
Understanding Deep Learning

Simon J.D. Prince

May 27, 2024

If you enjoy this book, here are four ways you can help me:

1. Spread the word via social media. Posts in languages other than English particularly welcome. Tag me on [LinkedIn](#) or [X](#) and I'll probably say hi.
2. Write me an Amazon review. Preferably positive, but all publicity is good publicity...
3. Send me comments (see bottom of this page). I reply to everything eventually.
4. Buy a copy. I took 18 months completely off work to write this book and ideally I'd like to make minimum wage (or better) for this time. Also, I'd like to write a second edition, but I need to sell enough copies to do this. Thanks!

The most recent version of this document can be found at <http://udlbook.com>.

Copyright in this work has been licensed exclusively to The MIT Press,
<https://mitpress.mit.edu>, which released the final version to the public in December 2023.
Inquiries regarding rights should be addressed to the MIT Press, Rights & Permissions
Department.

This work is subject to a [Creative Commons CC-BY-NC-ND license](#).

I would really appreciate help improving this document. No detail too small! Please contact me with suggestions, factual inaccuracies, ambiguities, questions, and errata via [github](#) or by e-mail at udlbookmail@gmail.com.



This book is dedicated to Blair, Calvert, Coppola, Ellison, Faulkner, Kerpatenko, Morris, Robinson, Sträussler, Wallace, Waymon, Wojnarowicz, and all the others whose work is even more important and interesting than deep learning.



Contents

Preface	ix
Acknowledgements	xi
1 Introduction	1
1.1 Supervised learning	1
1.2 Unsupervised learning	7
1.3 Reinforcement learning	11
1.4 Ethics	12
1.5 Structure of book	15
1.6 Other books	15
1.7 How to read this book	16
2 Supervised learning	17
2.1 Supervised learning overview	17
2.2 Linear regression example	18
2.3 Summary	22
3 Shallow neural networks	25
3.1 Neural network example	25
3.2 Universal approximation theorem	29
3.3 Multivariate inputs and outputs	30
3.4 Shallow neural networks: general case	33
3.5 Terminology	35
3.6 Summary	36
4 Deep neural networks	41
4.1 Composing neural networks	41
4.2 From composing networks to deep networks	43
4.3 Deep neural networks	45
4.4 Matrix notation	48
4.5 Shallow vs. deep neural networks	49
4.6 Summary	52

5 Loss functions	56
5.1 Maximum likelihood	56
5.2 Recipe for constructing loss functions	60
5.3 Example 1: univariate regression	61
5.4 Example 2: binary classification	64
5.5 Example 3: multiclass classification	67
5.6 Multiple outputs	69
5.7 Cross-entropy loss	71
5.8 Summary	72
6 Fitting models	77
6.1 Gradient descent	77
6.2 Stochastic gradient descent	83
6.3 Momentum	86
6.4 Adam	88
6.5 Training algorithm hyperparameters	91
6.6 Summary	91
7 Gradients and initialization	96
7.1 Problem definitions	96
7.2 Computing derivatives	97
7.3 Toy example	100
7.4 Backpropagation algorithm	103
7.5 Parameter initialization	107
7.6 Example training code	111
7.7 Summary	111
8 Measuring performance	118
8.1 Training a simple model	118
8.2 Sources of error	120
8.3 Reducing error	124
8.4 Double descent	127
8.5 Choosing hyperparameters	132
8.6 Summary	133
9 Regularization	138
9.1 Explicit regularization	138
9.2 Implicit regularization	141
9.3 Heuristics to improve performance	144
9.4 Summary	154
10 Convolutional networks	161
10.1 Invariance and equivariance	161
10.2 Convolutional networks for 1D inputs	163
10.3 Convolutional networks for 2D inputs	170

10.4	Downsampling and upsampling	171
10.5	Applications	174
10.6	Summary	179
11	Residual networks	186
11.1	Sequential processing	186
11.2	Residual connections and residual blocks	189
11.3	Exploding gradients in residual networks	192
11.4	Batch normalization	192
11.5	Common residual architectures	195
11.6	Why do nets with residual connections perform so well?	199
11.7	Summary	199
12	Transformers	207
12.1	Processing text data	207
12.2	Dot-product self-attention	208
12.3	Extensions to dot-product self-attention	213
12.4	Transformer layers	215
12.5	Transformers for natural language processing	216
12.6	Encoder model example: BERT	219
12.7	Decoder model example: GPT3	222
12.8	Encoder-decoder model example: machine translation	226
12.9	Transformers for long sequences	227
12.10	Transformers for images	228
12.11	Summary	232
13	Graph neural networks	240
13.1	What is a graph?	240
13.2	Graph representation	243
13.3	Graph neural networks, tasks, and loss functions	245
13.4	Graph convolutional networks	248
13.5	Example: graph classification	251
13.6	Inductive vs. transductive models	252
13.7	Example: node classification	253
13.8	Layers for graph convolutional networks	256
13.9	Edge graphs	260
13.10	Summary	261
14	Unsupervised learning	268
14.1	Taxonomy of unsupervised learning models	268
14.2	What makes a good generative model?	269
14.3	Quantifying performance	271
14.4	Summary	273
15	Generative Adversarial Networks	275

15.1	Discrimination as a signal	275
15.2	Improving stability	280
15.3	Progressive growing, minibatch discrimination, and truncation	286
15.4	Conditional generation	288
15.5	Image translation	290
15.6	StyleGAN	295
15.7	Summary	297
16	Normalizing flows	303
16.1	1D example	303
16.2	General case	306
16.3	Invertible network layers	308
16.4	Multi-scale flows	316
16.5	Applications	317
16.6	Summary	320
17	Variational autoencoders	326
17.1	Latent variable models	326
17.2	Nonlinear latent variable model	327
17.3	Training	330
17.4	ELBO properties	333
17.5	Variational approximation	335
17.6	The variational autoencoder	335
17.7	The reparameterization trick	338
17.8	Applications	339
17.9	Summary	342
18	Diffusion models	348
18.1	Overview	348
18.2	Encoder (forward process)	349
18.3	Decoder model (reverse process)	355
18.4	Training	356
18.5	Reparameterization of loss function	360
18.6	Implementation	362
18.7	Summary	367
19	Reinforcement learning	373
19.1	Markov decision processes, returns, and policies	373
19.2	Expected return	377
19.3	Tabular reinforcement learning	381
19.4	Fitted Q-learning	385
19.5	Policy gradient methods	388
19.6	Actor-critic methods	393
19.7	Offline reinforcement learning	394
19.8	Summary	395

