# GPU parallel computing for machine learning in Python

how to build a parallel computer

Yoshiyasu Takefuji

# GPU parallel computing for machine learning in Python
a step-by-step approach to building a GPU parallel computer

by Yoshiyasu Takefuji

Contents

Introduction
Chapter 1 How to build a GPU parallel computer
Chapter 2 How to install GPU software for machine learning
Chapter 2.1 GPU parallel machine based on Ubuntu16.04
- CUDA installation
- Miniconda installation
- virtualenv installation
- cudnnlib installation
- keras (machine learning framework)
- pytorch (machine learning framework)
- chainer (machine learning framework)

Chapter 2.2 GPU parallel machine based on Windows 10
- cygwin installation
- CUDA installation
- virtualenv installation
- keras installation

## Introduction

This book illustrates how to build a GPU parallel computer. If you don't want to waste your time for building, you can buy a built-in-GPU desktop/laptop machine. All you need to do is to install GPU-enabled software for parallel computing. Imagine that we are in the midst of a parallel computing era. The GPU parallel computer is suitable for machine learning, deep (neural network) learning. For example, GeForce GTX1080 Ti is a GPU board with 3584 CUDA cores. Using the GeForce GTX1080 Ti, the performance is roughly 20 times faster than that of an INTEL i7 quad-core CPU. We have benchmarked the MNIST hand-written digits recognition problem (60,000 persons: hand-written digits from 0 to 9). The result of MNIST benchmark for machine learning shows that GPU of a single GeForce GTX1080 Ti board takes only less than 48 seconds while the INTEL i7 quad-core CPU requires 15 minutes and 42 seconds.

A CUDA core is most commonly referring to the single-precision floating point units in an SM (streaming multiprocessor). A CUDA core can initiate one single precision floating point instruction per clock cycle. CUDA is a parallel computing platform and application programming interface (API) model created by Nvidia. It allows software developers and software engineers to use a CUDA-enabled graphics processing unit (GPU) for general purpose processing. The GPU parallel computer is based on SIMD ( single instruction, multiple data) computing.

The first GPU for neural networks was used by Kyoung-Su Oh, et al. for image processing published in 2004 (1). A minimum GPU parallel computer is composed of a CPU board and a GPU board.

This book contains the important issue on which CPU/GPU board you should buy and also illustrates how to integrate them in a single box by considering the heat problem. The power consumption of GPU is so large that we should take care of the temperature and heat from the GPU board in the single box. Our goal is to have the faster parallel computer with lower power dissipation.

Software installation is another critical issue for machine learning in Python. Two operating system examples including Ubuntu16.04 and Windows 10 system will be described. This book shows how to install CUDA and cudnnlib in two operating systems. Three frameworks including pytorch, keras, and chainer for machine learning on CUDA and cudnnlib will be introduced.  Matching problems between operating system (Ubuntu, Windows 10), library (CUDA, cudnnlib), and machine learning framework (pytorch, keras, chainer) are discussed.

# Chapter 1 How to build a GPU parallel computer

In order to build a GPU parallel computer, it is easy to use a **barebones PC.** According to Wikipedia, the barebones PC is a **computer** that has minimal components. A typical **barebones system** includes a case, motherboard, CPU, hard drive, RAM, and power supply. Most **barebones** systems are sold as kits, in which the components must be assembled by the user.

We have investigated barebones PCs available in the market. We have decided to use an inexpensive barebones PC, shuttle SZ170R8V2 which can accommodate a NVIDIA GTX 1080 Ti and an INTEL i7 quad-core CPU with 500W power supply as shown in Fig. 1.1

Fig. 11 Shuttle SZ170R8V2 (barebones PC)

Fig. 2 shows a Shuttle SZ170R8V2 motherboard. Shuttle SZ170R8V2 motherboard can accommodate a CPU (i7 7700 quad-core) and 4 memory slots where 4 memory cards (DDR4 PC4-17000 16GB) are used. INTEL Core i7 7700 CPU should be chosen instead of INTEL Core i7 7700k. The power consumption of the maximum FPU load of INTEL Core i7 7700 CPU is 65W [1] while that of INTEL Core i7 7700k CPU is 95W [2]. i7-7700k supports overclocking designated by the "K" suffix. In order to reduce the power consumption, INTEL Core i7 7700 CPU is selected. INTEL Core i7 7700 CPU should be inserted to the center of the motherboard as shown in Fig. 1.2.

