Hands on Machine Learning with Scikit-Learn &TensolFlow

Chapter 9

Up and Running with TensorFlow

Created by Yusuke FUJIMOTO

はじめに

- この資料は「Hands-On Machine Learning with Scikit-Learn and TensorFlow - O'Reilly Media」を読んだ際の（主にソースコードに関する）簡単な解説を残したものです。
- 全部を解説したわけではないので注意
- 余裕があればソースコード周りの背景知識もまとめたい
- 何かあったら yukkyo12221222@gmail.com まで

Chapter 9

Up and Running with TensorFlow

ポイント

- TensorFlow
  - google 製の計算ライブラリ（機械学習に限らない）
- 実行時は、主に以下の2ステップある
  - 計算グラフ（≒設計図）を作る
  - 計算を実行する
- テンソル（多次元配列）計算を扱う
  - 例：0次元テンソル＝スカラー、1次元テンソル＝ベクトル、2次元テンソル=行列、etc...

- なんで計算グラフ作るの？直接計算すれば良くない？
  - ニューラルネットワークの学習時は、微分が相互に影響し合う。それを手で全部書き下すのは厳しい
-

TF の関数メモ

```
tf.variable()   # 変数定義
tf.constant()   # 定数定義
tf.transpose()  # 転置
tf.matmul()     # 行列としてのかけざん
tf.random_uniform()   # 乱数生成, np.rand() 同じように使える
tf.assign()     # 値を書き換える（更新する）
tf.reduce_mean()   # 平均を求める（meanではないので注意）
```

Implementing Grandient Descent

- 勾配法を tensorflow で実装する。以下の2通りで。
  - 手で勾配を計算する
  - 自動で tensorflow に微分させる

```
error = y_pred - y
mse = tf.reduce_mean(tf.square(error), name="mse")
grad = 2/m * tf.matmul(tf.transpose(X), error)
training_op = tf.assign(theta,
                        theta - learning_rate * grad)
```

自動微分させる場合は以下のようにしてできる

```
gradients = tf.gradients(mse, [theta])[0]
```

Technique	勾配を計算するのにグラフを横切る数？	精度	その他
Numerical diff	$n_{inputs} + 1$	Low	実装が楽
Symbolic diff	N/A	High	色んなグラフ作れる
Forward-mode autodiff	n_{inputs}	High	Uses *dual numbers*
Reverse-mode autodiff	$n_{outputs} + 1$	High	TF で実装されてる

Using an Optimizer

勾配法以外にも学習（パラメータの更新）ができる

``

Exercises

- 省略。やらないとまずい

参考サイト

- PCAの最終形態GPLVMの解説
- 計算グラフの微積分：バックプロパゲーションを理解する | コンピュータサイエンス | POSTD