20 Why does deep learning work?	401
20.1 The case against deep learning	401
20.2 Factors that influence fitting performance	402
20.3 Properties of loss functions	406
20.4 Factors that determine generalization	410
20.5 Do we need so many parameters?	414
20.6 Do networks have to be deep?	417
20.7 Summary	418
21 Deep learning and ethics	420
21.1 Value alignment	420
21.2 Intentional misuse	426
21.3 Other social, ethical, and professional issues	428
21.4 Case study	430
21.5 The value-free ideal of science	431
21.6 Responsible AI research as a collective action problem	432
21.7 Ways forward	433
21.8 Summary	434
A Notation	436
B Mathematics	439
B.1 Functions	439
B.2 Binomial coefficients	441
B.3 Vector, matrices, and tensors	442
B.4 Special types of matrix	445
B.5 Matrix calculus	447
C Probability	448
C.1 Random variables and probability distributions	448
C.2 Expectation	452
C.3 Normal probability distribution	456
C.4 Sampling	459
C.5 Distances between probability distributions	459
Bibliography	462
Index	513



Preface

The history of deep learning is unusual in science. The perseverance of a small cabal of scientists, working over twenty-five years in a seemingly unpromising area, has revolutionized a field and dramatically impacted society. Usually, when researchers investigate an esoteric and apparently impractical corner of science or engineering, it remains just that — esoteric and impractical. However, this was a notable exception. Despite widespread skepticism, the systematic efforts of Yoshua Bengio, Geoffrey Hinton, Yann LeCun, and others eventually paid off.

The title of this book is “Understanding Deep Learning” to distinguish it from volumes that cover coding and other practical aspects. This text is primarily about the *ideas* that underlie deep learning. The first part of the book introduces deep learning models and discusses how to train them, measure their performance, and improve this performance. The next part considers architectures that are specialized to images, text, and graph data. These chapters require only introductory linear algebra, calculus, and probability and should be accessible to any second-year undergraduate in a quantitative discipline. Subsequent parts of the book tackle generative models and reinforcement learning. These chapters require more knowledge of probability and calculus and target more advanced students.

The title is also partly a joke — *no-one* really understands deep learning at the time of writing. Modern deep networks learn piecewise linear functions with more regions than there are atoms in the universe and can be trained with fewer data examples than model parameters. It is neither obvious that we should be able to fit these functions reliably nor that they should generalize well to new data. The penultimate chapter addresses these and other aspects that are not yet fully understood. Regardless, deep learning will change the world for better or worse. The final chapter discusses AI ethics and concludes with an appeal for practitioners to consider the moral implications of their work.

Your time is precious, and I have striven to curate and present the material so you can understand it as efficiently as possible. The main body of each chapter comprises a succinct description of only the most essential ideas, together with accompanying illustrations. The appendices review all mathematical prerequisites, and there should be no need to refer to external material. For readers wishing to delve deeper, each chapter has associated problems, Python notebooks, and extensive background notes.

Writing a book is a lonely, grinding, multiple-year process and is only worthwhile if the volume is widely adopted. If you enjoy reading this or have suggestions for improving it, please contact me via the accompanying website. I would love to hear your thoughts, which will inform and motivate subsequent editions.



Acknowledgments

Writing this book would not have been possible without the generous help and advice of these individuals: Kathryn Hume, Kevin Murphy, Christopher Bishop, Peng Xu, Yann Dubois, Justin Domke, Chris Fletcher, Yanshuai Cao, Wendy Tay, Corey Toler-Franklin, Dmytro Mishkin, Guy McCusker, Daniel Worrall, Paul McIlroy, Roy Amoyal, Austin Anderson, Romero Barata de Morais, Gabriel Harrison, Peter Ball, Alf Muir, David Bryson, Vedika Parulkar, Patryk Lietzau, Jessica Nicholson, Alexa Huxley, Oisin Mac Aodha, Giuseppe Castiglione, Josh Akyrbekov, Alex Gouglaki, Joshua Omilabu, Alister Guenther, Joe Goodier, Logan Wade, Joshua Guenther, Kylan Tobin, Benedict Ellett, Jad Araj, Andrew Glennerster, Giorgos Sfikas, Diya Vibhakar, Sam Mansat-Bhattacharyya, Ben Ross, Ivor Simpson, Gaurang Aggarwal, Shakeel Sheikh, Jacob Horton, Felix Rammell, Sasha Luccioni, Akshil Patel, Alessandro Gentilini, Kevin Mercier, Krzysztof Lichocki, Chuck Krapf, Brian Ha, Chris Kang, Leonardo Viotti, Kai Li, Himan Abdollahpouri, Ari Pakman, Giuseppe Antonio Di Luna, Dan Oneață, Conrad Whiteley, Joseph Santarcangelo, Brad Shook, Gabriel Brostow, Lei He, Ali Satvaty, Romain Sabathé, Qiang Zhou, Prasanna Vigneswaran, Siqi Zheng, Stephan Grein, Jonas Klesen, Giovanni Stilo, Huang Bokai, Kevin McGuinness, Qiang Sun, Zakaria Lotfi, Yifei Lin, Sylvain Bouix, Alex Pitt, Stephane Chretien, Robin Liu, Bian Li, Adam Jones, Marcin Świerkot, Tommy Löfstedt, Eugen Hotaj, Fernando Flores-Mangas, Tony Polichroniadis, Pietro Monticone, Rohan Deepak Ajwani, Menashe Yarden Einy, Robert Gevorgyan, Thilo Stadelmann, Gui JieMiao, Botao Zhu, Mohamed Elabbas, Satya Krishna Gorti, James Elder, Helio Perroni Filho, Xiaochao Qu, Jaekang Shin, Joshua Evans, Robert Dobson, Shibo Wang, Edoardo Zorzi, Stanisław Jastrzębski, Pieris Kalligerous, Matt Hewitt, Zviika Haramaty, Ted Mavroidis, Nikolaj Kuntner, Amir Yorav, Massoud Mokhtari, Xavier Gabaix, Marco Garosi, Vincent Schönbach, Avishek Mondal, Victor S.C. Lui, Sumit Bhatia, Julian Asilis, Hengchao Chen, Siavash Khallaghi, Csaba Szepesvári, Mike Singer, Mykhailo Shvets, Abdalla Ibrahim, Stefan Hell, Ron Raphaeli, Diogo Tavares, Aristotelis Siozopoulos, Jianrui Wu, Jannik Münz, Penn Mackintosh, Shawn Hoareau, Qianang Zhou, Emma Li, Charlie Groves, Xiang Lingxiao, Trivikram Muralidharan, Rajat Binaykiya, Germán del Cacho Salvador, Alexey Bloudov, Paul Colognese, Bo Yang, Jani Monoses, Adenilson Arcanjo, Matan Golani, Emmanuel Onzon, Shenghui Yan, Kamesh Kompella, Julius Aka, Johannes Brunnemann, Varniethan Kettheeswaran, Alex Ostrovsky, Daniel Burbank, Gavrie Philipson, Roozbeh Ehsani, Len Spek, Christoph Brune, Mohammad Nosrati, Bian Li, Runqi Chen, Qifu Hu, Rasmi Elasmar, Ronaldo Butrus, Carles Mesado, Jeffrey Wolberg, Olivier Koch, Edoardo Lanari, Fanmin Shi, Neel Maniar, Maksym Taran, Falk Langhammer, Reinaldo Lepsch, Max Talberg, Vishal Jain, Christian Arnold, Charles Hill, Nikita Panin, Steven Dillmann, Suhas Mathur, Harris Abdul Majid, Guolong Lin, Charles Elkan, Benedict Kuester, Vladimir Ivanov, Mohammad-Hadi Sotoudeh, Daniel Enériz Orta, Ian Jeffrey, Kwok Chun, Yu Liu, Tom Vettenburg, Aravinda Perera, Daniel Gigliotti, Iftikhar Ramnandan, Adnan Siddiquei, Will Knottenbelt, Valerio Di Stefano, Srikanth Jayaraman, Goldie Srulovich, Rafał Rolczyński, Anthony Ip, and Andre Coelho.

