1 Introduction

La gymnastique cerebrale n'est pas susceptible d'ameliorer l'organisation du cerveau en augmentant le nombre de cellules, car, on le sait, les elements nerveux ont perdu depuis l'epoque embryonnaire la propriety de proliferer; mais on peut admettre comme une chose tres vraisemblable que l'exercice mental suscite dans les regions cerebrales plus sollicitees un plus grand developpment de l'appareil protoplasmique et du systeme des collaterales nerveuses. De la sorte, des associations deja creees entre certains groupes de cellules se renforceraient notablement au moyen de la multiplication des ramilles terminales des appendices protoplasmiques et des collaterals nerveuses; mais, en outre, des connexions intercellulaires tout a fait nouvelles pourraient s'etablir grace a la neoformation de collaterales et d'expansions protoplasmiques. [Santiago Ramón y Cajal [165]]

The long-term research goal behind this book is to understand intelligence in brains and machines. Intelligence, like consciousness, is one of those words that:

(1) was coined a long time ago, when our scientific knowledge of the world was still fairly primitive;
(2) is not well defined, but has been and remains very useful both in everyday communication and scientific research; and
(3) for which seeking a precise definition today is premature, and thus not particularly productive.

Thus, rather than trying to define intelligence, we may try to gain a broader perspective on intelligent systems, by asking which systems are "intelligent", and how they came about on planet Earth. For this purpose, imagine an alien from an advanced civilization on a distant galaxy charged with reporting to her alien colleagues on the state of intelligent systems on planet Earth. How would she summarize her main findings?

1.1 Carbon-Based and Silicon-Based Computing

At a fundamental level, intelligent systems must be able to both compute and store information, and thus it is likely that the alien would organize her summary along these two axes. Along the computing axis, the first main finding she would have to report is that currently there are two computing technologies that are dominant on Earth: carbon-based computing implemented in all living systems, and silicon-based computing

https://doi.org/10.1017/9781108955652.002 Published online by Cambridge University Press

implemented in a growing number of devices ranging from sensors, to cellphones, to laptops, to computer clusters and clouds. Carbon-based computing has a 3.8 billion-year-long history, driven by evolution. In contrast, silicon-based computing is less than 100 years old, with a history driven by human (hence carbon-based) design rather than evolution. Other computing technologies, from DNA computing to quantum computing, currently play minor roles, although quantum computing can be expected to significantly expand in the coming two decades.

Along the storage axis, the main finding the alien would have to report is that there are at least two different styles of storage: the digital/Turing-tape style, and the neural style which is at the center of this book (Figure 1.1). In the digital style, information is stored neatly at different discrete locations, or memory addresses, of a physical substrate. In the neural style of computing, information is stored in a messy way, through some kind of holographic process, which distributes information across a large number of synapses. Think of how you may store your telephone number in a computer as opposed to your brain. In Turing machines, storage and processing are physically separate and information must be transferred from the storage unit to the computing unit for processing. In neural machines, storage and processing are intimately intertwined. In the digital style, storage tends to be transparent and lossless. In the neural style, storage tends to be opaque and lossy.

Remarkably, carbon-based computing discovered both ways of storing information. It first discovered the digital style of storage, using chemical processes, by storing information using DNA and RNA molecules which, to a first degree of approximation, can be viewed as finite tapes containing symbols from a four-letter alphabet at each position. Indeed, biological systems store genetic information, primarily about genes and their control, at precise addresses along their DNA/RNA genome. And every cell can be viewed as a formidable computer which, among other things, continuously measures and adjusts the concentration of thousands of different molecules. It took roughly 3.3 billion years of evolution of carbon-based digital computing for it to begin to discover the neural style of information processing, by developing the first primitive nervous circuits and brains, using tiny electrical signals to communicate information between neurons. Thus, about 500 million years ago it also began to discover the neural style of information storage, distributing information across synapses. In time, this evolutionary process led to the human brain in the last million year or so, and to language in the last few hundred thousand years.

It is only over the very last 100 years, using precisely these tiny electrical signals and synapses, that the human brain invented silicon-based computing which, perhaps not too surprisingly, also uses tiny electrical signals to process information. In some sense, the evolution of storage in silicon-based computing is an accelerated recapitulation of the evolution of storage in carbon-based computing. Silicon-based computing rapidly adopted the digital Turing style of storage and computing we are so familiar with. As an aside, it is, ironically, striking that the notion of tape storage was introduced by Turing precisely while thinking about modeling the brain which uses a different style of storage. Finally, in the last seven decades or so, human brains started trying to simulate on digital computers, or implement in neuromorphic chips, the neural style of

https://doi.org/10.1017/9781108955652.002 Published online by Cambridge University Press

computing using silicon-based hardware, beginning the process of building intelligent machines (Figure 1.1). While true neuromorphic computing in silicon substrate is an active area of research, it must be stressed that the overwhelming majority of neural network implementations today are produced by a process of virtualization, simulating the neural style of computing and storage on digital, silicon-based, machines. Thus, for most of these neural networks, there are no neurons or synapses, but only fantasies of these objects stored in well-organized digital memory arrays. Silicon computing is fast enough that we often forget that we are running a neural fantasy. As we shall see later in this book, thinking about this virtualization and about computing in native neural systems, rather than their digital simulations, will be key to better understand neural information processing.

