TensorFlow Examples and Tutorials

Tutorial index

0 - Prerequisite

- Introduction to Machine Learning.
- Introduction to MNIST Dataset.

1 - Introduction

- **Hello World** (notebook). Very simple example to learn how to print "hello world" using TensorFlow 2.0.
- **Basic Operations** (notebook). A simple example that cover TensorFlow 2.0 basic operations.

2 - Basic Models

- **Linear Regression** (notebook). Implement a Linear Regression with TensorFlow 2.0.
- **Logistic Regression** (notebook). Implement a Logistic Regression with TensorFlow 2.0.
- **Word2Vec (Word Embedding)** (notebook). Build a Word Embedding Model (Word2Vec) from Wikipedia data, with TensorFlow 2.0.

3 - Neural Networks

Supervised

- **Simple Neural Network** (notebook). Use TensorFlow 2.0 'layers' and 'model' API to build a simple neural network to classify MNIST digits dataset.
- **Simple Neural Network (low-level)** (notebook). Raw implementation of a simple neural network to classify MNIST digits dataset.
- **Convolutional Neural Network** (notebook). Use TensorFlow 2.0 'layers' and 'model' API to build a convolutional neural network to classify MNIST digits dataset.

- **Convolutional Neural Network (low-level)** (notebook). Raw implementation of a convolutional neural network to classify MNIST digits dataset.
- **Recurrent Neural Network (LSTM)** (notebook). Build a recurrent neural network (LSTM) to classify MNIST digits dataset, using TensorFlow 2.0 'layers' and 'model' API.
- **Bi-directional Recurrent Neural Network (LSTM)** (notebook). Build a bi-directional recurrent neural network (LSTM) to classify MNIST digits dataset, using TensorFlow 2.0 'layers' and 'model' API.
- **Dynamic Recurrent Neural Network (LSTM)** (notebook). Build a recurrent neural network (LSTM) that performs dynamic calculation to classify sequences of variable length, using TensorFlow 2.0 'layers' and 'model' API.

Unsupervised

- **Auto-Encoder** (notebook). Build an auto-encoder to encode an image to a lower dimension and re-construct it.
- **DCGAN (Deep Convolutional Generative Adversarial Networks)** (notebook). Build a Deep Convolutional Generative Adversarial Network (DCGAN) to generate images from noise.

4 - Utilities

- **Save and Restore a model** (notebook). Save and Restore a model with TensorFlow 2.0.
- **Build Custom Layers & Modules** (notebook). Learn how to build your own layers / modules and integrate them into TensorFlow 2.0 Models.

5 - Data Management

- **Load and Parse data** (notebook). Build efficient data pipeline with TensorFlow 2.0 (Numpy arrays, Images, CSV files, custom data, ...).
- **Build and Load TFRecords** (notebook). Convert data into TFRecords format, and load them with TensorFlow 2.0.
- **Image Transformation (i.e. Image Augmentation)** (notebook). Apply various image augmentation techniques with TensorFlow 2.0, to generate distorted images for training.

TensorFlow v1

The tutorial index for TF v1 is available here: TensorFlow v1.15 Examples. Or see below for a list of the examples.

Dataset

Some examples require MNIST dataset for training and testing. Don't worry, this dataset will automatically be downloaded when running examples. MNIST is a database of handwritten digits, for a quick description of that dataset, you can check this notebook.

Official Website: http://yann.lecun.com/exdb/mnist/.

Installation

To download all the examples, simply clone this repository:

```
git clone https://github.com/aymericdamien/TensorFlow-Examples
```

To run them, you also need the latest version of TensorFlow. To install it:

```
pip install tensorflow
```

or (with GPU support):

```
pip install tensorflow_gpu
```

For more details about TensorFlow installation, you can check TensorFlow Installation Guide

TensorFlow v1 Examples - Index

The tutorial index for TF v1 is available here: TensorFlow v1.15 Examples.

0 - Prerequisite

- Introduction to Machine Learning.
- Introduction to MNIST Dataset.

1 - Introduction

- **Hello World** (notebook) (code). Very simple example to learn how to print "hello world" using TensorFlow.

- **Basic Operations** (notebook) (code). A simple example that cover TensorFlow basic operations.
- **TensorFlow Eager API basics** (notebook) (code). Get started with TensorFlow's Eager API.