I'm particularly grateful to Daniyar Turmukhambetov, Amedeo Buonanno, Andrea Panizza, Mark Hudson, and Bernhard Pfahringer, who provided detailed comments on multiple chapters of the book. I'd like to especially thank Andrew Fitzgibbon, Konstantinos Derpanis, and Tyler Mills, who read the whole book and whose enthusiasm helped me complete this project. I'd also like to thank Neill Campbell and Özgür Şimşek, who hosted me at the University of Bath, where I taught a course based on this material for the first time. Finally, I'm extremely grateful to my editor Elizabeth Swayze for her frank advice throughout this process.

Chapter 12 (transformers) and chapter 17 (variational autoencoders) were first published as blogs for Borealis AI, and adapted versions are reproduced with permission of Royal Bank of Canada along with Borealis AI. I am grateful for their support in this endeavor. Chapter 16 (normalizing flows) is loosely based on the review article by Kobyzev et al. (2020), on which I was a co-author. I was very fortunate to be able to collaborate on Chapter 21 with Travis LaCroix from Dalhousie University, who was both easy and fun to work with, and who did the lion's share of the work.

Attribution

- Chessboard image in figure 1.13 adapted from <http://tinyurl.com/yc2d54d4>.
- Cogs image in figures 1.2, 1.4, 1.10 adapted from <http://tinyurl.com/2c7tttr8>.
- Penguin image in figures 19.1–19.5 and 19.6–19.9 adapted from <http://tinyurl.com/ycz9je56>.
- Fish image in figures 19.2–19.5, 19.7, 19.10–19.12 adapted from <http://tinyurl.com/4ueyhtsu>.

Chapter 1

Introduction

Artificial intelligence, or *AI*, is concerned with building systems that simulate intelligent behavior. It encompasses a wide range of approaches, including those based on logic, search, and probabilistic reasoning. *Machine learning* is a subset of AI that learns to make decisions by fitting mathematical models to observed data. This area has seen explosive growth and is now (incorrectly) almost synonymous with the term AI.

A *deep neural network* is a type of machine learning model, and when it is fitted to data, this is referred to as *deep learning*. At the time of writing, deep networks are the most powerful and practical machine learning models and are often encountered in day-to-day life. It is commonplace to translate text from another language using a *natural language processing* algorithm, to search the internet for images of a particular object using a *computer vision* system, or to converse with a digital assistant via a *speech recognition* interface. All of these applications are powered by deep learning.

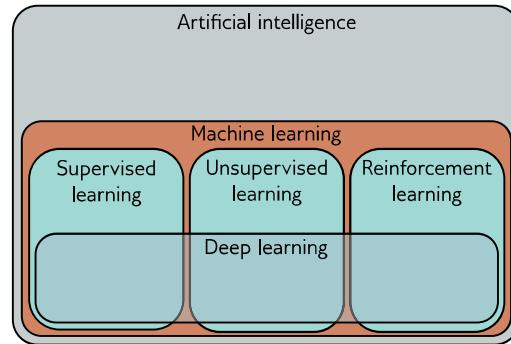
As the title suggests, this book aims to help a reader new to this field understand the principles behind deep learning. The book is neither terribly theoretical (there are no proofs) nor extremely practical (there is almost no code). The goal is to explain the underlying *ideas*; after consuming this volume, the reader will be able to apply deep learning to novel situations where there is no existing recipe for success.

Machine learning methods can coarsely be divided into three areas: supervised, unsupervised, and reinforcement learning. At the time of writing, the cutting-edge methods in all three areas rely on deep learning (figure 1.1). This introductory chapter describes these three areas at a high level, and this taxonomy is also loosely reflected in the book's organization. Whether we like it or not, deep learning is poised to change our world, and this change will not all be positive. Hence, this chapter also contains a brief primer on AI ethics. We conclude with advice on how to make the most of this book.

1.1 Supervised learning

Supervised learning models define a mapping from input data to an output prediction. In the following sections, we discuss the inputs, the outputs, the model itself, and what is meant by “training” a model.

Figure 1.1 Machine learning is an area of artificial intelligence that fits mathematical models to observed data. It can coarsely be divided into supervised learning, unsupervised learning, and reinforcement learning. Deep neural networks contribute to each of these areas.



1.1.1 Regression and classification problems

Figure 1.2 depicts several regression and classification problems. In each case, there is a meaningful real-world input (a sentence, a sound file, an image, etc.), and this is encoded as a vector of numbers. This vector forms the model input. The model maps the input to an output vector which is then “translated” back to a meaningful real-world prediction. For now, we focus on the inputs and outputs and treat the model as a black box that ingests a vector of numbers and returns another vector of numbers.

The model in figure 1.2a predicts the price of a house based on input characteristics such as the square footage and the number of bedrooms. This is a *regression* problem because the model returns a continuous number (rather than a category assignment). In contrast, the model in figure 1.2b takes the chemical structure of a molecule as an input and predicts both the freezing and boiling points. This is a *multivariate regression* problem since it predicts more than one number.

The model in figure 1.2c receives a text string containing a restaurant review as input and predicts whether the review is positive or negative. This is a *binary classification* problem because the model attempts to assign the input to one of two categories. The output vector contains the probabilities that the input belongs to each category. Figures 1.2d and 1.2e depict *multiclass classification* problems. Here, the model assigns the input to one of $N > 2$ categories. In the first case, the input is an audio file, and the model predicts which genre of music it contains. In the second case, the input is an image, and the model predicts which object it contains. In each case, the model returns a vector of size N that contains the probabilities of the N categories.

1.1.2 Inputs

The input data in figure 1.2 varies widely. In the house pricing example, the input is a fixed-length vector containing values that characterize the property. This is an example of *tabular data* because it has no internal structure; if we change the order of the inputs and build a new model, then we expect the model prediction to remain the same.

