1 When and How to Use Machine Learning

1.1 Overview

This chapter aims to motivate the study of machine learning, having in mind as the intended audience students and researchers with an engineering background.

Learning Objectives and Organization of the Chapter. By the end of this chapter, the reader should be able to addresses the following basic questions:

- What is machine learning for?
- Why are machine learning and "AI" so popular right now?
- What can engineers contribute to machine learning?
- What is machine learning?
- When to use machine learning?

Each of these questions will be addressed in a separate section in the rest of this chapter.

1.2 What is Machine Learning For?

Machine learning is currently the dominant form of **artificial intelligence (AI)** – so much so that the label "AI" has by now become synonymous with the data-driven pattern recognition methods that are the hallmark of machine learning. Machine learning algorithms underlie a vast array of applications, with real-life implications for individuals, companies, and governments. Here are some examples:

- Governments use it to decide on visa applications.
- Courts deploy it to rule on requests for bail.
- Police departments rely on it to identify suspects.
- Schools and universities apply it to assign places.
- Banks deploy it to grant or deny credit.
- Financial institutions run it to operate in the stock market.
- Companies apply it to optimize their hiring decisions.
- Online vendors leverage it to provide recommendations and track users' preferences.
- Individuals interact with it when "conversing" with virtual personal assistants and finding the best route to a destination. (Quotation marks seem necessary when using verbs that imply intentionality to algorithms.[1])

Not all cases of machine learning or AI use are equally legitimate, successful, or even morally justifiable. The "algorithm" (another necessary use of quotation marks) relied on

[1] The story of the ELIZA computer program provides an interesting cautionary tale.

by the UK government to predict students' marks in 2020 was actually based on a single, poorly thought-out, formula. Machine learning-based face-recognition tools are notorious for being inaccurate and biased. So are many suspect-identifying, credit-scoring, and candidate-selecting tools based on machine learning that have been widely reported to produce unfair outcomes.

This discussion points to an important distinction: It is one thing to design technically sound data-driven solutions, and it is another thing altogether to ensure that these solutions are deployed to benefit society. While this book will focus on the technical problem, there is clearly a need for engineers to work alongside regulators and social scientists to ensure that proper legal and ethical guidelines are applied. In the words of the American statistician Nate Silver, "The numbers have no way of speaking for themselves. We speak for them, we imbue them with meaning".

1.3 Why Study Machine Learning Now?

In 1950, **Claude Shannon**, the father of information theory, published a seminal paper entitled *Programming a Computer for Playing Chess*. In the paper, he envisions

(1) Machines for designing filters, equalizers, etc.
(2) Machines for designing relay and switching circuits.
(3) Machines which will handle routing of telephone calls based on the individual circumstances rather than by fixed patterns.
(4) Machines for performing symbolic (non-numerical) mathematical operations.
(5) Machines capable of translating from one language to another.
(6) Machines for making strategic decisions in simplified military operations.
(7) Machines capable of orchestrating a melody.
(8) Machines capable of logical deduction.

Despite being over 70 years old, this passage reads almost like a summary of recent breakthroughs in machine learning. In the same year, **Alan Turing**, writing for the Conference on Information Theory, explained the idea of learning machines as follows: "If, as is usual with computing machines, the operations of the machine itself could alter its instructions, there is the possibility that a learning processing could by this means completely alter the programme in the machine." Using language that has hardly aged, Turing outlines what computer scientists would describe today as **"Programming 2.0"**: Programmers no longer need to explicitly code algorithmic steps to carry out a given task; instead, it is the machine that automatically infers the algorithmic steps from data. Most commonly, data are in the form of examples that "supervise" the machine by illustrating the desired relationship between input and some target variables or actions; think, for instance, of a data set of emails, each labeled as either "spam" or "not spam".

