

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/327212186>

Machine Learning is Changing the Rules.

Article · April 2018

CITATION

1

READS

662

1 author:



Peter Morgan

Deep Learning Partnership

8 PUBLICATIONS 157 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Quantum Computing [View project](#)



EXO-200 [View project](#)

O'REILLY®

Compliments of
ActiveState®

Machine Learning Is Changing the Rules

Ways Businesses Can Utilize
AI to Innovate



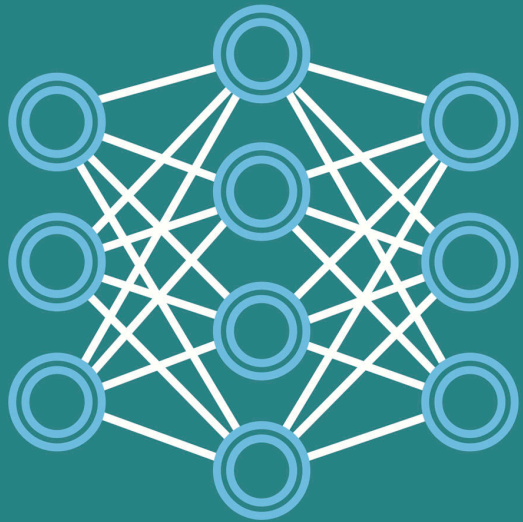
Peter Morgan

ActiveState®

FOR MACHINE LEARNING

See why 97%
of Fortune 1000
use our software

- Ramp-up your ML knowledge
- Try our ML-ready Python builds
- Fast-track ML into production



Learn more about ActiveState for ML >

Machine Learning Is Changing the Rules

*Ways Businesses Can
Utilize AI to Innovate*

Peter Morgan

Machine Learning Is Changing the Rules

by Peter Morgan

Copyright © 2018 O'Reilly Media. All rights reserved.

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (<http://oreilly.com/safari>). For more information, contact our corporate/institutional sales department: 800-998-9938 or corporate@oreilly.com.

Editors: Rachel Roumeliotis and Andy Oram

Production Editor: Justin Billing

Copyeditor: Octal Publishing, Inc.

Proofreader: Amanda Kersey

Interior Designer: David Futato

Cover Designer: Karen Montgomery

Illustrator: Rebecca Demarest

April 2018:

First Edition

Revision History for the First Edition

2018-03-28: First Release

The O'Reilly logo is a registered trademark of O'Reilly Media, Inc. *Machine Learning Is Changing the Rules*, the cover image, and related trade dress are trademarks of O'Reilly Media, Inc.

While the publisher and the author have used good faith efforts to ensure that the information and instructions contained in this work are accurate, the publisher and the author disclaim all responsibility for errors or omissions, including without limitation responsibility for damages resulting from the use of or reliance on this work. Use of the information and instructions contained in this work is at your own risk. If any code samples or other technology this work contains or describes is subject to open source licenses or the intellectual property rights of others, it is your responsibility to ensure that your use thereof complies with such licenses and/or rights.

This work is part of a collaboration between O'Reilly and Active State. See our *[statement of editorial independence](#)*.

978-1-492-03533-6

[LSI]

To Richard, Fernando, and Ilona; kindred spirits.

Table of Contents

Acknowledgments.....	vii
ActiveState: A Machine Learning Report.....	1
What Is a Disruptor, in Business Terms?	1
What Is Machine Learning?	2
Some Examples of Machine Learning (Industry Use Cases)	5
Healthcare	5
Finance	7
Transportation	12
Technology	14
Energy	17
Science	17
How Businesses Can Get Started in Machine Learning	19
Why Big Data Is the Foundation of Any Machine Learning Initiative	20
What Is a Data Scientist?	21
Automation of the Data Science Life Cycle	23
The Build-Versus-Buy Decision	24
Buying a Commercial-Off-the-Shelf Solution	24
Open Source Machine Learning Solutions	27
Additional Machine Learning Frameworks	28
Open Source Deep Learning Frameworks	29
Commercial Open Source	30
AI as a Service (Cloud Machine Learning)	31
Languages	35
Data Science Notebooks	35
Pros and Cons of Machine Learning Open Source Tools	35
Looking Ahead: Emerging Technologies	36
Conclusions: Start Investing in Machine Learning or Start Preparing to be Disrupted	36

Acknowledgments

First, my thanks go to O'Reilly for asking that I write this report and then supporting me all the way through to the end. To my friends and family for always being there, and for all the wonderful research scientists and engineers who make the field of artificial intelligence as exciting and engaging as it is. The rate of change we are seeing in this domain is truly breathtaking.

ActiveState: A Machine Learning Report

Machine learning has been garnering a lot of press lately, and for good reason. In this report, we look at those reasons, how machine learning is achieving the results it has been getting, and what your business can and should be doing so as not to be left behind by competitors that embrace this technology.

What Is a Disruptor, in Business Terms?

This term, whose use in business is attributed to Harvard Business School Professor Clayton Christensen, refers to any new technology that totally changes the rules and rewards governing a market. We have seen many of these throughout history. The major disruptions include agriculture, the industrial revolution, and the computer revolution. And now, one could argue, we are witnessing the biggest revolution (or market disruption) of all: the artificial intelligence revolution. The agricultural revolution enabled us to grow crops, store food, engage in trade, as well as build villages, towns, and eventually cities, and move on from our nomadic, hunter gatherer lifestyle. The industrial revolution replaced a lot of human and animal labor with machines and also enabled mass production of goods. Think of the steam engine and the car replacing horse transportation, and of machines in factories replacing human manual labor, such as weaving looms and the robots in car manufacturing plants. The digital revolution put a PC on every desk, with killer apps such as word processing, spreadsheets, and web browsers for accessing the internet. It also led to smartphones for business and consumer use and connectivity. Recall that such market disruptions initially replaced workers (not to mention horses) but new jobs were created, and mass unemployment was avoided.

It turns out that we are living in very interesting and unprecedented times, with the emergence of several new technologies including machine learning, block-chain technology, biotechnology, and quantum computing all on their own exponential growth curves. For more about emerging technologies and exponential

trends, see “Looking Ahead: Emerging Technologies” on page 36, as well as the references at the end of this report.^{1, 2, 3}

What Is Machine Learning?

What is machine learning, why is it so hot, and why does it have the ability to be a disruptor? “Machine learning” is the technology buzzword capturing a good deal of press of late—but is it warranted? We would argue yes; but first off, let’s define what it is and why it is so important. Rather than programming everything you want a computer to do using strict rule-based code, the machine learning process works from large datasets, with the machines learning from data, similar to how we humans and other biological brains process information. Given such a powerful paradigm shift, the potential for disruption is great indeed. Not only do computer professionals and business leaders need to learn how to design and deploy these new systems; they will also need to understand the impact this new technology will have on their businesses.

The three terms *machine learning*, *deep learning*, and *artificial intelligence* (AI) are often used interchangeably. What’s the difference? **Figure 1-1** illustrates the distinction between them. We can see that artificial intelligence covers all learning algorithms, including regression, classification, and clustering and cognitive tasks such as reasoning, planning, and navigation. In fact, the holy grail of AI is (and always has been) to build machines capable of doing *everything* a human being can do, and better. The brain, with its roughly 100 billion neurons and 4 billion years of evolution, is a pretty sophisticated and massively complex work of biological engineering, so perhaps we shouldn’t be too surprised that we haven’t yet managed to replicate all of its features in silicon. But we are making progress. This quest is known as *artificial general intelligence*, or AGI, and the ultimate goal is to design and build *artificial superintelligence*, or ASI.

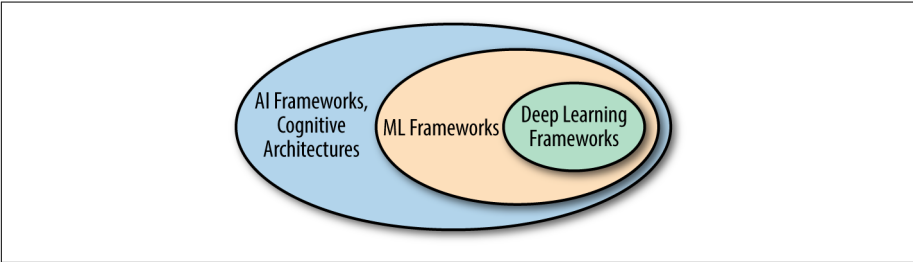


Figure 1-1. Comparing AI, machine learning, and deep learning

Inside the AI oval is machine learning with its wide variety of algorithms, including *support vector machines*, *K-means clustering*, *random forests*, and hundreds more that have been developed over the past several decades. In fact, machine learning is a branch of statistics whereby the algorithms learn from the data as it

is input into the system. Finally, we have deep learning, also known as *artificial neural networks* (ANNs) because these algorithms are modeled on how the brain processes data, although currently in a simplified framework.^{4, 5} The word *deep* refers to the layered networks of nodes that make up the architecture (see [Figure 1-2](#)) and that are sometimes referred to as *deep neural networks*, or DNNs. In practice, these DNNs can have hundreds of layers and billions of nodes. Computations occur at each node, calling for massively parallel processing. Some examples of DNN models are AlexNet, ResNet, Inception-v4, and VGG-19.⁶ DNNs now regularly outperform humans on difficult problems like face recognition and games such as Go.⁷

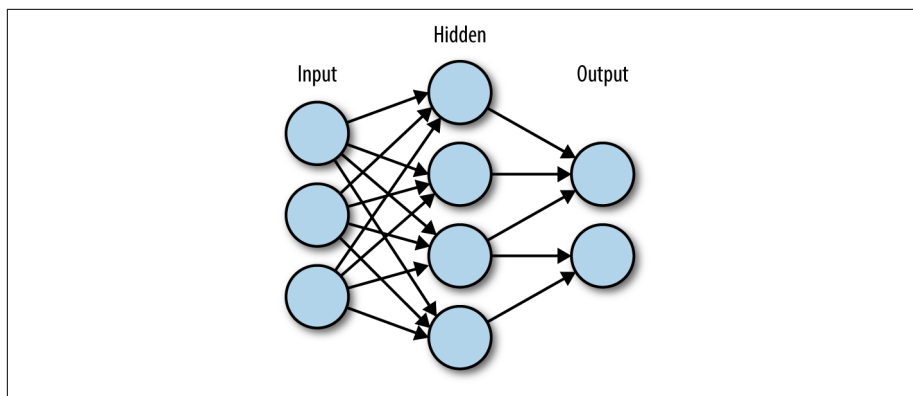


Figure 1-2. Schematic of an artificial neural network

Because these algorithms are oversimplifications of how the brain works, leading practitioners in the field, such as Geoff Hinton at the University of Toronto and Google, say that these current deep learning algorithms are too simple to get us to general intelligence and something a lot more like the brain is needed.⁸ One of the drawbacks is that these deep learning neural nets need large amounts of training data to gain the accuracy required, which translates to massive processing power. Specialized hardware such as graphics processing units (GPUs), field-programmable gate arrays (FPGAs), and application-specific integrated circuits (ASICs) are all being designed and built to optimize the calculations (basically very large matrix multiplications) needed for the deep learning processing. If you're curious about recent hardware developments in this area, check out [NVIDIA GPU](#), [Google TPU](#), and the [Graphcore IPU](#). A further brief discussion on hardware is given in “[Looking Ahead: Emerging Technologies](#)” on page 36, along with references.