Conversely, the input in the restaurant review example is a body of text. This may be of variable length depending on the number of words in the review, and here *input*

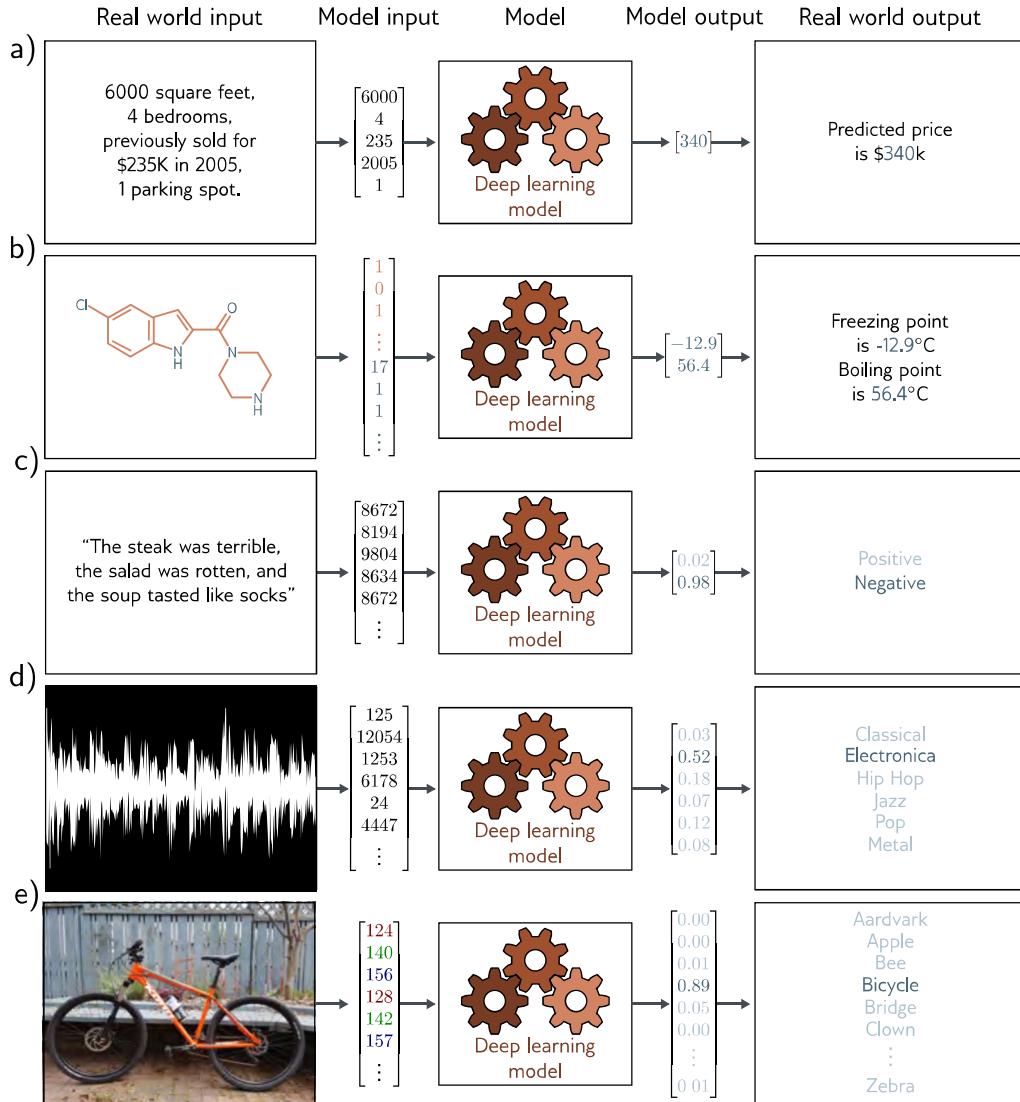


Figure 1.2 Regression and classification problems. a) This *regression* model takes a vector of numbers that characterize a property and predicts its price. b) This *multivariate regression* model takes the structure of a chemical molecule and predicts its freezing and boiling points. c) This *binary classification* model takes a restaurant review and classifies it as either positive or negative. d) This *multiclass classification* problem assigns a snippet of audio to one of N genres. e) A second multiclass classification problem in which the model classifies an image according to which of N possible objects it might contain.

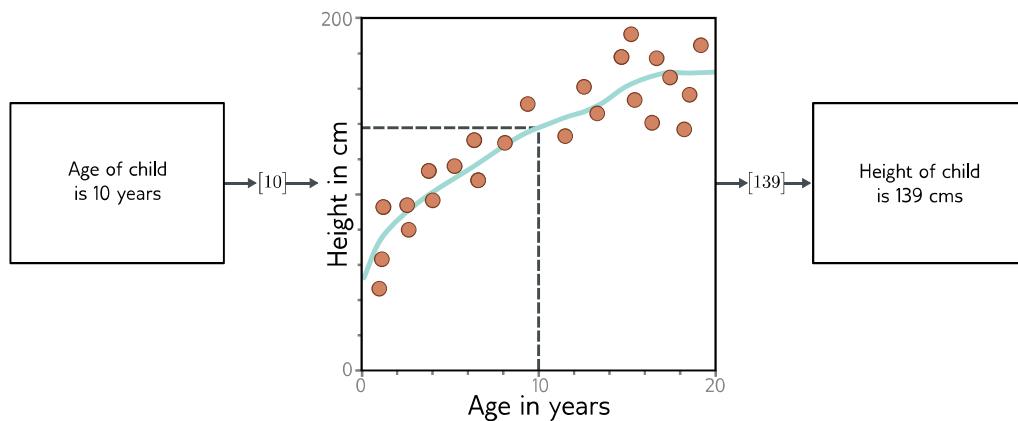


Figure 1.3 Machine learning model. The model represents a family of relationships that relate the input (age of child) to the output (height of child). The particular relationship is chosen using training data, which consists of input/output pairs (orange points). When we train the model, we search through the possible relationships for one that describes the data well. Here, the trained model is the cyan curve and can be used to compute the height for any age.

order is important; *my wife ate the chicken* is not the same as *the chicken ate my wife*. The text must be encoded into numerical form before passing it to the model. Here, we use a fixed vocabulary of size 10,000 and simply concatenate the word indices.

For the music classification example, the input vector might be of fixed size (perhaps a 10-second clip) but is very high-dimensional. Digital audio is usually sampled at 44.1 kHz and represented by 16-bit integers, so a ten-second clip consists of 441,000 integers. Clearly, supervised learning models will have to be able to process sizeable inputs. The input in the image classification example (which consists of the concatenated RGB values at every pixel) is also enormous. Moreover, its structure is naturally two-dimensional; two pixels above and below one another are closely related, even if they are not adjacent in the input vector.

Finally, consider the input for the model that predicts the freezing and boiling points of the molecule. A molecule may contain varying numbers of atoms that can be connected in different ways. In this case, the model must ingest both the geometric structure of the molecule and the constituent atoms to the model.

1.1.3 Machine learning models

Until now, we have treated the machine learning model as a black box that takes an input vector and returns an output vector. But what exactly is in this black box? Consider a model to predict the height of a child from their age (figure 1.3). The machine learning

model is a mathematical equation that describes how the average height varies as a function of age (cyan curve in figure 1.3). When we run the age through this equation, it returns the height. For example, if the age is 10 years, then we predict that the height will be 139 cm.

