Lab. Installation of TensorFlow GPU Version

1. 서론
   1)Windows 10에서 TensorFlow GPU(tensorflow-gpu)를 설치해보자.
   2)NVIDIA driver를 설치하고, Anaconda를 기반으로 GPU를 지원하는 TensorFlow version을 설치히보자.
   3)CUDA 및 cuDNN library 등은 Anaconda 환경에서 한꺼번에 설치되게 된다.
   4)Windows 10 버전을 기준으로 설명하지만, 다른 Windows 버전에서도 크게 다르지 않을 것으로 생각된다.
   5)만약 우분투 버전에서 GPU가 지원되는 Tensorflow를 설치하고자 하시는 경우
   https://hiseon.me/data-analytics/tensorflow/tensorflow-install/의 글을 참고할 것.

2. Windows nvidia driver 설치
   1)GPU를 지원하는 TensorFlow를 설치하기 전에, 가장 먼저 NVIDA driver를 설치해 주어야 한다.
   2)NVIDIA driver가 설치되지 않았을 경우, 아래의 페이지에서 GPU model에 맞는 version의 driver를 download 받는다.

   https://www.nvidia.co.kr/Download/index.aspx?lang=kr

   3)혹은 이미 설치되어 있는 경우에는 driver를 update해 준다.
   4)Download 받은 설치 파일을 실행하여, NVIDIA driver 설치를 진행한다.
   5)nvidia driver가 성공적으로 설치 되었을 경우, nvidia-smi 명령어를 이용하여 GPU 정보및 process의 사용률 등을 확인 할 수 있다.

   C:\Program Files\NVIDIA Corporation\NVSMI

   6)위의 위치에서 nvidia-smi 명령 실행한다.

```
Thu Sep 12 15:00:00 2019
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 398.36                 Driver Version: 398.36                    |
|-------------------------------+----------------------+----------------------+
| GPU  Name            TCC/WDDM | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  GeForce GTX 860M   WDDM  | 00000000:01:00.0 Off |                  N/A |
| N/A   48C    P8    N/A /  N/A |     40MiB /  4096MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

3. Python 및 TensorFlow 패키지 종속성 설치
   1)Python 3.6.x version을 설치한다. 설치할 때 PATH 환경변수에 등록한다.

```
python -V
---------------------------
```

```
    Python 3.6.8

  2)관리자 권한의 Command창에서 다음의 종속 모듈을 설치한다.

    C:\>pip3 install six numpy wheel
    C:\>pip3 install keras_applications==1.0.6 --no-deps
    C:\>pip3 install keras_preprocessing==1.0.5 --no-deps


4. Bazel 설치
  1)TensorFlow를 컴파일하는 데 사용되는 빌드 도구인 Bazel 0.23.0을 설치한다.
  2)이것은 C++를 빌드하도록 Bazel을 설정한다.
  3)Bazel 실행 파일의 위치를 %PATH% 환경 변수에 추가한다.
  4)Bazel 0.23.0 또는 이전 버전을 설치한다.

    https://docs.bazel.build/versions/master/install-windows.html

  5)Install the prerequisites
    -Visual C++ Redistributable for Visual Studio 2015
    -MSYS2 x86_64

  6)먼저 Visual C++ Redistributable for Visual Studio 2015를 설치한다.
    -vc_redist.x64.exe을 설치한다.

  7)Install MSYS2
    -https://www.msys2.org/
    - or
    -https://sourceforge.net/projects/msys2/files/Base/x86_64/msys2-x86_64-20190524.exe/download
    -Download msys2-x86_64-20190524.exe
    -설치 후 마지막 화면에서 [Run MSYS2 now] check 후 [Finish]를 클릭
    -Update the package database and core system packages with:
      $pacman -Syu
      :: Synchronizing package databases...
       mingw32               554.2 KiB   993K/s 00:01 [######################] 100%
       mingw32.sig           119.0   B  58.1K/s 00:00 [######################] 100%
       mingw64               556.0 KiB  1549K/s 00:00 [######################] 100%
       mingw64.sig           119.0   B  0.00B/s 00:00 [######################] 100%
       msys                  185.3 KiB  12.1M/s 00:00 [######################] 100%
       msys.sig              119.0   B  0.00B/s 00:00 [######################] 100%
      :: Starting core system upgrade...
      warning: terminate other MSYS2 programs before proceeding
      resolving dependencies...
      looking for conflicting packages...

