

O'REILLY®

# AI and Machine Learning for Coders

A Programmer's Guide to Artificial Intelligence



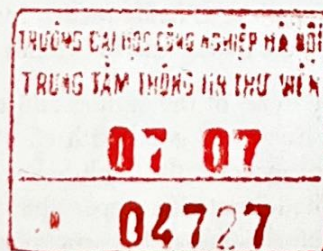
Laurence Moroney  
Foreword by Andrew Ng



---

# AI and Machine Learning for Coders

*A Programmer's Guide to Artificial Intelligence*



*Laurence Moroney*

Beijing • Boston • Farnham • Sebastopol • Tokyo

**O'REILLY®**



---

# Table of Contents

Foreword.....	xiii
---------------	------

Preface.....	xv
--------------	----

---

## Part I. Building Models

<b>1. Introduction to TensorFlow.....</b>	<b>1</b>
What Is Machine Learning?	1
Limitations of Traditional Programming	3
From Programming to Learning	5
What Is TensorFlow?	7
Using TensorFlow	8
Installing TensorFlow in Python	9
Using TensorFlow in PyCharm	9
Using TensorFlow in Google Colab	12
Getting Started with Machine Learning	13
Seeing What the Network Learned	18
Summary	19
<b>2. Introduction to Computer Vision.....</b>	<b>21</b>
Recognizing Clothing Items	21
The Data: Fashion MNIST	22
Neurons for Vision	23
Designing the Neural Network	25
The Complete Code	26
Training the Neural Network	29



Exploring the Model Output	29
Training for Longer—Discovering Overfitting	30
Stopping Training	30
Summary	32
<b>3. Going Beyond the Basics: Detecting Features in Images.....</b>	<b>33</b>
Convolutions	34
Pooling	35
Implementing Convolutional Neural Networks	37
Exploring the Convolutional Network	39
Building a CNN to Distinguish Between Horses and Humans	42
The Horses or Humans Dataset	42
The Keras ImageDataGenerator	43
CNN Architecture for Horses or Humans	44
Adding Validation to the Horses or Humans Dataset	47
Testing Horse or Human Images	49
Image Augmentation	52
Transfer Learning	55
Multiclass Classification	59
Dropout Regularization	63
Summary	65
<b>4. Using Public Datasets with TensorFlow Datasets.....</b>	<b>67</b>
Getting Started with TFDS	68
Using TFDS with Keras Models	70
Loading Specific Versions	73
Using Mapping Functions for Augmentation	73
Using TensorFlow Addons	74
Using Custom Splits	74
Understanding TFRecord	76
The ETL Process for Managing Data in TensorFlow	78
Optimizing the Load Phase	80
Parallelizing ETL to Improve Training Performance	81
Summary	83
<b>5. Introduction to Natural Language Processing.....</b>	<b>85</b>
Encoding Language into Numbers	85
Getting Started with Tokenization	86
Turning Sentences into Sequences	87
Removing Stopwords and Cleaning Text	91
Working with Real Data Sources	93



Getting Text from TensorFlow Datasets	93
Getting Text from CSV Files	97
Getting Text from JSON Files	99
Summary	102
<b>6. Making Sentiment Programmable Using Embeddings.....</b>	<b>103</b>
Establishing Meaning from Words	103
A Simple Example: Positives and Negatives	104
Going a Little Deeper: Vectors	105
Embeddings in TensorFlow	106
Building a Sarcasm Detector Using Embeddings	106
Reducing Overfitting in Language Models	109
Using the Model to Classify a Sentence	119
Visualizing the Embeddings	120
Using Pretrained Embeddings from TensorFlow Hub	123
Summary	125
<b>7. Recurrent Neural Networks for Natural Language Processing.....</b>	<b>127</b>
The Basis of Recurrence	127
Extending Recurrence for Language	130
Creating a Text Classifier with RNNs	132
Stacking LSTMs	134
Using Pretrained Embeddings with RNNs	139
Summary	146
<b>8. Using TensorFlow to Create Text.....</b>	<b>147</b>
Turning Sequences into Input Sequences	148
Creating the Model	152
Generating Text	154
Predicting the Next Word	154
Compounding Predictions to Generate Text	155
Extending the Dataset	156
Changing the Model Architecture	157
Improving the Data	158
Character-Based Encoding	162
Summary	163
<b>9. Understanding Sequence and Time Series Data.....</b>	<b>165</b>
Common Attributes of Time Series	167
Trend	167
Seasonality	167



