

# A DEEP DIVE INTO MACHINE LEARNING



DIG INTO MACHINE LEARNING. ANYTIME. ANYWHERE.

# A Deep Dive into Machine Learning

Copyright © 2018 SitePoint Pty. Ltd.

■ **Editor:** Adam Roberts

## Notice of Rights

All rights reserved. No part of this book may be reproduced, stored in a retrieval system or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical articles or reviews.

## Notice of Liability

The author and publisher have made every effort to ensure the accuracy of the information herein. However, the information contained in this book is sold without warranty, either express or implied. Neither the authors and SitePoint Pty. Ltd., nor its dealers or distributors will be held liable for any damages to be caused either directly or indirectly by the instructions contained in this book, or by the software or hardware products described herein.

## Trademark Notice

Rather than indicating every occurrence of a trademarked name as such, this book uses the names only in an editorial fashion and to the benefit of the trademark owner with no intention of infringement of the trademark.



Published by SitePoint Pty. Ltd.

48 Cambridge Street Collingwood

VIC Australia 3066

Web: [www.sitepoint.com](http://www.sitepoint.com)

Email: [books@sitepoint.com](mailto:books@sitepoint.com)

## About SitePoint

SitePoint specializes in publishing fun, practical, and easy-to-understand content for web professionals. Visit [sitepoint.com](http://sitepoint.com) to access our blogs, books, newsletters, articles, and community forums. You'll find a stack of information on JavaScript, PHP, Ruby, mobile development, design, and more.

# Table of Contents

<b>Preface .....</b>	<b>vii</b>
----------------------	------------

## **Chapter 1:    Become a Machine Learning Learning**

<b>Machine .....</b>	<b>8</b>
----------------------	----------

Big Picture .....	9
-------------------	---

Building Blocks .....	10
-----------------------	----

Real-world Inspiration .....	10
------------------------------	----

<b>Chapter 2:    Algorithm and Blues.....</b>	<b>11</b>
---	-----------

An Overview of Machine Learning Algorithms .....	12
--	----

<b>Chapter 3:    Present Tense: TensorFlow.....</b>	<b>13</b>
---	-----------

Learning.....	14
---------------	----

Code .....	14
------------	----

## **Chapter 4:    TensorFlow.js and Machine Learning**

<b>with JavaScript.....</b>	<b>16</b>
-----------------------------	-----------

TensorFlow.js.....	17
--------------------	----

Synaptic and Brain.js .....	18
-----------------------------	----

<b>Chapter 5: Solid Snake: Machine Learning with Python.....</b>	<b>19</b>
Pytorch.....	21
<b>Chapter 6: Pass the Torch .....</b>	<b>22</b>
<b>Chapter 7: Keras.....</b>	<b>24</b>
<b>Chapter 8: Caffe2.....</b>	<b>26</b>
<b>Chapter 9: CNTK, OK?.....</b>	<b>28</b>
Microsoft Cognitive Toolkit.....	29
<b>Chapter 10: Apache Spark MLlib.....</b>	<b>30</b>
<b>Chapter 11: Azure Machine Learning Service .....</b>	<b>32</b>
<b>Chapter 12: Google Cloud ML.....</b>	<b>34</b>
Cloud ML Engine .....	35
Cloud TPUs .....	35
<b>Chapter 13: Amazon SageMaker and IBM Watson...37</b>	

Amazon SageMaker.....	38
IBM Watson.....	38
 <b>Chapter 14: Core ML 2.....</b>	<b>40</b>
 <b>Chapter 15: Android: TensorFlow Lite and ML Kit .....</b>	<b>43</b>
TensorFlow Lite .....	44
ML Kit.....	44

# Preface

This little ebook began its life as an email course published by SitePoint. It provides a beginner-friendly introduction to machine learning, shining a torch across the landscape of machine learning and presenting lists of tools and resources for delving into this cutting-edge field.

# Become a Machine Learning Learning Machine

Chapter

# 1



When I thought about doing this series of “deep dives”—ongoing, thorough overviews of cutting-edge, complicated or misunderstood subjects—“machine learning and AI” was one topic I always came back to (the other one being “blockchain”, but that’s a task for a different day). A large number of people are excited about it; a smaller number know what they’re actually talking about.