      Packages (3) mintty-1~3.0.2-1  msys2-runtime-3.0.7-6  pacman-5.1.3-3

      Total Download Size:   17.33 MiB
      Total Installed Size:  60.77 MiB
      Net Upgrade Size:       0.77 MiB

      :: Proceed with installation? [Y/n]

    -y를 눌러서 계속 진행한다.
```

```
    -설치가 끝나면 창을 닫는다.
    -시작 > 프로그램에서 MSYS2 64bit > MSYS2 MINGW 64-bit
    -Open the MSYS2 terminal, and run this command:
      Instructor@DESKTOP-NU7GQVV MINGW64 ~
      $ pacman -S zip unzip patch diffutils git
      resolving dependencies...
      looking for conflicting packages...

      Packages (39) db-5.3.28-2  expat-2.2.7-1  gdbm-1.18.1-2  heimdal-7.7.0-1
                    libgdbm-1.18.1-2  openssh-8.0p1-1  perl-5.30.0-1
                    perl-Authen-SASL-2.16-2  perl-Convert-BinHex-1.125-1
                    perl-Encode-Locale-1.05-1  perl-Error-0.17027-1
                    perl-File-Listing-6.04-2  perl-HTML-Parser-3.72-4
                    perl-HTML-Tagset-3.20-2  perl-HTTP-Cookies-6.04-1
                    perl-HTTP-Daemon-6.01-2  perl-HTTP-Date-6.02-2
                    perl-HTTP-Message-6.18-1  perl-HTTP-Negotiate-6.01-2
                    perl-IO-Socket-SSL-2.066-1  perl-IO-stringy-2.111-1
                    perl-LWP-MediaTypes-6.02-2  perl-MIME-tools-5.509-1
                    perl-MailTools-2.21-1  perl-Net-HTTP-6.19-1
                    perl-Net-SMTP-SSL-1.04-1  perl-Net-SSLeay-1.85-3
                    perl-TermReadKey-2.37-4  perl-TimeDate-2.30-2
                    perl-Try-Tiny-0.30-1  perl-URI-1.76-1  perl-WWW-RobotRules-6.02-2
                    perl-libwww-6.39-1  vim-8.1.1777-1  diffutils-3.7-1  git-2.22.0-1
                    patch-2.7.6-1  unzip-6.0-2  zip-3.0-3

      Total Download Size:    29.35 MiB
      Total Installed Size:  140.11 MiB

      :: Proceed with installation? [Y/n] y


  8)Download Bazel
    -https://github.com/bazelbuild/bazel/releases/tag/0.24.1
    -bazel-0.24.1-windows-x86_64.zip(86.2MB)

  9)Bazel 설치
    -Java JDK 8 or later Installation
    -msys64 설치폴더가 'c:\msys64' 인 경우 'c:\msys64/usr/bin' 을 환경변수 PATH 에 추가.
    -다 되고나면, where msys-2.0.dll 를 수행해서 제대로 path 처리됐는지 확인
      where msys-2.0.dll
      ---------------------------------------
      C:\msys64\usr\bin\msys-2.0.dll

    -Download받은 파일이 압축을 풀고 C:/bazel 폴더를 만들고 그 안에 bazel.exe를 놓는다.
    -Bazel 실행 파일의 위치를 %PATH% 환경 변수에 추가한다.
      --C:\bazel