That said, the reason that deep learning is receiving all this attention is because it is outperforming pretty much all other machine learning algorithms when it comes to classifying images⁹ (see [Figure 1-3](#)), language processing, and time-series data processing. And with advancements in hardware and algorithm opti-

mizations, the time to achieve this accuracy is also dropping exponentially. For example, a team at Facebook, along with other teams, recently announced that they had processed ImageNet, a well-known image dataset, to a world-ranking accuracy in under an hour,¹⁰ whereas four years ago we might have expected this kind of result to take around one month—a time frame not really suitable for business applications. This processing time is expected to drop to under a minute over the next few years, with continued improvements in hardware and algorithmic optimizations.

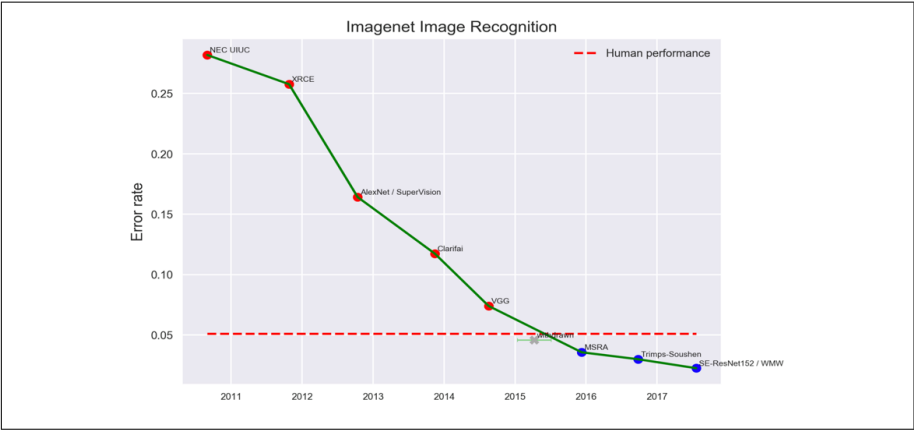


Figure 1-3. ImageNet error rate is now around 2.2%, less than half that of average humans

Figure 1-4 shows the increasing popularity in these technologies by search results. Finally, some machine learning and AI technical cheat sheets are available [here](#).

Now let’s take a look at how some companies are using machine learning to increase efficiency, innovate on new products and services, and boost profitability.

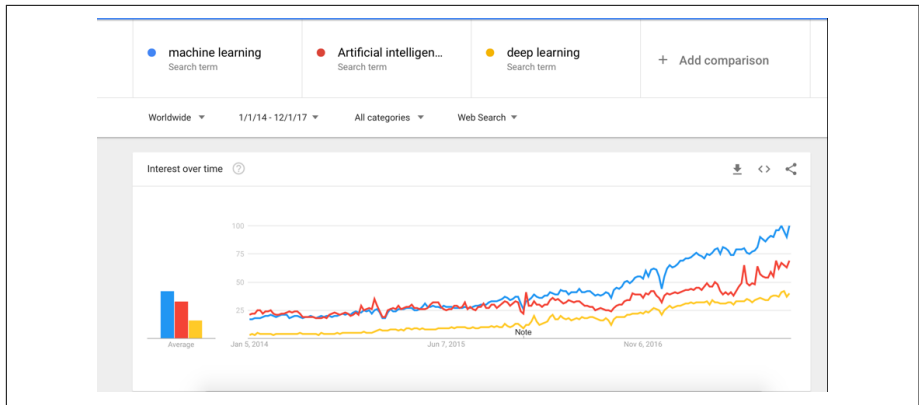


Figure 1-4. Search trends in deep learning, artificial intelligence, and machine learning using Google Trends

Some Examples of Machine Learning (Industry Use Cases)

If we think about it for a minute, by definition, every industry is going to be affected by the development and application of artificial intelligence algorithms. Intelligence injected into processes, products, and services will help business become more efficient, innovative, and profitable. Clearly, we don't have the space to talk about all business domains in this short report, so we have selected the following six sectors to provide an interesting cross-section of use cases. You'll learn how AI is being used in these various industry sectors today and which companies are successfully deploying AI, along with how and where. Here are the sectors we highlight:

- Healthcare
- Finance
- Transportation
- Technology—software and hardware systems
- Energy
- Science

Healthcare

AI is being used in various areas of healthcare, including the following:

- Genomics
- Drug discovery

- Cancer research
- Image scanning
- Surgery
- Longevity research
- Resource usage
- Robot carers

Let's take a look at genomics for some illustrative examples. Deep learning is being used extensively in genomics research and in the development of new biotech products. Few domains exploit so well the essential power of deep learning: its capability to ingest extremely large datasets and find patterns in that data. The human genome has approximately four billion base pairs, so this is a big dataset. Now think of analyzing thousands or millions of genomes, and looking at petabytes of data.

Fortunately, with the advent of deep learning-optimized hardware (GPU and ASIC) and cloud services (machine learning as a service, or MLaaS), deep learning algorithms can analyze these huge datasets in realistic timescales—days rather than the months required just a few years ago. This makes it time and cost effective to use this methodology in the field of genomics to understand the human genome. The results are exploited to fight cancer and other diseases such as Alzheimer's and Parkinson's as well as to accelerate drug discovery for myriad diseases (such as ALS) and mental health issues (such as schizophrenia) that afflict humans today.

DeepVariant is a **genomics framework** recently open sourced by Google Alphabet,¹¹ in conjunction with its healthcare company **Verily**. **The code is on GitHub**, with a license allowing anyone to download, use, and contribute to it. You can find genomics datasets on the web,^{12, 13} or, if you are a healthcare company, you might, of course, use your own.

Microsoft is also using AI to help **improve the accuracy of gene editing with CRISPR**. Several companies have been set up specifically to use machine learning to accelerate medical research and development. These include **BenevolentAI** and **Deep Genomics**.

Founded in 2013, BenevolentAI is based in London and is the largest private AI company in Europe. By applying AI to the **mass analysis** of vast amounts of scientific information, such as scientific papers, patents, clinical trials, data, and images, it is augmenting the insights of experienced scientists with the analytical tools they need to create usable and deep knowledge that dramatically speeds up scientific discovery. In the biotech industry, a new paper is published every 30 seconds; BenevolentAI applies AI algorithms to read and understand these papers. It then creates intelligent hypotheses about likely cures for various disea-

ses. The company has already made major breakthroughs, **including one related to ALS**. BenevolentAI also uses all of this data analysis to design and predict new molecules.

Deep Genomics is a Toronto-based company founded in 2015 by Brendan Frey. Frey was a researcher with Geoff Hinton and then professor of engineering and medicine at the University of Toronto before forming his company. Deep Genomics' founding belief is that the future of medicine will rely on artificial intelligence because biology is too complex for humans to understand. Deep Genomics is building a biologically accurate data- and AI-driven platform that supports geneticists, molecular biologists, and chemists in the development of therapies. For the company's Project Saturn, for example, researchers will use the platform to search across a vast space of more than 69 billion molecules with the goal of generating a library of 1,000 compounds that can be used to **manipulate cell biology and design therapies**.

Finally, it is worth noting that both **Google** and **Microsoft** cloud services offer genomics as a service (GaaS). Researchers can use these powerful platforms to analyze vast datasets, either public or proprietary. (For further information, read the **whitepaper on genomics** from the team at Google Cloud Platform [GCP]). Other papers describing machine learning techniques applied to genomics research are listed in the References section;^{14, 15} the interested reader is encouraged to view them.

Finance

Some areas of FinTech that employ AI include the following:

- Trading
- Investment
- Insurance
- Risk management
- Fraud detection
- Blockchain

Let's take a look at two aspects of finance for which AI is currently being deployed: blockchain and algorithmic trading.

AI and blockchain

Blockchain is a technology whose primary purpose is to decentralize and democratize products and services that run over public and private distributed computing systems such as the internet and company intranets, respectively. It is being hailed as internet 2.0 and as an incredible enabler of many things social, eco-

conomic, and political, including a fairer and more equitable distribution of resources and a huge enabler of innovation. With blockchain, a distributed ledger is replicated on thousands of computers around the world and is kept secure by powerful encryption algorithms.

Blockchain provides the opportunity to introduce new products and services, reduce costs of existing services, and significantly reduce transaction times, perhaps from days to seconds. Examples of blockchain applications include legal agreements (contracts), financial transactions, transportation infrastructure, accommodation (hotels, apartments, and smart locks), the energy grid, the Internet of Things (IoT), and supply-chain management.

There are also many blockchain-based products and services that haven't been thought of yet and are appearing almost daily as initial coin offerings (ICOs) or other ideas. ICOs are like initial public offerings (IPOs); however, with ICOs tokens are issued instead of stocks as a way of raising money. Instead of purchasing stocks, the public is given the opportunity to purchase tokens associated with a particular blockchain-based product or service. Evidence that blockchain is here to stay is provided by blockchain as a service (BaaS) offerings from both **IBM** and **Microsoft**. Blockchain's longevity is guaranteed also by the open source standards organizations Hyperledger, R3, and EEA, all of which have dozens of corporate members.

What happens when we begin to merge AI and the blockchain into a single, powerful integrated system? The combination gains power from blockchain's promise of near-frictionless value exchange and AI's ability to accelerate the analysis of massive amounts of data. The joining of the two is marking **the beginning of an entirely new paradigm**.

For instance, we can maximize security while services remain immutable by employing artificially intelligent agents that govern the chain. **State Street is doing just this by issuing blockchain-based indices**. Data is stored and made secure by using blockchain, and the bank uses AI to analyze the data while it remains secure. State Street reports that 64% of wealth and asset managers polled expected their firms to adopt blockchain in the next five years. IBM Watson is also merging blockchain with AI via the **Watson IoT group**. In this development, an artificially intelligent blockchain lets joint parties collectively agree on the state of the IoT device and make decisions on what to do based on language coded into a smart contract.

Finally, society is becoming increasingly reliant on data, especially with the advent of AI. However, a small handful of organizations with both massive data assets and AI capabilities have become powerful, giving them increasing control and ownership over commercial and personal interactions that poses a danger to a free and open society. We therefore need to think about unlocking data to achieve more equitable outcomes for both owners and users of that data, using a

thoughtful application of both technology and governance. Several new companies have started up that combine decentralized AI and blockchain technologies to do just this. Let's take a brief look at them here:

SingularityNET

SingularityNET enables AI-as-a-service (AIaaS) on a permissionless platform so that anyone can use AI services easily. The company provides a protocol for AI to AI communication, transaction, and market discovery. Soon, its robot Sophia's intelligence will run on the network, letting her learn from every other AI in the SingularityNET, and users will be able to communicate with her. SingularityNet recently raised \$36 million in about one minute in its recent ICO selling the AGI token.

Ocean Protocol

Ocean Protocol is a decentralized data exchange protocol that unlocks data for AI. Estimates show that a data economy worth \$2–3 trillion could be created if organizations and people had the tools to guarantee control, privacy, security, compliance, and pricing of data. Ocean Protocol provides the base layer for these tools using a set of powerful, state-of-the-art blockchain technologies. The cofounders also created the global decentralized database **Big-chainDB**. Exchanged as Ocean Token.

eHealth First

This is an IT platform for personalized health and longevity management whose stated aim is to help to prolong the user's life. It is based on blockchain, AI, and natural language processing (NLP). Using neural network algorithms, the platform will process the ever-growing body of publications in medical science allowing new scientific discoveries to be turned more quickly into treatments. Exchanged as EHF tokens.