There are thousands and thousands of start-ups and services which claim to harness machine learning (or neural networks or deep learning—pick your buzzword). Sometimes, their ML tech is nothing more than a glorified chat bot that’ll take a narrow range of inputs and come back with pre-chosen responses. Other times, it’s genuinely impressive, or scary, what this stuff can do. It’s sometimes hard, given the hype, to work out which is which.

But if you have any interest at all in the bleeding edge of technology, you should learn what machine learning is, and how to spot the legit ML tech. In this book, I’ll do my best to help.

## Big Picture

- Wait But Why’s post on [the coming artificial intelligence revolution](#) [waitbutwhy] is a typically excellent, good first step when thinking about AI and its potential impact on the world.
- A good look at [the mind of Google Assistant](#) [standard].
- [Examples of AI doing amazing things](#) [medium/archieai].
- One that wasn’t included: [AlphaGo Zero](#) [deepmind], the network that learned to be the best Go player in the world by playing itself.
- A Reddit thread of answers to the question: [If you had to show one paper to someone to show that machine learning is beautiful, what would you choose?](#) [reddit].

## Building Blocks

Some resources to help you frame everything else you see:

- [A machine learning glossary from Google](#) [developers.google].
- [A guide to how deep learning works, aimed at “everyone”](#) [medium.freecodecamp].

## Real-world Inspiration

- I found [this piece from Robin Sloan](#) [robinsloan], where he used machine learning to turn his text editor into a science fiction expert, able to autocomplete appropriately sci-fi stuff, hugely inspiring. It’s not AlphaGo-level abilities, but it does make the subject much more accessible to my mind.
- Finally, [MariFlow](#) [youtube/sethbling] is a self-driving Mario Kart example using a recurrent neural network.

# Algorithm and Blues

Chapter

2

## An Overview of Machine Learning Algorithms

Let's get a general idea of common algorithms involved in machine learning. This will help to form a foundation of knowledge on the subject.

- [A good intro to the most common ML algorithms](#) [analyticsvidhya], as well as just a good intro to the kinds of things that will crop up in the rest of the content I'll be sending your way.
- [A beginner's guide to the top 10 machine learning algorithms](#) [kdnuggets].
- [A video guide to writing your first machine learning code](#) [youtube/googledevelopers].
- If you a) have some more time, and b) enjoy math, the book *[The Mathematics of Machine Learning](#)* [github/mml-book] will be right up your alley, offering the mathematical foundations of ML, plus algorithms that use these.

# Present Tense: TensorFlow

Chapter

3

TensorFlow [tensorflow] is the open-source machine learning network that might be your best bet early on, so I thought it made sense to start here when it came to the actual code side of things. TensorFlow itself is a big subject, but this should be enough to get you started on your journey!

First, some intro content to set the scene.

- 9 tidbits to understand about TensorFlow [hackernoon].
- And TensorFlow Lite [developers.googleblog] built for smaller, less powerful devices like Raspberry Pis.

## Learning

- Here's a TensorFlow tutorial [github/aymericdamien] full of examples to get you started.
- Google's TensorFlow-packed crash course in machine learning [developers.google] is pretty definitive.
- An intro to TensorFlow Probability [medium/tensorflow], a toolbox for when you need to do things like quantify uncertainty, you want a generative model of data, stuff like that.
- This collection of machine learning recipes [youtube] uses TensorFlow for a lot of them.

## Code

- The Ruby implementation of TensorFlow [github/somaticio].
- The R implementation [tensorflow.rstudio].
- TensorFlowOnSpark [github/yahoo] is a Yahoo effort to establish distributed

TensorFlow with Apache Spark.

- [A collection of TensorFlow models](#) [github/sarasra].
- [Assorted examples found across the Web](#) [learningtensorflow].
- [TensorPack](#) [github/tensorpack] is a neural network training interface built on TF.
- [Sonnet](#) [github/deepmind] is another neural network library build on TF, this one from DeepMind, owned by Alphabet, who are behind some stuff like AlphaGo, that neural network that kinda mastered Go.
- [An implementation of YOLO: real-time object detection](#) [github/gliese581gg].