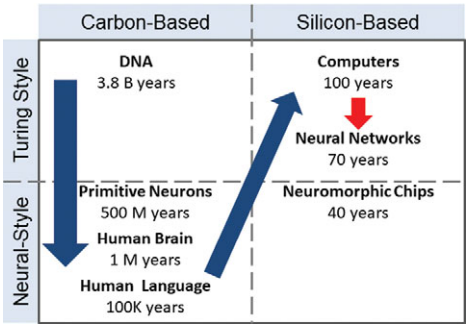

Figure 1.1 Evolution of computing and intelligence on planet Earth with approximate time scales. Computing on Earth can be organized along two axis: processing (carbon-based vs. silicon-based) and storage style (Turing vs. neural). Evolution began with carbon-based processing and Turing-style storage approximately 3.8B years ago. Primitive neurons and brains began emerging 500M years ago. Primate brains are a few million years old and human language is a few hundred thousand years old. Over the last 100 years or so, human brains developed silicon-based computing and computers rooted in the idea of Turing machines. AI and ANNs (artificial neural networks) have been developed over the last 70 years or so (red arrow). Most neural networks used today are virtual, in the sense that they are implemented in digital machines using the Turing style of storage. Neuromorphic chips, with mostly Turing-style but occasionally also neural-style of storage, have been in development for the past 40 years. Likewise, over the past 40 years, digital computers and artificial neural networks have been applied to biology, from molecular biology and evolution to neuroscience, to better understand carbon-based computing (arrow not shown).

Today, the carbon-based and silicon-based computing technologies are vastly different and carbon-based computing is still in many ways far more sophisticated. The differences are at all levels: physical sizes, time scales, energy requirements, and overall architectures. For instance, the human brain occupies slightly less than two liters of space and uses on the order of 20–40 W of power, roughly the equivalent of a light bulb, to effortlessly pass the Turing test of human conversation. In comparison, some of our supercomputers with their basketball-court size use three to four orders of magnitude more energy – something on the order of 100,000 W – to match, or slightly outperform,

https://doi.org/10.1017/9781108955652.002 Published online by Cambridge University Press

humans on a single task like the game of Jeopardy or GO, while miserably failing at passing the Turing test. This huge difference in energy consumption has a lot to do with the separation of storage and computing in silicon computers, versus their intimate and inextricable intertwining in the brain.

In spite of these differences, in the quest for intelligence these two computing technologies have converged on two key ideas, not unlike the well-known analogy in the quest of flight, where birds and airplanes have converged on the idea of using wings. In addition to using tiny electrical signals, both carbon-based and silicon-based intelligent systems have converged on the use of learning, including evolutionary learning and lifetime learning, in order to build systems that can deliver intelligent behavior and adapt to variations and changes in their environments. Thus it should not be too surprising that machine learning is today one of the key and most successful areas of artificial intelligence, and has been so for at least four decades.

As we have seen, on the silicon-side humans are learning how to emulate the neural-style of storing information. As an aside, and as an exercise in inversion, one may wonder whether evolution discovered how to emulate the Turing style of storage, for a second time, in brains. There is some evidence of that in our symbolic processing in general, in the discovery of individuals with superior autobiographical memory, or hyperthymesia [441, 644], who tend to index their life by dates, and in "enfants savants" and other individuals with superior arithmetic and other related capabilities, often connected to autism spectrum disorders (e.g. [383, 268, 370]). Unfortunately, we still know too little about information storage in the brain to really address this question, which touches on some of the main challenges for AI today.

1.2 Early Beginnings Until the Late 1940s

We now turn to a brief history of neural networks and deep learning. The goal here is not to be comprehensive, but simply to connect some of the most salient historical points in order to gain a useful perspective on the field. Additional pointers can be found, for instance, in [653]. Although one can trace the beginnings of artificial intelligence back to the Greek philosophers and even more ancient times, a more precise beginning that is relevant for this book can be identified by considering shallow learning as the precursor of deep learning. And shallow learning began with the discovery of linear regression in the late 1700s.

