Recap from the previous lecture

- Intro to Machine Learning
- Terminology
- Matrix representation
- Loss and Mean Square Error
- Gradient Descent
- Perceptron algorithm
- Classification example using scikit-learn
Tensor Definition

• Tensors are a generalization of matrices and arrays
• A vector (1-D) is a first order tensor
• A matrix (2-D) is a second order tensor

TensorFlow

• TensorFlow is an open-source library for high performance numerical computation
• It was originally developed by Google Brain team
• It supports machine learning and deep learning
• It provides a variety of toolkits that allows development at different levels of abstraction¹

https://www.tensorflow.org/

¹ One can pick a high level toolkit and start developing without knowing all the underlying details.
TensorFlow Hierarchy

- TensorFlow Estimators: High-level, object-oriented API
- `tf.layers`, `tf.losses`, `tf.metrics`: Reusable libraries for common model components
- Python TensorFlow: Provides Ops, which wrap C++ Kernels
- C++ TensorFlow: Kernels work on one or more platforms

Figure 1. TensorFlow toolkit hierarchy.

Source: https://developers.google.com/machine-learning/crash-course/first-steps-with-tensorflow/toolkit

---

TensorFlow “Hello, world!”

```python
import tensorflow as tf

# Set up a linear classifier.
classifier = tf.estimator.LinearClassifier(feature_columns)

# Train the model on some example data.
classifier.train(input_fn=train_input_fn, steps=2000)

# Use it to predict.
predictions = classifier.predict(input_fn=predict_input_fn)
```
Installation

• Python PIP
  $ pip install tensorflow
• Anaconda Package Manager
  $ conda install -c conda-forge tensorflow

Demo

• We will build a linear regression model using TensorFlow using the `LinearRegressor` class
• The dataset is based on the California census data from 1990
• We will build a correlation matrix to better understand our data
• We will evaluate our model using the Root Mean Squared Error (RMSE)
• Modify the parameters of our model to improve accuracy