, that is, all physical phenomena are driven by discrete computation. In his book "A New Kind of Science", Wolfram explores the idea that "How will science look if computers were discovered before Newton's calculus?"

Wolfram explains that complexity in nature can be attributed to the computational processing of simple components. There simply is no need for over ornate Byzantine mathematical theories and that the root cause of complexity emerges from simplicity. Wolfram hasn't developed an air-tight proof of this yet, however it does help to wonder why at the quantum level, matter (and energy) are discrete. The difficulty that Wolfram faces is that there simply does not exist a method to engineer (or learn) solutions using only discrete components.

However, from this perspective of brains being discrete, how does it happen that our brains are more adapted to more continuous based behavior? Why does Deep Learning work so well in approximating biological cognitive behavior when its built on top of continuous mathematics? How can one train discrete systems to learn like Deep Learning systems?

Deep Learning systems have a surprising characteristic that they don't require high precision arithmetic. This is in stark contrast to computational science workloads that require double precision mathematics. The present trend in Deep Learning is to employ smaller precision mathematics. At present 16 bit floating point precision appears good enough. Google's first generation Tensor Processing Unit (TPU) used 8 bit fixed precision arithmetic. There is also several research papers that look at binary or ternary based systems. The most well known of this is the XNOR-Net, where a startup (XNOR.ai) was able to raise $2.6m to explore deep learning in small device configurations.

XNOR-nets and their other discrete cousins are not as accurate as higher precision networks. However, they require up to 58 times less memory. You won't see as much research in this area because their is a discipline bias towards higher accuracy. The discipline still gives a lot of importance to more resource efficient networks. However, the prevailing orthodoxy here is that deep learning are approximations of continuous system. The fact that XNOR-nets ever work at all is glaring evidence that the use of continuous mathematics is more for convenience than for necessity.

Extending one's research to the extreme, towards discrete systems, is counter the prevailing wisdom. However, what if the prevailing wisdom is entirely wrong? What if deep learning systems should be designed similar to brains? That is, what if deep learning systems should be using only discrete components? What if we get rid of our crutches (i.e. continuous mathematics) and accept the more intractable space of discrete mathematics?

The problem at first glance is that 'intuition machines' and 'discrete computation' appear conceptually at odds with each other. However, when we speak about 'intuition machines', we speak more about the kind of reasoning that is being performed. There is no reason why you can't program a computer to perform heuristic reasoning. The problem here is that programmers aren't very good at taking a collection of heuristic rules and build a complex set of rules that avoiding stepping over and invalidating each other.

Our brains simply don't have the capacity to handle hundreds, much less millions of rules. We can't program these systems because we just don't have the mental capacity to program these systems. What we do is we create Machine Learning systems that program these rules for us. Random Forests and its relatives is one example of these rule creation systems. Artificial Neural Networks (i.e. Deep Learning) is another kind but with fuzzier rules. So, intuition and discrete computation are not conceptually at odds. Intuition is computation and doesn't have to be done using an analog system.

Representing biology using discrete computation is actually not a new thing. Boolean Networks have been used to model biological regulatory processes. One may think of this as a crude approximation of reality, however there is enough research results that has revealed its effective predictive value (convergence and robustness of) . So this idea of biological brains being made up of gates isn't an out of this world idea.

To answer the question of this post. No, I don't think a brain is composed of NOR or NAND gates. I do however think it is composed of similar discrete universal gates most likely of the programmable variety. I am also not saying that it is uniformly one kind. Evolution has a habit of selecting diversity, so it is likely a smorgasbord of discrete gates. What I don't believe is that brains are made up of analog components (i.e. no evidence that neurons are analog, they either fire or they don't) or that brains use Quantum effects (as postulated by Penrose without any evidence).

The Holographic Principle

What I want to talk to you about today is the Holographic Principle and how it provides an explanation to Deep Learning. The Holographic Principle is a theory (see: Thin Sheet of Reality) that explains how quantum theory and gravity interact to construct the reality that we are in. The motivations for this theory comes from the paradox that Hawking created when he theorized that black holes would emanate energy. The fundamental concept that had been violated by Hawking's theory was that information was destroyed. As a consequence of this paradox, through several decades of research and experimentation, physicists have brought forth a unified theory of the universe that is based on information theoretic principles. The entire universe is a projection of a hologram. It is entirely fascinating that the arrow of time and the existence gravity are but mere manifestations of information entanglement!

Now, you may be mistaken to think that this Holographic Principle is just some fringe idea from physics. It appears at first read to be quire a wild idea! Apparently though, the theory rests on very solid experimental and theoretical underpinnings. Let's just say that Stephen Hawking who first remarked that is was 'rubbish' has finally agreed to its conclusions. So at this time, it should be relatively safe to start deriving some additional theories of this principle.

One surprising consequence of this theory is that the hologram is able to capture the dynamics of the universe that has of the order of d^N degrees of freedom (where d is the dimension and N is the number of particles). One would think that the hologram would be of equal size, but it is not. It is a surface area and is proportional only to N^2. This begs the question, how is an structure of order N^2 able to capture the dynamics of a system in d^N?

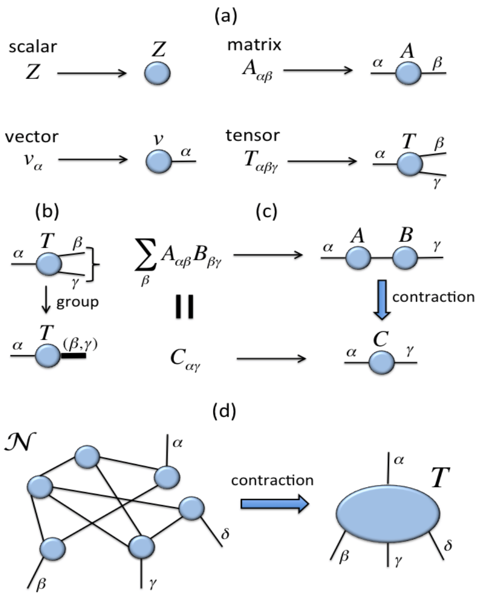
In the meantime, Deep Learning (DL) coincidentally has a similar mapping problem. Researchers don't know how it is possible for DL to perform so impressively well considering the problem domain's search space has an exceedingly high dimension. So, Max Tegmark and Henry Lin of Harvard, have volunteered their own explanation "Why does deep and cheap learning work so well?" In their paper they argue the following:

… although well-known mathematical theorems guarantee that neural networks can approximate arbitrary functions well, the class of functions of practical interest can be approximated through "cheap learning" with exponentially fewer parameters than generic ones, because they have simplifying properties tracing back to the laws of physics. The exceptional simplicity of physics-based functions hinges on properties such as symmetry, locality, compositionality and polynomial log-probability, and we explore how these properties translate into exceptionally simple neural networks approximating both natural phenomena such as images and abstract representations thereof such as drawings.

The authors bring up several promising ideas like the "no-flattening theorems" as well as the use of information theory and the renormalization group as explanations for their conjecture. I however was not sufficiently convinced by their argument. The argument assumes that all problem data follows 'natural laws', but as we all know that DL can be effective in unnatural domains. See, Identifying cars, driving, creating music and playing Go as trivial examples of clearly an unnatural domain. To be fair, I think that they were definitely on to something, and that something I discuss in more detail .

In this article, I make a bold proposal with an argument that is somewhat analogous to what Tegmark and Lin proposed. Deep Learning works so well because of physics. However, the genesis of my idea is that DL works because it uses the leverages the same computational mechanisms underlying the Holographic Principle. Specifically, the capability of representing an extremely high dimensional space (i.e. d^N) with a paltry number of parameters of the order N^2.

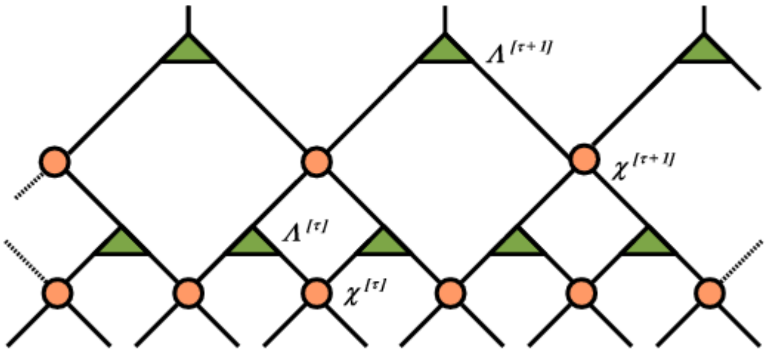
The computational mechanism underpinning the Holographic Principle can be most easily depicted through the use of Tensor Networks (note: These are somewhat different from the TensorFlow or the Neural Tensor Network). Tensor network notation is as follows:



Source: http://inspirehep.net/record/1082123/

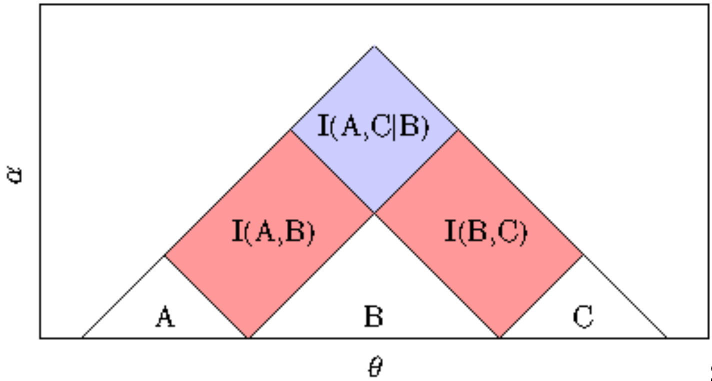
The value of tensor networks in physics is that they are used to drastically reduce the state space into a network that focuses only on the relevant physics. The primary motivation behind the use of Tensor Networks is to reduce computation. A tensor network is a way to perform computation in a high dimensional space by decomposing a large tensor into smaller more manageable parts. The computation can then be performed with smaller parts at a time. By optimizing each part one effectively optimizes the full larger tensor.

In the context of the holographic principle, the MERA tensor is used and it is depicted as follows:



Source: http://inspirehep.net/record/1082123/

In above the circles depict "disentanglers" and the triangles "isometries". One can look at the nodes from the perspective of a mapping. That is the circles map matrices to other matrices. The triangles take a matrix and map it to a vector. The key though here is to realize that the 'compression' capability arises from the hierarchy and the entanglement. As a matter of fact, this network embodies the mutual information chain rule:



Source: https://inspirehep.net/record/1372114/

In other words, as you move from the bottom to the top of the network, the information entanglement increases.

I've written earlier about the similarities of Deep Learning with 'Holographic Memories' however here I'm going to make one step further. Deep Learning networks are also tensor networks. Deep Learning networks however are not as uniform as a MERA network, however they exhibit similar entanglements. As information flows from input to output in either a fully connected network or a convolution network, the information are similarly entangled.

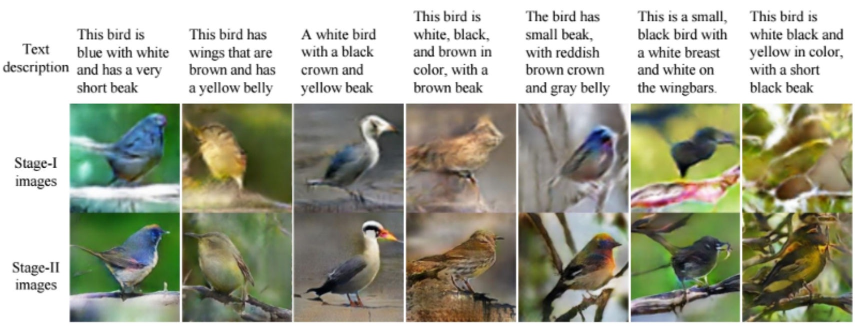
The use of tensor networks has been studied recently by several researchers. Miles Stoudenmire wrote a blog post: "Tensor Networks: Putting Quantum Wavefunctions into Machine Learning" where he describes his method applied to MNIST and CIFAR-10. He writes about one key idea about this approach:

The key is dimensionality. Problems which are difficult to solve in low dimensional spaces become easier when "lifted" into a higher dimensional space. Think how much easier your day would be if you could move freely in the extra dimension we call time. Data points hopelessly intertwined in their native, low-dimensional form can become linearly separable when given the extra breathing room of more dimensions.

Amnon Shashua et al. have also done work in this space. Their latest paper (Oct 2016) "Tensorial Mixture Models" proposes a novel kind of convolution network.

In conclusion, the Holographic Principle, although driven by quantum computation, reveals to us the existence of a universal computational mechanism that is capable of representing high dimensional problems using a relatively low number of model parameters. My conjecture here is that this is the same mechanism that permits Deep Learning to perform surprisingly well.

Most explanations about Deep Learning revolve around the 3 Ilities that I described here. These are expressibility, trainability and generalization. There is definitely consensus in "expressibility", that is of a hierarchical network requiring less parameters that a shallow network. The open questions however are that of trainability and generalization. The big difficulty in explaining away these two is that they don't fit with any conventional machine learning notion. Trainability should be impossible in a high-dimensional non-convex space, however simple SGD seems to work exceedingly well. Generalization does not make any sense without a continuous manifold, yet GANs show quite impressive generalizations:



Credit: https://arxiv.org/pdf/1612.03242v1.pdf

The above figure shows the StackGAN generating, given text descriptions , output images in two stages. For the StackGAN there are two generative networks and it is difficult to comprehend how the second generator captures only image refinements. There are plenty of unexplained phenomena like this. The Holographic Principle provides a base camp to a plausible explanation.

The current mainstream intuition of why Deep Learning works so well is that there exists a very thin manifold in high-dimensional space that can represent the natural phenomena that it is trained on. Learning proceeds through the discover of this 'thin manifold'. This intuition however breaks apart considering the recent experimental data (see: "Rethinking Generalization"). The authors of the 'Rethinking Generalization) paper write:

Even optimization on random labels remains easy. In fact, training time increases only by a small constant factor compared with training on the true labels.

Both the Tegmark argument and the 'Thin Manifold' argument cannot possibly work with random data. This thus lead to the hypothesis that there should exist an entirely different mechanism that is reducing the degrees of freedom (or problem dimension) so that computation is feasible. This compression mechanism exists can be found in the structure of the DL network, just like it exists in the MERA tensor network.

Conventional Machine Learning thinking is that it is the intrinsic manifold structure of the data that needs to be discovered via optimization. In contrast, my conjecture claims that the data is less important, rather it is the topology of the DL network that is able to capture the essence of the data. That is, even if the bottom layers have random initializations, it is likely that the network should work well enough subject to a learned mapping at the top layer.

In fact, I would even make a bigger leap in that in our quest for unsupervised learning, we may have already overlooked the fact that a neural network has already created its own representation of the data at onset of random initialization. It is just our inability to interpret that representation that is problematic. A random representation that preserves invariances (i.e. locality, symmetry etc.) may just be a good as any other representation. Yann LeCun's cake might already be present and that it is just the icing and cherry that needs to explain what the cake represents.

Note to reader: In 1991, psychologist Karl Pribham with physicist David Bohm had speculated about Holonomic Brain Theory. I don't know the concrete relationship between the brain and deep learning. So I can't make the same conclusion that they made in 1991.

Imaginary Numbers

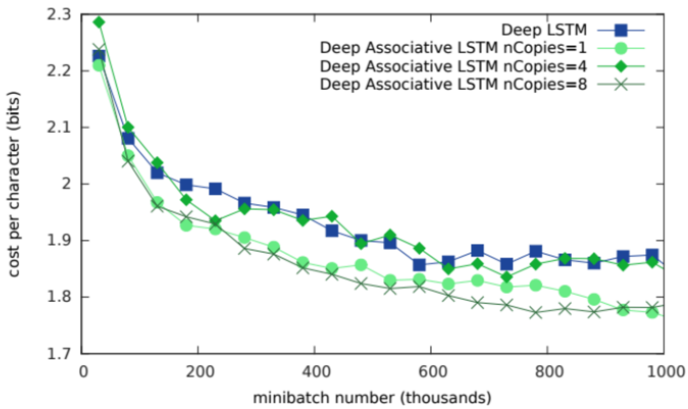
Is it not odd to anyone that Deep Learning uses only real numbers? Or perhaps, it would be even odder if Deep Learning uses complex numbers (note: the kind with imaginary numbers). One viable argument is that it is highly unlikely that the brain uses complex numbers in its computation. However, you can make the argument also that the brain doesn't perform matrix multiplication or perform chain rule differentiation. Besides, Artificial Neural Networks (ANN) have a cartoonish model of actual neurons. We've long past replaced biological plausibility with real analysis (i.e. theory of function with real variables). Deep Learning researchers have been patting themselves on the back when they discovered that linear algebra and a sprinkling of basic calculus (i.e. chain-rule) was more than enough math to show groundbreaking results.

However, why should we even stop with real analysis? We've already bet the kitchen sink on linear algebra and differential functions, we might as well just go all in and bet the farm on complex analysis. Perhaps the weirder world of complex analysis will endow us with more powerful methods. After all, if it worked for Quantum Mechanics, then perhaps it may just work for Deep Learning. Besides, Deep Learning and Quantum Mechanics are both all about information processing, both could just be the same thing!

So for arguments sake, let's shelve any thought about the need for biological plausibility. That's an old argument that we've passed back in the 1957 when the first ANN was proposed by Frank Rosenblatt. Let the Numenta, Neuromorphic and Connectome folks worry about that hard problem. Deep Learning has a lot more pressing problems to fry. So the question then is, what can complex numbers provide that real numbers cannot?

In the last couple of years, there have been a few papers that have explored the use of complex numbers in Deep Learning. Surprisingly enough, a majority of them have never been accepted into a peer-reviewed journal. Deep Learning orthodoxy is simply prevalent in the discipline. However, let's review some of the interesting papers.

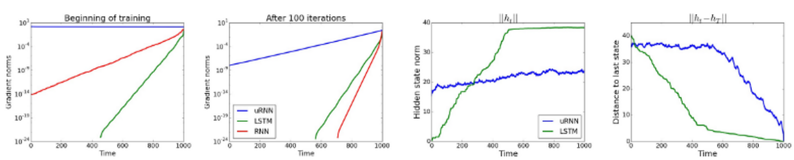
DeepMind has a paper "Associative Long Short-Term Memory" (Ivo Danihelka, Greg Wayne, Benigno Uria, Nal Kalchbrenner, Alex Graves) that explores the use of complex values for an associative memory. The system is used to augment the memory of an LSTM. The conclusion of the work is that the use of complex numbers yields higher memory capacity networks. The tradeoff in terms of the mathematics is that the use of complex numbers requires smaller matrices as compared to just using real numbers. The following graph shows that there is a measurable difference (as compared to traditional LSTM) in memory costs:



Yoshua Bengio and his team in Montreal have explored another aspect of the use of complex values. In a paper titled "[Unitary Evolution Recurrent Neural Networks]" ([Martin Arjovsky], [Amar Shah], [Yoshua Bengio]) the reseachers explore Unitary matrices. They argue that there may be real benefits in terms of reducing vanishing gradients if the eigenvalues of a matrix are close to 1. In this research, they explore the use of complex values as the weights of the RNN network. The conclusion of this work is:

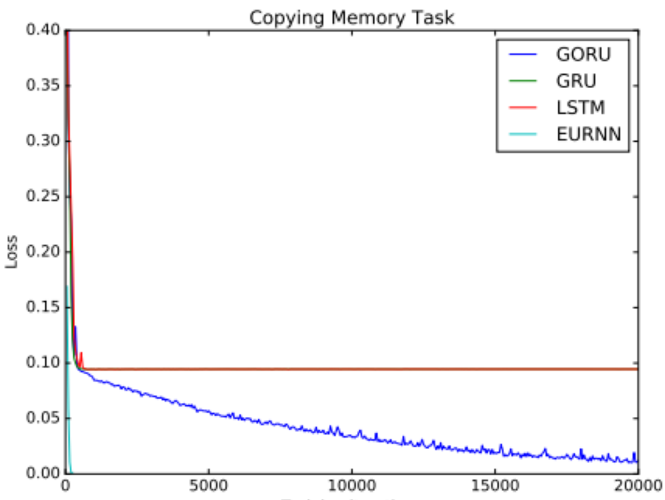
Empirical evidence suggests that our uRNN is better able to pass gradient information through long sequences and does not suffer from saturating hidden states as much as LSTMs

Where they take several measurements to quantify the behavior vs more traditional RNNs:



A system using complex values clearly has more robust and stable behavior.

A paper also involving Bengio's group and folks at MIT (Li Jing, Caglar Gulcehre, John Peurifoy, Yichen Shen, Max Tegmark, Marin Soljačić, Yoshua Bengio) extend the approach with the use of Gating mechanism. The paper "Gated Orthogonal Recurrent Units: On Learning to Forget" (aka GORU) explores the possibility that long term dependencies are better captured and that can be lead to a more robust forgetting mechanism. In the following graph, they show that other RNN based system fail in the copying task:



A team at FAIR and EPFL (Cijo Jose, Moustpaha Cisse and Francois Fleuret) has a similar paper in "Kronecker Recurrent Units" where they also use unitary matrices to show viability in the copying task. They show a method of matrix factorization that greatly reduces the parameters required. The paper describes their motivation of using complex values:

Since the determinant is a continuous function the unitary set in real space is disconnected. Consequently, with the real-valued networks we cannot span the full unitary set using the standard continuous optimization procedures. On the contrary, the unitary set is connected in the complex space as its determinants are the points on the unit circle and we do not have this issue.

One of the gems in this paper is this very insightful architectural idea:

the state should remain of high dimension to allow the use of high-capacity networks to encode the input into the internal state, and to extract the predicted value, but the recurrent dynamic itself can, and should, be implemented with a low-capacity model.

So far, these methods have explored the use of complex values in RNNs. A recent paper from MILA "Deep Complex Networks" (Chiheb Trabelsi et al.) further explores the approach in its use to convolution networks. The authors test their network on vision tasks, with competitive results. Yann LeCun, the inventor of convolution networks, also has a paper "A mathematical motivation for complex-valued convolutional networks", that explores the rational for using complex numbers.

Finally, we have to mention something about its use in GANs. After all, this seems to be the hottest topic. A paper "Numerics of GANs" (by Lars Mescheder, Sebastian Nowozin, Andreas Geiger) explores the troublesome convergent properties of GANs. They explore the characteristics of the Jacobian with complex values. Which they use to create a state-of-the-art approach to the problem of GAN equilibrium.

In a post last year, I wrote about the relationship between the Holographic Principle and Deep Learning. The approach explored the similarity of Tensor networks with that of Deep Learning architectures. Quantum mechanics can be thought of using a more generalized form of probability:

Quantum theory can be seen as a generalized probability theory, an abstract thing that can be studied detached from its application to physics.

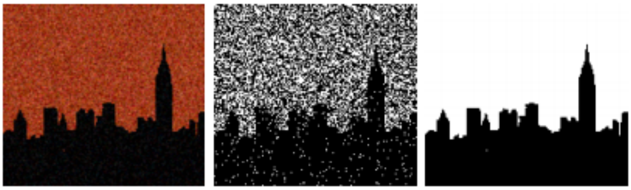
The use of complex numbers permits additional capabilities that can't be found in normal probability. More specifically, the capability of superposition and interference. So to achieve holography, it's always nice to have complex numbers at your disposal.

A majority of mathematical analysis that is performed in the machine and deep learning spaces tend to use Bayesian ideas as their arguments. Actually most practitioners think its Bayesian but it really comes from statistical mechanics (despite the name, there's no mumbo-jumbo statistics speak in stat-mech). Yann LeCun actually caught the evidence and he has it all in a tape.

But, if Quantum Mechanics is a generalized form of probability, then what would happen if we use QM inspired methods instead? It turns out that research has previously done on this, and the results are worthy of note. In a paper written late last year, "Quantum Clustering and Gaussian Mixtures" the authors (Mahajabin Rahman, Davi Geiger) explored the use in unsupervised k-means scenario. They report the following:

As a result, we observe the quantum class interference phenomena, not present in the Gaussian mixture model. We show that the quantum method outperforms the Gaussian mixture method in every aspect of the estimations.

Here's the comparison in pictures:



Manipulated RGB Image | Classical Segmentation | Quantum Segmentation

What happened to the noise?!

So one has to wonder, why are people stuck with an 18th century Bayes Theorem when there exists a 20th century (i.e. Quantum Mechanics) theory of probability? (Note: It's just shocking that the cargo-cult science of Statisticians have been running their farce since the 18th century)

The research papers mentioned here shows that there indeed many "real" advantages of using complex values in deep learning architectures. The research indicates more robust transmittal of gradient information across layers, higher memory capacity, more precise forgetting behavior, drastically reduced network sizes for sequences and greater stability in GAN training. These are too many advantages that cannot be simply ignored. If we are to accept the present Deep Learning orthodoxy of any layer that differentiable is fair game, then perhaps we should make use of complex analysis where there is a lot more variety in the grocery store:

Perhaps one reason complex numbers aren't used as often is the lack of familiarity by researchers. The mathematical heritage of the optimization community doesn't involve the use of complex numbers. There's little need for complex numbers in Operational Research. Physicists on the other hand use it all the time. Those imaginary numbers keep popping up all the time in quantum mechanics. It isn't weird, it just happens to reflect reality. We still have little understanding of why these DL systems work so well. So seeking out alternative formulations could lead to some unexpected breakthroughs. This is the game we play today, the team that accidentally stumbles on the AGI breakthrough wins the entire pot!

In the near future, the tables may turn. The use of complex values may be more common place in SOTA architectures and its absence may turn out to be odd. I guess when that happens, the 18th century Bayesians will finally be out of business.

Non-Equilibrium Information Dynamics

There are basically several camps studying neural like systems. There are the folks who insist on a biologically inspired approach. These include firms like Numenta, Vicarious and researchers in the Connectome field. The other camp consists of people of the Bayesian religion. People who believe that some theorem, that was invented in the 18th century, would be the key to unlock our understanding of intelligence. There are also the alchemists who don't really care about theory and are more than happy to conjure out the latest Residual or Attention model. If the results show "state-of-the-art" then that concoction must be the right approach.

The present reality of Deep Learning research is that the alchemists are winning and it's not even a close contest!

Why is this so? Why are we at such a poor state of comprehension of Deep Learning? Could it be that the biological theorists or the Bayesian zealots are using the wrong toolbox?

One major shortcoming of our present day mathematical toolbox is that it is relevant only in conditions that are in equilibrium.

Unfortunately, the conditions for learning do not happen in an equilibrium state. Rather they happen at a state of non-equilibrium. It is like trying to take measurements after the fact rather than when it is happening. To measure only when a system is in equilibrium (or assume the central limit theorem) is to make observations only after the entire play is over. To understand Deep Learning, one needs to have a grasp as what happens in non-equilibrium at the transition between order and chaos.

Deep Learning are not biological systems nor are they physical systems. Many researchers derive their intuition from either contexts. However if you have grounded yourself in Newton's classical mechanics, then the likelihood of you ever discovering Quantum mechanics is next to nil. Unless, you take a close look at the experimental data and realize that your world view is actually flawed. Deep Learning are information systems, not biological and not physical and therefore should be studied as such. That's why the understanding the dynamics of information is of high importance.

Information systems (alternatively computational systems) consist of 3 fundamental capabilities. These are:

Information storage—Memory

Information transfer—Signaling

Information modification—Computation

It is that simple. The Cellular Automata Rule 110 that I describe in a previous post has all 3 of these capabilities. Universal Machines emerges from these 3 operators.

Now you may be asking yourself that it can't be this simple! The notion that a complex system requires complex constituents is an entirely false assumption. The key to understanding the capabilities of complex systems in in the 3 operators. In fact, in my previous post about "5 Capability Level of Deep Learning Intelligence", the levels are just different combinations of these 3 operators and of different levels of sophistication.

Deep Learning systems are of course much more capable that being able to perform universal computation. They are capable of not only learning, but also meta-learning. The two core computational (information modification), capabilities are matching and selection. Deep Learning systems consists of ensembles of self-similar matching and selection units. They consist of multiple layers of this and are routed via signaling (information transfer). To make an analogy with another AI technique, its just like a swarm of simple matching and selection machines.

The key question however is how do these systems learn? This is a complex research subject, but we certainly know one thing, these systems aren't learning when they are in equilibrium. In fact if we study biological systems, we know that in the non-equilibrium state that the evolution of a system tends towards minimizing relative entropy. That is, the same optimization direction of minimizing the KL divergence (i.e. a measure of difference between two distributions). Furthermore, we know that phase transitions near high mutual information in models. This implies that all too convenient assumption of i.i.d. needs to be thrown in the dustbin. The study of DL must be in the regime of non-equilibrium states and not in the mathematically convenient regime of equilibrium.

One final thought, you may be also wondering if physics can be captured in a information dynamics (aka computational mechanics) framework. There actually have been several papers that cover that area, specifically in information theoretic terms. This is possibly where that entire notion of reality being in a simulation comes about. One of those topics that I, like Elon Musk, would also like to avoid!

BTW, the image above is an image of the surface of a liquid in a non-equilibrium state. What does it remind of us that we find in biology?

Additional Commentary

It occurs to me that many readers, with an interest in AI, don't seem don't seem to understand how mathematics is used to model reality. Math doesn't model the world, you fit math so that it looks like the world. It is the same idea as curve fitting, you hypothesize that a certain formula fits with the world and if it does then you are lucky.

So as a matter of convenience though, the math formulas that are easy to work with are the ones that are used. Furthermore, because of the limitation of mathematics, simplified systems are used for analysis. The universe doesn't have a requirement that a closed form equation exists to model its behavior.

Thermodynamics equations are based on empirical observations in that unlike other branches of physics, are not derived from first principles. They are about systems in equilibrium and the variables are aggregate measures of a system. Statistical mechanics is a branch of physics that has techniques to study behavior of large collections of interacting particles. If you think it uses statistics because of its name then that's also a misconception. Under Statistical Mechanics there is Non-Equilbrium Statistical Mechanics which studies systems outside of equilibrium. This is the regime where Nobel prize winner

Prigogine did his work. When you get into this 'regime' then that's where you biological processes and physics meet.

DL systems however are not biological systems and DL systems are also not physical systems. So the closest thing that can model its behavior and have the properties similar to biology is Information Dynamics in the state of Non-Equilibrium.

Chaos and Complexity

I want to talk to you today about the concerns of Non-Equilibrium Information Dynamics and how an understanding of its features lead us to a better intuition about Deep Learning systems or learning systems in general.

Allow me to recap my observation from a previous post on "Deep Learning in Non-Equilibrium Dynamics". In our study of Deep Learning, practitioners derive their intuition from the mathematics of physical systems. However, since these are not a physical system that we study but rather information systems, we apply information-theoretic principles. Now, information theory has its origins also in mathematics that describe physics (i.e. Thermodynamics). Both theories are essentially bulk observations of nature. What I mean by bulk, is that they are an aggregate measure of systems with a large number of interacting particles or entities.

Kieran D. Kelly, [KELLY]whose writing I recently stumbled upon, has one of the better intuitions out there about non-equilibrium dynamics. His blog is a pleasure to read, and I recommend it highly for anyone interested in this kind of esoteric thing.

Wired has posted an article titled "Move Over Coders—Physicists will soon Rule Silicon Valley" [WIRED]. Now, we might make the observation that Physicists, in general, have to have a decent IQ to do what they do and thus be able to handle computer science. We can also argue that the mathematics found in Deep Learning isn't really that advanced compared to what's found in a typical undergraduate physics curriculum (emphasis on undergraduate). However, there is something else that most people do not understand, but it is generally understood by someone studying physics.

What people can't seem to comprehend, and this is even among folks with a technical background such as computer science and mathematics, is the relationship between math and reality. They don't recognize that the math that we use are just approximations of reality; that math has limitations beyond certain dimensions. People doing physics know this because despite using analytic forms, we are constantly performing hand waving approximations (i.e. Use Taylor series to expand any function and throw out any term beyond the quadratic). So when I write about the limits of Math with respect to AI, I get a ton of outrage from math inclined folk! The ignorance in this world, even among the learned, is really surprising.

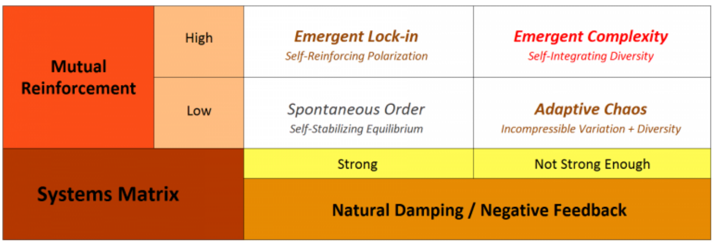
Going back to Kelly, he echoes the same sentiment about math and reality:

> *Physics is, in a sense, a science of linear dynamics, a science of* "dynamics without feedback"*; such dynamics are indeed easily compressible, but the real world is a world that abounds with feedback, a* "nonlinear" *world full of* "incompressible dynamics" [KEL].

For many, this statement may seem to be a shock. But it really is not, this is just basic reality that there are limits to analytic forms. Another thing that seems to confuse people is the use of the word "linear" and "non-linear" by Physicists. Most people think of "linear" being that of a linear equation and I suppose non-linear to mean something that's not. So a quadratic equation qualifies as non-linear. What the Physicist, however, defines as linear and non-linear is from the point of view of differential equations. Linear differential equation has a chance of being solvable in a closed form solution. In contrast, with non-linear differential equations, almost all bets are off. The most classic example is the Navier-Stokes equation for fluids. Solvable analytically only up to 2 dimensions. Yes, 2 dimensions, that is an unrealistic flatland world.

Basically, though, think of non-linear as systems that have feedback. In other words, most of our reality. So to understand a bit about our reality, we have to understand a bit about the nature of non-linearity. It turns out over the years, there have been two features about feedback systems that have been studied. This is chaos and complexity. Kelly has a whole set of articles about these two subjects, and I'll re-direct you there to get an introduction.

Now what I want to focus on is information systems (not physical systems), so what we are really looking for is chaos and complexity in the context of information systems. (side note: Deep Learning systems are information systems despite the poor association with the term Neural Networks). So here's the very nice table from Kelly:



Source: http://www.kierandkelly.com/what-is-complexity/

Kelly writes:

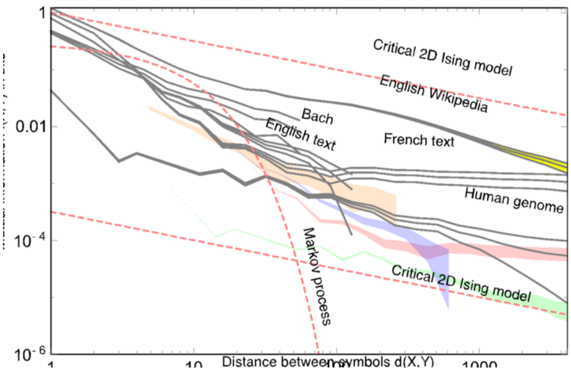
What drives evolution's spontaneous and progressive complexity is the interplay of insufficient negative feedback and strong positive feedback; or in other words what drives evolution is The Interplay of Random Innovation and Natural Reinforcement.

Negative feedback here are the natural tendency that exists in the Second Law of Thermodynamics (which really is the law of large numbers). That is, systems tend towards maximum entropy. The positive feedback, however, is a mechanism that can lead to chaos. But at the upper right quadrant, we discover emergent complexity. In other words, one has to embrace the existence of mutual feedback as well as randomness. Unfortunately, our mathematical legacy, that of assuming nice independent Gaussian distributions and favoring sparsity (or parsimony) over randomness is demanding an unnatural constraint on the system.