Work by pioneers such as Shannon and Turing ushered in the **cybernetic age** and, with it, the first wave of AI. **First-wave AI systems** – known today as **good old fashioned AI (GOFAI)** – were built on **deduction** and on handcrafted collections of facts. Within two decades, this logic-based approach was found to be inadequate vis-à-vis the complexity of perception in the real world, which can hardly be accommodated within GOFAI's world of well-defined distinct objects. By the late 1980s, GOFAI had fallen under the weight of its own exaggerated promises.

The **second, current, wave of AI** is dominated by machine learning, which is built on **induction**, that is, generalization from examples, and statistical pattern recognition. Linear methods, such as support vector machines, were the techniques of choice in the earlier years of the second wave, while more recent years, stretching to today, have witnessed the rise of **deep learning**. (Accordingly, some partition this second wave into two separate waves.[2])

The breakthroughs during the latter part of the second wave have met many of the goals set out in Shannon's paper from 1950, thanks mostly to the convergence of **"big data"** and **"big compute"**. For example, AlphaGo Zero – a more powerful version of the AI that famously beat a human world champion at the game of Go – required training more than 64 GPU workers and 19 CPU parameter servers for weeks, with an estimated hardware cost of US$25 million. OpenAI's highly publicized video game-playing program needed training for an equivalent of 45,000 years of game play, costing millions of dollars in rent access for cloud computing. The focus on scale has also led to concerns about the **accessibility** of the technology, with large multinationals scooping up most of the world's data in large data centers. The trend is expected to continue for the foreseeable future, thanks also to the rise of the **Internet of Things (IoT)**.

This book focuses on second-wave AI ideas and principles with the ambition of helping rebuild the bridge between machine learning and the fields of information theory and signal processing that existed at the time of Shannon's and Turing's work.

As a final note, it is useful to emphasize that the scope of machine learning should not be conflated with that of AI. The field of AI is far broader, including also first-wave AI methods based on logic, symbolic reasoning, and deduction, as well as other areas such as planning. The road to achieving **"general AI"** appears to be long and to stretch beyond the boundaries of machine learning: In the words of the American computer scientist Fei-Fei Li, with today's AI, i.e., with machine learning, "A machine … can make a perfect chess move while the room is on fire." An outlook on the topic of general AI is provided in Chapter 15.

1.4 What is Machine Learning?

To introduce the data-driven statistical pattern recognition approach at the core of machine learning, let us first review the standard **domain knowledge-based model-driven design methodology** that should be familiar to all engineers.

1.4.1 Domain Knowledge-Based Model-Driven Design

Engineers are often faced with the problem of designing algorithms to be deployed in a given real-world setting to meet certain performance requirements. To choose examples close to Shannon's work, a group of engineers may be tasked with devising an algorithm to compress images in such a way that they can be reconstructed with limited distortion; or to invent algorithms capable of communicating reliably over a wireless channel.

To address this type of problem, the standard domain knowledge-based model-driven design methodology complies with the block diagram shown in Fig. 1.1, carrying out the following steps:

[2] Besides deduction and induction, there is a third form of reasoning, or inference, that AI has yet to master: abduction. Abductive inferences involve intuition, guesswork, and the ability to form hypotheses to be revised via observation.

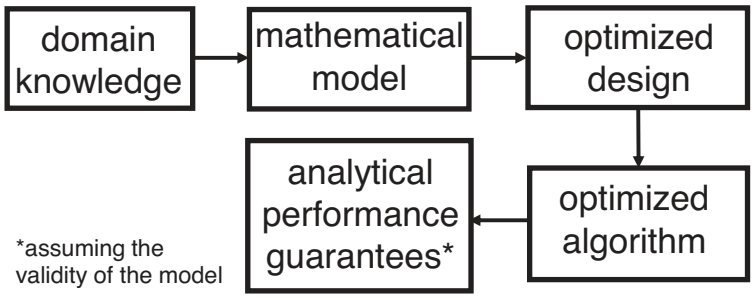

Figure 1.1 Conventional domain knowledge-based model-driven design methodology.