5. TensorFlow Source Code Download
  -C:/Downloads> bash

    Instructor@DESKTOP-NU7GQVV MINGW64 ~
    # cd /c/Downloads
```

157 Instructor@DESKTOP-NU7GQVV /c/Downloads
158 # git clone https://github.com/tensorflow/tensorflow.git
159
160
161 6. CUDA 10.0 설치
162 1)https://developer.nvidia.com/cuda-10.0-download-archive
163 2)Operating System : Windows
164 3)Architercture : x86_64
165 4)Version : 10
166 5)Installer Type(설치유형) : exe(local)
167 6)cuda_10.0.130_411.31_win10.exe(2.1GB)
168 7)다운받은 파일을 double-click하여 실행한다.
169 8)화면의 명령에 따라 진행한다.
170
171
172 7. cuDNN 7 Installation
173 1)https://developer.nvidia.com/rdp/cudnn-archive
174 2)위에서 CUDA를 10.0을 설치했기 때문에 이 버전에 맞는 cuDNN을 다운로드해야 한다.
175 -Download cuDNN v7.6.2 (July 22, 2019), for CUDA 10.0
176 -cuDNN Library for Windows 10
177 -javaexpert@nate.com/P@$$W0rdfornvidia
178 -cudnn-10.0-windows10-x64-v7.6.2.24.zip
179
180 3)압축을 푼다.
181 -cudnn-10.0-windows10-x64-v7.6.2.24\cuda\bin\cudnn64_7.dll을 다음 폴더에 복사한다.
182 --C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.0\bin
183 -cudnn-10.0-windows10-x64-v7.6.2.24\cuda\include\cudnn.h를 다음의 폴더에 복사한다.
184 --C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.0\include
185 -C:\Downloads\cudnn-10.0-windows10-x64-v7.6.2.24\cuda\lib\x64\cudnn.lib를 다음에 폴더에 복사한다.
186 --C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.0\lib\x64
187
188
189 8. TensorFlow Installation
190 # cd tensorflow
191 Instructor@DESKTOP-NU7GQVV /c/Downloads/tensorflow
192 # python ./configure.py
193
194 WARNING: --batch mode is deprecated. Please instead explicitly shut down your Bazel server using the command "bazel shutdown".
195 You have bazel 0.24.1 installed.
196 Please specify the location of python. [Default is C:\ProgramData\Anaconda3\python.exe]:
197
198
199 Found possible Python library paths:
200 C:\ProgramData\Anaconda3\lib\site-packages
201 Please input the desired Python library path to use. Default is [C:\ProgramData\Anaconda3\lib\site-packages]
202
203 Do you wish to build TensorFlow with XLA JIT support? [y/N]: N
204 No XLA JIT support will be enabled for TensorFlow.
205
206 Do you wish to build TensorFlow with ROCm support? [y/N]: N
207 No ROCm support will be enabled for TensorFlow.

```
208
209     Do you wish to build TensorFlow with CUDA support? [y/N]: Y
210     CUDA support will be enabled for TensorFlow.
211
212     Found CUDA 10.0 in:
213         C:/Program Files/NVIDIA GPU Computing Toolkit/CUDA/v10.0/lib/x64
214         C:/Program Files/NVIDIA GPU Computing Toolkit/CUDA/v10.0/include
215     Found cuDNN 7 in:
216         C:/Program Files/NVIDIA GPU Computing Toolkit/CUDA/v10.0/lib/x64
217         C:/Program Files/NVIDIA GPU Computing Toolkit/CUDA/v10.0/include
218
219
220     Please specify a list of comma-separated CUDA compute capabilities you want to build with.
221     You can find the compute capability of your device at: https://developer.nvidia.com/cuda-gpus.
222     Please note that each additional compute capability significantly increases your build time and
        binary size, and that TensorFlow only supports compute capabilities >= 3.5 [Default is:
        3.5,7.0]:
223
224
225     Please specify optimization flags to use during compilation when bazel option "--config=opt" is
        specified [Default is /arch:AVX]:
226
227
228     Would you like to override eigen strong inline for some C++ compilation to reduce the
        compilation time? [Y/n]:
229     Eigen strong inline overridden.
230
231     Preconfigured Bazel build configs. You can use any of the below by adding "--config=<>" to
        your build command. See .bazelrc for more details.
232         --config=mkl            # Build with MKL support.
233         --config=monolithic     # Config for mostly static monolithic build.
234         --config=ngraph         # Build with Intel nGraph support.
235         --config=numa           # Build with NUMA support.
236         --config=dynamic_kernels        # (Experimental) Build kernels into separate shared
            objects.
237         --config=v2             # Build TensorFlow 2.x instead of 1.x.
238     Preconfigured Bazel build configs to DISABLE default on features:
239         --config=noaws          # Disable AWS S3 filesystem support.
240         --config=nogcp          # Disable GCP support.
241         --config=nohdfs         # Disable HDFS support.
242         --config=nonccl         # Disable NVIDIA NCCL support.
243
244
245     #    bazel build --config=opt --config=cuda --define=no_tensorflow_py_deps=true
        //tensorflow/tools/pip_package:build_pip_package
246
247
248
249 9.  Anaconda Installation
250
251 10. Windows TensorFlow GPU 설치
252   1)Anaconda가 설치 완료될 경우, Anaconda Prompt 를 사용하거나 환경변수가 설정되었을 경우 PowerShell 등에
      서 conda 명령어를 사용할 수 있다.
```

2)conda 명령어가 사용이 가능한지 아래의 명령어로 버전을 출력한다.

