Save Model

MokkeMeguru[1]

2020-02-12 Wed

[1]meguru.mokke@gmail.com

Outline

Presentaion agenda

モデルを保存する

推論をしたい&訓練を継続したい
⇒ モデルの重みを保存

モデルを保存する: コード

```
1   # save model
2   saver = tf.train.Saver() # モデル保存のためのオブジェクト
3   save_path = "./tmp/sgd_save"
4   ckpt = tf.train.get_checkpoint_state(save_path)
5
6   with tf.Session() as sess:
7       if ckpt:
8           print('restore variable')
9           last_model = ckpt.model_checkpoint_path
10          # 保存されたモデルの取り出し
11          saver.restore(sess, last_model)
12      else:
13          # 保存されてなければ初期化
14          sess.run(init_op)
15      for step in range(500):
16          # ...
17          if step % 100 == 0:
18              # ...
19              # モデルの保存
20              saver.save(sess, save_path + "/model.ckpt",
21                         global_step=global_step)
```

モデルを保存する: 出力

```
pipenv run python save_model.py --task training
# a - 0 / b - 0/loss - 81.5751572 / global_step - 0
# a - 0.943696678 / b - 1.84093821/loss - 43.5924454 / global_step - 100
# a - 1.7148298 / b - 3.28322673/loss - 48.596 / global_step - 200
# a - 2.35165763 / b - 4.41729975/loss - 15.8343315 / global_step - 300
# a - 2.79292226 / b - 5.27085066/loss - 7.97473288 / global_step - 400
ls -la tmp/sgd_save/
# ...
# checkpoint
# ...
# model.ckpt-401.data-00000-of-00001
# model.ckpt-401.index
# model.ckpt-401.meta
```

訓練の再開

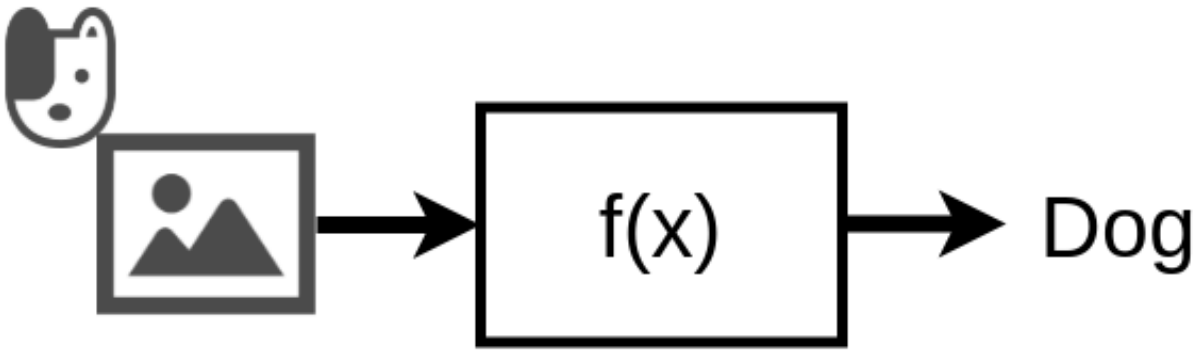
```
pipenv run python save_model.py --task training
# restore variable
# a - 2.79316497 / b - 5.27649879/loss - 19.928009 / global_step - 401
# a - 3.14751577 / b - 5.94900417/loss - 14.7261763 / global_step - 501
# a - 3.46371508 / b - 6.48496056/loss - 7.34989548 / global_step - 601
# a - 3.68623185 / b - 6.88744593/loss - 2.83048487 / global_step - 701
# a - 3.86673498 / b - 7.20387936/loss - 2.55219769 / global_step - 801
ls -la tmp/sgd_save/
# ...
# checkpoint
# ...
# model.ckpt-802.data-00000-of-00001
# model.ckpt-802.index
# model.ckpt-802.meta
```

Presentaion agenda

推論

得られたモデルの重みを用いて何らかを入力し、何かを得ること

ex1. 画像を入力して、その画像のカテゴリ (犬 / 猫) を得る
ex2. ランダムな値を入れて、適当な画像を得る



推論: コード I

```
1   def inference(sess):
2       # Interactive Session : 入力などを受け付ける必要がある場合に用いる Session
3       sess = tf.InteractiveSession()
4       with tf.variable_scope('inputs'):
5           x = tf.placeholder(dtype=tf.float32, shape=[])
6           y = tf.placeholder(dtype=tf.float32, shape=[])
7
8       # setup model
9       y_hat, log_op = simple_model(x)
10
11      # save model
12      saver = tf.train.Saver()
13      save_path = "./tmp/sgd_save"
14      ckpt = tf.train.get_checkpoint_state(save_path)
15
16      # load checkpoint
17      if ckpt:
18          print('restore variable')
19          last_model = ckpt.model_checkpoint_path
20          saver.restore(sess, last_model)
21      else:
22          # 訓練済みモデルがなければエラー
23          raise Exception('for inference, we need trained model')
```

推論: コード II

```
24
25      while True:
26          # input
27          input_x = input('-->')
28          print('input:', input_x)
29          if not input_x.isdigit():
30              break
31          input_x = int(input_x)
32          evaled_y_hat = sess.run([y_hat], feed_dict={x: input_x})
33          print('output:', evaled_y_hat)
34      sess.close()
```

推論: 出力

```
pipenv run python save_model.py --task inference
# -->2
# input: 2
# output: [17.979458]
# -->10
# input: 10
# output: [57.866394]
# -->12
# input: 12
# output: [67.838135]
# -->8
# input: 8
# output: [47.89466]
# --> quit
# input: quit
```

Presentaion agenda

課題

1. save_model.py を用いて訓練してみなさい
   複数回コードを実行し、loss が下がっていることを確認しなさい
2. save_model.py を次のように変更し訓練を行ってみなさい
   行なった結果の損失の変化の差を観察しなさい
   - モデルの保存先を `./tmp/sgd_save2/` にしなさい。
   - モデルの最適化関数を `GradientDescentOptimizer` から `AdamOptimizer` に変更しなさい。