Intuition Fabric

Provides democratized deep learning AI on the Ethereum blockchain. Although still very much in the design phase, the stated mission of this AI blockchain company is to distribute wealth and knowledge more equally throughout the world so that everyone makes a fair living and has opportunity for a decent quality of life.

OpenMined

The mission of the OpenMined community is to make privacy-preserving deep learning technology accessible both to consumers, who supply data, and to machine learning practitioners, who train models on that data. Given recent developments in cryptography (homomorphic encryption), AI-based products and services do not need a copy of a dataset in order to create value from it. Data provided could be anything from personal health information to social media posts. No tokens.

Synapse AI

A decentralized global data marketplace built on the blockchain. Users are paid for sharing their data and earn passive income by helping machines learn and become smarter. This can be considered a crowdsourcing of intelligence. Exchanged as Syn token.

DeepBrain

This is a chatbot based blockchain AI platform founded in Singapore. The DBC token is traded on a smart contract based on NEO.

Longgenesis

A collaboration between Bitfury and Insilico Medicine, in an attempt to solve two of the humanity's most pressing problems: ownership of personal data and longevity using AI and blockchain. Longgenesis life data marketplace and ecosystem is fueled by the LifePound token.

Algorithmic trading

Neural networks can process time-series data perfectly well, as witnessed in the way that humans and other animals process the streaming data incident on their senses from the external environment. So, it is not surprising that we can apply ANNs to financial data in order to make trading decisions. In technical terms, ANNs are a nonparametric approach to modeling time-series data, based on minimizing an entropy function.

Stock market prediction is usually considered as one of the most challenging issues among time-series predictions due to its noise and volatile features. How to accurately predict stock movement is still very much an open question. Of course, algorithmic trading has been blamed for past frightening spikes and drops, although they were quickly corrected. That's a good reason to search for better, more robust algorithms. In the literature, a recent trend in the machine learning and pattern recognition communities considers that a deep nonlinear topology should be applied to time-series prediction. An improvement over traditional machine learning models, DNNs can successfully model complex real-world data by extracting robust features that capture the relevant information and achieve even better performance than before.

In the paper "A deep learning framework for financial time-series using stacked autoencoders and long-short term memory," Bao et al.¹⁶ present a novel deep learning framework in which wavelet transforms (WT), stacked autoencoders (SAEs) and long short-term memory (LSTM) are combined for stock price forecasting. SAEs are the main part of the model and are used to learn the deep features of financial time-series in an unsupervised manner. WT are used to denoise the input financial time-series and then feed them into the deep learning framework. LSTMs are used to predict time-series when there are time steps with arbitrary size because LSTMs are well suited to learn from experience.

The authors apply their method to forecast the movements of each of six stock indices and check how well their model is in predicting stock-moving trends. Testing the model in various markets brings the opportunity to solve this problem and shows how robust the predictability of the model is. Their results show that the proposed model outperforms other similar models in both predictive accuracy and profitability performance, regardless of which stock index is chosen for examination.

In the paper “High-Frequency Trading Strategy Based on Deep Neural Networks,” Arevalo et al.¹⁷ use DNNs and Apple Inc. (AAPL) tick-by-tick transactions to build a high-frequency trading strategy that buys stock when the next predicted average price is above the last closing price, and sells stock in the reverse case. This strategy yields an 81% successful trade during the testing period.

The use of deep reinforcement learning (RL) algorithms in trading is examined in a recent blog post, “[Introduction to Learning to Trade with Reinforcement Learning](#)”, whose author has previously worked in the Google Brain team. Because RL agents are learning policies parameterized by neural networks, they can also learn to adapt to various market conditions by seeing patterns in historical data, given that they are trained over a long time horizon and have sufficient memory. This allows them to be much more robust to the effects of changing markets and to avoid the aforementioned flash crash scenarios. In fact, you can directly optimize the RL agents to become robust to changes in market conditions by putting appropriate penalties into the reward function.

Ding et al.¹⁸ combine a neural tensor network and a deep convolutional neural network (CNN) to extract events from news text and to predict short-term and long-term influences of events on stock price movements, respectively. Improvements in prediction accuracy, and therefore profitability, of 6% trading on the S&P 500 index were obtained.

In their paper, Dixon et al.¹⁹ describe the application of DNNs to predicting financial market movement directions. In particular, they describe the configuration and training approach and then demonstrate their application to backtesting a simple trading strategy over 43 different CME Commodity and FX future mid-prices at five-minute intervals. They found that DNNs have substantial predictive capabilities as classifiers if trained concurrently across several markets on labelled data.

Heaton et al.²⁰ provide a nice overview of deep learning algorithms in finance. Further aspects and worked examples of using deep neural networks in the algorithmic trading of various financial asset classes are covered in the blogs listed in the References section.^{21, 22, 23, 24}

In conclusion, deep learning presents a general framework for using large data-sets to optimize predictive performance. As such, deep learning frameworks are well suited to many problems in finance, both practical and theoretical. Due to their generality, it is unlikely that any theoretical models built from existing axiomatic foundations will be able to compete with the predictive performance of deep learning models. We can use deep neural networks to predict movements in financial asset classes, and they are more robust to sudden changes in market prices. They can also be used for **risk management** so as to avoid any trading-driven booms and busts.

Transportation

AI is being used in the various areas of transportation, including the following:

- Self-driving cars
- Route optimization
- Smart cities
- Flight
- Shipping

Let's take a look at self-driving cars. Self-driving vehicles are set to disrupt cities and transportation overall. Potentially, they can transform not only the way people and goods move around the world, but also patterns of employment, new transport potential for many populations, and the organization of urban environments.

Every year, **1.25 million people lose their lives on the world's roads**. Causes of death include speeding, alcohol, distractions, and drowsiness. Self-driving vehicles are expected to reduce this number significantly, by at least 99%. Not only could self-driving cars reduce the road toll each year, but time spent commuting could be time spent doing what one wants while the car handles all of the driving.

Driverless cars will enable new ride- and car-sharing services. New types of cars will be invented, resembling offices, living rooms, or hotel rooms on wheels. Travelers will simply order up the type of vehicle they want based on their destination and activities planned along the way.

Ultimately, self-driving vehicles will reshape the future of society. The self-driving car market is expected to rise rapidly to an **estimated \$20 billion by 2024, with a compound annual growth rate of around 26%**.

Machine learning algorithms are used to enable the vehicle to safely and intelligently navigate through the driving environment, predicting movements and avoiding collisions with objects such as people, animals, and other vehicles. Self-driving cars come equipped with various **external and internal sensors to track**

both the environment and the driver, respectively. Sensors include Lidar (light radar), radar, infrared, ultrasound, microphones, and cameras. To elaborate:

- Originating in the early 1960s, Lidar is a surveying method that measures distance to a target by illuminating that target with a pulsed laser light and measuring the reflected pulses with a sensor. Differences in laser return times and wavelengths can then be used to make digital 3D-representations of the target.
- Ultrasonic sensors track and measure positions of objects very close to the car, like curbs and sidewalks, as well as other cars when parking.

Information collected by these sensors is then used to calculate distances, speeds, and types of objects surrounding the vehicle, as well as to predict motion through time and space.

A GPS system is used for navigation and vehicle-to-vehicle communication. Finally, a powerful in-car computer, comprising GPU processors that are usually built into the trunk of the car, runs machine learning algorithms such as CNNs to identify and track objects and to navigate through the environment.²⁵ These algorithms are updated online as new and better software becomes available and, along with improvements in training data and hardware, are expected to take us to Level 5 (fully autonomous) self-driving cars.

NVIDIA presently has the largest market share of the self-driving car on-board processors with its DRIVE PX GPUs. The DRIVE PX Xavier processor, with more than seven billion transistors, is the most complex system on a chip (SoC) ever created, representing the work of more than 2,000 NVIDIA engineers over a 4-year period and an investment of \$2 billion in research and development. It is built around a custom 8-core CPU, a new 512-core Volta GPU, a new deep learning accelerator, computer vision accelerators, and new 8K HDR video processors. These on-board computers have 30 trillion operations per second (TOPS) of processing power while consuming just 30 watts of power, so it is like having a supercomputer in your car. The DRIVE PX Xavier is the first AI car supercomputer designed for fully autonomous Level 5 robotaxis and will be available Q1 2018.

The next generation platform from NVIDIA, the DRIVE PX Pegasus, delivers more than 320 TOPS and extends the DRIVE PX AI computing platform to handle Level 5 driverless vehicles. It will be available to NVIDIA automotive partners in the second half of 2018. Like a datacenter on wheels, NVIDIA DRIVE PX Pegasus will help make possible a new class of vehicles that can operate without a driver—fully autonomous vehicles without steering wheels, pedals, or mirrors, and interiors that feel like a living room or office. These vehicles will arrive on demand to whisk passengers safely to their destinations, bringing mobility to everyone, including the elderly and disabled. Millions of hours of lost time will

be recaptured by drivers as they work, play, eat, or sleep on their daily commutes. And countless lives will be saved by vehicles that are never fatigued, impaired, or distracted—increasing road safety, reducing congestion, and freeing up valuable land currently used for parking lots.

The AI performance and capabilities of the PX Pegasus platform are expected to ensure the reliability and safety of self-driving cars as well as autonomous trucking fleets. A unified architecture enables the same software algorithms, libraries, and tools that run in the datacenter to also perform inferencing in the car. A cloud-to-car approach enables cars to receive over-the-air updates to add new features and capabilities throughout the life of a vehicle. You can [find further details here](#) and [software libraries are available here](#).

Along with the incumbent car manufacturers such as GM, Ford, Mercedes, Volkswagen, and Toyota, a host of new companies have entered this market, including the likes of Waymo (spun out from Google in 2016), Tesla, Uber, Baidu, [NuTonomy](#), [Oxbotica](#), and [Aurora](#). AT CES 2018, [NVIDIA and Aurora announced that they are working together](#) to create a new Level 4 and Level 5 self-driving hardware platform.

[Waymo](#) currently drives [more than 25,000 autonomous miles each week](#), largely on complex city streets. That's on top of 2.5 billion simulated miles it drove just in 2016. By driving every day in different types of real-world conditions, Waymo's cars are taught to [navigate safely](#) through all kinds of situations. The company's vehicles have sensors and software that are designed to detect pedestrians, cyclists, vehicles, road work, and more from a distance of up to two football fields away in all directions. Waymo's cars are currently undergoing a public trial in Phoenix, Arizona, and as of November 2017, Waymo's fully self-driving vehicles are test-driving on public roads, without anyone in the driver's seat. Soon, members of the public will have the opportunity to use these vehicles in their daily lives. GM also says it will launch [a robot taxi service in 2019](#).

Technology

AI is being used in the various areas of the technology industry, including the following:

- DevOps
- Systems-level—compilers, processors, and memory
- Software development

Let's take a look at software development. Most would claim that the ultimate aim of technology is to make human life easier and more pleasurable by automating the tasks we find mundane and repetitious or that simply keep us away from doing the things we'd really love to be doing. Some types of programming might

fit into this category, and automating the mundane aspects of software development would make programmers happier and more productive. Also, businesses would like to reduce costs and improve the speed and accuracy of any workflow process, including programming, so this automation of much of the software development life cycle (SDLC) is inevitable. Finally, there's a chronic shortage of accomplished programmers, thus automation of the SDLC is strongly needed.

