


# Week 2

# Resources for larning Deep Learning

<https://d2l.ai/index.html>

 DIVE INTO  
DEEP LEARNING

Preface

Installation

Notation

1. Introduction

2. Preliminaries

3. Linear Neural Networks  
for Regression

4. Linear Neural Networks  
for Classification

5. Multilayer Perceptrons

6. Builders' Guide

7. Convolutional Neural  
Networks

8. Modern Convolutional  
Neural Networks

9. Recurrent Neural  
Networks

10. Modern Recurrent  
Neural Networks

11. Attention Mechanisms  
and Transformers

12. Optimization  
Algorithms

13. Computational  
Performance

Dive into Deep Learning

Preview Version

PyTorch

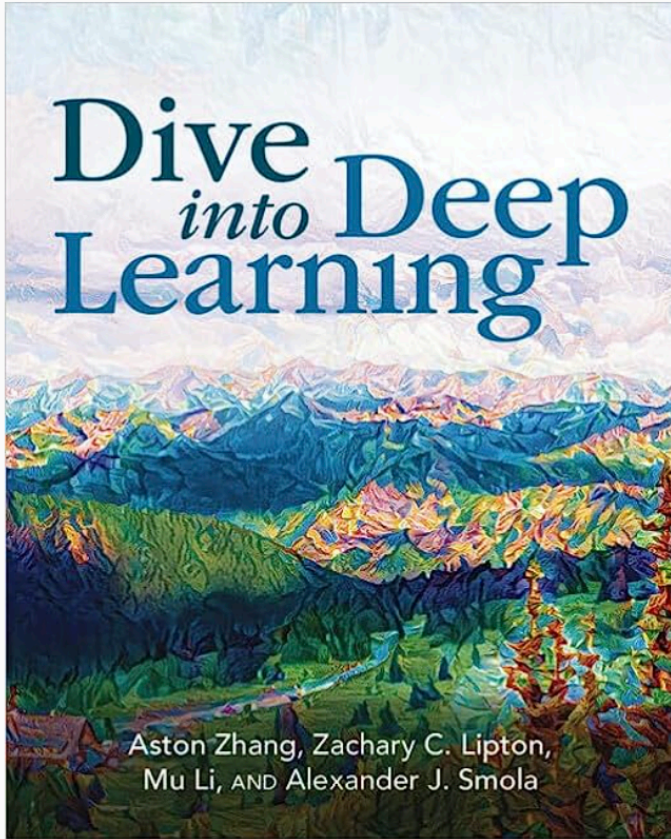
MXNet

Notebooks

Courses

GitHub

中文版



Aston Zhang, Zachary C. Lipton,  
Mu Li, AND Alexander J. Smola

Dive into Deep Learning

Interactive deep learning book with code, math, and  
discussions

Implemented with PyTorch, NumPy/MXNet, JAX, and  
TensorFlow

Adopted at 500 universities from 70 countries

☆ Star

21,597

Follow @D2L\_ai

- [Feb 2023] The book is forthcoming on Cambridge University Press ([order](#)). The Chinese version is the [best seller](#) at the largest Chinese online bookstore. Follow D2L's [open-source project](#) for the latest updates.
- [Dec 2022] JAX implementation is available! New topics of [reinforcement learning](#), [Gaussian processes](#), and [hyperparameter optimization](#) are added!

