Part I

Foundations

1 Introduction

In the fifteenth century, the printing press revolutionized the world by overcoming the genomic bottleneck that allows for only two billion characters of our DNA to be passed on from generation to generation. The printed text allows for unlimited knowledge to be passed on between generations.

A common distinction between the capabilities of humans and machines is that humans are generalists and machines are specialists. The deep learning revolution has resulted in many specialized machine learning systems with super-human capabilities under the title AlphaX. A few noteworthy examples are AlphaGo (Silver et al., 2016) for playing Go, AlphaZero (Silver et al., 2017) for playing chess, AlphaHoldem (Zhao et al., 2022) for playing poker, AlphaD3M (Drori, Krishnamurthy, Rampin, Lourenco, One, Cho, Silva and Freire, 2018) for automated machine learning, AlphaStock (Wang, Zhang, Tang, Wu and Xiong, 2019) for trading stocks, AlphaStar (Vinyals et al., 2019) for playing multi-player strategy games, AlphaDogfight (Pope et al., 2021) for flying fighter jets, AlphaFold (Jumper et al., 2021) for protein structure prediction, and AlphaCode (Li et al., 2022) for competition-level code generation.

In contrast, recent deep learning Transformers, also called foundation models, trained with one trillion parameters, are generalists. Consider the task of learning a university-level course. A human may learn at most a few hundred courses with great effort during an entire lifetime, whereas a foundation model is soon able to learn all courses in days with super-human performance. Understanding such a machine is very different from that of a human.

Deep learning and artificial intelligence (AI) are revolutionizing the world again in the twenty-first century by overcoming the human perception of reality, which is limited by our brains and senses. Machines are revealing to humans insights and new understandings of reality, in which, by comparison, individual human capabilities are mundane.

1.1 Deep Learning

Deep learning is narrowly defined as optimizing neural networks that have many layers. In the broader sense, deep learning encompasses all methods, architectures, and applications involving neural network representations. Deep neural

networks are inspired by neurons and their connections in the brain. The backpropagation algorithm is the most commonly used approach for optimizing deep neural networks. Backpropagation is based on computing gradients of a loss function using the chain rule in reverse mode differentiation. Backpropagation and gradient-based methods for optimizing neural networks are very different from biological learning mechanisms in the brain. Deep neural networks may also be optimized using genetic algorithms or Hebbian learning rules, which are inspired by learning in biological neural networks. This book focuses on the broad definition of deep learning, encompassing methods, architectures, and applications that use neural network representations optimized using backpropagation.

1.2 Outline

The book is divided into five parts: (1) *Foundations*, (2) *Architectures*, (3) *Generative Models*, (4) *Reinforcement Learning*, and (5) *Applications*.

1.2.1 Part I: Foundations: Backpropagation, Optimization, and Regularization

Part I, *Foundations*, consists of three chapters. Chapter 2 defines neural networks and presents forward propagation and backpropagation. Neural networks are defined as a composition of functions consisting of a linear and a non-linear part. The chapter defines the network inputs, pre-activations, non-linear activation functions, activation units, and outputs. These are used to introduce forward propagation in neural networks. Next, the chapter presents loss functions and their gradients, derivatives of non-linear activation functions, and the chain rule. These are used to explain backpropagation in a neural network, which is the cornerstone of training neural networks by gradient descent. Multiple examples illustrate the algorithms and provide the backpropagation derivations using the chain rule in reverse mode differentiation. Finally, the chapter presents initialization and normalization strategies for neural networks and the key deep learning software libraries and platforms.

Chapter 3 presents optimization in deep learning, focusing on gradient descent which iteratively finds a local minimum by taking steps in the direction of the steepest descent. Three main problems with training neural networks using gradient descent and their solutions are discussed: (1) the total loss function with respect to the neural network weights, which is a sum of many individual losses for many samples – the solution is mini-batch or stochastic gradient descent; (2) the derivative of the total loss, which is computed with respect to all of the network weights – the solution is backpropagation; and (3) the directions of gradients for consecutive time steps which follow optimized step sizes are orthogonal, forming a zig-zag pattern, which is slow, especially in flat regions. The solution is adaptive gradient descent methods that use previous gradients to determine the step size. Next, the chapter presents second-order methods, including

practical quasi-Newton approaches. Finally, the chapter discusses gradient-free optimization approaches such as evolution strategies.

Chapter 4 presents regularization as a technique that can be used to prevent overfitting and explains generalization, bias, and variance. The chapter presents three methods for regularization: (1) adding a penalty term to the cost function – the penalty term is usually a function of the number of parameters in the model; (2) dropout, which is a technique that randomly sets several of the weights in a neural network to zero, which helps to prevent overfitting by reducing the variance of the network; and (3) data augmentation, which is a technique that involves modifying the input data to the neural network by applying random transformations. This technique also helps prevent overfitting by increasing the size of the training set.

1.2.2 Part II: Architectures: CNNs, RNNs, GNNs, and Transformers

Part II, *Architectures*, is about deep learning architectures and consists of four chapters. The first three chapters in this part present successful deep learning representations since they share weights across space, time, or neighborhoods.

Chapter 5 presents convolutional neural networks (CNNs), which are a type of neural network that is designed to recognize patterns in images. The network comprises a series of layers, with each layer performing a specific function. The first layer is typically a convolutional layer, which performs a convolution operation on the input image. The convolution operation is a mathematical operation that extracts information from the input image. The output of the convolutional layer is then passed to a pooling layer, which reduces the number of neurons in the network. Multiple convolutions and pooling layers are followed by a series of fully connected layers responsible for classification or other applications performed on the image. Convolutional neural networks perform well in practice across a broad range of applications since they share weights at multiple scales across space. Finally, the chapter describes CNN architectures such as residual neural networks (ResNets), DenseNets, and ODENets.

