
TensoFlow-World-Resources Documentation

Release 1.0

Amirsina Torfi

Jan 08, 2019

1	Introduction	1
1.1	Motivation	1
1.2	How to make the most of this effort	2
2	Entrance to TensorFlow World	3
2.1	Installation	3
2.2	Getting Started	4
2.3	Going Deeper in TensorFlow	4
3	Programming with TensorFlow	5
3.1	Reading data and input pipeline	5
3.2	Variables	6
3.3	TensorFlow Utilities	6
4	TensorFlow Tutorials	9
4.1	Linear and Logistic Regression	9
4.2	Convolutional Neural Networks	9
4.3	Recurrent Neural Networks	10
4.4	Autoencoders	11
4.5	Generative models	11
4.6	Multiple GPUs	11
5	TensorFlow Projects	13
5.1	Comprehensive Tutorials	13
5.2	Models	13
6	Published Resources	15
6.1	Online Courses and Documentations	15
6.2	Books	15
7	Contributing	17
7.1	Pull Request Process	17
7.2	Final Note	17
8	Contributor Covenant Code of Conduct	19
8.1	Our Pledge	19
8.2	Our Standards	19

8.3	Our Responsibilities	20
8.4	Scope	20
8.5	Enforcement	20
8.6	Attribution	20
9	LICENSE	21

The purpose of this project is to introduce a shortcut to developers and researcher for finding useful resources about TensorFlow.

1.1 Motivation

There are different motivations for this open source project.

1.1.1 Why using TensorFlow?

A deep learning is of great interest these days, the crucial necessity for rapid and optimized implementation of the algorithms and designing architectures is the software environment. TensorFlow is designed to facilitate this goal. The strong advantage of TensorFlow is its flexibility in designing highly modular model which also can be a disadvantage too for beginners since lots of the pieces must be considered together for creating the model. This issue has been facilitated as well by developing high-level APIs such as [Keras](#) and [Slim](#) which gather lots of the design puzzle pieces. The interesting point about TensorFlow is that **its trace can be found anywhere these days**. Lots of the researchers and developers are using it and *its community is growing with the speed of light!* So the possible issues can be overcome easily since they might be the issues of lots of other people considering a large number of people involved in TensorFlow community.

1.1.2 What's the point of this open source project?

There are other similar repositories similar to this repository and are very comprehensive and useful and to be honest they made me ponder if there is a necessity for this repository! A great example is [awesome-tensorflow](#) repository which is a curated list of different TensorFlow resources.

The point of this repository is that the resources are being targeted. The organization of the resources is such that the user can easily find the things he/she is looking for. We divided the resources to a large number of categories that in the beginning one may have a headache!!! However, if someone knows what is being located, it is very easy to find

the most related resources. Even if someone doesn't know what to look for, in the beginning, the general resources have been provided.

1.2 How to make the most of this effort

The written and visual resources have been split. Moreover, As one can search in the documentation, the number of categories might look to be too much. For finding the most relevant resources, please at first look through the general resources.

Entrance to TensorFlow World

In this section, different TensorFlow topics and their associated resources will be addressed.

2.1 Installation

First of all, the TensorFlow must be installed!

2.1.1 Written Resources

- [Installing TensorFlow](#): Official TensorFlow installation
- [Install TensorFlow from the source](#): A comprehensive guide on how to install TensorFlow from the source using python/anaconda
- [TensorFlow Installation](#): A short TensorFlow installation guide powered by NVIDIA
- [7 SIMPLE STEPS TO INSTALL TENSORFLOW ON WINDOWS](#): A concise tutorial for installing TensorFlow on Windows

2.1.2 Visual Resources

- [Install TensorFlow on Ubuntu](#): A comprehensive tutorial on how to install TensorFlow on Ubuntu
- [Installation of TensorFlow](#): The video covers how to setup TensorFlow
- [Installing CPU and GPU TensorFlow on Windows](#): A tutorial on TensorFlow installation for Windows
- [Installing the GPU version of TensorFlow for making use of your CUDA GPU](#): A GPU-targeted TensorFlow installation

2.2 Getting Started

This part points to resources on how to start to code with TensorFlow

2.2.1 Written Resources

- [Getting Started With TensorFlow](#): This guide gets you started programming in TensorFlow
- [learning tensorflow](#): A great resource to start
- [Welcome to TensorFlow World](#): A simple and concise start to TensorFlow