Let's now look at some of the efforts we have seen toward automating the SDLC. We can separate these into three categories, each of which is sufficiently different so as to have some of its own unique characteristics.

- Web development
- Application programming
- Machine learning development

Web development

By web development, we mean mostly frontend HTML programming. Motivated by the purpose statement “The time required to test an idea should be zero,” Airbnb is investing in a machine learning platform that will **recognize sketches** or drawings and turn them into **actionable code**. The Airbnb team built an initial prototype dubbed sketch2code, using about a dozen hand-drawn components as training data, open source machine learning algorithms, and a small amount of intermediary code to render components from its design system into the browser. The company developed a working theory that if machine learning algorithms can classify a complex set of thousands of handwritten symbols (such as handwritten Chinese characters) with a high degree of accuracy, it should be able to classify the 150 components within its system and teach a machine to recognize them. The Airbnb team firmly believes that AI-assisted design and development will be **baked into the next generation of tooling**. For further details, see this great hands-on **blog post by Emil Wallner**, read the **pix2code paper** and check out some of the related automation open source code for **pix2code** and **Keras** on GitHub.

Application programming

Application programming refers to anything other than frontend web development, and it involves programming applications in essentially any language, be that C, C++, Ruby, Python, Java, Go, Swift, or some other language.

A dream of AI has always been to build systems that can write computer programs. In this spirit, researchers at MIT's Computer Science and Artificial Intelligence Laboratory (CSAIL) developed a new system that allows programmers to **transplant code from one program into another**. The programmer can select the

code from one program and an insertion point in a second program, and the system will automatically make the modifications necessary—such as changing variable names—to integrate the code into its new context. Their results show that this system, named CodeCarbonCopy or CCC for short, can successfully transfer donor functionality into recipient applications.

“CodeCarbonCopy enables one of the holy grails of software engineering: automatic code reuse,” says Stelios Sidiroglou-Douskos, a research scientist at CSAIL and first author on the associated paper.²⁶ “It’s another step toward automating the human away from the development cycle.” In ongoing work, the researchers are looking to generalize their approach to file formats that permit more flexible data organization and programs that use data structures other than arrays, such as trees or linked lists. Other machine learning use cases for the automation of application software development include **caption generation on images**, and identifying and **fixing bugs in code**.²⁷ Two further papers that discuss using machine learning to automate the understanding and development of code are included in the following endnotes.^{28, 29}

Machine learning development

A powerful idea is that humans create AI that creates even more intelligent AI, which eventually will solve all our problems—including scientific, economic, and social challenges. Such a notion can be considered a kind of bootstrapping mechanism to get us to the singularity—a point in time whereby the machines are as generally intelligent as humans.¹ Beyond this point is impossible to predict, hence the term singularity.

Let’s investigate where we are on this journey. It turns out that Google has created a machine learning framework called **AutoML** that it is using to optimize the design and development of further machine learning models. Jeff Dean, head of the Google Brain team in Mountain View, described what he termed “automated machine learning” as one of the most promising research avenues his team was exploring.

AutoML was developed as a solution to the lack of top-notch talent in AI programming. There aren’t enough cutting-edge developers to keep up with demand, so the team at Google Research came up with machine learning software that can create self-learning code. The system runs thousands of simulations to determine which areas of the code can be improved, makes the changes, and continues the process until its goal is reached.

AutoML is often better at coding machine learning systems than the researchers who made it. In an image recognition task, for example, it reached a record-high **82% accuracy**. AutoML is the genesis for the next generation of machine learning and will help to accelerate the technology timeline. Google **recently announced** AutoML as a Service on its cloud platform. The hope lies in a future in which AI

takes care of time-consuming tasks like programming, thus freeing humans to do the things machines can't do. Automation of algorithmic design is discussed further in the papers listed in the References section.^{30, 31, 32, 33, 34}

Energy

AI is being used in the various areas of energy management, such as the following:

- Smart metering
- Distribution
- Energy efficiency
- Datacenter

Let's take a look at datacenter energy management. AI is being used in various areas of the tech industry, such as inside the hyperscalers (Google, Microsoft, Amazon, IBM, Apple). This includes **energy management** inside of Google's datacenters. Back in July 2016, Google announced that by using DeepMind's machine learning algorithms, the company had **reduced the amount of energy used for cooling its datacenters** by up to 40%. Using a system of neural networks trained on different operating scenarios and parameters within the datacenters, Google created a more efficient and adaptive framework to understand datacenter dynamics and optimize efficiency.

Then, in March 2017, DeepMind announced it had been in **talks with the UK government** about making the national grid more efficient by applying the same algorithms to the national power supply. Demis Hassabis, cofounder and CEO of DeepMind, hopes that the UK's energy usage could be reduced by as much as 10%, just through AI-led optimization. This would reduce the cost of energy for consumers and reduce global warming, especially if scaled to every country in the world.

Science

AI is being used in the various areas of science:

- Artificial scientists
- Physics
- Materials science
- Mathematics
- Theorem provers

With the well-publicized success of deep learning in other fields, including image classification, gaming, and NLP, it was only a matter of time before scientists became interested in seeing whether it could be used to solve problems in their respective fields. This includes analyzing large scientific datasets and coming up with new hypotheses. We discuss several areas in which deep learning models have surpassed other techniques in analyzing and finding patterns in data and coming up with hypotheses of their own.

First up, we take a look at particle physics. Various particle physics groups, including at CERN,^{35, 36, 37, 38} have been using a variety of machine learning algorithms to sift through data and identify particle signatures. A popular **Kaggle contest** used machine learning techniques to identify the Higgs boson.³⁹

Next, working on the GPU-accelerated Titan supercomputer at Oak Ridge National Laboratory (ORNL), a team developed an algorithm that **automatically generates neural networks**. **MENNDL**, which is short for Multi-node Evolutionary Neural Networks for Deep Learning, evaluates, tests, and recommends neural networks for unique datasets like those that scientists collect. In the past, the team would have spent months testing neural networks to find one that would work for a given problem. MENNDL did this in just 24 hours.

Deep learning methods have also been used to measure the various properties of neutrinos. At Fermilab, the NOvA collaboration is using hand-crafted CNNs to analyze neutrino experiment datasets⁴⁰ to enhance its results. The team concludes:

With minimal event reconstruction, we were able to build and train a single algorithm, which achieved excellent separation of signal and background for both of the NOvA electron-neutrino appearance and muon-neutrino disappearance oscillation channels. This algorithm, CVN, is a powerful approach to the problem of event classification and represents a novel proof of concept—that CNNs can work extremely well with non-natural images like the readout of a sampling calorimeter. We expect this approach to be transferable to a wide range of detector technologies and analyses.

Astrophysics is another area in which deep learning algorithms are being used to great effect, aiding in the discovery of various astronomical objects and phenomena including gravitational lensing and exoplanets. Researchers from the Stanford Linear Accelerator Center (SLAC) have for the first time shown that neural networks can **accurately analyze the complex distortions in space-time** known as **gravitational lenses** **10 million times faster** than traditional methods.⁴¹ The scientists used neural networks to analyze images of strong gravitational lensing, where the image of a faraway galaxy is multiplied and distorted into rings and arcs by the gravity of a massive object such as a galaxy cluster. The distortions provide important clues about how mass is distributed in space and how that distribution changes over time. These properties are linked to dark matter, which

comprises up to 80% of all matter in the universe, and to dark energy that's accelerating the expansion.

To train the neural networks in what to look for, the researchers fed them about half a million simulated images of gravitational lenses, taking about a day. After they were trained, the networks were able to analyze new lenses almost instantaneously with a precision that was comparable to traditional analysis methods. "Analyses that typically take weeks to months to complete, that require the input of experts and that are computationally demanding, can be done by neural nets within a fraction of a second, in a fully automated way and, in principle, on a cell phone's computer chip," said postdoctoral fellow Laurence Perreault Levasseur.

Discovery of exoplanets is another area augmented by machine learning in general, and neural nets in particular.^{42, 43} Finding exoplanets isn't easy. Compared to their host stars, exoplanets are cold, small, and dark, but with the help of machine learning, progress has recently been made. The measured brightness of a star decreases slightly when an orbiting planet blocks some of the light. The Kepler space telescope observed the brightness of 200,000 stars for four years to hunt for these characteristic signals caused by transiting planets. Using a dataset of more than 15,000 labeled Kepler signals, researchers from Google and the University of Texas, Austin, created a TensorFlow model to distinguish planets from non-planets. After they had a working model, the two research teams then looked at data from 670 stars that were already known to host exoplanets, discovering two new exoplanets in the process. Using this technique, **further exoplanet discoveries are expected in the future.**

Creating an artificial scientist that could explore datasets and then come up with hypotheses of its own is the dream of many a scientist, not least because it would accelerate scientific discovery, not only in basic research, but also in applied areas such as medicine and materials science. Clearly this would be of great benefit to society with the development of cures for diseases and of life-enhancing drugs, along with stronger, lighter materials. These "robot scientists" are already in use, coming up with hypotheses and equations in a wide variety of areas, including dynamical systems,⁴⁴ **quantum experiments,**⁴⁵ **genomics, drug discovery, scanning journals and scientific databases,** solving the Schrodinger equation,^{46, 47} investigating the **quantum many-body problem,** discovering phase transitions,⁴⁸ exploring transport phenomena,⁴⁹ and research in many more areas.⁵⁰ The interested reader can find further areas of science in which machine learning is being used in **this article from O'Reilly** and in the presentations and accepted papers from the **NIPS 2017 conference.**

How Businesses Can Get Started in Machine Learning

We have seen that the spectacular success of AI results from a combination of large datasets, sophisticated learning algorithms, and dedicated hardware. You

will need all three elements to implement an AI strategy. Datasets can come from internal to your organization or from external sources, either proprietary or open source. Some examples of open source datasets are provided in the subsections that follow. The machine learning algorithms will either be proprietary or open source, whereas the hardware (compute, storage, network) can be on-premises or cloud based. The subsections that follow provide examples of open source datasets, machine learning frameworks, and cloud-based MLaaS.

Why Big Data Is the Foundation of Any Machine Learning Initiative

The good news is that, if you started a big data initiative half a decade ago, it's about to pay off. Through exponentially increasing advances in hardware and data, the time is right to capitalize on this Cambrian explosion of machine learning technologies. We've spoken briefly about advances in hardware (CPU, GPU, FPGA, and ASIC), but thanks to the web there is also an explosion in valuable data that individuals and companies can use for innovation and increased profits.

Data can come from anywhere, both internal and external to the company. A lot of companies and organizations are open sourcing datasets with which you can help train your machine learning algorithms. Other companies make a living by collating and selling annotated (labeled) datasets or models trained on various datasets. **Figure 1-5** presents some examples of data sources.