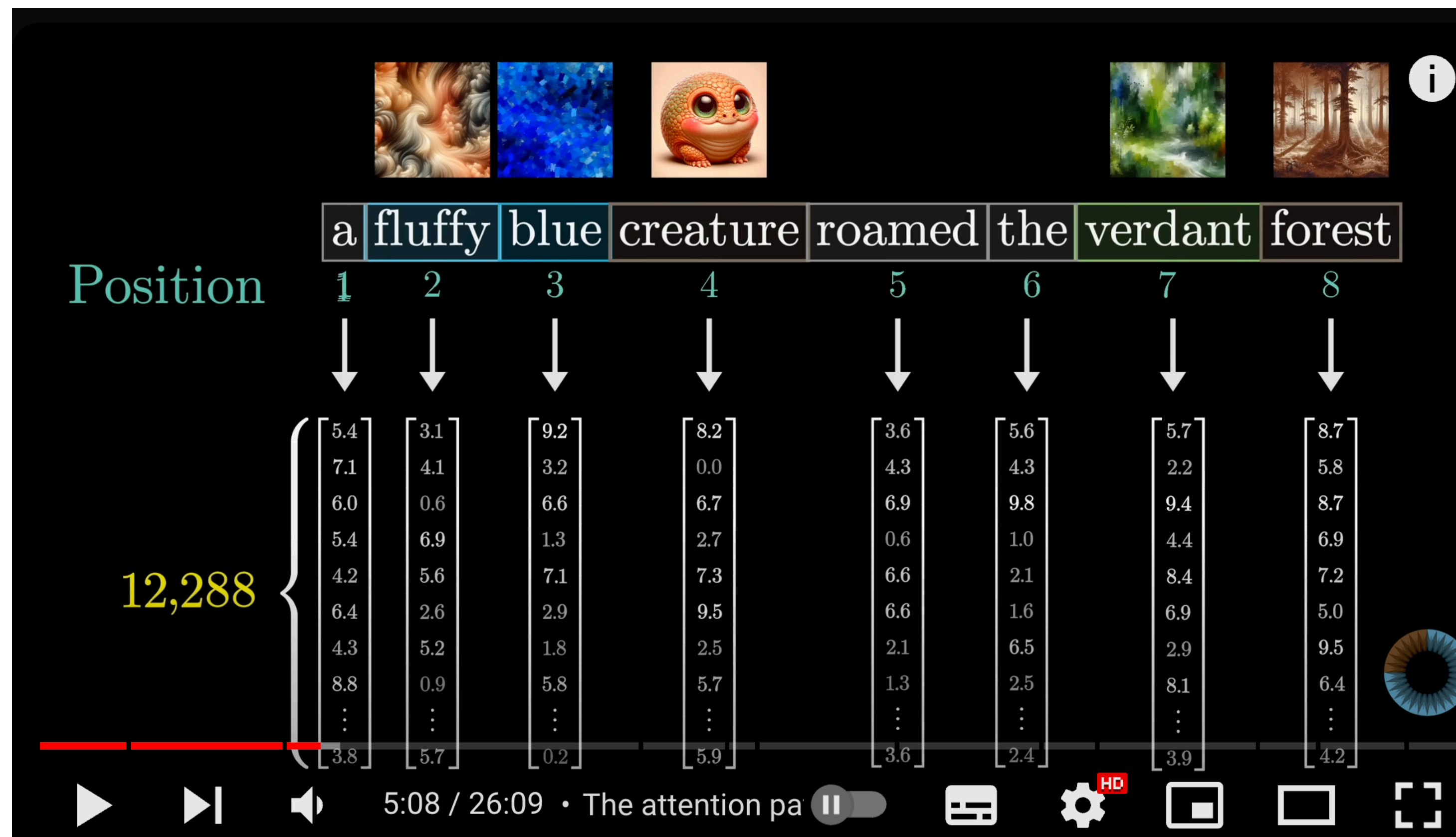
<https://www.amazon.com/Dive-into-Learning-Aston-Zhang/dp/1009389432/> check out our [new API](#) for implementation and new topics like generalization in [classification](#) and [deep learning](#), [ResNeXt](#), [CNN design space](#), and transformers for [vision](#)

<https://www.deeplearningbook.org/>

[Errata in published editions](#)