2.2.2 Visual Resources

- [Gentlest Introduction to Tensorflow](#)
- [TensorFlow in 5 Minutes](#)
- [Deep Learning with TensorFlow - Introduction to TensorFlow](#)
- [TensorFlow Tutorial \(Sherry Moore, Google Brain\)](#)
- [Deep Learning with Neural Networks and TensorFlow Introduction](#)
- [A fast with TensorFlow](#)

2.3 Going Deeper in TensorFlow

Advanced machine learning users can go deeper in TensorFlow in order to *hit the root*. Scratching the surface may never take us too further!

2.3.1 Written Resources

- [TensorFlow Mechanics](#): More experienced machine learning users can dig more in TensorFlow
- [Advanced TensorFlow](#): Advanced Tutorials in TensorFlow
- [We Need to Go Deeper: A Practical Guide to Tensorflow and Inception](#)
- [Wide and Deep Learning - Better Together with TensorFlow](#): A tutorial by Google Research Blog

2.3.2 Visual Resources

- [TensorFlow DeepDive](#): More experienced machine learning users can dig more in TensorFlow
- [Go Deeper - Transfer Learning: TensorFlow and Deep Learning](#)
- [Distributed TensorFlow - Design Patterns and Best Practices](#): A talk that was given at the Advanced Spark and TensorFlow Meetup
- [Fundamentals of TensorFlow](#)
- [TensorFlow Wide and Deep - Advanced Classification the easy way](#)
- [Tensorflow and deep learning - without a PhD](#): A great tutorial on TensorFlow workflow

The references here, deal with the details of programming and writing TensorFlow code.

3.1 Reading data and input pipeline

The first part is always how to prepare data and how to provide the pipeline to feed it to TensorFlow. Usually providing the input pipeline can be complicated, even more than the structure design!

3.1.1 Written resources

- [Dataset API for TensorFlow Input Pipelines](#): A TensorFlow official documentation on *Using the Dataset API for TensorFlow Input Pipelines*
- [TensorFlow input pipeline](#): Input pipeline provided by Stanford.
- [TensorFlow input pipeline example](#): A working example.
- [TensorFlow Data Input](#): TensorFlow Data Input: Placeholders, Protobufs & Queues
- [Reading data](#): The official documentation by the TensorFlow on how to read data
- [basics of reading a CSV file](#): A tutorial on reading a CSV file
- [Custom Data Readers](#): Official documentation on this how to define a reader.

3.1.2 Visual resources

- [Tensorflow tutorial on TFRecords](#): A tutorial on how to transform data into TFRecords

3.2 Variables

Variables are supposed to hold the parameters and supersede by new values as the parameters are updated. Variables must be clearly set and initialized.

3.2.1 Written Resources

Creation, Initialization

- [Variables Creation and Initialization](#): An official documentation on setting up variables
- [Introduction to TensorFlow Variables - Creation and Initialization](#): This tutorial deals with defining and initializing TensorFlow variables
- [Variables](#): An introduction to variables

Saving and restoring

- [Saving and Loading Variables](#): The official documentation on saving and restoring variables
- [save and restore Tensorflow models](#): A quick tutorial to save and restore Tensorflow models

Sharing Variables

- [Sharing Variables](#): The official documentation on how to share variables

3.2.2 Visual Resources

- [Deep Learning with Tensorflow - Tensors and Variables](#): A Tensorflow tutorial for introducing Tensors, Variables and Placeholders
- [Tensorflow Variables](#): A quick introduction to TensorFlow variables
- [Save and Restore in TensorFlow](#): TensorFlow Tutorial on Save and Restore variables

3.3 TensorFlow Utilities

Different utilities empower TensorFlow for faster computation in a more monitored manner.

3.3.1 Written Resources

Supervisor

- [Supervisor - Training Helper for Days-Long Trainings](#): The official documentation for TensorFlow Supervisor.
- [Using TensorFlow Supervisor with TensorBoard summary groups](#): Using both TensorBoard and the Supervisor for profit
- [Tensorflow example](#): A TensorFlow example using Supervisor.