Autocorrelation	168
Noise	168
Techniques for Predicting Time Series	169
Naive Prediction to Create a Baseline	169
Measuring Prediction Accuracy	172
Less Naive: Using Moving Average for Prediction	172
Improving the Moving Average Analysis	173
Summary	174
<b>10. Creating ML Models to Predict Sequences.....</b>	<b>175</b>
Creating a Windowed Dataset	176
Creating a Windowed Version of the Time Series Dataset	178
Creating and Training a DNN to Fit the Sequence Data	180
Evaluating the Results of the DNN	181
Exploring the Overall Prediction	183
Tuning the Learning Rate	185
Exploring Hyperparameter Tuning with Keras Tuner	187
Summary	191
<b>11. Using Convolutional and Recurrent Methods for Sequence Models.....</b>	<b>193</b>
Convolutions for Sequence Data	193
Coding Convolutions	194
Experimenting with the Conv1D Hyperparameters	197
Using NASA Weather Data	200
Reading GISS Data in Python	201
Using RNNs for Sequence Modeling	202
Exploring a Larger Dataset	205
Using Other Recurrent Methods	207
Using Dropout	208
Using Bidirectional RNNs	211
Summary	213

---

## Part II. Using Models

<b>12. An Introduction to TensorFlow Lite.....</b>	<b>217</b>
What Is TensorFlow Lite?	217
Walkthrough: Creating and Converting a Model to TensorFlow Lite	219
Step 1. Save the Model	220
Step 2. Convert and Save the Model	221
Step 3. Load the TFLite Model and Allocate Tensors	221



Step 4. Perform the Prediction	222
Walkthrough: Transfer Learning an Image Classifier and Converting to TensorFlow Lite	224
Step 1. Build and Save the Model	224
Step 2. Convert the Model to TensorFlow Lite	225
Step 3. Optimize the Model	227
Summary	229
<b>13. Using TensorFlow Lite in Android Apps.....</b>	<b>231</b>
What Is Android Studio?	231
Creating Your First TensorFlow Lite Android App	232
Step 1. Create a New Android Project	232
Step 2. Edit Your Layout File	234
Step 3. Add the TensorFlow Lite Dependencies	236
Step 4. Add Your TensorFlow Lite Model	238
Step 5. Write the Activity Code to Use TensorFlow Lite for Inference	238
Moving Beyond “Hello World”—Processing Images	241
TensorFlow Lite Sample Apps	244
Summary	246
<b>14. Using TensorFlow Lite in iOS Apps.....</b>	<b>247</b>
Creating Your First TensorFlow Lite App with Xcode	247
Step 1. Create a Basic iOS App	247
Step 2. Add TensorFlow Lite to Your Project	249
Step 3. Create the User Interface	250
Step 4. Add and Initialize the Model Inference Class	252
Step 5. Perform the Inference	255
Step 6. Add the Model to Your App	256
Step 7. Add the UI Logic	258
Moving Beyond “Hello World”—Processing Images	260
TensorFlow Lite Sample Apps	263
Summary	263
<b>15. An Introduction to TensorFlow.js.....</b>	<b>265</b>
What Is TensorFlow.js?	265
Installing and Using the Brackets IDE	267
Building Your First TensorFlow.js Model	268
Creating an Iris Classifier	272
Summary	276



<b>16. Coding Techniques for Computer Vision in TensorFlow.js.....</b>	<b>277</b>
JavaScript Considerations for TensorFlow Developers	278
Building a CNN in JavaScript	279
Using Callbacks for Visualization	281
Training with the MNIST Dataset	284
Running Inference on Images in TensorFlow.js	290
Summary	290
<b>17. Reusing and Converting Python Models to JavaScript.....</b>	<b>293</b>
Converting Python-Based Models to JavaScript	293
Using the Converted Models	295
Using Preconverted JavaScript Models	297
Using the Toxicity Text Classifier	297
Using MobileNet for Image Classification in the Browser	300
Using PoseNet	303
Summary	306
<b>18. Transfer Learning in JavaScript.....</b>	<b>307</b>
Transfer Learning from MobileNet	308
Step 1. Download MobileNet and Identify the Layers to Use	308
Step 2. Create Your Own Model Architecture with the Outputs from MobileNet as Its Input	310
Step 3. Gather and Format the Data	312
Step 4. Train the Model	318
Step 5. Run Inference with the Model	319
Transfer Learning from TensorFlow Hub	321
Using Models from TensorFlow.org	324
Summary	326
<b>19. Deployment with TensorFlow Serving.....</b>	<b>329</b>
What Is TensorFlow Serving?	329
Installing TensorFlow Serving	332
Installing Using Docker	332
Installing Directly on Linux	333
Building and Serving a Model	334
Exploring Server Configuration	337
Summary	340
<b>20. AI Ethics, Fairness, and Privacy.....</b>	<b>341</b>
Fairness in Programming	342
Fairness in Machine Learning	345



Tools for Fairness	347
The What-If Tool	347
Facets	348
Federated Learning	351
Step 1. Identify Available Devices for Training	351
Step 2. Identify Suitable Available Devices for Training	352
Step 3. Deploy a Trainable Model to Your Training Set	352
Step 4. Return the Results of the Training to the Server	353
Step 5. Deploy the New Master Model to the Clients	353
Secure Aggregation with Federated Learning	354
Federated Learning with TensorFlow Federated	355
Google's AI Principles	356
Summary	357
<b>Index.....</b>	<b>359</b>



---

# Preface

Welcome to *AI and Machine Learning for Coders*, a book that I've been wanting to write for many years but that has only really become possible due to recent advances in machine learning (ML) and, in particular, TensorFlow. The goal of this book is to prepare you, as a coder, for many of the scenarios that you can address with machine learning, with the aim of equipping you to be an ML and AI developer *without* needing a PhD! I hope that you'll find it useful, and that it will empower you with the confidence to get started on this wonderful and rewarding journey.