# TensorFlow.js and Machine Learning with JavaScript

Chapter

4



There's a universal law in development: if something exists, someone has made a JavaScript version of it. Actually, that's maybe a bit conservative: "many JavaScript versions of it". While JavaScript is less popular for machine learning than Python, R, Java or C/C++, I thought it made sense to take a look at it, since, for some of you, JavaScript is your go-to. So let's get started. This will mostly cover TensorFlow.js, then I'll link to some other options in the space. There are many, but the ones I've highlighted seem to be the most active/well-maintained.

- First up, [this article](#) [hackernoon] does a good job of busting some myths, and providing a gentle intro to ML with JS.

## TensorFlow.js

TensorFlow.js is an open-source machine learning library, allowing devs to use the high-level API to train and run machine learning models in browser (and also Node, as of a recent update). These models use the client's GPU, and can be easily added to a project with a script tag. If you like, you can also convert a pre-trained model from TensorFlow and Keras to the TensorFlow.js format.

- [Have a play](#) [playground.tensorflow] with a neural network in the browser, then fork it and make it better.
- Do you have 7 minutes? [Get to know TensorFlow.js, then](#) [medium.freecodecamp].
- [And the official intro to TensorFlow.js](#) [medium/tensorflow].
- [18 tips for training models with TensorFlow.js](#) [itnext].
- [A basic tutorial with TensorFlow.js](#) [medium/@tristansokol], solving a very simple problem.
- [A practical guide to TensorFlow.js](#) [blog.yellowant].

- Using TensorFlow.js to perform visual recognition on images with JavaScript [jamesthom].
- This developer used TensorFlow.js to build “front-end artificial intelligent” [blog.apptension], demonstrated via a spot of Flappy Bird.
- Use TensorFlow.js to work out where a webcam user’s gaze is directed [cpury.github].
- This developer made Amazon Alexa respond to sign language using TensorFlow.js [youtube/abhisheksingh].

## Synaptic and Brain.js

Synaptic [caza] bills itself as an architecture-free neural network library that’ll run on Node or a browser.

- Classifying handwritten digits with Synaptic.js [blog.webkid].
- A model of our immune system [medium/coinmonks] built using Synaptic.js.

Then, to finish up, a couple of links to show off Brain.js [github/brainjs].

- A guide to implementing Brain.js [itnext].
- And a video guide to using Brain.js to distinguish between light or dark colors [youtube].

# **Solid Snake: Machine Learning with Python**

Chapter

# 5

From what I've learned in my research for this book, when it comes to machine learning code, Python is just the most popular choice. It's relatively readable and simple to understand, which means users can spend less of their energy worrying about syntax errors. It also has a bunch of useful libraries available out of the box, and its popularity creates a virtuous circle, so a bunch of open-source ML libraries have been created by the community. Here's [a good summary](#) [hub.packtpub] of these and other arguments.

Now, let's take a look at what's out there for the budding Python machine learning developer.

- A step-by-step guide to [your first machine learning project](#) [machinelearningmastery] in Python.
- [100 Days of ML Code](#) [github/avik-jain] uses Python code for the exercises, so is a good way to go deep on both ML and Python's place in it.
- The Python code for [common machine learning algorithms](#) [github/susanli2016].
- [The best Python libraries for machine learning](#) [datascienceplus].
- [Some Jupyter notebooks](#) [github/ageron] that provide machine learning fundamentals in Python, using Scikit-Learn and TensorFlow (remember that!)
- [Helpful cheat sheets](#) [startupsventurecapital] for Python-based machine learning.
- [A simplified MultiAgent Python Snake game](#) [youtube/crazymuse] via deep reinforcement learning.
- If you're enjoying practicing machine learning, and you're thinking this may become an actual pursuit/career for you, here are [five useful books for learning machine learning in Python](#) [quantstart].

# Pytorch

PyTorch is a Python library for machine learning, built by Facebook and based on Lua-based machine learning library Torch (which we'll get into later on in the Deep Dive). I thought it made sense to look into this while we have a Python focus.

- [An introduction to PyTorch](#) [analyticsvidhya].
- [A comparison between PyTorch and TensorFlow](#) [towardsdatascience].
- [A PyTorch template project](#) [github/victoresque].
- [More tutorials!](#) [github/morvanzhou].
- [A PyTorch deep learning minicourse](#) [github/atcold].

