1. AI, ML, and DL

1.1. Artificial Intelligence

- Concisely, AI can be described as the effort to automate intellectual tasks normally performed by humans.
    - ➢ In fact, for a fairly long time, most experts believed that human-level artificial intelligence could be achieved by having programmers handcraft a sufficiently large set of explicit rules for manipulating knowledge stored in explicit databases.
        - ✧ This is known as ***Symbolic***.
    - ➢ Although symbolic AI proved suitable to solve well-defined, logical problems, such as playing chess, it turned out to be intractable to figure out explicit rules for solving more complex, fuzzy problems, such as image classification, speech recognition, or natural language translation
    - ➢ A new approach arose to take symbolic AI's place: ***Machine Learning***.

1.2. Machine Learning

- Although visionary and far ahead of its time, the Analytical Engine (first-known general-purpose mechanical computer) wasn't meant as a general-purpose computer when it was designed in the 1830s and 1840s, because the concept of general-purpose computation was yet to be invented.
    - ➢ It was merely meant as a way to use mechanical operations to automate certain computations from the field of mathematical analysis – hence the name Analytical Engine

- Lady Lovelace's (friend and collaborator of Charles Babbage, the inventor of Analytical Engine) questioned: "could a general-purpose computer "originate" anything, or would it always be bound to dully execute processes we humans fully understand? Could it ever be capable of any original thought? Could it learn from experience? Could it show creativity?"
    - ➢ Alan Turing later took this remark and introduced the ***Turing test***. He had the opinion that computers could in principle be made to emulate all aspects of human intelligence

- The usual way to make a computer do useful work is to have a human programmer write down rules – a computer program – to be followed to turn input data into appropriate answers
    - ➢ Machine learning turns this around: the machine looks at the input data and the corresponding answers, and figure out what the rules should be.

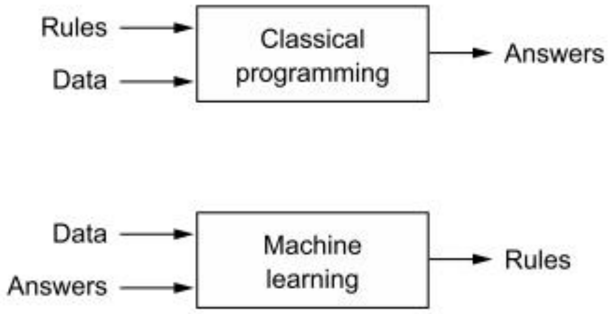


Figure 1.2 Machine learning: a new programming paradigm

- A machine learning system is trained rather than explicitly programmed
  - It's presented with many examples relevant to a task, and it finds statistical structure in these examples that eventually allows the system to come up with rules for automating the task

- However, unlike statistics, machine learning tends to deal with large, complex datasets (e.g. a dataset of millions of images, each consisting of tens of thousands of pixels) for which classical statistical analysis such as Bayesian analysis would be impractical
  - As such, machine learning, and especially deep learning, exhibits comparatively little mathematical theory – maybe too little – and is fundamentally and engineering discipline

1.3. Learning rules and representations from data

- A machine learning model transforms its input data into meaningful outputs
  - This is a process that is "learned" from exposure to known examples of inputs and outputs
  - Therefore, the central problem in machine learning and deep learning is to meaningfully transform data
    - i.e. to learn useful representations of the input data at hand – representations that get us closer to the expected output
  - machine learning models are all about finding appropriate representations for their input data – transformations of the data that make it more amenable to the task at hand.

- Finding useful representation by hand is hard, the resulting rule-based system is brittle
  - Every time you come across a new example of new representation that breaks your carefully thought-out rules, you will have to add new data transformations and new rules, while taking into account their interaction with every previous rule
  - That's why we should automate it, we could try systematically searching for different sets of automatically generated representations of the data and rules based on them, identifying good ones by using as feedback the percentage of digits being correctly classified in some development dataset. This is machine learning

- Machine learning algorithms aren't usually creative in finding transformations; they are merely searching through a predefined set of operations, called a ***hypothesis space***.
  - Precisely, machine learning is searching for useful representations and rules over some input data,

within a predefined space of possibilities, using guidance from a feedback signal.

1.4. The "deep" in deep learning

- Deep learning is a specific subfield of machine learning: a new take on learning representations from data that puts an emphasis on learning successive layers of increasingly meaningful representations.
    - How many layers contribute to a model of the data is called the ***depth*** of the model
    - Other appropriate names for the field could have been *layered representations learning* or *hierarchical representations learning*.
    - Modern deep learning often involves tens or even hundreds of successive layers of representations, and they're all learned automatically from exposure to training data.