2 - Basic Models

- **Linear Regression** (notebook) (code). Implement a Linear Regression with TensorFlow.
- **Linear Regression (eager api)** (notebook) (code). Implement a Linear Regression using TensorFlow's Eager API.
- **Logistic Regression** (notebook) (code). Implement a Logistic Regression with TensorFlow.
- **Logistic Regression (eager api)** (notebook) (code). Implement a Logistic Regression using TensorFlow's Eager API.
- **Nearest Neighbor** (notebook) (code). Implement Nearest Neighbor algorithm with TensorFlow.
- **K-Means** (notebook) (code). Build a K-Means classifier with TensorFlow.
- **Random Forest** (notebook) (code). Build a Random Forest classifier with TensorFlow.
- **Gradient Boosted Decision Tree (GBDT)** (notebook) (code). Build a Gradient Boosted Decision Tree (GBDT) with TensorFlow.
- **Word2Vec (Word Embedding)** (notebook) (code). Build a Word Embedding Model (Word2Vec) from Wikipedia data, with TensorFlow.

3 - Neural Networks

Supervised

- **Simple Neural Network** (notebook) (code). Build a simple neural network (a.k.a Multi-layer Perceptron) to classify MNIST digits dataset. Raw TensorFlow implementation.
- **Simple Neural Network (tf.layers/estimator api)** (notebook) (code). Use TensorFlow 'layers' and 'estimator' API to build a simple neural network (a.k.a Multi-layer Perceptron) to classify MNIST digits dataset.
- **Simple Neural Network (eager api)** (notebook) (code). Use TensorFlow Eager API to build a simple neural network (a.k.a Multi-layer Perceptron) to classify MNIST digits dataset.

- **Convolutional Neural Network** (notebook) (code). Build a convolutional neural network to classify MNIST digits dataset. Raw TensorFlow implementation.
- **Convolutional Neural Network (tf.layers/estimator api)** (notebook) (code). Use TensorFlow 'layers' and 'estimator' API to build a convolutional neural network to classify MNIST digits dataset.
- **Recurrent Neural Network (LSTM)** (notebook) (code). Build a recurrent neural network (LSTM) to classify MNIST digits dataset.
- **Bi-directional Recurrent Neural Network (LSTM)** (notebook) (code). Build a bi-directional recurrent neural network (LSTM) to classify MNIST digits dataset.
- **Dynamic Recurrent Neural Network (LSTM)** (notebook) (code). Build a recurrent neural network (LSTM) that performs dynamic calculation to classify sequences of different length.

Unsupervised

- **Auto-Encoder** (notebook) (code). Build an auto-encoder to encode an image to a lower dimension and re-construct it.
- **Variational Auto-Encoder** (notebook) (code). Build a variational auto-encoder (VAE), to encode and generate images from noise.
- **GAN (Generative Adversarial Networks)** (notebook) (code). Build a Generative Adversarial Network (GAN) to generate images from noise.
- **DCGAN (Deep Convolutional Generative Adversarial Networks)** (notebook) (code). Build a Deep Convolutional Generative Adversarial Network (DCGAN) to generate images from noise.

4 - Utilities

- **Save and Restore a model** (notebook) (code). Save and Restore a model with TensorFlow.
- **Tensorboard - Graph and loss visualization** (notebook) (code). Use Tensorboard to visualize the computation Graph and plot the loss.
- **Tensorboard - Advanced visualization** (notebook) (code). Going deeper into Tensorboard; visualize the variables, gradients, and more...

5 - Data Management

- **Build an image dataset** (notebook) (code). Build your own images dataset with TensorFlow data queues, from image folders or a dataset file.

- **TensorFlow Dataset API** (notebook) (code). Introducing TensorFlow Dataset API for optimizing the input data pipeline.
- **Load and Parse data** (notebook). Build efficient data pipeline (Numpy arrays, Images, CSV files, custom data, ...).
- **Build and Load TFRecords** (notebook). Convert data into TFRecords format, and load them.
- **Image Transformation (i.e. Image Augmentation)** (notebook). Apply various image augmentation techniques, to generate distorted images for training.

6 - Multi GPU

- **Basic Operations on multi-GPU** (notebook) (code). A simple example to introduce multi-GPU in TensorFlow.
- **Train a Neural Network on multi-GPU** (notebook) (code). A clear and simple TensorFlow implementation to train a convolutional neural network on multiple GPUs.