More precisely, the model represents a family of equations mapping the input to the output (i.e., a family of different cyan curves). The particular equation (curve) is chosen using *training data* (examples of input/output pairs). In figure 1.3, these pairs are represented by the orange points, and we can see that the model (cyan line) describes these data reasonably. When we talk about *training* or *fitting* a model, we mean that we search through the family of possible equations (possible cyan curves) relating input to output to find the one that describes the training data most accurately.

It follows that the models in figure 1.2 require labeled input/output pairs for training. For example, the music classification model would require a large number of audio clips where a human expert had identified the genre of each. These input/output pairs take the role of a teacher or supervisor for the training process, and this gives rise to the term *supervised learning*.

1.1.4 Deep neural networks

This book concerns deep neural networks, which are a particularly useful type of machine learning model. They are equations that can represent an extremely broad family of relationships between input and output, and where it is particularly easy to search through this family to find the relationship that describes the training data.

Deep neural networks can process inputs that are very large, of variable length, and contain various kinds of internal structures. They can output single real numbers (regression), multiple numbers (multivariate regression), or probabilities over two or more classes (binary and multiclass classification, respectively). As we shall see in the next section, their outputs may also be very large, of variable length, and contain internal structure. It is probably hard to imagine equations with these properties, and the reader should endeavor to suspend disbelief for now.

1.1.5 Structured outputs

Figure 1.4a depicts a multivariate binary classification model for semantic segmentation. Here, every pixel of an input image is assigned a binary label that indicates whether it belongs to a cow or the background. Figure 1.4b shows a multivariate regression model where the input is an image of a street scene and the output is the depth at each pixel. In both cases, the output is high-dimensional and structured. However, this structure is closely tied to the input, and this can be exploited; if a pixel is labeled as “cow,” then a neighbor with a similar RGB value probably has the same label.

Figures 1.4c–e depict three models where the output has a complex structure that is not so closely tied to the input. Figure 1.4c shows a model where the input is an audio file and the output is the transcribed words from that file. Figure 1.4d is a translation

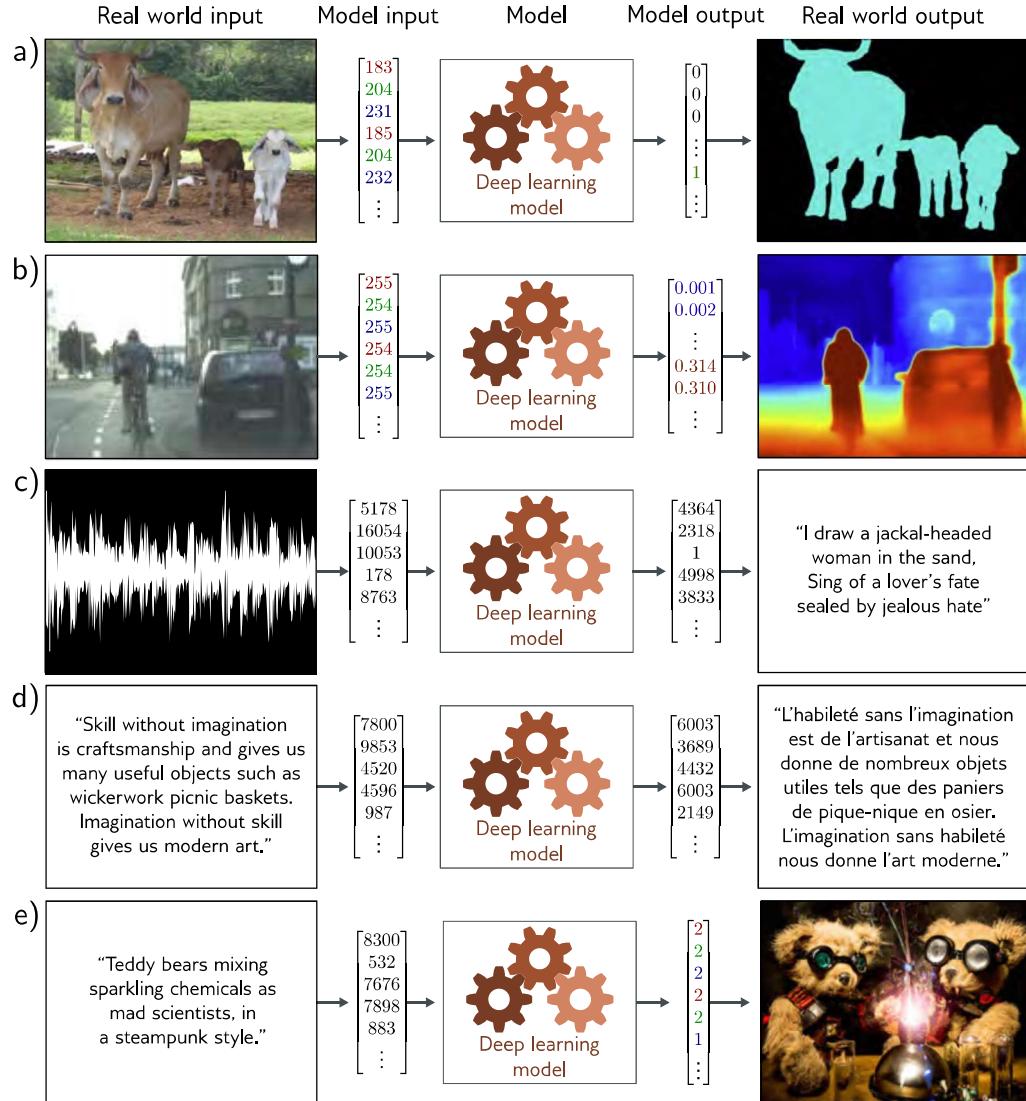


Figure 1.4 Supervised learning tasks with structured outputs. a) This semantic segmentation model maps an RGB image to a binary image indicating whether each pixel belongs to the background or a cow (adapted from Noh et al., 2015). b) This monocular depth estimation model maps an RGB image to an output image where each pixel represents the depth (adapted from Cordts et al., 2016). c) This audio transcription model maps an audio sample to a transcription of the spoken words in the audio. d) This translation model maps an English text string to its French translation. e) This image synthesis model maps a caption to an image (example from <https://openai.com/dall-e-2/>). In each case, the output has a complex internal structure or grammar. In some cases, many outputs are compatible with the input.

model in which the input is a body of text in English, and the output contains the French translation. Figure 1.4e depicts a very challenging task in which the input is descriptive text, and the model must produce an image that matches this description.

In principle, the latter three tasks can be tackled in the standard supervised learning framework, but they are more difficult for two reasons. First, the output may genuinely be ambiguous; there are multiple valid translations from an English sentence to a French one and multiple images that are compatible with any caption. Second, the output contains considerable structure; not all strings of words make valid English and French sentences, and not all collections of RGB values make plausible images. In addition to learning the mapping, we also have to respect the “grammar” of the output.

