Python 3.7

Deep Learning Models

A collection of various deep learning architectures, models, and tips for TensorFlow and PyTorch in Jupyter Notebooks.

Traditional Machine Learning

- Perceptron
  [TensorFlow 1: GitHub | Nbviewer]
  [PyTorch: GitHub | Nbviewer]
- Logistic Regression
  [TensorFlow 1: GitHub | Nbviewer]
  [PyTorch: GitHub | Nbviewer]
- Softmax Regression (Multinomial Logistic Regression)
  [TensorFlow 1: GitHub | Nbviewer]
  [PyTorch: GitHub | Nbviewer]
- Softmax Regression with MLxtend's plot_decision_regions on Iris
  [PyTorch: GitHub | Nbviewer]

Multilayer Perceptrons

- Multilayer Perceptron
  [TensorFlow 1: GitHub | Nbviewer]
  [PyTorch: GitHub | Nbviewer]

- Multilayer Perceptron with Dropout
  [TensorFlow 1: GitHub | Nbviewer]
  [PyTorch: GitHub | Nbviewer]
- Multilayer Perceptron with Batch Normalization
  [TensorFlow 1: GitHub | Nbviewer]
  [PyTorch: GitHub | Nbviewer]
- Multilayer Perceptron with Backpropagation from Scratch
  [TensorFlow 1: GitHub | Nbviewer]
  [PyTorch: GitHub | Nbviewer]

Convolutional Neural Networks

Basic

- Convolutional Neural Network
  [TensorFlow 1: GitHub | Nbviewer]
  [PyTorch: GitHub | Nbviewer]
- Convolutional Neural Network with He Initialization
  [PyTorch: GitHub | Nbviewer]

Concepts

- Replacing Fully-Connnected by Equivalent Convolutional Layers
  [PyTorch: GitHub | Nbviewer]

Fully Convolutional

- Fully Convolutional Neural Network
  [PyTorch: GitHub | Nbviewer]

LeNet

- LeNet-5 on MNIST
  [PyTorch: GitHub | Nbviewer]
- LeNet-5 on CIFAR-10
  [PyTorch: GitHub | Nbviewer]
- LeNet-5 on QuickDraw
  [PyTorch: GitHub | Nbviewer]

AlexNet

- AlexNet on CIFAR-10
  [PyTorch: GitHub | Nbviewer]

VGG

- Convolutional Neural Network VGG-16
  [TensorFlow 1: GitHub | Nbviewer]
  [PyTorch: GitHub | Nbviewer]
- VGG-16 Gender Classifier Trained on CelebA
  [PyTorch: GitHub | Nbviewer]
- VGG-16 Dogs vs Cats Classifier
  [PyTorch: GitHub | Nbviewer]
- Convolutional Neural Network VGG-19
  [PyTorch: GitHub | Nbviewer]

DenseNet

- DenseNet-121 Digit Classifier Trained on MNIST
  [PyTorch: GitHub | Nbviewer]
- DenseNet-121 Image Classifier Trained on CIFAR-10
  [PyTorch: GitHub | Nbviewer]

ResNet

- ResNet and Residual Blocks
  [PyTorch: GitHub | Nbviewer]
- ResNet-18 Digit Classifier Trained on MNIST
  [PyTorch: GitHub | Nbviewer]
- ResNet-18 Gender Classifier Trained on CelebA
  [PyTorch: GitHub | Nbviewer]
- ResNet-34 Digit Classifier Trained on MNIST
  [PyTorch: GitHub | Nbviewer]
- ResNet-34 Object Classifier Trained on QuickDraw
  [PyTorch: GitHub | Nbviewer]
- ResNet-34 Gender Classifier Trained on CelebA
  [PyTorch: GitHub | Nbviewer]
- ResNet-50 Digit Classifier Trained on MNIST
  [PyTorch: GitHub | Nbviewer]
- ResNet-50 Gender Classifier Trained on CelebA
  [PyTorch: GitHub | Nbviewer]

- ResNet-101 Gender Classifier Trained on CelebA
  [PyTorch: GitHub | Nbviewer]
- ResNet-101 Trained on CIFAR-10
  [PyTorch: GitHub | Nbviewer]
- ResNet-152 Gender Classifier Trained on CelebA
  [PyTorch: GitHub | Nbviewer]

Network in Network

- Network in Network CIFAR-10 Classifier
  [PyTorch: GitHub | Nbviewer]

Normalization Layers

- BatchNorm before and after Activation for Network-in-Network CIFAR-10 Classifier
  [PyTorch: GitHub | Nbviewer]
- Filter Response Normalization for Network-in-Network CIFAR-10 Classifier
  [PyTorch: GitHub | Nbviewer]

Metric Learning

- Siamese Network with Multilayer Perceptrons
  [TensorFlow 1: GitHub | Nbviewer]

