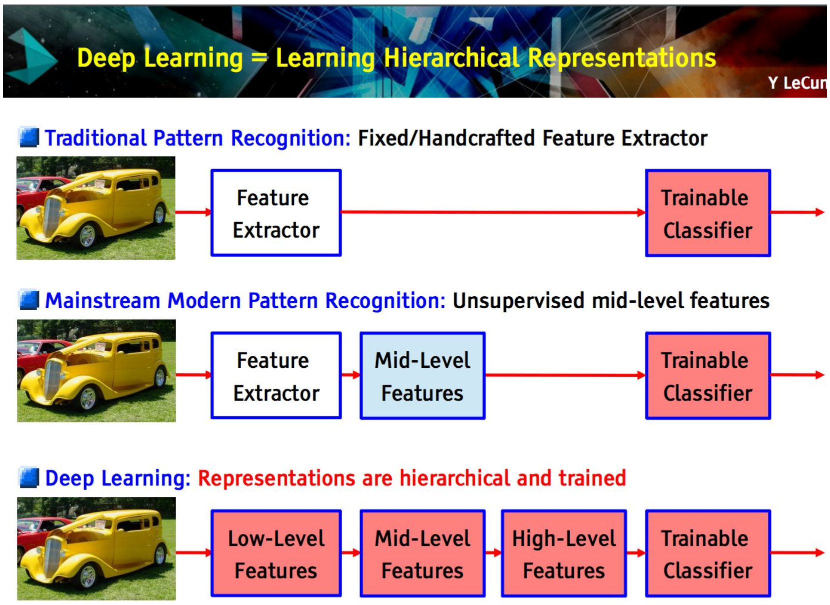
Musings on Deep Learning

Colin McDonnell

1 Introduction

Deep learning has seen a massive surge in interest recently as some breakthrough results and the advent of GPU computing made it competitive with more traditional machine learning approaches. Driven and enabled by these hardware capabilities, a new layer-centric abstraction is being developed with impressive initial results. Though the technical capabilities of the field are exploding, the understanding of its potential is not. Largely its success has been observed in the context of classic AI problems: image classification, natural language processing, recommendation algorithms. However this is akin to early television programs that merely showed a live stream of a radio broadcaster speaking into a microphone; it doesn't demonstrate the power of the medium of deep learning. Even AI pioneer Yann LeCun has fallen into the trap of incremental thinking, despite all his technical achievements. See Figure 1 for a slide of a presentation he recently gave at CVPR 2015 in early June.



Deep Learning = Learning Hierarchical Representations

Y LeCun

Traditional Pattern Recognition: Fixed/Handcrafted Feature Extractor

Feature Extractor → Trainable Classifier →

Mainstream Modern Pattern Recognition: Unsupervised mid-level features

Feature Extractor → Mid-Level Features → Trainable Classifier →

Deep Learning: Representations are hierarchical and trained

Low-Level Features → Mid-Level Features → High-Level Features → Trainable Classifier →

Figure 1: Slide from Yann LeCun's keynote at CVPR 2015.

Note that LeCun has forced deep learning into the context of "old-school AI". This slide presents deep learning as mere "automated feature detection" with integrated classifiers, when it is so much more. This paper explores the space of possibilities and presents some mental models for how to think about deep learning and its applications.

2 An Initial Point

I want to make one particular point as early as possible in this analysis: the limitations of deep learning are not equivalent to the limitations of present-day deep learning. This is a pretty obvious logical fallacy, though one that has tripped people up throughout the history of human progress. Everyone loves a contrarian and no one wants to look naive, so there is a tendency to be guarded or even hostile towards new exciting technologies. When I took an introductory AI class in Fall 2013, my professor, Patrick Winston, projected a picture of a group of students eating cake at an event on the screen. He declared that no system in the near future is likely to be capable of describing what is happening in the scene. The next year a photo captioning system was presented by Google and Stanford. This past year, my roommate is taking his class. Apparently that picture is gone and is replaced by two pictures: one of a person drinking from a soda bottle and one of a cat lapping from a bowl. He declared that no system in the near future is likely to be capable of describing what shared activity happening in both scenes.

These strong unsubstantiated opinions are widespread, even among very clever people like Patrick Winston. In the history of deep learning, opponents have vigorously declared the technology incapable of scaling beyond single-layer nets, operating on temporal data, learning hierarchical representations, achieving state of the art in activity recognition and image classification, displaying robustness to translation and rotation, and incorporating persistent memory into its computation. All of these have been proven wrong, as will the current flavor-of-the-month misconceptions surrounding its ability to achieve state of the art object detection, unsupervised learning, and variable-size representations. The only true limitations on deep learning in the general case (arbitrary size, connectivity, and activation function) is its inability to compute uncomputable functions, which I for one find forgivable.