- In deep learning, these layered representations are learned via models called ***neural networks***, structured in literal layers stacked on top of each other.
    - The term "neural network" refers to neurobiology.
        - But although some of the central concepts in deep learning were developed in part of drawing inspiration from our understanding of the brain (visual cortex in particular), deep learning models are not models of the brain.
        - There is no evidence that the brain implements anything like the learning mechanisms used in modern deep learning models.

- In the following example

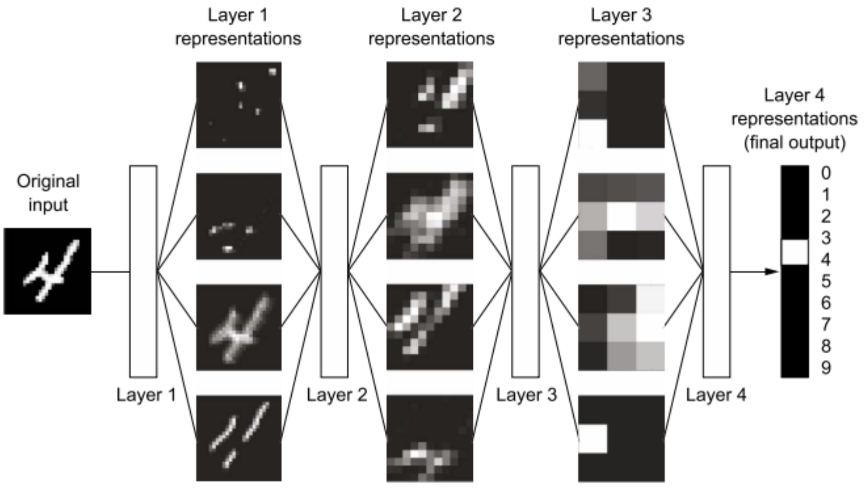


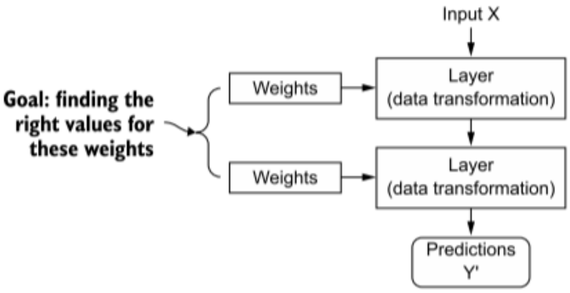
Figure 1.6 **Data representations learned by a digit-classification model**

- The network transforms the digit image into representations that are increasingly different from the original image and increasingly informative about the final result.
    - You can think of a deep network as a multistage *information distillation* process, where information goes through successive filters and comes out increasingly *purified*.
- It's a simple idea – but, as it turns out, very simple mechanisms, sufficiently scaled, can end up

looking like magic

1.5. Understanding how dep learning works

- ***Weights***
  - The specification of what a layer does to its input data is stored in the layer's weights, which in essence are a bunch of numbers.



Input X
Goal: finding the right values for these weights
Weights
Layer (data transformation)
Weights
Layer (data transformation)
Predictions Y'

Figure 1.7 A neural network is parameterized by its weights.

  - In technical terms, we'd say that the transformation implemented by a layer is ***parameterized*** by its weights. (Weights are also sometimes called the ***parameter*** of a layer)
  - In this context, ***learning*** means finding a set of values for the weights of all layers in a network, such that the network will correctly map example inputs to their associated targets.
    - A deep neural network can contain tens of millions of parameters. Finding the correct values for all of them may seem like a daunting task, especially given that modifying the value of one parameter will affect the behavior of all the others.

- ***Loss function***
  - To control the output of a neural network, you need to be able to measure how far this output is from what you expected. This is the job of the ***loss function*** of the network.
  - The loss function takes the predictions of the network and the true target and computes a distance score, capturing how well the network has done on this specific example

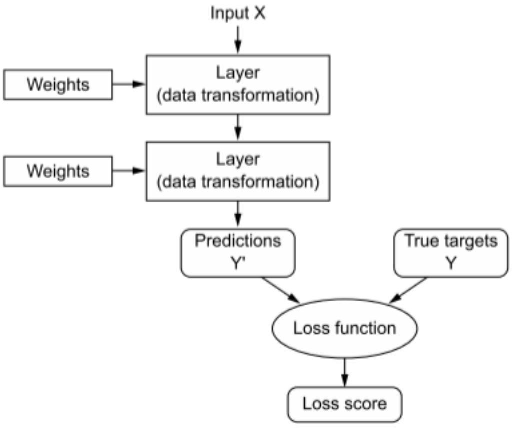


Figure 1.8 A loss function measures the quality of the network's output.

- ***Optimizer***
  - The fundamental trick in deep learning is to use this score as a feedback signal to adjust the value of the weights a little, in a direction that will lower the loss score for the current example.