Fortunately, this “grammar” can be learned without the need for output labels. For example, we can learn how to form valid English sentences by learning the statistics of a large corpus of text data. This provides a connection with the next section of the book, which considers *unsupervised learning models*.

1.2 Unsupervised learning

Constructing a model from input data without corresponding output labels is termed *unsupervised learning*; the absence of output labels means there can be no “supervision.” Rather than learning a mapping from input to output, the goal is to describe or understand the structure of the data. As was the case for supervised learning, the data may have very different characteristics; it may be discrete or continuous, low-dimensional or high-dimensional, and of constant or variable length.

1.2.1 Generative models

This book focuses on *generative unsupervised models*, which learn to synthesize new data examples that are statistically indistinguishable from the training data. Some generative models explicitly describe the probability distribution over the input data and here new examples are generated by sampling from this distribution. Others merely learn a mechanism to generate new examples without explicitly describing their distribution.

State-of-the-art generative models can synthesize examples that are extremely plausible but distinct from the training examples. They have been particularly successful at generating images (figure 1.5) and text (figure 1.6). They can also synthesize data under the constraint that some outputs are predetermined (termed *conditional generation*). Examples include image inpainting (figure 1.7) and text completion (figure 1.8). Indeed, modern generative models for text are so powerful that they can appear intelligent. Given a body of text followed by a question, the model can often “fill in” the missing answer by generating the most likely completion of the document. However, in reality, the model only knows about the statistics of language and does not understand the significance of its answers.



Figure 1.5 Generative models for images. Left: two images were generated from a model trained on pictures of cats. These are not real cats, but samples from a probability model. Right: two images generated from a model trained on images of buildings. Adapted from Karras et al. (2020b).

The moon had risen by the time I reached the edge of the forest, and the light that filtered through the trees was silver and cold. I shivered, though I was not cold, and quickened my pace. I had never been so far from the village before, and I was not sure what to expect. I had been walking for hours, and I was tired and hungry. I had left in such a hurry that I had not thought to pack any food, and I had not thought to bring a weapon. I was unarmed and alone in a strange place, and I did not know what I was doing.

I had been walking for so long that I had lost all sense of time, and I had no idea how far I had come. I only knew that I had to keep going. I had to find her. I was getting close. I could feel it. She was nearby, and she was in trouble. I had to find her and help her, before it was too late.

Figure 1.6 Short story synthesized from a generative model of text data. The model describes a probability distribution that assigns a probability to every output string. Sampling from the model creates strings that follow the statistics of the training data (here, short stories) but have never been seen before.



Figure 1.7 Inpainting. In the original image (left), the boy is obscured by metal cables. These undesirable regions (center) are removed and the generative model synthesizes a new image (right) under the constraint that the remaining pixels must stay the same. Adapted from Saharia et al. (2022a).

I was a little nervous before my first lecture at the University of Bath. It seemed like there were hundreds of students and they looked intimidating. I stepped up to the lectern and was about to speak when something bizarre happened.

Suddenly, the room was filled with a deafening noise, like a giant roar. It was so loud that I couldn't hear anything else and I had to cover my ears. I could see the students looking around, confused and frightened. Then, as quickly as it had started, the noise stopped and the room was silent again.

I stood there for a few moments, trying to make sense of what had just happened. Then I realized that the students were all staring at me, waiting for me to say something. I tried to think of something witty or clever to say, but my mind was blank. So I just said, "Well, that was strange," and then I started my lecture.

Figure 1.8 Conditional text synthesis. Given an initial body of text (in black), generative models of text can continue the string plausibly by synthesizing the "missing" remaining part of the string. Generated by GPT3 (Brown et al., 2020).

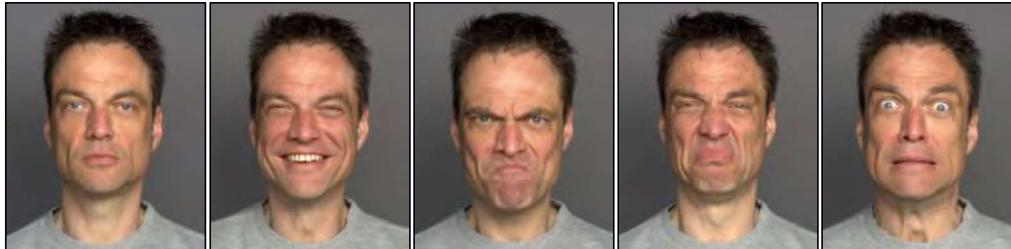


Figure 1.9 Variation of the human face. The human face contains roughly 42 muscles, so it's possible to describe most of the variation in images of the same person in the same lighting with just 42 numbers. In general, datasets of images, music, and text can be described by a relatively small number of underlying variables although it is typically more difficult to tie these to particular physical mechanisms. Images from Dynamic FACES database (Holland et al., 2019).

1.2.2 Latent variables

Some (but not all) generative models exploit the observation that data can be lower dimensional than the raw number of observed variables suggests. For example, the number of valid and meaningful English sentences is considerably smaller than the number of strings created by drawing words at random. Similarly, real-world images are a tiny subset of the images that can be created by drawing random RGB values for every pixel. This is because images are generated by physical processes (see figure 1.9).

This leads to the idea that we can describe each data example using a smaller number of underlying *latent variables*. Here, the role of deep learning is to describe the mapping between these latent variables and the data. The latent variables typically have a simple

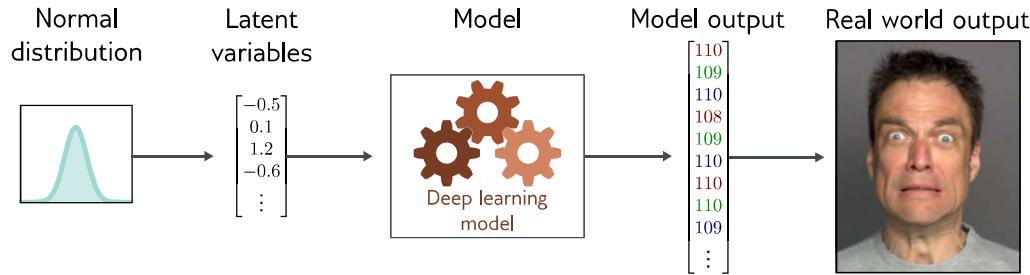


Figure 1.10 Latent variables. Many generative models use a deep learning model to describe the relationship between a low-dimensional “latent” variable and the observed high-dimensional data. The latent variables have a simple probability distribution by design. Hence, new examples can be generated by sampling from the simple distribution over the latent variables and then using the deep learning model to map the sample to the observed data space.



Figure 1.11 Image interpolation. In each row the left and right images are real and the three images in between represent a sequence of interpolations created by a generative model. The generative models that underpin these interpolations have learned that all images can be created by a set of underlying latent variables. By finding these variables for the two real images, interpolating their values, and then using these intermediate variables to create new images, we can generate intermediate results that are both visually plausible and mix the characteristics of the two original images. Top row adapted from Sauer et al. (2022). Bottom row adapted from Ramesh et al. (2022).