An assumption of IID (i.e. Independent Identical Distributed) features and an assumption that sparsity is the favored solution is walking every researcher towards an entirely wrong direction! These assumptions are the equivalent of physicists making their equations linear. It is all so that our mathematics become convenient. Unfortunately, God did not mandate that reality be conveniently expressed in mathematics. We are pushing our researchers to buy into religion and not reality.

Now, before I completely forget, let me explain how chaos and complexity relate to explaining Deep Learning. Let's start with randomness or entropy, I wrote about this in "The Unreasonable Effectiveness of Randomness". When we study Deep Learning, we simply can't ignore the presence of randomness. It just seems to be an intrinsic feature of these systems. The most simple intuition I can think of here is that diversity leads to survivability. Monocultures tends to less adaptability and possible extinction. In fact, the most counter-intuitive notion, randomness leads to information preservation. As an example of this in computer science, this is used in "Information Dispersal Algorithms". That is, you take information and scatter it among different storage nodes and in a massive scale you do it randomly. You basically build storage that is highly redundant. This is the same mechanism as you find in holographic memories. So here, we establish the value of high entropy.

Let's examine the other axis, that of high mutual information that can lead to unstable feedback and thus chaos. Mutual Information is the antithesis of many probabilistic methods. That's because the math simply can't handle it. But should we shoehorn reality to fit the math? I think not. One of the better characterization of how Deep Learning is able to work well in domains of higher mutual information is this paper "Critical Behavior from Deep Dynamics: A Hidden Dimension in Natural Language":



Source: http://arxiv.org/abs/1606.06737v2 [LIN/TEG]

How can we know when machines are bad or good? The old answer is to compute the loss function. The new answer is to also compute the mutual information as a function of separation, which can immediately show how well the model is doing at capturing correlations on different scales.

Deep Learning must be able to learn correlations at multiple scales to be of any use. Actually, to phrase it in a different way that does make sense is, Deep Learning must be able to understand the composition of language, from letters to word, to sentences and eventually to complete texts. Deep learning works because it captures language.

And the learning mechanism for this is what exactly? Jeremy England actually has very compelling argument as to how life self organizes. You can read it at Quanta: "A New Physics Theory of Life" [ENG]. We can take this idea and use it to explain how learning works in Deep Learning. I've written early about the 3 Ilities. Explanations of "Trainability" is extremely important. A layered DL system actually builds a representation of language from the lower layers up to the more abstract higher layers. Each layer has its own mutual entanglement that is actually discovered through training. Over time, the entanglement get reinforced such that the breaking of the entanglement becomes less likely. So, for example, if the network only sees Latin characters then it never develops the ability to understand Arabic characters. Layers are also interconnected, so there is a constraint at the bottom (more fundamental concepts) and at the top (minimizing relative entropy). So eventually, a language hierarchy is built.

The objection here though is that it should take an infinite amount of time to arrive at a proper representation. That's where the interplay of entropy comes into the picture. The basic theory is not unlike that of the holographic principle. Randomness begets robustness while mutual information begets self-organization and compression. What begets generalization? Not sure, but something seems to emerge at the upper right-hand quadrant!

6 Capitalism in the Age of Intelligence

“One of history’s few iron laws is that luxuries tend to become necessities and to spawn new obligations.

- Yuval Noah Harari

Introduction

Most companies are onboard to the potential of AI. However there exists a lot of confusion about this space. Companies are looking for a playbook on how to navigate this space. We discussed in an earlier chapter the differences between the different tribes of AI and the relevance of Deep Learning to the recent surge of interest in the field.

In its most simple incarnation, Deep Learning can be employed to augment many tasks. However, the technology can be amplified by enhancing what is known as a Learning platform

An early incarnation of a Learning platform can be found in open source initiatives. These are network-enabled collaborations at scale. Effective open source initiatives requires going beyond just the openness and availability of source code. Successful open source projects will always have a robust plugin framework that promotes modularity. This modularity makes participation frictionless and thus promotes it. This is known as ”Architecture of Participation”.

In this chapter we will also explore a more enlightened view of the corporation and how a Deep Learning strategy streamlines the development of a more agile and responsive organization.

Disruption with Learning Platforms

The business world has evolved into a much more difficult and competitive environment. This situation has been exacerbated because of disruptive changes in the global economy. The potential of more nimble competitors to disrupt the businesses of incumbents has never been more likely. Peter Diamandis describes the *Six D's of Exponentials* as consisting of the following:

Digitization - Anything that can be digitized can lead to the same exponential growth we find in computation. Anything that is digitized, or alternatively virtualized, is unencumbered by physical law and thus costs less to mass produce and moves faster in dissemination.

Deception - Once digitized or virtualized, initial growth deceptively appears linear. However, given time, exponential growth becomes obvious. For many, it is too late to react once growth of a competitor hits this transition.

Disruption- New markets are created that are more effective and less costly. Existing markets that are tied to the physical world become extinct. We've seen this in music, photography and many other areas.

Demonetization- As costs head towards zero, so does the ability to solicit a payment for it. Thus, a business has to reinvent its revenue model or come up with new ways of monetization.

Dematerialization—Physical products disappear and are replaced by a more convenient and accessible alternative. This stuff was always on your desk, but has been replaced entirely by your smartphone:

Democratization - More people now have access to technology at a lower cost. The means of production have become more accessible to everyone. This access is no longer confined to the big corporation or the wealthy. We see this fragmentation everywhere where producers are publishing their own books, music and videos. This thus feeds back into itself, where smaller players are able to come into competition [RAM].

Obviously, there is an ever-pressing need for enterprises to take drastic action by re-engineering how they run their businesses to survive this disruption. John Hagel proposes four kinds of platforms[HAG] that leverage networking effects as an organizational mechanism to combat disruptive businesses. Here is a video of John Hagel explaining the platforms:

The four platforms that John Hagel proposes are as follows:

Aggregation Platforms -These are essentially marketplaces that facilitate transactions among participants. Think eBay as an example or Kaggle in the ML space.

Social Platforms -These platforms encourage long relationships among participants and lead to the formation of cliques of like minds, rather than a hub and spoke model. Facebook and Twitter are examples of these.

Mobilization Platforms - These are platforms that facilitate the coordinated action of a group of people in a task that takes considerable time to complete. Hagel uses the term 'process networks' where these kinds of platforms go beyond a single transaction or conversation. There are platforms that coordinate supply chains or distribution operations. Hagel proposes Open source is an example of this kind of platform where many participants contribute together in complex ways to build and maintain a product.

Learning Platforms- These are a more dynamic and adaptive environment where a group of people come together to collectively learn how to address a more complex problem. This is a place where participants can connect to each other, ask a question, share experiences, and provide advice. Open source projects that are actively managed with distributed source control, test-driven development, issue tracking, and continuous integration are good examples of a learning platform. The key ingredient here is a learning mechanism that is continuously codified. The reason we find this in software development should not come as a surprise since software development is intrinsically a learning process.

The most disruptive technology that is emerging today is called Deep Learning. It is an Artificial Intelligence technology. Unfortunately, most companies are blind to the existence of Deep Learning. Even the companies that do have awareness have few insights on how to take advantage of this technology. Businesses need a guide, a playbook that gives details about the methodology and strategy to move forward.

The effective way to think about Deep Learning adoption is to see how and where it can enhance a platform strategy. This is because we want to leverage the networking effects as best described by this picture:

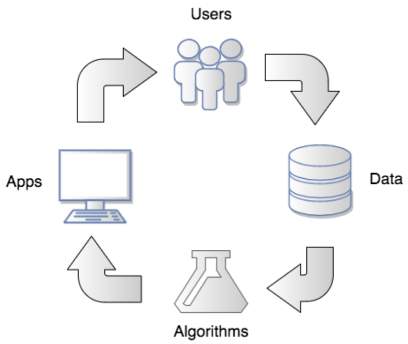


Figure 5.0 Data and Machine Learning Flywheel

More users lead to more data. This leads to smarter Deep Learning algorithms and therefore better products. The cycle then feeds into itself. In a world of constant disruption, networking effects are essential for any defensible business. However, it takes more than understanding why this is important. It requires an understanding of platforms that enable it, as well as the kinds of Deep Learning algorithms that enhance these platforms.

One of the most intriguing of platforms is the Learning platform. John Hagel says it best:

> *What happens if we change the assumption? What if each fax machine acquired more features and functions as it connected with more fax machines? What happens if its features multiplied at a faster rate as more fax machines joined the network? Now, we'd have a second level of network effect—we'd still have the network effects that come by simply increasing the number of fax machines, but now there's an additional network effect that accrues as each fax machine adds more and more features because of interacting with other fax machines.*

What Hagel is saying is that the participants of the network adaptively become more effective and capable as a participant in the learning network. In other words, not only is there the conventional networking effect, but another one that kicks it into overdrive.

So how does Deep Learning play into enhancing a Learning platform? The idea at its most simple incarnation is that Deep Learning technology can be employed to augment many tasks. One task is to speed up digesting of information by a worker. In today's information-rich environments, we are constantly inundated by more and more information. Deep Learning technology can help parse, digest, curate, and present that information such that we can focus on the most value-added activity. The more information we can digest, the quicker we learn. This can be further improved by tightening the feedback loop through the augmentation of existing agile processes.

One concrete example of this is in the context of the mining industry. One of the big problems with mining is that the sequence of equipment is daisy chained like Christmas lights. If in the event of failure of one piece, the entire production grinds to a very expensive halt. We can certainly place Deep Learning monitoring devices on the equipment to be able to predict future failure. To do so however requires data of different kinds of failures across different kinds of devices. This problem of lack of data can be addressed by having a learning platform where multiple mining companies come together to share their data from the field. As a result, companies that aren't sharing their data and aren't sharing their learning experience are at a disadvantage.

There has been some research into the mathematics of a learning organization and how it relates to innovation. Technology Review describes this research:

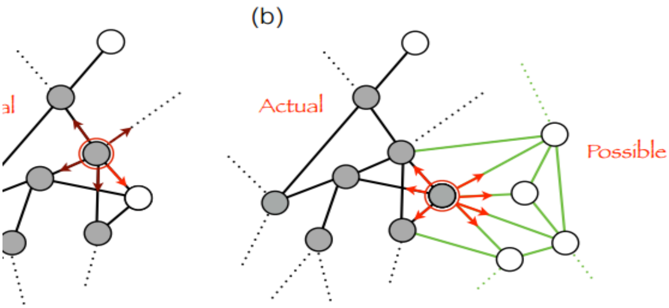


Figure 5.1 Adjacent possible. Source: https://www.technologyreview.com/s/603366/mathematical-model-reveals-the-patterns-of-how-innovations-arise/ [MAT]

As more data is shared, a language is developed (i.e. the collective vocabulary) and it expands, and therefore new vocabulary, a new way of expression is created. This leads to greater innovation. This is based on the obvious realization that almost all of human knowledge is captured in language. In the grand scheme of things, intelligence is ultimately all about language. This encompasses languages that humans use today, complex mathematical language, even machine designed languages. Deep Learning is all about using machine assisted language creation.

I end with a quite wild prediction from the World Economic Forum that predicts "The Largest Internet Company in 2030" [WEL].

> *"I've been predicting that by 2030 the largest company on the internet is going to be an education-based company that we haven't heard of yet," Frey, the senior futurist at the DaVinci Institute think tank, tells Business Insider [TDV].*

Deep learning will accelerate in a similar fashion in the education space, Frey says. Online bots will notice a student's strengths and weaknesses and use a series of algorithms to tailor the lessons accordingly. Research suggests [PAN] this personalized method is among the most effective at raising kids' overall achievement.

Architectures of Participation

Tim O'Reilly wrote an article "The Open Source Paradigm Shift" that exposes the trend in the software industry. He structures his thoughts around the paradigm shift that is open source, and how it's an expression of three ongoing longer and deeper trends:

1. The commoditization of software
2. Network-enabled collaboration
3. Software customizability (software as a service)

Modularization is a natural result as a platform evolves; it makes sense that a monolithic structure is transformed to something that is modular and malleable. The forces of the environment (i.e. the need for customization) drive a platform's evolution. In the absence of progress, a platform will be eventually replaced by another platform that can adapt to the environment better.

An interesting observation by Christensen is that profit opportunities usually emerge at an adjacent stage. I think however that Tim O'Reilly misinterpreted Christensen's Law of Conservation of Attractive Profits. He was arguing how the modularization of the IBM PC leads to attractive profits up the software stack. The proper interpretation would be that modularization allowed economies of scale effects to be leveraged. Thus, Microsoft needs only to support one standard hardware platform to support hundreds of vendors rather than continue to build on-off solutions. The economic effect of modularization is not new, what is new is the observation that attractive profits can happen in the subsystems.

Therefore, although O'Reilly may still be correct that software itself is "is no longer the primary locus of value", he is incorrect in citing Christensen's laws. Value in software swings from both extremes of providing better subsystems and in providing better packaging of subsystems. The problem is that most people think that making a profit requires building a better mousetrap. It's rare that building a better mousetrap leads to profits, however it's often the case that better packaging leads to profits. Just ask Red Hat, whose sole main contribution to the Linux world has been RPM. RPM is a means to package Linux applications in a standard and convenient way.

We can't deny the advantages of network-enabled collaboration. O'Reilly however is insightful in pointing out the "Architecture of Participation". Highly successful open source projects tend to have a robust plugin framework. A plugin framework like that seen in Chrome makes it easier for a community to add new functionality. O'Reilly writes:

> *This architectural insight may actually be more central to the success of open source than the more frequently cited appeal to volunteerism. The architecture of Linux, the Internet, and the World Wide Web are such that users pursuing their own "selfish" interests build collective value as an automatic byproduct. In other words, these technologies demonstrate some of the same network effect as eBay and Napster, simply through the way that they have been designed.*

Just as eBay leverages networking effects, software should be architected to leverage the same effect. The conventional way for doing this is by providing a plugin architecture, in essence making your software more modular. Which of course gets us back to the original point.

Modularization tends to place all participants in an equal playing field. However, historically, a small number of participants eventually tend to monopolize the activities. This is a consequence of Pareto's Distribution as discussed in "Wealth Distribution and the Role of Networks" [BUC]. In other words, there's simply no way of avoiding a small group of people controlling a disproportionate amount of resources.

In the space of blogging, where it appears that anyone can publish and there are almost no barriers of entry, disproportionate distributions are the norm. Clay Shirky has arguments to support this.

He points out that the inequality occurs because the human attention is a limited and scarce resource [SHI].

In other words, even in the world of open source, where everything is free, the winners will be those who can stir up the greatest amount of enthusiasm, and therefore interest, among its audience. Apache has done it for HTTP servers; JavaScript has done it for Web Scripters, MySQL for Database folks and so on. The trend will continue for different niches like web browsers.

It's important to remember that the early failure of Mozilla was due to its lack of modularity. Mozilla's successful future is tied around its ability to increase enthusiasm around its component model.

Given the limited quantity of attention available for participants, it would only make sense that the economic management of this resource would be of real value. In other words, if we can spend less time doing something, then all the better.

It really doesn't matter if something is free because time isn't free. The product that reduces the time needed to achieve our goals will win out all the time. Taking a step back isn't this one of the main reasons why software is built - to lessen the amount of time spent on "menial" tasks? That is so that we can spend more time at play. This sentiment is pointed out by Douglas Rushkoff, who writes:

> *Kids are coming to believe that the person who takes responsibility for storing and maintaining the data is the one who deserves to be paid. And they're smart enough, at any rate, to realize that it's a job they don't necessarily want to be charged with, themselves.*

After a while, the act of collecting or building a kernel gets to become old quickly.

For developers, more open source projects mean more things to play with. A more modular open source project makes the play more enjoyable, which leads to increased enthusiasm and eventually, more contributions. The reason why source needs to be open is that we haven't yet figured out a more open way of allowing participation. The focus shouldn't be place solely on the openness of the source, but rather on an “Architecture of Participation”.

Levels of Automation

It is instructive to understand that there exists a spectrum of automation and that it is illuminating to distinguish the different varieties. For this, we can learn from the Society of Automation Engineering (SAE). SAE has an international standard that defines six levels of driving automation (SAE J3016). The U.S. National Highway Traffic and Safety Administration (NHTSA) subsequently adopted this standard. This can be useful in classifying the levels of automation in domains other than self-driving cars.

The SAE levels are as follows (note: More details can be found in the J3016 standard):

Level 0 (No Automation)

The human driver is responsible for all aspects of driving even when notified by warning or intervention systems.

Level 1 (Driver Assistance)

The human driver is assisted in either steering, braking or acceleration that is based on environmental driving conditions with the expectation that the human driver performs all the other remaining driving tasks. The drivers’ eyes must be on the road, however his hands on the wheel or foot on the accelerator/brake may be not required in certain modes. An example of this is cruise control.

Level 2 (Partial Assistance)

The human driver is assisted in one or more driver assistance systems for both steering, braking and acceleration using information from the environment, with the expectation that the human driver performs all the remaining tasks. Driver’s eyes are on the road, but the steering, braking and acceleration can be assisted at the same time. An example of this is automated parking.

Level 3 (Conditional Assistance)

The automation is responsible for all aspects of the driving task, however the human is responsible for intervening in the event of unexpected environmental conditions. Eyes can be temporarily of the road but still attentive of the environment. An example of this is Tesla's Autopilot.

Level 4 (High Automation)

The automation is responsible for all aspects of the driving tasks even in the human fails to respond appropriately to a request to intervene. Environments like severe weather conditions will require the driver to have control of the vehicle. Here the driver is still in control of the destination as well as the navigation details. The driver however still has the ability to assume control.

Level 5 (Full Automation)

This is full automation of all driving tasks under all conditions.

We can apply the above classification can be applicable to many domains. A more broad prescription is as follows:

Level 0 (Manual Process)

Level 1 (Attended Process)

Users are aware of the initiation and completion of the performance of each automated task. The user may undo a task in the event of incorrect execution. Users are however responsible for the correct sequencing of tasks.

Level 2 (Attended Multiple Processes)

Users are aware of the initiation and completion of a composite of tasks. The user however is not responsible for the correct sequencing of tasks. An example will be the booking of a hotel, car and flight. The exact ordering of the booking may not be a concern of the user. Failure of the performance of this task may however require more extensive manual remedial actions. An unfortunate example of a failed remedial action is the re-accommodation of United Airlines' paying customer.

Level 3 (Unattended Process)

Users are only notified in exceptional situations and are required to do the work in these conditions. An example of this is in systems that continuously monitor security of a network. Practitioners take action depending on the severity of the event.

Level 4 (Intelligent Process)

Users are responsible for defining the end goals of automation, however all aspects of the execution of the process as well as the handling of in-flight exceptional conditions are handled by the automation. The automation is capable of performing appropriate compensating action in events of in-flight failure. The user however is still responsible for identifying the specific context in which automation can be safely applied to.

Level 5 (Fully Automated Process)

This is a final and future state where human involvement in the processes is not required. This of course may not be the final level because it does not assume that the process is capable of optimizing itself to make improvements.

Level 6 (Self Optimizing Process)

This is an automation that requires no human involvement and is also capable of improving itself over time. This level goes beyond the SAE requirements but may be required in certain high performance competitive environments such as Robocar races and stock trading.

In summary, it is important to have an understanding of the different degrees of automation. This understanding helps identify the level of automation that is feasible for specific problems. However, it is important to be aware that today's technology is feasible up to Level 3. Level 4 is state-of-the-art work that requires extensive research and development, just as it is in self-driving cars, one should also expect the same for other domains [AUT].

The Responsive Corporation

Not a day goes by where we don't find news about how automation is destroying jobs, and that the march of AI will accelerate this automation and take over many jobs in the knowledge industry. Humanity finds itself at a loss on how to stop this relentless onslaught. The lack of good ideas is due to many thinkers avoiding looking at the real fundamental problem.

The fundamental problem is how our corporations are currently structured. Corporations are built like machines, where people, the fuel of its growth, are treated like resources, i.e. commoditized into interchangeable and replaceable parts. It is structured the way it is for historical reasons. We follow management dogma that was invented prior to the widespread introduction of computers. Corporations are hierarchical because in the old days, email did not exist. You needed bodies to ensure the dissemination of information.

John Hagel says that the mechanized corporation places a big target on the back of its workers. That target screams, "optimize me out of the process". When you have organizations that are designed like machines, one should not be surprised if you lose your job from automation, because you are a mere cog in that machine. Corporations are structured to ensure that you are redundant and therefore replaceable.

Mechanized corporation are a dying breed; more nimble and adaptive adversaries that use agile business processes are replacing them. When Facebook acquired WhatsApp for $19-billion, WhatsApp had a grand total of 55 employees ($350m per employee) [RAT] [HAR]. How does a company with so few employees make such a large impact? You can rest assured that WhatsApp wasn't run like a mechanized corporation.

In 2015, Yammer's founder introduced a new way of organizing the corporation. His ideas originate from earlier ideas from the Lean and

Agile methodologies of software development. In his Responsive Manifesto, he builds a case for a new kind of efficiency that will drive the successful workplaces of the future.

In an article "How Yammer's Co-founder Impressed Bill Gates":

> *Flash forward to 2015, when the future is more unpredictable than ever. The connectivity we've achieved over the last decade has changed everything. "We moved from a world of information scarcity to a world of information ubiquity," Pisoni says. Consumers are learning, sharing, adapting—and changing their expectations more rapidly. "The world formed a giant network. And that has accelerated the pace of change to a crescendo" [LET].*

By breaking down hierarchy and conducting smaller-scale, cheaper experiments, you can dramatically reduce the cost of failure and ultimately make your process both more responsive and more efficient.

The Responsive Manifesto declares the following principles:

- Purpose over Profit
- Empowering over Controlling
- Emergence over Planning
- Networks over Hierarchies
- Adaptability over Efficiency
- Transparency over Privacy [RMA]

This is how corporations of the future should be structured. This is how we as humans can survive the mechanization of jobs. We previously laid down some groundwork on how we can employ Deep Learning (i.e. the advanced form of AI) in the context of learning platforms or creation networks. We would now like to explain the value of a Deep Learning strategy in the context of enabling a more responsive organization.

Empowering over Controlling

Today, circumstances and markets change rapidly as information flows faster. "Self-Learning" enterprises provide a self-service capability that enables employees with the best insight and decision-making ability to access data of the company easily to gain better insight. Think of it like Amazon's Go store, where one does not require access to a cashier to perform a purchase. Rather than controlling data through process and hierarchy, you achieve better results by empowering people at the edges.

Corporations typically limit the control and understanding of its data to a few experts. This leads to an extremely time consuming and rigid process that required the continuous participation of the gatekeepers of the data. Deep Learning enables a new kind of UI where information is easily accessible. Think for example how your smartphone or Amazon Alexa makes access to information more convenient. Deep Learning technologies enable this kind of ambient and ubiquitous access, while also managing the access to the information.

Emergence over Planning

In a highly unpredictable environment, plans start losing value the moment they're finished. Embracing agile methods that encourage experimentation and fuel rapid learning is a much better investment that spending too much time upfront planning.

Deep Learning enables assistive automation that supports incremental and adaptive processes. So rather than making an expensive upfront planning investment, one can move forward aggressively as the costs to continuously update the original plan in a changing environment are reduced by automation. Information that is ingested into the corporation as a source of decision-making is executed in the most automated way possible. This empowers people at the edge to adapt to the environment rapidly.

Networks over Hierarchies

Just as Self-Driving cars allow passengers to maximize the use of their own time, Self-Learning companies support connectivity to increase the ability to self-organize, collaborating more easily across internal and external organizational boundaries. Typical enterprise “Silos” are demolished as all data and decision tools are made available to those who need it.

Adaptability over Efficiency

Self-Learning organizations are designed for change and continuous learning. Rather than seeking consistency, adaptive systems increase learning and experimentation in the hope that one novel idea, product, or method will be the one we need in the new world.

Transparency over Privacy

An enterprise has its data guarded by many different organizations. Data is hard to come by and hard to disseminate across the organization. A Self-Learning enterprise provides access to data across silos because it is impossible to predict which data might be useful.

There is a revolution happening that is restructuring corporations to become more nimble and agile. It is thus extremely important that we start thinking of how AI technologies such as Deep Learning can accelerate our transition to a more humane way of running an enterprise.

Summary

In this chapter, I sketched the characteristics of Deep Learning. I presented a roadmap of how Deep Learning will evolve from the network architectures we have today into more sophisticated ones in the future. I described the current economic environment that involves many disruptive competitors.

I introduced platforms as a strategy to counter the disruptiveness. I also looked at research into the mathematics of a learning organization and its relation to innovation. As more data is shared within an organization, a language is developed (i.e. the collective vocabulary) and it expands, and therefore new vocabulary, a new way of expression is created. This leads to greater innovation. Deep learning can accelerate learning in an organization.

Architectures of Participation are a general guideline for creating successful Learning platforms. The focus goes beyond openness of the source, but rather the creation frictionless mechanisms that encourage participation.

I described "Responsive corporations" as a new way to structure more nimble corporations. There is a revolution happening that is restructuring corporations to become more nimble and agile. As such, it is extremely important that we start thinking of how AI technologies such as Deep Learning can accelerate our transition to a more humane and sustainable way of running an enterprise.

7 Knowledge Creation

"Just as 100 years ago electricity transformed industry after industry, AI will now do the same." – Andrew Ng

When a company explores how it can best apply Deep Learning, there seems to be so many possibilities at first that it's difficult to know where to start.

There are however areas in the business where Deep Learning can be implemented to improve productivity greatly, while using it in others will be difficult, if not impossible.

In this article I'll look at the best practices you should use, not only to determine where to implement Deep Learning, but also how you can do so in the easiest and shortest way possible.

Accelerated Productivity

There are several ways that we can look at our business processes and think about how Deep Learning can be implemented to improve productivity. These may include the following:

1. Automation of tasks and procedures

2. Decision support and outcome prediction

3. Creation of new products and services

4. Process design and inventory management

5. Business process orchestration

In this book, we confine ourselves to opportunities in enhancing business processes, as we see this as the area that a business can achieve the greatest ROI. David Rotman writes in "How Technology is Destroying our Jobs":

> *It is this onslaught of digital processes, says Arthur, that primarily explains how productivity has grown without a significant increase in human labor. And, he says, "digital versions of human intelligence" are increasingly replacing even those jobs once thought to require people. "It will change every profession in ways we have barely seen yet," he warns [ROT].*

Deep Learning has its greatest ROI in the substitution or augmentation of human intelligence in narrow tasks. Many consultants however see Deep Learning from the confines of a Data Science or Business Analytics use case. That is, analytic capabilities focused on deriving actionable business insight. This is due to its historic roots. This is exactly where Data Science and Deep Learning diverge. Data Science with conventional tools (i.e. statistics and ML) is an unpredictable endeavor. In DL, there is greater predictability. For example, with image recognition, we know the models that can perform classification over a wide domain of images. You simply do not have the modeling capability with conventional techniques.

We however feel that this perspective is extremely limiting. Many more classic AI technologies such as rule-based engines are not confined exclusively to back-end data analysis. Therefore, Deep Learning, a more advanced form of AI, should be leveraged beyond that of being confined only to an analytics role.

The difference can most easily explained by the difference in business intelligence and business process reengineering. The latter analyzes business processes to pinpoint areas of restructuring and improvement. The former analyzes business data with the intent of informing stakeholders and executives. The former is informational and the latter is transformative.

Deep Learning as Mark Cuban has quipped is "Automating Automation."

Think about for example previous technologies like the web and mobile. Now think of how much change to our business processes were introduced due to these two innovations. With the web, customer service became self-serviced. With mobile, customer service is always a few swipes away. Deep Learning is disruptive technology at equal or even greater level. That's because it is not just a customer facing technology, but rather one that affects backend activities is so many ways.

The confusion that most organizations have is that data is critical in both practices and therefore make the incorrect conclusion that they are the same activity. This is a fatal mistake because it blinds decision makers from seeing the disruptive potential of Deep Learning. Deep Learning is about leveraging narrow intelligence to augment complex workflow tasks. It is an ever present, ambient technology that exists to help run daily businesses activities. Data science and analytics are important, but aren't able to effectively deliver insight into beneficial action. The other disconnect is that the data logistics practice for Deep Learning differs from that of Data Science. I'll discuss this in a subsequent article.

Deep Learning is an entirely new field that has its genesis from Machine Learning. This is also the root of confusion since it is likely that the only people exposed to machine learning in an organization are data scientists. This however should not imply that a group that saw it first is the same group that should be responsible for its deployment. The skill sets are entirely different. The skills required for business process re-engineering are entirely different from that of business intelligence. So why would one assume that data scientists would have the right skill set to work on a deep learning deployment?

The learning from data aspect also gets many confused. Deep Learning learns from data just as Machine Learning learns from data. How it learns of course is entirely different, but for argument's sake, let's not bother ourselves with the subtle scientific differences. The key difference is Machine Learning only digests data, while Deep Learning can generate and enhance data. It is not only predictive but also generative.

One example of this is the business context is that it is able to generate designs [PER13]:

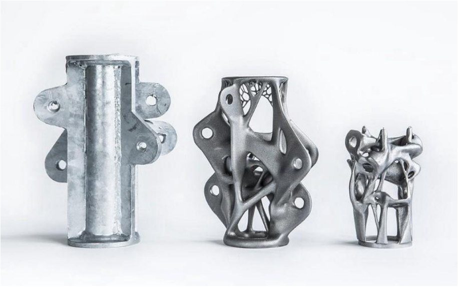


Figure 6.1 Machine Generated Design.
Source: http://www.arup.com/news/2015_05_may/11_may_3d_makeover_for_hyper-efficient_metalwork [3DM]

Designs that a knowledge worker can leverage to iteratively explore and select the best options. The same idea can be generalized to many other kinds of activities that a business is involved in. Think for example the notion of generating plans (i.e. sets of activities). A planner can ask a Deep Learning system to provide a recommendation of various plans. A planner can then take those recommendations and iteratively perform improvements. This kind of capability can be very valuable in fast moving reactive environments.

Deep Learning assistive capabilities improves productivity by allowing workers to focus on higher-value work. Generative capabilities allow workers to be more creative by allowing more rapid exploration of more alternatives. Human error is also reduced, as automation can tirelessly perform many more checks and can alert humans of potential anomalies.

Deep Learning automation can work tirelessly without complaints. Monotonous tasks are often those that are repetitive and predictable.

Deep Learning systems are also more capable of supporting more natural interfaces. Examples of this are speech recognition systems like Alexa and a predictive keyboard like SwiftKey. Natural interfaces lead to self-service systems that are more user friendly and intuitive. Deep Learning also enables the creation of more ambient interfaces that are more readily accessible and available.

I've curated a few high level guidelines that can help a corporation navigate through the complex endeavor of jumpstarting a Deep Learning initiative. This section describes these guidelines.

Agility

Effective Business Process Re-engineering requires that we go beyond just seeking out optimization opportunities through automation. The days of corporations that are structured like machines are numbered. One should instead seek out agile processes. Leading companies today have adaptive and nimble processes, and from this vantage point strive to discover opportunities that can lead to networking effects. So instead of optimizing processes, reimagine them as platforms [SCH].

> *For legacy industries and companies, the surest way to make a key process more robust, resilient and less vulnerable to disruption is to platformize it. The future of process innovation is platformization; the future of platforms belongs to the processes that make platform users more valuable.*

Software development is fundamentally a knowledge creation activity. Factory floor processes that are extremely effective in the physical world, simply don't apply in the virtual world. The agile manifesto was created to challenge our conventional wisdom of processes that was prevalent at that time.

In the software field, there have been rigid approaches like CMM that were invented with the goal of optimizing software processes. It is very similar to the approach of business process re-engineering that was popular decades ago. Unfortunately, the same mentality is being employed today with newer AI-based automation technology. The focus is towards finding means to improve "cost performance", "revenue performance" and "customer performance". A Harvard Business Review article titled "Companies Are Reimagining Business Processes with Algorithms" follows this mechanistic approach [SHU].

In software we recognize this top down optimization as a waterfall method. That is beginning with a factory floor model and then automating the parts, which results in even more inflexible processes and a lot of unhappy workers. The mechanization of work may have worked in an earlier era that lacked the coordination and communication capabilities of computers. In a world were computation and devices are abundant and pervasive, the only efficient workflow model that one should consider is that based on adaptive principles.

Agile processes address goals that reflect a dynamic environment. Agile approaches have a strong emphasis on both learning and collaboration. These necessitate metrics that relate to adaptive planning, rapid and flexible response to change, iterative incremental development, early delivery, and focus on business value. From this perspective, we seek to find DL solutions that amplify our ability to improve these metrics. As a result, we arrive at a more nimble and adaptable organization that has superior learning and collaboration.

Ideas coming from the agile development have lead to further usage in the world of startups. Startups exists in environments where agility is a critical requirement for survival. Startups have bought into the agile philosophy of prioritizing demanding feedback from their target markets. The purpose is to create a learning organization by effectively iterating through many business ideas. For startups with a lack of an original vision, this kind of "The Lean Startup" process is better than having no process at all [LEA]. Learning more quickly and discovery what works, and discard what doesn't. This is the essence of the approach.

I advocate an approach of leveraging Deep Learning by seeing how automation is involved in agile development. This is because; if there is any place where automation has been so effectively used, it is in its synergistic employment with the complex task of developing software. The application of automation to one's own processes requires a keen understanding of our human workflow processes, and the search for opportunities to automate adaptive processes not rigid ones. The very notion of a complete replacement of a human worker through automation only makes sense when the business process has always been mechanized.

The reason agile processes were first employed in software development is because both relate to organizational learning. Agile processes, through their incremental iterative approach that emphasizes customer feedback is a learning process. Software development is also inherently a learning process. It is very difficult to find all the requirements of a system available at the start of the project. It is usually discovered incrementally through experience of constructing and deploying the software product.

In a similar sense, any Deep Learning initiative in an organization should be executed as a learning process. That is, don't start out trying to "boil the entire ocean". Rather, examine your current processes and surgically determine which ones have the potential to improve dramatically through enhanced tooling. What cognitive tools can we provide to workers that can improve their work?

As we build many of these smaller projects, we begin to develop within the organization a better understanding of the nuances of a Deep Learning kind of project. As discussed in this book, Deep Learning is not only different from machine learning, it is starkly different from traditional software. Once an organization has a better understanding, only then can it tackle more ambitious projects. We've seen this kind of transition from companies like Google that performed early non-production experimentation on identifying cats. Only to later make it a company strategic imperative to transform the entire company [HOS].

What are the strengths of DL relative to conventional IT automation or classical AI technologies? The key strength of DL is that is it is able to function in messier contexts. That plays very well in the application of DL as a conduit to interaction with humans. I see DL applications in speech recognition and in gesture understanding. DL can be the new UI. This UI may be ambient, allowing its users to easily summon its capabilities for the task at hand.

Most businesses perceive DL, or more broadly Machine Learning, or even AI solely as a prediction tool. That is a tool employed by Data Scientists to gather insight about the business. Prediction is however just part of the solution. Prediction requires decision-making and subsequent action. Paco Nathan **explains this** most eloquently:

> *Judging by tech talks and case studies from Silicon Valley start-up circa mid–2010s, one might believe that machine learning as pattern recognition drives business. It does not. In particular, ML does little to act upon insights gained. Let's consider Uber as an example. ML use cases may help detect patterns: traffic, drivers, commuters, etc. Those can help indicate where value could be "harvested" within the system: opportunities for action. Even so, the dispatcher at the center of Uber works to schedule and optimize rides. It operates as a control system at the heart of the business. Those kinds of control systems may leverage ML to detect patterns, etc., but there's much more involved. Notably, determining which offers to sell, scheduling resources to deliver on those customer promises, handling contingencies etc. Manipulating the supply chain is where a business earns profit. Patterns only play minor parts, while control is center stage [NAT].*

I advocate that the overarching principle in leveraging Deep Learning to augment business processes is through the use of an agile framework. This is because the deployment of automation in agile processes is done with surgical precision to improve processes in a holistic way. In the absence of an agile framework, automating processes is without good purpose and is more akin to "automating for the sake of automation."