1.2.1 Linear Regression

The discovery of linear regression in the late 1700s resulted from the work of Carl Friedrich Gauss (1777–1855) and Adrien-Marie Legendre (1752–1833). Many if not most of the features of machine learning and deep learning are already present in the basic linear regression framework (Figure 1.2), such as having:

(1) an initial set of data points;

https://doi.org/10.1017/9781108955652.002 Published online by Cambridge University Press

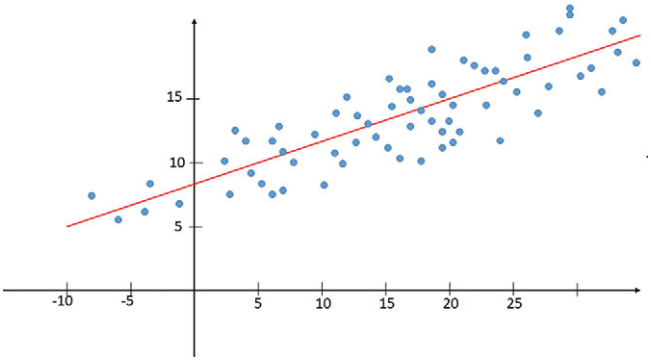

Figure 1.2 Linear regression in two dimensions.

(2) a class of possible models;

(3) a problem of model fitting;

(4) a problem of prediction using fitted models;

(5) a problem of model comparison; and so forth.

All these features are present in the deep learning framework of today, and by and large a cynic could say that most of deep learning today is akin to linear regression on steroids. However there are two fundamental points where linear regression is misleading. First, there is a closed mathematical formula for the coefficients of the "best" model, which will be reviewed in a coming chapter. In deep learning the models are more complex and there is no analytical formula; the parameters of the model must be learnt progressively through a process of optimization. Second, and especially in two or three dimensions, the linear model is easily interpretable and our visual system can see the data points and the model. In deep learning one is typically fitting non-linear surfaces in high-dimensional spaces, a process that cannot be visualized directly, and the parameters of the models tend to be more opaque. However, even in the simple case of a linear model, or a linear neuron, opacity is already present due to the neural style of storage: information about the data is stored, through a shattering process, in the coefficients of the linear model. This storage process is irreversible: one cannot retrieve the original points from the coefficients. Only some of their basic statistical properties, such as means and covariances (the sufficient statistics), are retained.

Of course, in addition to using non-linear models, deep learning today is often applied in situations characterized by very large numbers of data points in high-dimensional spaces (e.g. 1 billion humans in genome/DNA space or in photo/pixel space), where traditional linear regression has rarely ventured, for obvious historical and technical reasons.

https://doi.org/10.1017/9781108955652.002 Published online by Cambridge University Press

1.2.2 Neuroscience Origins

Next, having traced the beginning of shallow (linear) learning, we can now turn to the true beginnings of deep learning. Machine learning and artificial intelligence as we know them today began to be developed in the late 1940s and early 1950s. They were fundamentally inspired by questions and knowledge available at the time about the brain. Thus it is useful to briefly summarize the state of neuroscience around 1950. Although we know much more today, many of the basic principles were already in place by 1950.

Briefly, by 1950 scientists had already gathered a good deal of essential information about the brain, its structure, and its function. Charles Darwin (1809–1882)'s *On the Origin of Species* had been published almost a century earlier in 1859. Thus they were well aware of the fact that the human brain has been shaped by evolutionary forces and that "nothing in biology makes sense except in the light of evolution"[1]. And in as much evolution operates by tinkering, rather than design, they could expect the brain to have a messy structure. Information about the coarse anatomy of the brain and some of its different components was also known. For instance, since the work of Pierre Paul Broca (1824–1880), Carl Wernicke (1848–1905), and others in the 19th century, the existence of very specialized areas for speech production and comprehension, as well as other functions was known, although not with the level of detail we have today. The detailed anatomical work of Santiago Ramón y Cajal (1852–1934), using staining methods that had been pioneered by Camillo Golgi (1843–1926), had revealed the delicate architecture of the brain, the various cortical layers, and the remarkable arborization and shape of different kinds of neurons, from pyramidal cells in the cortex to Purkinje cells in the cerebellum. The word "synapse", describing the contacts neurons make with each other, had been introduced by Charles Sherrington (1857–1952). Anesthetic and other drugs that can chemically alter brain states and consciousness were also well known. Finally the studies of Alan Hodgkin (1914–1998) and Andrew Huxley (1917–2012) of how neurons transmit electric signals along their axons started before, and interrupted by, the Second World War, had resumed and resulted in the publication of the famous Hodgkin–Huxley model in 1952.