Autoencoders

Fully-connected Autoencoders

- Autoencoder (MNIST)
  [TensorFlow 1: GitHub | Nbviewer]
  [PyTorch: GitHub | Nbviewer]
- Autoencoder (MNIST) + Scikit-Learn Random Forest Classifier
  [TensorFlow 1: GitHub | Nbviewer]
  [PyTorch: GitHub | Nbviewer]

Convolutional Autoencoders

- Convolutional Autoencoder with Deconvolutions / Transposed Convolutions
  [TensorFlow 1: GitHub | Nbviewer]
  [PyTorch: GitHub | Nbviewer]

- Convolutional Autoencoder with Deconvolutions and Continuous Jaccard Distance
  [PyTorch: GitHub | Nbviewer]
- Convolutional Autoencoder with Deconvolutions (without pooling operations)
  [PyTorch: GitHub | Nbviewer]
- Convolutional Autoencoder with Nearest-neighbor Interpolation
  [TensorFlow 1: GitHub | Nbviewer]
  [PyTorch: GitHub | Nbviewer]
- Convolutional Autoencoder with Nearest-neighbor Interpolation -- Trained on CelebA
  [PyTorch: GitHub | Nbviewer]
- Convolutional Autoencoder with Nearest-neighbor Interpolation -- Trained on Quickdraw
  [PyTorch: GitHub | Nbviewer]

Variational Autoencoders

- Variational Autoencoder
  [PyTorch: GitHub | Nbviewer]
- Convolutional Variational Autoencoder
  [PyTorch: GitHub | Nbviewer]

Conditional Variational Autoencoders

- Conditional Variational Autoencoder (with labels in reconstruction loss)
  [PyTorch: GitHub | Nbviewer]
- Conditional Variational Autoencoder (without labels in reconstruction loss)
  [PyTorch: GitHub | Nbviewer]
- Convolutional Conditional Variational Autoencoder (with labels in reconstruction loss)
  [PyTorch: GitHub | Nbviewer]
- Convolutional Conditional Variational Autoencoder (without labels in reconstruction loss)
  [PyTorch: GitHub | Nbviewer]

Generative Adversarial Networks (GANs)

- Fully Connected GAN on MNIST
  [TensorFlow 1: GitHub | Nbviewer]
  [PyTorch: GitHub | Nbviewer]
- Fully Connected Wasserstein GAN on MNIST
  [PyTorch: GitHub | Nbviewer]
- Convolutional GAN on MNIST
  [TensorFlow 1: GitHub | Nbviewer]

[PyTorch: GitHub | Nbviewer]

- Convolutional GAN on MNIST with Label Smoothing
  [TensorFlow 1: GitHub | Nbviewer]
  [PyTorch: GitHub | Nbviewer]
- Convolutional Wasserstein GAN on MNIST
  [PyTorch: GitHub | Nbviewer]
- "Deep Convolutional GAN" (DCGAN) on Cats and Dogs Images
  [PyTorch: GitHub | Nbviewer]
- "Deep Convolutional GAN" (DCGAN) on CelebA Face Images
  [PyTorch: GitHub | Nbviewer]

Graph Neural Networks (GNNs)

- Most Basic Graph Neural Network with Gaussian Filter on MNIST
  [PyTorch: GitHub | Nbviewer]
- Basic Graph Neural Network with Edge Prediction on MNIST
  [PyTorch: GitHub | Nbviewer]
- Basic Graph Neural Network with Spectral Graph Convolution on MNIST
  [PyTorch: GitHub | Nbviewer]

Recurrent Neural Networks (RNNs)

Many-to-one: Sentiment Analysis / Classification

- A simple single-layer RNN (IMDB)
  [PyTorch: GitHub | Nbviewer]
- A simple single-layer RNN with packed sequences to ignore padding characters (IMDB)
  [PyTorch: GitHub | Nbviewer]
- RNN with LSTM cells (IMDB)
  [PyTorch: GitHub | Nbviewer]
- RNN with LSTM cells (IMDB) and pre-trained GloVe word vectors
  [PyTorch: GitHub | Nbviewer]
- RNN with LSTM cells and Own Dataset in CSV Format (IMDB)
  [PyTorch: GitHub | Nbviewer]
- RNN with GRU cells (IMDB)
  [PyTorch: GitHub | Nbviewer]
- Multilayer bi-directional RNN (IMDB)
  [PyTorch: GitHub | Nbviewer]

- Bidirectional Multi-layer RNN with LSTM with Own Dataset in CSV Format (AG News)
  [PyTorch: GitHub | Nbviewer]

Many-to-Many / Sequence-to-Sequence

- A simple character RNN to generate new text (Charles Dickens)
  [PyTorch: GitHub | Nbviewer]