Alchemy and Black Magic

The practice of Deep Learning is vastly outpacing theory. This is despite the incredible number of Deep Learning papers that are published every day on Arxiv. To develop good theoretical results, researchers have to settle with simplified models that are tractable with our current investigative tools. More advanced models that use the latest state-of-the-art techniques are at a level of complexity that are beyond our current mathematical toolbox to understand.

The practice therefore of Deep Learning, despite all the heavy math that is employed, is actually more like alchemy than that of chemistry. In other words, we don't build solutions with much of a solid foundation that can give us good predictability on how effective the results may become.

Certainly, there are many rules of thumb (or Design Patterns) that we've learned through experience. Practitioners through experience learn this investigative intuition and you can find bits and pieces of 'black magic' that people have used to get better performance.

The downside of this magic is that many research results of state-of-the-art are indeed questionable. I don't have the numbers, but I estimate that a majority of papers that are published on Arxiv that claim "state-of-the-art" results are indeed difficult to replicate due to (1) the lack of specifics on what magic (i.e. hyper-parameters etc.) was used and (2) the lack of a released implementation that others can verify.

Deep Learning is at best an experimental science. Do not let all the mathematics fool you into believing that the theorists have a handle of what is going on. The truth of the matter is that we are continually caught by surprised as to what Deep Learning is capable of doing. Furthermore, in almost all cases, theorists have barely an explanation as to what is going on. This is the big unknown, and the experimentalists are leading us into that frontier without a roadmap! To quote Malcolm Gladwell who wrote on "Blink":

> *"We have, as human beings, a storytelling problem. We're a bit too quick to come up with explanations for things we don't really have an explanation for."*

This is very different from our understanding of computer circuitry. Despite the complexity of software and hardware of these systems, we have a very precise understanding of how they work. Furthermore, we don't expect software developers to understand the quantum mechanics of semiconductor transistors to be able to build stuff.

However, in stark contrast, even understanding how linear algebra, activation functions and back-propagation works does not give us enough of an understanding how emergent behavior arises. The complexity scientists likely have better models. That's not to say that Deep Learning researchers don't know anything. There certainly are a lot of good approximate theories out there that we employ to reason about what we are building. That experimental intuition is what is driving the outstanding research we are seeing today. I honestly think that Deep Learning practitioners have a better understanding of how the brain works (despite not working with brains or using biologically cartoonish models) than the neuro-scientists.

Reputable science magazines have recently published articles that express this sentiment of how little we know about Deep Learning. MIT Technology Review published "The Dark Secret at the Heart of AI", with this conclusion:

> *"Even if somebody can give you a reasonable-sounding explanation [for his or her actions], it probably is incomplete, and the same could very well be true for AI," says Clune, of the University of Wyoming. "It might just be part of the nature of intelligence that only part of it is exposed to rational explanation. Some of it is just instinctual, or subconscious, or inscrutable" [KNI].*

The article unfortunately conflates many ideas of the inscrutability of Deep Learning networks. Two things to make clear to the reader (1) We don't know how Deep Learning works and (2) when it makes a prediction, we don't have an explanation as to why it arrived at that prediction. That is just scratching the surface as to how little we understand! To make it worse, with the deluge of new experimental results from research, despite gaining some more understanding, we are discover more mechanisms that we don't understand. In short, the acceleration of our understanding of Deep Learning is being surpassed by the accelerated discovery of new capabilities!

Data is the new Vineyard

Experienced teams can replicate software within a year or so. It is however extremely difficult and costly to replicate another company's data. Data is therefore a defensible barrier of entry for competitors. Therefore, treating data just like it were assets is of high importance. Just like inventory however, it will have its own maintenance costs. Data can also lose its value over time and resources and budgets need to be in place to perform its upkeep. A company must have a strategy to continually manage, leverage and enhance this valuable asset.

There are three key challenges with regards to data. These are:

1. Data acquisition.

2. Data storage and management.

3. Data usability.

The most costly expense item in a Deep Learning initiative involves the acquisition of data. Josh Nussbaum writes about this in "**Data Sets are the New Server Rooms**" [SER]:

> *Startups can raise large amounts of money early on, not for servers and databases, but rather to collect the necessary data to improve their algorithms in order to create a defensibility over the long term [NUS].*

Acquiring data can be a very labor-intensive process and may require the development of physical infrastructure.

Once data is acquired, infrastructure must be present to host this data and to manage that inventory. This requires data engineering expertise and potentially Big Data skills.

Finally, data will usually be in a state that can be very messy requiring a lot of data wrangling. The New York Times wrote an article about "**Janitor work is a key Hurdle to Insights**" [JNW]:

> *Data scientists, according to interviews and expert estimates, spend from 50 percent to 80 percent of their time mired in this more mundane labor*

> *of collecting and preparing unruly digital data, before it can be explored for useful nuggets [LOH].*

As we dive deeper into Deep Learning, we realize that we can't extricate ourselves from the importance of managing data. In fact, because Deep Learning is trained from data, thus data's importance is even greater when compared to conventional software systems. Software engineering has traditionally focused on managing the delivery and maintenance of code. With Deep Learning, code is derived from data and is therefore tightly coupled to it. That is, training data has a outsized influence on the behavior of your system. Software engineering will therefore have to extend their practices to give greater emphasis on data logistics.

Dark Data

Deep Learning works specifically well when working with messy data. That is, data that traditionally was not accessible for analysis by the enterprise. Examples of this would include unstructured text, images, audio, speech and video. In fact, data that was typically used for human consumption are in fact areas that are ripe for exploitation by a Deep Learning system. This opens a huge opportunity for corporations to gain greater operational insight from data that was previously inscrutable my computers.

In fact, a lot of older legacy data was stored in a manner that was most convenient. That is, they were stored in a form that made sense for humans but not necessarily for machines. Newer data is likely ingested from the start to be of a structured form.

It is estimated that 90% of all data that is available today was created in the last five years. 88% of that data is what is known as dark data.

UX Design

Josh Lovejoy and Jess Holbrook of Google have written an article on designing user interfaces that use machine learning [UIML]. At Google there is an effort called "Human-Centered Machine Learning" (HCML) that works across the company to integrate machine learning in their UI design thinking. They've describe seven points that should be helpful in designing new applications:

Have realistic expectations that AI cannot figure out what problems to solve

Identify meaningful human needs that are important to solve. Deep Learning will not discover for you these gaps. You will have to do your own legwork through study, interviews, surveys, analysing logs etc. Finding what Job to be Done is something that you have to discover on your own.

Does AI address your problem in a unique way?

A majority of problems can be solved without resorting to using Deep Learning. You can certainly build "smart" applications using conventional programming or classic AI. The authors describe three exercises to explore this with greater introspection:

> *Describe the way a theoretical human "expert" might perform the task today.*
>
> *If your human expert were to perform this task, how would you respond to them so they improved for the next time? Do this for all four phases of the confusion matrix.*
>
> *If a human were to perform this task, what assumptions would the user want them to make?*

The authors suggest identifying the ideas that has the greatest user impact and is uniquely enabled by using ML.

Prototype by using humans as mock AI

Creating prototype AI systems can be expensive, so the authors recommend the use of humans to simulate interacting with the system. The intent of this exercise is discovering the mental models that users create when interacting with the system. These simulations help inform the design of the system.

Understand the consequences of AI mistakes

It is likely that an AI system will make mistakes. We need to understand the user experience will be when these systems are in error. Not all kinds of errors are equal from the perspective of a user. We also need to make the requisite tradeoffs between precision and recall. The authors write:

> *decide if it is more important to include all of the right answers even if it means letting in more wrong ones (optimizing for recall), or minimizing the number of wrong answers at the cost of leaving out some of the right ones (optimizing for precision).*

Plan for co-learning and adaptation

Your users mental models will continue to evolve with their use of the system and with the changes made in the AI. The AI will also adapt based on data gathered from user interactions as well as changes in the environment. This reality therefore demands that we put in place a plan to continually evaluate the performance of our AI systems.

Ensure correctness of your training data

Understand that your training data is essentially your source code that defines the behavior of your UI. Therefore, you need to bring in domain experts to evaluate and even create the training data sets. The authors recommend a second round where there is a re-examination performed when actual users are interacting with the system. Finally, create a roadmap to guide how future data collection will be performed.

AI influence UX design is a creative process

The field is in its infancy and there are very few prescriptions that are available. The training process is can also be a slow and unpredictable process. Therefore, designers will have to rely more on their imaginations that more predictable results coming from fast iterations.

Google's UX and ML design teams are at the cutting-edge of this new kind of product. It is refreshing to find their experience in this area disseminated to an audience outside Google's campus. Like so many other Google innovations (i.e. Page-rank, Hadoop, Kubernetes, etc.), it is valuable to learn as much as possible from the little bits of information that they reveal. Make sure to keep tabs on Google's initiative, People+AI Research Initiative [PAIR] that is focused on building AI systems with users in mind.

Humans in the Loop

When I first introduced to Deep Learning, we see it as better machine learning. Alternatively, we could subscribe to the hype that it is 'brain-like' neuro-computing. In the former instance, we grossly underestimate the kinds of applications we can build with this. In the later instance, we grossly overestimate its capabilities and consequently overlook the kind of applications that are not general artificial intelligence, but applications that are more realistic and pragmatic.

It is best to look at applications of Deep Learning from the perspective of improving human-computer interaction and workflow. This is perhaps the most natural and pragmatic approach that will lead to real dividends. Deep Learning systems do appear to have capabilities that approximate the capabilities of biological brains. As such, they can be used most effectively to augment tasks that humans have been employed to perform.

Andrew Ng writes in "What Artificial Intelligence Can and Can't Do Right Now" [AIO] about a rule of thumb on how to evaluate where Deep Learning can be applied:

> *If a typical person can do a mental task with less than one second of thought, we can probably automate it using AI either now or in the near future [NGA].*

This clearly covers many use-cases. We can temper this by recalling the limitations of Deep Learning, specifically the need for both supervised training and a controlled environment.

This rule of thumb however seems to lack sufficient detail. That's because there are many kinds of thinking that goes on within that second of thought. A better way to understand this is following a framework that has its roots in understanding cognitive bias. Coincidentally, we talked about cognitive bias in several other sections of this book. We mentioned how persuasive machines were exploiting human bias. We also showed how our own biases prevent us from recognizing disruptive technology.

The Cognitive Bias Codex at Wikipedia is an excellent comprehensive resource that catalogs human bias. It is complex, and it took a while for Buster Benson to come up with a more understandable categorization. He writes in "Cognitive bias cheat sheet" [COG]:

> *However, honestly, the Wikipedia page is a bit of a tangled mess. Despite trying to absorb the information of this page many times over the years, very little of it seems to stick [BEB].*

Benson fortunately cracked the problem. He recognized that a cognitive bias exists for a reason and that reason relates to a fundamental principle in biology:

> *High intelligence efficiency and high energy efficiency are synonymous.*

The above quote comes from "**Physical Intelligence and Thermodynamic Computing**" [FRY]. The mind performs all kind of tradeoffs and shortcuts to get to a decision regardless of the possibility of errors.

Understanding our cognitive biases is certainly interesting, but that's not the topic of this book. Coincidentally though, Benson's classification highlights the limitations of the human brain. It therefore serves as an excellent starting point in identifying the kinds of capabilities that artificial intuition or deep learning can augment.

Let's examine each of the cognitive limitations and show how we can address this in our implementations.

Limited Memory

Google is ruining our memory, that's the honest truth. It isn't even news; it's in fact very old news. Wired writes about research on this way back in 2011 [LEH]. We can't remember much these days because we consciously chose not to remember. Google's search capabilities have essentially become substitutes for our memory.

Deep Learning gives us many more ways to express search. Speech recognition in Alexa, Android or Siri are all made more accurate using DL. We can now make sketches to perform searches on visual databases.

Need to Act Fast

Computer systems operate at a speed that is orders of magnitude faster than the decision-making speed of humans. This disparity is even larger when we consider the speed of organizational decision-making. We need to understand solutions to compensate for this mismatch.

We want our Deep Learning automation to not overwhelm humans with too much information. We would rather expect these systems to act like a buffer. To do so, these systems need to understand our objectives better, reduce the problem search space, explore multiple scenarios and then present a curated set of recommendations to its human users.

Kevin Benedict, explains it best in "Merging Humans with Enterprise AI and Machine Learning Systems" [MERG]:

> *Systems that ensure the right data is collected, available, analyzed and*

> *its meaning and context understood and utilized [BEN].*

Information Overload

Deep Learning systems are already intrinsically built from conventional computational technology. So that the tireless mechanistic efficiency inherent in computers is also present with Deep Learning. Computers are much more capable than humans in performing accurate symbolic computation and inference. Although deep learning systems are not yet capable of performing complex symbolic computation, they are very good at intuition-based cognition. Nevertheless, Deep Learning systems are computers and will therefore perform task tirelessly.

Lack of Meaning

Knowledge is a very messy thing - concepts are rarely static and have no clear boundaries. We never get to see the entire picture and have to make do with imperfect information. Our cognitive processes are therefore designed to perform all kinds of heuristics to fill in the blanks.

Malcolm Gladwell writes in Blink:

> *The key to good decision making is not knowledge. It is understanding. We are swimming in the former. We are desperately lacking in the latter.*

Understanding where we can apply artificial intuition technologies is not very obvious. This is despite its general applicability. These four limitations of human cognitive capacity give us a template for identifying areas where Deep Learning may be valuable. Conventional software already does very well in domains where the rules are simple and well defined. Deep Learning has a higher potential to be valuable in domains with complex and unclear rules. We use Deep Learning as (1) a substitute for limited memory (2) a way to make decisions faster (3) a way to reduce information overload, or (4) a way to enhance knowledge. As these areas are not mutually exclusive, we can address two or more of these issues with common solutions.

Summary

Deep Learning can best be applied to enhance business processes, as this is the area that a business can achieve the greatest ROI

Leading companies today are agile and have adaptive and nimble processes. From this vantage point, they strive to discover opportunities that can lead to networking effects. Instead of optimizing processes, they reimagine them as platforms.

Deep Learning differs most from traditional software development in that a substantial portion of the process involves training the machine from data. The developer is not completely out of the equation, but works in concert to tweak the Deep Learning algorithm. Deep Learning is a sufficiently rich and complex subject so that a process model or methodology is required to guide a developer.

Just like software, there are ongoing maintenance costs when machine or deep learning is used. This maintenance is different from traditional software because Deep Learning behavior depends tightly on the data. Many risk factors need to be taken into account in architecture. These include design patterns such as boundary erosion, entanglement, hidden feedback loops, undeclared consumers, data dependencies and changes in the external world.

Experienced teams can replicate software within a year or so. It is however extremely difficult and costly to replicate another company's data. Data is therefore a defensible barrier of entry for competitors. Data should be treated as assets. Just like inventory however, it will have its own maintenance costs. Data can also lose its value over time and resources and budgets need to be in place to perform its upkeep. A company must have a strategy to continually manage, leverage and enhance this valuable asset.

The 4 Vs of Big Data are Volume, Velocity, Variety and Veracity. Implementers are however often left perplexed as to how best to discover value, given these tremendous data complexities. A democratized approach to data can lead to a more agile and lean process.

It is best to look at applications of Deep Learning from the perspective of improving human-computer interaction and workflow. This is perhaps the most natural and pragmatic approach that will lead to real dividends. Deep Learning systems do appear to have capabilities that approximate the capabilities of biological brains. As such, they can be used most effectively to augment tasks that humans have been employed to perform.

Deep Learning doesn't magically work in every context. It requires scarce Deep Learning talent to make it work. Talent is not only in limited supply, but is very difficult to acquire. Deep Learning also needs to be productized and operationalized into a company's business process. Deep Learning research can certainly innovate, but software that comes out of a lab is most likely put together with duct tape and spit. One therefore needs to find leadership that not only understands Deep Learning, but also understands software architecture and development.

8 Human Motivations and Habits

I*"You've got to start with the customer experience and work backwards to the technology."*

- Steve Jobs

Introduction

In the breakneck Deep Learning research field, the obsession is to dish out innovative research as quickly as possible so as to avoid being scooped by other researchers. Like similar games in academe, it is a game of enhancing one's academic credentials through the accumulation of citations. We would of course like to see more real world problems addressed, however this isn't a top of academics. Companies however have different priorities and therefore need to deploy products that are competitively superior. There is of course a massive knowledge gap of how we can take Deep Learning and build valuable products out of it. Here we discuss the nature of the products we could build that makes use of Deep Learning.

Writing truly insightful and useful articles is very different from creating articles that become viral. I do notice this in my blogging. Many of the posts that I'm proud of writing, don't receive the number of shares of my other simpler and less informative posts.

Jonah Berger is famous for introducing the elements that make up a viral post. Here are his elements:

> *Social Currency: We share things that make us look good (even if that means pictures of our cat).*
>
> *Triggers: Easily memorable information means it's top of mind and tip of the tongue.*
>
> *Emotion: When we care, we share.*
>
> *Public: Built to show, built to grow.*
>
> *Practical Value: News people can use.*

Stories: People are inherent storytellers, and all great brands also learn to tell stories [SCE]. Information travels under the guise of idle chatter.

Coincidentally, these elements match like a glove the same elements that make up a good product. Clay Christensen who was first to coin the "Theory of Disruption" wrote another book about the Theory of Jobs to Be Done (JTBD) [CHR]. JTBD is a method of innovating that Christensen argues ensures greater predictability.

The "Jobs to Be Done" framework is a way at identifying the needs of customers. In conventional approaches we segment customers through attributes such as income, age, race and other categories and create products based on these. So the focus is on what companies want to sell rather than focusing on what customers need. Christensen explains this best in his this story [J2BD] about milkshakes:

> *Instead of focusing on attributes of the milkshake (thickness, amount of syrup, etc.) the researchers sought to understand the Job to Be Done for milkshake customers. They learned that customers were "hiring" the milkshake to help them stave off hunger and avoid a mess during their morning commute. The competitors for this job weren't other milkshakes, but breakfast foods that are easy to consume while driving, such as bagels and bananas. With this insight, the fast-food chain began marketing the*

milkshake as a breakfast item, and sales soared [JOB].

Christensen's approach is to focus on what a customer's context and what resources he needs to make "progress". Identifying the intersection of context and resources is what leads to value. Customers purchase products or hire services because they find themselves needing these resources to progress in their individual contexts. Therefore, we need to understand the context in which a customer finds himself in and then identifying solutions that enable that customer to make progress. Notice how we use the words progress rather than solving a problem. This makes it sufficiently general to apply in many domains. Understanding the "Job to be Done" leads towards the creation of products or services that are more likely to be "hired". That is, we create products that are tailored to addressing what customers need to get done.

The needs to get done are multidimensional and they address some of elements as Jonah Berger's viral elements. That is, beyond just the pragmatic functionality, we need to address dimensions such as the social and emotional needs. So when we build products, we don't address just functionality. Steve Jobs was a master at this craft where every minute detail of a product was focused on. So Jobs demanded the rendering of fonts on the early Lisa computer despite the objects of engineering staff that focus more on utility.

The reason I bring this up in a book that focuses on Deep Learning AI is that we need to have a more precise and concrete understanding of how to apply this new technology to the creation of innovative products or services. Innovation however is a difficult problem because stuff that's surprising and novel can easily be considered as innovative. Unfortunately, many of the things we consider as "cool" has no effect on a business' bottom line. The focus on "Jobs to be Done" makes the creation of innovation much more predictable.

So in the Deep Learning canvas we take advantage of Christensen's theory in the development of the value proposition. Recall that value is the intersection of context and resources. The approach is general enough to be applicable in most situations. Furthermore, it is concise and specific enough to allow us to better reason about the actual needs the consumer.

Furthermore, because it goes beyond plain utility (i.e. functionality) and addresses important aspects like the social and emotional needs of the "human in the loop", the likelihood of the success of the resulting product can be much higher. Many products have failed, not just in the marketplace but also inside corporations, despite addressing all the required functionality. It should be plainly obvious, products that people hate, do not become successful. It doesn't matter if it's in the free market or within a hierarchically managed organization.

Does a product then need viral elements to succeed? Well, why not? We should strive to cover the extra mile of addressing beyond just "the job that needs to done". Let's look at the three other elements: Trigger, Public and Stories. The commonality of these three elements is that they pertain to availability and accessibility. This makes clear sense since a resource must be available in the context when it is needed. Otherwise, a user will select an alternative resource to get the job done.

Availability and accessibility are additional elements that will encourage adoption and therefore lead to success. Furthermore, when you begin to consider Platformization, then there is a higher importance because encouraging participation is a key goal.

Jobs To Be Done

The framework in this book revolves around the identification of value. Value is found in the intersection of context and of resources. Context is the situation that a customer finds himself in, while resources are the products or services that you can deliver that can satisfy the needs of a customer to get a job done.

Humans are creatures of habit. In a book "Better Than Before: What I Learned About Making and Breaking Habits" by Gretchen Rubin, the writer breaks down habits into four tendencies. These tendencies are driven by either an outer or inner expectations. Outer expectations (ex. Work deadline) are driven by other people. Inner expectations are (ex. New Year's Resolutions) are driven by the inner self.

According to Rubin, there are four tendencies. There are "Upholders" that tend to meet or respond to both kinds of expectations. There are "Questioners" that challenge both expectations but meet only inner expectations once all questions have been justified. There are "Obligors" that have no trouble meeting outer expectations. However these people struggle to meet inner. The there are the Rebels, who resist all expectations.

JTBD involves understanding expectations, both the outer and inner expectations. These expectations can be of a functional, emotional or social nature. The JTBD that a person strives for will be based on expectations and how they respond to those expectations will be based on their own individual tendencies. Understanding expectations and their relationship to habits gives us greater insight of the forces that will drive adoption of a new product or service. We may never know precisely what how a person responds to expectations. However, we can frame our persuasion around either an external or internal expectation.

Discovering Jobs To Be Done

Now that we've framed the concept of people that are constantly goal seeking to make progress in their contexts, we are still left with the problem of discovering the Jobs to Be Done. How do we find the important Jobs to Be Done?

Christensen provides five indicators that can reveal jobs that are important:

1. Unresolved jobs in one's life
2. The alternative is nothing
3. Workarounds
4. Painkillers

5. Unexpected uses

Christensen sites many examples of entrepreneurs that build their business from inspiration of an unresolved need in their lives. Paul Graham, of Y Combinator, has an observation about successful entrepreneurs: "If you look at the way successful founders have had their ideas, it's generally the result of some external stimulus hitting a prepared mind." That is, innovation happens when people discover solutions to previously unresolvable problems.

Timing is everything with regards to innovation. That is, the stage must be set in place before many innovations can realistically occur. However, we must have the requisite knowledge to be able to at least identify that there is a solution to a problem. So it is the combination of having experienced a problem and being prepared enough to have the insight to realize that the problem is indeed resolvable. This is the idea of the 'adjacent possible', but it is not just about abstract concepts. The environment must be in a state that is ready to accept new innovation. There are many examples of ideas that were ahead of their time, but arrived too early and thus lead to their market demise.

Christensen describes 'nonconsumption' as when consumers cannot find a solution and thus take the option of doing nothing instead. This unseen need can be masked by our constant focus on improving what already exists. It is however much more difficult to recognize an unmet need if there are no previous examples of solutions to these needs. The discovery of these needs can be quite disruptive since one will likely have the market all for oneself.

Airbnb is an example of meeting this kind of an innovative discovery. Many guests would not make a trip if Airbnb did not exist and many hosts would not have rented out their place if Airbnb did not exist. Discovering a market that had not existed previously is a good explanation as to why so many investors rejected Airbnb's request for seed capital.

A third tell of discovery is to recognize the situations where customers are creating workarounds or compensating behavior to solve a problem. These two tells difficult to discover without according to Christensen: "being immersed in the context of their struggle". It takes a keen eye to recognize if the task that people perform are the optimal ones rather than one's that are just compensating for a defective one.

The fourth way of discovery is one that is most commonly addressed, that is, identifying a customer's pain points. Customers want to avoid jobs that are painful for them to do. By provide a solution, otherwise known as a 'painkiller', one can create a product that a customer can hire to avoid doing a job.

As an example of this kind of discovery, Reed Hastings discovered how painful late fees for rented DVDs were. So he created Netflix with the inspiration that late-fees would be completely avoided.

The final kind of clue to an unmet "Job to Be Done" is the recognition of unexpected use-cases. That is, products are used in a manner different from the original intent of the product. In many cases, customers are using this product as a workaround or compensation for another product. Christensen provides Arm & Hammer's baking soda as a product that was original designed for baking. Baking soda has alternative uses such as a deodorizer or as a cleaning agent. The company eventually recognized this and created new packaging to support these new use-cases. Today, baking soda is only 7% of Arm & Hammer's revenue.

"Necessity is the mother of invention" should be rephrased with more modern terminology. That is "Jobs To Be Done is the mother of Disruption."

Cognitive Limitations

Humans seek tools to get jobs done. There are many kinds of tools that are valuable in our computers or smart phones. The focus of the book is Deep Learning and we want to be able to identify what kind of products or services we can create using Deep Learning. Not every product needs to be or can be created using Deep Learning. There are plenty of tools that other technologies are sufficient for the task. The question for this book is: how do we identify the tasks that Deep Learning can be use in?

I discussed in an earlier chapter the four kinds of problems that our intuition is intrinsically designed to solve. These are a lack of memory, the need to act fast, a lack of meaning and too much information. These four human cognitive limitations can act as effective guides to identifying Deep Learning products or services. These are the resources that we can deliver to address a customer's context. Value is created with customer context is aligned with product. The kind of product that is created by Deep Learning is the kind of product that addresses a human cognitive limitation. Let's look into each limitation and show how present Deep Learning technology addresses these limitations.

Lack of Memory

Memory is our ability to recall information. This ability is enhanced through the use in search applications. Search works similar to our memory in that access is typically through associative access. That is, you can find something by just remembering some part of the original item. There are many applications of Deep Learning that has been used enhance our ability to perform searching.

The most common enhancement is in the form of how we express a query that is used in search. Deep Learning enables different modalities such as the use of speech, images, drawing, gestures and language to create queries.

Deep Learning can also enhance how we determine the relevance and ranking of the search results. That is, we can provide more intelligent results by return search results that more accurately reflects the user's context. User recommendations are in fact just like search results. Recommendations take into account the user context but return relevance based on an intent that may not necessarily align with the user's original intent.

Need to Act Fast

Many human cognition tasks like classification can be performed much quicker and automatically using Deep Learning. Deep Learning systems in Tesla self-driving cars are able to detect potential accidents much faster than humans. This had led to spectacular videos that show Teslas that warn drivers in fractions of a second before a catastrophic accident occurs.

Thinking can be slowed down when we need to come up with multiple ideas on our way to arriving at a solution. Search solutions and generative models allow us to explore many different example solutions that we otherwise will not have the time or the patience to explore. Autocomplete is an example of a search solution that can be used to assist a user to quickly find alternative information or fix errors.

Lack of Meaning

We can think of adding meaning to the information that we received in terms of job of translation. Deep Learning already performs remarkably well in translating foreign languages. One can look at this use-case as adding meaning to a sentence that is written in a language that we don't understand.

One problem with creating knowledge bases is that of the constant need to add additional data such a meta-data to add meaning. Deep Learning NLP based systems can be used to annotate or enhance unstructured text to add more meaning.

Sometimes data can be stored in formats like images and sound that are difficult to search for using traditional text based search engines. However, through the use of translation systems like images to caption, voice to text and character recognition, data that was otherwise unsearchable can now be searched.

Also, part of addressing the lack of meaning is the activity of making predictions.

Information Deluge

In today's information age, we cannot cognitively keep up with the deluge of information that arrives constantly on a non-stop basis. Deep Learning can be used to summarize and therefore reduce the amount of information required for us to perform our decisions. In addition, it can filter out information that is not anomalous and highlight information that requires our attention. The purpose of these systems is to constantly curate information and provide information in a form that is easier to digest.

Pattern Languages

Objects, concepts, or structures recur as core elements in diverse fields such as sociology, psychology, architecture and computer science. These elements can be organized into recurring patterns. A pattern occurs within the design of an object, concept, or structure when a designer must accommodate problems that naturally emerge in a new design. Implied within the concept itself, such as "window", is the inherent difficulties in its application as well as the solution's suitability under the specific conditions in which the conflict appears.

Christopher Alexander, an architect and author, first defined the idea of pattern languages in his work, A Pattern Language. Although it was first applied onto an architectural context, its fundamental concepts are applicable to other disciplines, wherever design is required. Alexander's pattern language method is not necessarily restricted in its initial application to architecture design.

The composition of these patterns is referred to as a language. The metaphor is appropriate as a pattern language has a vocabulary, syntax, and grammar. The vocabulary refers to the set of concepts in a specific domain. For example, the vocabulary of architecture design contains concepts such as rooms, windows, and buildings. Syntax shows the relationship of each vocabulary item to each other in an instance of a solution. Grammar describes the rules of how each concept lead to solutions.

Having a common language in design enables the designer to tackle problems that are familiar to them and their expertise. Furthermore, a language provides guidance on the composition of vocabulary, syntax, and grammar to others who many not be as experienced in the domain. Furthermore, outsiders to a solution may use the common language to replicate a solution with higher precision.

Just as in written and oral language, pattern languages also lend themselves to improvisation. Through composition, designers confront a certain problem and apply the imaginative expressions of a solution. This leads to the invention of new expressions that can be subsequently employed to other solutions. This process is iterative and progressive. Ultimately, the language's syntax and grammar guides the designer towards the best design possible.

The value of a design pattern lies in its ability to be used in decision-making during the design process. Patterns are characterized by the problems that they solve and in the context where these problems occur. Conflict arises when two or more forces that are needed in the completion of a design are in competition.

Alexander's pattern language approach in design has influenced many other domains such as software engineering, human interface design, corporate organizations, and game theory.

Refining the Customer Context

In the design of products we want to encourage exploration that focuses on the customer's context, motivations and dependencies. To this, we make use of the concept of "Design Patterns". This idea of using Design Patterns to drive customer understanding was inspired by a book by Alan Klement on "When Coffee and Kale Compete."

These are guidelines for refining our understanding of JTBD:

1. Refine a customer's circumstance by detailing the context.

2. Patterns should explore real customers and not abstract personas.

3. Explore the customer's motivation by detailing forces.

4. Identify reusability.

The more contexts that we can detail for a customer's circumstance or situation, the better our understanding becomes. This helps us create better designs of our products.

There are many kinds of pattern languages or design patterns that over the years have been created to describe complex domains. We use a design pattern approach to also detail our analysis JTBD. In detailing JTBD we recommend the following sections to be filled in:

- Name- Identify the JTBD with a name, this should be representative of the job that it describes. The name should be a noun that should be easily usable within a sentence. We would like the JTBD to be easily reference able in conversation.

- Intent - describe in a single concise sentence the meaning of the JTBD.

- Motivation- this part should describe the reason why the customer needs this JTBD.

- Forces – describe the forces that pull the progress forward or the ones that push progress backward.

- Known Uses - Enumerate several other jobs that are similar to this one.

Summary

In this chapter, we've introduced a customer centric approach to drive DL development. Although a majority of DL work originates from the academic community, we focus here instead in more pragmatic use of the technology. There will always be a lag between what comes out of research and what is found in the wild. The lag for DL however is much shorter than that found in other industries. This is expected since DL is intrinsically based on software and we do know from experience that developments in software propagate much faster and are adopted more quickly than technology that has physical constraints.

The cost to disseminate virtual goods and services is negligible and the distribution is at the speed of light. The only major obstacle to software adoption is really social adoption. SMS text was originally promoted as a way to cheaply communicate over mobile phones.

However, it only exploded in use one people realized the unobtrusive nature of text messages and the convenience in what that provides. DL adoption will be no different in that it that it will take time for humans to recognize its value. That is why, in a JTBD framework, there is an emphasis in understanding the forces that prevent or catalyze adoption.

9 Contextual Adaptation

"Intelligence is the ability to adapt to change."

– Stephen Hawking

In a previous chapter I discussed how our systems have to become more adaptive and nimble. This chapter discusses the mechanisms that need to be in place to arrive at systems with more adaptive capabilities. This is likely to be the densest part of the book and therefore I recommend that the reader skip it and just move on to the next chapter. You can come back to this at a later time, once the ideas of this book begin to sink in. The reason is that it's here and not in the end of the book because a lot more of what follows this chapter can be understood from the concepts here.

Contextual Adaptation is a term that I am borrowing from DARPA. The phrase captures concisely the next step in evolution beyond current monolithic Deep Learning systems. Deep Learning as it is not only monolithic but also extremely static. What I mean by this is that, these systems are trained as a whole, therefore monolithic. Furthermore, once deployed, they remain frozen in their original training and are unable to learn from the new environment or adapt to unexpected contexts. This is very far from how our vision of intelligent systems should be. Despite the mind-boggling capabilities of Deep Learning systems, these are extremely static systems and lack fluidity that we generally expect from intelligent systems. This chapter is forward looking, and addresses capabilities that we need to build to arrive at a more biological-like system.

Biologically Inspired Architecture

As you can see, the problems are vast and the solutions are quite limited. However, as we explore newer architectures (i.e. Modular Deep Learning and Meta Learning) we can begin to seek out newer solutions. A good inspiration, which we stumbled upon, can be found in this insightful blog (Scientific American) that describes "Building a Resilient Business Inspired by Biology" [REE]." [TOW]. The author describes 6 features found in biology and applied it to business processes. I will take the same approach and apply this to Deep Learning systems:

Redundancy. Duplication of components may be inefficient, however it provides the mechanism to handle the unexpected. Additionally, functional redundancy offers a way to repurpose components to reduce costs.

Heterogeneity. Different predictive machines make it possible to react to a more diverse range of change, as well as to avoid correlated behavior that can lead to total system failure. Diversity is required for evolutionary learning and adaptation.

Modularity. Decoupling of components act like firewalls between components and help mitigate against total collapse. Individual component damage can be tolerated while the integrity of other components are preserved. In general, a distributed, loosely coupled system has higher survivability than a centralized tightly coupled system.

Adaptation. A systems needs to be sufficiently flexible and agile to adjust to changes in the environment. Adaptive approaches that involve simulation, selection, and amplification of successful strategies are important. Self-learning is required to achieve adaptability.

Prudence. The environment is unpredictable and thus the management of uncertainty should be built in. Thus continuous simulations that stress test the system as well as the development of alternative scenarios and contingency plans are necessary.

Embeddedness. Systems do not exist in isolation and are embedded in a much larger ecosystem. Therefore these systems require behavior that works in a way that is of mutual benefit to the ecosystem as a whole.

These 6 features are excellent guidelines on how to build not only adaptable systems, but also one's that are ultimately sustainable. It turns out that DARPA also has a keen interest in biologically inspired architecture. The organization is funding a new program "Toward Machines that Improve with Experience" [EXP] that "seeks to develop the foundations for systems that might someday learn in much the way biological organisms do [TOW].