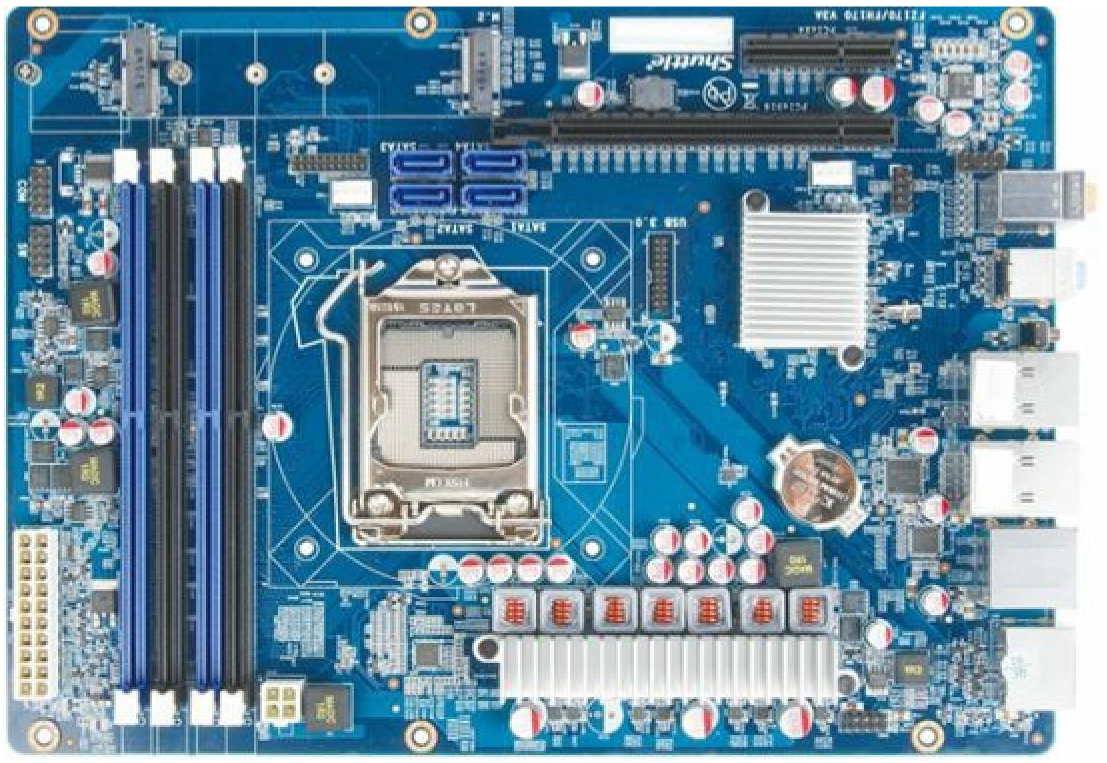
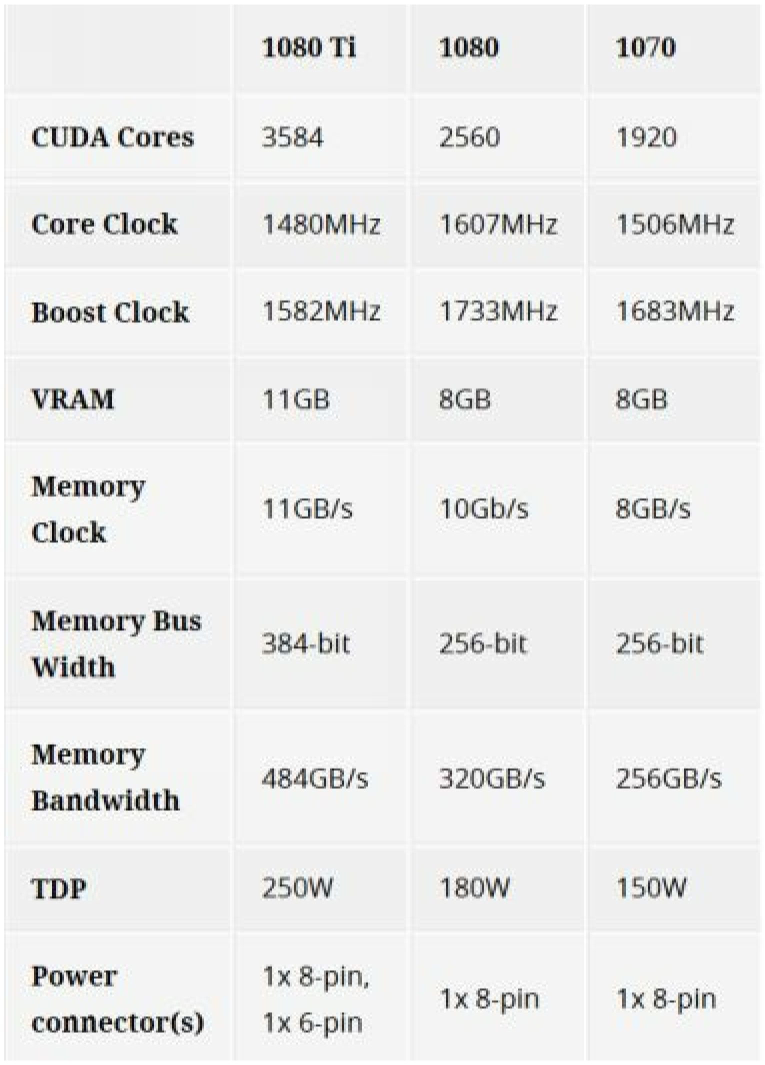