TensorFlow Debugger

- [TensorFlow Debugger \(tfdbg\) Command-Line-Interface Tutorial](#): Official documentation for using debugger for MNIST
- [How to Use TensorFlow Debugger with tf.contrib.learn](#): A more high-level method to use the debugger.
- [Debugging TensorFlow Codes: A Practical Guide for Debugging TensorFlow Codes](#)
- [Debug TensorFlow Models with tfdbg](#): A tutorial by Google Developers Blog

MetaGraphs

- [Exporting and Importing a MetaGraph](#): Official TensorFlow documentation
- [Model checkpointing using meta-graphs in TensorFlow](#): A working example

Tensorboard

- [TensorBoard - Visualizing Learning](#): Official documentation by TensorFlow.
- [TensorFlow Ops](#): Provided by Stanford
- [Visualisation with TensorBoard](#): A tutorial on how to create and visualize a graph using TensorBoard
- [Tensorboard](#): A brief tutorial on Tensorboard

3.3.2 Visual Resources

- [Hands-on TensorBoard \(TensorFlow Dev Summit 2017\)](#): An introduction to the amazing things you can do with TensorBoard
- [Tensorboard Explained in 5 Min](#): Providing the code for a simple handwritten character classifier in Python and visualizing it in Tensorboard
- [How to Use Tensorboard](#): Going through a bunch of different features in Tensorboard

This section is dedicated to provide tutorial resources on the implementation of different models with TensorFlow.

4.1 Linear and Logistic Regression

4.1.1 Written Resources

- [TensorFlow Linear Model Tutorial](#): Using `TF.Learn` API in TensorFlow to solve a binary classification problem
- [Linear Regression in Tensorflow](#): Predicting house prices in Boston area
- [Linear regression with Tensorflow](#): Make use of tensorflow for numeric computation using data flow graphs
- [Logistic Regression in Tensorflow with SMOTE](#): Implementation of Logistic Regression in TensorFlow
- [A TensorFlow Tutorial - Email Classification](#): Using a simple logistic regression classifier
- [Linear Regression using TensorFlow](#): Training a linear model by TensorFlow
- [Logistic Regression using TensorFlow](#): Training a logistic regression by TensorFlow for binary classification

4.1.2 Visual Resources

- [Deep Learning with Tensorflow - Logistic Regression](#): A tutorial on Logistic Regression
- [Deep Learning with Tensorflow - Linear Regression with TensorFlow](#): A tutorial on Linear Regression

4.2 Convolutional Neural Networks

4.2.1 Written Resources

- [Convolutional Neural Networks](#): Official TensorFlow documentation

- [Convolutional Neural Networks using TensorFlow](#): Training a classifier using convolutional neural networks
- [Image classifier using convolutional neural network](#): Building a convolutional neural network based image classifier
- [Convolutional Neural Network CNN with TensorFlow tutorial](#): It covers how to write a basic convolutional neural network within TensorFlow with Python
- [Deep Learning CNNs in Tensorflow with GPUs](#): Designing the architecture of a convolutional neural network (CNN)

4.2.2 Visual Resources

- [Deep Learning with Neural Networks: Convolutional Neural Networks with TensorFlow](#)
- [TensorFlow Tutorial: Convolutional Neural Network](#)
- [Understanding Convolution with TensorFlow](#): A tutorial on Convolution operation with TensorFlow
- [Deep Learning with Tensorflow: Convolutional Network with TensorFlow](#)

4.3 Recurrent Neural Networks

4.3.1 Written Resources

- [Recurrent Neural Networks: TensorFlow official documentation](#)
- [How to build a Recurrent Neural Network in TensorFlow](#): How to build a simple working Recurrent Neural Network in TensorFlow
- [Recurrent Neural Networks in Tensorflow](#): Building a vanilla recurrent neural network (RNN) from the ground up in Tensorflow
- [RNNs in Tensorflow - a Practical Guide and Undocumented Features](#): Going over some of the best practices for working with RNNs in Tensorflow
- [RNN / LSTM cell example in TensorFlow and Python](#): Covering how to code a Recurrent Neural Network model with an LSTM in TensorFlow
- [Sequence prediction using recurrent neural networks\(LSTM\) with TensorFlow](#): How to approximate a sequence of vectors using a recurrent neural networks
- [TensorFlow RNN Tutorial](#): Recurrent Neural Networks for exploring time series and developing speech recognition capabilities