3 History and Technical Overview

Now that that is thoroughly addressed, a brief technical and historical overview of deep learning will likely be valuable. The history of deep learning is a tale of dimensional proliferation -- each new generation is some generalization of the preceding status quo. The original “perceptron” is what we would call a single neuron today: a set of weighted inputs that are processed according to a function we’ll call the “neural function”. The quintessential neural function is a summation of the inputs paired with a nonlinear thresholding function (frequently the Heaviside step function). However, this function can take any form whatsoever, and in fact much of the interesting behavior of neural networks is achieved through modifications of this function. The perceptron is trained through manipulation of the weights and modifications to the neural function (classically, a shift of the thresholding quantity).

The single perceptron was quickly generalized to an entire line or plane of neurons, each with a set of weighted inputs, that “vote” on the proper classification of an instance. It was these single layer perceptrons are the titular characters in Minsky and Papert’s fateful book *Perceptrons,* which proved the inability of a single-layer neural network to compute an XOR function. This book is frequently credited (discredited?) with starting the AI winter that spanned the the 70s and 80s.

This problem was quickly rectified by adding many layers in series, resulting in so-called multi-layer perceptrons (MLPs) capable of computing all Boolean operators. MLPs are most frequently trained with the backpropagation algorithm, which was originally developed to train support vector machines but found wide utility for artificial neural networks. The existence of multiple layers enabled a vast new set of hyperparameters for researchers to fiddle with -- the number of neurons in each layer, the pattern of connections between successive layers, the number of layers in the complete network, the threshold function to use in the various layers (which may or may not be stochastic), and whether to have “parallel layers” that compute different kernel functions. Different regimes of these hyperparameters have been of sufficient interest to researchers to merit a name of their own. These include convolutional layers, loss layers, ReLU layers, FC layers, dropout, weight decay, and various forms of pooling.

These various parameters and hyperparameters are sufficiently general to describe nearly any feedforward neural network. Unfortunately feedforward neural networks are not optimal for processing time series data of arbitrary duration. For that we need to consider a time dimension, which is manifested in neural networks in the form of *feedback.* A neural network with loops is called recurrent, and many types of recurrent networks have been explored. As an example of how feedback can enable a neural “memory”, consider Figure 2 which depicts the long short term memory concept.

Just by modifying the neural functions of a four-layer neural network and adding a loop, we’ve created a neural flip-flop, the fundamental building block of arbitrarily complex memory circuits. There are many types of recurrent neural networks being studied currently, notable the “Neural Turing Machine” published by DeepMind and echo state networks/liquid state machines, and bidirectional associative memory. The low-level implementation details are irrelevant to this analysis however -- the relevant information is that deep learning systems are capable of memory storage and retrieval and all cognitive activities enabled thereby.

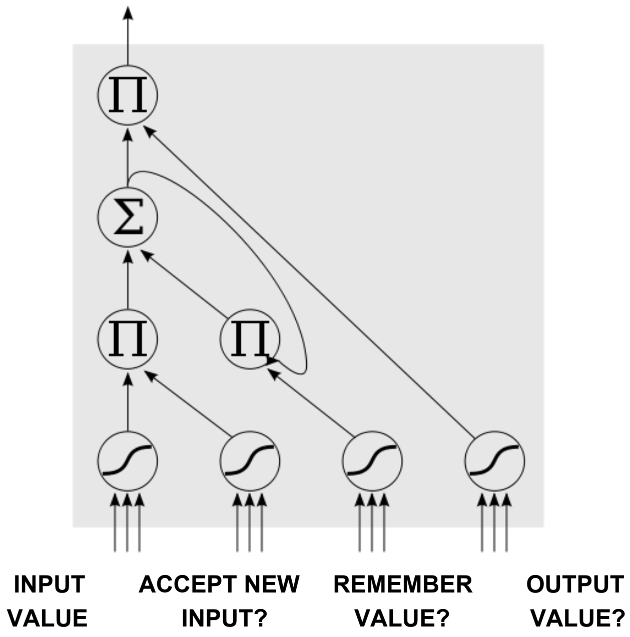


Figure 2: A diagram of a long short term memory unit.