Fig. 1.2 Shuttle SZ170R8V2 motherboard

A minimum GPU parallel computer is composed of a motherboard with a CPU, a GPU board, and SSD or Hard disk storages. We use a 3.5 inch 2TB hard disk. We may replace the hard disk with SSD. Because of the BIOS problem, your SSD may not be recognized by the current BIOS of the motherboard. It is safer to use the cheap hard disk instead of the expensive SSD.

The latest GPU board of NVIDIA GeForce GTX 1080 Ti has 3584 CUDA cores. GeForce GTX has several series: GTX 1070, GTX 1080, and GTX 1080 Ti. Table 1 shows the detailed specification of NVIDIA GeForce GXT boards..

Table 1 Specification of NVIDIA GPU GTX boards



| | 1080 Ti | 1080 | 1070 |
|---|---|---|---|
| **CUDA Cores** | 3584 | 2560 | 1920 |
| **Core Clock** | 1480MHz | 1607MHz | 1506MHz |
| **Boost Clock** | 1582MHz | 1733MHz | 1683MHz |
| **VRAM** | 11GB | 8GB | 8GB |
| **Memory Clock** | 11GB/s | 10Gb/s | 8GB/s |
| **Memory Bus Width** | 384-bit | 256-bit | 256-bit |
| **Memory Bandwidth** | 484GB/s | 320GB/s | 256GB/s |
| **TDP** | 250W | 180W | 150W |
| **Power connector(s)** | 1x 8-pin, 1x 6-pin | 1x 8-pin | 1x 8-pin |

You should be aware of the power consumption of every GPU board. For example, a single NVIDIA GeForce GTX 1080 Ti board requires 600W recommended by NVIDIA instead of 250W. However, the peak power is around 267W for NVIDIA GTX 1080 Ti where the maximum GPU temperature is 91 degree Celcius [3]. In order to use the NVIDIA GeForce GTX 1080 Ti board, six-pin/eight-pin PCI Express connectors are used as shown in Fig. 1.3.
.

