

What this book is about

Neural networks are one of the most beautiful programming paradigms ever invented. In the conventional approach to programming, we tell the computer what to do, breaking big problems up into many small, precisely defined tasks that the computer can easily perform. By contrast, in a neural network we don't tell the computer how to solve our problem. Instead, it learns from observational data, figuring out its own solution to the problem at hand.

Automatically learning from data sounds promising. However, until 2006 we didn't know how to train neural networks to surpass more traditional approaches, except for a few specialized problems. What changed in 2006 was the discovery of techniques for learning in so-called deep neural networks. These techniques are now known as deep learning. They've been developed further, and today deep neural networks and deep learning achieve outstanding performance on many important problems in computer vision, speech recognition, and natural language processing. They're being deployed on a large scale by companies such as Google, Microsoft, and Facebook.

The purpose of this book is to help you master the core concepts of neural networks, including modern techniques for deep learning. After working through the book you will have written code that uses neural networks and deep learning to solve complex pattern recognition problems. And you will have a foundation to use neural networks and deep learning to attack problems of your own devising.

A principle-oriented approach

One conviction underlying the book is that it's better to obtain a solid understanding of the core principles of neural networks and deep learning, rather than a hazy understanding of a long laundry list of ideas. If you've understood the core ideas well, you can rapidly understand other new material. In programming language terms, think of it as mastering the core syntax, libraries and data structures of a new language. You may still only "know" a tiny

Neural Networks and Deep Learning

What this book is about

On the exercises and problems

- ▶ [Using neural nets to recognize handwritten digits](#)
- ▶ [How the backpropagation algorithm works](#)
- ▶ [Improving the way neural networks learn](#)
- ▶ [A visual proof that neural nets can compute any function](#)
- ▶ [Why are deep neural networks hard to train?](#)
- ▶ [Deep learning](#)
- ▶ [Appendix: Is there a simple algorithm for intelligence?](#)
- ▶ [Acknowledgements](#)
- ▶ [Frequently Asked Questions](#)

If you benefit from the book, please make a small donation. I suggest \$5, but you can choose the amount.

Donate



Alternately, you can make a donation by sending me Bitcoin, at address

1Kd6tXH5SDAmiFb49J9hknG5pqj7KStSAX

Sponsors



Deep Learning Workstations starting at \$6,999: [learn more](#)



Deep Learning Workstations, Servers, and Laptops



Thanks to all the [supporters](#) who made the book possible, with especial thanks to Pavel Dudrenov.

fraction of the total language - many languages have enormous standard libraries - but new libraries and data structures can be understood quickly and easily.

This means the book is emphatically not a tutorial in how to use some particular neural network library. If you mostly want to learn your way around a library, don't read this book! Find the library you wish to learn, and work through the tutorials and documentation. But be warned. While this has an immediate problem-solving payoff, if you want to understand what's really going on in neural networks, if you want insights that will still be relevant years from now, then it's not enough just to learn some hot library. You need to understand the durable, lasting insights underlying how neural networks work. Technologies come and technologies go, but insight is forever.

A hands-on approach

We'll learn the core principles behind neural networks and deep learning by attacking a concrete problem: the problem of teaching a computer to recognize handwritten digits. This problem is extremely difficult to solve using the conventional approach to programming. And yet, as we'll see, it can be solved pretty well using a simple neural network, with just a few tens of lines of code, and no special libraries. What's more, we'll improve the program through many iterations, gradually incorporating more and more of the core ideas about neural networks and deep learning.

This hands-on approach means that you'll need some programming experience to read the book. But you don't need to be a professional programmer. I've written the code in Python (version 2.7), which, even if you don't program in Python, should be easy to understand with just a little effort. Through the course of the book we will develop a little neural network library, which you can use to experiment and to build understanding. All the code is available for download [here](#). Once you've finished the book, or as you read it, you can easily pick up one of the more feature-complete neural network libraries intended for use in production.

On a related note, the mathematical requirements to read the book are modest. There is some mathematics in most chapters, but it's

Thanks also to all the contributors to the [Bugfinder Hall of Fame](#).

Resources

[Michael Nielsen on Twitter](#)

[Book FAQ](#)

[Code repository](#)

[Michael Nielsen's project announcement mailing list](#)

[Deep Learning](#), book by Ian Goodfellow, Yoshua Bengio, and Aaron Courville

[cognitivemedium.com](#)



By [Michael Nielsen](#) / Jun 2019

usually just elementary algebra and plots of functions, which I expect most readers will be okay with. I occasionally use more advanced mathematics, but have structured the material so you can follow even if some mathematical details elude you. The one chapter which uses heavier mathematics extensively is [Chapter 2](#), which requires a little multivariable calculus and linear algebra. If those aren't familiar, I begin [Chapter 2](#) with a discussion of how to navigate the mathematics. If you're finding it really heavy going, you can simply skip to the [summary](#) of the chapter's main results. In any case, there's no need to worry about this at the outset.

It's rare for a book to aim to be both principle-oriented and hands-on. But I believe you'll learn best if we build out the fundamental ideas of neural networks. We'll develop living code, not just abstract theory, code which you can explore and extend. This way you'll understand the fundamentals, both in theory and practice, and be well set to add further to your knowledge.

In academic work, please cite this book as: Michael A. Nielsen, "Neural Networks and Deep Learning", Determination Press, 2015

Last update: Tue Jun 11 16:58:53 2019

This work is licensed under a Creative Commons Attribution-NonCommercial 3.0 Unported License. This means you're free to copy, share, and build on this book, but not to sell it. If you're interested in commercial use, please [contact me](#).