The parameters used for training a neural network are typically input weights and modifications to a neural function such as a thresholding gate. In addition to these, there are hyperparameters relating to the structure of the network: the number and size of layers, the connection patterns, etc. An area of research that has yet to be explored deeply is the concept of metahyperparameters: a set of rules by which a network can change its own hyperparameters in response to the training data it is receiving. Such a network could dynamically adopt the structure and connectivity best suited to the data it is trying to classify. Additionally, learning can be encoded in the creation of new neural connections, as opposed to the weights of pre-determined connections, which some have argued is important for approaching the performance of the human brain. The idea of metahyperparameters is an exciting frontier which, as with all the changes before, generalizes the status quo along a totally new dimension. As we consider the present and future applications of deep learning, we assume a level of robustness and flexibility that is currently not achievable but which we expect will soon emerge as a result of these structural modification techniques.

4 Ways To Think About Deep Learning

That concludes a brief and necessary technical and historical overview of deep learning. Now we can change levels of abstraction completely. We consider a number of mental models that can help researchers, engineers, entrepreneurs, and laypeople find opportunities to apply deep learning. Each model we consider is progressively more abstract, more general, and more powerful.

4.1 deep learning as feature extractor

The first model is predominant in the minds of most researchers, as we've alluded to before. This model sees deep learning in the context of artificial intelligence research: raw data goes in, relevant patterns and quantities are identified, and some desired result come out. The network is trained on a set of training data, then real-life examples can be analyzed with the learned weights. The examples of this are well characterized: image classification, speech-to-text, etc. There is little more to add here, as this conception of deep learning is already prevalent.

4.2 deep learning as problem solver

The recent DeepDream project from DeepMind suggests another completely different use modality for deep learning. In this project, DeepMind took a neural network that had been trained on a corpus of images, passed in a input image consisting of pure noise, then modified the image specifically to increase the strength of a given classification. The ultimate results of this are images with features that have been morphed to look more like the classifier object in question. See Figure 3 for an example in which an image of spaghetti was iteratively modified to max out a neural net trained on pictures of dogs. The result ruins spaghetti forever.

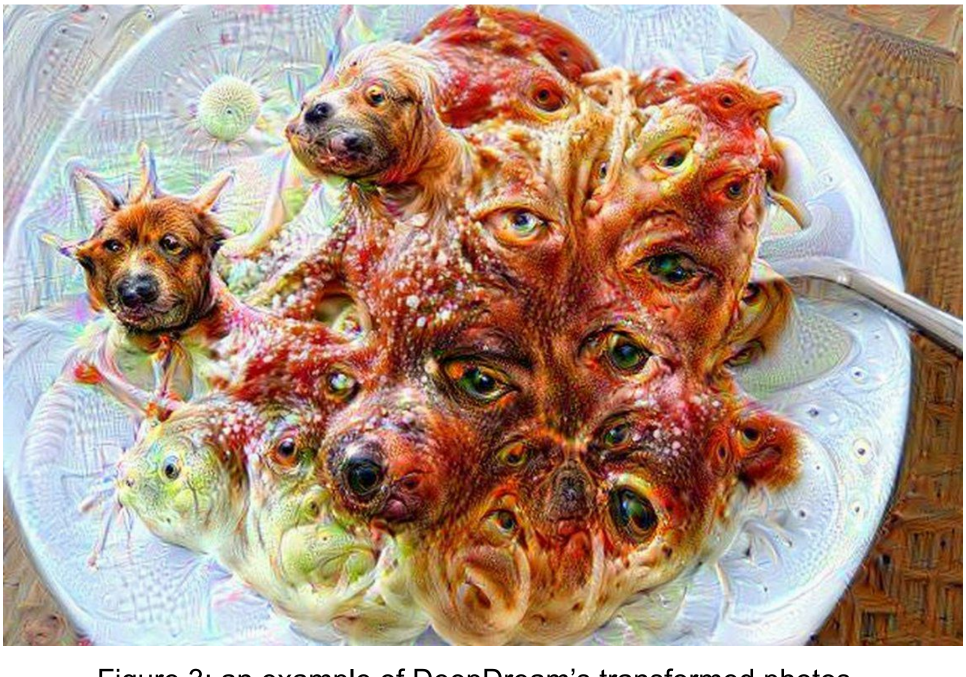


Figure 3: an example of DeepDream's transformed photos.

