平台技术说明

1. 技术流程图



说明：

①mnist\_data\_steup.py 将原始影像数据转成hdfs数据，并存储在hdfs

create\_pickle\_train(image\_path, mask\_path, img\_pixel=10, channels=3)

“””

:remark 影像转成array

:param image\_path: 影像路径

:param mask\_path: 对应的影像掩膜路径

:param img\_pixel: 图像大小（如：mnist 28）

:param channels: 图像波段数（如：mnist 波段为 1）

:return: array

“””

def writeMNIST(sc, dir\_name,dir\_name1,img\_pixel,channels, output, format, num\_partitions): '''

:remark 将图像数据转成需要的格式 存储成HDFS

:param sc: SparkContext

:param dir\_name: 输入存放影像路径

:param dir\_name1: 输入的影像对应的掩膜路径

:param img\_pixel: 图像大小（如：mnist 28）

:param channels: 图像波段数（如：mnist 波段为 1）

:param output: 转成HDFS输出路径

:param format: 需要转成的数据格式（csv,pickle,tfr）

:param num\_partitions: 实际图像分类数 mnist 10分类 所有为10

:return: HDFS

'''

②mnist\_spark.py 分配hdfs上的数据到各个工作节点上

不需修改

③mnist\_dist.py 各节点执行该程序得到最后的模型

只需按模型模板编写即可

1. 提交命令

cd /home/install/git

①数据转换

spark-submit \

--master yarn \

--deploy-mode cluster \

--queue default \

--num-executors 8 \

--executor-memory 32G \

--driver-memory 6G \

--conf spark.dynamicAllocation.enabled=false \

--conf spark.yarn.maxAppAttempts=1 \

--archives hdfs://dm01-08-01.tjidc.dcos.com:8020/user/root/data/mnist.zip#mnist \

--jars hdfs://dm01-08-01.tjidc.dcos.com:8020/spark-tensorflow/spark-tensorflow-connector-1.0-SNAPSHOT.jar \

myfloder/Arable/mnist\_data\_setup.py \

--imgPixel 10 \

--channels 3 \

--output hdfs://dm01-08-01.tjidc.dcos.com:8020/user/root/data/pickle \

--format pickle

②train

hdfs dfs -rmr model\_arable1

hdfs dfs -mkdir model\_arable1

spark-submit \

--master yarn \

--deploy-mode cluster \

--queue default \

--num-executors 8 \

--executor-memory 32G \

--driver-memory 6G \

--py-files TensorFlowOnSpark/tfspark.zip,myfloder/Arable/mnist\_dist.py \

--conf spark.dynamicAllocation.enabled=false \

--conf spark.yarn.maxAppAttempts=1 \

--conf spark.yarn.executor.memoryOverhead=12288 \

myfloder/Arable/mnist\_spark.py  \

--images hdfs://dm01-08-01.tjidc.dcos.com:8020/user/root/data/pickle/train1/images \

--labels hdfs://dm01-08-01.tjidc.dcos.com:8020/user/root/data/pickle/train1/labels \

--format pickle \

--mode train \

--model hdfs://dm01-08-01.tjidc.dcos.com:8020/user/root/model\_arable \

--epochs 1 \

--steps 10000 \

--acc 0.7 \

--dropout 0.5 \

--learning\_rate 1e-6 \

--model\_name model.ckpt

③ retrain

spark-submit \

--master yarn \

--deploy-mode cluster \

--queue default \

--num-executors 8 \

--executor-memory 32G \

--driver-memory 6G \

--py-files TensorFlowOnSpark/tfspark.zip,myfloder/Arable/mnist\_dist.py \

--conf spark.dynamicAllocation.enabled=false \

--conf spark.yarn.maxAppAttempts=1 \

--conf spark.yarn.executor.memoryOverhead=12288 \

myfloder/Arable/mnist\_spark.py  \

--images hdfs://dm01-08-01.tjidc.dcos.com:8020/user/root/data/pickle/train2/images \

--labels hdfs://dm01-08-01.tjidc.dcos.com:8020/user/root/data/pickle/train2/labels \

--format pickle \

--mode retrain \

--model hdfs://dm01-08-01.tjidc.dcos.com:8020/user/root/model\_arable \

--epochs 1 \

--steps 100000 \

--acc 0.9 \

--dropout 0.3 \

--learning\_rate 1e-8 \

--model\_name model.ckpt

④inference

spark-submit \

--master yarn \

--deploy-mode cluster \

--queue default \

--num-executors 8 \

--executor-memory 32G \

--driver-memory 6G \

--py-files TensorFlowOnSpark/tfspark.zip,myfloder/Arable/mnist\_dist.py \

--conf spark.dynamicAllocation.enabled=false \

--conf spark.yarn.maxAppAttempts=1 \

--conf spark.yarn.executor.memoryOverhead=12288 \

myfloder/Arable/mnist\_spark.py  \

--images hdfs://dm01-08-01.tjidc.dcos.com:8020/user/root/data/pickle/test/images \

--labels hdfs://dm01-08-01.tjidc.dcos.com:8020/user/root/data/pickle/test/labels \

--format pickle \

--mode inference \

--model model\_arable \

--output predictions