At the same time, on the theoretical side, mathematicians from George Boole (1815–1864), to Georg Cantor (1845–1918), to Kurt Gödel (1906–1978), and especially Alan Turing (1912–1954) had laid the foundations of logic, computability, and computer science. The mathematician John von Neumann (1903–1957) had designed the basic computer architecture still in use today, participated in the development of one of the first computers, studied cellular and self-reproducing automata, and written a little book *The Computer and the Brain* (originally unfinished and published first in 1958 after his death) [770]. Claude Shannon (1916–2001) had laid the foundations of information theory in *A Mathematical Theory of Communication* published in 1948. Perhaps most notably for this book, Warren McCulloch (1898–1969) and Walter Pitts (1923–1969) had published *A Logical Calculus of the Ideas Immanent in Nervous Activity* in 1943 and,

[1] This is the title of an essay published a little later, in 1973, by evolutionary biologist Theodosius Dobzhansky (1900–1975).

https://doi.org/10.1017/9781108955652.002 Published online by Cambridge University Press

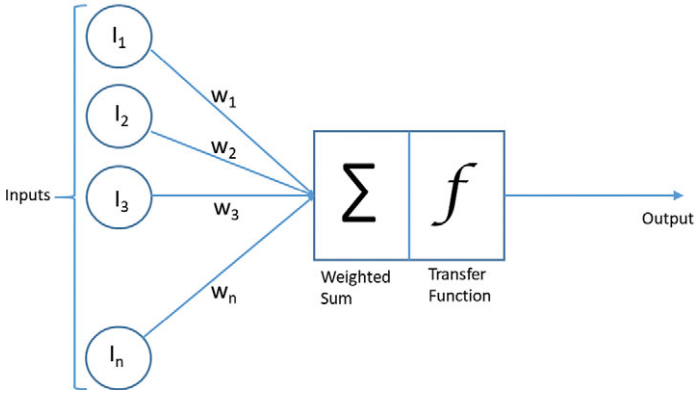

Figure 1.3 Simple neuronal model with n inputs and n synaptic weights. The neuron operates by computing the weighted sum of its inputs, also called the activation, and then passing it through a transfer function, also called activation function, to produce its output. Crucially, this computing units comes with its own storage unit. Information is stored in the synaptic weights.

together with their subsequent work, started to introduce a simple neural model, which is essentially what is still used today in most deep learning applications (Figure 1.3).

It is against this rich intellectual backdrop, that scientists could begin to think in more precise ways about learning and memory and try to seek detailed mechanistic explanations for Ivan Pavlov (1849–1936)'s experiments on conditioning reflexes or, more broadly, the question of neural storage: how and where one stores information in the brain, such as one's name or telephone number.

A back of the envelope calculation shows that long-term memories cannot be encoded in permanent electrical signals reverberating throughout the brain. As a result, long-term memory bits must ultimately be encoded in the structure and biochemical composition of the brain, not its electrical activity, and synapses are the ideal place where this encoding ought to take place, in order to be able to rapidly influence neural activity during recall and other tasks. The hypothesis that memories may be encoded in synapses can be traced back at least to an article by Cajal [165], where it is hinted in broad strokes, the term synapse not having been created yet. This is the quote given at the beginning of this chapter, which can roughly be translated by: "Mental exercise is not likely to improve the organization of the brain by increasing the number of cells because, as we know, nerve cells have lost the ability to replicate since the embryonic stage; but one can accept as being very likely that mental exercise triggers the growth of the protoplasmic apparatus and branching network of arborizations in the areas that are most activated. In this way, associations already created between certain groups of cells become reinforced in particular through the expansion of terminal arborizations and connections; in addition, entirely new connections between cells could be created through the formation of new branches and new protoplasmic expansions.".

Thus the simple idea was born that learning and memory must be implemented somehow at the level of synapses (their position, shape, composition, creation/elimination,

https://doi.org/10.1017/9781108955652.002 Published online by Cambridge University Press

Table 1.1 Length scales: in the human brain, and the brain rescaled by a factor of 10^6.

Object	Scale in Meters	Rescaled by 10^6	Rescaled Object
Diameter of Atom	10^{-10}	10^{-4}	Hair
Diameter of DNA	10^{-9}	10^{-3}	
Diameter of Synapse	10^{-7}	10^{-1}	Fist
Diameter of Axon	10^{-6}	10^{0}	
Diameter of Neuron	10^{-5}	10^{1}	Room
Length of Axon	10^{-3}–10^{0}	10^{3}–10^{6}	Park-Nation
Length of Brain	10^{-1}	10^{5}	State
Length of Body	10^{0}	10^{6}	Nation

strengthening/weakening) – and roughly remains the guiding model even today. Needless to say, we have gained a much better appreciation of the biochemical processes involved in synaptic modifications, including complex patterns of gene expression and epigenetic modifications, and the complex production, transport, sequestration, and degradation of protein, RNA, and other molecular species (e.g. [317, 488, 767, 568, 713]).