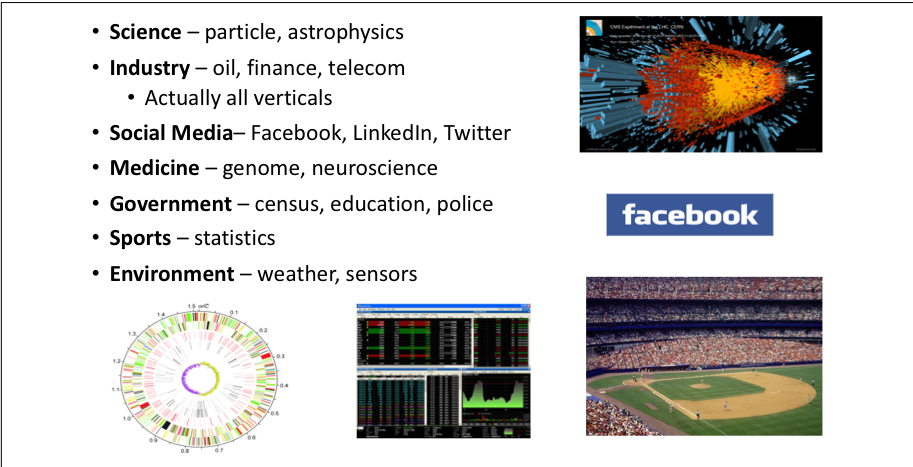


Figure 1-5. Where does data come from?

Here are some examples of open source datasets. The interested reader is encouraged to explore these further:

Images

MNIST, CIFAR-10, ImageNet, PASCAL VOC, Mini-Places2, Food 101, flickr8k, flickr30k, COCO

Text

IMDB, Penn Treebank, Shakespeare Text, bAbI, Hutter-prize

Video

UCF101, Kinetics, YouTube-8M, CMU mocap

Speech

AudioSet, wav2letter, openSLR, TIMIT, BABEL

What Is a Data Scientist?

A data scientist is someone who solves business problems through the identification, collection, and analysis of data. Companies hire data scientists to help solve existing business problems and to suggest and help design new products and services. It is now possible to create products and services that were not feasible before, and even to enter entire new fields and markets, by exploiting the explosion of data and exponential hardware improvements. Applying statistical methods, including machine learning, data scientists can wrangle raw data to identify patterns and make predictions hitherto infeasible using traditional compute and datasets. Distributed computing technologies such as Hadoop and Spark can process huge datasets (sometimes at petabyte scale) that would have been impossible a few years ago. Data scientists are sometimes referred to as “unicorns” because they require a rare combination of computer science, statistics and domain knowledge. A typical data science workflow might look like the following:

1. Data collection

Data sources, both public and private, are identified and the data is collected and stored in the company's data storage system in varying formats. This could include SQL or NoSQL databases, which are used for storing structured and unstructured data, respectively.

2. Data preparation

The process whereby the data is examined and cleaned up. This includes normalizing the data, addressing missing or anomalous data, and removing duplicates. Often, much of this process can be automated using open source or proprietary data preparation tools, such as those listed in the References section.^{51, 52, 53}

3. Feature engineering

Feature engineering attempts to increase the predictive power of learning algorithms by creating features from raw data that help facilitate the learning

process. More recently, automation of feature engineering helps data scientists reduce data exploration time, allowing them to try many ideas quickly. On the other hand, it enables people who are not familiar with data science to quickly extract value from their data with little effort, time, and cost. Tools include [FeatureTools \(MIT\)](#), [OneBM \(IBM\)](#), and [ExploreKit \(Berkeley\)](#).

4. Feature selection

Depending on the actual problem being addressed, certain features of the data will be more useful than others in solving the problem. These are highlighted and extracted using domain knowledge, data analysis tools, and statistical methods. Feature selection can be done manually or automated.

5. Sampling

In statistical analysis, one can calculate how large a sample of data is needed to yield a given accuracy in the data analysis. The actual sample size depends on the distribution of the data, whether it's normal, Poisson, binomial, and so on.

6. Algorithm implementation

At this point, we can decide on the specific algorithms we are going to use in our data analysis. The idea is to try various algorithms to see which gives the highest accuracy and/or the most precision. Often an ensemble of different algorithms yields the best results. This stage entails a certain amount of trial and error, along with data science experience.

7. Hyperparameter tuning

Machine learning algorithms consist of various hyperparameters, such as learning rates, initialization values, decay rates, regularization, number and types of layers, activation functions—there are many depending on the specific type of algorithm we are using. Tuning hyperparameters can be a time-consuming process and often more of an art than a science. Fortunately, this process has recently become more automated through the use of various tools such as [AutoML](#), as discussed earlier.

8. Model selection

We need to decide which model we are going to use on a particular dataset. For example, if we decide to use deep learning, we would need to select from the various available neural network models such as VGGnet, ResNet, Inception, and AlexNet.

9. Model serving (scoring)

Now it's time to test and compare our machine learning models. By assigning a score to the results of each model, possibly depending on a multitude of factors like accuracy, time to run, and so forth, we can rank the models in order of efficiency and select the top-performing model(s).

10. Prediction insights

By running the models on some data of interest, we are now in the position to use our models to make predictions. This can be classification, regression (fitting a curve through the data), or the clustering of unsupervised data.

11. Metrics

Again, we score the capability of our models by comparing their performance against various metrics that we have determined to be important for our tests. From these tests we can choose what we consider to be the one best model with which to move forward.

12. Proof of Concept

We need to test the chosen model; that is, deploy it on our hardware, whether that is locally on our laptop, or in a large-scale distributed test environment.

13. Productize and scale

Finally, we productize our machine learning model by deploying it into our production environment. This will typically be a multinode distributed environment running in a datacenter, or perhaps geographically dispersed for resiliency. **TensorFlow Serving** is an example of a machine learning framework that productizes machine learning models.

The bad news is that the data scientist you need to hire is hard to come by

Due to the breadth and depth of data science skills, data scientists are hard to come by. These skills include mathematics, statistical analysis, computer science (including machine learning and programming skills), and domain expertise. A person skilled in any one of these areas is difficult to find, let alone all of the above. The journey to becoming a fully-fledged data scientist can be a challenging one. Neural networks, reinforcement learning, generative adversarial networks (GANs), and transfer learning are all fast becoming part of the data scientist's day-to-day toolkit.

Luckily, technology can help fill the gaps in your personnel

Fortunately, various products and statistical libraries exist so that we don't need to reinvent the wheel when statistically analyzing these very large datasets, remembering you will still need a few data engineers on hand in order to build and maintain your big data cluster(s).

Automation of the Data Science Life Cycle

Another strong trend we see is the automation of the data science pipeline with tools and projects such as **Automatic Statistician**, along with those discussed and

referenced earlier. By the end of 2018, most data science toolkits will include tools for data preparation, exploratory data analysis (EDA), automated feature engineering, hyperparameter tuning, model deployment, and other pipeline tasks.^{54, 55} The field of data science has progressed rapidly in the past five years and **will continue to do so**.

The Build-Versus-Buy Decision

Of course, the CDO/CTO/CIO always faces the build-versus-buy decision—is it more feasible and cost effective to license a company's data science solution, or to build it from scratch using open source technologies? There are pros and cons for both approaches—let's take a look at these now.

Buying a Commercial-Off-the-Shelf Solution

Buying a commercial-off-the-shelf (COTS) solution might be best for those companies that lack the personnel to create AI on their own or have non-unique business data. Although buying a proprietary COTS solution is never inexpensive, it can save a lot of time and labor if data science capability is scarce or difficult to hire. Proprietary COTS machine learning solutions include the following:

SAS

Founded in 1976, SAS grew out of the need for a computerized statistics program to analyze vast amounts of agricultural data collected through United States Department of Agriculture (USDA) grants. An abbreviation for Statistical Analysis System, today SAS has more than 14,000 employees in 148 countries.

RapidMiner

RapidMiner is a data science platform that supports all steps of the machine learning process including data preparation, results visualization, model validation, and optimization. It was developed starting in 2001 by a team at the Artificial Intelligence Unit of the Technical University of Dortmund, Germany. Formerly known as YALE (Yet Another Learning Environment), the name of the software was changed to RapidMiner in 2007.

Ayasdi

Ayasdi's machine intelligence platform combines scalable computing and big data infrastructure with the latest machine learning, statistical, and geometric algorithms and topological data analysis (TDA) to enable data scientists, domain experts, and business people to be more productive. TDA is based on topology, the mathematical discipline that studies the notion of shape. It draws on the philosophy that all data has an underlying shape and that shape has meaning. As a framework, TDA uses dozens of machine learning algorithms to automatically determine the best shape of the data. In doing so, it

uncovers subtle patterns that can be missed by other approaches that rely on a single algorithmic dimension. It can determine what in the data is important in an unsupervised fashion—meaning users often find answers to questions they didn't know to ask. Finally, topological data analysis provides justification for its findings and recommendations. Ayasdi was founded by three researchers at Stanford University in 2008.

BigML

Founded in 2011, BigML's mission is to make machine learning simple and beautiful for everyone. BigML is a consumable, programmable, and scalable machine learning platform designed to solve and automate classification, regression, time-series forecasting, cluster analysis, anomaly detection, association discovery, and topic modeling tasks. With a user-friendly interactive interface, BigML makes it easy to fully automate end-to-end machine learning tasks.

H2O

H2O.ai makes machine learning accessible and allows business users to extract insights from data without needing expertise in deploying or tuning machine learning models. Founded in 2011, H2O provides an open source machine learning platform. Products include Sparkling Water (machine learning on Spark) and Driverless AI, which automates machine learning. Solutions include claims processing, credit scoring, customer churn, fraud detection, ICU monitoring, operational intelligence, and predictive maintenance.

Wolfram Alpha

Wolfram Alpha is the computational knowledge engine from the company founded by Stephen Wolfram, and a leader in applying computing to the sciences. Wolfram Alpha's long-term goal is to make all systematic knowledge immediately computable and accessible to everyone. It works by using its vast store of expert-level knowledge and algorithms to automatically answer questions, do analysis, and generate reports. It contains over 10-trillion pieces of data from primary sources with continuous updating and more than 50,000 types of algorithms and equations.

Matlab

MathWorks, the maker of Matlab, is a leading developer of mathematical computing software for engineers and scientists. Matlab's Neural Network Toolbox (NNT) available in the 2017b release, uses the frameworks Caffe and TensorFlow-Keras and models including AlexNet, GoogLeNet, VGG-16, VGG-19, and ResNet-50 to deploy deep learning models. It provides algorithms, pretrained models, and apps to create, train, visualize, and simulate both shallow and deep NNs. You can perform classification, regression, clustering, dimensionality reduction, time-series forecasting, and dynamic sys-

tem modeling and control. For time-series classification and prediction, the toolbox provides LSTM networks. With NNT, you can visualize intermediate layers and activations, modify network architecture, and monitor training progress.

IBM Watson

Watson is IBM's machine learning platform. It covers all machine learning analysis such as NLP, image classification, and time-series analysis, in all business domains, including healthcare, FinTech, transportation, and retail. Watson on the IBM Cloud allows you to integrate AI into applications and store, train, and manage data securely and at scale.

Dataiku

Dataiku's collaborative Data Science Software (DSS) platform helps to automate the data science work flow. It acts as a central hub for technical and nontechnical users to prototype, build, scale, deploy, and manage advanced data science products. The DSS platform contains a collaborative and team-based user interface for data scientists and beginner analysts with plug-ins to common machine learning platforms, such as scikit-learn, MLlib, and H2O.