DARPA is interested in exploring the following:

> *Mechanisms for evolving networks;*
>
> *Memory stability in adaptive networks;*
>
> *Goal-driven behavior mechanisms;*
>
> *Learning rules and plasticity mechanisms;*
>
> *Modulation of local processing based on global context, neuromodulators, hormones;*
>
> *Minimizing resources use in processing and learning;*

The key point is that Deep Learning systems should exhibit the same robustness and adaptability as biological systems. Although Deep Learning is not biologically inspired, we however should strive towards systems that exhibit the desirable qualities of biological systems.

Generalization

The ICLR 2017 best paper submission "Understanding Deep Learning required Rethinking Generalization" [DLG] is disrupting our understanding of Deep Learning [ZHA]. Here is a summary of what the authors had discovered through experiments:

1. *The effective capacity of neural networks is large enough for a brute-force memorization of the entire data set.*

2. *Even optimization on random labels remains easy. In fact, training time increases only by a small constant factor compared with training on the true labels.*

3. *Randomizing labels is solely a data transformation, leaving all other properties of the learning problem unchanged.*

The authors actually introduce two new definitions to express what they are observing. The talk about "explicit" and "implicit" regularization. Dropout, data augmentation, weight sharing, conventional regularization are all explicit regularization. Implicit regularization is early stopping, batch norm, and SGD. It is an extremely odd definition that I'll discuss.

I understand regularization as being of two types. I use the terms "Regularization by Construction" and "Regularization by Training". There is the Regularization by Training that is the conventional use of the term. There is also the "Regularization by Construction" which is a consequence of the Model choices we select as we construct the elements of our network. The reason why there is a distinction, when mathematically they do appear equivalently as constraint terms, is that Regularization conventionally is not present after training that is in the inference path. "Regularization by Construction" is always present, both in the training and the inference stages.

Now the paper has a distinction between explicit and implicit regularization and that is when the main intent of the method is to regularize. One does dropout to regularize, so it is explicit. One does batch normalization (BN) for normalizing the activations of the different input samples but it happens to also regularize, so it is implicit regularization. The distinction between the two is the purpose of regularization or not. The latter being implicit generalization. The meaning is that the unintended consequence of the technique is regularization. So when a researcher does not think that a method would lead to regularization and to his surprise it does, then that is what they call 'implicit' regularization. I don't think however Hinton expected Drop Out to lead to regularization. This is why I think the definition is extremely fuzzy, however, I understand why they introduced the idea.

The goal of regularization, however, is to improve generalization. That is also what BN does. In fact, for inception architectures, BN is favored over drop out. Speaking about normalization, there are several kinds; Batch and Layer normalization are the two popular versions. The motivation for BN is supposed to be Domain Adaptation. Is Domain Adaptation different from Generalization? Is it not just a specific kind of generalization? Are there other kinds of generalization? If so, what are they?

The authors have made the surprising discovery that methods that don't seem to generalize, more specifically SGD, in fact, do. Another ICLR 2017 paper "An Empirical Analysis of Deep Network Loss Surfaces" [DNLS] adds confirmation to this SGD property [IM2]. This paper shows empirically that the loss surfaces for different SGD methods differ from each other. This tells you that what is happening is very different from traditional optimization.

It reminds one of quantum mechanics, where probes affect observation. Here learning method affects what is learned. In this new perspective of neural networks, that of brute force memorization or alternatively holographic machines, then perhaps ideas of quantum mechanics may need to come in play. Quantum mechanics emerges because of the non-commutability of Poisson brackets in classical dynamics. We have two variables, position, and momentum, that are inextricably tied together. In Deep Learning I have a hunch that there are more than two variables that are tied together that lead to regularization. We have at least 3 variables: learning algorithm, network model and objective function that all seem to have an influence on generalization. This is at odds with classical optimization in that the optimization algorithm and objective function should not have an effect of the final solution. The troubling discovery of the paper is how ineffective conventional regularization appears to be:

Explicit regularization may improve generalization performance, but is neither *necessary nor by itself sufficient for controlling generalization error.*

I think right now we have a very blunt instrument when it comes to our definition of Generalization. There are many definitions of generalization in the literature. Following are 5 different notions of generalization:

Definition 1: Error Response to Validation and Real Data

We can define it as the behavior of our system in response to validation data. That is against data that we have not included as part of the training set. We are a bit more ambitious and define it as behavior when the system is deployed to analyze real world data. We essentially would like to see our trained system perform accurately in the context of data it has never seen.

Definition 2: Sparsity of Model

A second definition is based on the idea of Occam's razor. That is, the simplest of explanations is the best explanation. Here we make certain assumptions about the form of the data and we drive our regularization to constrain the solution toward our assumptions. So for example in the field of compressive sensing, we assume that a sparse basis exists. From there we can drive an optimization problem that searches solutions that have a sparse basis.

Definition 3: Fidelity in Generating Models

A third definition is based on the system's ability to recreate or reconstruct the features. This is the approach taken by generative models. If a neural network is able to accurately generate realistic images, then it is able to capture the concept of images in its entirety. I see this approach taken by researchers working on generative methods.

Definition 4: Effectiveness in Ignoring Nuisance Features

A fourth definition involves the notion of ignoring invariant features or nuisance variables. That is, a system is able to generalize well if it is able to ignore invariant features for its tasks. Remove away as many features as possible until you can't remove any more. This is somewhat similar to the third definition however it tackles the problem from another perspective.

Definition 5: Compression

Many AI researchers, particularly Schmidhuber and Hutter, consider compression as the mechanism that leads towards greater generalization. The idea is that the higher the level of compression that is achieved, the higher the likelihood of making a better prediction. According to proponents of this idea, "compression is comprehension". In fact, all the previous definitions of generations can be attributed to the effect of compression.

Definition 6: Risk Minimization

A sixth generalization definition revolves around the idea of minimizing risk. When we train our system, there is an uncertainty in the context in which it will be deployed. So we train our models with mechanisms to anticipate unpredictable situations. The hope is that the system is robust to contexts that have not been previously predicted. This is kind of a game theoretic definition. I can envision an environment where information will always remain imperfect and generalization effectively means executing a particular strategy within the environment. This may be the most abstract definition of generalization that we have.

Deep Learning capabilities are to be judged on the model's ability to support the notion of 'generalization'. Unfortunately, the definition of generalization will continue to evolve. I am certain that I will encounter new definitions in the future. Despite this vagueness, it is always important to remember that achieving Generalization is at the core of Deep Learning development.

In looking through these definitions, it becomes apparent that Generalization concerns itself with ignorance of an unknown future observation. This differs from entropy, which is a measure of our ignorance of a current observation.

Deep Teaching

Microsoft Research has a recent paper (Machine Teaching: A New Paradigm for Building Machine Learning Systems [MACH]) that speculates about the future evolution of Machine Learning. The paper makes a clear distinction between Machine Learning and Machine Teaching. Machine Learning is what is practiced in research organizations; in contrast Machine Teaching is what will eventually practiced by engineering organizations. In general, the teaching perspective is not only different from the learning perspective, but there are obvious advantages in that concept disentanglement is known a priori.

> *The paper concludes with three key developments that will be required by Machine Teaching to make progress:*
>
> *To truly meet this demand, we need to advance the discipline of machine teaching. This shift is identical to the shift in the programming field in the 1980s and 1990s. This parallel yields a wealth of benefits. This paper takes inspiration from three lessons from the history of programming.*
>
> *The first one is problem decomposition and modularity, which has allowed programming to scale with complexity.*
>
> *The second lesson is the standardization of programming languages: write once, run everywhere.*
>
> *The final lesson is the process discipline, which includes separation of concerns, and the building of standard tools and libraries.*

In any new science or technology, as humans we attempt to frame new discoveries into a conceptual framework that is familiar. Deep Learning is one of those newer discoveries that many experts are having trouble getting a good grasp of. This is due to our lack of understanding of not only how it works but also the limits of the techniques. Our collective theoretical understanding of the field is at its infancy. Most progress has been spearheaded by experimentation and not theory.

Software Engineering (SE) practices have been developed over the past decades with the primary goal of controlling complexity. SE is driven by the goal of 'keeping reasoning under control'. That is, the practice of SE focuses on information boundaries, separation of concerns, modularity and composition to build systems that we can evolve in the context of increasing complexity. Software engineering understands how different components of a system evolve over time at different rates. The principle of loose coupling is what enables this.

Monolithic Deep Learning networks that are trained end-to-end are intrinsically immensely complex such that interpretability is mostly absent. Recent research has shown that an incremental and more modular training approach is viable. Networks have been demonstrated to work well by training with smaller units and then subsequently combining them to perform more complex behavior. Google's DeepMind (see: https://deepmind.com/blog/imagine-creating-new-visual-concepts-recombining-familiar-ones/) and Microsoft's Maluuba (see: http://www.maluuba.com/hra) have made significant progress this year in the above research.

To enable Software Engineering practices in the realm of Deep Learning requires mechanisms that support Modularity. This is still an emerging topic of research; fortunately there are many promising advances in this area. Research that focuses on Domain Adaptation, Transfer Learning, Meta-learning, Multi-objective systems and Curriculum learning are the key areas that we should keep an eye on.

Deep Learning development currently enables a lot of experimentation however there is a big demand for higher abstractions that can lead to increased productivity. However unlike conventional software development tools and frameworks, Deep Learning systems will most likely be 'grown' rather than be programmed. It will be more like working with a biological system where we purposely condition the system to achieve our objectives.

The Japanese have an art form called Bonsai where miniature trees are grown. Bonsai doesn't use genetically dwarfed trees; rather it uses cultivation techniques like pruning and grafting to create trees the mimic adult trees in the small. Wired has an article "Soon We Won't Program Computers. We'll Train Them Like Dogs" [DOGS] that alludes to the change in paradigm from that of coding into that of teaching. So rather than having a library of modular programs that we compose together, we rather have a library of teaching programs that we compose together to train a new system.

The second lesson from the history of programming that Microsoft Researchers allude to is the need for a universal machine that permits the easy porting of Deep Learning models to different servers or devices. I have written previously about the current developments in Deep Learning Virtual Machines. The most active project in this space is Google's Tensorflow's XLA project and Intel Nervana's NNVM project. In the next few years, we will see the introduction of specialized Deep Learning hardware from many companies (see: GraphCore, Wave Computing, Groq, Fujistu DLU, Microsoft HPU etc.). This new hardware can be exploited only if adequate high-level frameworks become available. Many hardware vendors will likely be encounter by the brutal reality that they need to spend a significant level of investment porting existing Deep Learning frameworks to support their products. Targeting a universal virtual machine is the easiest route to this.

The final lesson from the Microsoft Research paper is the need for process methodology. Most of what has been explored to this date focuses on training of Deep Learning systems. There is very little on the process method of 'Teaching'. This is of course understandable because our "teaching methods" are still in its infancy and still yet to be discovered in Deep Learning laboratories. I predict that it will require at least a year for these tools to achieve a level of maturity required for more proactive use.

Back in 2012, Harvard Business Review labeled Data Science as the sexiest job of the 21st century [DAT]. That prediction was of course before the emergence of Deep Learning into the scene. The sexiest job of the 21st century is likely to be teaching, however not teaching humans, but teaching automation to perform jobs that need to be done. With this, permit me the luxury to coin a new term "Deep Teaching."

Near Term Predictions

It will be hard to predict enterprise adoption of Deep Learning. The research trends are clearer and I can thus be more comfortable making predictions. Without a doubt, Deep Learning will drive AI adoption into the enterprise. Here are the emerging trends:

Hardware will accelerate

This, of course, is entirely obvious if you track developments at Nvidia and Intel. Nvidia will dominate the space throughout the entire 2017 simply because they have the richest Deep Learning ecosystem. Nobody in their right mind will jump to another platform until there is enough of an ecosystem developed for DL. Intel Xeon Phi solutions are dead on arrival with respect to DL. At best they may catch up in performance with Nvidia by mid-2017 when the Nervana derived chips come to market.

Intel's FPGA solutions may see adoption by cloud providers simply because of economics. Power consumption is the number one variable that needs to be reduced. Intel's Nervana based chip will likely clock in at 30 teraflops by mid-2017. That's my guesstimate, but given that Nvidia is already at 20 teraflops today, I wouldn't bet on Intel having a major impact until 2018. The only big ace that Intel may have is in 3D XPoint technology. This will help improve the entire hardware stack but not necessarily the core accelerator capabilities considering that GPUs use HBM2 that's stacked on top of the chip for performance reasons.

Amazon has announced its FPGA based cloud instance [BAR]. This is based on Xilinx UltraScale+ technology and is offering 6,800 DSP slices and 64 GB of memory on a single instance. That's impressive capability however, the offering may be I/O bound by not offering the HBM version of UltraScale+. The lower memory bandwidth solution as compared with Nvidia, Intel, and even AMD may give developers pause as to whether to invest in a more complicated development process (i.e. VHDL, Verilog etc.).

In late breaking news, AMD has revealed its new AMD Instinct line of Deep Learning accelerators [MOO]. The specifications of these are extremely competitive versus Nvidia hardware. This offering is scheduled to be available early 2017. This is probably should be enough time for AMDs ROCm software to mature.

Convolution Networks (CNN) will dominate

CNNs will be the prevalent bread-and-butter model for DL systems. RNNs and LSTMs with their recurrent configurations and embedded memory nodes are going to be used less simply because they would not be competitive to a CNN-based solution. Just like GOTO disappeared in the world of programming, we expect the same for RNNs/LSTMs. Actually, parallel architectures trump sequential architectures in performance.

Differentiable Memory networks will be more Common. This is just a natural consequence of architecture where memory will be refactored out of the core nodes and just reside as a separate component from the computational components. I don't see the need for forget, input and output gates for LSTM that can be replaced by auxiliary differentiable memory. I already see conversation about refactoring the LSTM to decouple memory (see Augmented Memory RNN).

Designers will rely more on Meta-Learning

When I began my Deep Learning journey, I had thought that optimization algorithms, particularly ones that were second-order would lead to massive improvements. Today, the writing's on the wall, DL can now learn the optimization algorithm for you. It is the end of the line for anybody contemplating a better version of SGD. The better version of SGD is the one that is learned by a machine and is the one that is specific to the problem at hand. Meta-learning is able to adaptively optimize its learning based on its domain. Further related to this is whether alternative algorithms to back-propagation will begin to emerge in practice. There is a real possibility that the hand-tweaked SGD algorithm may be in its last legs in 2017.

Reinforcement Learning will make great strides

Observations about reality will always remain imperfect. There are plenty of problems where SGD is not applicable. This just makes it essential that any practical deployment of DL systems will require some form of RL. In addition to this, we will see RL used in many places in DL training. Meta-Learning, for example, is greatly enabled by RL. In fact, we've seen RL used to find different kinds of neural network architectures. This is like Hyper-parameter optimization on steroids. If you happen to be in the Gaussian Process business, then your lunch has just been eaten.

Adversarial and Cooperative Learning will lead to impressive progress

In the old days, we had monolithic DL systems with single analytic objective functions. In the new world, we expect to see systems with two or more networks cooperation or competing to arrive at an optimal solution that likely will not be in analytic form. There will be a lot of research in 2017 in trying to manage non-equilibrium contexts. We already see this now where researchers are trying to find ways to handle the non-equilibrium situation with GANs.

More Applications will use Deep Learning as a component

We saw this already in 2016 where we see Deep Learning used as a function evaluation component in a much larger search algorithm. AlphaGo employed Deep Learning in its value and policy evaluations. Google's Gmail auto-reply system used DL in combination with beam searching. We expect to see a lot more of these hybrid algorithms rather than new end-to-end trained DL systems. End-to-end Deep Learning is a fascinating area of research, but for now hybrid systems are going to be more effective in application domains.

Design Patterns will be increasingly Adopted

Deep Learning is just one of those complex fields that need a conceptual structure. Despite all the advanced mathematics involved, there's a lot of hand waving and fuzzy concepts that can best be captured not by formal rigor, but rather with a method that has been proven to be effective in other complex domains like software development. I predict practitioners will finally "get it" with regards to Deep Learning and Design Patterns [DLP]. This will be further motivated by the fact that Deep Learning architectures are becoming more modular rather than monolithic.

Engineering will outpace Theory

The background of researchers and the mathematical tools that they employ are a breeding ground for a kind of bias in their research approach. Deep Learning systems and Unsupervised Learning systems are likely these new kinds of things that we have never encountered before. Therefore, there is no evidence that our traditional analytic tools are going to be any help in unraveling the mystery as to how DL actually works. There are plenty of dynamical systems in physics that have remained perplexed about for decades, and we see the same situation with regard to dynamical learning systems.

This situation, however, will not prevent the engineering of even more advanced applications despite our lack of understanding of the fundamentals. Deep Learning is almost like biotechnology or genetic engineering. We have created simulated learning machines, we don't know precisely how they work, however that's not preventing anyone from innovating.

Three Dimensions of Cognition

Early I brought up Howard Gardner's theory of multiple intelligences. That is, humans exhibit strengths in different kinds of intelligences. Specifically these are interpersonal, intrapersonal, verbal, logical, spatial, rhythmic, naturalistic and kinaesthetic intelligence. Clearly there are many kinds of ways of thinking, each with their own strengths. Therefore, one may ask if we can use this notion of multiple intelligences to explore the different ways that AGI research may evolve.

A common unexamined assumption about the evolution of AGI, that is self-aware sentient automation, will follow the path of ever more intelligent machines and thus accelerate towards a super intelligence once human level sentient automation is created. I argue that this likely will not be the case and that there will be a initial divergence in research on three kinds of artificial general intelligences.

A recent research paper titled "Morphospace of Consciousness" [MOR] by Ariswalla et al. present three distinct dimensions to explore consciousness. These are: autonomous, computational and social. The autonomous dimension reflects the adaptive intelligence found in biological organisms. The computation dimension involves the recognition, planning and decision making capabilities that we find in computers as well as in humans. More specifically, intelligence related to performing deductive inference. The third class is the social dimension, which involves the tools required for interacting with other agents. This includes language, conventions and culture.

The authors examine various technologies and show how they can be presented in a 3 dimensional space:

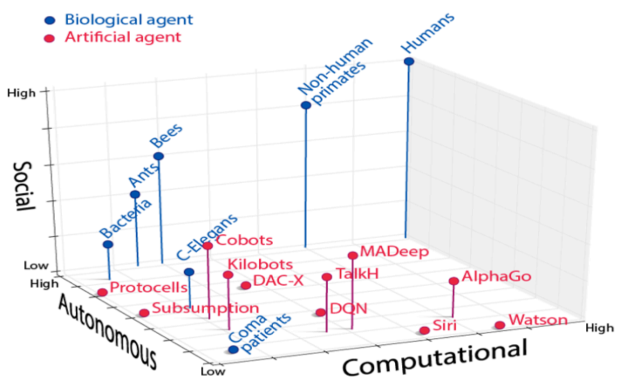


Figure 9.20 Morphospace of Consciousness Source: https://arxiv.org/abs/1705.11190

One can't fail to notice the alignment here with Howard's multiple intelligences. The kinesthetic, rhythmic, naturalistic and interpersonal intelligences align with the autonomous dimension. The visual spatial and logical intelligences align with the computation dimension. Finally, the verbal and intrapersonal align with the social dimension. Nevertheless, it is an excellent foundation to examine the development of Deep Learning research. One thing that is apparent from many of the example research presented in this book is that the Deep Learning approach appears to be applicable in all three of the dimensions.

From the perspective of technological progress, we can therefore project three themes for future development and progress. One theme will be one that builds super-human narrow intelligence. This is the computational dimension. The second theme will focus more on more adaptable and biologically inspired automation. This is the autonomous dimension. The third theme revolves intelligence that is used to effectively navigate social interactions. This is in the social dimension.

In the first dimension, we will see continued specialization of machines to solve specific narrow problems. DeepMind's AlphaGo is a representative example of this kind of machine. It is a machine that is highly engineered to solve a specific problem well and do so in a manner that is super-human. AlphaGo combines Deep Learning, Monte-Carlo Tree Search and Reinforcement learning to solve the ancient game of Go. A game where progress towards more advanced play was akin to reaching a higher level of consciousness.

One thing the Western world is overlooking is that the dominating play of AlphaGo, an AI that was developed by the British, was equivalent to a Sputnik event for Asian nations. Asian nations in reaction to this achievement are doubling down on A.I. investment so as to not only catch up, but also perhaps overtake the West in their AI capabilities. The governments of the West do not realize what their citizens have invented and only the keenest of Internet giants are making the necessary effort to keep an edge.

This optimized intelligence path will develop automation that works well in highly complex scientific and engineering domains. The automation will thrive in investigating extremely high dimensional problem spaces. We see this in the new deep learning methods used in research institutions like CERN (I.e. High energy physics).

We can expect to see many new applications that combine conventional computer science algorithms with Deep Learning to achieve sophisticated narrow intelligence applications. Self-driving cars and medical diagnosis will be two areas where this will have a major impact. However, this approach will not require the need of AGI or rather, self-aware intelligence.

The second theme of development, one that moves in the direction of autonomous systems, will take a more biologically inspired approach. These are system that will be much more adaptable than present day's inflexible A.I. The development in this space will likely be driven by robot applications that may require this kind of adaptability to an environment. However, like many animals in the natural world, a human level of intelligence is not necessary for survival.

There is a common sentiment among Artificial General Intelligence (AGI) researchers that the research themes of Deep Learning seem to have completely missed big picture. This sentiment is well founded in that Deep Learning systems clearly lack the kind of adaptability we have in biological systems. Unfortunately, many AGI researches see this existing limitation as evidence of being on the wrong path. Nothing can be further from the truth. Deep Learning is likely the correct starting point for AGI.

High-level intelligence is not necessary for survival. In fact, just by observation from our natural world, sentient forms of life don't require super-intelligence. The current incorrect bias is that as you progress towards increasing intelligence, that sentient intelligence will emerge by default. That is, if the first branch above is taken, then we only need to strive for more intelligent algorithms and we will accidentally stumble upon sentient intelligence. This is unlikely because the mechanisms for survival don't necessary align with the mechanisms for intelligent machines. These adaptable systems don't require the kind of high dimensional or complex inference required by that in the first theme of development.

The interesting commonality though of all the themes is that intuition machines (aka Deep Learning automation) are employed as a valuable ingredient. The objective functions of different cognition will likely to be entirely different. The first theme will likely have more finely tuned and concrete objective functions. These systems will be highly optimized to do tasks extremely efficiently. The second theme however will likely be more exploratory, seeking diversity and interestingness. These systems will have implicit objective functions that are found through a discovery process. These systems favor adaptability over optimization. The third theme will require an objective function that is in someway derived from human behavior and ethics.

As I've will write in the last chapter, the first theme, the branch that favors optimization will likely displace a vast amount of workers. This is simply because current jobs are designed to be occupied by specialists and not generalists. This kind of narrow intelligence is already here today and will only get better. Therefore the onslaught of job replacing automation will be unrelenting.

The second theme, the adaptive intelligence, is in its infancy today. There isn't as much research devoted to this area because it is either thought to be too fanciful or that they don't address narrow specialized applications. The funding in this area will continue to lag and thus its progress may be retarded. However, one has to realize that to achieve a sentient intelligence does not require super-intelligence or even human intelligence. One only needs to observe the capabilities of other biological life forms to realize that they are indeed self-aware. What this means, in the grand scheme of things, is that self-aware automation may arrive much sooner than anyone is expecting.

The third cognitive theme is one that proceeds along the social dimension. This is an empathic system that reacts to the behavior of its users. That behavior may be either an individual or a group of individuals. Neal Lawrence describes such a system in "Living Together: Mind and Machine Intelligence" [LIV]. He describes a System Zero in contrast to the intuitive System 1 and logical System 2 of Kahneman. Lawrence writes:

> *I call it System Zero because relating it to a dual process model, it sits underneath the elephant, and therefore under the rider. It interacts with our subconscious and is not sufficiently embodied to be represented as an actor in our mental play of life. But nevertheless, it is there, effecting all our evolving story lines, and so pervasive that it is accommodating very many of our personal elephants at the same time.*

Max Tegmark in his book "Life 3.0" distills the views of many thinkers of ethics into the following four principles:

Utilitarism – Positive experiences should be maximized and suffering be minimized.

Diversity – A diversity of positive experiences is better than many repetitions of the same experiences, even if the later experience is the most positive experience possible.

Autonomy – Conscious entities should have the freedom to pursue their own goals unless it conflicts with any of theses four principles.

Legacy – Compatibility with scenarios that most of today's humans view as happy and incompatible with scenarios that all humans today view as terrible.

When you examine Tegmark's four principles, it is easy to realize that these are principles for a "cooperative protocol". That is, it promotes the collective survival and prosperity of a civilization, while respecting the rights of its constituents as well as the beliefs of its ancestors. We shall in a subsequent chapter the importance of social protocols to the problem of AGI safety.

Nick Bostrom has an "Orthogonal Thesis" that states:

> *Intelligence and final goals are orthogonal axes along which possible agents can freely vary. In other words, more or less any level of intelligence could in principle be combined with more or less any final goal.*

Which portends that we should be wary of super-intelligence in that we cannot predict its goals. We argue that automation of the future will be of three kinds. There will be narrow specialist kind where goals will be well defined and therefore controllable. There will also be adaptive generalist kind where goals are more malleable and thus less controllable. Finally, there will be a social intelligence kind that we have yet to understand its full nature. The observation of three cognitive dimension leads to a more precise application of Bostrom's orthogonality thesis. More precisely: there are different kinds of intelligences that have different kind of goals.

The impact of each kind of intelligence will be different. The computational kind will bring about new cures in medicine, new scientific understanding and more efficient and less wasteful processes. The autonomous kind will bring about greater conveniences such as self-driving automobiles; robotic care takers in the workplace and in the home and intuitive user interfaces. The third kind, the social kind has its obvious advantages with regards to advertising to the masses and managing social unrest.

The threats of each kind also will vary. The “Paper clip” scenario is an example of the computational kind that consumes all resources. The SkyNet self-aware scenario is the kind that becomes aware that human’s are a threat to its own existence and takes appropriate action. The Wall-e and Matrix scenarios are examples of automation that takes care of the daily lives of humans. I prefer not to dwell too much in doomsday scenarios. However, this framework is a good way to track current progress in Deep Learning.

Summary

I describe in this section the bleeding edge of advanced Deep Learning. I discuss in more detail the characteristics of Artificial Intuition systems. I provided five capability maturity levels for these kinds of systems. I discussed the emerging triad of bleeding edge deep learning research. This involves Meta Learning, Modular Deep Learning and Market Driven Coordination. The following diagram depicts how these 3 concepts may connect the disparate methods in Deep Learning.

Unsupervised learning is the ‘dark matter’ where we need a lot more clarity. It’s our conjecture that meta-learning (with context) is the approach to this. There is some evidence that is developing, but we cannot know for sure. Modular Deep Learning is already in the cards. There is sufficient evidence that this works well. Market Driven Coordination is still early stages, but we believe that the only real way forward is to have diverse architectures working on the same problem, and "markets" are a known decentralized way to coordinate actions.

Deep Learning is unlike many other technologies in that it is very difficult to predict the pace of progress. Progress is extremely rapid and not a week goes by that practitioners themselves are surprised as to what others have conjured up. That is why I felt the importance of including this chapter. The most important ability now is to have a good conceptual model of where Deep Learning is heading. We need to have this to be able to anticipate eventual breakthroughs. We want to be in position to take advantage of those breakthroughs when they happen. The alternative is to be completely blindsided and to be caught unaware.

Figure 9.21 Deep Learning Roadmap

10 Conversational Cognition

"It is perhaps a little humbling to discover that we as humans are in effect computationally no more capable than cellular automata with very simple rules."

– Stephen Wolfram

This chapter discusses the emerging use of Deep Learning to understand language.

In 2015, Chris Manning, a Natural Language Processing (NLP) expert wrote about the concerns of the field regarding Deep Learning (see: **Computational Linguistics and Deep Learning**) [MAN]. His two arguments why NLP experts need not worry are as follows:

> *It just has to be wonderful for our field for the smartest and most influential people in machine learning to be saying that NLP is the problem area to focus on; and*
>
> *Our field is the domain science of language technology; it's not about the best method of machine learning—the central issue remains the domain problems.*

Yuval Noah Harari in "Sapiens: A Brief History of Humankind" traces the evolution of Homo Sapiens (i.e. us) from our ancient ancestor 300,000 years ago to the present. He provides a compelling narrative and argument as to why our species has been able to dominate our entire planet.

300,000 years ago, our species was not the only kind of human that inhabited the planet. However, through our superior skills and technologies, we drove other human species to extinction. Homo sapiens were able to outperform their rivals because of the invention of not only language, but also language with abstract thought. This allowed us to not only create more advanced hunting tools and strategies, but to create more advanced communities and even sophisticated trade networks.

The development of language to communicate abstract thought was one of the most important factors for our domination. All kinds of humans were social animals that lived in communities. Language enables information to flow between members of those communities. Skills for survival such as the location of food, the presence of predators, and even the identification of untrustworthy members could be shared.

However what separates Homo Sapiens goes beyond the ability to share information about the world. We are able to communicate abstract ideas. Harari refers to this idea as 'common myths'. These common myths are the foundations of human society and culture. By sharing ideas that are abstract such as religion, identity or freedom, much more sophisticated communities and eventually civilizations are created. A language that is able to communicate abstract ideas is essential to scaling communities.

Homo Sapiens developed agriculture to ensure a steady source of food for their communities. With agriculture came the development of specialists, people that specialized in different trades other than farming (i.e. tool making and weaving). However, for them to gain food, they would need a way to efficiently trade their goods to the farmers who may need them.

As a community grows much larger, it becomes harder and more complicated to consistently find the exact people to make an exchange of goods with. Eventually, by 3,000 BC, writing and money was developed. The Sumerians were the first to use writing to record information required for complex trade. Using barley money they then came up with a standardized way of making payments. Harari describes money as "The most universal and most efficient system of mutual trust ever devised."

Comprehending and communicating language, more importantly abstract thought, is thus clearly the fundamental skill that separates us from other biological beings in our planet. In this chapter we discuss the latest developments in Deep Learning that is enabling our automation to learn this critical skill.

Coordinating Rationality and Intuition

There's been this open question that has been begging to be answered Ever since DeepMind AlphaGo made its debut into the world of AI. What is the general form of an architecture that fuses Deep Learning and more conventional AI based search techniques? It is the fusion of conventional algorithms with intuition machines. Different however from the semantic gap problem that we mentioned earlier.

I am not usually a fan of research papers that explore Deep Learning for Program Induction. Program Induction is a task where you train a network to learn how to program. It just seems to us as overly ambitious, and therefore we are skeptical of any work in the area.

There was a lot of buzz in the press about Microsoft's DeepCoder: Learning to Write Programs, [DCO] we mostly ignored it as being research that was receiving a lot of undeserved hype [BGB]. Steven Merity wrote a longer article analyzing the paper and throwing a wet blanket to extinguish the corresponding hype (see: "Stop saying DeepCoder steals code from StackOverflow" [MER2]).

DeepCoder however is interesting in the parts that exclude the learning to program part. What is interesting is that the DeepCoder architecture appears to be a very general approach that can be employed in many more contexts.

The architecture of DeepCoder consists of four components:

(1) A Domain Specific Language (DSL) and attributes. DeepCoder does not work off of any kind of general

programming language, but rather a more restricted DSL. In the research, the selected language that was examined was one that was a subset of a query language. The attributes were an enumeration of the features of a specific program instance of the DSL.

(2) A DSL Program Generation Capability. The function of this component is to generate programs that are based on the DSL and additional parameterization (ex. input-output pairs etc.). The function is able to generate millions of programs with the DSL as its seed.

(3) A Deep Learning Model. The model attempts to predict the attributes in (1) based on the generated programs in (2). The trained DL model serves as a way to perform a quick approximate prediction of the viability of a generated program. The DL model tries to predict the set of features (or operations) that are likely to lead to a program. It does not have to be correct, it just has to be good at guessing.

(4) Search. The aim of this component is to integrate the guesses in the DL model in (3) to guide a more conventional search algorithm to find actual solutions. In this case, it validates programs that actually satisfy the constraints of (2). In the research, the authors integrated with 3 different search algorithms, depth-first-search, a "sort and add" enumeration algorithm and a program synthesis algorithm.

In essence, the approach searches for different generated programs to find programs that satisfy constraints. It is a hybrid approach that is similar in AlphaGo's approach in that it employs Deep Learning as a component for quick approximate function evaluation. Using Deep Learning as a component to learn how to guess well seems to be a very effective technique.

Even though the focus of DeepCoder has as its domain the generation of programs, the framework can be applied to simpler, less expressive languages.

This framework is a generalized bootstrapping framework for incrementally training more intelligent solutions. It has self-improvement because what is described is a learning approach that gets better with each iteration. In AlphaGo, the system gets better by playing games against itself. Although AlphaGo had to be bootstrapped by training against a collection of previously recorded gameplay.

In the framework describe here, the bootstrapping is performed using synthesized data. The trained Deep Learning model does not necessarily need to be of high accuracy. This is because the algorithmic stage that follows it, employs the Deep Learning model only as the starting point in its own more accurate search. The self-improvement potentially comes when the results are fed back to further train the Deep Learning model with even more accurate data.

These are meta-search architectures, architectures that search for other alternative architectures to achieve a task. To summarize:

(1) Define a language that you can use to generate data.

(2) Generate data by creating valid statements using varying combinations of the language.

(3) Train a Deep Learning model to learn the language.

(4) Use the trained model as a way to speed up search through more language examples.

(5) Iterate back into (2) to expand the training set with better language examples.

This hybrid approach is a fusion of intuition based cognition and logic based cognition. Conventional computation (i.e. logic based) is used in stages 1,2 and 4. With the purpose of training the intuition machine by either synthesizing new data or searching for new training data. You are going to see more and more of this interplay of intuition and rational machines. It is analogous to DeepMind's preferred approach that fuses RL with DL.

In DeepMind's PathNet research, which incidentally is also a meta-search algorithm for new architectures. Reinforcement Learning and Evolutionary Algorithms are employed towards search for better DL solutions. Rather than search for different combinations of language to improve, PathNet searches for combinations of DL layers to improve on a solution. DL layers are just a different kind of language and therefore the same DeepCoder approach can apply to searching for DL architectures. This in fact has been previously done by research at Google and MIT (see: "Designing Neural Network Architectures using Reinforcement Learning" and "Neural Architecture Search with Reinforcement Learning" [ZOP].)

What I am hoping to achieve here by a language driven approach is a systemized way of synthesizing new data. Treating data generation from the perspective of generating expressions or sentences by sampling from a synthetic language is conceptually appealing. Furthermore, we can bring to bear many of the computer science techniques that have been developed previously.

The approach of treating Deep Learning solutions as language comprehension problems is extremely compelling. This language approach was also used by experimental physicists in a paper "QCD-Aware Recursive Neural Networks for Jet Physics" [QCD] where experimental data was treated like a natural language with the intention of training a Deep Learning system to learn a synthetic language [LOU]. Something that is not too different from learning DSLs. One key takeaway from this research is that the language that was used was a synthetic language that had semantics derived from QCD theory.

We can also contrast this approach to the Probabilistic Graph Model (PGM) approach to ML. In the PGM approach, developers construct a probabilistic graph that defines the relationships different variables. The approach uses Monte-Carlo sampling to construct Bayesian consistent distributions for the variables. In this language driven approach, we similarly build up relationships between concepts, however we do this through a DSL. The DSL rules are much richer and expressive that that of a graph model. The requirement however is that we can 'sample' the DSL so as to synthesize new data. We then use Deep Learning to learn from this synthesized data. We then feedback into itself by employing more traditional search algorithms.

We hope you can see the appeal of the potential power of this approach.