Figure 1.12 Multiple images generated from the caption “A teddy bear on a skateboard in Times Square.” Generated by DALL-E-2 (Ramesh et al., 2022).

probability distribution by design. By sampling from this distribution and passing the result through the deep learning model, we can create new samples (figure 1.10).

These models lead to new methods for manipulating real data. For example, consider finding the latent variables that underpin two real examples. We can interpolate between these examples by interpolating between their latent representations and mapping the intermediate positions back into the data space (figure 1.11).

1.2.3 Connecting supervised and unsupervised learning

Generative models with latent variables can also benefit supervised learning models where the outputs have structure (figure 1.4). For example, consider learning to predict the images corresponding to a caption. Rather than directly map the text input to an image, we can learn a relation between latent variables that explain the text and the latent variables that explain the image.

This has three advantages. First, we may need fewer text/image pairs to learn this mapping now that the inputs and outputs are lower dimensional. Second, we are more likely to generate a plausible-looking image; any sensible values of the latent variables should produce something that looks like a plausible example. Third, if we introduce randomness to either the mapping between the two sets of latent variables or the mapping from the latent variables to the image, then we can generate multiple images that are all described well by the caption (figure 1.12).

1.3 Reinforcement learning

The final area of machine learning is reinforcement learning. This paradigm introduces the idea of an agent which lives in a world and can perform certain actions at each time step. The actions change the state of the system but not necessarily in a deterministic way. Taking an action can also produce rewards, and the goal of reinforcement learning

is for the agent to learn to choose actions that lead to high rewards on average.

One complication is that the reward may occur some time after the action is taken, so associating a reward with an action is not straightforward. This is known as the *temporal credit assignment problem*. As the agent learns, it must trade off *exploration* and *exploitation* of what it already knows; perhaps the agent has already learned how to receive modest rewards; should it follow this strategy (exploit what it knows), or should it try different actions to see if it can improve (explore other opportunities)?

1.3.1 Two examples

Consider teaching a humanoid robot to locomote. The robot can perform a limited number of actions at a given time (moving various joints), and these change the state of the world (its pose). We might reward the robot for reaching checkpoints in an obstacle course. To reach each checkpoint, it must perform many actions, and it's unclear which ones contributed to the reward when it is received and which were irrelevant. This is an example of the temporal credit assignment problem.

A second example is learning to play chess. Again, the agent has a set of valid actions (chess moves) at any given time. However, these actions change the state of the system in a non-deterministic way; for any choice of action, the opposing player might respond with many different moves. Here, we might set up a reward structure based on capturing pieces or just have a single reward at the end of the game for winning. In the latter case, the temporal credit assignment problem is extreme; the system must learn which of the many moves it made were instrumental to success or failure.

The exploration-exploitation trade-off is also apparent in these two examples. The robot may have discovered that it can make progress by lying on its side and pushing with one leg. This strategy will move the robot and yields rewards, but much more slowly than the optimal solution: to balance on its legs and walk. So, it faces a choice between exploiting what it already knows (how to slide along the floor awkwardly) and exploring the space of actions (which might result in much faster locomotion). Similarly, in the chess example, the agent may learn a reasonable sequence of opening moves. Should it exploit this knowledge or explore different opening sequences?

It is perhaps not obvious how deep learning fits into the reinforcement learning framework. There are several possible approaches, but one technique is to use deep networks to build a mapping from the observed world state to an action. This is known as a *policy network*. In the robot example, the policy network would learn a mapping from its sensor measurements to joint movements. In the chess example, the network would learn a mapping from the current state of the board to the choice of move (figure 1.13).

1.4 Ethics

It would be irresponsible to write this book without discussing the ethical implications of artificial intelligence. This potent technology will change the world to at least the

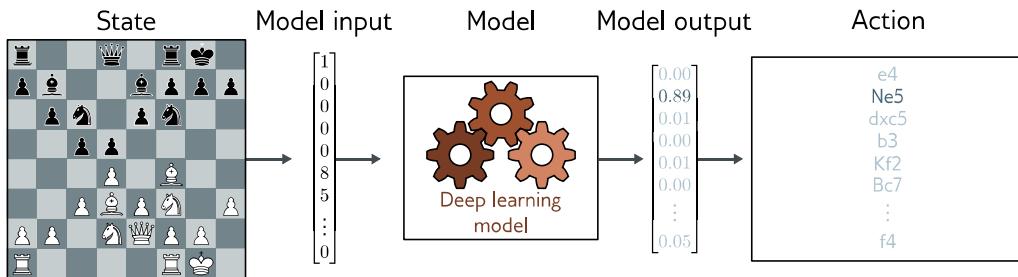


Figure 1.13 Policy networks for reinforcement learning. One way to incorporate deep neural networks into reinforcement learning is to use them to define a mapping from the state (here position on chessboard) to the actions (possible moves). This mapping is known as a *policy*.

same extent as electricity, the internal combustion engine, the transistor, or the internet. The potential benefits in healthcare, design, entertainment, transport, education, and almost every area of commerce are enormous. However, scientists and engineers are often unrealistically optimistic about the outcomes of their work, and the potential for harm is just as great. The following paragraphs highlight five concerns.

Bias and fairness: If we train a system to predict salary levels for individuals based on historical data, then this system will reproduce historical biases; for example, it will probably predict that women should be paid less than men. Several such cases have already become international news stories: an AI system for super-resolving face images made non-white people look more white; a system for generating images produced only pictures of men when asked to synthesize pictures of lawyers. Careless application of algorithmic decision-making using AI has the potential to entrench or aggravate existing biases. See Binns (2018) for further discussion.

Explainability: Deep learning systems make decisions, but we do not usually know exactly how or based on what information. They may contain billions of parameters, and there is no way we can understand how they work based on examination. This has led to the sub-field of explainable AI. One moderately successful area is producing local explanations; we cannot explain the entire system, but we can produce an interpretable description of why a particular decision was made. However, it remains unknown whether it is possible to build complex decision-making systems that are fully transparent to their users or even their creators. See Grennan et al. (2022) for further information.

Weaponizing AI: All significant technologies have been applied directly or indirectly toward war. Sadly, violent conflict seems to be an inevitable feature of human behavior. AI is arguably the most powerful technology ever built and will doubtless be deployed extensively in a military context. Indeed, this is already happening (Heikkilä, 2022).

Concentrating power: It is not from a benevolent interest in improving the lot of the human race that the world's most powerful companies are investing heavily in artificial intelligence. They know that these technologies will allow them to reap enormous profits. Like any advanced technology, deep learning is likely to concentrate power in the hands of the few organizations that control it. Automating jobs that are currently done by humans will change the economic environment and disproportionately affect the livelihoods of lower-paid workers with fewer skills. Optimists argue similar disruptions happened during the industrial revolution and resulted in shorter working hours. The truth is that we simply do not know what effects the large-scale adoption of AI will have on society (see David, 2015).