## Who Should Read This Book

If you're interested in AI and ML, and you want to get up and running quickly with building models that learn from data, this book is for you. If you're interested in getting started with common AI and ML concepts—computer vision, natural language processing, sequence modeling, and more—and want to see how neural networks can be trained to solve problems in these spaces, I think you'll enjoy this book. And if you have models that you've trained and want to get them into the hands of users on mobile, in the browser, or via the cloud, then this book is also for you!

Most of all, if you've been put off entering this valuable area of computer science because of perceived difficulty, in particular believing that you'll need to dust off your old calculus books, then fear not: this book takes a code-first approach that shows you just how easy it is to get started in the world of machine learning and artificial intelligence using Python and TensorFlow.

## Why I Wrote This Book

I first got seriously involved with artificial intelligence in the spring of 1992. A freshly minted physics graduate living in London in the midst of a terrible recession, I had been unemployed for six months. The British government started a program to train 20 people in AI technology, and put out a call for applicants. I was the first



participant selected. Three months later, the program failed miserably, because while there was plenty of *theoretical* work that could be done with AI, there was no easy way to do it *practically*. One could write simple inference in a language called Prolog, and perform list processing in a language called Lisp, but there was no clear path to deploying them in industry. The famous “AI winter” followed.

Then, in 2016, while I was working at Google on a product called Firebase, the company offered machine learning training to all engineers. I sat in a room with a number of other people and listened to lectures about calculus and gradient descent. I couldn’t quite match this to a practical implementation of ML, and I was suddenly transported back to 1992. I brought feedback about this, and about how we should educate people in ML, to the TensorFlow team—and they hired me in 2017. With the release of TensorFlow 2.0 in 2018, and in particular the emphasis on high-level APIs that made it easy for developers to get started, I realized the need was there for a book that took advantage of this, and widened access to ML so that it wasn’t just for mathematicians or PhDs anymore.

I believe that more people using this technology and deploying it to end users will lead to an explosion in AI and ML that will prevent another AI winter, and change the world very much for the better. I’m already seeing the impact of this, from the work done by Google on diabetic retinopathy, through Penn State University and PlantVillage building an ML model for mobile that helps farmers diagnose cassava disease, Médecins Sans Frontières using TensorFlow models to help diagnose antibiotic resistance, and much, much more!

## Navigating This Book

The book is written in two main parts. Part I (Chapters 1–11) talks about how to use TensorFlow to build machine learning models for a variety of scenarios. It takes you from first principles—building a model with a neural network containing only one neuron—through computer vision, natural language processing, and sequence modeling. Part II (Chapters 12–20) then walks you through scenarios for putting your models in people’s hands on Android and iOS, in browsers with JavaScript, and serving via the cloud. Most chapters are standalone, so you can drop in and learn something new, or, of course, you could just read the book cover to cover.

## Technology You Need to Understand

The goal of the first half of the book is to help you learn how to use TensorFlow to build models with a variety of architectures. The only real prerequisite to this is understanding Python, and in particular Python notation for data and array processing. You might also want to explore Numpy, a Python library for numeric calculations. If you have no familiarity with these, they are quite easy to learn, and



you can probably pick up what you need as you go along (although some of the array notation might be a bit hard to grasp).

For the second half of the book, I generally will not teach the languages that are shown, but instead show how TensorFlow models can be used in them. So, for example, in the Android chapter (Chapter 13) you'll explore building apps in Kotlin with Android studio, and in the iOS chapter (Chapter 14) you'll explore building apps in Swift with Xcode. I won't be teaching the syntax of these languages, so if you aren't familiar with them, you may need a primer—*Learning Swift* (<https://oreil.ly/MnEVD>) by Jonathan Manning, Paris Buttfield-Addison, and Tim Nugent (O'Reilly) is a great choice.

## Online Resources

A variety of online resources are used by, and supported in, this book. At the very least I would recommend that you keep an eye on TensorFlow (<https://www.tensorflow.org>) and its associated YouTube channel (<https://www.youtube.com/tensorflow>) for any updates and breaking changes to technologies discussed in the book.

The code for this book is available at <https://github.com/lmoroney/tfbook>, and I will keep it up to date there as the platform evolves.

## Conventions Used in This Book

The following typographical conventions are used in this book:

### *Italic*

Indicates new terms, URLs, email addresses, filenames, and file extensions.

### Constant width

Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, data types, environment variables, statements, and keywords.

### Constant width bold

Used for emphasis in code snippets.



This element signifies a note.