Using BigDL in Telefonica Open Cloud

Issue 01
Date 2018/3/19
## Contents

1 Overview ................................................................................................................................. 1  
  1.1 BigDL Overview ................................................................................................................ 1  

2 Getting Started ...................................................................................................................... 3  
  2.1 Provision BigDL VM .......................................................................................................... 3  
  2.2 Run BigDL examples .......................................................................................................... 4  
  2.3 Conclusion ......................................................................................................................... 6
1.1 BigDL Overview

BigDL is a distributed deep learning framework for Apache Spark that was developed by Intel and contributed to the open source community with licensed under the Apache 2.0. As the following diagram shows, BigDL is implemented as a library on top of Spark, users can write their deep learning applications as standard Spark programs, which can directly run on top of existing Spark or Hadoop clusters.

Figure 1-1 BigDL in the Apache Spark* stack

BigDL applies the following scenarios if you want to:

- Analyze a large amount of data on the same big data Spark cluster on which the data reside (in, say, HDFS, Apache HBase*, or Hive);

- Add deep learning functionality (either training or prediction) to your big data (Spark) programs or workflow;

- Use existing Hadoop/Spark clusters to run your deep learning applications, which you can then easily share with other workloads (e.g., extract-transform-load, data warehouse, feature engineering, classical machine learning, graph analytics). An undesirable alternative to using BigDL is to introduce yet another distributed framework alongside Spark just to implement deep learning algorithms.
This article gives a demo for end users how to create a development and test environment based on an image, which is embedded with BigDL software and other depended packages. On this environment, end users could test their developed models. If the models work correctly, end users could deploy the models in production in the further step, such as in MRS cluster.
2.1 Provision BigDL VM

In Telefonica OpenCloud, we have released a public image named by “Ubuntu 16.04 for BigDL”. It’s an image built based on Ubuntu 16.04 64bit. Inside the image we have embedded with a lot of software or packages depended by BigDL, such as:

* JDK (1.8.0_92)
* Scala (2.11.8)
* Jupyter (1.0.0)
* bigdl distribution (0.4.0)
* TensorBoard 1.0.0a6
* BigDL-Tutorials-branch-0.4
* MXNet (1.0.0)
* Spark (2.1.1)
* MKL (2018.0.1)
* git (2.15.0)
* python 2.7.14
* pip
* numpy
* scipy
* pandas
* scikit-learn
* matplotlib
* seaborn
* wordcloud
* Anaconda (5.0.1)
Therefore you don’t have to install them manually. Normally we will update packages this image once half a year.

Before you start, you need to provision an ECS VM by visiting ECS console and following the directions in the wizard:

![ECS console screenshot]

Notes: for your convenience during usage, we suggest you assign an EIP to the VM for accessing it afterwards. And we propose the minimal flavor for this VM to be 2vCPU and 4GB memory.

2.2 Run BigDL examples

After the VM is provisioned successfully, you could access it by SSH using xShell or Putty with private key for linux account. Switch to root account using:

```
sudo su - root
```

You could see some environment variables have been loaded from /root/.bashrc:

```
export BIGDL_HOME=/opt/anaconda/lib/python2.7/site-packages/bigdl
export SPARK_HOME=/opt/anaconda/lib/python2.7/site-packages/pyspark
export BIGDL_MASTER_HOME=/opt/intel-analytics/BigDL-master
export JAVA_HOME=/opt/anaconda2/pkgs/java-jdk-8.0.92-1
export PATH="/opt/anaconda2/bin:$PATH;$JAVA_HOME/bin:/opt/apache-maven-3.5.2/bin"
export SPARK_DRIVER_MEMORY=2g
```

And we have already uploaded BigDL source code and BigDL Tutorials to path /opt/intel-analytics:
Run the notebook by:

```
root@bigdl:/opt/intel-analytics# jupyter notebook
--notebook-dir=/opt/intel-analytics/BigDL-Tutorials-branch-0.4 --ip=* --no-browser --allow-root
```

```
[W 16:20:57.485 NotebookApp] WARNING: The notebook server is listening on all IP addresses and not using encryption. This is not recommended.

[I 16:20:57.519 NotebookApp] JupyterLab alpha preview extension loaded from /opt/anaconda2/lib/python2.7/site-packages/jupyterlab

JupyterLab v0.27.0

Known labextensions:

[I 16:20:57.521 NotebookApp] Running the core application with no additional extensions or settings

[I 16:20:57.527 NotebookApp] Serving notebooks from local directory: /opt/intel-analytics/BigDL-Tutorials-branch-0.4

[I 16:20:57.527 NotebookApp] 0 active kernels

[I 16:20:57.527 NotebookApp] The Jupyter Notebook is running at: http://[all ip addresses on your system]:8888/?token=fc741974a1c21d74c18436cd9956eb9081ce32e3f39c2a84

[I 16:20:57.527 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).

[C 16:20:57.528 NotebookApp]

Copy/paste this URL into your browser when you connect for the first time, to login with a token:

http://localhost:8888/?token=fc741974a1c21d74c18436cd9956eb9081ce32e3f39c2a84

Last line shows the URL to visit notebook by browser:

```
jupyter
```

Navigate to example http://[EIP]:8888/notebooks/notebooks/neural_networks/rnn.ipynb

Run the notebook. It will take a few minutes, and in the end you will get prediction results and see a loss graph like this:
2.3 Conclusion

In this article we demonstrated an example for how to test models of BigDL in Telefonica Open Cloud. It’s an example to use MNIST dataset to train a multi-layer feed forward neural network. You can get the learn details from: http://yann.lecun.com/exdb/mnist/.

The other useful links for reference:

https://bigdl-project.github.io/0.4.0/index.html

https://github.com/intel-analytics/BigDL/
https://github.com/intel-analytics/BigDL-Tutorials