4.3.2 Visual Resources

- [Deep Learning with Neural Networks and TensorFlow: Recurrent Neural Networks \(RNN\)](#)
- [An Introduction to LSTMs in Tensorflow](#): A brief tutorial
- [Deep Learning with Tensorflow - The Recurrent Neural Network Model](#): A tutorial on the Recurrent Neural Network Models
- [Sequence Models and the RNN API](#): TensorFlow Dev Summit 2017
- [RNN Example in Tensorflow](#): A quick tutorial

4.4 Autoencoders

4.4.1 Written Resources

- [Deep Autoencoder with TensorFlow: An open source project](#)
- [Variational Autoencoder in TensorFlow: A tutorial on Variational Autoencoder](#)
- [Diving Into TensorFlow With Stacked Autoencoders: A nice brief tutorials](#)
- [Convolutional Autoencoders in Tensorflow: Implementing a single layer CAE](#)
- [Variational Autoencoder using Tensorflow: Facial expression low dimensional embedding](#)

4.4.2 Visual Resources

- [Deep Learning with Tensorflow - Autoencoder Structure: Tutorial on Autoencoder models](#)
- [Deep Learning with Tensorflow - RBMs and Autoencoders: Tutorial on Restricted Boltzmann machines and AEs](#)

4.5 Generative models

4.5.1 Written Resources

- [Generative Adversarial Nets in TensorFlow: Implementing GAN using TensorFlow, with MNIST data](#)
- [Generative Adversarial Networks: A working example of Generative Adversarial Networks](#)

4.5.2 Visual Resources

- [TensorFlow Tutorial - Adversarial Examples: A tutorial on a working example for generative models](#)

4.6 Multiple GPUs

4.6.1 Written Resources

- [Using GPUs: Official TensorFlow documentation](#)
- [Deep Learning with Multiple GPUs on Rescale: TensorFlow Tutorial](#)

TensorFlow Projects

This section is dedicated to provide resources that are mainly open source projects developed by TensorFlow. Those might be comprehensive tutorials on working example.

5.1 Comprehensive Tutorials

- [TensorFlow-World](#): Concise and ready-to-use TensorFlow tutorials with detailed documentation
- [TensorFlow-Tutorials](#): Introduction to deep learning based on Google's TensorFlow framework
- [TensorFlow Tutorials](#): Organized tutorials in TensorFlow
- [TensorFlow-Examples](#): Providing working examples in TensorFlow
- [Tensorflow Tutorials using Jupyter Notebook](#): TensorFlow tutorials written in Python plus Jupyter Notebook
- [TensorFlow for Deep Learning Research](#): Code examples for the Stanford course CS 20SI

5.2 Models

- [TensorFlow Models](#): Machine learning models implemented in TensorFlow
- [Tensorflow VGG16 and VGG19](#): Implementation of VGG 16 and VGG 19 based on tensorflow-vgg16 and Caffe to Tensorflow
- [ResNet in TensorFlow](#): Implementation of Deep Residual Learning for Image Recognition
- [Inception in TensorFlow](#): Train the Inception v3 architecture
- [A TensorFlow implementation of DeepMind WaveNet paper](#): TensorFlow implementation of the WaveNet generative neural network architecture for audio generation
- [3D Convolutional Neural Networks for Speaker Verification](#): Implementation of 3D Convolutional Neural Networks for Speaker Verification application in TensorFlow.

- [Domain Transfer Network \(DTN\)](#): The implementation of [Unsupervised Cross-Domain Image Generation in TensorFlow](#)
- [Neural Style](#): The Neural Style algorithm implementation that synthesizes a pastiche
- [SqueezeNet in TensorFlow](#): Tensorflow implementation of SqueezeNet

Published Resources

This section is dedicated to provide published resources on TensorFlow, Such as websites, blogs, and books.