```
conda --version
-----------------------------------------------
conda 4.7.10
```

3)Anaconda가 제대로 실행 될 경우, 아래의 명령어를 이용하여 GPU가 지원되는 Tensorflow를 설치한다.

```
conda install tensorflow-gpu
-----------------------------------------------------------------
C:\WINDOWS\system32>conda install tensorflow-gpu
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

 environment location: C:\ProgramData\Anaconda3

 added / updated specs:
  - tensorflow-gpu


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    _tflow_select-2.1.0        |              gpu           3 KB
    absl-py-0.7.1              |           py37_0         158 KB
    astor-0.8.0                |           py37_0          47 KB
    conda-4.7.11               |           py37_0         2.8 MB
    cudatoolkit-10.0.130       |                0       250.0 MB
    cudnn-7.6.0                |       cuda10.0_0       157.1 MB
    gast-0.2.2                 |           py37_0         155 KB
    grpcio-1.16.1              |   py37h351948d_1         850 KB
    keras-applications-1.0.8   |            py_0          33 KB
    keras-preprocessing-1.1.0  |            py_1          36 KB
    libprotobuf-3.8.0          |       h7bd577a_0         1.8 MB
    markdown-3.1.1             |           py37_0         132 KB
    protobuf-3.8.0             |   py37h33f27b4_0         527 KB
    tensorboard-1.14.0         |   py37he3c9ec2_0         3.1 MB
    tensorflow-1.14.0          |gpu_py37h5512b17_0          5 KB
    tensorflow-base-1.14.0     |gpu_py37h55fc52a_0       95.3 MB
    tensorflow-estimator-1.14.0|            py_0         291 KB
    tensorflow-gpu-1.14.0      |       h0d30ee6_0          3 KB
    termcolor-1.1.0            |           py37_1          8 KB
    ------------------------------------------------------------
                                           Total:       512.3 MB

The following NEW packages will be INSTALLED:

  _tflow_select      pkgs/main/win-64::_tflow_select-2.1.0-gpu
  absl-py            pkgs/main/win-64::absl-py-0.7.1-py37_0
  astor              pkgs/main/win-64::astor-0.8.0-py37_0
  cudatoolkit        pkgs/main/win-64::cudatoolkit-10.0.130-0
```

```
  cudnn              pkgs/main/win-64::cudnn-7.6.0-cuda10.0_0
  gast               pkgs/main/win-64::gast-0.2.2-py37_0
  grpcio             pkgs/main/win-64::grpcio-1.16.1-py37h351948d_1
  keras-applications pkgs/main/noarch::keras-applications-1.0.8-py_0
  keras-preprocessi~ pkgs/main/noarch::keras-preprocessing-1.1.0-py_1
  libprotobuf        pkgs/main/win-64::libprotobuf-3.8.0-h7bd577a_0
  markdown           pkgs/main/win-64::markdown-3.1.1-py37_0
  protobuf           pkgs/main/win-64::protobuf-3.8.0-py37h33f27b4_0
  tensorboard        pkgs/main/win-64::tensorboard-1.14.0-py37he3c9ec2_0
  tensorflow         pkgs/main/win-64::tensorflow-1.14.0-gpu_py37h5512b17_0
  tensorflow-base    pkgs/main/win-64::tensorflow-base-1.14.0-gpu_py37h55fc52a_0
  tensorflow-estima~ pkgs/main/noarch::tensorflow-estimator-1.14.0-py_0
  tensorflow-gpu     pkgs/main/win-64::tensorflow-gpu-1.14.0-h0d30ee6_0
  termcolor          pkgs/main/win-64::termcolor-1.1.0-py37_1

The following packages will be UPDATED:

  conda                                 4.7.10-py37_0 --> 4.7.11-py37_0


Proceed ([y]/n)?
```

4)Windows에서 설치시 항상 관리자권한의 Command에서 실행해야 한다.

5)위의 설치 과정을 보면 tensorflow-gpu 패키지가 필요로 하는 CUDA (cudatoolkit) 과 cuDNN 라이브러리 등이 자동으로 함께 설치되는 것을 알 수 있다.

6)'y'를 입력하여 설치를 진행한다.

11. tensorflow-gpu package 설치 확인

1)제대로 Tensorflow가 설치되었는지 확인하기 위해서는 아래와 같은 명령어를 입력한다.

```
>>> import tensorflow as tf
>>> print(tf.__version__)
1.14.0
```