1. **Definition of a mathematical model based on domain knowledge**. Based on domain knowledge, i.e., information about the "physics" of the problem, one constructs a mathematical model of the problem of interest that involves all the relevant variables. Importantly, the mathematical model is **white-box**, in the sense that the relationship between inputs and outputs variables is mediated by quantities and mechanisms that have physical counterparts and interpretations.
2. **Model-based optimization**. Given the mathematical model, one formulates an optimization problem with the aim of identifying an optimal algorithm. Importantly, the optimized algorithm offers analytical performance guarantees under the assumption that the model is correct.

Example 1.1

To illustrate this methodology, consider the mentioned problem of designing a reliable wireless digital communication system. The outlined conventional methodology starts by establishing, or retrieving, domain knowledge about the physics of radio propagation from a transmitter to a receiver in the environment of interest. This step may entail consulting with experts, relying on textbooks or research papers, and/or carrying out measurement campaigns. As a result of this first step, one obtains a **mathematical model** relating input variables, e.g., input currents at the transmitter's antennas, and output variables, e.g., output currents at the receiver's antennas. The resulting white-box model is inherently **interpretable**. For instance, the relationship between the signals at the transmitter's and receiver's antennas may depend on the number and type of obstacles along the way.

The second step is to leverage this model to define, and hopefully solve, a **model-based optimization problem**. For the example at hand, one should optimize over two algorithms – the first, the encoder, mapping the information bits to the input currents of the transmit antennas, and the second, the decoder, mapping the output currents at the receiver to the decoded bits. As an optimization criterion, we can adopt the average number of incorrectly decoded bits, which we wish to minimize while ensuring a power or energy constraint at the transmitter. Driven by the model, the problem can be mathematically stated and ideally solved, yielding provably optimal algorithms, whose performance can be backed by analytical guarantees. Crucially, any such guarantee is only reliable and trustworthy to the extent that one can trust the starting mathematical model.

1.4.2 Inductive Bias-Based Data-Driven Design: Machine Learning

Machine learning follows an **inductive bias-based** – rather than domain knowledge-based – **data-driven** – rather than model-driven – design methodology that is illustrated by the block diagram in Fig. 1.2. This methodology proceeds as follows:

1. **Definition of an inductive bias**. As discussed, the conventional design methodology assumes the availability of a sufficiently detailed and precise knowledge about the problem domain, enabling the definition of a trustworthy, white-box, mathematical model of the "physics" of the variables of interest. In contrast, when adopting a machine learning methodology, one implicitly assumes that such knowledge is incomplete, or at least that the available models are too complex to be used in the subsequent step of model-based optimization. To account for this **model deficit** or **algorithmic deficit**, the machine learning methodology starts off by restricting the type of algorithms that may work well for the problem at hand. Broadly speaking, the selected class of algorithms encompasses **black-box** mappings between input and output variable; think of polynomial functions with a given degree. This class of mappings, which we will refer to as **model class**, constitutes (part of) the **inductive bias** that the methodology builds upon to extract information from data.[3]
2. **Training**. Machine learning aims to extract patterns and regularities from data that may generalize outside the available data set, i.e., to the "real world". This process, known as **training**, operates within the confines set by the inductive bias: Based on data, a specific (black-box) model is selected within the model class posited by the inductive bias, and no models outside it are allowed as the outputs of training. As opposed to the analytical performance guarantees provided by the conventional methodology, only statistical metrics can generally be evaluated on the trained model. Their validity and relevance rests on the extent to which the available data are deemed to be representative of the conditions one would encounter when deploying the system.

A popular cartoon[4] summarizes – and somewhat simplifies! – this process in the form of Algorithm 1.1.