Furthermore, one may conjecture that the essence of the learning algorithms for modifying synapses must be relatively simple since they are shared by vertebrates and invertebrates (e.g. Aplysia) and thus were discovered very early by evolution. This is not to say, of course, that the actual implementation of the algorithms in biological wetware may not be extremely complicated, requiring again changes in gene expression, epigenetic modifications, and many other cellular processes. But the basic underlying algorithms ought to be relatively simple, and thus began the quests for such algorithms.

1.2.3 The Deep Learning Problem

While elegant in its simplicity and power, the idea that changes in synapses over time and in different parts of the brain, intimately coupled with electrical activity, is ultimately responsible for learning and memory formation faces a formidable challenge which is not immediately apparent given the very small length-scale of synapses. To clearly see this challenge, it is useful to rescale synapses by a factor of one million (Table 1.1) to obtain a better visualization of the problem. With this rescaling, a synapse becomes the size of a human fist (10 centimeters), the body of a neuron the size of a house (~30 meters), the brain a sphere with radius ~100 kilometers, and the longest axons can run about 10^6 meters, or 1,000 kilometers. Imagine now the task of an individual who is trying to learn how to play tennis, or how to play the violin. How can a fist-sized motor synapse in Los Angeles adjust itself, deciding for instance whether to strengthen or weaken itself, to better control a tennis racket or a bow that is located in San Francisco? The synapse in Los Angeles knows nothing about tennis, or music, or the laws of mechanics, and it is essentially a blind machine: it can only sense its immediate biochemical environment on the scale of its diameter. *This is the epicenter of the deep learning problem.*

More broadly, how can very large numbers of essentially blind synapses, deeply buried

https://doi.org/10.1017/9781108955652.002 Published online by Cambridge University Press

inside a jungle of interconnected neurons, adjust themselves in a coordinated manner to produce memories and human intelligent behavior?

1.2.4 Hebbian Learning

Donald Hebb (1904–1985) is credited with being one of the first to sense this deep mystery, still largely unsolved today, and to attempt to provide one of the first ideas for a possible solution in his book *The Organization of Behavior*, which appeared in 1949. Hebb, who was primarily a psychologist, did not use any mathematical formalism and stated his ideas in rather vague terms. Buried in his book, one can find the statement: "*Let us assume that the persistence or repetition of a reverberatory activity (or 'trace') tends to induce lasting cellular changes that add to its stability. When an axon of cell A is near enough to excite cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased*". This is often paraphrased in compact form by: "Neurons that fire together, wire together".

While not mathematically precise, Hebb's conceptualization is the first to propose a general purpose, seemingly unsupervised, algorithm for the adjustment of synapses on a large-scale. As previously mentioned, the detailed molecular implementation of learning and memory algorithms in the brain is very complex, regardless of the potential simplicity of the underlying algorithms (as additional references to a vast literature, see for instance [324, 323, 789, 758]). However Hebb's basic insight has remained an important source of inspiration for machine learning, deep learning, and neuroscience. However, it is not clear at all how the large-scale application of Hebbian learning could lead to coherent memories and intelligent behavior. This problem will be examined in a later chapter of this book.

There are several ways of providing a precise mathematical implementation of Hebb's vague idea of self-organized learning. Without going into details yet, the most simple embodiment is given by learning rules of the form:

$$\Delta w_{ij} \propto O_i O_j \quad \text{or} \quad \Delta w_{ij} \propto (O_i - \mu_i)(O_j - \mu_j) \tag{1.1}$$

where w_{ij} denotes the strength of the synaptic connection from neuron j to neuron i, $\propto$ denotes proportionality, O_i represents the output of the postsynaptic neuron with corresponding average μ_i, O_j represents the output of the presynaptic neuron with corresponding average μ_j. The term Δw_{ij} represents the incremental change resulting from the activity of the pre- and post-synatpic neurons. Obviously a more complete formulation would have to include other important details, which will be discussed later, such as the time scales over which these quantities are computed and the interpretation of the outputs (e.g. in terms of firing rates). For the time being, it is worth noting two things in this formulation. First, the learning rule is a quadratic polynomial in the outputs of the two neighboring neurons. Second, as described, this formulation of Hebb's rule is symmetric in that it does not depend on whether the connection runs from the axon of neuron i to the dendrites of neuron j, or vice versa.

https://doi.org/10.1017/9781108955652.002 Published online by Cambridge University Press