Chapter 6 introduces recurrent neural networks (RNNs), which share weights across time. This chapter describes backpropagation through time, its limitations, and the solutions in the form of long short-term memory (LSTM) and gated-recurrent unit (GRU). Next, the chapter describes sequence-to-sequence models, followed by encoder–decoder attention and self-attention and embeddings.

Chapter 7 presents graph neural networks (GNNs), which share weights across neighborhoods. The chapter begins with the definitions of graphs and their representations. Graph neural networks are introduced and applied to irregular structures such as networks. They are used for three tasks: (1) predicting properties of nodes; (2) predicting properties of edges; and (3) predicting properties of sub-graphs or properties of entire graphs.

The second part of the book concludes with Chapter 8, which covers state-of-

the-art Transformers, also known as foundation models, which have become a mainstream architecture in deep learning. Transformers have disrupted various fields, including natural language processing, computer vision, audio processing, programming, and education. Large Transformer models currently consist of more than one trillion parameters, and the number of parameters of Transformers is increasing by orders of magnitude each year; it is on track to surpass the number of connections in the human brain. Transformers may be classified into three types of architectures: (1) autoencoding Transformers, which is a stack of encoders; (2) auto-regressive Transformers, which is a stack of decoders; and (3) sequence-to-sequence Transformers, which is a stack of encoders connected to a stack of decoders. New scalable deep learning architectures such as Transformers are revolutionizing how machines perceive the world, make decisions, and generate novel output.

1.2.3 Part III: Generative Models: GANs, VAEs, and Normalizing Flows

The task of classification maps a set of examples to a label. In contrast, generative models map a label to a set of examples. Part III, *Generative Models*, consists of two chapters.

Chapter 9 introduces generative adversarial network (GAN) theory, practice, and applications. The chapter begins by describing the roles of the generator and discriminator. Next, the advantages and limitations of different loss functions are described. Generative adversarial network training algorithms are presented, discussing the issues of mode collapse and vanishing gradients while providing state-of-the-art solutions. Finally, the chapter concludes with a broad range of applications of GANs.

Chapter 10 introduces variational inference and its extension to black-box variational inference used in practice for inference on large datasets. Both reverse Kullback–Leibler (KL) and forward KL approaches are presented. The chapter covers the variational autoencoder algorithm, which consists of an encoder neural network for inference and a decoder network for generation, trained end-to-end by backpropagation. The chapter describes how the variational approximation of the posterior is improved using a series of invertible transformations, known as normalizing flows, in both discrete and continuous domains. Finally, state-of-the-art examples of deep variational inference on manifolds are presented.

1.2.4 Part IV: Reinforcement Learning

Part IV covers *Reinforcement Learning*, a type of machine learning in which an agent learns by interacting with an environment.

Chapter 11 begins by defining a stateless multi-armed bandit, presenting the trade-off between exploration and exploitation. Next, the chapter covers basic principles of state machines and Markov decision processes (MDPs) with

known transition and reward functions. Finally, the chapter presents reinforcement learning in which the transition and reward functions are unknown, and therefore the agent interacts with the environment by sampling the world. Monte Carlo sampling and temporal difference sampling are described with examples. The chapter concludes by presenting the Q-learning algorithm.

Chapter 12 presents deep reinforcement learning through value-based methods, policy-based methods, and actor–critic methods. Value-based methods covered include deep Q-networks and present prioritized replay. Policy-based methods described include policy gradients and REINFORCE. Next, the chapter covers actor–critic methods, including advantage actor–critic and asynchronous advantage actor–critic. Advanced hybrid approaches, such as natural policy gradient, trust region policy optimization, proximal policy optimization, and a deep deterministic policy gradient, are presented. Next, the chapter covers model-based reinforcement learning approaches, including Monte Carlo tree search (MCTS), AlphaZero, and world models. The chapter concludes by presenting imitation learning and exploration strategies for environments with sparse rewards.

1.2.5 Part V: Applications

The book concludes with Part V, which covers a dozen state-of-the-art applications of deep learning in a broad range of domains: autonomous vehicles, climate change and monitoring, computer vision, audio processing, voice swapping, music synthesis, natural language processing, automated machine learning, learning-to-learn courses, protein structure prediction and docking, combinatorial optimization, computational fluid dynamics, and plasma physics. Each deep learning application is briefly described, along with a visualization or system architecture.

1.2.6 Appendices

The first appendix, Matrix Calculus, defines the partial derivatives of a function with respect to variables and is helpful for gradient computations in backpropagation and optimization. The second appendix summarizes best practices in scientific writing and reviewing. A section on scientific writing addresses the abstract, introduction, related work, the structure of the text, figures, captions, results, discussion, and the reader's perspective and provides this book's style sheet. A section on reviewing explains the review process, including best practices for evaluating and rating scientific work and writing a rebuttal.

1.3 Code

Hundreds of Python functions are automatically generated on each topic for each chapter by program synthesis using deep learning. All of the code is made available on the book's website at `www.dlbook.org`.

1.4 Exercises

Each chapter has around a dozen human-generated theoretical and programming exercises and their solutions. In addition, hundreds of questions and solutions on each topic are automatically generated by program synthesis using deep learning. All questions and solutions are made available on the book's website at `www.dlbook.org`.