Bonsai

Bonsai abstracts away the complexity of machine learning libraries like TensorFlow so that developers, data scientists, and subject matter experts can more effectively program and manage AI models. As machine learning and deep learning algorithms evolve, you can recompile and retrain Bonsai's code to take advantage of low-level technology advances. Applications include increasing automation of control systems such as robotics, warehouse operations, smart manufacturing, and smart sensors. It also optimizes real-time decision support for business systems including energy efficiency, supply chain, network optimization, and planned maintenance.

DataRobot

Founded in 2012, with more than 270 employees, DataRobot offers an automated machine learning platform. It automatically builds and evaluates hundreds of models, enabling users to rapidly build and deploy highly accurate machine learning models. Incorporating a library of hundreds of the most powerful open source machine learning algorithms, the DataRobot platform automates, trains, and evaluates predictive models in parallel. DataRobot brings data science to the masses—what were previously costly initiatives that only a data scientist could design or interpret are now, with automated machine learning, viable for the mainstream market.

Pros and cons of machine learning COTS

Cost is obviously a critical factor in using proprietary software, as opposed to open source for which the code is free. However, with this cost, you get support

and Service-Level Agreements (SLAs) whereby the vendor is responsible and can be held accountable for fixing bugs. Reliability is expected to be better, and speed to fix bugs should be faster with COTS software compared with open source frameworks.

Open Source Machine Learning Solutions

Building your own solution (e.g., through Python) is appropriate for those organizations that can hire a data scientist or that have unique data. Following is a list of some open source machine learning and deep learning packages:

KNIME

Zurich-based KNIME, with more than 1,500 modules, hundreds of ready-to-run examples, a comprehensive range of integrated tools, and a wide choice of advanced algorithms, provides a comprehensive data scientist toolbox. Through Keras, users have access to a variety of different state-of-the-art deep learning frameworks, such as TensorFlow and CNTK.

NuPIC

NuPIC (Numenta Platform for Intelligent Computing) is an implementation of hierarchical temporal memory (HTM), a theory of intelligence based on the neuroscience of the neocortex. NuPIC is suited to a variety of problems, including anomaly detection and prediction of streaming data sources.

scikit-learn

scikit-learn was started in 2007 as a Google Summer of Code project. A popular open source machine learning toolkit, it contains algorithms for supervised and unsupervised learning, including Gaussian processes and neural networks. It has been a consistent entrant in the Google Summer of Code program, held each summer. It includes dataset loading and transformation, along with model selection and evaluation. Before the release of TensorFlow in November 2015, scikit-learn was the most popular open source machine learning framework. It is built on the Python packages NumPy, SciPy, and matplotlib.

MLlib

MLlib is Apache Spark's scalable machine learning library consisting of common learning algorithms and utilities, including classification, regression, clustering, collaborative filtering, and dimensionality reduction. It uses the linear algebra package Breeze.

Stan

Stan is a framework used for statistical modeling, data analysis, and prediction in the social, biological, and physical sciences, engineering, and business. Users specify log density functions in Stan's probabilistic programming language and get full Bayesian statistical inference with Markov Chain

Monte Carlo (MCMC) sampling, approximate Bayesian inference with variational inference, and penalized maximum likelihood estimation with optimization. Stan's math library provides differentiable probability functions and linear algebra. It is a part on the NumFocus open source software organization.

Weka

Weka (Waikato Environment for Knowledge Analysis) is data mining software written in Java. It is a collection of machine learning algorithms that can either be applied directly to a dataset or called from the users' Java code. Weka contains tools for data preprocessing, classification, regression, clustering, association rules, and visualization.

Core ML

Core ML is an open source machine learning framework used across Apple products, including Siri. Core ML enables users to build apps with intelligent new features using just a few lines of code. In addition to supporting extensive deep learning with more than 30 layer types, it also supports standard ML models such as tree ensembles, support vector machines (SVMs), and generalized linear models. Features include face tracking, face detection, object tracking, language identification, tokenization, lemmatization, part of speech, and named entity recognition.

Prediction.io

Apache PredictionIO is an open source machine learning server built on top of an open source stack for developers and data scientists to create predictive engines for machine learning tasks. Built on Spark, it supports machine learning and data processing libraries such as Spark MLlib and OpenNLP. Following a successful launch in 2012, PredictionIO was acquired by Salesforce and open sourced to the Apache Foundation in July 2016.

Additional Machine Learning Frameworks

You can find additional machine learning frameworks, both proprietary and open source, at the following locations:

- [Collection on GitHub](#)
- [Machine learning software](#)
- [Comparison of statistical packages](#)
- [List of statistical packages](#)

Open Source Deep Learning Frameworks

As we saw earlier, deep learning is a subset of machine learning that involves neural networks as the underlying data analysis technology. Let's look at a number of those frameworks here.

TensorFlow

By far the most popular deep learning framework in use by developers and data scientists today is the Google open source framework, **TensorFlow**. It is the second highest ranked software framework on **GitHub** ranked by number of stars. It supports all the most common hardware platforms and is used extensively within Google.

Other deep learning frameworks

MXnet

MXnet is the Amazon-led deep learning framework.

CNTK

CNTK is the Microsoft-developed framework.

Keras

Keras was developed by Francois Chollet, who works at Google. It provides a high-level, user-friendly interface to TensorFlow and CNTK, allowing for easy and fast prototyping through user modularity and extensibility.

PyTorch

PyTorch, as the name suggests, is a Python framework based on Torch. Using dynamic computation graphs, it was developed by Soumith Chintala and is used for research within Facebook.

Caffe2

Caffe was initially developed by the Berkeley BVLC group, whereas Caffe2 was extended and is used internally in production at Facebook.

Chainer

A deep learning framework from Preferred Networks, who are based in Japan.

DL4J

A Java-based framework initially developed by Adam Gibson, commercialized through the company SkyminD.

Neon

Intel's framework developed to run on its Nervana deep learning ASIC, the Neural Network Processor (NNP).

Gluon

An API developed by Amazon and Microsoft that supports MXnet, CNTK, and some other deep learning frameworks. It offers a full set of plug-and-play neural network building blocks, including predefined layers, optimizers, and initializers. It comes as part of MXnet.

ONNX

Open Neural Network eXchange is an open source format created by Facebook and Microsoft that enhances interoperability between deep learning frameworks. ONNX enables models to be trained in one framework and transferred to another for inference, for example.

Commercial Open Source

Commercial open source frameworks are ones whereby two types of product can be offered: a free open source edition and a commercial edition (which might include more than one type of license). Open source software with commercial support combines the customizability and community of open source with the dedicated support of a commercial partner. These hybrid options are appropriate for teams that want the flexibility of open source packages but also need a support safety net for mission-critical applications. Here's what commercial support generally means:

- SLA-backed support so that companies can meet their deadlines, rather than relying on community message boards.
- Regular maintenance releases so that upgrades can be planned.
- License indemnification, so if a developer makes changes to an open source library in a way that violates that library's licensing terms, the commercial support vendor rather than the company is held liable.

Examples of commercial open source vendors include the following:

Anaconda

Anaconda, previously Continuum Analytics, offer a data science package manager with major open source data science libraries prebundled. It offers Anaconda Distribution (free edition) and Anaconda Enterprise, its commercial edition. Prebundled packages include Jupyter, NumPy, pandas, pip, scipy, dask, spyder, and scikit-learn. Both editions include over 1,000 popular data science packages along with the Conda package and virtual environment manager for Windows, Linux, and macOS. The commercial product allows organizations to collaborate, govern, scale, and secure Python and R in enterprise environments.

RStudio

RStudio is an integrated development environment (IDE) for the R programming language. It is available in open source and commercial editions and runs on the desktop (Windows, macOS, and Linux) or in a browser connected to RStudio Server. It includes a code editor, debugging, and visualization tools. RStudio develops free and open tools for R and enterprise-ready professional products for teams to scale and share work.

ActiveState

ActiveState Python for machine learning includes the packages TensorFlow, Keras, scikit-learn, pandas, NumPy, scipy, and matplotlib. ActivePython comes in a community edition and various commercial editions (Business, Enterprise, and OEM). ActiveState Enterprise Edition provides guaranteed technical support, security, legal indemnification, and quality assurance, yielding the advantages of open source while minimizing the risks. Language distributions are available on all platforms (Window/Linux/macOS) as well as Big Iron (AIX, Solaris, and HP-UX).

AI as a Service (Cloud Machine Learning)

Of course, there are always cloud services available, which offer a range of data science services and machine learning frameworks including most of those already mentioned. The most common services are provided by the big four:

- **Amazon Web Services (AWS)**
- **Google Cloud Platform (GCP)**
- **Microsoft Azure**
- **IBM Cloud**

The following subsections provide login info for the various cloud providers and descriptions of some of their machine learning related service offerings.

AWS

Amazon Machine Learning (Amazon ML; see **Figure 1-6**) is a cloud-based service that allows developers of all skill levels to use machine learning technology. AI services include Rekognition image and video (vision), Lex (conversational chatbots), and Comprehend, Translate, Transcribe, and Polly (language). AWS uses Kinesis for real-time streaming data, which you can use to ingest data into your machine learning platform.

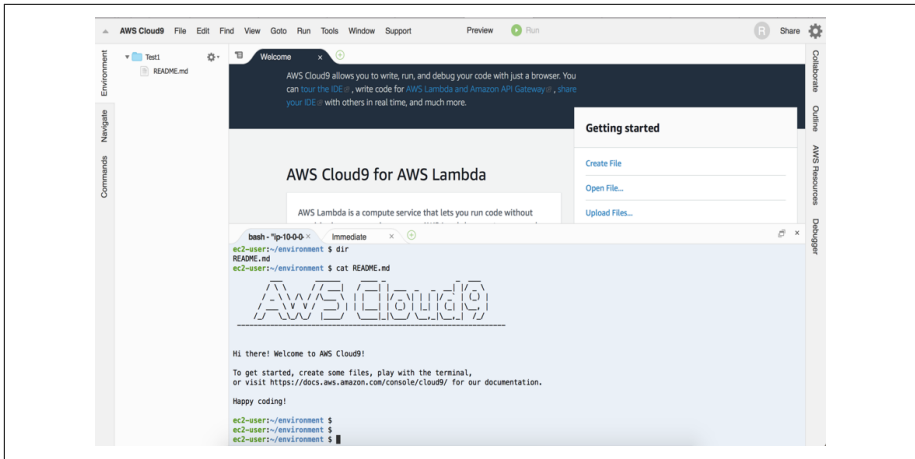


Figure 1-6. Login screen for AWS

To ease the difficulty of the data science process, **Amazon SageMaker** enables data scientists and developers to build, train, and deploy machine learning models with high-performance machine learning algorithms, broad framework support, and training, tuning, and inference. SageMaker has a modular architecture so that data scientists can use any or all of its capabilities in existing machine learning workflows.

AWS Deep Learning AMIs are preconfigured environments with which you can quickly build deep learning applications. Built for Amazon Linux and Ubuntu, the AMIs come preconfigured with popular deep learning frameworks such as MXnet with Gluon, TensorFlow, CNTK, Caffe2, PyTorch, and Keras to train custom AI models and experiment with new algorithms.