The new use modality is the application of deep learning as a candidate solution generator. The process used by DeepDream to modify the input image to increase a given fitness metric is akin to that used by an evolutionary algorithm. Before deep learning, the fitness function used in evolutionary or genetic algorithms had to be hard-coded. Now, a number of training examples can be used to teach a neural network how to evaluate a set of examples, and the resulting set of weights encodes the desired fitness function. This approach can be used to learn fitness functions that are hard to quantify or high-dimensional: the efficiency of an antenna, the binding strength of an enzyme, the quality of a movie script, and perhaps even the beauty or elegance of a designed product.

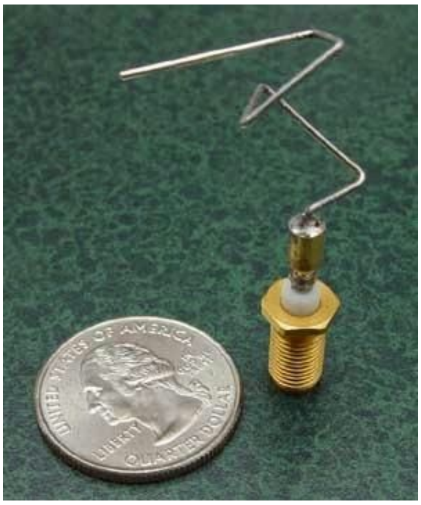


FIGURE 4: The 2006 NASA ST5 spacecraft antenna, a topology discovered by an evolutionary algorithm to have the best radiation pattern.

There are many successful examples of designs -- both hardware and software -- that are the product of an evolutionary algorithm. The best known example is the antenna design shown in Figure 4, developed by NASA for the ST5 spacecraft in 2005. A decade prior, in 1996, Adrian Thomson published a thesis describing a tone discriminator circuit consisting of only 40 logic gates, designed by an evolutionary algorithm. To quote damninteresting.com:

> “Dr. Thompson peered inside his perfect offspring to gain insight into its methods, but what he found inside was baffling. The plucky chip was utilizing only thirty-seven of its one hundred logic gates, and most of them were arranged in a curious collection of feedback loops. Five individual logic cells were functionally disconnected from the rest— with no pathways that would allow them to influence the output— yet when the researcher disabled any one of them the chip lost its ability to discriminate the tones. Furthermore, the final program did not work reliably when it was loaded onto other FPGAs of the same type.”

Needless to say, evolutionary algorithms are both capable of coming up with some impressive solutions. Expanding the scope of problems that can be subjected to these methods is potentially even more damn interesting.

Similarly, recent neuromorphic robotic control systems let robots discover ways to activate their actuators to achieve goals (say walking) while minimizing certain quantities (such as fall-induced acceleration spikes). These systems are a degenerate case of deep learning; it is a simple matter to take the telemetry data from a set of trial runs to train a neural-network-augmented robot how to move using well-known fall-avoidance and goal-achievement heuristics. This represents a completely new approach to robotics. Today the cutting edge of robotics research deals with underactuated dynamics models, however with deep learning researchers can start considering “overactuated robotics”: robots sporting an excess of redundant multi-DOF actuators and capable of deep learning a unique control algorithm adapted to the quirks of its actuators and the variation of its mechanical construction (just like humans).

4.3 deep learning as data transformation

The previous two mental models of deep learning are both derived from the context of the artificial intelligence research world. The first sees deep learning as a robust way of extracting relevant features from a data set. The second uses those extracted features as a fitness function to generate and evaluate candidate solutions to underconstrained problems. However neural networks can do yet more interesting things.

The classic problems of artificial intelligence have very well-formed inputs and outputs. In image classification, the algorithm takes in a matrix of pixel values and outputs a string designating the label of the image. In speech processing, the input is a sampled audio signal and the output is a synchronized string of text. However, deep learning can -- and indeed, should -- be used to solve far messier problems. The input can be heterogeneous or jumbled, comprise multiple channels, or vary in bandwidth over time. The desired output can vary stochastically or be poorly or ambiguously defined, underconstrained, or not defined. The characteristic robustness of deep learning is exactly what qualifies is to deal with these messy problems, but as of yet it has yet to break fully out of the context of well-defined traditional AI problems. In Figure 5 we break down the space of applications according to the “messiness” of the inputs and outputs.

	well-formed output	messy output
well-formed input	Includes most classic AI problems (image classification, NLP), specific well-defined algorithms, DSP, useful application-	Unsupervised, broad search for statistical regularities in a data set, selecting for features with high SNR. For example, extracting

	specific computations (Rayleigh-Sommerfeld, for example).	heart rate from video using Gaussian color magnification. “What can this data tell us?”
messy input	Synthesizing diverse data sources to perform a given well-defined task: for instance, predict the stock market. Lends itself to open-ended prediction tasks and optimization problems.	Largely unexplored at the moment. Includes automated scientific research, design thinking, narrative generation, creativity, artificial general intelligence...