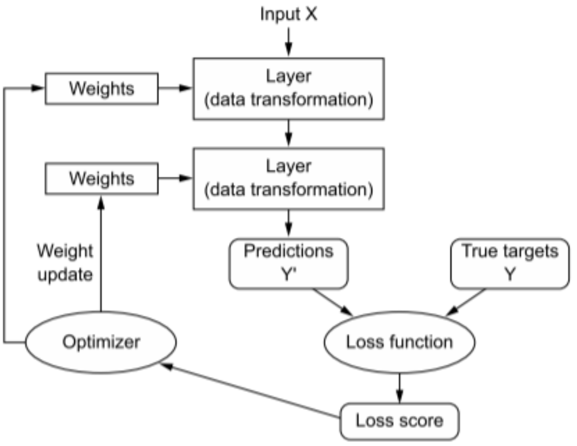


Figure 1.9 The loss score is used as a feedback signal to adjust the weights.

  - This adjustment is the job of the ***optimizer***, which implements what's called the ***Backpropagation*** algorithm: the central algorithm in deep learning.

- The process
  - Initially, the weights of the network are assigned random values, so the network merely implements a series of random transformations.
    - Naturally, its output is far from what it should ideally be, and the loss score is accordingly very high.
  - With every example the network processes, the weights are adjusted a little in the correct direction, and the loss score decreases.
    - This is the ***training loop***, which, repeated a sufficient number of times (typically tens of iterations over thousands of examples), yields weight values that minimize the loss function.
    - A network with a minimal loss is one for which the outputs are as close as they can be to the targets: a trained network.
    - It's a simple mechanism that, once scaled, ends up looking like magic

1.6. What deep learning has achieved so far

- Deep learning has enabled the following breakthroughs, all in historically difficult areas of machine learning
  - Near-human-level image classification
  - Near-human-level speech transcription
  - Near-human-level handwriting transcription
  - Dramatically improved machine translation
  - Dramatically improved text-to-speech conversion
  - Digital assistants such as Google Assistant and Amazon Alexa
  - Near-human-level autonomous driving

- Improved ad targeting, as used by Google, Baidu, or Bing
- Improved search results on the web
- Ability to answer natural language questions
- Superhuman Go playing

1.7. Don't believe that short-term hype

- Although deep learning has led to remarkable achievements in recent years, expectations for what the field will be able to achieve in the next decade tend to run much higher than what will likely be possible
  - The risk with high expectations for the short term is that, as technology fails to deliver, research investment will dry up, slowing progress for a long time

- In the 1960s where symbolic AI was first introduced, projections about AI were flying high.
  - One of the best-known pioneers and proponents of the symbolic AI approach was Marvin Minsky, who claimed in 1967, "Within a generation … the problem of creating "artificial intelligence" will substantially be solved". 3 years later, in 1970, he made a more precisely quantified prediction: "in from 3 to 8 years we will have a machine with the general intelligence of an average human being"
  - A few years later, as these high expectations failed to materialize, researchers and government funds turned away from the field, making the start of first AI ***winter***.

- In the 1980s, a new take on symbolic AI, ***expert systems***, started gathering steam among large companies
  - A few initial success stories triggered a wave of investment, which corporations around the world starting their own in-house AI departments to develop expert systems.
  - These systems had proven expensive to maintain, difficult to scale, and limited in scope, and interest died down. Thus begun the second AI ***winter***.

- We may be currently witnessing the third cycle of AI hype and disappointment, and we're still in the phase of intense optimism.

1.8. The promise of AI

- AI research has been moving forward amazingly quickly in the past ten years, in large part due to a level of funding never before seen in the short history of AI, but so far relatively little of this progress has made its way into the products and processes that form our world
  - Most of the research findings of deep learning aren't yet applied, or at least are not applied to the full range of problems they could solve across all industries

- AI is coming, in a not-so-distant future.
  - AI will be your assistant, even your friend

  - ➢ It will answer your questions, help educate your kids, and watch over your health
  - ➢ It will deliver your groceries to your door and drive you from point A to point B.
  - ➢ It will be your interface to an increasingly complex and information-intensive world
  - ➢ Even more importantly, AI will help humanity as a whole move forward, by assisting human scientists in new breakthrough discoveries across all scientific fields, from genomics to mathematics

- Don't believe the short-term hype, but do believe in the long-term vision. It may take a while for AI to be deployed to its true potential – a potential the full extent of which no one has yet dared to dream – but AI is coming, and it will transform our world in a fantastic way

2. A brief history of machine learning

- It's safe to say that most of the machine learning algorithms used in the industry today aren't deep learning algorithms.
  - ➢ Deep learning isn't always the right tool for the job – sometimes there isn't enough data for deep learning to be applicable, and sometimes the problem is better solved by a different algorithm