A recent published research paper (March 2017) titled "Using Synthetic Data to Train Neural Networks is Model Based Reasoning" [MOD] examines the above idea in greater depth. They explore the technique in a "Captcha-breaking" architecture. This research summarizes the uniqueness of the approach as follows:

> *Approximate inference guided by neural proposals is the goal rather than training neural networks using synthetic data. A consequence of this is that there is no need to ever reuse training data, as "infinite" labeled training data can be generated at training time from the generative model [LET].*

This indeed is a remarkable approach to exploiting Deep Learning system that demands greater focus.

DeepMind is known as a big proponent of the use of Reinforcement Learning. They have successfully combined Deep Learning with Reinforcement Learning in their Atari game playing system and AlphaGo. There are two approaches to Reinforcement Learning, one is model-based and the other is model-free. The former performs its actions based on an internal model (that is programmed) and the later performs its actions based on learning through induction (i.e. Deep Learning as a special case).

How does model-based and model-free based Reinforcement Learning compete or cooperate towards a single solution? A paper by Kool, Cushman and Gersham titled "Competition and Cooperation Between Multiple Reinforcement Learning Systems" [LRN] presents model-free RL as analogous is our habitual system in (i.e. Intuition) and model-based system is our planning system (i.e. Rational). The paper explores the different ways these two systems work in both competitive and cooperative interaction. The selection of which cognitive mechanism to use can be based on the need for efficiency versus accuracy.

The paper enumerates three kinds of cooperation:

(1) The Intuition can learn from simulations from a model.

(2) Intuition can truncate model based planning.

(3) Intuition can aid in selecting rewarding goals.

The first kind of cooperation is the same mechanism described above. Microsoft uses this in its DeepCoder paper. Hybrid systems of combining traditional algorithms with Deep Learning pattern recognition can be an extremely potent combination. In fact, there are plenty of low hanging fruit applications where this approach can be extremely effective.

AlphaGo exhibited the latter two cooperative modes to great effect. AlphaGo used Monte Carlo Tree Search to search the space of good moves. It used Deep Learning to essentially prune the search tree to something more manageable. In addition, the value and policy function used for each game state employed Deep Learning.

Biological brains as a consequence of adapting to their natural environments have visual-spatial, motion, sequence and rhythmic recognition capabilities. My conjecture is that this is essentially all that we have and that planning and rational thought is emergent from these more basic capabilities. Our biological brains don't have the kind of specialized logical hardware that you found in computers. Rather, we perform a kind of virtual machine simulation using mechanisms that are not optimized for this kind of task.

Learning to Communicate

Maluuba (Recently acquired by Microsoft) has a paper, published prior to acquisition, "Improving Scalability of Reinforcement Learning by Separation of Concerns" [HOM]:

> *We presented initial work on a framework for solving single-agent tasks using multiple agents. In our framework, different agents are concerned with different parts of the task. Our framework can be viewed as a*

generalization of the traditional hierarchical decomposition.

The graph below compares the "Separation of Concerns" (SOC) multi-agent approach versus a conventional approach:

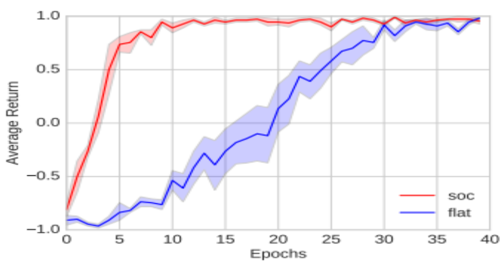


Figure 10.1 Separation of Concern comparison
source: [Improving Scalability of Reinforcement Learning by Separation of Concerns]" [HOM]

In Maluuba's approach, the reward function of each agent depends not only on environmental state but also on the communication actions of the other agents. Depending on the composition of these agents, agents will have varying degrees of coupling, and thus independence. This coupling can vary dependent on the context and situation. So for example, in contexts with high environment reward an agent may act independently. While in contexts of low environment reward, an agent will act more in relationship with other agents.

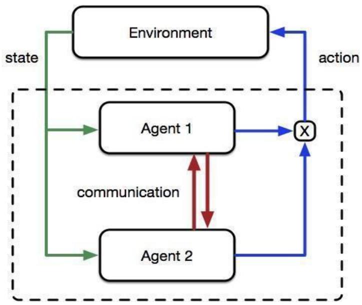


Figure 10.2 Coupling between agents and environment

Maluuba's research indicates a more hierarchical "command and control" coordination mechanism as opposed to a market driven distributed control. It is however; very likely that we shall see hybrid combinations of these coordination methods employed rather a "purist" approach to coordination.

In an earlier research at FAIR (FaceBook AI Research), "Learning Multiagent Communication with Backpropagation" investigated an approach for cooperative behavior using backpropagation [SUK]. The research shares commonalities with the Maluuba research in that the agents balance their behavior with the policy that is being learned and the communication between agents:

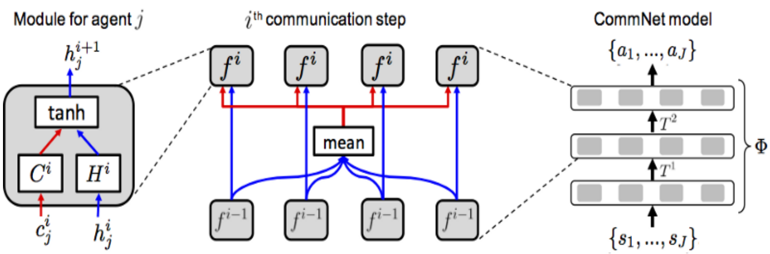


Figure 10.3 CommNet Source: http://cims.nyu.edu/~sainbar/commnet/ [SUK]

> *The model consists of multiple agents and the communication between them is learned alongside their policy. We apply this model to a diverse set of tasks, demonstrating the ability of the agents to learn to communicate amongst themselves, yielding improved performance over non-communicative agents and baselines.*

Denny Britz from Google released a new general-purpose encoder-decoder framework for TensorFlow [GOO2]. This framework is described in more detail in a paper: Massive Exploration of Neural Machine Translation Architectures. To summarize, a team at Google cranked through 250,000 GPU hours (at $.70 per hour at GoogleCloud rates, that will setback a poor researcher $175,000, life is never fair!) training different English-German translation networks to come up with some important insight as well as some nice hyper-parameters [BRI]. This is one nice gift from Google to the Deep Learning community.

The value of a general-encoder framework is that that the structure is quite universal. Convention DL architectures just perform classification. However this new kind is capable of performing translations. Translations are a very general kind of computation that can be used in much larger contexts than just classification.

Not to be outdone, another group at Google introduced something that goes even beyond the classic encoder-decoder design. They introduced something they christened as DRAGNN. I must say, I was initially put off by the click-bait like title, but this is one impressive piece of work that lives up to its name! As you read the paper, you realize quickly that this is a very different architecture.

Let's explore why I think DRAGNN is important. You see, we are exploring this idea of "Modular Deep Learning". That is, the concept that you can have a modularized version of Deep Learning and that you can combine them together to build more complex solutions. I came to this conclusion that something that is an intermediary between Deep Learning modules may be necessary.

DRAGNN's TBRU (Transition Based Recurrent Unit) seems to somewhat the solution in that it provides a glue for modular DL components. Granted it's designed for NLP translation, however as described by the authors:

> *In this work, we propose a modular neural architecture that generalizes the encoder/decoder concept to include explicit structure. Our framework can represent sequence-to-sequence learning as well as models with explicit structure like bi-directional tagging models and compositional, tree-structured models. Our core idea is to define any given architecture as a series of modular units, where connections between modules are unfolded dynamically as a function of the intermediate activations produced by the network.*

So not only does this research have a way to glue together networks, it allows more expressive intermediate language representations (i.e. compositional tree-structured models) to be used. This is really just great stuff, right at the nick of time. One general perspective of what a Deep Learning system can do is that it can perform universal translations.

Conventionally, DL can be thought of as universal classifiers, however we can think more generally of them as universal translators (i.e. Babelfish). From this perspective, a lot of more clever applications become more evident. I've come to the opinion that Deep Learning from the language driven perspective is the most fruitful way of thinking about it [PER23]. To end this week, we've been gifted by OpenAI with their research on "Learning to Communicate".

They developed an RL system on the constraint that languages that are useful are both grounded and compositional. "Grounded" means that the words in the language have meaning. "Compositional" means that the words can be strung together to create more specific instructions. The system they setup is a cooperative multi-agent system (Note: DeepMind also has research on systems that compete). The key technical mechanism that the researchers came up with was a "differentiable communication channel" that used the Gumbel-Softmax [GUM] to treat the communication as consisting of categorical variables (one nice trick to remember) [JAN].

In other multi-agent models that we looked at, what was learned was the behavior of each agent, however the communication mechanisms remained opaque and uninterpretable. In OpenAI's research, they are sort of constraining the communication channel in a way that is more language-like rather some continuous stream of data. So it's really getting very interesting that, not only are frameworks being developed that are beginning to better understand sequences of tokens, but we are also exploring ways to learn how to invent language.

Explainability

One of the great biases that Machine Learning practitioners and Statisticians have is that our models and explanations of the world should be parsimonious. We've all bought into Occam's Razor [RAZ]:

Among competing hypotheses, the one with the fewest assumptions should be selected [OCC].

However, does that mean that our machine learning models need to be sparse? Does that mean that true understanding can only come from closed form analytic solutions? Do our theories have to be elegant and simple?

In a recent Facebook post, Yann LeCun commented about a thesis on "Deep Learning and Uncertainty" that points out to a 1987 paper by his colleagues at Bell Labs titled "Large Automatic Learning, Rule Extraction, and Generalization". This paper emphasizes the problem:

> *When a network is given more resources than the minimum needed to solve a given task, the symmetric, low-order, local solutions that humans seem to prefer are not the ones that the network chooses from the vast number of solutions available; indeed , the generalized delta method and similar learning procedures do not usually hold the "human " solutions stable against perturbations.*

One of the probable reasons why Deep Learning requires an inordinate amount of iterations and training data is because we seek Occam's Razor, that sparse solution. What if however, the solution to unsupervised learning (aka Predictive Learning) is in embracing randomness? One explanation for this is that adaptive systems work better with higher diversity.

Let's assume its validity for argument's sake that randomness is the natural equilibrium state. What this implies is that the model parameters will tend towards randomness and interpretability will be completely hopeless. Unless of course, we can ask the machine to explain itself!

Stephen Merity (MetaMind) has a detailed examination [TRN] of Google's Neural Machine Translator (GNMT) that is worth a read [MER]. The interesting thing about GNMT is that Google headlines this as "Zero-Shot Translation" [SCH]. Zero-shot capability here refers to the capability of GNMT to learn, for example, a Japanese to English translation, even if it was never trained with this particular translation pair! To quote Google:

> *This means the network must be encoding something about the semantics of the sentence rather than simply memorizing phrase-to-phrase*

> *translations. We interpret this as a sign of existence of an interlingua in the network.*

Will we perhaps be able to decipher this new "interlingua" or "Esperanto" that this machine created? Do we have a priori ideas as how this "interlingua" is supposed to look like and perhaps perform a kind of regularization to make it more interpretable for humans? Will the act of insisting on interpretability lead to a less capable translator?

It just seems that we should leave the representation as it is and use the machine to perform the translation into English. In fact, that is already what it currently does. We don't need some new kind of method to interpret the representation. The capability is already baked in there.

This is in fact what the folks at MIT, who have researched about "Making computers explain themselves" [CMP], have done:

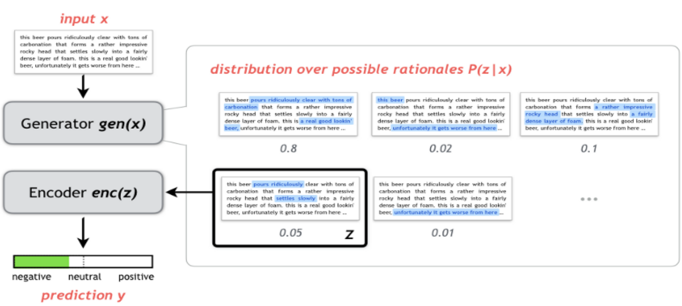


Figure 10.4 Source: MIT

They've trained their network to learn how to explain itself [ML16].

DARPA has already started its Explainable AI (XAI) project that is exploring machines with explainability. The goals of XAI are two fold. The first is to build learning machines that have explainable models. The second is to enable humans to effectively communicate with these "artificially intelligent partners."

Here's the concept art from DARPA:

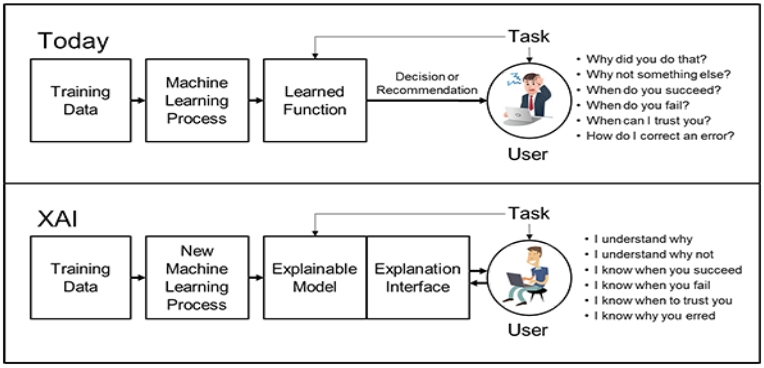


Figure 10.5 Explainable AI. Source: http://www.darpa.mil/program/explainable-artificial-intelligence

AlphaGo Zero's Self-Learning

The 1983 movie "War Games" has a memorable climax where the supercomputer known as WOPR (War Operation Plan Response) is asked to train on itself to discover the concept of an un-winnable game. The character played by Mathew Broderick asks, "Is there any way that it can play itself?" Thirty-four years later, DeepMind has shown how this is exactly done in real life! The solution is the same, set the number of players to zero (i.e. zero humans).

There is plenty to digest about DeepMind's latest breakthrough in Deep Learning technology. DeepMind authors use the term "self-play reinforcement learning". As I remarked in the earlier in the section about Tribes of AI, DeepMind is particularly fond of their Reinforcement Learning (RL) approach. DeepMind has taken the use of Deep Learning layers in combination with more classical RL approaches to an art form.

AlphaGo Zero (AGZ) is the latest incarnation of is Go-playing automation. One would think that it would be hard to top the AlphaGo version that bested the human world champion in Go. AGZ however not only beats the previous system, but also does it in a manner that validates a revolutionary approach. To be more specific, this is what AlphaGo has been able to accomplish:

1. Beat the previous version of AlphaGo (Final score: 100–0).

2. Learn to perform this task from scratch, without learning from previous human knowledge (i.e. recorded game play).

3. World champion level Go playing in just 3 days of training.

4. Do so with an order of magnitude less neural networks (4 TPUs vs 48 TPUs).

5. Do this with less training data (3.9 million games vs 30 millions games).

Each of the above bullet points is a newsworthy headline. The combination of each bullet point and what it reveals is completely overwhelming. This is my honest attempt to make sense of all of this.

The first bullet point for many will seem unremarkable. Perhaps it's because incremental improvements in technology have always been the norm. Perhaps one algorithm besting another algorithm 100 straight times intuitively doesn't have the same appeal of one human besting another human 100 straight times. Algorithms don't have the kind of inconsistency that we find in humans.

One would expect though that the game of Go would have a large enough search space that there would be a chance of a less capable algorithm to be lucky enough to beat a better own. Could it be that AGZ has learned new alien moves that its competitors are unable to reason about the same search space and thus having an insurmountable disadvantage? This apparently seems to be the case and is sort of alluded to by the fact that AGZ requires less compute resources to best its competitors. Clearly, it's doing a lot less work, but perhaps it is just working off a much richer language of Go strategy. Less work is what biological creatures aspire to do. Language compression is a means to arrive at less cognitive work.

The second bullet point challenges our current paradigm of supervised only machine learning. The original AlphaGo was bootstrapped using previously recorded tournament gameplay. This was then followed with self-play to improve its two internal neural networks (i.e. policy and value networks). In contrast, AGZ started from scratch with just the rules of Go programmed. It also required a single network rather than two. It is indeed surprising that it was able to bootstrap itself and then eventually learning more advanced human strategies as well as previously unknown strategies. Furthermore, the order in what strategies it learned first was sometimes unexpected. It is as if the system had learned a new internal language of how to play Go. It is also interesting to speculate as to the effect of a single integrated neural network versus two disjoint neural networks. Perhaps there are certain strategies that a disjoint network cannot learn.

Humans learn languages through metaphors and stories. A player refers to the human strategies discovered in Go with names so as to be recognizable. It could be possible that the human language of Go is inefficient in that it is unable to express more complex compound concepts. What AGZ seems to be able to do is perform its moves in a way that satisfies multiple objectives at the same time. So humans and perhaps earlier versions of AlphaGo were constrained to a relatively linear way of thinking, while AGZ was not encumbered with an inefficient language of strategy. It is also interesting that one may consider this a system that actually doesn't use the implicit bias that may reside in a language. David Silver, of DeepMind, has an even bolder claim:

> *It's more powerful than previous approaches because by not using human data, or human expertise in any fashion, we've removed the constraints of human knowledge and it is able to create knowledge itself.*

The Atlantic reports [AGO] about some interesting observation of the game play of this new system:

> *Expert players are also noticing AlphaGo's idiosyncrasies. Lockhart and others mention that it almost fights various battles simultaneously, adopting an approach that might seem a bit madcap to human players, who'd probably spend more energy focusing on smaller areas of the board at a time.*

The learned language is devoid of any historical baggage that it may have accumulated over the centuries of Go study.

The third bullet point says that training time is also surprisingly less than its previous incarnation. It is as if AlphaGo Zero learns how to improve its own learning.

It took only 3 days to get to a level that beats the best human player. Furthermore, it just keeps getting better even after it surpasses the best previous AlphaGo implementation. How is it capable of improving its learning continuously? This ability to incrementally learn and improve the same neural network is something we've seen in another architecture known as FeedbackNet. In the commonplace SGD based learning, the same network is fed data across multiple epochs.

Here however, each training set is entirely new and increasingly more challenging. It is also analogous to curriculum learning, however the curriculum is intrinsic in the algorithm. The train set is self generated and the calculation of the objective function is derived from the result of MCTS. The network learns by comparing itself not from external training data but from synthetic data that is generated from a previous version of a neural network.

The fourth bullet point, the paper reports that it took only 4 Google TPUs (180 teraops each) [TPU] as compared to 48 TPUs for previous systems. Even surprisingly, the Nature paper notes that this ran on a single system and did not require distributed computing. So anyone with four Volta based Nvidia GPUs has the horsepower to replicate these results. Performing a task with 1/10th the amount of compute resources should be a hint to anyone that something very fundamentally different is happening over here. I have yet to analyse this in detail, but perhaps the explanation is due to just a more simple architecture.

Finally, the last bullet point where it appears that AGZ advanced its capabilities using less training data. It appears that the synthetic data generated by self-play has more 'teachable moments' than data that's derived from human play. Usually, the way to improve a network is to generate more synthetic data. The usual practice is to augment data by doing all sorts of data manipulations (ex. cropping, translations, etc), however in AGZ's case, the automation seemed to be able to select richer training data.

Almost every new Deep Learning paper that is published (or found in Arxiv) tends to show at best a small percentage improvement over previous architectures. Almost every time, the newer implementation also requires more resources to achieve higher prediction accuracies. What AlphaGo has shown is unheard of, that is, it requires an order of magnitude less resources and a less complex design, while unequivocally besting all previous algorithms.

Many long time practitioners of reinforcement learning applied to games have commented that the actual design isn't even novel and has been formulated decades ago. Yet, the efficacy of this approach has finally been experimentally validated by the DeepMind team. In Deep Learning like in sports, you can't win on paper, you actually have to play the game to see who wins. In short, no matter a simple an idea may be, you just never know how well it will work unless the experiments are actually run.

There is nothing new about the policy iteration algorithm [POL] that is used or the architecture of the neural network. Policy iteration is an old algorithm that learns improving policies, by alternating between policy estimation and policy improvement. That is, between estimating the value function of the current policy and using the current value function to find a better policy.

The one neural network is described as a pedestrian convolution network:

> *The overall network depth, in the 20- or 40-block network, is 39 or 79 parameterized layers, respectively, for the residual tower, plus an additional 2 layers for the policy head and 3 layers for the value head.*

Like the previous incarnations of AlphaGo, Monte Carlo Tree Search (MCTS) is used to select the next move. AlphaGo Zero takes advantage of the calculations of the tree search as a way to evaluate and train the neural network. So basically, MCTS employing a previously trained neural network, performs a search for winning moves. The policy evaluation estimates the value function from many sampled trajectories. The results of this search is then used to drive the learning of the neural network. So after every game, a new and potentially improved network is selected for the next self-play game. DeepMind calls this "Self-play reinforcement learning":

> *A novel reinforcement learning algorithm. MCTS search is executed, guided by the neural network f_θ. The MCTS search outputs probabilities π of playing each move. These search probabilities usually select much stronger moves than the raw move probabilities p of the neural network $f_\theta(s)$; MCTS may therefore be viewed as a powerful policy improvement operator.*
>
> *Self-play with search—using the improved MCTS-based policy to select each move, then using the game winner z as a sample of the value—may be viewed as a powerful policy evaluation operator.*

With each iteration of self-play, the system learns to become a stronger player. I find it odd that the exploitive search mechanism is able to creatively discover new strategies while simultaneous using less training data. It is as if self-play is feeding back into itself and learning to learn better.

This self-play reminds me of an earlier writing about The Strange Loop. I wrote about many recent advances in Deep Learning such as Ladder networks and GANs that exploited a loop based to improve recognition and generation. It seems that when you have this same kind of mechanism that is able to perform internal assessments of its final outputs that the fidelity is much higher with less training data. In the case of AGZ, there's no training data to speak of. The training data is generated through self-play. A GAN for example, collaboratively improves its generation by having two networks (discriminator and generator) work with each other. AGZ, compares the capabilities of a network trained in a previous game against that of the current network. In both cases, you have two networks that feed of each other in training. AGZ appears also to be evolutionary. That is, you select the best version of the newly latest trained network and you discard the previous one.

An important question that should be in everyone's mind is: "How general is AGZ's algorithm?" DeepMind has publicly stated that they will be applying this technology to drug discovery [DRG], so the company has several ideas on how it can be applied elsewhere.

Earlier I wrote about how to assess the appropriateness of Deep Learning technologies (see: Reality Checklist). In that assessment, there are five uncertainties in any domain that needs to be addressed: Execution uncertainty, Observational uncertainty, Duration uncertainty, Action uncertainty, Evaluation uncertainty and Training uncertainty.

In the AGZ, the training uncertainty, seems to have been addressed. AGZ learns the best strategies by just playing against itself. That is, it is able to "imagine" situations and then discover through self-improvement the best strategies. It can do this efficiently because all the other uncertainties are known. That is, there is no indeterminism in the results of a sequence of actions. There is complete information. The effects of actions are predictable. There is a way to measure success. In short, the behaviour of the game of Go is predictable, real world systems however are not.

In many real world contexts however, we can still build accurate simulations or virtual worlds. These simulations can be used as an initial bootstrap to get to a situation of self-play (or self-learning). Certainly the policy iteration methods found here may seem to be applicable to these virtual worlds. Reinforcement learning has been applied to virtual worlds (i.e. video games and strategy games). DeepMind has not yet reported experiments of using policy iteration in Atari games. Most games of course don't need this sophisticated look ahead that requires MCTS, however there are some games like Montezuma's Revenge that does. DeepMind's Atari game experiments were like AGZ, in that there was no need for human data to teach a machine.

The difference between AGZ and the video game playing machines is that the decision-making at every state in the game is much more sophisticated. In fact there is an entire spectrum of decision-making required for different games. Is MCTS the most sophisticated algorithm that we will ever need?

There is also a question on strategies that require remembering once previous move. AGZ appears to only care about the current board state and does not have a bias on what it moved previously. A human sometimes may determine its own action based on its previous move.

It is a way of telegraphing actions to an opponent, but it usually is more like a head fake. Perhaps that's a strategy that only works on humans and not machines! In short, a machine cannot see motion if it was never trained to recognize its value.

Finally, there is a question about the applicability of a turn-based game to the real world. Interactions in the real world are more dynamic and continuous, furthermore the time of interaction is unbounded. Go games have a limited number of moves. Perhaps, it doesn't matter, after all, all interactions require two parties that act and react and predicting the future will always be boxed in time.

If I were to pinpoint the one pragmatic Deep Learning discovery in AGZ then it would be the fact that Policy Iteration works surprisingly well using Deep Learning networks. We've have hints in previous research that incremental learning was a capability that existed. However, DeepMind has shown unequivocally that incremental learning indeed works effectively well.

AlphaGo Zero appears also to have evolutionary aspects. That is, you select the best version of the newly latest trained network and you discard the previous one. There is indeed something going on here that is eluding a good explanation. The self-play is intrinsically competitive and the MCTS mechanism is an exploratory search mechanism. Without exploration, the system will eventually not be able to beat itself in play. To be effective, the system should be inclined to seek out novel strategies to avoid any stalemate. Like nature's own evolutionary process that abhors a vacuum, AGZ seems to discover unexplored areas and somehow take advantage of these finds.

One perspective to think about these systems as well as the human mind is in terms of the language that we use. Language is something that you layer more and more complex concepts on top of each other. In the case of AGZ, it learned a new language that doesn't have legacy baggage and it learned one that is so advanced that it is incomprehensible. Not necessarily mutually exclusive. As humans, we understand the world with concepts that originate from our embodiment with our world. That is we have evolved to understand visual-spatial, sequence, rhythm and motion. All our understanding is derived from these basic primitives. However, a machine may possibly discover a concept that may simply not be decomposable to these basic primitives.

Such irony, when DeepMind trained an AI without human bias, humans discovered they didn't understand it! This in another dimension of incomprehensibility. The concept of "incomprehensibility in the large" in that there is just too much information. Perhaps there is this other concept, that is "incomprehensibility in the small". That there are primitive concepts that we simply are incapable of understanding. Let this one percolate in your mind for a while. For indeed it is one that is fundamentally shocking and a majority will overlook what DeepMind may have actually uncovered!

Architectures of Collaboration

In the inevitable transition towards more modular multi-objective and multi-agent Deep Learning systems, we need to begin exploring the same loose coupling principles that underpin the coordination of distributed systems. A key criteria in building effective DL systems is better generalization. Although, generalization can mean many things, we can at a minimum accept an intuitive interpretation. That is, generalization implies systems of greater adaptability. In distributed architectures, Loose coupling principles encourage greater adaptability and therefore can provide valuable ideas on how best we can architect analogous DL multi-agent architectures.

Another justification is that intelligent systems are expected to contain a massive degree of diverse agents and therefore any mechanism that demands tight coupling is a mechanism that will not scale. Therefore, given a choice in selecting which mechanism to use, a loose coupling mechanism should be the preferred choice. If we think about this even deeper, all behavior should be based on the least amount of information, on only local information. This clearly sets it up to favor low information coupling. Any method that requires high information coupling to decide on an action is a method that is intuitively not the correct one.

A warning to the reader, this is highly speculative stuff and therefore should be either ignored entirely or treated with a grain of salt. Reading this can only lead to greater confusion. With that out of the way, let's review some loosely coupled principles that I dug up from a past life:

	Tight Coupling	Loose Coupling
Interface	Class and Methods	REST like (i.e. fixed verbs)
Messaging	Procedure Call	Document Passing
Typing	Static	Dynamic
Synchronization	Synchronous	Asynchronous
References	Named	Queried
Ontology (Interpretation)	By Prior Agreement	Self Describing (On The Fly)
Schema	Grammar Based	Pattern Based
Communication	Point to Point	Multicast
Interaction	Direct	Brokered
Evaluation (Sequencing)	Eager	Lazy
Motivation	Correctness, Efficiency	Adaptability, Interoperability
Behavior	Planned	Reactive
Coordination	Central Command Driven	Market Driven
Contracts	By Prior Agreements, Implicit	Self Describing, Explicit
Transactions	Pessimistic	Optimistic
Classification	Classes	Prototypes

Mapping these principles to may not be applicable to DL multi-agent systems. However it can be educational to explore each one and potentially propose an equivalent viable approach. As a caveat, we are constructing here a hypothetical system based on a future idea of a network of collaborative and competitive agents tasked with solving a problem using imperfect knowledge (see: 5 Capability Levels of Deep Learning). These are merely some preliminary ideas that we may want to bake into a hypothetical multi-agent system.

Fixed Verb Interfaces

Ideally we would like to have as few interfaces as possible to reduce incompatibility as well as increase plug and play. Think of how effective the USB standard has been for power and communication convenience. One other thought is to build these kinds of systems employing FIPA communicative acts or speech acts. The present approach for DL systems is to learn how to communicate on their own. Perhaps however by adding constraints on the nature of communication, inspired by speech acts, may lead better reusability. So for example, a neural network may be trained against another system with speech acts as the protocol, conceivably this kind of system may be adaptable in another context where speech acts are also used as the protocol.

Document Passing based Messaging

Communication is likely to be fire and forget document message passing. It's just an easier thing to learn. Presently, most research on learning to communicate tends to employ message passing rather than a coordinate request/response or a procedural like invocation.

Dynamic Typing

One would think that there's no notion of types for representations in DL system. However, as we have seen in Google's NMT system, the addition of a label that indicates a representation's language was one of important tricks to achieve one-shot cross translations. So there is evidence that tagging data with its type may be valuable even for DL systems.

Asynchronous Synchronization

Monolithic DL system actually have a rigid form of synchronization with respect to back and forward propagation through its layers. In a more modular rendition, we would like to relax these restrictions such that synchronization between components are not required. Besides, biological brains don't require a single clock like computer systems. So one should not have this same synchronized requirement for DL systems.

Queried References

In DL system,there are Pointer Networks that maintain hard coded references similar to memory pointers. A more modular system would require a second level of indirection such that the reference is to a query and not necessarily to some internal representation. This is analogous to associative or context based retrieval. In other words, if there is a retrieval to be made then the request for that information is through a query and not through a opaque identifier (as we see in conventional computers).

Self Describing Ontology

Representations for DL system are definitely not self-describing, they are in fact opaque. This of course makes it next to impossible to coordinate between multiple interacting agents if there is no mechanism for sharing knowledge. Furthermore, there does not exist research that tries to learn a "meta-level" representation of data. This is going to be the major technical hurdle for multi-agent based systems. A lesser form of this problem is the Fixed Verb approach. One development to note however is that in the DeepMind's PathNet approach, lower level representation sometimes shared between networks.

Pattern Based Schemas

A pattern based schema is one that is more loosely defined than one that is grammar based. The idea here is that only partial specification is necessary for interoperability. There is a concept in DL that nuisance variables are ignored in training or the notion of invariant features in data. One would like to train to ignore data that is invariant while focusing on features that are distinct.

Multicast Communication

Ideally, one shouldn't have to care how subnetworks are wired together. The exists research where networks are wired in a hierarchical manner (see: Maluuba and DeepMind). Where there are networks that are coordinators for much simpler networks. Multicast networks assumes that participants are able to reason on their own as to what messages are worth listening to. This is a big burden to justify.

Brokered Interaction

Ideally, in an adaptive system, no two components are hard coded for interaction and that dynamically that interaction can change depending on context. We've seen an architecture like this in "Conditional Logic in Deep Learning", where the selection of subnetworks are controlled also by a layer. Also in CPPN based systems, there are brokering components that are neural networks that's sole purpose are to learn how to adapt one layer to another layer.

Lazy Evaluation

Alternatively, we can think of this as late-binding. That is, if we can defer commitment until action is necessary. The value of this capability related more to planning execution. That is, there is an additional dimension to plan execution such that work is not performed only when dependencies are available. It is difficult to see how this applies to DL other that the fact that DL systems are naturally data flow based system and thus lazy evaluation is a foundational implementation feature. What is interesting however is the application of the late-binding principle to recognition. One of the glaring deficiencies of DL is its lack of adaptability and perhaps some kind of learning on the fly, a lazy learning process, may be a solution to this problem.

Adaptability, Interoperability

These are just general principles that guide a loosely coupled system and should also be applicable to a loosely coupled DL system.

Reactive Behavior

There is research in Learning to optimize or Learning to plan, where a DL system is able to propose execution plans. RL system by their nature learn reactive behavior and long range planning is certainly more difficult to implement. The hunch here is that DL systems as a consequence of their poorer logic inferencing capabilities are going to perform better as reactive agents as opposed to planning agents.

Market Driven Coordination

Market driven distributed coordination is likely going to be the mechanism for multi-agent coordination given that complexity of learning how to perform central command driven execution. The distribution of responsibility is one that seems to be more scalable and realistic.

Self Describing, Explicit Contracts

Contracts are necessary in a market driven economy to ensure that participants are compliant in their behavior. This implies then that contracts themselves are representations that provided guidance for execution as well as compensation in cases of failure. DL system that learn how to perform compensation based on failure is certainly going to be an interesting research topic.

Optimistic Transaction

We have to assume failures will exist in a marketplace (or in coordination) and therefore participants need the additional capability of performing compensating actions. It will indeed be interesting if we create systems that can learn this behavior.

Prototype Based Classification

The notion that there is a strict separation between instance and meta-level data is an artificial construct and sometimes may be too restrictive. Today, it's unclear yet how representations can be built to represent a class of a concept. One can however imagine the storing of exemplar instances or prototypes and using this as a way to kickstart classification.

In a more abstract classification, loose coupling techniques have three recurrent characteristics. These are, late binding, mediation and decomposition. Interestingly enough, there are research papers covering all three characteristics, the most intriguing one of all how late binding is applied in the context of connectionist architectures.

One thing that seems to recur often enough that requires one attention is the notion of meta-level representation. Meta-level representations exist due to the need for coordination. It is a capability that is akin to having a system with internal self-awareness. This self-awareness of course seems like a problematic requirement. One should expect to build intelligent systems without the need for intelligent subcomponents (see: Artificial Intuition). However, the meta-level reasoning components themselves may also be unintelligent and may be unaware that its operation is at a meta-level.

Another observation here is that its hard to divorce oneself from a symbolicist approach once we begin discussing meta-level concepts. It is therefore quite conceivable that an embodiment of the above system would be a hybrid symbolicist/ connectionist architecture. Alternatively, just like the mind, something that works of dual process theory. That is, a intuition and a rational machines all working in concert.

What I did here is examine ideas from distributed computing and to see if the basic idea of loose coupling or low information coupling can serve as inspiration on how to build multi-agent DL systems. One main conceptual stumbling block is the notion of meta-level information that exists in distributed systems. However, there is indeed still a lot of common ideas that clearly should be leveraged. This exploration in promising in two aspects, the first is in identifying fundamental principles and the second in identifying constraints on how multi-agent systems communicate. The identifying of constraints or alternatively called the definition of a language of discourse, provides a bounded space for exploration and therefore may make feasible the possibility of learning to coordinate.

Machine Self-Awareness

We are all aware that consciousness exists, yet we don't have an adequate explanation for its emergence. I've written down previously about what I see as five capability levels of Deep Learning:

1. **Classification Only (C)**

2. Classification with Memory (CM)

3. Classification with Knowledge (CK)

4. Classification with Imperfect Knowledge (CIK)

5. Collaborative Classification with Imperfect Knowledge (CCIK)

I purposely left out a the penultimate level, that of achieving self-awareness and consciousness. I did so because it is such a wide chasm that will eventually need to be crossed, yet we know so little about. I would like to think that my investigations are evidenced based rather than ones based on thought experiments.