The **AWS Deep Learning AMIs** run on Amazon EC2 P2 instances as well as P3 instances that take advantage of NVIDIA's Volta architecture. The AMIs come integrated with the Intel Math Kernel Library (MKL) and installed with Jupyter notebooks. To simplify package management and deployment, the AMIs install the Anaconda Data Science Platform for large-scale data processing, predictive analytics, and scientific computing. You can find more information at <https://aws.amazon.com/machine-learning/amis/> or in the **AWS documentation for AMIs**.

You can find additional details for all of the above services at <https://aws.amazon.com/aml/>.

Finally, you can access Amazon machine learning documentation [here](#).

GCP

Google Cloud's AI (Figure 1-7) provides modern machine learning services with pretrained models and a service to generate tailored models. Google Cloud ML Engine makes it easy for users to build large-scale machine learning models that cover a broad set of scenarios from regression models to image classification. It is integrated with other GCP products such as Cloud Storage, Dataflow, and Datalab. GCP uses Dataflow or Dataproc for real-time streaming data, which you can use to ingest data into a machine learning platform. Cloud Datalab is an interactive tool created to explore, analyze, transform, and visualize data and build machine learning models on GCP.

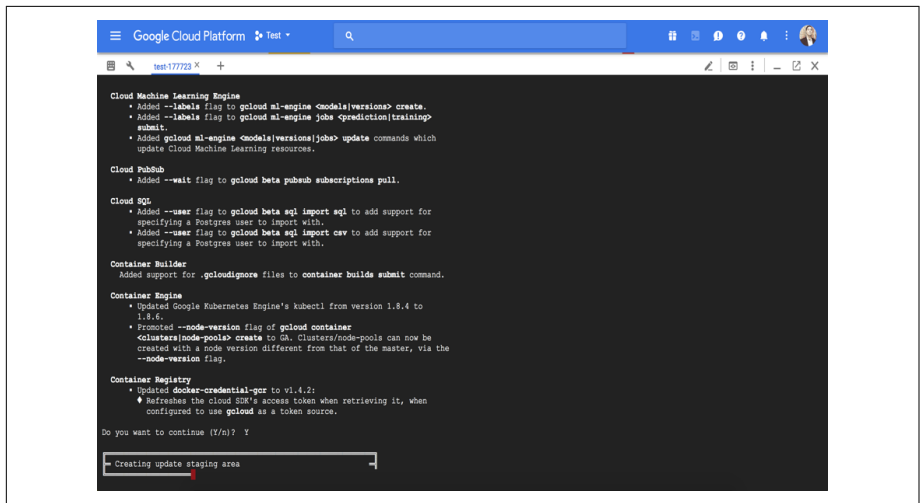


Figure 1-7. Login screen for GCP

Google Cloud Vision API classifies images into thousands of categories, detects individual objects and faces within images, and finds printed words contained within images. Video analysis, provided by the Google Cloud Video Intelligence API, makes videos searchable and discoverable by extracting metadata, identifying key nouns, and annotating the content of the video.

Google Natural Language API reveals the structure and meaning of text through machine learning models. It extracts information about people, places, and events mentioned in text documents as well as understanding sentiment. Google Cloud Translation API provides a simple interface for translating an arbitrary string into any supported language (e.g., French to English).

Google Cloud Speech API converts audio to text by applying neural network models in an API that recognizes more than 110 languages. DialogFlow is an end-to-end development suite for building conversational interfaces (chatbots)

for websites and mobile applications that are capable of natural and rich interactions between your users and your business.

Finally, a strong differentiator that GCP has is the integration of TensorFlow, Kubernetes (the distributed container service), and TPUs into the GCP stack, as they were all developed by Google. As mentioned earlier, AutoML is a service that automatically trains models on various datasets. You can find additional information in the [Cloud Machine Learning Engine Documentation](#).

Azure

Microsoft Azure (Figure 1-8) offers a range of **cognitive services** on its cloud platform.

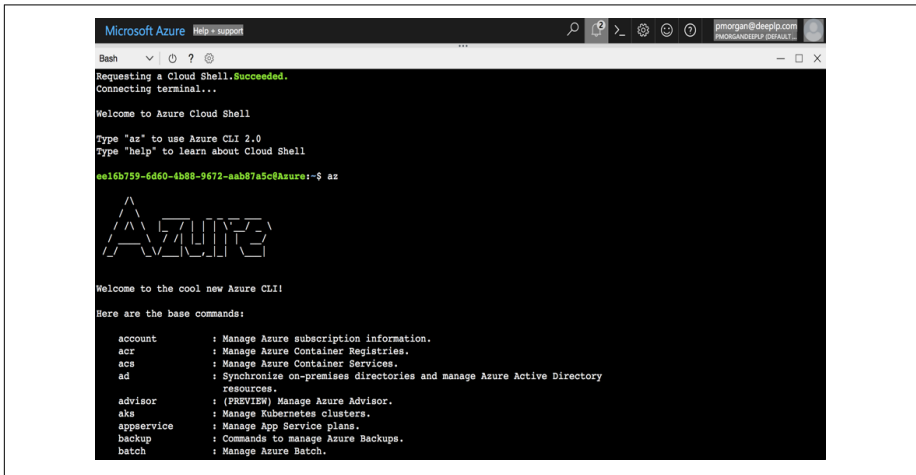


Figure 1-8. Login screen for Microsoft Azure

Infrastructure services include Spark and Azure Container Services (AKS), based on Kubernetes. Azure uses Stream Analytics for real-time streaming data, which you can use to ingest data into a machine learning platform.

You can find additional information on the Azure AI Platform [here](#). Documentation for Azure machine learning is available at <https://docs.microsoft.com/en-gb/azure/machine-learning/preview/>.

IBM Cloud

IBM's AI and machine learning offering is essentially **Watson running in the IBM Cloud**. Services include natural language understanding (NLU), including text-to-speech and chatbots, image classification, and video analysis.

Languages

The common languages of data science are Python, R, and Julia. All have their pros and cons. Python is a popular multipurpose language with a large database of users, many coming from the scientific community. R is particularly suited to statistical analysis, growing as it did from the statistics community. Julia combines the user friendliness of Python with the speed of compiled languages such as C/C++. It thus solves the “two-language problem”: the tendency to prototype your data science models in a high-level scripting language and then port them to a faster language like C++ for production. For this reason, Julia is proving to be a very popular and well-received new language among the data science community.

Data Science Notebooks

Data science notebooks are proving very popular in the data science community due to their ease of use and capabilities. One can code in them as well as display visualizations and text for the data science analysis. They are also very convenient to share work among collaborative data science efforts. Notebooks come as standalone as well as being offered by the major cloud providers. Jupyter notebooks (the name being a clever concatenation of JULia, PYThon and R) is the most popular notebook, and several of the following are in fact built on top of the Jupyter kernel:

- Jupyter
- JupyterLab
- Zeppelin
- CoLaboratory
- JunoLab (Julia)

Cloud notebooks:

- Azure
- AWS
- GCP
- IBM

Pros and Cons of Machine Learning Open Source Tools

Open source software is free, which is perfect for those who are budget constrained or just wanting to prototype and try the source code for themselves. Open source is also attractive because the tools are not likely to shrivel up or take

an unexpected direction, as proprietary offerings sometimes do. The open source communities around the more popular frameworks such as TensorFlow and MXnet are in the tens of thousands, so there is a great deal of contribution and support in general. The flexibility you have in running open source code is something a lot of developers appreciate because they can fork the code repositories and have full access to the codebase to extend and change the source as they see fit.

Looking Ahead: Emerging Technologies

So, what lies ahead in this exponentially accelerating world of data science and technology in general? Well, in a nutshell: faster hardware, better algorithms, and more comprehensive datasets. Hardware improvements include processors optimized for data processing, such as ASICs and neuromorphic chips. ASICs include the [Google TPU](#), [Graphcore IPU](#), [Intel Nervana](#), and [Wave Computing](#).

In terms of better algorithms, we are looking at improvements in the following areas: unsupervised learning (learning with no or very few labels), transfer learning, neuroscience inspired biologically plausible algorithms,^{56, 57, 58, 59} and a movement of the research field toward more general intelligence algorithms^{60, 61, 62} including active inference.⁶³

As described above, we are seeing various efforts to combine AI with blockchain. Further afield, we see two new types of processors (neuromorphic and quantum) beginning to come out of the labs and being deployed in experimental and prototype production environments. Neuromorphic computing efforts include IBM's [TrueNorth](#), the Human Brain Project's [SpiNNaker](#) and [BrainScaleS](#), and Intel's [Liohi neuromorphic processor](#), which the company announced at CES 2018. In the quantum computing space we have corporations [IBM](#), [Microsoft](#), [Google](#), and [Intel](#) all working on their own projects, together with an ecosystem of smaller companies like [DWave](#) and startups growing to support the coming wave of quantum computing in the enterprise.

Conclusions: Start Investing in Machine Learning or Start Preparing to be Disrupted

AI tools, both open source and proprietary, are already well established in the market and are gaining traction at an exponential rate. Rapid development of various new functionality and feature sets, plus the rapid advancements in hardware, both on-premises and in the cloud, will make AI and machine learning even more pervasive. It will be embedded in more applications throughout the business world, both internally and in new products and services delivered to end users. In a [recent report](#), Deloitte examined bottlenecks to adoption of machine learning in businesses and the recent progress expected to alleviate said

bottlenecks and accelerate adoption. In **another report**, PWC predicted that AI will add more than \$15.7 trillion to the global economy by 2030, potentially the largest market disruption of all time.

The computer and telecommunications revolution clearly replaced a lot of manual data entry along with messengers, but new jobs were created, and productivity increased considerably with the use of computer processors, storage, communications, and applications such as spreadsheets and other productivity and collaborative software tools. This time might be different, however, as it's not clear that new jobs for humans can be created after artificial intelligence supersedes human intelligence in performing a wide range of tasks. This means that social and economic changes might be needed, such as the introduction of a universal basic income, which is currently in trials in various parts of the world. A basic income would allow humans to pursue other tasks that would be more appealing to us—such as intellectual pursuits, starting our own businesses, or generally being free to choose whatever we'd like to spend our time on.

All in all, it is clear that those who embrace machine intelligence will do well; those who don't might very well be left behind in what is shaping up to become the fourth industrial revolution, and the largest one so far.

References

- [1] Kurzweil, Ray, *The Singularity is Near* (New York: Penguin Books, 2006).
- [2] Ford, Martin, *The Rise of the Robots* (New York: Basic Books, 2015).
- [3] Barrat, James, *Our Final Invention* (New York: St. Martins, 2015).
- [4] LeCun et al., “Deep Learning,” *Nature* 521 (May 2015): 436–444. <https://www.nature.com/articles/nature14539>.
- [5] Goodfellow et al., *Deep Learning* (Cambridge: MIT Press, 2016).
- [6] Canziani et al. “An Analysis of Deep Neural Network Models for Practical Applications.” *arXiv* (April 2017). <https://arxiv.org/abs/1605.07678>.
- [7] Silver et al. “Mastering the Game of Go Without Human Knowledge,” *Nature* 550 (19 Oct 2017): 354–359. <http://www.nature.com/articles/nature24270>.
- [8] Sabour, S. et al. “Dynamic Routing Between Capsules,” *arXiv* (Nov 2017). <https://arxiv.org/abs/1710.09829>.
- [9] Hu et al. “Squeeze-and-Excitation Networks.” *arXiv* (Sept 2017). <https://arxiv.org/abs/1709.01507>.
- [10] You et al. “ImageNet Training in Minutes.” *arXiv* (Dec 2017). <https://arxiv.org/abs/1709.05011>.