2.1. Early Neural Networks

- Although the core ideas of neural networks were investigated in toy forms as early as the 1950s, the approach took decades to get started
  - ➢ For a long time, the missing piece was an efficient way to train large neural networks.
  - ➢ This changed in the mid-1980s, when multiple people independently rediscovered the Backpropagation algorithm – a way to train chains of parametric operations using gradient-descent optimization and started applying it to neural networks

- The first successful practical application of neural nets came in 1989 from Bell Labs, when Yann LeCun combined the earlier ideas of convolutional neural networks and backpropagation, and applied them to the problem of classifying handwritten digits.
  - ➢ The resulting network, dubbed ***LeNet***, was used by the US Postal Service in the 1990s to automate the reading of ZIP code on mail envelopes

2.2. Kernel Methods

- As neural networks started to gain some respect among researchers in the 1990s, thanks to the first success, a new approach to machine learning rose to fame and quickly sent neural nets back to oblivion:

Kernel methods

- ***Kernel Methods*** are group of classification algorithms, the best known of which is the ***Support Vector Machine*** (SVM).
- The modern formulation of an SVM was developed by Vladimir Vapnik and Corinna Cortes in the early 1990s at Bell Labs and published in 1995.

- SVM is a classification algorithm that works by finding "decision boundaries" separating two classes. SVMs proceed to find these boundaries in two steps

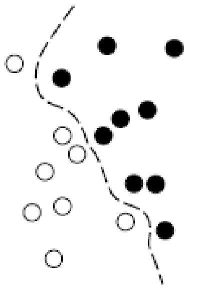


Figure 1.10
A decision boundary

  - 1. The data is mapped to a new high-dimensional representation where the decision boundary can be expressed as a hyperplane
    - If the data was 2 dimensional, a hyperplane would be a single line
  - 2. A good decision boundary (a separation hyperplane) is computed by trying to maximize the distance between the hyperplane and the closest data points from each class, a step called ***maximizing the margin***. This allows the boundary to generalize well to new samples outside of the training dataset

- The technique of mapping data to a high-dimensional representation where a classification problem becomes simpler may look good on paper, but in practice it's often computationally intractable. That's where the ***kernel trick*** comes in (the key idea that kernel methods are named after)
  - The gist is that: to find good decision hyperplanes in the new representation space, you don't have to explicitly compute the coordinates of your points in the new space; you just need to compute the distance between pairs of points in that space, which can be done efficiently using a kernel function
  - A ***kernel function*** is a computationally tractable operation that maps any two points in your initial space to the distance between these points in your target representation space, completely bypassing the explicit computation of the new representation.
    - Kernel functions are typically crafted by hand rather than learned from data – in the case of an SVM, only the separation hyperplane is learned.

- At the time they were developed, SVMs exhibited state-of-the-art performance on simple classification problems and were one of the few machine learning methods backed by extensive theory and amenable to serious mathematical analysis, making them well understood and easily interpretable. Because of these useful properties, SVMs became extremely popular in the field for a long time
  - But SVMs proved hard to scale to large datasets and didn't provide good results for perceptual

problems such as image classification.

- Because an SVM is a shallow method (focus on learning only one or two layers of representations of the data), applying an SVM to perceptual problems requires first extracting useful representations manually (a step called ***feature engineering***), which is difficult and brittle
- For example, if you want to use an SVM to classify handwritten digits, you can't start from the raw pixels; you should first find by hand useful representations that make the problem more tractable, like the pixel histograms mentioned earlier.

2.3. Probabilistic Modelling

- Probabilistic modelling is the application of the principles of statistics to data analysis. This is one of the earliest forms of machine learning, and it's still widely used to this day.
  - One of the best-known algorithms in the category is the ***Naïve Bayes algorithm***
    - This is a type of machine learning classifier based on applying Bayes' theorem while assuming that the features in the input data are all independent (a strong, or "naive" assumption, which is where the name comes from)
    - This form of data analysis predates computers and was applied by hand decades before its first computer implementation (most likely dating back to the 1950s)
  - A closely related model is ***logistic regression*** (logreg for short), which is sometimes considered to the "Hello World" of modern machine learning.
    - Much like Naïve Bayes, logreg predates computing by a long time, yet it's still useful to this day, thanks to its simple and versatile nature. It's often the first thing a data scientist will try on a dataset to get a feel for the classification task at hand.

2.4. Tree-Based Methods

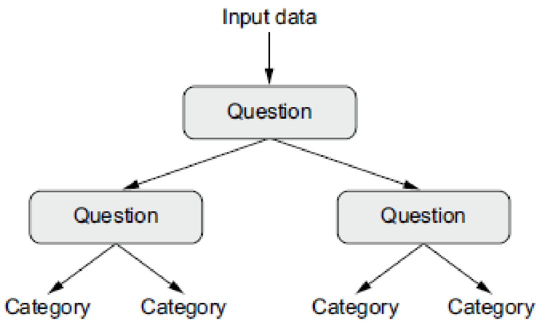


Figure 1.11 A decision tree: the parameters that are learned are the questions about the data. A question could be, for instance, "Is coefficient 2 in the data greater than 3.5?"