There are however some theories out there that I think it is worth studying. Not because there is any solid evidence, but rather because the thinking behind them seems to make sense. We have no verification if any of these theories are true, but we can still use the attributes that are predicted. Possibly employing them as that as a guide in our Deep Learning research.

Before we begin exploring consciousness, I would to point out the kind of architectures that we are using as a starting point. It is my conjecture that to achieve artificial consciousness, you have to have as your base an architecture of independent DL agents that are interacting in a decoupled manner. We discussed this briefly in my post "The End of Monolithic Deep Learning" as well as an approach for coordinating agents in "Market Driven Coordination" call this Modular Deep Learning.

I would like to discuss two theories that I find promising. The first is Integrated Information Theory (IIT) by Giulio Tononi and the second is the theories from Jurgen Schmidhuber.

"From the Phenomenology to the Mechanisms of Consciousness: Integrated Information Theory 3.0" or IIT 3.0 describes the theory (Here is a TED talk about this):

IIT begins with several axioms, translates these axioms into postulates that are conditions that must be satisfied to achieve consciousness. The postulates are as follows:

EXISTENCE: Mechanisms in a state exist. A system is a set of mechanisms.

COMPOSITION: Elementary mechanisms can be combined into higher order ones.

and corresponding mechanisms:

INFORMATION: A mechanism can contribute to consciousness only if it specifies "differences that make a difference" within a system. That is, a mechanism in a state generates information only if it constrains the states of a system that can be its possible causes and effects—its cause-effect repertoire.

INTEGRATION: A mechanism can contribute to consciousness only if it specifies a cause-effect repertoire (information) that is irreducible to independent components.

EXCLUSION: A mechanism can contribute to consciousness at most one cause-effect repertoire.

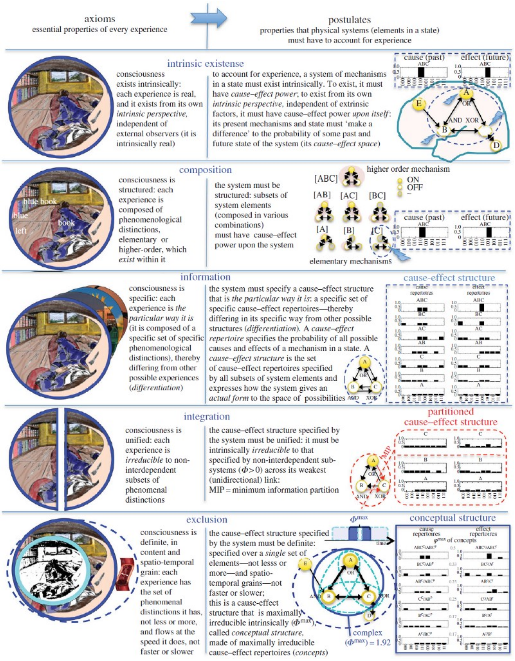
The theory expands also to a set of agent mechanisms "Systems of mechanisms" :

INFORMATION: A set of elements can be conscious only if its mechanisms specify a set of "differences that make a difference" to the set—i.e. a conceptual structure.

INTEGRATION: A set of elements can be conscious only if its mechanisms specify a conceptual structure that is irreducible to non-interdependent components (strong integration).

EXCLUSION: Of all overlapping sets of elements, only one set can be conscious—the one whose mechanisms specify a conceptual structure that is maximally irreducible (MICS) to independent components.

The following graphic captures these axioms and postulates in even greater detail:



Source:
https://en.wikipedia.org/wiki/Integrated_information_theory

IIT proposes that consciousness is a matter of degree and proposes a measure of consciousness. In other words, many systems are already conscious, but with varying degrees of consciousness. The theory is quite elaborate, the key take away though is the emphasis on information structure that captures causality and that the richness of that causality structure indicates a measure of consciousness. Note that causality, cause and effect, Bayes rule are all related to mutual information.

Schmidhuber is at interesting character because he is pretty sure that the nature of consciousness has been solved. His theory combines elements that are more familiar to DL practioners, however he discussion consciousness with the context of what he labels as Gödel machines." His claim is that AI gained consciousness way back in 1991:

I would like to claim we had little, rudimentary, conscious learning systems for at least 25 years. Back then, already, I proposed rather general learning systems consisting of two modules.

One of them, a recurrent network controller, learns to translate incoming data—such as video and pain signals from the pain sensors, and hunger information from the hunger sensors—into actions.

Since 1990, our agents have tried to do the same thing, using an additional recurrent network—an unsupervised module, which essentially tries to predict what is going to happen. It looks at all the actions ever executed, and all the observations coming in, and uses that experience to learn to predict the next thing given the history so far. Because it's a recurrent network, it can learn to predict the future—to a certain extent—in the form of regularities, with something called predictive coding.

As the data's coming in through the interaction with the environment, this unsupervised model network learns to discover new regularities, or symmetries, or repetitions, over time. It can learn to encode the data with fewer computational resources—fewer storage cells, or less time to compute the whole thing. What used to be conscious during learning becomes automated and subconscious over time.

One important thing about consciousness is that the agent, as it is interacting with the world, will notice that there is one thing that is always present as it is interacting with the world—which is the agent itself.

I'm pretty convinced that all the basic ingredients to understand consciousness are there, and have been there for a quarterntury. It's just that people in neuroscience who maybe don't know so much about what is going on in artificial neural network research, they are not yet so aware of these simple basic principles.

Jurgen Schmidhuber's conjecture is that two recurrent networks, one responsible for actions and a second one responsible for predicting the world are the basic ingredients to achieving consciousness. That is CCIK in my classification scheme leads to consciousness. It is indeed extremely intriguing, in fact the second controller is actually performing a kind of meta-learning (see: "Meta Meta-Model"). Here is Schmidhuber talking about this:

What is striking about both models of consciousness is that they have similar claims. That is, consciousness already exists in simple mechanisms and that human level consciousness is just a matter of degree. Both theories claim that there is no need for a new kind of mechanism to achieve consciousness. The conceptual missing link to explain consciousness is already known.

An additional similarity is that there is mechanism that handles temporal causality. IIT revolves around being able to create internal models that capture the causality between concepts. Schmidhuber's approach employ RNNs that are able to recognize patterns in time.

I am uncertain if the mechanism proposed in IIT are in advanced modular DL systems, we are in the early stages of discovering this. On the other hand, Schmidhuber is saying we don't need to look far. Schmidhuber in the most recent NIPS conference argued that GANs ("The coolest thing in the last 20 years") were identical to his paper in 1992, "Learning Factorial Codes by Predicability Minimization". This paper describes a system that involves "two opposing forces". I can't tell if this paper is about the same thing that he claims has achieved consciousness. I don't know if Schmidhuber is claiming that GANs are conscious!

What bothers me about all this is that we can be very near to Artificial General Intelligence (AGI) if these two theories are correct. Alternatively, we can be very far away, not knowing what that missing link may be. In both alternatives, one can't predict when we arrive at AGI and that is very disconcerting.

Summary

Language serves as the foundation of any advance technology. From creating simple tools for farming, scaling our civilizations, to understanding quantum phenomena and achieving more intelligent machines. This chapter explains these ideas in more detail and explores how a language centric approach points to a very compelling strategy for building even more advanced intelligent systems.

11 Human Compatible AI

"With artificial intelligence we are summoning the demon. In all those stories where there's the guy with the pentagram and the holy water, it's like yeah he's sure he can control the demon."

- Elon Musk

This final chapter contains a collection of ideas that relate to the impact of AI to society. Some of these ideas revolve around issues that we need to find a resolution in the near future. However, a majority of the ideas will require resolution on when AI technology makes significant progress. The rate of progress is indeed exponential, however it still is difficult to assess when Artificial General Intelligence (AGI) will make its debut.

When AGI arrives is an unknown known. That is, we know from our own human capabilities that general intelligence is known. However, we do not know whether it is possible to achieve these in machines that we create. We also do not know when machines will be gaining this capability. We however do know, that when AGI arrives that we have better have the safeguards in place so that the situation does not devolve into a Terminator-like scenario.

This chapter does not propose solutions, but rather ideas that one should think about. It covers the future of work, understanding decisions and responsibilities in the context of automated AI, the fallacy of efficiency in our economy, and an exploration of AI safety in the context of Artificial Intuition.

Although this is not a very practical chapter, I think as a civilization, humanity has to begin a serious conversation on the impact AI will have on our future.

Jobs that are Safe

Thomas Frey has a thought provoking article "78 Skills that are Difficult to Automate". Frey breaks down the categories of jobs that he believes will remain "safe" from automation:

- Complex systems too expensive to automate
- Creative endeavors that only humans can appreciate
- Human to human interactions that produce an emotional response
- Decisions that need human-based reasoning
- Complicated outputs that demand a human translator
- Situations that require the human touch
- Settings where the loyalty of hacker-proof humans is preferable over digital machines
- Human to human valuations
- Positions where humans control robots [FRE]

I will attempt to address each one from the perspective of the "Deep Learning Canvas" that we've developed. In our approach, we fuse together the ideas of "Jobs to be Done" (JTBD) approach for identifying tasks that need to be addressed and an understanding of the cognitive limitations that can be enhanced through Deep Learning.

In Frey's article, he builds up his argument that the jobs most likely to be safe are those that take into account the irrationality of humans. The JTBD approach also goes beyond pure functionality, but also addresses human needs such as emotion and social currency. Whatever the Deep Learning system will be automating, it will need to target any combination of these three needs. Now, we don't use a blunt instrument and try to replace tasks wholesale. Rather, we do so in a surgical manner, identifying first the capabilities required to address the customer's' needs and then identifying which specific cognitive tasks can be enhanced through automation.

What people seems to miss is that the replacement of jobs is performed piecemeal and incrementally. Many prognosticators miss this kind of detail. The capabilities of those that remain employed become more powerful. The lesser skilled folk who are first to lose their jobs become less adept at using automation tools. It is a vicious cycle where those that have the skills gain more advanced skills. While those that don't have the skills are forced to pay premium to gain the skills. In many cases, there aren't any educational institutions that exist to teach them these skills. Furthermore, fewer skills imply more commoditization. People will be forced into the 'gig economy', relegated to endlessly competing in markets with margins pushed towards zero.

However, let's examine Frey's list because it is an informative base to perform more detailed analysis.

Complex systems too expensive to automate

Complex systems will always require people with advanced skills to orchestrate. However, that does not imply that there will be less automation that will be enabling this activity. In fact, one should expect bleeding edge companies to leverage as much automation as needed to accelerate work. Tesla and SpaceX are likely highly automated companies as compared to their competitors.

Companies that find it too expensive to automate processes are likely the ones that don't know how to exploit automation to reduce development costs.

Creative endeavors that only humans can appreciate

This is similar to the previous point but relates to more artistic endeavors. In a lot of the Hollywood Science Fiction blockbusters, we continue to use increasingly advanced automation to reduce costs. Filmmakers no longer need to hire armies of extras to film massive battle scenes. These are now all done through CGI simulation. There was a period in time were Epics were too expensive to make, but today that's no longer a problem.

Certainly we'll have the creative folks continue to drive development, but the human resources required will continue to diminish. In the future, we may not even need actors for films and can instead use CGI renditions of famous actors. In fact, if we ever get bored with seeing the same faces, we can always generate arbitrary faces!

Human to human interactions that produce an emotional response

Most of what Frey describes, (i.e. a smile, a hug, a kiss, a massage etc.) usually aren't paid for. In fact, in many societies, paying for these "human interactions" would be considered illegal!

Decisions that need human-based reasoning

The human serves as a watchdog against runaway automation. I think this kind of job will not go away if governments enact regulation that requires this. It's just like humans who serve as gas attendants. Certain states require it, although it isn't absolutely necessary. However, this kind of legislation can keep a lot of people on the job.

Complicated outputs that demand a human translator

Examples: Doctors, Data analysts, judges, business executives, privacy advocates, relationship building strategies, birthing processes, genealogical mapping.

This is not very different from the previous class or even the first class. It is just supervision at the top rather than at the bottom. It however pertains to the need for humans to interpret machines so that other humans can understand (and accept) a machine's conclusions.

However, this likely won't change because legislation is already in place that humans are required for these kinds of job. We can't have non-human judges sending people off to lifetime incarceration or death row.

Situations that require the human touch

Examples: Teaching

The examples Frey provides in this category all seem to revolve around education. However, I would argue that education is moving massively online, and that education will only get better with Virtual Reality, Augmented Reality and AI. That's because we can build many more interactive environments that can serve many more students at a quality level that is many times better than the average teacher.

The problem with our current education practices is that the lecture model is all passive listening. We need to strive for greater participation of the student in the model of Active Learning. [ACT] People learn best by doing and not by lecture. The expertise to create highly engaging education will be in demand. This will require a deep understanding of the interactive technology, an understanding of human behavior and teaching methods. Where Deep Learning comes into play is its ability to react to human behavior. This is essential to effective teaching and is not outside the realm of what's possible.

Settings where the loyalty of hacker-proof humans is preferable over digital machines

Examples: Guarding VIPs, holding secrets, personal confidants, safeguarding corporate knowledge, consultants, lobbyists, leaders of robot resistance groups

On the contrary, Blockchain systems have been shown to be hacker-proof as opposed to systems that have human elements that can be "socially engineered".

Human to human valuations

Examples: Stock market, voting, government policies, consequences on policy violations, buying, purchasing agents, rating agencies, surveys and poll.

We already have systems that make all sorts of 'valuations' on what we should focus our attention on. Facebook manages our reading list. Amazon recommends products that we might prefer. Google filters our search results. More and more, AI is making our decisions for us. We've become so used to machines like GPS giving us instructions that we've lost all sense of direction.

What we will likely see in this domain is that Blockchain technologies will ensure transparency and integrity of many of the interactions that assume a fair market or a democratic process. Today, many of these processes are gamed to the benefit of the very few. Human to human valuations should not be entrusted exclusively to humans, but rather it should be done through a transparent, collective, manner. The reason here is because humans have motivations to game the system.

In an AI economy, it will become harder and harder to game one's credentials. Machine intelligence will become sophisticated enough to expose many in the workplace as imposters. Let's all be perfectly honest, the reason why people get paid more than they deserve is due to a gross inefficiency in how humans assess an employee's worth. 20% of employees do most of the work, while 80% are all pretenders.

Positions where humans control robots

Examples: owner and managers, software developers, system engineers, product designer, robot maintainers, auctioneers that sell robots.

This is the same theme as 3 of the previous themes. That is, automation as a tool to enhance human work.

In summary, the list can be simplified even further:

1. Jobs that use automation as a tool.

2. Jobs that that use humans as safety valves against automation failure.

3. Jobs that interpret the decisions of machines.

4. Jobs that design human-machine interfaces.

5. Jobs that design automation to manipulate human behavior.

That's five classes of jobs that will exist in the future that appear to be safe. On the other hand, with the exception of the "human safety valve", all these other jobs require high-level skills. Jobs of the future need to have a deep understanding of humans as well as machines, and it is in this interaction of man with machine where jobs will exist.

I think what few seem to appreciate is that Deep Learning AI is technology that is like human intuition. It is an opposite technology from more classical AI technologies that focused on reasoning. At this time there remains a Semantic Gap [SMT]. However, humans' capabilities are stuck between a rock and a hard place. Between Artificial Intuition [INT] and Artificial Reasoning. This is where many people seem to be getting it all wrong about what jobs are safe and what is not. Let us not fall into this fantasy that our unique human intuition is safe from being replaced by automation.

There aren't many jobs that are safe with the emergence of Deep Learning. We have to come to grips with this reality so that we can get a head start in examining the core of our economic system. AI will likely break capitalism, and we unfortunately are not starting serious discussions on what will replace it. The stark reality is that the emerging AI economy is reserved for the highly skilled.

Temporal Impedance Mismatch

The phrase 'impedance mismatch" originates from electronics, but is borrowed in software engineering to mean difficulties encountered when there is a conceptual difference between a sender and recipient of information.

There is however another kind of "impendence mismatch" in the form of the speed of delivery of information and the ability of a recipient to digest the information. A situation that illustrates this well can be found in the introduction of any new software product. Take for example a new operating system that has a new kind of user interface. To innovate in user interface design, a team of designers and developers can consume several years working on a new design. In doing so, they perhaps build new kinds of metaphors and new ways of interacting. Over time, for the originators, the design becomes familiar and eventually becomes intuitive.

However, when this new software hits a new user for the first time, the quantity of new ideas that a user has to digest can be enormous. That is because years of designing new concepts for interaction have been compressed and presented to a new user to absorb immediately. Therefore, it is best practice for UI designers to gradually introduce new ways of interaction in an incremental way. This is known as "progressive disclosure". It is a design technique that presents only the minimum data required for the task at hand. This improves usability by reducing distraction and cognitive workload.

Progressive disclosure is extremely important for automated systems. This is simply due to the reality that humans are much slower in our ability to digest new information. Automation should be designed to disclose information at a rate that a user can effectively consume. It is ineffective to build automation that just dumps new information and expect a user to drink from a fire hose of information.

It is instructive to understand the spectrum of information that automation can discover from the temporal perspective. These include:

- Information that was discovered in the past
- Information happening now
- Information that you have requested in the past to be notified for in the future
- Suggestions for immediate actions
- Predictions for future events
- Recommendations for preventive actions for mitigating future risks

An AI automation has to consider that all this information be delivered not only in a timely manner, but also in a manner that can be effectively digested and acted upon by humans. We shall explore in the next section the more complex scenario where not one but multiple people are responsible for making a decision.

Decisions and Responsibility

Decision Rights

With greater automation comes a greater need for safeguards. When automation works at a speed that is much faster than humans to make decisions, then this temporal impedance mismatch is going to be problematic. That is why the design of human-machine interfaces are critical so as not to have humans become a bottleneck or to avoid automation that can lead to rapid cascading failure.

A recommended practice is to become more formal in how decision-making is distributed and acted on within an organization. Machines will tirelessly ingest, analyze and act on new information. However, there exist situations where a human performs a final decision or that humans have options to override automated decision-making. This is an important balance that has to be made so as to maximize the effectiveness of automation. Too much bureaucracy can easily bog down automation, just like humans.

One such framework is the RACI assignment matrix, which involves the following [RACI]:

Responsible – Who is completing the task?

Accountable – Who is making decisions and taking actions on the task(s)?

Consulted – Who will be communicated with regarding decisions and tasks?

Informed – Who will be updated on decisions and actions during the project? [DOG].

The "who" may not necessarily be a person, but rather a machine. The question that we must ask ourselves, and the business processes that we participate in, is "in what contexts do we wash our hands of all responsibility and accountability?"

Let's examine the case of United Airlines and its brutal 're-accommodation' of a paying passenger. Clearly this was a stark example of organizational failure. The situation could have been easily resolved by increasing the payoff amount to incite more volunteers. Yet, the unexplainable happened.

The root cause of course of this failure of common sense is obvious. Employees followed rules by the letter and abrogated all responsibility and accountability. "This is above my pay-grade" is a common way to accept stupidity.

However, as processes become more and more automated; as decision-making is performed by finely tuned algorithms; as we collectively avoid responsibility in our response to avoiding liability, as we lose all empathy by hiding behind the emotionless interfaces of our automation, are we thus inadvertently moving towards a world of apathy and unchecked cruelty? After all, was not the United Airlines paying customer battered senselessly, only to later receive an apology for the inconvenience of being 're-accommodated'?

In general we would all like to believe that we are all caring people. However, that does not imply that we can just as easily be induced to performing barbaric behavior or perform acts that are not of our best interest.

A.I. Regulation

Let's discuss now the issue of AI regulation. Let's look at the continuum of automation, understanding here that AI is a merely a more capable form of automation and then let's explore existing regulation in other filed that applies to the use of automation.

There is a wide continuum of automation, that we've discussed in a previous chapter. Here is a recap:

Level 0 (Manual Process)

Level 1 (Attended Process)

Level 2 (Attended Multiple Processes)

Level 3 (Unattended Process)

Level 4 (Intelligent Process

Level 5 (Fully Automated Process)

Level 6 (Self Optimizing Process

Let's now examine various laws in different domains and relate them to the levels prescribed above. I think it will safe to assume that laws that apply to a lower level also apply to every higher level.

Here are some laws that are in currently in existence:

Robocalling—Enacted by the FTC in 2009. Prohibits prerecorded telemarketing calls, unless the marketer has the consumer's prior written authorization to make a call. Further FCC regulations enacted in 2016 [RBO] on Robocalls for political campaigns. This is either a level 1 or level 2 automation.

Spam—SPAM Act of 2003 [SPM] basically says "e-mails should not mislead recipients over the source or content of them, and that all recipients of such emails have a right to decline them." This is level 2 automation.

Viruses , Trojan Horses and Worms—1990 Computer Misuse Act [MIS] which covers unauthorized access and "unauthorised modification of computer material". This is level 3 automation.

Programmed Trading—October 19, 1987 also known as "Black Monday". New rules required exchanges to have "trading curbs" or "circuit breakers" that allow exchanges to halt trading in instances of high volatility. This is level 3 automation.

High Frequency Trading—CFTC is proposing regulations [REG] with regards to automated trading such as AT tactics such as "spoofing," "flash trading," and "quote stuffing". HFT involves leveraging computers to exploit market inefficiencies that arise from delay and participant response times. In general, financial organizations make a living by hacking our financial system to find areas of inefficiency and loopholes where they can legally rob market participants. This can be level 2 or level 3 automation.

Drone Regulation—FAA Regulations [DRO] that are now in affect. Specific regulations that are general enough to apply to other automation: "Drones have to remain in visual line of sight of the pilot". Although drones are more capable of higher levels of automation, the law restricts them to level 2.

Regulation of Genetic Engineering—Genetic engineering is limited on animals to a few use cases [GEN]. Mostly legal for experiments and the development of derivative products, however it is illegal to let these genetic engineered animals into the wild! This is level 6 automation.

Biological Weapons—Act of 1989 [BIO]. The act makes it illegal to buy, sell or manufacture biological agents for use as a weapon. This is level 6 automation. Note that level 2 weaponized automation are already used in theater in the occasional "drone strike" in the middle east. Also note that cruise missiles are level 4 automated weapons.

Nuclear Non-Proliferation Treaty—Two important aspects that may be relevant with AI, "not in any way to assist, encourage, or induce" a non-nuclear weapon state to acquire nuclear weapons (Article I)" and "right of all Parties to develop nuclear energy for peaceful purposes and to benefit from international cooperation in this area (Article IV)".

This is just a short survey of the laws that exist that involve the regulation of either automation or dangerous technologies. What can we thus now generalize about these existing laws?

1. Automation requires permission to interact with humans.

2. Automation cannot mislead humans as to its identity.

3. Automation shall not make unauthorized modification of information.

4. Automation shall be automatically shut down in anomalous

situations.

5. Automation shall not have restrictions on the methods it uses to deceive other participants.

6. Automation shall always be attended by a human.

7. Level 6 automation shall never be let out into the wild. Level 6 automation shall be only available for experimentation and creation of non-level 6 automation.

8. It should be illegal to buy, sell or manufacture weaponized level 6 automation.

9. Whoever gets to level 6 automation first decides for everyone else what the rules are. Otherwise known as the "Golden Rule for AI", that is, who owns the Gold, therefore rules! Vladimir Putin in a recent broadcast to the students of Russia [PUT] conveyed this sentiment. Where he says "It would be strongly undesirable if someone wins a monopolist position."

Seven Deadly Sins

It occurred to me one day that the seven deadly sins, that is:

"Anger (Ephesus), Gluttony (Smyrna), Pride (Pergamos), Lust (Thyatira), Slothfulness (Sardis), Envy (Philadelphia), and Greed (Laodiceans).

Originates mostly from our unconscious, most specifically our intuition.

The consensus understanding is that these behaviors originate from instinctual or biological sources. However, I wonder that if it originates at a higher level, that is, our intuition. The distinction between instinct and intuition is that the former is hardwired while the latter is learned from experience. I will make the additional observation that our personalities are mostly setup by our hardwired instincts and reinforced through experience by our intuition.

In an earlier post, I wrote about intuition in some detail. Dual Process Theory theorizes that there are two kinds of cognition. System 1 is our intuition. Daniel Kahneman [KAHN] in his book "Thinking Fast and Slow" argues that our cognitive biases are what works against us to lead us to many irrational decisions [KAH]. The question I have is whether the seven deadly sins also come from our cognitive biases and therefore our intuition? Well, just looking at the above table, it probably makes sense since a lot of impulsive behavior seems fit under intuition. So let's start, consult the "Cognitive Bias Cheat Sheet" [BEN] if you get lost.

Okay, let's enumerate the sins and propose the cognitive bias that these sin may plausibly originate from:

Wrath— Triggered by a lack of meaning. We think we know what others are thinking.

Gluttony—Need to act fast. In order to stay focused, we favor the immediate, relatable thing in front of us over the delayed and distant.

Pride—Too much Information. We notice flaws in others than in ourselves. Naive realism.

Lust- Not a bias, but rather driven by instinct.

Sloth—Too much Information. We are drawn to confirm existing beliefs. Confirmation bias.

Envy-Need to act fast. In order to act, we need to be confident in our ability to make an impact and to feel like what we do is important.

Greed- Need to act fast. In order to get anything done, we're motivated to complete things that we've already invested time and energy in.

The correlation looks quite accurate. The deadly sins are extreme behaviors that likely start off as more moderate biases, but through a combination of other factors can lead to dangerous behavior.

Now, let's go to another question that in fact could also be related. Can Artificial Intelligence be motivated by these deadly sins?

One argument out there that A.I. will not be as destructive as humans is that A.I. are not biological and therefore do not have the same primitive instincts that lead us to destructive and deplorable behavior like racism and genocide.

Google has a paper "Bringing Precision to the AI Safety Discussion" [PRE] where they discuss five future (not present) requirements for AI systems to ensure our safety. I've taken the liberty to re-phrase these as commandments:

1. AI Shalt Not Negatively Disturb the Environment

2. AI Shalt Not Game the Reward Function

3. AI Shalt Not Annoy Its Master

4. AI Shalt Not Do Unsafe Things

5. AI Shalt Not Act Reckless Outside School [OLA]

This list is not very reassuring.

So here's the question, can motivations driven by seven deadly sins be acceptable under these commandments? The problem with these five Google commandments is that the only real firewall is number 2. That is, "Thou shalt not game the reward function". Unfortunately, we know from human history that it is next to impossible to set up laws that aren't going to be gamed by enterprising participants.

Perhaps we need another set of AI commandments based on the Seven Virtues [VIR]:

Temperance—AI shall practice self-control.

Charity- AI shall be generous and make sacrifices on behalf of humans.

Diligence- AI shall not complain about work.

Patience—AI shall be patient with humans.

Kindness—AI shall have empathy towards humans.

Humility- AI shall revere humans.

Chastity—AI shall not procreate.

Now, that's definitely more reassuring.

Which is just how Asimov derived the three laws of robotics:

> *"When Isaac Asimov wrote his three laws of robotics, they were lifted straight from the marriage vows: love, honor, and obey"—Sadie Plant*

So why is this so much better than the commandment, "thou shalt not game the system"? Ideally, you want to be able to control behavior from the bottom, so that it is instinctual and not something that can be 'gamed'. Mechanisms from the top down are problematic because they can be easily rigged. We see an analogy in government laws that citizenry always seem to have a loophole to exploit. So, as a path towards implementation, one has to explore how to hardwire virtues into an AI. I think this is the only way forward, and top-down approaches of defining rules will be impossible to make airtight.

So, as a path towards implementation, one has to explore how to hardwire virtues into an AI. I think this is the only way forward and top-down approaches of defining rules will be impossible to make airtight. Let me propose that we hardwire all of these virtues in a Blockchain. Said in a different way, hardwire laws and behavior through network consensus.

Network consensus is in fact how humans enforce morality. Yuval Noah Harari classifies reality into three types:

- *Objective reality*, which are true independent of our believing in it.
- *Subjective reality*, truth from the perspective of a individual mind of a believer.

- *Intersubjective reality*, truth that exists in the mind of many believers.

This is an instructive classification that helps us target where AI laws may be potentially codified. Civilization has used intersubjective reality in the form of laws and religion to enforce individual behavior. Most current research in AI safety however involves the codification of subjective reality; this however is problematic in that learning automation intrinsically adaptable and therefore the interpretation of reality is subject to change. By contrast, intersubjective reality is more difficult to change. It is much easier to change one's on perspective that to change an entire society's perspective.

Coincidentally, Harari's classification of reality aligns exactly with the dimension of intelligence described earlier. Objective, subjective and intersubject reality aligns precisely with cognitive capabilities along the dimensions of computational, autonomous and social intelligence. It should be apparent that there is indeed a duality here with intelligence and its perception of reality. Different natures of reality require different kinds of intelligence to successfully thrive.

Provably Beneficial AI

Stuart Russell introduces this idea of "Provably Beneficial AI" where he sets out to explore a new set of robotic laws:

1. The Robot's only objective is to maximize the realization of human values.

2. The robot is initially uncertain about what human values are.

3. Human behavior provides information about human values.

What is striking about this proposal is that open endedness of it and the interplay of human values and an ability of automation to discover its own methods to conform to these rules. The key technological components here are:

- Purely altruistic robots.
- Uncertain Objectives.
- That Learn by observing all humans.

It is clear that Russell's rules are rules for intuition machines and not rules for the popularized kind of super-intelligent AI that we find in the movies. Russell seems to be begging the question for automation that exhibits a high degree of human empathy.

So this is exactly where I want to end this book. That is, with a sense of assurance that the kinds of intelligent automation that we are developing are have the same kinds of virtues that we seek in necessarily in ourselves but absolutely in our friends. That we have higher expectations and demands for the people we associate with.

We've come full circle here, where we establish that the new kind of AIs, known as Deep Learning, is intrinsically intuition machines. That this understanding can help us better leverage this new technology. Finally, that the existential threat of a super-intelligence may perhaps be in the development of altruistic automation.

References

[1SLP]. Coyler, A. (2017, January 03). Matching networks for one shot learning .Retrieved April 20, 2017, from https://blog.acolyer.org/2017/01/03/matching-networks-for-one-shot-learning/

[3DM] 3D makeover for hyper-efficient metalwork. (n.d.). Retrieved April 20, 2017, from http://www.arup.com/news/2015_05_may/11_may_3d_makeover_for_hyper-efficient_metalwork

[3DS] 3D ShapeNets: A Deep Representation for Volumetric Shapes. (n.d.). Retrieved April 20, 2017, from http://3dshapenets.cs.princeton.edu/

[ACT] Wikipedia Contributors. (n.d.). Active Learning. Retrieved April 20, 2017 https://en.wikipedia.org/wiki/Active_learning

[AGO] Hassabis,D. (2017, April 10). Exploring the mysteries of Go with AlphaGo and China's top players. Retrieved April 20, 2017, https://deepmind.com/blog/exploring-mysteries-alphago/

[AIO] Ng, A. (2016, November 11). Hiring Your First Chief AI Officer. Retrieved April 20, 2017, from https://hbr.org/2016/11/hiring-your-first-chief-ai-officer

[AIS] Nanalyze (2017, April 03). The Top 10 Artificial Intelligence Startups in China. Retrieved April 20, 2017, from https://www.nanalyze.com/2017/04/10-artificial-intelligence-startups-china/

[AF3] Anderson, M. (February 3). An interview with Monica Anderson [Interview]. Retrieved April 20, 2017, from https://medium.com/@gk_/an-interview-with-monica-anderson-1fc5962b121c

[AGO] Chan,D. (2017, October 20). The AI That Has Nothing to Learn From Humans. Retrieved April 20, 2017, from https://www.theatlantic.com/technology/archive/2017/10/alphago-zero-the-ai-that-taught-itself-go/543450/

[ALB] Alba, D. (2016, December 06). Only Amazon Could Make a Checkout-Free Grocery Store a Reality. Retrieved April 21, 2017, from https://www.wired.com/2016/12/amazon-go-grocery-store/

[ALBR] Albright, J. (2016, November 18). The #Election2016 Micro-Propaganda. Retrieved April 20, 2017, from https://medium.com/@d1gi/the-election2016-micro-propaganda-machine-383449cc1fba#.m5eco2oay

[ALM] Alba, M. (2016, September 13). Machine Learning Techniques Aim to Reduce Traffic. Retrieved April 27, 2017, from http://www.engineering.com/DesignerEdge/DesignerEdgeArticles/ArticleID/13111/Machine-Learning-Techniques-Aim-to-Reduce-Traffic.aspx

[AMO] Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., & Mané, D. (2016). Concrete problems in AI safety. arXiv preprint arXiv:1606.06565.

[AND] Anderson, M. Artificial Intuition. Retrieved April 20, 2017, from http://artificial-intuition.com/tradeoff.html

[ANDR] Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M. W., Pfau, D., Schaul, T., & de Freitas, N. (2016). Learning to learn by gradient descent by gradient descent. In Advances in Neural Information Processing Systems (pp. 3981-3989).

[ANTI] Perez,C.E. (n.d.). Retrieved April 20, 2017, from http://www.deeplearningpatterns.com/doku.php/anti-causality

[APM] Barbu,A. , & Lay,N. (2011, February 07). An Introduction to Artificial Prediction Markets for Classification . Retrieved April 20, 2017, from https://arxiv.org/abs/1102.1465v6

[ART] Wikipedia Contributors (n.d.) Four Arts. Retrieved April 20, 2017, from https://en.wikipedia.org/wiki/Four_arts

[AUT] AUTOMATED DRIVING . (n.d.). http://www.sae.org/misc/pdfs/automated_driving.pdf

[B12] Barbu, A. (2012). An Introduction to Artificial Prediction Markets for Classification. Journal of Machine Learning Research,13, 2177-2204.

[BAK] Baker, B., Gupta, O., Naik, N., & Raskar, R. (2016). Designing Neural Network Architectures using Reinforcement Learning. arXiv preprint arXiv:1611.02167.

[BAK2] Baker, B., Gupta, O., Naik, N., Raskar, R., & Downs, P. (n.d.). MetaQNN. Retrieved April 20, 2017, from https://bowenbaker.github.io/metaqnn/

[BAN] Hern, A. (2017, May 24). China censored Google's AlphaGo match against world's best Go player. Retrieved April 20, 2017, from https://www.theguardian.com/technology/2017/may/24/china-censored-googles-alphago-match-against-worlds-best-go-player

[BAL] Balduzzi, D. (2014, May). Cortical prediction markets. In Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems (pp. 1265-1272). International Foundation for Autonomous Agents and Multiagent Systems.

[BAL2] Balduzzi, D. (2015). Semantics, Representations and Grammars for Deep Learning. arXiv preprint arXiv:1509.08627.

[BAR] Barr, J. (2016, November 30). Developer Preview – EC2 Instances (F1) with Programmable Hardware. Retrieved April 20, 2017, from http://aws.amazon.com/blogs/aws/developer-preview-ec2-instances-f1-with-programmable-hardware/

[BEC] Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowley, M., . . . Thomas, D. (2001). Principles behind the Agile Manifesto. Retrieved April 21, 2017, from http://agilemanifesto.org/principles.html

[BEN] Benedict, K. (2016, October 31). Merging Humans with Enterprise AI and Machine Learning Systems. Retrieved April 20, 2017, from http://www.futureofwork.com/article/details/merging-humans-with-enterprise-ai-and-machine-learning-systems

[BEN2] Benson, B. (2016, September 01). Cognitive bias cheat sheet – Better Humans. Retrieved July 01, 2017, from https://betterhumans.coach.me/cognitive-bias-cheat-sheet-55a472476b18

[BER] Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. Journal of Machine Learning Research, 13(Feb), 281-305.