1.3 From 1950 to 1980

1.3.1 Shallow Non-Linear Learning

A few years later, Frank Rosenblatt (1928–1971) proposed the first learning algorithm for a neuron represented by a linear threshold function (or perceptron) [618], the perceptron learning algorithm. If a single layer of n perceptrons is given, it is easy to see that each perceptron learns independently from the other perceptrons. In Chapter 7, we will prove that, for such a layer, the perceptron learning algorithm can be rewritten in the form:

$$\Delta w_{ij} \propto (T_i - O_i) I_j \tag{1.2}$$

where Δw_{ij} is the change in the synaptic weight w_{ij} connecting input j to output i, T_i is the binary target for output i, $O_i = f(S_i) = \sum_j f(w_{ij} I_j)$ where f is a threshold function, I_j is the jth component of the input, and all these quantities are computed on-line, i.e. for each training example. The perceptron algorithm can be viewed as a supervised Hebbian algorithm in that it requires a target T_i for each output unit i and each training example, but is written as a product of a postsynaptic term $T_i - O_i$ and a presynaptic term I_j. Again note that this learning rule is a quadratic function of the input, output, and target variables. As we shall see, when the data is linearly separable with respect to each output, the perceptron learning algorithm is capable of finding n suitable hyperplanes that correcly separate the data.

A slightly more general version of the perceptron algorithm, the Delta rule, was introduced by Bernard Widrow and Marcian Hoff [786] in the form:

$$\Delta w_{ij} \propto (T_i - O_i) g'(S_i) I_j \tag{1.3}$$

where the new term $g'(S_i)$ represents the derivative of the transfer function g, and $O_i = g(S_i)$. This again can be viewed as a supervised Hebbian rule for shallow (one-layer) differentiable networks, and it is easy to see that the Widrow–Hoff rule performs gradient descent with respect to the least square error function $\mathcal{E} = \frac{1}{2} \sum_i (T_i - O_i)^2$.

1.3.2 First Forays into Deep Architectures and their Challenges

In the following two decades, some progress was made through a number of individual efforts, but with little global coordination. On the neuroscience side, David Hubel (1926–2013) and Torsten Wiesel began probing the mysteries of the visual system. As the man who introduced the term synapse had put it: "*A shower of little electrical leaks conjures up for me, when I look, the landscape; the castle on the height, or when I look at him, my friend's face and how distant he is from me they tell me. Taking their word for it, I go forward and my other senses confirm that he is there.*" (Charles Sherrington in *Man on his Nature*, 1940). Huber and Wiesel published the results of some of their famous experiments [365] showing that, under anesthesia and very specific stimulus conditions, there are neurons in the early stages of the cat visual cortex that behave like feature detectors by responding, for instance, to bars of a particular orientation at a particular

https://doi.org/10.1017/9781108955652.002 Published online by Cambridge University Press

location in the visual field. Furthermore, these feature detector neurons are organized into fairly regular arrays covering both the entire visual field and the space of features (e.g. bars at all possible orientations).

On the computational side, Alexey Ivakhnenko (1913–2007) in Russia seems to have been among the first to formally consider multi-layer architectures of simple processing units, and deep learning algorithms for training them [371, 372], such as gradient descent, although the influence of his work at the time seems to have been limited. The idea of gradient descent goes back at least to the work of Augustin-Louis Cauchy (1789–1857) in 1847 and Jacques Hadamard (1865–1963) in 1908. As we shall see, deriving gradient descent in systems with multiple layers requires the chain rule of calculus, which itself goes back to Gottfried Leibniz (1646–1716) and Isaac Newton (1643–1727). It is also in Russia, around the same period that Vladimir Vapnik and Alexey Chervonenkis (1938–2014) began to develop their statistical theory of learning ([752, 750, 751] and references therein).

More influential at the time were Marvin Minsky (1927–2016) and Seymour Papert with their 1969 book [503] proving a number of interesting results about perceptrons. In this book, they also raised concerns regarding the possibility of finding good learning algorithms for deep (multi-layer) perceptrons: "*The perceptron has shown itself worthy of study despite (and even because of!) its severe limitations. It has many features to attract attention: its linearity; its intriguing learning theorem; its clear paradigmatic simplicity as a kind of parallel computation. There is no reason to suppose that any of these virtues carry over to the many-layered version*." While such statements may have not encouraged research in neural networks, claims that this brought neural network research to a halt have been exaggerated, as can be seen from the literature. Even in the original perceptron book, right after the sentence above, the authors wrote the following hedging sentence: "*Nevertheless, we consider it to be an important research problem to elucidate (or reject) our intuitive judgment that the extension to multilayer systems is sterile. Perhaps some powerful convergence theorem will be discovered, or some profound reason for the failure to produce an interesting 'learning theorem' for the multilayered machine will be found*".