Figure 5: breakdown of DL applications by messiness of inputs and outputs.

4.3.1 well formed input → well-formed output

This category includes the majority of deep learning applications at the moment, such as image recognition and NLP have generally well-defined inputs and outputs. Additionally, many known algorithms of interest can simply be learned by a deep learning system instead of needing to be implemented manually. As an example, consider the Rayleigh-Sommerfeld algorithm, a digital signal processing algorithm that reconstructs the location of pinholes from the diffraction pattern of an incident plane wave, as shown in Figure 6. This algorithm has recently been used in medical imaging to resolve micron-scale features with nothing but an LED, a pinhole, and a CCD sensor. It would be trivial to take a set of input images like that on the left of Figure 5, manually generate x-y coordinates of the center of each ripple, then train a neural net to learn the Rayleigh-Sommerfeld algorithm completely autonomously. Thus in some ways, deep learning finally achieves the holy grail of programming: a system that lets you specify what you want, not how to get it.

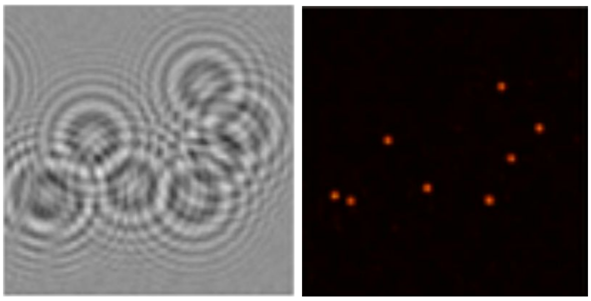


Figure 6: reconstruction of pinholes from diffraction pattern

by Rayleigh-Sommerfeld algorithm.

4.3.2 well formed input → messy output

This category is harder to wrap your head around. What sorts of problems have a well-defined input but no well-defined output? Is it even possible to train a neural network without a well-defined output for each input instance? Not in the traditional sense, but this category is far from empty. Consider a broad, unsupervised search for statistical regularities in a data set, selecting for features with high signal-to-noise ratio. This could turn up features that we would never have known to look for in the data but are nonetheless very prominent after some processing. For example, consider the recent research on Eulerian video magnification from MIT CSAIL. This research looks at interesting features that can be extracted from a video feed, yielding surprising results. From a standard 60fps video, the researchers were able to pick up the heart rate of people in the shot based on invisible color changes in their face, exaggerate small periodic motions, and even extract audio signals based on tiny vibrations in objects in the scene. These are the sorts of interesting features that might be revealed by a system in this category. In a “big data” world, the most valuable insights are the ones we don’t yet know to look for.

4.3.3 messy input → well-formed output

This category of system is simpler to understand. Any system that synthesizes diverse data sources to perform a given well-defined task is transforming a messy input into a well-defined output. For instance, consider a system that is capable of scraping data from the Internet -- stock prices and trading volumes, news articles, Twitter, public balance sheets, etc -- and is told to, as accurately as possible, predict the stock market. The input is heterogeneous, multi-channel, and time-varying in both content and structure, but the objective function -- alignment with the stock market -- remains constant and well-defined. As this example suggests, this category contains mostly open-ended prediction tasks and multi-dimensional dynamic optimization problems.

4.3.4 messy input → messy output

This category is very poorly defined and as such largely unexplored at the moment. It includes a vast array of interesting problems, of which a few are automated scientific research, design thinking, narrative generation, and artificial general intelligence. It will likely require self-modifying neural nets capable of autonomously seeking out new sources of information in the physical world or on the Internet.

4.4 deep learning as impedance matching

We consider a characterization of deep learning as an “impedance-matching” layer between domains. When connecting two circuits together for the purposes of power exchange, it is important to calibrate the components such that loss is minimized.

This process is known as impedance-matching, but the term metaphorically refers to any process of improving communication between two distinct contexts.