- ***Decision Trees*** are flowchart-like structures that let you classify input data points or predict output values given inputs. They are easy to visualize and interpret.
  - Decision trees learned from data began to receive significant research interest in the 2000s, and by 2010 they were often preferred to kernel methods

- In particular, the ***Random Forest*** algorithm introduced a robust, practical take on decision-tree learning that involves building a large number of specialized decision trees and then ensembling their outputs.

  - Random forests are applicable to a wide range of problems – they are almost always the second-best algorithm for any shallow machine learning task.

- When the popular machine learning competition website Kaggle got started in 2010, random forests quickly became a favorite on the platform – until 2014, when ***gradient boosting machines*** took over
  - A gradient boosting machine, much like a random forest, is a machine learning technique based on ensembling weak prediction models, generally decision trees.
    - It uses ***gradient boosting***, a way to improve any machine learning model by iteratively training new models that specialize in addressing the weak points of the previous models.
  - Applied to decision trees, the use of the gradient boosting technique results in models that strictly outperform random forests most of the time, while having similar properties.
  - It may be one of the best, if not the best, algorithm for dealing with non-perceptual data today

2.5. Back to Neural Networks

- Around 2010, although neural networks were almost completely shunned by the scientific community at large, a number of people still working on neural networks started to make important breakthroughs
  - The groups of Geoffrey Hinton at the University of Toronto
  - Yoshua Bengio at the University of Montreal
  - Yann LeCun at New Yort University
  - IDSIA in Switzerland

- In 2011, Dan Ciresan from IDSIA began to win academic image-classification competitions with GPU-trained deep neural networks – the first practical success of modern deep learning.
- But the watershed moment came in 2012. With the entry of Hinton's group in the yearly large-scale image-classification challenge ImageNet.
  - The ImageNet challenge was notoriously difficult at the time, consisting of classifying high-resolution color images into 1,000 different categories after training on 1.4 million images.
    - Top-five accuracy measure how often the model selects the correct answer as part of its top five guesses (out of 1,000 possible answers)
  - In 2011, the top-five accuracy of the wining model, based on classical approaches to computer vision, was only 74.3%
  - In 2012, a team led by Alex Krizhevsky and advised by Geoffrey Hinton was able to achieve a top-five accuracy of 83.6% - a significant breakthrough.
  - The competition has been dominated by deep convolutional neural networks every year since.
  - By 2015, the winner reached an accuracy of 96.4%, and the classification task on ImageNet was considered to be a completely solved problem.

- Since 2012, deep convolutional neural networks (convnets) have become the go-to algorithm for all computer vision tasks; more generally, they work on all perceptual tasks
  - At any major computer vision conference after 2015, it was nearly impossible to find presentations that didn't involve convnets in some form.

- At the same time, deep learning has also found applications in many other types of problems, such as natural language processing.
    - ➢ It has completely replaced SVMs and decision trees in a wide range of applications.
        - ✧ E.g. for several years, the European Organization for Nuclear Research, CERN, used decision tree-based methods for analyzing particle data from the ATLAS detector at the Large Hadron Collider (LHC), but CERN eventually switched to Keras-based deep neural networks due to their higher performance and ease of training on large datasets.

2.6. What makes deep learning different

- The primary reason deep learning took off
    - ➢ Offered better performance
    - ➢ More importantly, makes problem-solving much easier, because it completely automates what used to be the most crucial step in a machine learning workflow: feature engineering
        - ✧ Previous machine learning techniques – shallows learning – only involved transforming the input data into one or two successive representation spaces, usually via simple transformation such as high-dimensional non-linear projections (SVMs) or decision trees.
        - ✧ Humans had to go to great lengths to make the initial input data more amenable to processing by these models: they had to manually engineer good layers of representations for their data. This is called ***feature engineering***.
        - ✧ Deep learning, on the other hand, completely automates this step: with deep learning, you learn all features in one pass rather than having to engineer them yourself. This has greatly simplified machine learning workflows, often replacing sophisticated multistage pipelines with a single, simple, end-to-end deep learning model.