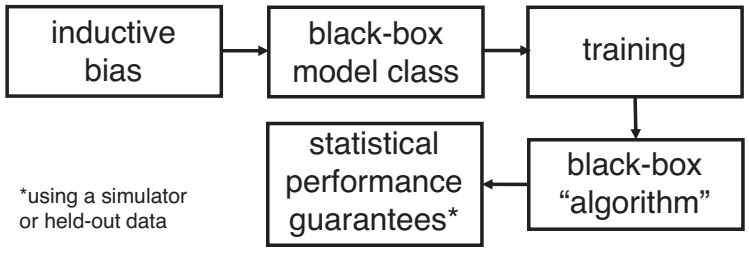

Figure 1.2 Inductive bias-based data-driven design methodology adopted by machine learning.

[3] The definition of an inductive bias may be the result of abductive inference.
[4] https://xkcd.com/1838/

Algorithm 1.1: Machine learning as an algorithm (according to an xkcd cartoon)

while *numbers don't look right* **do**
  pour data into pile of linear algebra
  collect answers
end

Example 1.2

To elaborate on the methodology followed by machine learning, consider again the problem of designing the transmitter and receiver of the wireless communication link described in Example 1.1. Assume that a model for the propagation environment is not available; perhaps we are faced with a new communication setting, and we do not have the resources to carry out an extensive measurement campaign or to consult with experts to obtain an accurate "physics"-based model. What we assume to be true, instead, is that a general class of **black-box models**, such as linear functions or neural networks, can implement effective transmitting and receiving algorithms. We may know this because of prior experience with similar problems, or through some trial-and-error process. This choice defines the **inductive bias**.

We also assume that we can collect data by transmitting over the channel. **Training** leverages these data to select a specific model within the given model class posited as part of the inductive bias. In the problem at hand, training may output two neural networks, one to be used by the transmitter and one by the receiver. The performance obtained upon deployment can be estimated using (additional) available data, as long as the latter are deemed to be sufficiently rich to accurately reflect real-world conditions.

Concluding this section, a couple of remarks are in order.

- **Inductive bias and domain knowledge**. As emphasized in the discussion above, the selection of the inductive bias should be guided as much as possible by domain knowledge. While not enough domain knowledge is assumed to be available to determine an accurate mathematical model, information about the problem is invaluable in selecting a model class that may effectively generalize outside the available data. Domain knowledge may be as basic as the identification of specific **invariances** in the desired input–output mapping. For instance, in the problem of classifying images of cats against images of dogs, it does not matter *where* the cat or dog is in the image. This translational invariance can be leveraged to select a model class whose constituent classifiers produce the same output irrespective of a shift in the input. A recent example of the importance of choosing a suitable inductive bias is provided by the success of DeepMind's AlphaFold system in predicting the structure of a protein from its primary sequence. A key element of AlphaFold is, in fact, the way in which its predictive model class encodes symmetry principles that facilitate reasoning over protein structures in three dimensions.
- **White-box vs. black-box models**. Care was taken in the discussion above to distinguish between white-box models – the type of mathematical models assumed by the conventional engineering methodology – and black-box models and model classes – the type of input–output mappings assumed as part of the inductive bias by machine learning. While we use

in both cases we use the term "models", their application and significance are different, and are key to understanding the distinction between the two methodologies.

1.5 Taxonomy of Machine Learning Methods

There are three main broad classes of machine learning algorithms, which differ in the way data are structured and presented to the learning agent: supervised, unsupervised, and reinforcement learning. This section provides a short introduction.

1.5.1 Supervised Learning

In supervised learning, the learning agent is given a **training data set** including multiple samples. Each data sample consists of a pair **"(input, desired output)"**: Each data point hence "supervises" the learner as to the best output to assign to a given input. This is illustrated by the example in Fig. 1.3, where the input variable is given by a point in the two-dimensional plane, while the target variable is binary, with the two possible values represented as a circle and a cross.

In supervised learning, the goal is to generalize the input–output relationship – which is partially exemplified by the training data – outside the training set. In the example in Fig. 1.3, the problem amounts to **classifying** all points in the plane as being part of the "circle class" or of the "cross class". The trained model may be expressed as a line separating the regions of inputs belonging to either class.