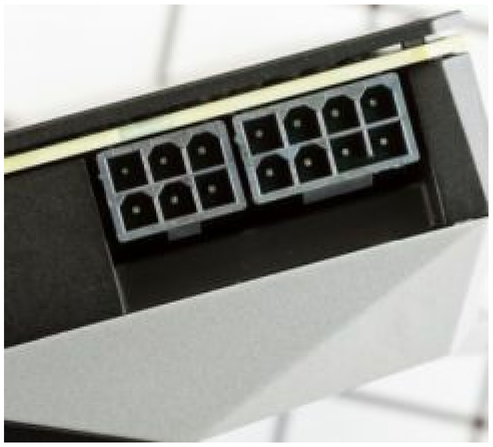


Fig 1.3 PCI Express connector
left: singular six-pin PCI Express connector
right: singular eight-pin PCI Express connector

Once you build the Shuttle PC with INTEL Core i7 7700 CPU installed in the CPU socket, 4x16GB memory cards inserted, and the hard disk, you should add a NVIDIA GeForce GTX 1080 Ti board. You must download the latest Ubuntu 16.04 image from one of Ubuntu archive sites in your country. As of today (2017.6.16), ubuntu-16.04.2-desktop-amd64.iso should be burned to a DVD media.

In order to see what is going on in software installation, you need to use a HDMI display, a keyboard, and a mouse. Connect the DVD player with Ubuntu 16.04 media to the motherboard by a USB cable. Turn on the power of the motherboard. Follow the instruction of Ubuntu installation. Once you setup the IP address of the GPU parallel machine, you don't need the HDMI display. In order to use remote-login, you must install openssh-server by:

```
$ sudo apt install openssh-server.
```

You should update and upgrade the Ubuntu system by:

```
$ sudo apt update
$ sudo apt upgrade
```

It is important to install vncserver on the GPU parallel machine. To install vncserver, you should type:

```
$ sudo apt install tightvncserver
```

Once you installed tightvncserver, run tightvncserver

```
$ vncserver
```

.vnc directory will be created

```
$ cd .vnc
```

You should edit xstartup file by using nano or vi editor.
cat command shows the content of xstartup file.

```
$ cat xstartup
#!/bin/sh
MODE="GNOME"
unset SESSION_MANAGER
metacity &
gnome-settings-daemon &
gnome-panel &
```

To kill tightvncserver, type

```
$ vncserver -kill :1
```

Note that 1 indicates tightvncserver process number.

To run tightvncserver again, type

```
$ vncserver
```

In order to remote login to the GPU parallel machine from the remote machine, you should install tightvncviewer.
To install tightvncviewer, download tightviewer from the following site:
http://www.tightvnc.com/

.

References:
1. http://www.tomshardware.com/reviews/intel-kaby-lake-core-i7-7700k-i7-7700-i5-7600k-i5-7600,4870-9.html
2. http://www.tomshardware.com/reviews/intel-kaby-lake-core-i7-7700k-i7-7700-i5-7600k-i5-7600,4870-8.html
3.https://www.techpowerup.com/reviews/NVIDIA/GeForce_GTX_1080_Ti/2