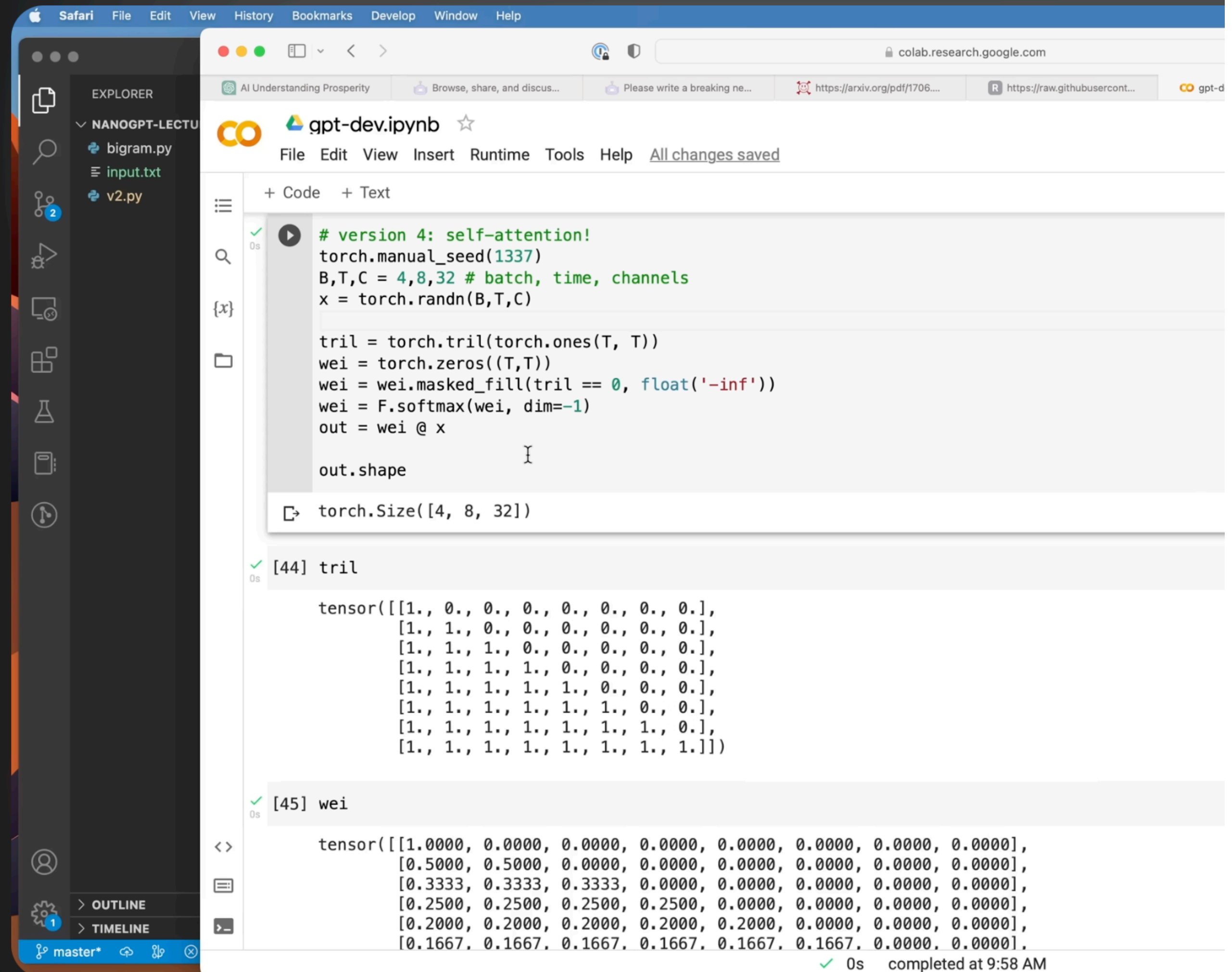
## Deep Learning

- [Table of Contents](#)
- [Acknowledgements](#)
- [Notation](#)
- [1 Introduction](#)
- [Part I: Applied Math and Machine Learning Basics](#)
  - [2 Linear Algebra](#)
  - [3 Probability and Information Theory](#)
  - [4 Numerical Computation](#)
  - [5 Machine Learning Basics](#)
- [Part II: Modern Practical Deep Networks](#)
  - [6 Deep Feedforward Networks](#)
  - [7 Regularization for Deep Learning](#)
  - [8 Optimization for Training Deep Models](#)
  - [9 Convolutional Networks](#)
  - [10 Sequence Modeling: Recurrent and Recursive Nets](#)
  - [11 Practical Methodology](#)
  - [12 Applications](#)
- [Part III: Deep Learning Research](#)
  - [13 Linear Factor Models](#)
  - [14 Autoencoders](#)
  - [15 Representation Learning](#)
  - [16 Structured Probabilistic Models for Deep Learning](#)
  - [17 Monte Carlo Methods](#)
  - [18 Confronting the Partition Function](#)
  - [19 Approximate Inference](#)
  - [20 Deep Generative Models](#)
- [Bibliography](#)
- [Index](#)



<https://www.youtube.com/@3blue1brown>





The screenshot displays a Jupyter Notebook titled "gpt-dev.ipynb" in a web browser. The notebook is running on a Google Colab instance. The code in the notebook is as follows:

```
# version 4: self-attention!
torch.manual_seed(1337)
B,T,C = 4,8,32 # batch, time, channels
x = torch.randn(B,T,C)

tril = torch.tril(torch.ones(T, T))
wei = torch.zeros((T,T))
wei = wei.masked_fill(tril == 0, float('-inf'))
wei = F.softmax(wei, dim=-1)
out = wei @ x

out.shape
```

The output of the code is shown in the console:

```
torch.Size([4, 8, 32])
```

```
[44] tril
tensor([[1., 0., 0., 0., 0., 0., 0., 0.],
        [1., 1., 0., 0., 0., 0., 0., 0.],
        [1., 1., 1., 0., 0., 0., 0., 0.],
        [1., 1., 1., 1., 0., 0., 0., 0.],
        [1., 1., 1., 1., 1., 0., 0., 0.],
        [1., 1., 1., 1., 1., 1., 0., 0.],
        [1., 1., 1., 1., 1., 1., 1., 0.],
        [1., 1., 1., 1., 1., 1., 1., 1.]])
```

```
[45] wei
tensor([[1.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000],
        [0.5000, 0.5000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000],
        [0.3333, 0.3333, 0.3333, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000],
        [0.2500, 0.2500, 0.2500, 0.2500, 0.0000, 0.0000, 0.0000, 0.0000],
        [0.2000, 0.2000, 0.2000, 0.2000, 0.2000, 0.0000, 0.0000, 0.0000],
        [0.1667, 0.1667, 0.1667, 0.1667, 0.1667, 0.1667, 0.0000, 0.0000],
        [0.1667, 0.1667, 0.1667, 0.1667, 0.1667, 0.1667, 0.0000, 0.0000],
        [0.1667, 0.1667, 0.1667, 0.1667, 0.1667, 0.1667, 0.0000, 0.0000]])
```

The notebook interface also shows a file explorer on the left with files "bigram.py", "input.txt", and "v2.py". The bottom status bar indicates the notebook is on the "master" branch and was completed at 9:58 AM.

Let's build GPT: from scratch, in code, spelled out.

<https://www.youtube.com/@AndrejKarpathy>