Finally, in Japan, throughout the 1960s and 1970s Kehiko Fukushima began to think about computer vision architectures inspired by neurobiology and in particular the work of Hubel and Wiesel [286, 287, 289]. This line of work culminated in the neocognitron model [288], essentially a multi-layer convolutional neural network, whereby local operations, such as detection of particular features over a small patch, are repeated at each location of the input image, effectively convolving the image with the corresponding feature detector, and providing a foundation for enabling translation invariant recognition. Unfortunately the computer power to train convolutional neural networks was not readily available at the time. However, and even more fundamentally, Fukushima proposed to solve vision problems using the wrong learning algorithm: he proposed to learn the parameters of the neocognitron in a self-organized way using Hebbian learning. This approach never materialized in practice and the reasons behind this will play an important role later in the book. In particular, we will show that Hebbian learning applied to a feedforward convolutional neural network cannot solve vision tasks [102].

https://doi.org/10.1017/9781108955652.002 Published online by Cambridge University Press

Needless to say, there were many other developments in the period 1950–1980 that were directly relevant but cannot be surveyed, for instance, the development of complexity theory and the concept of NP-completeness [293], or additional work on neural networks by other scientists, such as Shun Ichi Amari [22, 23] or Stephen Grossberg [313, 314, 255, 315]).

1.4 From 1980 to Today

The machine learning expansion of the past few decades started in the 1980s, under the influence of several clusters of researchers, which originally worked fairly independently of each other and whose influences progressively merged. Three of these clusters were based in Southern California, the exception being Leslie Valiant at Harvard who was laying some of the foundations of computational learning theory by developing the probably approximately correct (PAC) framework [743]. Around the same time, Judea Pearl at UCLA was developing the theory and algorithms for Bayesian networks [567]. At Caltech, John Hopfield developed a discrete model of associative memory [355], followed by an analog version for optimization and other purposes [357, 720], that were particularly influential at the time. For one, these models established important connections to statistical mechanics, in particular the theory of spin glasses, and brought many physicists to the field from around the world. Other faculty members at the time at Caltech that were drawn into neural networks included Edward Posner, Robert McEliece, Demetri Psaltis, Yaser Abu-Mostafa, and Carver Mead who used analog VLSI to build some of the first analog neuromorphic circuits [496, 470, 495] with Misha Mahowald (1963–1996). Hopfield's connections also brought an influx of scientists and research from the Bell Laboratories. The fourth cluster, and perhaps the one which over time contributed the most to practical applications of neural networks, was the Parallel Distributed Processing (PDP) group at UCSD (and also CMU) led by David Rumelhart and James McClelland [490]. The PDP group made several contributions that are studied in detail in this book including the concepts of supervised and unsupervised learning, autoencoders and Boltzmann machines [9], and of course it co-developed and popularized the backpropagation learning algorithm, basically stochastic gradient descent for neural networks using the chain rule of calculus [627].

The basic backpropagation learning rule can be written as:

$$\Delta w_{ij} \propto B_i O_j \tag{1.4}$$

where O_j is the output of the presynaptic neuron and B_i is a postsynaptic backpropagated error obtained by applying the chain rule. While in this form the rule still seems to have a supervised Hebbian flavor with a quadratic form, it is important to note that the backpropagated error B_i is not the activity of the postsynaptic neuron. This again will have important consequences in later analyses.

Backpropagation is the single most important and most practical algorithm in this book, and the PDP group played a role in demonstrating its capabilities and promoting its early adoption in the mid-1980s. Terry Sejnowski, for instance, used it to pioneer

https://doi.org/10.1017/9781108955652.002 Published online by Cambridge University Press

several applications in speech synthesis (NETtalk)[665] and protein secondary structure prediction [588]. Finally, using backpropagation, it became possible to train the convolutional neural networks that Fukishima had originally proposed, in applications such as handwritten character recognition [219] or biometric fingerprint recognition [76].

The mid-1980s also correspond to the time when the first sustained machine learning conferences started, originally run by a combination of scientists from Caltech, Bell Labs, and the PDP group; beginning with a meeting in Santa Barbara, CA in 1985, followed by the first annual, by-invitation only Snowbird, UT conference in 1986 which also led to the first NIPS – now renamed NeurIPS – conference in Denver, CO and the establishment of the NIPS Foundation by Edward Posner in 1987.

By the end of the 1980s, the word "HMMs" (hidden Markov models), coming from the scientists using them in speech recognition, could be heard with increasing frequency at these conferences. In a few short years, this led to several developments including:

(1) the development of HMMs for bioinformatics applications which ushered machine learning methods into biology at the time of the human genome sequencing project [423, 79];
(2) the first attempts at combining graphical models with neural networks leading to, for instance, hybrid HMM/Neural Network (NN) and recursive neural networks [78, 99, 280];
(3) the recognition that HMMs and their learning algorithms (e.g. the EM algorithm) were special cases of a more general theory of graphical models [689] and probabilistic inference, the foundations of which had already been laid by Judea Pearl and several others.