# Chapter 2 How to install GPU software for machine learning

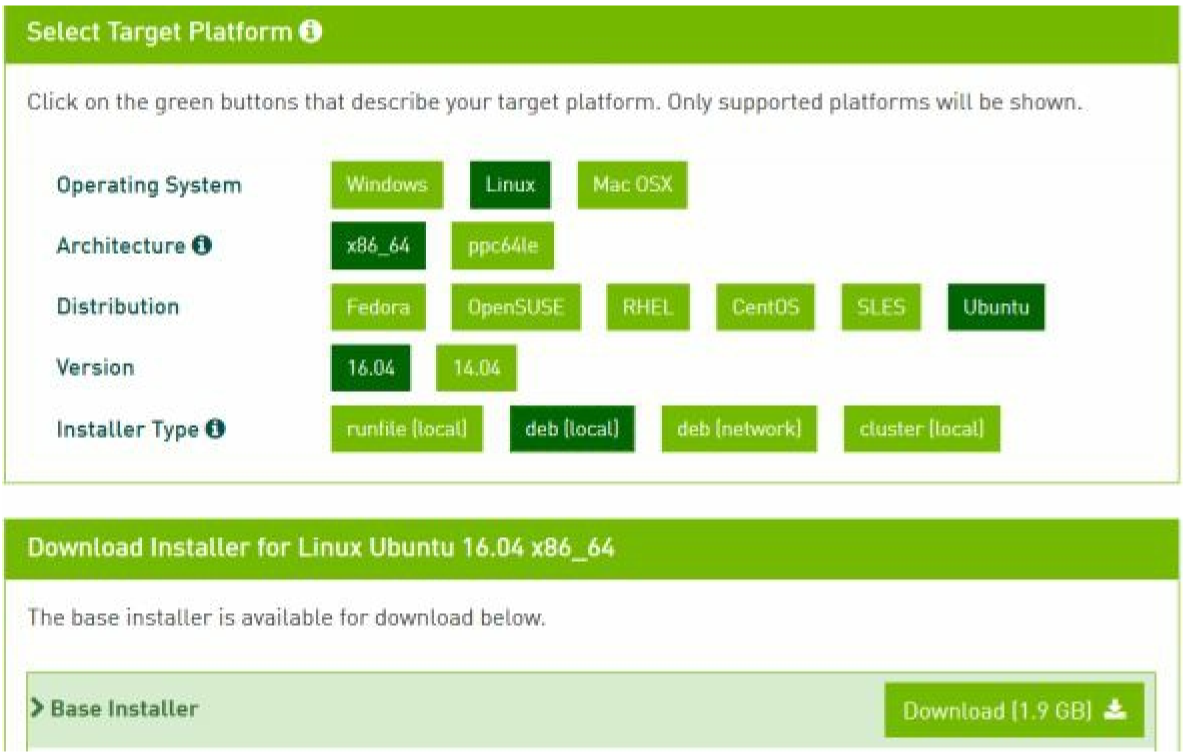
We need to install three kinds of software: CUDA, cudnnlib, and machine learning framework. In this book, two operating systems (Ubuntu 16.04 and Windows 10) are explained where different version of CUDA and cudnnlib should be installed depending on installed operating system. Three machine learning frameworks are introduced: pytorch, keras, and chainer. Chapter2.1 explains how to install GPU software on Ubuntu 16.04. Chapter2.2 shows how to install GPU software for Windows 10.

# Chapter 2.1 GPU parallel machine based on Ubuntu 16.04

In order to install CUDA software, you should access to the following site:
https://developer.nvidia.com/cuda-downloads

## CUDA installation

In Select Target Platform menu, pick "Linux", "x86_64", "Ubuntu", "16.04", "deb (local)" respectively. The downloaded file is 1.9GB size.



Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown.

| | | | | | | |
|---|---|---|---|---|---|---|
| Operating System | Windows | [✓ Linux] | Mac OSX | | | |
| Architecture | [✓ x86_64] | ppc64le | | | | |
| Distribution | Fedora | OpenSUSE | RHEL | CentOS | SLES | [✓ Ubuntu] |
| Version | [✓ 16.04] | 14.04 | | | | |
| Installer Type | runfile (local) | [✓ deb (local)] | deb (network) | cluster (local) | | |

Download Installer for Linux Ubuntu 16.04 x86_64

The base installer is available for download below.

> Base Installer — Download (1.9 GB)

Fig. 2.1.1 CUDA downloadable file for Ubuntu 16.04

Download the file, cuda-repo-ubuntu1604-8-0-local-ga2_8.0.61-1_amd64.deb

The following command, dpkg, can install cuda software.

```
$ sudo dpkg -i cuda-repo-ubuntu1604-8-0-local-ga2_8.0.61-1_amd64.deb
$ sudo apt update
$ sudo apt upgrade
$ sudo apt install cuda
```

## Miniconda installation

```
$ wget https://repo.continuum.io/miniconda/Miniconda2-latest-Linux-x86_64.sh
$ bash Miniconda2-latest-Linux-x86_64.sh
Welcome to Miniconda2 4.3.21 (by Continuum Analytics, Inc.)
In order to continue the installation process, please review the license
agreement.
Please, press ENTER to continue
>>>
```

```
$ conda update conda
```

## virtualenv installation

```
$ conda install virtualenv
```

Close terminal and reopen terminal

virtualenv command creates p27 environment (python2.7 environment)

```
$ virtualenv p27 -p $(which python)
```

## cudnnlib installation

In order to download cudnnlib, you have to access to the following site:
https://developer.nvidia.com/cudnn

This is very important to select the right cuDNN library.