6.1 Online Courses and Documentations

- [LearningTensorFlow](#): Beginner-level tutorials for a TensorFlow
- [Deep Learning by Google](#): A free online course developed by Google and Udacity
- [Tensorflow for Deep Learning Research](#): A comprehensive course by Stanford
- [Creative Applications of Deep Learning with TensorFlow](#): A non-free course on TensorFlow
- [Deep Learning with TensorFlow Tutorial](#): In this TensorFlow course, you will be able to learn the basic concepts of TensorFlow

6.2 Books

- [TensorFlow Machine Learning Cookbook](#): Quick guide to implementing TensorFlow in your day-to-day machine learning activities
- [Deep Learning with TensorFlow](#): Throughout the book, you'll learn how to implement deep learning algorithms for machine learning systems
- [First contact with TensorFlow](#): An online book on TensorFlow
- [Building Machine Learning Projects with TensorFlow](#): Learn how to implement TensorFlow in production
- [Learning TensorFlow](#): This book is an end-to-end guide to TensorFlow
- [Machine Learning with TensorFlow](#): Tackle common commercial machine learning problems with Google's TensorFlow library
- [Getting Started with TensorFlow](#): An easy-to-understand book on TensorFlow

- [Hands-On Machine Learning with Scikit-Learn and TensorFlow](#): By using examples, theory, the book help to gain an understanding of the machine learning concepts
- [Machine Learning with TensorFlow \(MEAP\)](#): An introduction to the concepts of TensorFlow

For typos, please do not create a pull request. Instead, declare them in issues or email the repository owner. Please note we have a code of conduct, please follow it in all your interactions with the project.

7.1 Pull Request Process

Please consider the following criterions in order to help us in a better way:

1. The pull request is mainly expected to be a link suggestion.
2. Please make sure your suggested resources are not obsolete or broken.
3. Ensure any install or build dependencies are removed before the end of the layer when doing a build and creating a pull request.
4. Add comments with details of changes to the interface, this includes new environment variables, exposed ports, useful file locations and container parameters.
5. You may merge the Pull Request in once you have the sign-off of at least one other developer, or if you do not have permission to do that, you may request the owner to merge it for you if you believe all checks are passed.

7.2 Final Note

We are looking forward to your kind feedback. Please help us to improve this open source project and make our work better. For contribution, please create a pull request and we will investigate it promptly. Once again, we appreciate your kind feedback and elaborate code inspections.

Contributor Covenant Code of Conduct

8.1 Our Pledge

In the interest of fostering an open and welcoming environment, we as contributors and maintainers pledge to making participation in our project and our community a harassment-free experience for everyone, regardless of age, body size, disability, ethnicity, gender identity and expression, level of experience, nationality, personal appearance, race, religion, or sexual identity and orientation.

8.2 Our Standards

Examples of behavior that contributes to creating a positive environment include:

- Using welcoming and inclusive language
- Being respectful of differing viewpoints and experiences
- Gracefully accepting constructive criticism
- Focusing on what is best for the community
- Showing empathy towards other community members

Examples of unacceptable behavior by participants include:

- The use of sexualized language or imagery and unwelcome sexual attention or advances
- Trolling, insulting/derogatory comments, and personal or political attacks
- Public or private harassment
- Publishing others' private information, such as a physical or electronic address, without explicit permission
- Other conduct which could reasonably be considered inappropriate in a professional setting

8.3 Our Responsibilities

Project maintainers are responsible for clarifying the standards of acceptable behavior and are expected to take appropriate and fair corrective action in response to any instances of unacceptable behavior.

Project maintainers have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, or to ban temporarily or permanently any contributor for other behaviors that they deem inappropriate, threatening, offensive, or harmful.

8.4 Scope

This Code of Conduct applies both within project spaces and in public spaces when an individual is representing the project or its community. Examples of representing a project or community include using an official project e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event. Representation of a project may be further defined and clarified by project maintainers.

8.5 Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported by contacting the project team at amirsina.torfi@gmail.com. The project team will review and investigate all complaints, and will respond in a way that it deems appropriate to the circumstances. The project team is obligated to maintain confidentiality with regard to the reporter of an incident. Further details of specific enforcement policies may be posted separately.

Project maintainers who do not follow or enforce the Code of Conduct in good faith may face temporary or permanent repercussions as determined by other members of the project's leadership.

8.6 Attribution

This Code of Conduct is adapted from the [Contributor Covenant][homepage], version 1.4, available at [<http://contributor-covenant.org/version/1/4/>][{}version]

[homepage]: <http://contributor-covenant.org> [version]: <http://contributor-covenant.org/version/1/4/>

CHAPTER 9

LICENSE

MIT License

Copyright (c) 2017 Amirsina Torfi

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.