Thus the 1990s saw a rapid expansion of research in all areas of probabilistic graphical models [696, 282, 386, 417, 234], which for almost a decade became the dominant topic at NIPS, and more broadly in machine learning. Somewhat like "HMMs", SVMs (support vector machines) [211, 217] – a sophisticated form of shallow learning – started at some point to gain popularity and led to a rapid expansion of research in the more general area of kernel methods [656, 657], which also became a dominant topic in machine learning for another decade or so.

Although graphical models and kernel methods were the dominant topics in the 1990s and 2000s, the idea of a second "neural network winter" has been overblown, just like the first one, and for similar reasons. A simple inspection of the literature shows that research on neural networks by several groups around the world continued unabated, leading to several new results. Among many others, these include: the development of LSTMs (long–short term memory units) [348]; the development of recursive deep learning methods [280]; and the application of recursive neural networks to a host of problems ranging from protein secondary structure prediction [75, 579], to protein contact map prediction [88, 237], and to the game of GO [791, 792], to name just a few.

Starting around 2011, convolutional neural networks implemented on Graphical Processing Units (GPUs) were used to win international computer vision competitions, such as the IJCNN 2011 traffic site recognition contest [199], achieving super-human performance levels. A successful application of GPUs and convolutional neural networks to

https://doi.org/10.1017/9781108955652.002 Published online by Cambridge University Press

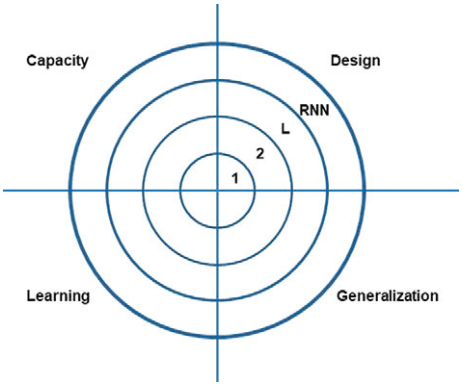

Figure 1.4 Roadmap. Concentric circles correspond to neural networks of increasing complexity, starting from single neurons, going to feedforward neural networks with increasing number L of layers, and then to recurrent and recursive neural networks. The sectors provide examples of important topics that need to be investigated for each level of complexity.

the ImageNet benchmark data set [422], involving million of images across thousands of categories, contributed significantly to a broader recognition of neural networks and their rebranding as "deep learning".

The current period of expansion of neural networks and deep learning into most areas of science and technology is driven by three major technological factors:

(1) the availability of unprecedented computing power, primarily in the form of GPUs;
(2) the availability of large training sets, thanks to progress in sensors, databases, the Internet, open-source projects, and so forth – although it is important to note that deep learning methods can also be adapted and deployed in scenarios where data is less abundant; and
(3) the development of well-maintained, industrial-strength, software libraries for neural networks such as TensorFlow, PyTorch, and Keras.

1.5 Roadmap

A mental roadmap for the rest of this book is depicted in Figure 1.4. We will start from the center of the bullseye and progressively expand towards the periphery. The concentric circles correspond to networks of increasing power and complexity, starting from feedforward networks with one layer, two layers, an arbitrary number L layers, and then recurrent and recursive networks. In the angular dimension, the sectors correspond to fundamental topics that need to be studied for each level of complexity. Examples of sectors we will study include the design and capacity of these networks, the corresponding learning algorithms and generalization capabilities. There is not yet a complete theory for each one of the resulting "boxes", however, as we shall see, there is substantial theory for many of them. The flip side of deep learning being like linear regression on steroids is that, like linear regression, it can be applied to essentially any domain provided one

https://doi.org/10.1017/9781108955652.002 Published online by Cambridge University Press

is willing to collaborate with domain experts. There are plenty of applications of deep learning to engineering and other problems that are covered in many articles and in other books. Here we will focus on applications of deep learning to problems in the natural sciences.

1.6 Exercises

EXERCISE 1.1 Identify as many other forms of computing as possible and evaluate their current status.

EXERCISE 1.2 Identify and discuss different forms of virtualization that may be contained in Figure 1.1.

EXERCISE 1.3 Why may it not be too surprising to find that both brains and computers use electrical signals to process information?

EXERCISE 1.4 Provide a back-of-the-envelope calculation for estimating the energetic cost of storing information, over the long-term, using patterns of electrical activity in brains or computers.

EXERCISE 1.5 Provide a back-of-the-envelope calculation for estimating how much power the brain requires for its operation, and compare it to the power required by the entire organism.

EXERCISE 1.6 Provide a back-of-the-envelope calculation for estimating how much power is required to power a desktop, versus a supercomputer, versus a server farm.

EXERCISE 1.7 The olfactory cortex appears to be close to the nose, and the auditory cortex appears to be close to the ears. Provide plausible explanations for why our primary visual cortex is located in the back of our heads, far away from the eyes.

https://doi.org/10.1017/9781108955652.002 Published online by Cambridge University Press