Since we have installed CUDA8.0, you have to pick cuDNN v6.0, cuDNN v5.1, or cuDNN v5 of CUDA8.0. Purchased GeForce GXT 1080 Ti is based on cuDNN v5.1 so that we have selected cuDNN v5.1 Library for Linux.

Download the following file cudnn-8.0-linux-x64-v5.1-tgz:
https://developer.nvidia.com/compute/machine-learning/cudnn/secure/v5.1/prod_20161129/8.0/cudnn-8.0-linux-x64-v5.1-tgz

```
$ mv cudnn-8.0-linux-x64-v5.1-tgz cudnn-8.0-linux-x64-v5.1.tgz
```

The following tar command creates two directories.

```
$ tar xvf cudnn-8.0-linux-x64-v5.1.tgz
```

created folder: cuda/include cuda/lib64

Copy several important files: cudnn.h and libcudnn*

chmod a+r command adds read permission for all classes

```
$ sudo cp cuda/include/cudnn.h  /usr/include/
$ sudo cp cuda/lib64/libcudnn*  /usr/local/cuda/lib64
$ sudo chmod a+r /usr/local/cuda/lib64/libcudnn*
```

Download two files although they are for Ubuntu14.04:

https://developer.nvidia.com/compute/machine-learning/cudnn/secure/v5.1/prod_20161129/8.0/libcudnn5_5.1.10-1+cuda8.0_amd64-deb

https://developer.nvidia.com/compute/machine-learning/cudnn/secure/v5.1/prod_20161129/8.0/libcudnn5-dev_5.1.10-1+cuda8.0_amd64-deb

And install two files by dpkg -i command:

```
$ sudo dpkg -i libcudnn5_5.1.10-1+cuda8.0_amd64.deb
$ sudo dpkg -i libcudnn5-dev_5.1.10-1+cuda8.0_amd64.deb
$ sudo echo "/usr/local/cuda-8.0/lib64" >/etc/ld.so.conf.d/cuda-8-0.conf
$ sudo ldconf
```

## keras (machine learning framework)

In order to activate p27 environment (python2.7 environment), run the following source command:

```
$ source p27/bin/activate
```

Once p27 is activated, prompt will be (p27).

In order to install keras and tensorflow-gpu, run the following command:

```
(p27) $ pip install keras tensorflow-gpu
```

"python -V" command shows python version.

```
(p27) $ python -V
python 2.7.12
```

"pip -V" command shows pip version.

```
(p27) $ pip -V
pip 9.0.1 from /home/takefuji/p27/lib/python2.7/site-packages (python 2.7)
```

To run mnist_cnn.py, run the following command:

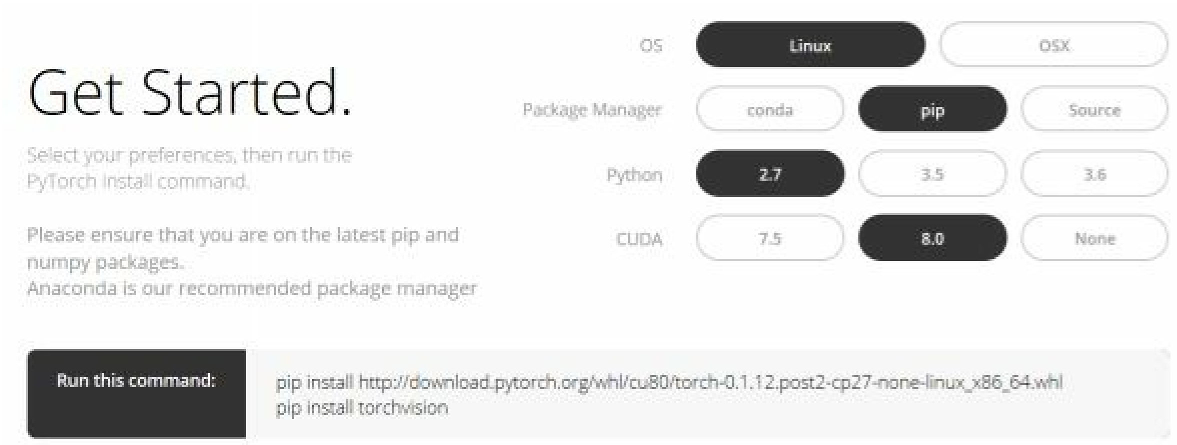
```
(p27) $ python mnist_cnn.py
Using TensorFlow backend.
x_train shape: (60000, 28, 28, 1)
60000 train samples
10000 test samples
Train on 60000 samples, validate on 10000 samples
Epoch 1/12
...
Test loss: 0.0271579699688
```