A practical example is the problem of classifying emails as spam or not spam, for which we may have access to a corpus of emails, each labeled as "spam" or "not spam".

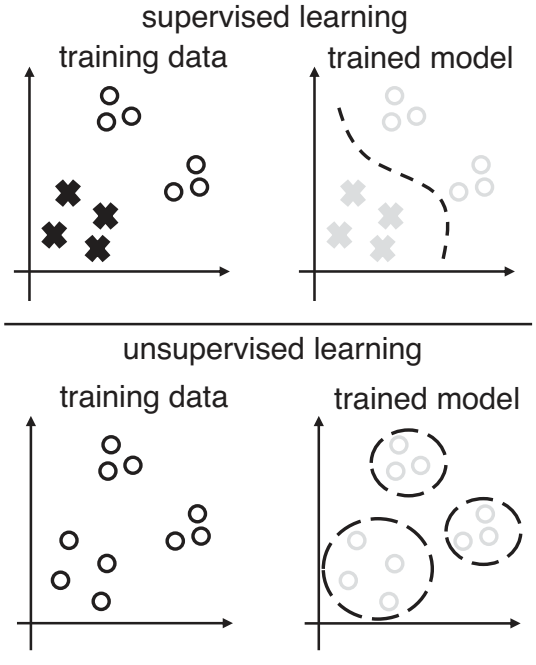

Figure 1.3 Supervised vs. unsupervised learning.

1.5.2 Unsupervised Learning

In unsupervised learning, the learning agent is again given a training set containing multiple samples. However, each sample only contains the "input", without specifying what the desired output should be. For instance, in Fig. 1.3, the learner is given several points on the plane without any label indicating the target variable that should be assigned to each point. Unsupervised learning generally leverages **similarities** and **dissimilarities** among training points in order to draw conclusions about the data.

A specific example of an unsupervised task – but by no means not the only one! – is **clustering**, whereby the data set is partitioned into groups of similar inputs. For the corpus of emails mentioned above, in the absence of the "spam"/"not spam" labels, clustering may aim to group together emails that are about similar topics.

1.5.3 Reinforcement Learning

In reinforcement learning, the learning agent is not given a fixed data set; instead, data are collected as the agent interacts with the environment over time. Specifically, based on its current "understanding" of the environment, at each given time step the agent takes an action; observes its impact on the evolution of the system; and adjusts its strategy for the following step. To enable adaptation, the agent is given a reward signal that can be used to reinforce useful behavior and discard actions that are harmful to the agent's performance.

With its focus on feedback-based dynamic adaptation, reinforcement learning is closely related to control theory. Unlike standard control theory, however, in reinforcement learning one does not assume the availability of a (white-box) model for the evolution of the system. The agent can only rely on data and on the rewards to optimize its operation. Reinforcement learning is quite different in nature from supervised and unsupervised learning, and it will not be further discussed in this book.

1.6 When to Use Machine Learning?

Based on the discussion so far in this chapter, we can draw some general conclusions about the relative merits of model-driven and data-driven design methodologies, as well as some guidelines on when to choose the latter.

1.6.1 Pros and Cons

Let us start by listing potential advantages of the machine learning methodology:

- **Lower cost and faster development**. Machine learning can reduce the time to deployment if collecting data is less expensive, or faster, than acquiring a "physics"-based mathematical model along with an optimal model-based algorithm.
- **Reduced implementation complexity**. Trained models can be efficiently implemented if the model class selected as part of the inductive bias contains low-complexity "algorithms", such as linear models or shallow neural networks.