# Pass the Torch

## Chapter 6

Torch, a Lua-based machine learning framework, is sold as an easy machine learning framework, thanks to Lua's simplicity. It's rather popular indeed among the machine learning set, so let's take a look!

- Getting started: installing Torch, followed by some examples [torch]. Then, Hello World in Torch [mdtux89.github].
- A hands-on intro to Torch [github/alexbw].
- A video Torch tutorial [youtube].
- More tutorials [torch] (recurrent neural networks, graph-style neural networks) for learning Torch.
- Facebook's Torch extensions, so you can ML like the pros [github/facebookarchive].
- High-performance, reusable recurrent neural networks, and long short-term memory modules in Torch [github/jcjohnson].
- Code to easily implement a multi-layer recurrent neural network [github/karpathy].
- Lastly for this chapter, TorchCraft [github/torchcraft] lets you use a machine learning network to play StarCraft. A complete overview of the project [arxiv].

Keras

Chapter

7



Keras is a Python-based neural network API, capable of running on top of TensorFlow or CNTK. It's built to make experimentation quick and easy. Sounds like our kind of API!

A bunch of the examples below involve convolutional neural networks, so here's a reminder on what that means.

- An intro to Python deep learning with Keras [machinelearningmastery].
- And a guide to running it on top of TensorFlow [visualstudiomagazine].
- A Keras cheatsheet [github/kailashahirwar].
- This demo uses a pre-trained, stock Keras image classifier [blog.stratospark] to identify food by image.
- A Keras implementation of a popular machine vision demo: extending an image beyond its original borders [github/bendangnuksung].
- A seven-step process to mastering deep learning [kdnuggets] with Keras.
- A guide to building a convolutional neural network in Python, with Keras [elitedatascience].
- Train a neural network on a custom image dataset, using Keras [pyimaginesearch].
- A convolutional neural network to generate ASCII art [github/osciart].
- Picasso [medium/merantix] is a free, open source deep neural network for Keras and TensorFlow designed to ensure neural networks are trained on the correct dataset.

Caffe2

Chapter

8

Caffe2 is an open-source deep learning framework. I've been using machine learning and deep learning kinda interchangeably, so here's a good explanation of the differences. It's fast, often used in convolutional neural networks, can run on mobile(!) and also comes up a lot in machine vision.

It's based on the framework Caffe, originally developed at the University of Berkeley. Caffe2 was developed at Facebook, so the company still has a lot of involvement. There's also a process underway to merge Caffe2 and PyTorch [pytorch], just to make things more confusing for us. (Getting this relatively straight in my mind has been a fun project!)

- This is a good introduction to Caffe2 [caffe2], outlining the framework's characteristics, strengths and usage.
- Caffe Model Zoo [github/caffe2] is a repo full of pre-trained networks that can be used immediately. An essential visit for the budding Caffe-enthusiast. A guide to using them [caffe2].
- How to implement Caffe2 in a mobile project [caffe2].
- An AI-powered camera demo, with Caffe2 [caffe].
- How to use Detectron, Facebook's open-source framework, for object detection and masking [hackernoon].

# CNTK, OK?

## Chapter 9

# Microsoft Cognitive Toolkit

Microsoft Cognitive Toolkit is an open-source deep learning framework. It's high-performance, makes setting up a neural network easy, and you can work with it via a Python, C++ or command line interface. (It used to be known as CNTK, which I will use here because brevity.)

- How to install CNTK from the source [github/astorfi], which is more compatible with your system configuration, and also in a Python virtual environment.
- Ruby binding for CNTK [github/tamuratak].
- CNTK images on Docker Hub [hub.docker].
- How to train a CNTK handwriting classifier model in Python [github/wmeints], then use the same model in Java via a REST API.
- The official model gallery [microsoft], full of examples and tutorials for using CNTK.
- One of these: complete an image recognition task with a convolutional neural network [github/microsoft].
- Then some simple, ready-to-use tutorials for CNTK [github/astorfi]
- Build a neural network with Python and CNTK [adventuresinmachinelearning].
- Build a handwriting classifier with CNTK [medium/@tuzzer].
- “Introduction to CNTK Succinctly” [syncfusion]. A book you can download as a free PDF (that's the online version).

# Chapter

# 10

## Apache Spark MLlib

To close out the Machine Learning Deep Dive, I'll be taking a look at two areas: cloud-based machine learning integrations and moving machine learning to mobile.