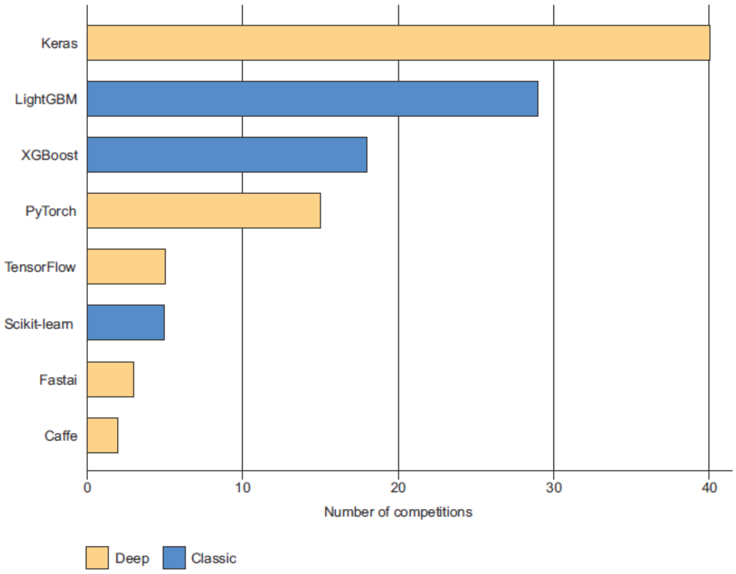
- Could shallow methods be applied repeatedly to emulate the effects of deep learning?
    - ➢ In practice, successive applications of shallow-learning methods produce fast-diminishing returns, because the optimal first representation layer in the three-layer model isn't the optimal first layer in a one-layer or two-layer model
    - ➢ What is transformative about deep learning is that it allows a model to learn all layers of representation **jointly**, at the same time, rather than in succession (***greedily***, as it is called)
        - ✧ With joint feature learning, whenever the model adjusts one of its internal features, al other features that depend on it automatically adapt to the change, without requiring human intervention.
        - ✧ Everything is supervised by a single feedback signal: every change in the model serves the end goal. This is much more powerful than greedily stacking shallow models, because it allows for complex, abstract representations to be learned by breaking them down into long series of intermediate spaces (layers); each space is only a simple transformation away from the previous one.

- These are the two essential characteristics of how deep learning learns from data
    - ➢ The incremental, layer-by-layer way in which increasingly complex representations are developed
    - ➢ These intermediate incremental representations are learned jointly

- ✧ Each layer being updated to follow both the representational needs of the layer above and the needs of the layer below.

2.7. The modern machine learning landscape

- In early 2019, Kaggle ran a survey asking teams that ended in the top five of any competition since 2017 which primary software tool they had used in the competition.
  - ➢ They tend to use either deep learning methods (most often via the Keras library) or gradient boosted trees (most often via the LightGBM or XGBoosting libraries)



- It's not just competition champions, either. Kaggle also runs a yearly survey among machine learning and data science professionals worldwide.
  - ➢ With tens of thousands of respondents, this survey is one of our most reliable sources about the state of the industry.
  - ➢ The following figure shows the percentage of usage of different machine learning software framework

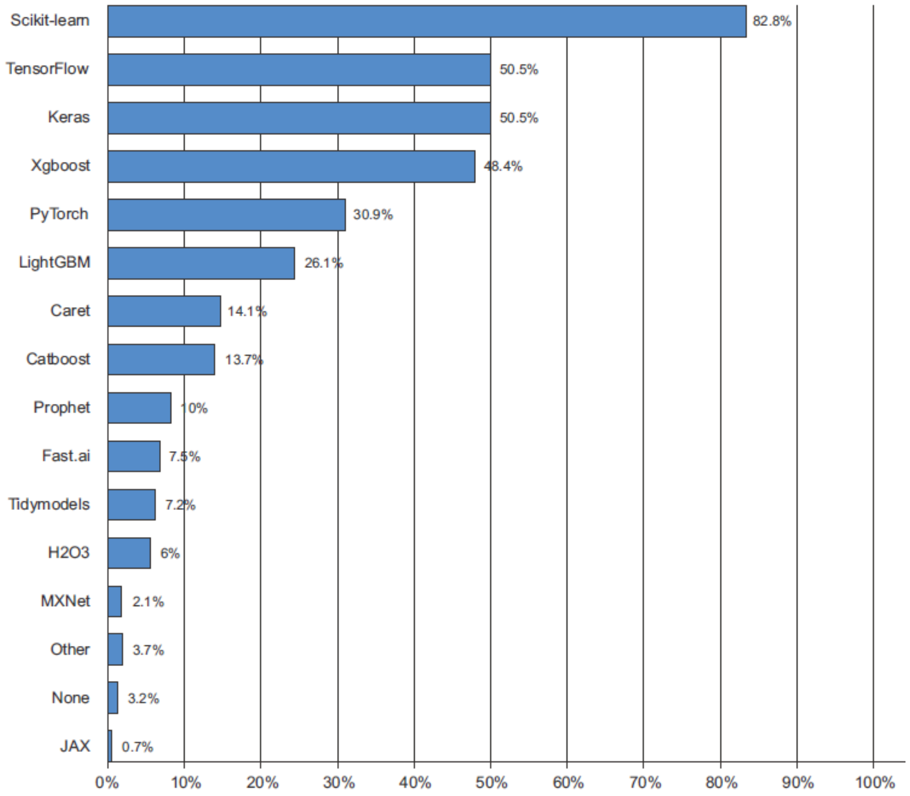


Figure 1.13 Tool usage across the machine learning and data science industry (Source: www.kaggle.com/kaggle-survey-2020)