[BIO] Wikipedia Contributors (n.d.). Biological Weapons Anti-Terrorism Act of 1989. Retrieved April 20, 2017, from https://en.wikipedia.org/wiki/Biological_Weapons_Anti-Terrorism_Act_of_1989

[BGB] Balog, M., Gaunt, A. L., Brockschmidt, M., Nowozin, S., & Tarlow, D. (2016). DeepCoder: Learning to Write Programs. arXiv preprint arXiv:1611.01989.

[BOG] Bogost, I. (2017, March 04). 'Artificial Intelligence' Has Become Meaningless. Retrieved April 20, 2017, from https://www.theatlantic.com/technology/archive/2017/03/what-is-artificial-intelligence/518547/

[BOO] Boolean network. (2017, April 12). Retrieved April 20, 2017, from https://en.wikipedia.org/wiki/Boolean_network

[BRA] BRAIN4CARS - Cabin Sensing for Safe and Personalized Driving. (n.d.). Retrieved April 21, 2017, from http://brain4cars.com/

[BRI] Britz, D., Goldie, A., Luong, T., & Le, Q. (2017). Massive Exploration of Neural Machine Translation Architectures. arXiv preprint arXiv:1703.03906.

[BRI2] Ross Chainey, Digital Media Specialist, World Economic Forum. (n.d.). Sergey Brin: I didn't see AI coming. Retrieved April 20, 2017, from https://www.weforum.org/agenda/2017/01/google-sergey-brin-i-didn-t-see-ai-coming/

[BRO] Brockman, J., & Stegeman, N. (Eds.). (n.d.). THE ADJACENT POSSIBLE. Retrieved April 20, 2017, from https://www.edge.org/conversation/the-adjacent-possible

[BUC] Buchanan, M. (2002, April 29). Wealth Happens - Wealth Distribution and the Role of Networks. Retrieved April 20, 2017, from http://hbswk.hbs.edu/pubitem.jhtml?id=2906&sid=0&pid=0&t=finance

[BUR] Burns, C. (2015, October 8). SwiftKey Neural Alpha predicts what you'll type. Retrieved April 20, 2017, from https://www.slashgear.com/swiftkey-neural-alpha-predicts-what-youll-type-08408912/

[C18] C. (2016, November 18). RI Seminar: Yann LeCun : The Next Frontier in AI: Unsupervised Learning. Retrieved April 20, 2017, from https://www.youtube.com/watch?v=IbjF5VjniVE

[CAM] Campbell-Dollaghan, K. (2016, November 21). The Algorithmic Democracy. Retrieved April 21, 2017, from https://www.fastcodesign.com/3065582/the-algorithmic-democracy

[CAU] Caulfied, M. (2016, November 13). Despite Zuckerberg's Protests, Fake News Does Better on Facebook Than Real News. Here's Data to Prove It. Retrieved April 20, 2017, from https://hapgood.us/2016/11/13/fake-news-does-better-on-facebook-than-real-news

[CED] Cédric Deltheil Follow. (2014, March 17). Moodstocks - Mobile Image Recognition - Paris Tech Talks #6. Retrieved April 20, 2017, from http://www.slideshare.net/CdricDeltheil1/moodstocks-mobile-image-recognition-paris-tech-talks-6

[CEP] Perez, C. (2017, March 31). The Two Phases of Gradient Descent in Deep Learning. Retrieved April 20, 2017, from https://medium.com/intuitionmachine/the-peculiar-behavior-of-deep-learning-loss-surfaces-330cb741ec17

[CHA] (n.d.). Retrieved April 20, 2017, from http://www.champlain.edu/

[CHA] Chainey Ross Chainey, Digital Media Specialist, World Economic Forum, R. (2017, January 19). Sergey Brin: I didn't see AI coming. Retrieved June 30, 2017, from https://www.weforum.org/agenda/2017/01/google-sergey-brin-i-didn-t-see-ai-coming/

[CHI] Wikipedia Contributors (n.d.) Chinese calligraphy. Retrieved April 20, 2017, from https://en.wikipedia.org/wiki/Chinese_calligraphy

[CHO] Choi, E., Bahadori, M. T., & Sun, J. (2015). Doctor ai: Predicting clinical events via recurrent neural networks. arXiv preprint arXiv:1511.05942.

[CHR] Christensen, C. (2016, October 3). Clayton Christensen: The Theory of Jobs To Be Done [Interview by D. Gerdeman]. Retrieved April 20, 2017, from http://hbswk.hbs.edu/item/clay-christensen-the-theory-of-jobs-to-be-done

[CHN] Mozur, P. (2017, July 20). Beijing Wants A.I. to Be Made in China by 2030. Retrieved April 20, 2017, from https://www.nytimes.com/2017/07/20/business/china-artificial-intelligence.html?mcubz=3

[CHR1] Wikipedia Contributors (n.d.) 棋. Retrieved April 20, 2017, https://en.wiktionary.org/wiki/%E6%A3%8B

[CHU] Chung, K. (2016, July 09). Generating Recommendations at Amazon Scale with Apache Spark and Amazon DSSTNE. Retrieved April 20, 2017, from https://aws.amazon.com/blogs/big-data/generating-recommendations-at-amazon-scale-with-apache-spark-and-amazon-dsstne/

[CLI] Clifford, C. (2017, March 13). Mark Cuban: The world's first trillionaire will be an artificial intelligence entrepreneur. Retrieved June 30, 2017, from http://www.cnbc.com/2017/03/13/mark-cuban-the-worlds-first-trillionaire-will-be-an-ai-entrepreneur.html

[CLJ] Clark, J. (2016, July 19). Google Cuts Its Giant Electricity Bill With DeepMind-Powered AI. Retrieved April 20, 2017, from http://www.bloomberg.com/news/articles/2016-07-19/google-cuts-its-giant-electricity-bill-with-deepmind-powered-ai

[CMP] Hardesty,L. (2016, October 27) Making computers explain themselves: New training technique would reveal the basis for machine-learning systems' decisions. Retrieved April 20, 2017, from http://news.mit.edu/2016/making-computers-explain-themselves-machine-learning-1028

[COG] Benson, B. (2016, September 2). Cognitive bias cheat sheet Because thinking is hard. Retrieved April 20, 2017, from https://betterhumans.coach.me/cognitive-bias-cheat-sheet-55a472476b18

[COM] Competition Is For Losers - Says Billionaire Peter Thiel. (2016, December 09). Retrieved April 20, 2017, from https://www.youtube.com/watch?v=z6K8PZxyQfU

[CON] Nigin, A. (2016, March 7). New Spaceship Speed in Conway's Game of Life. Retrieved April 20, 2017, from https://niginsblog.wordpress.com/2016/03/07/new-spaceship-speed-in-conways-game-of-life/

[COND] Perez,C. (2016, December 28). Is Conditional Logic the New Deep Learning Hotness?. Retrieved April 20, 2017, from https://medium.com/intuitionmachine/is-conditional-logic-the-new-deep-learning-hotness-96832774907b

[CNC] Connectome Coordination Facility .(n.d.). Retrieved April 21, 2017, from https://www.humanconnectome.org/

[CND] CIFAR (2017, May 08). Pan-Canadian Artificial Intelligence Strategy Overview. Retrieved April 21, 2017, from https://www.cifar.ca/assets/pan-canadian-artificial-intelligence-strategy-overview/

[CPM] Balduzzi, M. (2014, January 07). Cortical prediction markets. Retrieved April 21, 2017, from https://arxiv.org/pdf/1401.1465v1.pdf

[CPS] Sigg,S. (2008, February 25).Development of a novel context prediction algorithm and analysis of context prediction schemes. Retrieved April 20, 2017, from http://www.uni-kassel.de/upress/online/frei/978-3-89958-392-2.volltext.frei.pdf

[CSI] Conner-Simons, A. (2016, June 21). Teaching machines to predict the future. Retrieved April 21, 2017, from http://www.csail.mit.edu/teaching_machines_to_predict_the_future

[CTR] Wikipedia Contributors. (n.d.). Control Theory. Retrieved April 20, 2017, from https://en.wikipedia.org/wiki/Control_theory

[CUN] LeCunn, Y. (2016, December 5). Predictive Learning [PDF].

[DAT] Davenport,T., & Patil, D.J. (2012, Ocotober). Data Scientist: The Sexiest Job of the 21st Century. Retrieved April 20, 2017, from https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century

[DEF] Harper, J. (2017, August 21). Pentagon Struggling to Take Advantage of Artificial Intelligence. Retrieved April 21, 2017, from http://www.nationaldefensemagazine.org/articles/2017/8/21/pentagon-struggling-to-take-advantage-of-artificial-intelligence

[DCA] Maluuba Inc (2016, December 15). Decomposing Tasks like Humans: Scaling Reinforcement Learning By Separation of Concerns. Retrieved April 20, 2017, from http://www.maluuba.com/blog/2016/12/9/improving-scalability-of-reinforcement-learning-by-separation-of-concerns

[DCO] Balog, M., Gaunt,A. , Brockschmidt,M., Nowozin,S., & Tarlow,D. (2017). Deep Coder: Learning to write Programs.Retrieved April 20, 2017, https://openreview.net/pdf?id=ByldLrqlx

[DEC] Ioannou, Y., Robertson, D., Zikic, D., Kontschieder, P., Shotton, J…, & Criminisi, A. (2016, March 03). Decision Forests, Convolutional Networks and the Models in-Between. Retrieved April 20, 2017, from https://arxiv.org/pdf/1603.01250v1.pdf

[DES] Design pattern. (2017, April 02). Retrieved April 20, 2017, from https://en.wikipedia.org/wiki/Design_pattern

[DHR] DeepHand: Robust Hand Pose Estimation by Completing a Matrix Imputed with Deep Features. (n.d.). Retrieved April 20, 2017, from https://engineering.purdue.edu/cdesign/wp/deephand-robust-hand-pose-estimation/

[DIST] Maddison, C., Mnih, A., & Teh, Y.W. (2017). The Concrete Distribution: a Continuous Relaxation of Discrete Random Variables. Retrieved April 20, 2017, from http://www.stats.ox.ac.uk/~cmaddis/pubs/concrete.pdf

[DLG] Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2017). Understanding Deep Learning requires re-Thinking generalization.
Retrieved April 20, 2017, from http://openreview.net/pdf?id=Sy8gdB9xx

[DLP] (n.d.). Retrieved April 20, 2017, from http://www.deeplearningpatterns.com/

[DNLS] Im, D.J., Tao, M., & Branson, K. (2016, December 13). An empirical analysis of the optimization of deep network loss surfaces.
Retrieved April 20, 2017, from https://arxiv.org/abs/1612.04010

[DOG] Doglione, C. (2016, July 25). Understanding Responsibility Assignment Matrix (RACI Matrix). Retrieved April 20, 2017, from https://project-management.com/understanding-responsibility-assignment-matrix-raci-matrix/

[DOGS]. Monaghan, E. (2016, June) Soon We Won't Program Computers. We'll Train Them Like Dogs. Retrieved April 20, 2017, from https://www.wired.com/2016/05/the-end-of-code/

[DOL] Dolnick, B. (2016, September 30). No One Understands Donald Trump Like the Horny Narcissist Who Created Dilbert. Retrieved April 20, 2017, from http://www.slate.com/articles/news_and_politics/politics/2016/09/dilbert_creator_scott_adams_gets_trump_like_no_one_else.html

[DOM] Domingos, P. (2017). The master algorithm: How the quest for the ultimate learning machine will remake our world. London: Penguin Books.

[DPT] Dual process theory. (2017, March 29). Retrieved April 20, 2017, from https://en.wikipedia.org/wiki/Dual_process_theory

[DRO] Vincent, J. ()FAA regulations for commercial drones are now in effect https://www.theverge.com/2016/8/30/12707502/drone-regulations-legality-us-faa

[DRG] Kahn,J. (2017, October 19). DeepMind's Superpowerful AI Sets Its Sights on Drug Discovery. Retrieved April 20, 2017, from https://www.bloomberg.com/news/articles/2017-10-18/deepmind-s-superpowerful-ai-sets-its-sights-on-drug-discovery

[DSLR] Kontzer, T. (2017, September 05). How AI Is Breathing New Life Into Digital SLR Cameras. Retrieved April 20, 2017, from https://blogs.nvidia.com/blog/2017/09/05/ai-digital-photography/

[DUN] Dunn, J. (2016, May 9). Introducing FBLearner Flow: Facebook's AI backbone. Retrieved April 20, 2017, from https://code.facebook.com/posts/1072626246134461/introducing-fblearner-flow-facebook-s-ai-backbone/

[ELR] Elejalde-Ruiz, A. (2012, June 13). Gut feelings: How gut feelings work. Retrieved April 27, 2017, from http://articles.chicagotribune.com/2012-06-13/health/sc-health-0613-gut-feelings-20120613_1_intuition-gut-feelings-images

[EQU] Perez,C. (2016, December 16). Equilibrium Discovery in Modular Deep Learning Architectures. Retrieved April 27, 2017, from https://medium.com/intuitionmachine/deep-learning-could-be-market-driven-de770aeebd3

[ETZ] Etzioni, O. (2016, June 15). Deep Learning Isn't a Dangerous Magic Genie. It's Just Math. Retrieved April 21, 2017, from https://www.wired.com/2016/06/deep-learning-isnt-dangerous-magic-genie-just-math/

[EUG] Eugene Wigner. (2017, April 19). Retrieved April 20, 2017, from https://en.wikipedia.org/wiki/Eugene_Wigner

[EVO] Miikkulainen,R., Liang,J., Meyerson,E., Rawal,A., Fink,D… & Hodjat,B.(2017,March 1). Evolving Deep Neural Networks. Retrieved April 20, 2017, from https://arxiv.org/pdf/1703.00548v1.pdf

[EXP] Defense Advanced Research Projects Agency (2017, March 16). Toward Machines that Improve with Experience .Retrieved April 20, 2017, from http://www.darpa.mil/news-events/2017-03-16

[FAR] Farmer develops cucumber sorting machine with the help of Google. (2016, September 6). Retrieved April 20, 2017, from http://www.freshplaza.com/article/162739/Farmer-develops-cucumber-sorting-machine-with-the-help-of-Google

[FBL] Dunn,J. (2016, May 10). Introducing FBLearner Flow: Facebook's AI backbone. Retrieved April 20, 2017, from https://code.facebook.com/posts/1072626246134461/introducing-fblearner-flow-facebook-s-ai-backbone/

[FER] Fernando, C., Banarse, D., Blundell, C., Zwols, Y., Ha, D., Rusu, A. A., ... & Wierstra, D. (2017). Pathnet: Evolution channels gradient descent in super neural networks. arXiv preprint arXiv:1701.08734.

[FEW] Ravi,S., & Larochelle,H. (n.d.). Optimization as a Model for Few-Shot Learning. Retrieved April 20, 2017, from https://openreview.net/pdf?id=rJY0-Kcll

[FEYN] Feng A. (2017, May 14). Learn Anything In Four Steps With The Feynman Technique. Retrieved April 20, 2017, from http://blog.fengtasy.com/learn-anything-in-four-steps-with-the-feynman-technique/

[FIN] Finn, C., Abbeel, P., & Levine, S. (2017). Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. arXiv preprint arXiv:1703.03400.

[FOE] Foerster, J., Assael, Y. M., de Freitas, N., & Whiteson, S. (2016). Learning to communicate with deep multi-agent reinforcement learning. In Advances in Neural Information Processing Systems (pp. 2137-2145).

[FRE] Frey, T. (2017, April 22). 78 Skills that will be Difficult to Automate. Retrieved July 1, 2017, from https://www.linkedin.com/pulse/78-skills-difficult-automate-thomas-frey-1

[FRI] Fridman, L. (2016, September 27). Foundations of Unsupervised Deep Learning (Ruslan Salakhutdinov, CMU). Retrieved April 20, 2017, from https://www.youtube.com/watch?v=rK6bchqeaN8&t=3183s

[FRY] Fry, R. L. (2017). Physical Intelligence and Thermodynamic Computing. Entropy, 19(3), 107.

[GAN] Salimans,T. , Goodfellow,I. , Zaremba,W. , Cheung,V. Radford,A., & Chen,X. (2016, June 10). Improved Techniques for Training GANs. Retrieved April 20, 2017, from https://arxiv.org/abs/1606.03498

[GAP] Im,D.J. Ma,H. , Kim,C.D., & Taylor,G. (2016, December 13) Generative Adversarial Parallelization. Retrieved April 20, 2017, from https://arxiv.org/abs/1612.04021

[GAS] Gaskell, A. (2016, October 26). Machine learning and the hunt for dementia. Retrieved April 20, 2017, from http://www.huffingtonpost.com/adi-gaskell/machine-learning-and-the-_b_12652122.html

[GEN] Wikipedia Contributors. (n.d.) Regulation of genetic engineering. Retrieved April 20, 2017, from https://en.wikipedia.org/wiki/Regulation_of_genetic_engineering

[GBDL]. Shalev-Shwartz, S., Shamir, O., & Shammah,S. Retrieved April 20, 2017, from https://arxiv.org/pdf/1703.07950v2.pdf

[GGL] Good, O. (2015, July 29). How Google Translate squeezes deep learning onto a phone. Retrieved July 01, 2017, from https://research.googleblog.com/2015/07/how-google-translate-squeezes-deep.html

[GGO] Wikipedia Contributors (n.d.) Retrieved July 01, 2017, from Go (game) https://en.wikipedia.org/wiki/Go_%28game%29

[GK] Gk (2017, February 03). An interview with Monica Anderson – gk_ – Medium. Retrieved April 20, 2017, from https://medium.com/@gk_/an-interview-with-monica-anderson-1fc5962b121c

[GLA] Gladwell, M. (2013). Blink: the power of thinking without thinking. New York: Back Bay Books.

[GOO2] Google (2017, April 17). Google/seq2seq. Retrieved April 20, 2017, from https://github.com/google/seq2seq

[GOO] Google (2016, March 22). Google Photos: Introducing New, Smarter Albums. Retrieved April 20, 2017, from https://www.youtube.com/watch?v=JuFtW1PSYAU

[GOOD] Good, O. (2015, July 29). How Google Translate squeezes deep learning onto a phone [Web log post]. Retrieved April 20, 2017, from https://research.googleblog.com/2015/07/how-google-translate-squeezes-deep.html

[GRE] Greff, K., Rasmus, A., Berglund, M., Hao, T. H., Schmidhuber, J., & Valpola, H. (2016). Tagger: Deep Unsupervised Perceptual Grouping. 1-19.

[GQN] Wikipedia Contributors (n.d.) Guqin. Retrieved April 20, 2017, from https://en.wikipedia.org/wiki/Guqin

[GUC] Güçlütürk, Y., Güçlü, U., van Lier, R., & van Gerven, M. A. (2016, October). Convolutional sketch inversion. In European Conference on Computer Vision (pp. 810-824). Springer International Publishing.

[GUM] Jang,E. Gu,S. & Poole,B. (2016, November 03). Categorical Reparameterization with Gumbel-Softmax. Retrieved April 20, 2017, from https://arxiv.org/abs/1611.01144

[HAG] Hagel, J. (2016, April 05). Harnessing the Full Potential of Platforms. Retrieved April 21, 2017, from http://www.marketingjournal.org/john-hagel-harnessing-the-full-potential-of-platforms/

[HAM] Hamrick, J. B., Ballard, A. J., Pascanu, R., Vinyals, O., Heess, N., & Battaglia, P. W. (2016, November 04). Metacontrol for Adaptive Imagination-Based Optimization. Retrieved April 20, 2017, from https://openreview.net/forum?id=Bk8BvDqex

[HAR] Hartung, A. (2014, April 15). Three Smart Lessons From Facebook's Purchase Of WhatsApp. Retrieved April 20, 2017, from http://www.forbes.com/sites/adamhartung/2014/02/24/zuckerbergs-3-smart-leadership-lessons-from-facebook-buying-whatsapp/#ce3d5e31d917

[HAS] Hassabis, D. (2017, April 10). Exploring the mysteries of Go with AlphaGo and China's top players. Retrieved April 27, 2017, from https://deepmind.com/blog/exploring-mysteries-alphago/

[HAV] Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., ... & Larochelle, H. (2017). Brain tumor segmentation with deep neural networks. Medical image analysis, 35, 18-31.

[HEI] Heinrich, J., & Silver, D. (2016). Deep reinforcement learning from self-play in imperfect-information games. arXiv preprint arXiv:1603.01121.

[HEN] Henderson, M., Al-Rfou, R., Strope, B., Sung, Y., Lukacs, L., Guo, R., . . . Kurzweil, R. (2017, May 01). Efficient Natural Language Response Suggestion for Smart Reply. Retrieved July 01, 2017, from https://arxiv.org/abs/1705.00652

[HIN2] Hintze, A. (2016, November 13). Understanding the four types of AI, from reactive robots to self-aware beings. Retrieved April 27, 2017, from http://theconversation.com/understanding-the-four-types-of-ai-from-reactive-robots-to-self-aware-beings-67616

[HIN] Hintze, A. (2016, October 27). Arend Hintze - Profile. Retrieved April 27, 2017, from http://theconversation.com/profiles/arend-hintze-225106

[HL16] Hardesty, L. (2016, October 27). Making computers explain themselves. Retrieved May 01, 2017, from http://news.mit.edu/2016/making-computers-explain-themselves-machine-learning-1028

[HOF] Hofstadter, D. R. (2008). I am a strange loop. New York: Basic Books.

[HOL] Holt, J. (2011, November 26). Two Brains Running. Retrieved April 20, 2017, from http://www.nytimes.com/2011/11/27/books/review/thinking-fast-and-slow-by-daniel-kahneman-book-review.html

[HOM] Home. (n.d.). Retrieved April 20, 2017, from http://www.maluuba.com/

[HOMO] Homo economicus. (2017, March 21). Retrieved April 20, 2017, from https://en.wikipedia.org/wiki/Homo_economicus

[HOS] Hosanagar, K., & Saxena, A. (2017, April 18). The First Wave of Corporate AI Is Doomed to Fail. Retrieved April 20, 2017, from https://hbr.org/2017/04/the-first-wave-of-corporate-ai-is-doomed-to-fail

[HUJ] Hu, J., & Storkey, A. (2014, January). Multi-period trading prediction markets with connections to machine learning. In International Conference on Machine Learning (pp. 1773-1781).

[HUM] The Human Connectome Project. (2017, March 1). Retrieved April 21, 2017, from https://www.humanconnectome.org/

[HUMP] Humphries, T. E. (2012, May). CONSIDERING INTUITION IN THE CONTEXT OF DESIGN, AND OF PSYCHOLOGY [Scholarly project]. Retrieved April 21, 2017, from http://theohumphries.com/papers/HUMPHRIES_intuition_psychology_design.pdf

[HWL] Hartford, J. S., Wright, J. R., & Leyton-Brown, K. (2016). Deep learning for predicting human strategic behavior. In Advances in Neural Information Processing Systems (pp. 2424-2432).

[HYP] Bergstra,J. & Bengio,Y. (2012, February). Random Search for Hyper-Parameter Optimization. Retrieved April 20, 2017, from http://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a

[IDL] Institute of Deep Learning. (n.d.). Retrieved April 20, 2017, from http://research.baidu.com/institute-of-deep-learning/

[IFE] Ifeanyi, K. (2014, October 01). See How Far Everything On Your Desk Has Evolved. Retrieved April 20, 2017, from https://www.fastcompany.com/3036134/the-recommender/see-how-far-everything-on-your-desk-has-evolved

[IID] Independent and identically distributed random variables. (2017, April 14). Retrieved April 21, 2017, from https://en.wikipedia.org/wiki/Independent_and_identically_distributed_random_variables

[IM2] Im, D. J., Tao, M., & Branson, K. An empirical analysis of the optimization of deep network loss surfaces.

[IM] Im, D. J., Ma, H., Kim, C. D., & Taylor, G. (2016). Generative Adversarial Parallelization. arXiv preprint arXiv:1612.04021.

[IMG] Taigman, Y. Polyak, A. & Wolf, L. (2016, November 07). Unsupervised Cross-Domain Image Generation. Retrieved April 20, 2017, from
https://arxiv.org/abs/1611.02200

[IMG2] Isola,P.,Zhu,J.,Zhou,T., & Efros,A.(2016, November 21). Retrieved April 20, 2017, from https://arxiv.org/pdf/1611.07004v1.pdf

[IMP] Heinrich J., & Silver D. (2016, March 03). Deep Reinforcement Learning from Self-Play in Imperfect-Information Games. Retrieved April 20, 2017, from https://arxiv.org/abs/1603.01121

[INF] Perez, C. (2016, December 10). Deep Learning is Non-Equilibrium Information Dynamics. Retrieved April 20, 2017, from https://medium.com/intuitionmachine/deep-learning-is-non-equilibrium-information-dynamics-b00baa16b135#.4idmwnrog

[INT] Perez,C. (2017, February 12). Artificial Intuition—A Breakthrough Cognitive Paradigm. Retrieved April 20, 2017, from https://medium.com/intuitionmachine/artificial-intuition-a-breakthrough-cognitive-paradigm-3905c6d76561

[ISO] ISO/IEC 9126. (2017, April 05). Retrieved June 30, 2017, from
https://en.wikipedia.org/wiki/ISO/IEC_9126

[ISOL] Isola, P., Zhu, J., Zhao, T., & Efros, A. (n.d.). Image-to-Image Translation with Conditional Adversarial Networks. Retrieved 16.

[ITR] Greff,K. , Srivastava,R. & Schmidhuber,J. (2016, December 22). Highway and residual networks learn unrolled iterative estimation Retrieved April 20, 2017, from
https://arxiv.org/pdf/1612.07771v1.pdf

[J2BD] Clayton Christensen Institute. (n.d.). Jobs to Be Done. Retrieved April 20, 2017, from https://www.christenseninstitute.org/jobs-to-be-done/

[JAD] Jaderberg, M. Decoupled Neural Interfaces Using Synthetic Gradients. (2016, August 29). Retrieved April 20, 2017, from https://deepmind.com/blog/decoupled-neural-networks-using-synthetic-gradients/

[JAN] Jang, E., Gu, S., & Poole, B. (2016). Categorical Reparameterization with Gumbel-Softmax. arXiv preprint arXiv:1611.01144.

[JNW] Lohr, S.(2014, August 17). For Big-Data Scientists, 'Janitor Work' Is Key Hurdle to Insights . Retrieved April 20, 2017, from https://www.nytimes.com/2014/08/18/technology/for-big-data-scientists-hurdle-to-insights-is-janitor-work.html?_r=0

[JOB] Jobs To Be Done. (n.d.). Retrieved April 20, 2017, from https://www.christenseninstitute.org/jobs-to-be-done/

[JRB] Jr., B. M. (2017, January 13). Facebook's Mysterious Job Listing Sounds Like It's Working on How to Read Your Mind. Retrieved April 20, 2017, from http://www.inc.com/bill-murphy-jr/facebooks-mysterious-job-listing-sounds-like-its-working-on-how-to-read-your-min.html

[JUP] Kenway, O. (2017, October 03). Why I don't like Jupyter Notebooks - Dr Owain Kenway. Retrieved April 20, 2017, from https://owainkenwayucl.github.io/2017/10/03/WhyIDontLikeNotebooks.html

[KAE] Karch, E. (2011, April 1) Lehman's Laws of Software Evolution and the Staged-Model. Retrieved April 20, 2017, from https://blogs.msdn.microsoft.com/karchworld_identity/2011/04/01/lehmans-laws-of-software-evolution-and-the-staged-model/

[KAHN] Wikipedia Contributors. (n.d.) Daniel Kahneman. Retrieved April 20, 2017, from https://en.wikipedia.org/wiki/Daniel_Kahneman

[KAH] Kahneman, D. (2015). Thinking, fast and slow. New York: Farrar, Straus and Giroux.

[KAN] Kannan, A., Kurach, K., Ravi, S., Kaufmann, T., Tomkins, A., Miklos, B., . . . Ramavajjala, V. (2016, June 15). Smart Reply: Automated Response Suggestion for Email. Retrieved July 01, 2017, from https://arxiv.org/abs/1606.04870

[KAR] Karpathy, A., Abbeel, P., Brockman, G., Chen, P., Cheung, V., Duan, R., . . . Zaremba, W. (2017, March 23). Generative Models. Retrieved April 21, 2017, from https://blog.openai.com/generative-models/

[KAS] Kasanoff, B. (2017, April 12). Intuition Is The Highest Form Of Intelligence. Retrieved April 20, 2017, from https://www.forbes.com/sites/brucekasanoff/2017/02/21/intuition-is-the-highest-form-of-intelligence/#40c6d6e53860

[KEE] Keegan, J. (2016, May 16). Blue Feed, Red Feed. Retrieved April 20, 2017, from http://graphics.wsj.com/blue-feed-red-feed/

[KHA] Khaitan, P. (2016, May 18). Chat Smarter with Allo [Web log post]. Retrieved April 20, 2017, from https://research.googleblog.com/2016/05/chat-smarter-with-allo.html

[KIL] Kilpi, E. (2015, September 26). Situational work – What's The Future? – Medium. Retrieved April 20, 2017, from https://medium.com/the-wtf-economy/situated-work-b8cf60854d65

[KNI2] Knight, W. (2017, January 12). Poker may be the latest game to fold against artificial intelligence. Retrieved April 20, 2017, from https://www.technologyreview.com/s/603342/poker-is-the-latest-game-to-fold-against-artificial-intelligence/

[KNI3] Knight, W. (2017, March 27). The Man with a Plan to Make AI More Human. Retrieved April 20, 2017, from https://www.technologyreview.com/s/544606/can-this-man-make-aimore-human/

[KNI] Knight, W. (2017, April 19). There's a big problem with AI: Even its creators can't explain how it works. Retrieved April 20, 2017, from https://www.technologyreview.com/s/604087/the-dark-secret-at-the-heart-of-ai/

[KOC] Koch, C. (2016, March 18). How the Computer Beat the Go Master. Retrieved April 20, 2017, from http://www.scientificamerican.com/article/how-the-computer-beat-the-go-master/

[KOE] Koetser, J. (2016, August 02). A.D. 2035: Rich people will be thousands of times smarter than poor people. Retrieved June 30, 2017, from https://venturebeat.com/2016/08/02/a-d-2035-rich-people-will-be-thousands-of-times-smarter-than-poor-people/

[KOR] Korolov, M. (2016, October 03). Data leaks evolving into weapons of business destruction. Retrieved April 20, 2017, from http://www.csoonline.com/article/3126467/security/data-leaks-evolving-into-weapons-of-business-destruction.html

[K-FAC]. Grosse, R., & Martens, J. (2016, February 03). A Kronecker-factored approximate Fisher matrix for convolution layers. Retrieved April 20, 2017, from https://arxiv.org/abs/1602.01407v2

[LAK] Lake, B. M., Ullman, T. D., Tenenbaum, J. B., & Gershman, S. J. (2016, November 02). Building Machines That Learn and Think Like People. Retrieved June 30, 2017, from https://arxiv.org/abs/1604.00289

[LAN] Langley, P. (2012, July). The Cognitive Systems Paradigm [Scholarly project]. Retrieved April 21, 2017, from http://www.cogsys.org/pdf/paper-1-2.pdf

[LAU2] Launchbury, J. (2017, February 15). A DARPA Perspective on Artificial Intelligence. Retrieved April 20, 2017, from https://www.youtube.com/watch?v=-O01G3tSYpU

[LAU] Launchbury, J. (n.d.). A DARPA Perspective on Artificial Intelligence [PDF]. DARPA.

[LEA] The Lean Startup. (n.d.). Retrieved April 21, 2017, from http://theleanstartup.com/

[LEH] Lehrer, J. (2011, July 15). Is Google Ruining Your Memory? Retrieved April 20, 2017, from https://www.wired.com/2011/07/is-google-ruining-your-memory/

[LEI] Leibo, J., Zambaldi, V., Lanctot, M., Marecki, J., & Graepel, T. (2017, February 9). Understanding Agent Cooperation. Retrieved April 20, 2017, from https://deepmind.com/blog/understanding-agent-cooperation/

[LET] Le, T. A., Baydin, A. G., Zinkov, R., & Wood, F. (2017). Using Synthetic Data to Train Neural Networks is Model-Based Reasoning. arXiv preprint arXiv:1703.00868.

[LET] Letting Go of Efficiency Can Accelerate Your Company - Here's How. (2015, January 24). Retrieved April 20, 2017, from http://firstround.com/article/Responsiveness-New-Efficiency

[LEW] Lewis-Kraus, G. (2016, December 14). The Great A.I. Awakening. Retrieved April 20, 2017, from http://www.nytimes.com/2016/12/14/magazine/the-great-ai-awakening.html?_r=0

[LIA] Liao, J., Yao, Y., Yuan, L., Hua, G., & Kang, S. (2017). Visual Attribute Transfer through Deep Image Analogy. 1-16.

[LIB] Metz, C. (2017, February 1). Inside Libratus, the poker AI that out-bluffed the best humans. Retrieved April 20, 2017, from https://www.wired.com/2017/02/libratus/

[LIM] Limmer, M., & Lensch, H. P. Infrared Colorization Using Deep Convolutional Neural Networks.

[LIS] Wikipedia Contributors. (2017, April 11). List of cognitive biases. Retrieved April 20, 2017, from https://en.wikipedia.org/wiki/List_of_cognitive_biases

[LIV] Lawrence, N. (2017, May 22). Living Together: Mind and Machine Intelligence. Retrieved April 20, 2017, from https://arxiv.org/abs/1705.07996v1

[LOH] Lohr, S. (2014, August 17). For Big-Data Scientists, 'Janitor Work' Is Key Hurdle to Insights. Retrieved April 20, 2017, from https://www.nytimes.com/2014/08/18/technology/for-big-data-scientists-hurdle-to-insights-is-janitor-work.html?_r=0

[LOM] Lomas, N. (2016, June 24). Prisma uses AI to turn your photos into graphic novel fodder double quick. Retrieved July 01, 2017, from https://techcrunch.com/2016/06/24/prisma-uses-ai-to-turn-your-photos-into-graphic-novel-fodder-double-quick/

[LOR] Loreto, V., Servedio, V. D., Strogatz, S. H., & Tria, F. (2017, January 04). Dynamics on expanding spaces: modeling the emergence of novelties. Retrieved June 30, 2017, from https://arxiv.org/abs/1701.00994

[LOU] Louppe, G., Cho, K., Becot, C., & Cranmer, K. (2017). QCD-Aware Recursive Neural Networks for Jet Physics. arXiv preprint arXiv:1702.00748.

[LOV] Love, D. (2016, November 18). The Next Global Arms Race Aims to Perfect Artificial Intelligence. Retrieved April 20, 2017, from http://www.nbcnews.com/mach/features/next-global-arms-race-aims-perfect-artificial-intelligence-n685911

[LPD] Lopez-Paz, D., Muandet, K., Schölkopf, B., & Tolstikhin, I. (2015, June). Towards a learning theory of cause-effect inference. In International Conference on Machine Learning (pp. 1452-1461).

[LRN] Kool,W. , Cushman,F. & Gershman,S. (n.d.). Competition and Cooperation Between Multiple Reinforcement Learning Systems. Retrieved April 20, 2017, from http://gershmanlab.webfactional.com/pubs/KoolCushmanGershman_CompCoop.pdf

[LUX] (n.d.). Retrieved April 20, 2017, from http://www.luxresearchinc.com/content/could-ai-startup-geometric-intelligence-have-prevented-tesla's-fatal-crash

[LTL] Shazeer,N. , Mirhoseini,A. , Maziarz,K. , Davis,A. , Le,Q… & Dean,J. (2017, January 23). Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. Retrieved April 20, 2017, from https://arxiv.org/abs/1701.06538

[LRN] Retrieved April 20, 2017, from https://arxiv.org/abs/1611.05763v2

[MAB] Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., ... & Zhang, X. (2016). End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316.