Cloud-based ML lets you take the machine learning principles and libraries you've grown accustomed to and vastly increase your resources. There are a few competitors in this space, all established names. Here's [a comparison of the options I'll cover](#)—although it doesn't include this chapter's entry. Speaking of which ...

It doesn't have the catchiest name, but [Spark MLlib](#) is a simple, scalable machine library, built on top of the Spark cluster computing framework. MLlib plays nice with popular ML languages/frameworks and runs very, very fast. Let's take a look.

- [An argument for using MLlib](#) [infoword].
- [An official programming guide](#) [spark.apache].
- [An intro to using Spark MLlib with Python](#) [datacamp].
- [Build a recommendation engine with MLlib](#) [chimpler.wordpress].
- [Use MLlib to analyze flight delays](#) [mapr].
- [Perform Twitter sentiment analysis with MLlib](#) [github/p7h].
- [MLlib vs TensorFlow](#) [analyticsindiamag].
- But why choose just one? [Using MLlib and TensorFlow together](#) [adaltas].
- [An official guide for MLlib+TensorFlow, from Google](#) [cloud.google].

# Azure Machine Learning Service

Chapter

# 11



Azure Machine Learning Service is Microsoft's entry into the cloud machine learning game. It has a Python API that can plug into existing machine learning libraries like TensorFlow and PyTorch, and you can develop locally and then scale up. This is in contrast to its Machine Learning Studio product, which offers a more abstract, drag-and-drop interface for working with machine learning models, no coding required.

- An introduction to Azure ML Service [thenewstack].
- A workflow to use when using Azure ML [docs.microsoft].
- Some hands-on Azure ML labs [github/azure-readiness].
- Build a predictive model with Azure ML [blog.datasciencedojo].
- Train PyTorch models with Azure ML [docs.microsoft].
- Build an Azure ML service to predict car accidents [github/dwilson1988].
- Classify handwriting with TensorFlow and Azure [github/azure].
- Use deep learning in Azure ML Service to analyze the human footprint in the Amazon rainforest [medium/@weshoffman].

# Google Cloud ML

## Chapter 12

Google offers a variety of products for integrating AI into existing services: Cloud Talent Solution for making hiring easier via AI. I won't look at these directly, as they abstract away the machine learning fun and that's not really our focus here. Let's look at the fun stuff for machine learning devs instead.

## Cloud ML Engine

- [An introduction to Cloud ML Engine](#) [bmc].
- A guide to properly setting up a task.py file to [train machine learning models with Cloud ML Engine](#) [towardsdatascience].
- [This general intro to TensorFlow](#) [liufuyang.github] introduces Google Cloud ML Engine at part 4, but it might be worth following along for the whole thing to get a better picture of the whole project.
- Using molecular data, this model [predicts molecular energy](#) [cloud.google], using Apache Beam, TensorFlow and Google Cloud ML Engine.
- Or [predict income data from census data](#) [github/googlecloudplatform] with Keras and Cloud ML Engine.
- [More samples here!](#) [github/googlecloudplatform].

## Cloud TPUs

Google also offers TPUs, a class of application-specific circuit custom built to accelerate machine learning. You can use these in your own work, provided you're a Google Cloud customer, via its Cloud TPU [cloud.google] service.

- [An intro](#) [bmc] to exactly what TPUs are, and why they're handy for machine learning.
- [A quickstart guide](#) [cloud.google].

- And then a guide to setting up TPUs for machine learning [blog.goodaudience].
- Reference models and tools for Cloud TPUs [github/tensorflow]. For example, an implementation of the RetinaNet object detection model [github/tensorflow].
- Speaking of, how to train and serve a realtime mobile object detector in 30 minutes [medium/tensorflow].
- How to write a custom Estimator model for Cloud TPUs [medium/tensorflow].

# Amazon SageMaker and IBM Watson

Chapter

# 13

## Amazon SageMaker

SageMaker is Amazon’s low-level machine learning platform that allows developers to use familiar frameworks to build and run machine learning projects on AWS. Amazon has more user-friendly, plug-and-play, no-code-required tools like image and video analyzer Rekognition, but SageMaker is for those who want to get their hands dirty.