- From 2016 to 202, the entire machine learning and data science industry has been dominated by theses two approaches: deep learning and gradient boosted trees.
  - Specifically, gradient boosted trees is used for problems where structured data is available
    - People tend to use Scikit-learn, XGBoost, or LightGBM.
  - Deep learning is used for perceptual problems such as image classification
    - Most practitioners of deep learning use Keras, often in combination with its parent framework TensorFlow.
- Those tools are all Python libraries: Python is by far the most widely used language for machine learning and data science.

3. *Why deep learning?*

- The two key ideas of deep learning for computer vision – convolutional neural networks and backpropagation – were already well understood by 1990.
- The Long Short-Term Memory (LSTM) algorithm, which is fundamental to deep learning for timeseries, was developed in 1997 and has barely changed since.
- Why did deep learning only take off after 2012?
  - In general, 3 technical forces are driving advances in machine learning
    - Hardware
    - Datasets and benchmarks

        - ✧ Algorithmic advances
    - ➢ Because the field is guided by experimental findings rather than by theory, algorithmic advances only become possible when appropriate data and hardware are available to try new ideas (or to scale up old ideas, as is often the case).
        - ✧ Machine learning isn't mathematics or physics, where major advances can be done with a pen and a piece of paper. It's engineering science
    - ➢ The real bottlenecks throughout the 1990s and 2000s were data and hardware, but here's what happened during that time: the internet took off and high-performance graphics chips were developed for the needs of the gaming market

3.1. Hardware

- Between 1990 and 2010, off-the-shelf CPUs became faster by a factor of approximately 5,000. As a result, nowadays it's possible to run small deep learning models on your laptop, whereas this would have been intractable 25 years ago.
- But typical deep learning models used in computer vision or speech recognition require orders of magnitude more computational power than your laptop can deliver
    - ➢ Throughout the 2000s, companies like NVIDIA and AMD invested billions of dollars in developing fast, massively parallel chips (graphical processing unit, or GPUs) to power the graphics of increasingly photorealistic video games – cheap, single-purpose supercomputers designed to render complex 3D scenes on your screen in real time.
        - ✧ This investment came to benefit the scientific community when, in 2017, NVIDIA launched CUDA, a programming interface for its line of GPUs
    - ➢ A small number of GPUs started replacing massive clusters of CPUs in various highly parallelizable applications, beginning with physics modeling.
        - ✧ Deep neural networks, consisting mostly of many small matrix multiplications, are also highly parallelizable, and around 2011 some researchers began to write CUDA implementations of neural nets – Dan Ciresan and Alex Krizhevsky were among the first.

- What happened is that the gaming market subsidized supercomputing for the next generation of AI applications.
    - ➢ Sometimes, big things begin as games.
    - ➢ Today, the NVIDIA Titan RTX can deliver a peak of 16 teraFLOPS in single precision (16 trillion float32 operations per second), that is about 500 times more computing power than the world's fastest supercomputer from 1990, the Intel Touchstone Delta.
        - ✧ On a Titan RTX, it takes only a few hours to train an ImageNet model of the sort that would have won the ILSVRC competition around 2012 or 2013. Meanwhile, large companies train deep learning models on clusters of hundreds of GPUs

- What's more, the deep learning industry has been moving beyond GPUs and is investing in increasingly specialized, efficient chips for deep learning
    - ➢ In 2016, as its annual I/O convention, Google revealed its Tensor Processing Unit (TPU) project: a new chip design developed from the ground up to run deep neural networks significantly faster

and far more energy efficient than top-of-the-line GPUs.

- ➢ Today, in 2020, the third iteration of the TPU card represents 420 teraFLOPS of computing power. That is 10,000 times more than the Intel Touchstone Delta from 1990
- ➢ These TPU cards are designed to be assembled into large-scale configurations, called “pods”.
  - ✧ One pod (1024 TPU cards) peaks at 100 petaFLOPS. For scale, that’s about 10% of the peak computing power of the current largest supercomputer, the IBM Summit at Oak Ridge National Lab, which consists of 27,000 NVIDIA GPUs and peaks at around 1.1 exaFLOPS.

3.2. Data

- If deep learning is the steam engine of this revolution, then data is its coal: the raw material that powers our intelligent machines, without which nothing would be possible.
  - ➢ We have exponential progress in storage hardware over the past 20 years (following Moore’s law)
  - ➢ The game changer has been the rise of the internet, making it feasible to collect and distribute very large datasets for machine learning.

