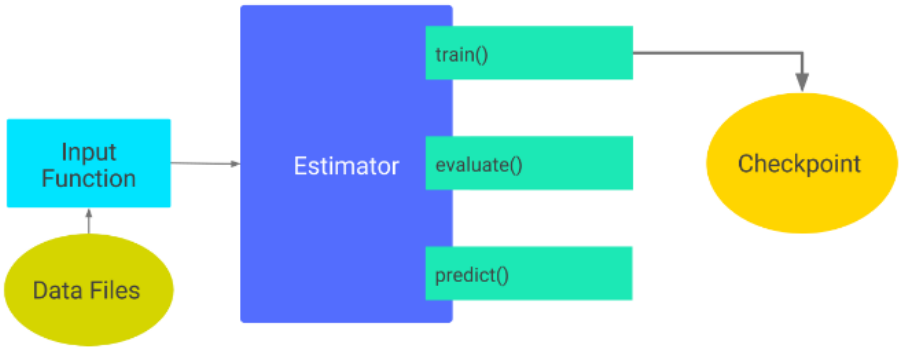
1) Installed TensorFlow without GPU support due to difficulties to access CODA
    a. Requires its own python environment, as it is likely to interfere with normal python environment.
        i. Experiencing severe difficulties and had to do several system wipes for everything to work out.
    b. In conclusion: Make sure it is separated from everything else!
        i. Considering separate system partition for specialized Linux installation just to cater to this software
2) Ran the TensorFlow tutorials.
    a. Getting started for ML Beginners
        i. Neural Networks – Increase number of hidden layers and neurons creates a more powerful model, but requires more data to train effectively.
        ii. Train the model
        iii. Estimator class – Two types.
            1. Premade estimators – Used DNN – Numerical feature columns.
            2. Custom estimators – partially self coded.
        iv. Evaluate
        v. Predicting
    b. Feature Columns
        i. Numeric columns
        ii. Bucketized columns – different weights, richer model, added flexibility to learn
        iii. Categorical identity columns
        iv. Hashed column – Specifying number of categories
        v. Crossed columns – Feature crossing, model learns separate weights for each combination of features
    c. Indicator and Embedding columns – Embedding columns increase your models capabilities.
    d. Getting started with TensorFlow
        i. Explaining the iris flower case.
    e. Checkpoints – Save and restore TensorFlow models built with Estimators.
        i. Checkpoints, which is a format dependent on the code that created the model
        ii. SavedModel, which is a format independent of the code that created the model

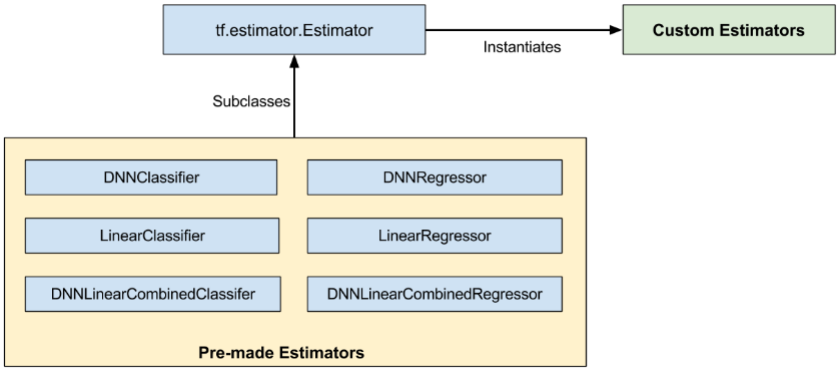


The first call to train().

        iii.
    f. Datasets – Working with data

i. Arguments
ii. Slices
iii. Manipulation
iv. Return
v. Reading CSV files
vi. Build the dataset
vii. Build a csv line parser
viii. Parse the lines

g. Creating custom estimators

i. 

3) Started the real tutorials.
   a. Quickly did the image recognition script. Disappointing result outside of imagenet database, 2% accuracy on unknow image
   b. Building a convolutional neural network for recognition of handwriting
      i. Input layer
      ii. Convolutional layer#1
      iii. Pooling layer #1
      iv. Convolutional layer#2
      v. Pooling layer #2
      vi. Dense layer
      vii. Logits layer
      viii. General predictions
      ix. Calculate loss
      x. Configure the training op
      xi. Add evaluation metrics.
      xii. Load training and test data.
      xiii. Create the estimatorsSet up a logging hook
      xiv. Train the model
      xv. Evaluate the model.
   c. Retrain Inceptions Final Layer for New Categories tutorial – Image recognition technique where previously learned models are transferred to new classes, thus shortening the image recognition process.
4) Research on Raster Vision (Developed by Azavea).
   a. Project holds a set of scripts for training and running object detection models on aerial and satellite imagery. Traditional imagery is in PNG files. Aerial imagery are usually large GeoTiff files with many objects. Annotations and predictions are represented in geospatial coordinates using GeoJSON files.
   b. TensorFlow Object Detection API provides good core object detection functionality using deep learning but cannot natively handle satellite and

aerial imagery. The module Rasterio is used to convert data into and out of a form that the TensorFlow API can handle.

c. For prediction: Images are split into small “chips”. Then it makes predictions on the chips and aggregate the predictions back together.
d. For training: Images are cropped out around objects, to create training chips.