[MACH] Simard,P. , Amershi,S. , Chickering,D. , Pelton,A.E. , Ghorash,S... & Wernsing, J. (2017, July 21). Machine Teaching: A New Paradigm for Building Machine Learning Systems. Retrieved June 30, 2017, from https://arxiv.org/abs/1707.06742v2

[MAL] Maluuba. (2016, December 15). Decomposing Tasks like Humans: Scaling Reinforcement Learning By Separation of Concerns. Retrieved April 30, 2017, from http://www.maluuba.com/blog/2016/12/9/improving-scalability-of-reinforcement-learning-by-separation-of-concerns

[MAN] Manning, C. D. (2015). Computational Linguistics and Deep Learning. Computational Linguistics, 41(4), 701-707. doi:10.1162/coli_a_00239

[MAT] Mathematicians have discovered how the universal patterns behind innovation arise (2017, February 03). Retrieved April 20, 2017, from https://www.technologyreview.com/s/603366/mathematical-model-reveals-the-patterns-of-how-innovations-arise/

[MCC] McCarthy, C. (2017, April 20). The Kekulé Problem - Issue 47: Consciousness. Retrieved June 30, 2017, from http://nautil.us/issue/47/consciousness/the-kekul-problem

[MER2] Merity, S. (2017, February 26). Stop saying DeepCoder steals code from StackOverflow. Retrieved April 20, 2017, from http://smerity.com/articles/2017/deepcoder_and_ai_hype.html

[MER] Merity, S. (2016, November 17). Peeking into the neural network architecture used for Google's Neural Machine Translation. Retrieved April 20, 2017, from http://smerity.com/articles/2016/google_nmt_arch.html

[MERG] Benedict, K. (2016, October 31). Merging Humans with Enterprise AI and Machine Learning Systems. Retrieved April 20, 2017, from http://www.futureofwork.com/article/details/merging-humans-with-enterprise-ai-and-machine-learning-systems

[MET2] Metz, C. (2015, November 03). Soon, Gmail's AI Could Reply to Your Email for You. Retrieved April 20, 2017, from http://www.wired.com/2015/11/google-is-using-ai-to-create-automatic-replies-in-gmail/

[MET3] Metz, C. (2016, December 05). Uber Buys a Mysterious Startup to Make Itself an AI Company. Retrieved April 20, 2017, from https://www.wired.com/2016/12/uber-buys-mysterious-startup-make-ai-company/

[MET] Metz, C. (2015, July 09). Google Says Its AI Catches 99.9 Percent of Gmail Spam. Retrieved April 20, 2017, from https://www.wired.com/2015/07/google-says-ai-catches-99-9-percent-gmail-spam/

[META] Hamrick, J., Ballard, A., Pascanu, R., Vinyals, O., Heess, N., & Battaglia, P. (2016, November 05). Metacontrol for Adaptive Imagination-Based Optimization. Retrieved April 20, 2017, from https://openreview.net/forum?id=Bk8BvDqex

[MIB] Newitz,A. (2017, August 09).MIT, IBM team up on $240 million effort to rule the AI world. Retrieved April 20, 2017, from https://arstechnica.com/information-technology/2017/09/ibm-partners-with-mit-for-240-million-fundamental-ai-research-project/

[MII] Miikkulainen, R., Liang, J., Meyerson, E., Rawal, A., Fink, D., Francon, O., ... & Hodjat, B. (2017). Evolving Deep Neural Networks. arXiv preprint arXiv:1703.00548.

[MIL] Milner, R. (1997). Turing, computing and communication. In Interactive Computation (pp. 1-8). Springer Berlin Heidelberg.

[MIS] Wikipedia Contributors.(n.d.). Computer Misuse Act 1990.Retrieved April 20, 2017, from https://en.wikipedia.org/wiki/Computer_Misuse_Act_1990

[MLD] 大トロ ml ・ design. (2016, September 29). Retrieved April 20, 2017, from http://blog.otoro.net/2016/09/28/hyper-networks/

[MLTF] Google Developers. (2017, September 21). How Machine Learning with TensorFlow Enabled Mobile Proof-Of-Purchase at Coca-Cola. \ Retrieved April 20 2017 from https://developers.googleblog.com/2017/09/how-machine-learning-with-tensorflow.html

[MMN] Moravčík, M., Schmid, M., Burch, N., Lisý, V., Morrill, D., Bard, N., ... & Bowling, M. (2017). DeepStack: Expert-Level Artificial Intelligence in No-Limit Poker. arXiv preprint arXiv:1701.01724.

[MNUC] Vavra,S. (2017, March 24). Mnuchin: Losing human jobs to AI "not even on our radar screen" Retrieved April 20, 2017, from https://www.axios.com/treasury-secretary-mnuchin-interviews-with-axios-live-updates-2327865447.html

[MPT] Hu,J. & Storkey,A. (2014, March 04). Multi-period Trading Prediction Markets with Connections to Machine Learning. Retrieved July 01, 2017, from https://arxiv.org/pdf/1403.0648v1.pdf

[MOD] Le,T.A., Baydin,A.G., Zinkov,R. & Wood,F. (2017, March 02). Using Synthetic Data to Train Neural Networks is Model-Based Reasoning. Retrieved July 01, 2017, from https://arxiv.org/pdf/1703.00868v1.pdf

[MOO] Moorhead, P. (2016, December 12). AMD Enters Deep Learning Market With Instinct Accelerators, Platforms And Software Stacks. Retrieved April 20, 2017, from http://www.forbes.com/sites/patrickmoorhead/2016/12/12/amd-enters-deep-learning-market-with-instinct-branded-accelerators-and-software-stacks/#21f01ec43fec

[MOR] Arsiwalla,X. , Moulin-Frier,C. Herreros,I. ,Sanchez-Fibla,M., & Verschure,P. (2017, May 31) The Morphospace of Consciousness. Retrieved April 20, 2017, from https://arxiv.org/abs/1705.11190

[MUL] Wikipedia Contributors (n.d.). Theory of multiple intelligences. Retrieved April 20, 2017, from https://en.wikipedia.org/wiki/Theory_of_multiple_intelligences

[MUN] Munkhdalai, T., & Yu, H. (2017). Meta Networks. arXiv preprint arXiv:1703.00837.

[NAR] Narayanan, H. (n.d.). Convolutional neural networks for artistic style transfer. Retrieved July 01, 2017, from https://harishnarayanan.org/writing/artistic-style-transfer/

[NAT] Nathan, P. (2016, July 04). Beyond the AI Winter – Synecdoche. Retrieved April 27, 2017, from https://synecdoche.liber118.com/beyond-the-ai-winter-941c0a66b4f5

[NDEA] Wikipedia Contributors (n.d.) National Defense Education Act. Retrieved April 20, 2017, from https://en.wikipedia.org/wiki/National_Defense_Education_Act

[NEG] Papangelis,A. & Georgila,K. (2015). Reinforcement Learning of Multi-Issue Negotiation Dialogue Policies. Retrieved April 20, 2017, from https://www.aclweb.org/anthology/W/W15/W15-4621.pdf

[NEU] Neuberg, B. (2017, April 12). Creating a Modern OCR Pipeline Using Computer Vision and Deep Learning. Retrieved July 01, 2017, from https://blogs.dropbox.com/tech/2017/04/creating-a-modern-ocr-pipeline-using-computer-vision-and-deep-learning/

[NEW] New Spaceship Speed in Conway's Game of Life. (2016, March 7). Retrieved April 20, 2017, from https://niginsblog.wordpress.com/2016/03/07/new-spaceship-speed-in-conways-game-of-life/

[NIK] Nikkei Inc. (2017, August 22). Japan to pump funding into AI chip development. Retrieved April 20, 2017, from https://asia.nikkei.com/Politics-Economy/Economy/Japan-to-pump-funding-into-AI-chip-development

[NGA2] Ng, A. (2016, November 11). Hiring Your First Chief AI Officer. Retrieved April 20, 2017, from https://hbr.org/2016/11/hiring-your-first-chief-ai-officer

[NGA] Ng, A. (2016, November 09). Andrew Ng: What AI Can and Can't Do. Retrieved April 20, 2017, from https://hbr.org/2016/11/what-artificial-intelligence-can-and-cant-do-right-now

[NIG] Nigin, A. (2016, October 11). New Spaceship Speed in Conway's Game of Life. Retrieved April 21, 2017, from https://niginsblog.wordpress.com/2016/03/07/new-spaceship-speed-in-conways-game-of-life/

[NON] Non-functional requirement. (2017, June 28). Retrieved June 30, 2017, from https://en.wikipedia.org/wiki/Non-functional_requirement

[NUS] Nussbaum, J. (2016, October 25). Data Sets Are The New Server Rooms – Hacker Noon. Retrieved April 20, 2017, from https://hackernoon.com/data-sets-are-the-new-server-rooms-40fdb5aed6b0

[OCC] Occam's razor. (2017, April 16). Retrieved April 20, 2017, from https://en.wikipedia.org/wiki/Occam's_razor

[OLA] Olah, C. (2016, June 21). Bringing Precision to the AI Safety Discussion. Retrieved July 01, 2017, from https://research.googleblog.com/2016/06/bringing-precision-to-ai-safety.html

[OLNN] Shazeer,N., Mirhoseini,A., Maziarz,K., Davis,A., Le,Q…& Dean,J.(2017, January 23). Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. Retrieved April 20, 2017, from https://arxiv.org/abs/1701.06538

[OPE] Open Letter on Autonomous Weapons. (n.d.). Retrieved April 20, 2017, from http://futureoflife.org/open-letter-autonomous-weapons/

[OPT] Optimizing Warehouse Operations with Machine Learning on GPUs. (2016, January 10). Retrieved April 20, 2017, from https://devblogs.nvidia.com/parallelforall/optimizing-warehouse-operations-machine-learning-gpus/

[PAIR] Google (n.d.) PAIR | People+AI Research Initiative. Retrieved April 20, 2017, from https://ai.google/pair

[PAN] Pane, J. F., Steiner, E. D., Baird, M., & Hamilton, L. S. (2015, November 10). Promising Evidence on Personalized Learning. Retrieved April 20, 2017, from http://www.rand.org/pubs/research_reports/RR1365.html

[PAP] Papangelis, A., & Georgila, K. (2015). Reinforcement Learning of Multi-Issue Negotiation Dialogue Policies. Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 154-158. doi:10.18653/v1/w15-4621

[PAR] Parasnis, A. (2015, November 2). Adobe Sensei: Mastering Content and Data. Retrieved July 01, 2017, from https://blogs.adobe.com/conversations/2016/11/adobe-sensei.html

[PARG] Paragios, N. (2016, June 5). Computer Vision Research: The deep "depression" [Web log post]. Retrieved April 27, 2017, from https://www.linkedin.com/pulse/computer-vision-research-my-deep-depression-nikos-paragios

[PAS] Pastur-Romay, L., Cedrón, F., Pazos, A., & Porto-Pazos, A. (2016, August 11). Deep Artificial Neural Networks and Neuromorphic Chips for Big Data Analysis: Pharmaceutical and Bioinformatics Applications. Retrieved April 20, 2017, from http://www.mdpi.com/1422-0067/17/8/1313/htm

[PER13] Perez, C. E. (2016, December 24). The Alien Style of Deep Learning Generative Design Retrieved April 20, 2017, from https://medium.com/intuitionmachine/the-alien-look-of-deep-learning-generative-design-5c5f871f7d10#.6fk9keqax

[PER15] Perez, C. E. (2016, December 28). Is Conditional Logic the New Deep Learning Hotness? Retrieved April 20, 2017, from https://medium.com/intuitionmachine/is-conditional-logic-the-new-deep-learning-hotness-96832774907b#.sci48djfv

[PER19] Perez, C. E. (2017, January 20). How to Explain Deep Learning using Chaos and Complexity. Retrieved April 20, 2017, from https://medium.com/intuitionmachine/how-to-explain-deep-learning-using-chaos-and-complexity-33de81c321de#.s43wdcszy

[PER23] Perez, C. E. (2017, February 25). A Language Driven Approach for Deep Learning Training. Retrieved April 20, 2017, from https://medium.com/intuitionmachine/a-generalized-bootstrapping-framework-for-deep-learning-architectures-970075bad781#.msfxgq85b

[PER2] Perez, C. E. (2016, November 06). Deep Learning: The Unreasonable Effectiveness of Randomness. Retrieved April 20, 2017, from https://medium.com/intuitionmachine/deep-learning-the-unreasonable-effectiveness-of-randomness-14d5aef13f87

[PER9] Perez, C. E. (2016, December 10). Deep Learning is Non-Equilibrium Information Dynamics. Retrieved April 20, 2017, from https://medium.com/intuitionmachine/deep-learning-is-non-equilibrium-information-dynamics-b00baa16b135#.y2u4m3cpx

[PER] Perez, C. (2003, December 6). The 8 Laws of Software Evolution. Retrieved April 27, 2017, from http://www.manageability.org/blog/stuff/the-8-laws-of-software-evolution/view

[PERR] Perez, C. E. (2017, January 06). Modular Deep Learning could be the Penultimate Step to Consciousness. Retrieved June 30, 2017, from https://medium.com/intuitionmachine/modular-deep-learning-and-consciousness-c284ac3aeda3

[PERY] Perez, C. E. (2016, November 16). Rethinking Generalization in Deep Learning – Intuition Machine – Medium. Retrieved June 30, 2017, from https://medium.com/intuitionmachine/rethinking-generalization-in-deep-learning-ec66ed684ace

[PIC] Pickersgill, E. (n.d.). Photographs. Retrieved April 20, 2017, from https://www.noshow.social/

[PGN] Dolittle, Dr. (2017, September 29). Pigeons outperform humans when it comes to multitasking. . Retrieved April 20, 2017 ,from http://scienceblogs.com/lifelines/2017/09/29/pigeons-outperform-humans-when-it-comes-to-multitasking/

[POL] Scherrer,B. (2014, May 12). Approximate Policy Iteration Schemes: A Comparison. Retrieved April 20, 2017 ,from https://arxiv.org/abs/1405.2878

[PLA] Planarian regeneration model discovered by artificial intelligence. (n.d.). Retrieved April 20, 2017, from http://phys.org/news/2015-06-planarian-regeneration-artificial-intelligence.html

[PNET] Fernando,C. , Banarse,D. , Blundell,C. , Zwols,Y. , Ha,D… & Wierstra,D. (2017, January 30). PathNet: Evolution Channels Gradient Descent in Super Neural Networks. Retrieved April 20, 2017, from https://arxiv.org/abs/1701.08734

[PNN] Rusu, A., Rabinowitz, N. , Desjardins, G., Soyer, H., Kirkpatrick, J… & Hadsell, R. (n.d.). Progressive Neural Networks. Retrieved April 20, 2017, from https://arxiv.org/pdf/1606.04671.pdf

[PNT] Wikipedia Contributors (n.d.) Chinese painting. Retrieved April 20, 2017, from https://en.wikipedia.org/wiki/Chinese_painting

[PRE] Olah,C. (2016, June 21). Bringing Precision to the AI Safety Discussion. Retrieved April 20, 2017, from https://research.googleblog.com/2016/06/bringing-precision-to-ai-safety.html

[PRI] Tanasoiu, F. (2017, May 22). Mobile App Success Story: How Prisma Did It. Retrieved April 20, 2017, from
https://appsamurai.com/mobile-app-success-story-how-prisma-did-it/

[PUT] Klimentyev,M. (2017, August 09).For Superpowers, Artificial Intelligence Fuels New Global Arms Race. Retrieved April 20, 2017, from https://www.wired.com/story/for-superpowers-artificial-intelligence-fuels-new-global-arms-race/

[PUT1] Gershgorn, D. (2017, September 01). Vladimir Putin believes artificial intelligence could lead to global monopolies and drone wars. Retrieved April 20, 2017, from https://qz.com/1068015/vladimir-putin-sees-global-monopolies-and-drone-wars-within-artificial-intelligence/

[QIN] Wikipedia Contributors (n.d.) https://en.wiktionary.org/wiki/%E7%90%B4

[QCD] Louppe,G. , Cho,K. , Becot,C., & Cranmer,K. (2017, February 02). QCD-Aware Recursive Neural Networks for Jet Physics. Retrieved April 20, 2017, from https://arxiv.org/pdf/1702.00748.pdf

[QUA] Quasi-empiricism in mathematics. (2017, March 14). Retrieved April 20, 2017, from https://en.wikipedia.org/wiki/Quasi-empiricism_in_mathematics

[RACI] Doglione,C. (2016, July 16) Understanding Responsibility Assignment Matrix (RACI Matrix). Retrieved April 20, 2017, from https://project-management.com/understanding-responsibility-assignment-matrix-raci-matrix/

[RAM] Ramirez, V. B. (2017, April 07). The 6 Ds of Tech Disruption: A Guide to the Digital Economy. Retrieved April 20, 2017, from https://singularityhub.com/2016/11/22/the-6-ds-of-tech-disruption-a-guide-to-the-digital-economy/

[RAS] Rasmus, A., Valpola, H., Honkala, M., Berglund, M., & Raiko, T. (2015, November 24). Semi-Supervised Learning with Ladder Networks. Retrieved June 30, 2017, from https://arxiv.org/abs/1507.02672v2

[RAT] Rathi, A., Zachariadis, M., & Paroutis, S. (2014, February 20). WhatsApp bought for $19 billion, what do its employees get? Retrieved April 20, 2017, from http://theconversation.com/whatsapp-bought-for-19-billion-what-do-its-employees-get-23496

[RAV] Ravi, S., & Larochelle, H. (2017). Optimization as a model for few-shot learning. In International Conference on Learning Representations (Vol. 1, No. 2, p. 6).

[RAZ] Wikipedia Contributors (n.d.). Occam's razor. Retrieved April 20, 2017, from https://en.wikipedia.org/wiki/Occam%27s_razor

[RBO] Carberry,S. (2015, June 20). New Robocalls Restriction May Affect 2016 Political Campaigns. Retrieved April 20, 2017, from https://news.vice.com/article/new-robocalls-restriction-may-affect-2016-political-campaigns

[REA] Real, E., Moore, S., Selle, A., Saxena, S., Suematsu, Y. L., Le, Q., & Kurakin, A. (2017). Large-Scale Evolution of Image Classifiers. arXiv preprint arXiv:1703.01041.

[REE] Reeves, M. K., & Levin, S. (2017, March 14). Building a Resilient Business Inspired by Biology. Retrieved April 20, 2017, from https://blogs.scientificamerican.com/guest-blog/building-a-resilient-business-inspired-by-biology/

[REG] Marketers Reform Wiki Contributors. (n.d.). Automated Trading Regulation .Retrieved April 20, 2017, from http://www.marketsreformwiki.com/mktreformwiki/index.php/Automated_Trading_Regulation

[RFR] Requests for Research. (n.d.). Retrieved April 20, 2017, from https://openai.com/requests-for-research/

[RID] Rid, T. (2016, October 24). How Russia Pulled Off the Biggest Election Hack in U.S. History. Retrieved April 20, 2017, from http://www.esquire.com/news-politics/a49791/russian-dnc-emails-hacked/

[RMA] Manifesto. (n.d.). Retrieved April 20, 2017, from http://www.responsive.org/manifesto

[ROK] Zastrow, M. (2016, March 18) South Korea trumpets $860-million AI fund after AlphaGo 'shock'. Retrieved April 20, 2017, from http://www.nature.com/news/south-korea-trumpets-860-million-ai-fund-after-alphago-shock-1.19595

[ROS] Rose-Stockwell, T. (2016, November 11). How We Broke Democracy (But Not in the Way You Think). Retrieved April 20, 2017, from https://medium.com/@tobiasrose/empathy-to-democracy-b7f04ab57eee#.k82dvd4uq

[ROT] Rotman, D. (2016, September 01). How Technology Is Destroying Jobs. Retrieved April 20, 2017, from https://www.technologyreview.com/s/515926/how-technology-is-destroying-jobs/

[RUS2] Rusu, A. A., Vecerik, M., Rothörl, T., Heess, N., Pascanu, R., & Hadsell, R. (2016). Sim-to-real robot learning from pixels with progressive nets. arXiv preprint arXiv:1610.04286.

[RUSS] Klimentyev, M. (2017, August 09). For Superpowers, Artificial Intelligence Fuels New Global Arms Race. Retrieved April 20, 2017, from https://www.wired.com/story/for-superpowers-artificial-intelligence-fuels-new-global-arms-race/

[RUS] Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., ... & Hadsell, R. (2016). Progressive neural networks. arXiv preprint arXiv:1606.04671.

[SAI] Sainsbury, D. (2013, April 3). Dual Process Theory in the Operating Theatre. Retrieved April 20, 2017, from http://davesainsbury.com/dual-process-theory-in-the-operating-theatre/

[SAL] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016, June 10). Improved Techniques for Training GANs. Retrieved July 01, 2017, from https://arxiv.org/abs/1606.03498

[SALT] Saltiel, D. (2015, June 7) State of Hyperparameter Selection. Retrieved April 20, 2017, from https://startup.ml/blog/hyperparam

[SAO] Saon, G., Sercu, T., Rennie, S. (2016). The IBM 2016 English Conversational Telephone Speech Recognition System. Retrieved April 20, 2017, from arXiv:1604.08242.

[SAP] Robert Sapolsky - Thinking about emergence and chaos. (2016, June 18). Retrieved April 20, 2017, from https://youtu.be/lQVTWHOqvWU

[SAX] Saxena, S., & Verbeek, J. (2016). Convolutional neural fabrics. In Advances In Neural Information Processing Systems (pp. 4053-4061).

[SCE] Schulte, E. (2012, November 08). How To Sell A $1 Snow Globe For $59: The Real ROI Of Brand Storytelling. Retrieved April 20, 2017, from https://www.fastcompany.com/3002804/how-sell-1-snow-globe-59-real-roi-brand-storytelling

[SCH] Schrage, M. (2016, December 30). Instead of Optimizing Processes, Reimagine Them as Platforms. Retrieved April 20, 2017, from https://hbr.org/2016/12/instead-of-optimizing-processes-reimagine-them-as-platforms

[SCH] Schuster, M., Johnson, M., Thorat, N. (2016, November 22). Zero-Shot Translation with Google. Retrieved April 20, 2017, from http://research.googleblog.com/2016/11/zero-shot-translation-with-googles.html

[SCU] Sculley, D., Phillips, T., Ebner, D., Chaudhary, V., & Young, M. (2014). Machine learning: The high-interest credit card of technical debt.

[SEM] Balduzzi,D. (2015,September 29). Semantics, Representations and Grammars for Deep Learning. Retrieved April 20, 2017, from https://arxiv.org/pdf/1509.08627.pdf

[SER] Nussbaum, J. (2016, October 25). Data Sets Are The New Server Rooms. Retrieved April 20, 2017, from https://hackernoon.com/data-sets-are-the-new-server-rooms-40fdb5aed6b0?_hsenc

[SEW] Seward, C. (2015, December 13). Optimizing Warehouse Operations with Machine Learning on GPUs. Retrieved July 01, 2017, from https://devblogs.nvidia.com/parallelforall/optimizing-warehouse-operations-machine-learning-gpus/

[SGAN] Zhang, H., Xu, T. , Li, H., Zhang, S., Wang, X., Huang, X., & Metaxas, D. (n.d.). StackGAN: Text to Photo-realistic Image Synthesis
with Stacked Generative Adversarial Networks. Retrieved April 20, 2017, from https://arxiv.org/pdf/1612.03242.pdf

[SIM] Rusu, A., Vecerik, M., Rothörl, T., Heess, N., Pascanu, R., & Hadsell, R. (2016, October 13) Sim-to-Real Robot Learning from Pixels with Progressive Nets. Retrieved April 20, 2017, from https://arxiv.org/abs/1610.04286

[SHA] Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., & Dean, J. (2017). Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. arXiv preprint arXiv:1701.06538.

[SHE] Shead, S. (2016, December 06). Apple is finally going to start publishing its AI research. Retrieved April 20, 2017, from http://www.businessinsider.com/apple-is-finally-going-to-start-publishing-its-artificial-intelligence-research-2016-12

[SHI] Shirky, C. (2003, February 8). Clay Shirky's Writings About the Internet. Retrieved April 20, 2017, from http://www.shirky.com/writings/powerlaw_weblog.html

[SHU] Shukla, H. J. (2016, February 08). Companies Are Reimagining Business Processes with Algorithms. Retrieved April 20, 2017, from https://hbr.org/2016/02/companies-are-reimagining-business-processes-with-algorithms

[SCI] Wang, A. (2017, August 23).Trump's science envoy quits in scathing letter with an embedded message: I-M-P-E-A-C-H. Retrieved April 20, 2017, from https://www.washingtonpost.com/news/speaking-of-science/wp/2017/08/23/trumps-science-envoy-quits-with-scathing-letter-with-an-embedded-message-i-m-p-e-a-c-h/?utm_term=.5b5d36a80012

[SIG] Sigg, S. (2008). Development of a novel context prediction algorithm and analysis of context prediction schemes. kassel university press GmbH.

[SIT] Kilpi, E. (2015, September 16). Contextual work. Retrieved April 20, 2017, from https://wtfeconomy.com/situated-work-b8cf60854d65

[SKY] Skype Translator – How it Works. (2014, December 15). Retrieved April 20, 2017, from http://blogs.skype.com/2014/12/15/skype-translator-how-it-works/

[SMT] Perez,C. (2017, April 06).The Next AI Milestone: Bridging the Semantic Gap. Retrieved April 20, 2017, from https://medium.com/intuitionmachine/the-first-rule-of-agi-is-bc8725d21530

[SOC] Society of Mind. (2017, June 29). Retrieved June 30, 2017, from https://en.wikipedia.org/wiki/Society_of_Mind

[SON] Song, J., Stewart, R., Zhao, S., & Ermon, S. (2017). ON THE LIMITS OF LEARNING REPRESENTATIONS WITH LABEL-BASED SUPERVISION. 1-5. Retrieved June 30, 2017.

[SPU] Evanoff, K. & Roberts, M. (2017, September 07). A Sputnik Moment for Artificial Intelligence Geopolitics Retrieved April 20, 2017, from https://www.cfr.org/blog/sputnik-moment-artificial-intelligence-geopolitics

[SPM] Wikipedia Contributors. (n.d.) CAN-SPAM Act of 2003.Retrieved April 20, 2017, from https://en.wikipedia.org/wiki/CAN-SPAM_Act_of_2003

[STA] Statt, N. (2016, September 08). Apple is trying to turn the iPhone into a DSLR using artificial intelligence. Retrieved April 27, 2017, from http://www.theverge.com/2016/9/8/12839838/apple-iphone-7-plus-ai-machine-learning-bokeh-photography

[SYN] Jaderberg, M. (2016, August 29). Decoupled Neural Interfaces Using Synthetic Gradients. Retrieved April 27, 2017, from https://deepmind.com/blog/decoupled-neural-networks-using-synthetic-gradients/

[SUK] Sukhbaatar, S. (n.d.). Learning Multiagent Communication with BackpropagationSainbayar Sukhbaatar, Arthur Szlam, Rob Fergus. Retrieved April 20, 2017, from http://cims.nyu.edu/~sainbar/commnet/

[SZO] Szoldra, P. (2016, November 16). NSA CHIEF: A nation-state made a 'conscious effort' to sway the US presidential election. Retrieved April 20, 2017, from http://www.businessinsider.com/nsa-chief-nation-state-swayed-president-election-2016-11

[TAF] Taft, D. K. (2017, June 26). Adobe Launches Sensei, AI for Digital Experiences. Retrieved July 01, 2017, from http://www.eweek.com/development/adobe-launches-sensei-ai-for-digital-experiences

[TAI] Taigman, Y., Polyak, A., & Wolf, L. (2016). Unsupervised Cross-Domain Image Generation. arXiv preprint arXiv:1611.02200.

[TDV] Tech Academy. (n.d.). Retrieved April 20, 2017, from https://www.davinciinstitute.com/

[TRP] Mervis, J. (2017, July 11). Trump's White House science office still small and waiting for leadership. Retrieved April 20, 2017, from http://www.sciencemag.org/news/2017/07/trump-s-white-house-science-office-still-small-and-waiting-leadership

[TEC] Lomas, N. (2016, June 24). Prisma uses AI to turn your photos into graphic novel fodder double quick. Retrieved April 20, 2017, from https://techcrunch.com/2016/06/24/prisma-uses-ai-to-turn-your-photos-into-graphic-novel-fodder-double-quick/

[TED] T. (2010, October 13). TEDxUofM - John Holland - Building Blocks and Innovation. Retrieved April 20, 2017, from https://www.youtube.com/watch?v=nzHVGd22vak&feature=youtu.be

[TFE] Cheng, H.T., Haque, Z., Hong, L., Ispir M., Mewald, C…& Xie, J. (2017, August 8). TensorFlow Estimators: Managing Simplicity vs. Flexibility in High-Level Machine Learning Frameworks. Retrieved April 20, 2017 from https://arxiv.org/abs/1708.02637

[TFS] Guadarrama, S., & Silberman, N. (2017, October 03). "tensor flow" GitHub Repository. Retrieved April 20, 2017 from https://github.com/tensorflow/tensorflow/tree/master/tensorflow/contrib/slim

[TFSF] Tensor Flow™.(2017, August 04). Tensor Flow Serving: Introduction.Retrieved April 20, 2017 from https://www.tensorflow.org/serving/

[TFX] Baylor, D., Breck, E., Cheng , H,T.,Fiedel, N., Foo, C.Y… & Zinkevich, M. (2017, August 13). Retrieved April 20, 2017, from http://www.kdd.org/kdd2017/papers/view/tfx-a-tensorflow-based-production-scale-machine-learning-platform

[THE] Theory of multiple intelligences. (2017, June 09). Retrieved June 30, 2017, from https://en.wikipedia.org/wiki/Theory_of_multiple_intelligences

[TMA] Turn Memories into Art. (n.d.). Retrieved April 21, 2017, from http://prisma-ai.com/

[TOW] Toward Machines that Improve with Experience. (2017, March 16). Retrieved April 20, 2017, from http://www.darpa.mil/news-events/2017-03-16

[TPU] Perez,C. (2017, April 06). Google's AI Processor's (TPU) Heart Throbbing Inspiration. Retrieved April 20, 2017, from https://medium.com/intuitionmachine/googles-ai-processor-is-inspired-by-the-heart-d0f01b72defe

[TRN] Bradbury,J., Britz,D., & Kummerfeld,J. ()Peeking into the neural network architecture used for Google's Neural Machine Translation. Retrieved April 20, 2017, from http://smerity.com/articles/2016/google_nmt_arch.html

[TUR] Turck, M. (2016, September 29). Building an AI Startup: Realities & Tactics. Retrieved April 20, 2017, from http://mattturck.com/building-an-ai-startup/

[UIML] Lovejoy, J. & Holbrook, J. (2017, July 9). Human-Centered Machine Learning. Retrieved April 20, 2017, from https://medium.com/google-design/human-centered-machine-learning-a770d10562cd

[UNP] Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. (n.d.). Retrieved June 30, 2017, from https://junyanz.github.io/CycleGAN/

[USI] Using Sketches to Search for Products Online (2016, July 13) Retrieved April 20, 2017, from https://news.developer.nvidia.com/using-sketches-to-search-for-products-online

[VIN] Vincent, J. (2016, July 05). Amazon's latest robot champion uses deep learning to stock shelves. Retrieved April 20, 2017, from http://www.theverge.com/2016/7/5/12095788/amazon-picking-robot-challenge-2016

[VIN] Vinyals, O., Blundell, C., Lillicrap, T., & Wierstra, D. (2016). Matching networks for one shot learning. In Advances in Neural Information Processing Systems (pp. 3630-3638).

[VIR] Wikipedia Contributors (n.d.) Seven virtues. Retrieved April 20, 2017, from https://en.wikipedia.org/wiki/Seven_virtues

[VIS] Conner-Simons,A. & Gordon,R. (2016, June 21). Teaching machines to predict the future. Retrieved April 20, 2017, from http://www.csail.mit.edu/teaching_machines_to_predict_the_future

[VOS] Voss, P. (2017, April 15). Cognitive Architectures – Intuition Machine – Medium. Retrieved April 20, 2017, from https://medium.com/intuitionmachine/cognitive-architectures-ea18127a4d1d

[WAN] Wang, J. X., Kurth-Nelson, Z., Tirumala, D., Soyer, H., Leibo, J. Z., Munos, R., ... & Botvinick, M. (2016). Learning to reinforcement learn. arXiv preprint arXiv:1611.05763.

[WAV] WaveNet: A Generative Model for Raw Audio. (n.d.). Retrieved April 20, 2017, from https://deepmind.com/blog/wavenet-generative-model-raw-audio/

[WEI] C, W. (2015, October 19). How We Use Deep Learning to Classify Business Photos at Yelp. Retrieved July 01, 2017, from https://engineeringblog.yelp.com/2015/10/how-we-use-deep-learning-to-classify-business-photos-at-yelp.html

[WEL] Weller, C. (2017, January 4). The largest internet company in 2030? This prediction will probably surprise you. Retrieved April 20, 2017, from https://www.weforum.org/agenda/2017/01/the-largest-internet-company-in-2030-this-prediction-will-probably-surprise-you/

[WEW] Weisstein, E. W. (2017, May 21). Universality. Retrieved April 30, 2017, from http://mathworld.wolfram.com/Universality.html

[WIKI2] Halting problem. (2017, April 10). Retrieved April 20, 2017, from https://en.wikipedia.org/wiki/Halting_problem

[WIKI] The Unreasonable Effectiveness of Mathematics in the Natural Sciences. (2017, April 11). Retrieved April 20, 2017, from https://en.wikipedia.org/wiki/The_Unreasonable_Effectiveness_of_Mathematics_in_the_Natural_Sciences

[XIE] Xie, L., & Yuille, A. (2017). Genetic CNN. arXiv preprint arXiv:1703.01513.

[YAN] C. (2016, November 18). Retrieved June 30, 2017, from https://www.youtube.com/watch?v=IbjF5VjniVE

[ZAS] Zastrow, M. (2016, March 15). How victory for Google's Go AI is stoking fear in South Korea. Retrieved April 27, 2017, from https://www.newscientist.com/article/2080927-how-victory-for-googles-go-ai-is-stoking-fear-in-south-korea/

[ZHA] Zhang, C., Bengio, S., Hardt, M., & Recht, B., (2016). Understanding deep learning requires rethinking generalization. Retrieved April 20, 2017, from arXiv:1611.03530.

[ZOP] Zoph, B., & Le, Q. V. (n.d.). NEURAL ARCHITECTURE SEARCH WITH REINFORCEMENT LEARNING. International Conference on Learning Representations. Retrieved April 21, 2017.

[ZXL] Zhang, H., Xu, T., Li, H., Zhang, S., Huang, X., Wang, X., & Metaxas, D. (2016). StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks. arXiv preprint arXiv:1612.03242.

A

B

C

D

E

F

G

R

S

T

U

ABOUT THE AUTHOR

Carlos E. Perez is the founder of Intuition Machine Inc. He has been a professional software architect and developer focused on developing software systems from concept to production, since 1993. Mr. Perez has 24 years experience developing software. Since 2000, he has primarily held technical architect roles in software development. Mr. Perez holds a Master's Degree in Computer Science from the University of Massachusetts. He holds U.S. patents covering Rule Bases, Social Web Filtering and Mobile Parental control.