Many of the tasks deep learning has proven so capable of performing are *transformation tasks:* they involve the conversion from one medium, format, or domain to another. For instance, image classification is a conversion of an image to a human-readable string representation of the objects shown in that image. This concept is easily generalizable: deep learning being put to use to convert audio into text, images of faces into emotional states (facial expression analysis), and text in one language into another language. These examples were carefully chosen to demonstrate the ability of deep learning to transform from a digital to a human domain (facial expression analysis), from human to digital (speech to text), and even between human domains (translation).

Today, a huge number of jobs that consists mostly of transformation tasks: spreadsheets into reports, medical records into insurance premiums, symptoms into diagnoses, and many more. Currently these jobs are held by humans, who both require a salary and also tend to suck at these sorts of jobs. Humans have limited serial processing capabilities and attention spans, and are subject to all manner of performance-affecting exogenous factors. Moreover, the ideas that are capable of expressing or logically operating upon is frequently limited by the language they know, a phenomenon known as the Sapir-Whorf hypothesis. Deep learning provides a cheaper and potentially superior alternative for any task that can be characterized as an inter-domain transformation.

4.5 deep learning as a hidden layer

As neural networks started comprising multiple layers, the concept of a “hidden layer” emerged: a computational neuron layer somewhere between the input and output layers that was treated as a black box. As deep learning advances, it will effectively become a “hidden layer” in the network of society as more systems, infrastructure, and products communicate and coordinate. By this we mean deep learning will be a hidden but vital glue that lets us stitch together the digital and physical worlds by transforming the dumb sensors of today into *observers* that can reliably understand certain states of the world and react accordingly.

This capability completes a feedback loop that will dramatically change the way society is managed on a global scale. Here we explore a high-level conceptual design of a system that intelligently automates and augments individual lives, industries, governments, and perhaps the entire global village, resulting in a dramatic shift in how resources are allocated, goods and services are provided, and people spend their time. See Figure 6 for a diagram describing the components of such a system.

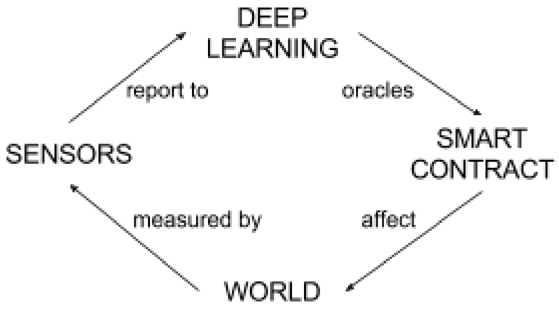


Figure 7: society-scale intelligent infrastructure with deep learning and smart contracts.

4.5.1 world measured by sensors

There is unlikely to be much confusion or objection on this point. Sensors are increasingly cheap, powerful, and perfusive in our pockets, on our wrists, in our homes and workplaces, on roads and bridges, and increasingly in our bodies. There are hundreds if not thousands of companies trying to extract useful insights from the data generated thereby, and an equal number of competing "standards" and proprietary communication and data storage schemes.

4.5.2 sensors report to deep learning systems

Increasingly, the features to be extracted from these sensors will become more abstract or high-level. This follows from an assumption on the part of the author that the Internet of Things industry must move beyond the "alarm clock triggers coffee pot" example in order to continue impressing venture capitalists. Such features include noninvasively detected heartbeat anomalies, a lack of yogurt in the fridge, a verbal confirmation of a meeting, or the sight of Grand Central Station in someone's augmented reality glasses.

4.5.3 deep learning oracles smart contracts

The concept of a smart contract has been recently re-popularized with the advent of scalable and secure computing platforms like the Bitcoin blockchain or Ethereum. Smart contracts are simply transactions that are triggered automatically when certain conditions are met. Conditions operate on data provided by oracles, trusted parties that report on some state of the world. Transaction here is meant in a very broad sense, essentially referring to any digitally-mediated action: an exchange of cash, a data transfer, an online purchase, an API call, whatever.

4.5.4 smart contract affects world

The triggering of a smart contract can affect change in the world in any number of ways. It can call an Uber, request a bridge inspection, display a particular banner on someone’s augmented reality headset, order a product online, request a service, etc.

5 Conclusion

The advent of deep learning in tandem with the invention of large, stable, trustless computing platforms like the Bitcoin blockchain finally allows intelligent automation and augmentation on a vast scale. In general, humans are bad at predicting the future and imagining the powerful effects of a new technology percolating into society. We encourage the reader to realize deep learning is bigger than the sum of its present capabilities, and deeply consider how deep learning will interact with the heterogenous universes of individual lives, new emerging technologies, and myriad of industries that drive us forward as a society.