3.3. Algorithms

- Until the late 2000s, we were missing a reliable way to train very deep neural networks. As a result, neural networks were still fairly shallow, using only one or two layers of representation; thus, they weren’t able to shine against more-refined shallow methods such as SVMs and random forests
  - ➢ The key issue was that of gradient propagation through deep stacks of layers. The feedback signal used to train neural networks would fade away as the number of layers increased.

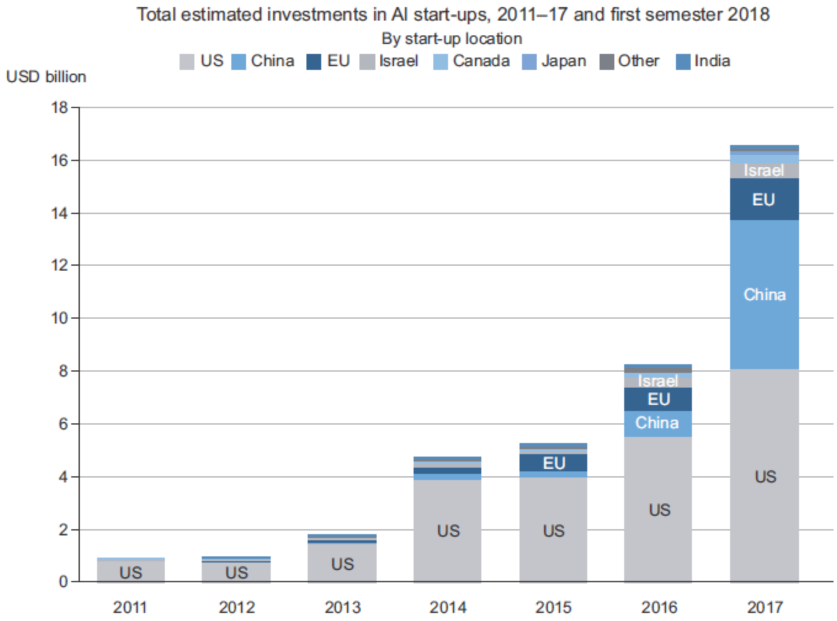
- This changed around 2009-2010 with the advent of several simple but important algorithmic improvements that allowed for better gradient propagation:
  - ➢ Better ***activation function*** for neural layers
  - ➢ Better ***weight-initialization schemes***, starting with layer-wise pretraining, which was then quickly abandoned
  - ➢ Better ***optimization schemes***, such as RMSProp and Adam

- Only when these improvements began to allow for training models with 10 or more layers did deep learning start to shine

- Finally, in 2014, 2015, and 2016, even more advanced ways to improve gradient propagation were discovered, such as batch normalization, residual connections, and depthwise separable convolutions

3.4. A new wave of investment

- As deep learning became the new state of the art from computer vision in 2012-2013, and eventually for all perceptual tasks, industry leaders took note. What followed was a gradual wave of industry investment far beyond anything previously seen in the history of AI



Total estimated investments in AI start-ups, 2011–17 and first semester 2018

By start-up location

3.5. The democratization of deep learning

- One of the key factors driving this inflow of new faces in deep learning has been the democratization of the toolsets used in the field
  - In the early days, doing deep learning required significant C++ and CUDA expertise, which few people possessed.
  - Nowadays, basic Python scripting skills suffice to do advanced deep learning research.
    - This has been driven most notably by the development of the now-defunct Theano library, and then the TensorFlow library – two symbolic tensor-manipulation frameworks for Python that support autodifferentiation, greatly simplify the implementation of new models – and by the rise of user-friendly libraries such as Keras, which makes deep learning as easy as manipulating LEGO bricks.

3.6. Will it last?

- Deep learning has several properties that justify its status as an AI revolution, and it’s here to stay. We may not use neural networks two decades from now, but whatever we use will directly inherit from modern deep learning and its core concepts.

- These important properties can be broadly sorted into three categories:
  - ➢ Simplicity – deep learning removes the need for feature engineering, replacing complex, brittle, engineering-heavy pipelines with simple, end-to-end trainable models that are typically built using only five or six different tensor operations
  - ➢ Scalability – deep learning is highly amenable to parallelization on GPUs or TPUs, so it can take full advantage of Morre’s law.
    - ✧ In addition, deep learning models are trained by iterating over small batches of data, allowing them to be trained on datasets of arbitrary size
  - ➢ Versatility and reusability – unlike many prior machine learning approaches, deep learning models can be trained on additional data without restarting from scratch, making them visible for continuous online learning.
    - ✧ Furthermore, trained deep learning models are repurposable and thus reusable; for example, it is possible to take a deep learning model trained for image classification and drop it into a video processing pipeline
    - ✧ This allows us to reinvest previous work into increasingly complex and powerful models. This also makes deep learning applicable to fairly small datasets