- The full “how it works” docs are worth reading [docs.aws.amazon]. I found the “batch transform” tool [docs.aws.amazon] particularly interesting.
- Is it right for you? [hackernoon]. Takeaways for me: it’s pretty easy to get started, but potential dealbreakers include the expense and the (possibly) time-consuming data uploads you’ll need to plan for.
- SageMaker examples [github/awslabs].
- Sagify [github/kenza-ai] is a command-line program that lets you train and deploy machine learning models onto Amazon SageMaker in a few steps.
- Speaking of Amazon, a guide to deploying a serverless machine learning model [ritchievink].

## IBM Watson

You might remember Watson as the system that beat humans at Jeopardy. IBM has taken that system, or at least its brand name, and built it out into tools for businesses, analysts and data scientists. The landscape here is a bit confusing, with many similarly named services, and some just aren’t aimed at those wanting to build their own models from scratch. Here are a few links that may help you though.

- A three-part guide [medium/ibm-watson-data-lab] to building a model that

can predict housing prices using Watson.

- Or one that models building violations [developer.ibm], using a Watson service that tunes and redeploys models as more data becomes available.
- Thanks to a partnership with Apple, Watson has also been integrated with the iOS CoreML framework [medium/flawless-app-stories]. Funnily enough, we're about to look at mobile machine learning, so consider that an excellent segue.

# Core ML 2 | Chapter 14



Mobile developers, this is your time to shine! [Core ML 2](#) is Apple’s official framework for machine learning on its products. The company uses it on apps like Siri and the camera, but you can use it on your stuff too. I understand not everyone is a Swift dev, but I like to see where machine learning is being used, and Apple in particular is doing some interesting stuff.

- [CreateML](#) [developer.apple] is Apple’s “no coding required” machine learning solution.
- [Core ML 2 vs Google’s ML Kit](#) [venturebeat].
- [An intro to Core ML](#) [appcoda].
- [How to recognize handwriting with Core ML](#) [github/brainadvent].
- [How to build Silicon Valley’s Not Hotdog app with Core ML in an afternoon](#) [heartbeat.fritz].
- [A big list of Core ML models](#) [github/likedan]. That page also has a list of models that *could* be converted to Core ML, so get cracking. Existing models I liked the look of:
  - [GoogLeNetPlaces](#) [github/chenyi1989] detects the location of a scene of an image (airport, forest, etc.).
  - [Nudity-CoreML](#) [github/ph1ps] detects nudity in an image.
  - [Exermote](#) [github/lausbert]. When someone exercises, and wears an iPhone on their arm, the model predicts.
- [A Core ML model that vocalizes sign language](#) [github/ardamavi].
- [A prototype of a hand pose and gesture tracking model](#) [youtube/2020cv] using TensorFlow, Keras and Core ML.
- An interesting piece with some implications for the security of your models:

an attempt at reverse-engineering Core ML models [heartbeat.fritz], based on how they're stored in apps.

# Android: TensorFlow Lite and ML Kit

## Chapter 15

Ah, Android. When compared to the polish of the iOS ecosystem, things always tend to get a little more diffuse, and machine learning tooling is no exception. Let's take a look at the two options you have for Android machine learning: TensorFlow Lite and ML Kit. [Here's a good intro](#) [medium/xebia-france] to the two sides of the coin, complete with their respective limitations.

## TensorFlow Lite

To take this Deep Dive full-circle, the official solution for running machine learning models on Android is [TensorFlow Lite](#) [tensorflow].

- [Build an object-detection library with TensorFlow Lite](#) [medium/midorks].
- [Build a custom machine learning model with TensorFlow Lite](#) [medium/over-engineering].
- And [a guide to using a TensorFlow Lite model for inference with ML Kit](#) [firebase.google].

## ML Kit

[Google's ML Kit from Firebase](#) is a machine learning SDK for Android and iOS. It's in beta, and was released at 2018's Google I/O. It's more user-friendly than TF Lite, so might be a good place to start.

- [An intro](#) [auto0].
- [How to use ML Kit to in iOS for face detection](#) [github/appcoda].
- [How ML Kit performs on iOS](#) [blog.xmartlabs].
- Finally, [a guide to using TensorFlow Lite and ML Kit to build a Pokidex](#) [github/the-dagger].