

# Practice of Tensorflow on Pai

# Setup Environment

- use daily environment (e.g., laptop)

## 1. download odps console

- `wget http://odps.alibaba-inc.com/official\_downloads/odpscmd/0.23.1/odps\_clt\_release\_64.tar.gz`

## 2. config your odps account

- `project_name=search_kg`
- `access_id = xxx`
- `access_key=xxx`
- `end_point=http://service-corp.odps.aliyun-inc.com/api`

## 3. config oss account

- `ossmd config --host=oss-cn-hangzhou-zmf.aliyuncs.com --id=LTAiNCeWlvNFRVws --key=U3LavOsucE7O3dI0vsg2cxze2TJqtu`

# Hello World

- prepare script “hello\_world.py”
- run command
  - `pai -name tensorflow100 -Dscript=file:///home/lisheng.ls/pai-tensorflow/test/hello_world.py`

# Data

- ODPS Table
- Volume (support only read mode)
- OSS (support both read and write mode)
  - checkpoint model

# Read Data

- **assume data is already exists**

- use osscmd upload your data

- **small data usage**

- **large-scale data**

- use tfreader api

- **demo**

- read\_oss.py

```
from tensorflow.python.lib.io import file_io
# from tensorflow.python.platform import gfile
data = file_io.read_file_to_string(file_path)

# with gfile.GFile(file_path) as f:
#     for line in f:
#         other code
```

# Parameters

参数名称	参数描述	参数值格式
script	必选, 本地TF算法文件	"file:///path/to/*.py" 或者: "project_name/resources/resource_name"
volumes	可选, 输入volume, 可多个, 以逗号隔开	"odps://prj_name/volumes/volume_name" "odps://prj_name/volumes/volume_name/part_name"
buckets	可选, 输入bucket, 可多个, 以逗号隔开, 每个bucket须以"/"结尾	"oss://bucket_name/?role_arn=xxx&host=yyy" "oss://bucket_1/?role_arn=xxx&host=yyy,oss://bucket_2/"
tables	可选, 输入表, 可多个, 以逗号隔开	"odps://prj_name/tables/table_name"
gpuRequired	可选, 标识是否使用GPU	yes/200
checkpointDir	可选, TF checkpoint目录	"oss://bucket_name/?role_arn=xxx&host=yyy"

```
-Dbuckets="oss://83267/?role_arn=acs:ram::1504587816182874:role/search-alinlp-oss-visit&host=oss-cn-hangzhou-zmf.aliyuncs.com"
```

```
pai -name tensorflow010 -Dbuckets="oss://83267/neural/data/?role_arn=acs:ram::1504587816182874:role/search-alinlp-oss-visit&host=oss-cn-hangzhou-zmf.aliyuncs.com" -Dscript="file:///home/lisheng.ls/pai-tensorflow/neural-responding/neural.tar.gz" -DentryFile="neural_responding.py" -DcheckpointDir="oss://83267/neural/models/?role_arn=acs:ram::1504587816182874:role/search-alinlp-oss-visit&host=oss-cn-hangzhou-zmf.aliyuncs.com";
```

# Save & Restore

```
# Once in a while, we save checkpoint, print statistics
if step % FLAGS.steps_per_checkpoint == 0:
    perplexity = math.exp(float(loss)) if loss < 300 else float("inf")
    print "global step %d learning rate %.4f perplexity %.2f" %
        model.global_step.eval(), model._lr_rate.eval(), perplexity
    # Save checkpoint and zero loss.
    # checkpoint_path = os.path.join(FLAGS.checkpointDir, "neural.ckpt")
    checkpoint_path = FLAGS.checkpointDir + "neural-responding-beam.ckpt"
    saver.save(sess, checkpoint_path, global_step=model.global_step)
    loss = 0.0
    sys.stdout.flush()
```

```
with tf.Session() as sess:
    model = seq2seq_attention_model.Seq2SeqAttentionModel(hps)
    model.build_graph()
    """Restore a checkpoint and decode it."""
    ckpt_state = tf.train.get_checkpoint_state(FLAGS.checkpointDir)
    if not ckpt_state:
        print "no saved model in dir[%s]" % FLAGS.checkpointDir
        return
    ckpt_path = os.path.join(
        FLAGS.checkpointDir, os.path.basename(ckpt_state.model_checkpoint_path)
    )
    saver = tf.train.Saver()
    saver.restore(sess, ckpt_path)
```

# Q&A

- <http://gitlab.alibaba-inc.com/algo/pai-tensorflow-doc>
- <https://web.stanford.edu/class/cs20si/>
- <https://github.com/oxford-cs-deepnlp-2017/lectures>