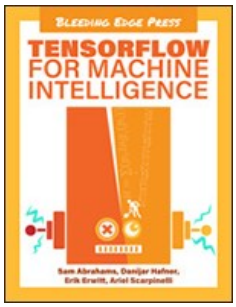

TensorFlow for Machine Intelligence: A Hands-On Introduction to Learning Algorithms

by Sam Abrahams, Danijar Hafner, Erik Erwitt and Ariel Scarpinelli

Preface

Welcome

Since its open source release in November 2015, **TensorFlow** has become one of the most exciting machine learning libraries available. It is being used more and more in research, production, and education. The library has seen continual improvements, additions, and optimizations, and the TensorFlow community has grown dramatically. With **TensorFlow for Machine Intelligence**, we hope to help new and experienced users hone their abilities with TensorFlow and become fluent in using this powerful library to its fullest!

Background education

While this book is primarily focused on the TensorFlow API, we expect you to have familiarity with a number of mathematical and programmatic concepts. These include:

- Derivative calculus (single-variable and multi-variables)
- Matrix algebra (matrix multiplication especially)
- Basic understanding of programming principles
- Basic machine learning concepts

In addition to the above, you will get more out of this book if they have the following knowledge:

- Experience with Python and organizing modules
- Experience with the **NumPy** library
- Experience with the **matplotlib** library
- Knowledge of more advanced machine learning concepts, especially feed-forward neural networks, convolutional neural networks, and recurrent nerual networks

When appropriate, we'll include refresher information to re-familiarize the reader with some of the concepts they need to know, to fully understand the math and/or Python concepts.

What you should expect to learn

This TensorFlow book introduces the framework and the underlying machine learning concepts that are important to harness machine intelligence.

After reading this book, you should know the following:

- A deep understanding of the core TensorFlow API
- The TensorFlow workflow: graph definition and graph execution
- How to install TensorFlow on various devices
- Best practices for structuring your code and project
- How to create core machine learning models in TensorFlow
- How to implement RNNs and CNNs in TensorFlow
- How to deploy code with TensorFlow Serving
- The fundamentals of utilizing TensorBoard to analyze your models

This book's layout

Section 1: Getting started with TensorFlow

The first section of this book helps get you on your feet and ready to use TensorFlow. The first chapter is the introduction, which provides a brief historical context for TensorFlow and includes a discussion about TensorFlow's design patterns as well as the merits and challenges of choosing TensorFlow as a deep learning library.

After the introduction, "TensorFlow Installation" goes over some considerations a person installing TensorFlow should think about, and includes detailed instructions on installing TensorFlow: both installing from binaries as well as building from source.

Section 2: TensorFlow and Machine Learning fundamentals

This second section begins in "Fundamentals of TensorFlow" by getting TensorFlow installed on your machine and diving deep into the fundamentals of the TensorFlow API *without* incorporating a lot of machine learning concepts. The goal is to isolate "learning TensorFlow" and "learning how to do machine learning with TensorFlow". This will include an in depth description of many of the most important pieces of the TensorFlow API. We'll also show how you can take a visual graph representation of a model and translate it into TensorFlow code, as well as verify that the graph is modeled correctly by using TensorBoard.

Once the core API concepts are covered, we continue with "Machine Learning Basics", in which we create simple machine learning models, such as linear regression, logistic regression, and clustering, inside of TensorFlow.

Section 3: Implementing advanced deep models in TensorFlow

The third section is comprised of two chapters, each focusing on a different type of more complex deep learning model. Each will describe the model in question and present visual graph representation of what we are trying to create. We'll discuss why the model is setup the way it is, note any mathematical quirks, and then go over how to set them up effectively in TensorFlow.

The first model we'll look at is the Convolutional Neural Network, or CNN, in "Object Recognition and Classification," where we'll talk about training TensorFlow models on image data. This will include a discussion on the math and purpose of convolutions, how to convert raw image data into a TensorFlow-compatible format, and how to test your final output.

In the chapter "Natual Language Processing with Reurrent Networks," we'll examine how to properly create Recurrent Neural Networks, or RNNs, in TensorFlow. Looking at a variety of natural language processing (NLP) tasks, we'll see how to use Long Short-Term Memory (networks) and incorporate pre-trained word vectors in your model.

Section 4: Additional tips, techniques, and features

The final section of the book will explore the latest features available in the TensorFlow API. Topics include preparing your model for deployment, useful programming patterns, and other select subjects.

Other machine learning libraries

TensorFlow is not the only open source machine learning library out there. Below is a short list of various options for deep learning:

- **Caffe** focuses on convolutional neural networks and image processing, and is written in C++.
- **Chainer** is another flexible machine learning Python library capable of utilizing multiple GPUs on one machine.
- **CNTK** is Microsoft's entry into the open source machine learning library game. It uses its own model definition language to build distributed models declaratively.
- **Deeplearning4j** is a Java library specifically focused on neural networks. It is built to be scaled and integrated with Spark, Hadoop, and other Java-based distributed software.
- **Nervana Neon** is an efficient Python machine learning library, which is capable of using multiple GPUs on a single machine.
- **Theano** is an extremely flexible Python machine learning library written in Python. It is popular in research, as it is quite user friendly and capable of defining complex models fairly easily. TensorFlow's API is most similar to Theano's.
- **Torch** is a machine learning library that focuses on GPU implementation. It is written in Lua, and is backed by research teams at several large companies.

It's beyond the scope of this book to have an in-depth discussion on the merits of each of these libraries, but it is worth looking into them if you have the time. The authors of TensorFlow took inspiration from several of them when designing the framework.

Further reading ~~~~~~~~~~~~

If, after reading this book, you're interested in pursuing more with TensorFlow, here are a couple of valuable resources:

- **The official TensorFlow website**, which will contain the latest documentation, API, and tutorials
- **The TensorFlow GitHub repository**, where you can contribute to the open-source implementation of TensorFlow, as well as review the source code directly
- **Officially released machine learning models implemented in TensorFlow**. These models can be used as-is or be tweaked to suit your own goals
- **The Google Research Blog** provides the latest news from Google related to TensorFlow applications and updates.
- **Kaggle** is a wonderful place to find public datasets and compete with other data-minded people
- Data.gov is the U.S. government's portal to find public datasets all across the United States

Alright, that's enough of a pep-talk. Let's get started with **TensorFlow for Machine Intelligence**!