- [11] Poplin, R. et al. “Creating a universal SNP and small indel variant caller with deep neural networks,” *bioRxiv* (January 2018). <https://www.biorxiv.org/content/early/2018/01/09/092890>.
- [12] Google. “1000 Genomes Project.” Last updated January 8, 2018. <https://cloud.google.com/genomics/data/1000-genomes>.
- [13] NCBI. “Genome Resources.” <https://www.ncbi.nlm.nih.gov/genome/>.
- [14] Alipanahi, B. et al. “Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning,” *Nature Biotechnology* 33 (July 2015): 831–838. <https://www.nature.com/articles/nbt.3300>.
- [15] Leung, M. et al. “Machine Learning in Genomic Medicine: A Review of Computational Problems and Data Sets,” *Proceedings of the IEEE* 104, no. 1 (January 2016): 176–197. <http://ieeexplore.ieee.org/document/7347331/>.
- [16] Bao, W. et al. “A deep learning framework for financial time series using stacked autoencoders and long-short term memory,” *PLoS ONE* 12, no. 7 (July 2017). <http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0180944>.
- [17] Arévalo, A. et al. “High-Frequency Trading Strategy Based on Deep Neural Networks,” *ICIC 2016: Intelligent Computing Methodologies*, Springer (July 2016): 424–436. https://doi.org/10.1007/978-3-319-42297-8_40.
- [18] Ding X et al. “Deep learning for event-driven stock prediction.” *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)* (July 2015). <https://www.aaai.org/ocs/index.php/IJCAI/IJCAI15/paper/view/11031>.
- [19] Dixon, M. F. et al. “Classification-Based Financial Markets Prediction Using Deep Neural Networks.” *Algorithmic Finance* (June 2017). <https://arxiv.org/abs/1603.08604>.
- [20] Heaton, J. B. et al. “Deep Learning in Finance,” *arXiv* (Jan 2018). <https://arxiv.org/abs/1602.06561>.
- [21] Honchar, A. “Neural networks for algorithmic trading” series, Medium, 19 Oct 2017. <https://medium.com/machine-learning-world/neural-networks-for-algorithmic-trading-enhancing-classic-strategies-a517f43109bf>.
- [22] Huang, S. “Predicting Cryptocurrency Price With Tensorflow and Keras,” Medium, 1 Jan, 2018. <https://medium.com/@huangkh19951228/predicting-cryptocurrency-price-with-tensorflow-and-keras-e1674b0dc58a>.
- [23] Longmore, K. “Deep Learning for Trading Part 1: Can it Work?” Robot-wealth blog, accessed Feb 23, 2018. <https://robotwealth.com/deep-learning-trading-part-1/>.
- [24] Sigmoidal. “Machine Learning for Trading—Overview.” Sigmoidal blog, accessed Feb 23, 2018. <https://sigmoidal.io/machine-learning-for-trading/>.

- [25] Janai, J. et al. “Computer Vision for Autonomous Vehicles: Problems, Datasets and State-of-the-Art,” *arXiv* (April 2017). <https://arxiv.org/abs/1704.05519>.
- [26] Sidiropoulos-Douskos, S. et al. “CodeCarbonCopy,” *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering* (September 2017): 95–105. <http://dx.doi.org/10.1145/3106237.3106269>.
- [27] Long, F. and M. Rinard. “Automatic Patch Generation by Learning Correct Code,” POPL’16, Jan 20–22, 2016, St. Petersburg, FL. <http://people.csail.mit.edu/fanl/papers/prophet-popl16.pdf>.
- [28] Hellendoorn, V. et al. “Are Deep Neural Networks the Best Choice for Modeling Source Code?” ESEC/FSE’17 (September 2017). <http://web.cs.ucdavis.edu/~devanbu/isDLgood.pdf>.
- [29] Balog, M. et al. “Deepcoder: Learning to Write Programs,” *arXiv* (November 2016). <https://arxiv.org/abs/1611.01989>.
- [30] Baker, B. et al. “Designing Neural Network Architectures using Reinforcement Learning,” *arXiv* (March 2017). <https://arxiv.org/abs/1611.02167>.
- [31] Li, K. and J. Malik. “Learning to Optimize,” *arXiv* (June 2016). <https://arxiv.org/abs/1606.01885>.
- [32] Zoph, B. and Q. Le. “Neural Architecture Search with Reinforcement Learning,” *arXiv* (February 2017). <https://arxiv.org/abs/1611.01578>.
- [33] Duan, Y. et al. “RL2: Fast Reinforcement Learning via Slow Reinforcement Learning,” *arXiv* (November 2016). <https://arxiv.org/abs/1611.02779>.
- [34] Wang, J. et al. “Learning to Reinforcement Learn,” *arXiv* (January 2017). <https://arxiv.org/abs/1611.05763>.
- [35] Castelvechi, D. “Artificial intelligence called in to tackle LHC data deluge.” *Nature* 528 (December 2015): 18–19. <http://www.nature.com/news/artificial-intelligence-called-in-to-tackle-lhc-data-deluge-1.18922>.
- [36] Baldi, P. et al. “Searching for exotic particles in high-energy physics with deep learning,” *Nature Communications* 5 (July 2014). <https://www.nature.com/articles/ncomms5308>.
- [37] Databases at CERN. “Distributed Deep Learning with Apache Spark and Keras.” January 25, 2017. <https://db-blog.web.cern.ch/blog/joeri-hermans/2017-01-distributed-deep-learning-apache-spark-and-keras>.
- [38] Oliveira, L. et al. “Learning Particle Physics by Example: Location-Aware Generative Adversarial Networks for Physics Synthesis,” *arXiv* (June 2017). <https://arxiv.org/abs/1701.05927>.

- [39] Chen, T. and T. He. “Higgs Boson Discovery with Boosted Trees.” *JMLR: Workshop and Conference Proceedings* 42 (2015) 69–80. <http://proceedings.mlr.press/v42/chen14.pdf>.
- [40] Aurisano, A. et al. “A Convolutional Neural Network Neutrino Event Classifier,” *arXiv* (August 2016). <https://arxiv.org/abs/1604.01444v3>.
- [41] Hezaveh, Y. et al. “Fast automated analysis of strong gravitational lenses with convolutional neural networks,” *Nature* 548 (August 2017): 555–557. <https://www.nature.com/articles/nature23463>.
- [42] Shallue, Chris and Andrew Vanderburg, “Earth to exoplanet: Hunting for planets with machine learning,” Google blog, December 14, 2017. <https://www.blog.google/topics/machine-learning/hunting-planets-machine-learning/>.
- [43] Vijayan, Jaikumar, “TensorFlow Helps NASA Discover New Exoplanet,” *eWeek*, December 14, 2017. <http://www.eweek.com/big-data-and-analytics/google-machine-learning-technology-helps-nasa-discover-new-exoplanet>.
- [44] “First-Generation Robot Scientist to Search for Underlying Laws of Nature,” *Daily Galaxy*, August 25, 2015. http://www.dailygalaxy.com/my_weblog/2015/08/robot-scientist-to-search-for-laws-of-nature-that-underlie-the-universe.html.
- [45] Choi, Charles Q., “Physicists Unleash AI to Devise Unthinkable Experiments,” *Scientific American*, March 22, 2016. <https://www.scientificamerican.com/article/physicists-unleash-ai-to-devise-unthinkable-experiments/>.
- [46] Steinke, Steven, “Solving the Schrödinger equation with deep learning,” *Becoming Human* blog post, September 27, 2017. <https://becominghuman.ai/solving-schrödingers-equation-with-deep-learning-f9f6950a7c0e>.
- [47] Mills, K. et al., “Deep learning and the Schrödinger equation,” *Phys. Rev. A* 96, Issue 4 (October 2017). <https://journals.aps.org/pra/abstract/10.1103/PhysRevA.96.042113>.
- [48] L. Wang. “Discovering phase transitions with unsupervised learning” *Phys. Rev. B* 94, 195105 (2016).
- [49] Farimani, A.B. et al., “Deep Learning the Physics of Transport Phenomena,” *arXiv* (September 2017). <https://arxiv.org/abs/1709.02432>.
- [50] Denil, M. et al., “Learning to Perform Physics Experiments via Deep Reinforcement Learning,” *arXiv* (August 2017). <https://arxiv.org/abs/1611.01843>.
- [51] Trifacta. <https://www.trifacta.com>.
- [52] Talend. <https://www.talend.com/products/data-preparation/>.
- [53] Tong, Kester et al., “Preprocessing for Machine Learning with tf.Transform,” Google Research blog, February 22, 2017. <https://research.googleblog.com/2017/02/preprocessing-for-machine-learning-with.html>.

- [54] Kobielus, James, “Developers Will Adopt Sophisticated AI Model Training Tools in 2018,” *Datanami*, January 3, 2018. <https://www.datanami.com/2018/01/03/developers-will-adopt-sophisticated-ai-model-training-tools-2018/>.
- [55] Jaderberg, Max, “Population based training of neural networks,” Deepmind blog, November 27, 2017. <https://deepmind.com/blog/population-based-training-neural-networks/>.
- [56] Sabour, S. et al., “Dynamic Routing Between Capsules,” *arXiv* (November 2017). <https://arxiv.org/abs/1710.09829>.
- [57] Hassabis et al., “Neuroscience-Inspired Artificial Intelligence,” *Neuron* 95, no. 2 (July 2017): 245–258. [http://www.cell.com/neuron/abstract/S0896-6273\(17\)30509-3](http://www.cell.com/neuron/abstract/S0896-6273(17)30509-3).
- [58] Marblestone et al., “Toward an Integration of Deep Learning and Neuroscience,” *Front Comput Neurosci*, (September 2016). <https://www.ncbi.nlm.nih.gov/pubmed/27683554?dopt=Abstract>.
- [59] Richards et al., “Towards Deep Learning with Segregated Dendrites,” *eLife* (December 2017). <https://elifesciences.org/articles/22901>.
- [60] Gardner, Howard. *Frames of Mind: The Theory of Multiple Intelligences*, 3rd ed. (New York: Basic Books, 2011).
- [61] Marcus, Gary. “Deep Learning: A Critical Appraisal.” *arXiv* (January 2018). <https://arxiv.org/abs/1801.00631>.
- [62] Rabinowitz, N. C. et al. “Machine Theory of Mind.” *arXiv* (February 2018). <https://arxiv.org/abs/1802.07740>.
- [63] Friston, K. J. et al. “Active Inference, Curiosity and Insight.” *Neural Computation* 29, no. 10 (October 2017): 2633–2683. https://www.mitpressjournals.org/doi/abs/10.1162/neco_a_00999.

About the Author

Peter Morgan is founder and CEO of **Deep Learning Partnership**, a company that consults and trains on the latest in deep learning and artificial intelligence algorithms and full-stack solutions. He also runs the **Deep Learning Lab** meetup group in London. Peter is currently authoring a book on quantum computing for Springer and a paper on active inference, a general theory of intelligence, with Professor Karl Friston at the UCL. In a past life, Peter was a theoretical high-energy physicist and a solutions architect for companies such as IBM, BT Labs, and Cisco Systems. He enjoys frisbee and golfing. You can find more information about Peter on his [LinkedIn profile](#).