Existential risk: The major existential risks to the human race all result from technology. Climate change has been driven by industrialization. Nuclear weapons derive from the study of physics. Pandemics are more probable and spread faster because innovations in transport, agriculture, and construction have allowed a larger, denser, and more interconnected population. Artificial intelligence brings new existential risks. We should be very cautious about building systems that are more capable and extensible than human beings. In the most optimistic case, it will put vast power in the hands of the owners. In the most pessimistic case, we will be unable to control it or even understand its motives (see Tegmark, 2018).

This list is far from exhaustive. AI could also enable surveillance, disinformation, violations of privacy, fraud, and manipulation of financial markets, and the energy required to train AI systems contributes to climate change. Moreover, these concerns are not speculative; there are already many examples of ethically dubious applications of AI (consult Dao, 2021, for a partial list). In addition, the recent history of the internet has shown how new technology can cause harm in unexpected ways. The online community of the eighties and early nineties could hardly have predicted the proliferation of fake news, spam, online harassment, fraud, cyberbullying, incel culture, political manipulation, doxxing, online radicalization, and revenge porn.

Everyone studying or researching (or writing books about) AI should contemplate to what degree scientists are accountable for the uses of their technology. We should consider that capitalism primarily drives the development of AI and that legal advances and deployment for social good are likely to lag significantly behind. We should reflect on whether it's possible, as scientists and engineers, to control progress in this field and to reduce the potential for harm. We should consider what kind of organizations we are prepared to work for. How serious are they in their commitment to reducing the potential harms of AI? Are they simply "ethics-washing" to reduce reputational risk, or do they actually implement mechanisms to halt ethically suspect projects?

All readers are encouraged to investigate these issues further. The online course at <https://ethics-of-ai.mooc.fi/> is a useful introductory resource. If you are a professor teaching from this book, you are encouraged to raise these issues with your students. If you are a student taking a course where this is not done, then lobby your professor to make this happen. If you are deploying or researching AI in a corporate environment, you are encouraged to scrutinize your employer's values and to help change them (or leave) if they are wanting.

1.5 Structure of book

The structure of the book follows the structure of this introduction. Chapters 2–9 walk through the supervised learning pipeline. We describe shallow and deep neural networks and discuss how to train them and measure and improve their performance. Chapters 10–13 describe common architectural variations of deep neural networks, including convolutional networks, residual connections, and transformers. These architectures are used across supervised, unsupervised, and reinforcement learning.

Chapters 14–18 tackle unsupervised learning using deep neural networks. We devote a chapter each to four modern deep generative models: generative adversarial networks, variational autoencoders, normalizing flows, and diffusion models. Chapter 19 is a brief introduction to deep reinforcement learning. This is a topic that easily justifies its own book, so the treatment is necessarily superficial. However, this treatment is intended to be a good starting point for readers unfamiliar with this area.

Despite the title of this book, some aspects of deep learning remain poorly understood. Chapter 20 poses some fundamental questions. Why are deep networks so easy to train? Why do they generalize so well? Why do they need so many parameters? Do they need to be deep? Along the way, we explore unexpected phenomena such as the structure of the loss function, double descent, grokking, and lottery tickets. The book concludes with chapter 21, which discusses ethics and deep learning.

1.6 Other books

This book is self-contained but is limited to coverage of deep learning. It is intended to be the spiritual successor to *Deep Learning* (Goodfellow et al., 2016) which is a fantastic resource but does not cover recent advances. For a broader look at machine learning, the most up-to-date and encyclopedic resource is *Probabilistic Machine Learning* (Murphy, 2022, 2023). However, *Pattern Recognition and Machine Learning* (Bishop, 2006) is still an excellent and relevant book.

If you enjoy this book, then my previous volume, *Computer Vision: Models, Learning, and Inference* (Prince, 2012), is still worth reading. Some parts have dated badly, but it contains a thorough introduction to probability, including Bayesian methods, and good introductory coverage of latent variable models, geometry for computer vision, Gaussian processes, and graphical models. It uses identical notation to this book and can be found online. A detailed treatment of graphical models can be found in *Probabilistic Graphical Models: Principles and Techniques* (Koller & Friedman, 2009), and Gaussian processes are covered by *Gaussian Processes for Machine Learning* (Williams & Rasmussen, 2006).

For background mathematics, consult *Mathematics for Machine Learning* (Deisenroth et al., 2020). For a more coding-oriented approach, consult *Dive into Deep Learning* (Zhang et al., 2023). The best overview for computer vision is Szeliski (2022), and there is also the impending book *Foundations of Computer Vision* (Torralba et al., 2024). A good starting point to learn about graph neural networks is *Graph Representation Learning* (Hamilton, 2020). The definitive work on reinforcement learning is *Reinforce-*

ment Learning: An Introduction (Sutton & Barto, 2018). A good initial resource is Foundations of Deep Reinforcement Learning (Graesser & Keng, 2019).

1.7 How to read this book

Most remaining chapters in this book contain a main body of text, a notes section, and a set of problems. The main body of the text is intended to be self-contained and can be read without recourse to the other parts of the chapter. As much as possible, background mathematics is incorporated into the main body of the text. However, for larger topics that would be a distraction to the main thread of the argument, the background material is appendicized, and a reference is provided in the margin. Most [notation](#) in this book is standard. However, some conventions are less widely used, and the reader is encouraged to consult appendix A before proceeding.

The main body of text includes many novel illustrations and visualizations of deep learning models and results. I've worked hard to provide new explanations of existing ideas rather than merely curate the work of others. Deep learning is a new field, and sometimes phenomena are poorly understood. I try to make it clear where this is the case and when my explanations should be treated with caution.

References are included in the main body of the chapter only where results are depicted. Instead, they can be found in the notes section at the end of the chapter. I do not generally respect historical precedent in the main text; if an ancestor of a current technique is no longer useful, then I will not mention it. However, the historical development of the field is described in the notes section, and hopefully, credit is fairly assigned. The notes are organized into paragraphs and provide pointers for further reading. They should help the reader orient themselves within the sub-area and understand how it relates to other parts of machine learning. The notes are less self-contained than the main text. Depending on your level of background knowledge and interest, you may find these sections more or less useful.

Each chapter has a number of associated problems. They are referenced in the margin of the main text at the point that they should be attempted. As George Pólya noted, "Mathematics, you see, is not a spectator sport." He was correct, and I highly recommend that you attempt the problems as you go. In some cases, they provide insights that will help you understand the main text. Problems for which the answers are provided on the associated website are indicated with an asterisk. Additionally, Python notebooks that will help you understand the ideas in this book are also available via the website, and these are also referenced in the margins of the text. Indeed, if you are feeling rusty, it might be worth working through the notebook on [background mathematics](#) right now.

Unfortunately, the pace of research in AI makes it inevitable that this book will be a constant work in progress. If there are parts you find hard to understand, notable omissions, or sections that seem extraneous, please get in touch via the associated website. Together, we can make the next edition better.