

萩原研 プログラミング研修

3日目：tensorflowによる実践演習

今週は、機械学習で使用されることが多いtensorflowを用いて演習を行っていきます。

今日の流れはざっとこんな感じです。

1. tensorflowの使い方を理解する。
2. 次に、画像処理における代表的な手法のCNNを実装する。
3. 次に、言語処理における代表的な手法のLSTMを実装する。

1. tensorflow

tensorflowを用いると順伝播計算を記述するだけで、誤差逆伝播計算を自動で行ってくれる。

このためプログラマーが行う処理の流れは以下のようになる。

1. 変数定義
2. 順伝播計算の定義
3. 誤差関数の定義
4. 最適化手法の定義
5. セッションの定義（毎回同じ文言を書くだけ）
6. 入力データを1へ入力し,4で定義した最適化手法を実行

例)784次元の入力画像から10次元の出力を得る単層ニューラルネット

1. 入力xと教師データt（外部入力による定数）、
重みwとバイアスb（変数）を定義。

外部入力による定数はplaceholderで、

変数はVariableで定義する。

```
x = tf.placeholder(tf.float32, [None, 784])
t = tf.placeholder(tf.float32, [None, 10])
w = tf.Variable(tf.zeros([784, 10]))
b = tf.Variable(tf.zeros([10]))
```

2. 出力yに至る計算過程を記述

```
u = tf.matmul(x,w)+b
y = tf.nn.softmax(u)
```

3. 教師データtと出力yを用いて、誤差関数lossを定義（ここではログ交差エントロピーを使用）

```
loss = -tf.reduce_sum(t * tf.log(y))
```

4. 最適化手法train_stepを定義

```
train_step = tf.train.AdamOptimizer().minimize(loss)
```

5. 以下を書くだけ

```
sess = tf.InteractiveSession()  
sess.run(tf.initialize_all_variables())
```

6. 4のtrain_stepを何度も実行

```
sess.run(train_step, feed_dict={x: 入力バッチ, t: 教師バッチ})
```

注意点.

tensorflowによって定義した変数を出力したい場合は下記のようにし、
sess.run()を実行しなくてはならない。ここでは、第1引数の値を求めるのに必要な外部入力を
feed_dict={}によって示す。

下記の例で、wは外部入力がなくとも値を持つためfeeddictは不要

yを計算するには入力データが必要なため、feeddictを用いてxに実際の入力を入れる。

```
sess.run(w)  
sess.run(y, feed_dict={x: 入力バッチ})
```

演習1-1. MNISTの分類

784-500-10のニューラルネットを学習してください。

- バッチサイズ（学習時に使用するデータ数）：100
 - 学習に使用するデータは訓練データ60000枚から毎回ランダムに100枚選択
- エポック数（データセットを何周学習させるか）：100
- 重みの初期値：ガウシアン初期化(平均0, 標準偏差0.1)
- バイアスの初期値：0
- SGDの学習係数：0.0001

ヒント.

入力データに対する正答率を知りたいときは以下の式を（5.）より前に書きsess.run()で呼び出す。

```
correct_prediction = tf.equal(tf.argmax(p, 1), tf.argmax(t, 1))  
accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))
```

例題では中間層が無いニューラルネットの場合を考えた。

784-500-10のように中間層を追加するには順伝播計算の部分を書き換える。