Ordinal Regression

- Ordinal Regression CNN -- CORAL w. ResNet34 on AFAD-Lite
  [PyTorch: GitHub | Nbviewer]
- Ordinal Regression CNN -- Niu et al. 2016 w. ResNet34 on AFAD-Lite
  [PyTorch: GitHub | Nbviewer]
- Ordinal Regression CNN -- Beckham and Pal 2016 w. ResNet34 on AFAD-Lite
  [PyTorch: GitHub | Nbviewer]

Tips and Tricks

- Cyclical Learning Rate
  [PyTorch: GitHub | Nbviewer]
- Annealing with Increasing the Batch Size (w. CIFAR-10 & AlexNet)
  [PyTorch: GitHub | Nbviewer]
- Gradient Clipping (w. MLP on MNIST)
  [PyTorch: GitHub | Nbviewer]

Transfer Learning

- Transfer Learning Example (VGG16 pre-trained on ImageNet for Cifar-10)
  [PyTorch: GitHub | Nbviewer]

Visualization and Interpretation

- Vanilla Loss Gradient (wrt Inputs) Visualization (Based on a VGG16 Convolutional Neural Network for Kaggle's Cats and Dogs Images)
  [PyTorch: GitHub | Nbviewer]
- Guided Backpropagation (Based on a VGG16 Convolutional Neural Network for Kaggle's Cats and Dogs Images)
  [PyTorch: GitHub | Nbviewer]

PyTorch Workflows and Mechanics

Custom Datasets

- Custom Data Loader Example for PNG Files
  [PyTorch: GitHub | Nbviewer]
- Using PyTorch Dataset Loading Utilities for Custom Datasets -- CSV files converted to HDF5
  [PyTorch: GitHub | Nbviewer]
- Using PyTorch Dataset Loading Utilities for Custom Datasets -- Face Images from CelebA
  [PyTorch: GitHub | Nbviewer]
- Using PyTorch Dataset Loading Utilities for Custom Datasets -- Drawings from Quickdraw
  [PyTorch: GitHub | Nbviewer]
- Using PyTorch Dataset Loading Utilities for Custom Datasets -- Drawings from the Street View House Number (SVHN) Dataset
  [PyTorch: GitHub | Nbviewer]
- Using PyTorch Dataset Loading Utilities for Custom Datasets -- Asian Face Dataset (AFAD)
  [PyTorch: GitHub | Nbviewer]
- Using PyTorch Dataset Loading Utilities for Custom Datasets -- Dating Historical Color Images
  [PyTorch: GitHub | Nbviewer]
- Using PyTorch Dataset Loading Utilities for Custom Datasets -- Fashion MNIST
  [PyTorch: GitHub | Nbviewer]

Training and Preprocessing

- Generating Validation Set Splits
  [PyTorch: GitHub | Nbviewer]
- Dataloading with Pinned Memory
  [PyTorch: GitHub | Nbviewer]
- Standardizing Images
  [PyTorch: GitHub | Nbviewer]
- Image Transformation Examples
  [PyTorch: GitHub | Nbviewer]
- Char-RNN with Own Text File
  [PyTorch: GitHub | Nbviewer]
- Sentiment Classification RNN with Own CSV File
  [PyTorch: GitHub | Nbviewer]

Parallel Computing

- Using Multiple GPUs with DataParallel -- VGG-16 Gender Classifier on CelebA
  [PyTorch: GitHub | Nbviewer]

Other

- Sequential API and hooks
  [PyTorch: GitHub | Nbviewer]
- Weight Sharing Within a Layer
  [PyTorch: GitHub | Nbviewer]
- Plotting Live Training Performance in Jupyter Notebooks with just Matplotlib
  [PyTorch: GitHub | Nbviewer]

Autograd

- Getting Gradients of an Intermediate Variable in PyTorch
  [PyTorch: GitHub | Nbviewer]

TensorFlow Workflows and Mechanics

Custom Datasets

- Chunking an Image Dataset for Minibatch Training using NumPy NPZ Archives
  [TensorFlow 1: GitHub | Nbviewer]
- Storing an Image Dataset for Minibatch Training using HDF5
  [TensorFlow 1: GitHub | Nbviewer]
- Using Input Pipelines to Read Data from TFRecords Files
  [TensorFlow 1: GitHub | Nbviewer]
- Using Queue Runners to Feed Images Directly from Disk
  [TensorFlow 1: GitHub | Nbviewer]
- Using TensorFlow's Dataset API
  [TensorFlow 1: GitHub | Nbviewer]

Training and Preprocessing

- Saving and Loading Trained Models -- from TensorFlow Checkpoint Files and NumPy NPZ Archives
  [TensorFlow 1: GitHub | Nbviewer]