But there are also potential drawbacks:

- **Limited performance guarantees**. While model-driven designs can be certified with analytical performance guarantees that are valid under the assumed model, the same is not true for data-driven solutions, for which only statistical guarantees can be provided.
- **Limited interpretability**. Machine learning models are black boxes that implement generic mappings between input and output variables. As such, there is no direct way to interpret their operation by means of mechanisms such as **counterfactual queries** ("what would have happened if … ?"). This issue can be partially mitigated if the selected model class contains structured models that encode a specific inductive bias. For instance, if a model class relies on given hand-crafted features of the data, one may try to tease apart the contribution of different features to the output.

1.6.2 Checklist

Given these pros and cons, when should we use machine learning? The following check list provides a useful roadmap.

1. **Model or algorithm deficit?** First off, one should rule out the use of domain knowledge-based model-driven methods for the problem at hand. There are two main cases in which this is well justified:
   - **model deficit**: a "physics"-based mathematical model for the variables of interest is not available.
   - **algorithm deficit**: a mathematical model is available, but optimal or near-optimal algorithms based on it are not known or too complex to implement.
2. **Availability of data?** Second, one should check that enough data are available, or can be collected.
3. **Stationarity?** Third, the time over which the collected data are expected to provide an accurate description of the phenomenon of interest should be sufficiently long to accommodate data collection, training, and deployment.
4. **Performance guarantees?** Fourth, one should be clear about the type of performance guarantees that are considered to be acceptable for the problem at hand.
5. **Interpretability?** Finally, one should assess to what degree "black-box" decisions are valid answers for the setting under study.

The checklist is summarized in Algorithm 1.2.

1.7 Summary

- Machine learning – often conflated with AI – is extensively used in the real world by individuals, companies, and governments, and its application requires both technical knowledge and ethical guidelines.
- The field of AI has gone through various waves of funding and hype, and its basic tenets are decades old, dating back to work done during the cybernetic age. The current, second, wave revolves around statistical pattern recognition – also known as machine learning.
- Machine learning is distinct from the domain knowledge-based model-driven design methodology that is most familiar to engineers. Rather than relying on optimization over

Algorithm 1.2: Should I use machine learning?

```
if there is a model or algorithm deficit then
    if data are available or can be generated then
        if environment is stationary then
            if model-based performance guarantees are not needed then
                if interpretability is not a requirement then
                    apply machine learning
                end
            end
        end
    end
end
```

a "physics"-based analytical model, machine learning adopts an inductive bias-based data-driven methodology.

- The inductive bias consists of a model class and a training algorithm. Based on training data, the training algorithm selects one model from the model class. The inductive bias should ideally be selected based on domain knowledge.
- Machine learning methods can be broadly partitioned into supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, the training algorithm is given examples of an input–output relationship; in unsupervised learning, only "inputs" are available; while in reinforcement learning the agent collects data as it interacts with the environment.
- Machine learning can help lower cost, deployment time, and implementation complexity, at the cost of offering more limited performance guarantees and interpretability as compared to the standard model-based methodology.

1.8 Recommended Resources

For further reading on the cybernetic age and on the early days of AI, I recommend the books by Kline [1] and by Wooldridge [2] (the latter includes the story of ELIZA), as well as Lem's "Cyberiad" [3]. The classical books by Pagels [4] and by Hofstadter [5] are also still well worth reading. For discussions on the role of AI and machine learning in the "real world", interesting references are the books by Levesque [6], Russell [7], and Christian [8]. The philosophical relationship between GOFAI and machine learning is covered by Cantwell Smith [9]. If you wish to learn more about the "extractive" aspects of AI – from natural resources, communities, individuals – Crawford's book [10] provides an informative introduction.

Dozens of papers on machine learning are posted every day on repositories such as arXiv.org. Incentives to publish are such that one should generally question results that appear exaggerated. Are the experiments reproducible? Are the data representative of real-world scenarios or instead biased in favor of the use of a specific method? At the time of writing, a particularly evident failure of machine learning is the lack of effective AI tools to diagnose COVID-19 cases [11], despite hundreds of (published) claims of success stories.