Test accuracy: 0.9911

## pytorch (machine learning framework)

In order to install pytorch framework, access to the following site for installation information:

http://pytorch.org/



Get Started.

Select your preferences, then run the PyTorch install command.

Please ensure that you are on the latest pip and numpy packages.
Anaconda is our recommended package manager

OS: Linux [✓], OSX
Package Manager: conda, pip [✓], Source
Python: 2.7 [✓], 3.5, 3.6
CUDA: 7.5, 8.0 [✓], None

Run this command:

```
pip install http://download.pytorch.org/whl/cu80/torch-0.1.12.post2-cp27-none-linux_x86_64.whl
pip install torchvision
```

Fig. 2.1.2 how to install pytorch

In order to install pytorch framework, run the following command as shown in Fig. 2.1.2:

```
(p27) $ pip install http://download.pytorch.org/whl/cu80/torch-0.1.12.post2-cp27-none-linux_x86_64.whl
(p27) $ pip install torchvision
```

Download examples from the following site:

https://github.com/pytorch/examples

```
(p27) $ wget https://github.com/pytorch/examples/archive/master.zip
(p27) $ unzip master.zip
(p27) $ cd examples-master/mnist
```

```
(p27) $ pip install -r requirements.txt
```

To run an mnist benchmark problem, run the following command:

```
(p27) $ python main.py
Train Epoch: 1 [0/60000 (0%)]   Loss: 2.390087
Train Epoch: 1 [640/60000 (1%)] Loss: 2.350225
...
Test set: Average loss: 0.0505, Accuracy: 9826/10000 (98%)
Train Epoch: 12 [0/60000 (0%)]  Loss: 0.034612
...
Train Epoch: 12 [59520/60000 (99%)]     Loss: 0.219132
Test set: Average loss: 0.0511, Accuracy: 9832/10000 (98%)
```

## Chainer (machine learning framework)

To install chainer, run the following command:

```
(p27) $ pip install -U chainer
(p27) $ pip install -U cupy
(p27) $ wget https://github.com/pfnet/chainer/archive/master.zip
(p27) $ unzip master.zip
(p27) $ cd chainer-master/examples/mnist
```

In order to use GPU, "--gpu=0" for single GPU, run the following command:

```
(p27) $ python train_mnist.py --gpu=0
GPU: 0
# unit: 1000
# Minibatch-size: 100
# epoch: 20
epoch main/loss validation/main/loss main/accuracy  validation/main/accuracy  elapsed_time
1 0.195361 0.084808    0.940017      0.974           8.60946
...
20 0.00508269 0.102755  0.998483       0.982          34.3499
```

# Chapter 2.2 GPU paralle machine based on Windows 10

Instead of building a GPU parallel computer, you can use the laptop with an NVIDIA GPU card. There are several laptops with GTX 1070 GPU as shown in Table 2.2.1 or GTX 1080 GPU as shown in Table 2.2.2.

Table 2.2.1 NVIDIA GTX 1070 embedded laptops
http://www.ultrabookreview.com/10939-laptops-nvidia-1070-1080/

| Model | Screen | Hardware | Graphics | TB3 | Weight |
|---|---|---|---|---|---|
| **Acer Predator 15 G9-593** | **15.6-inch FHD GSync 60 Hz** | **Skylake Core HQ / max 64 GB RAM** | **GTX 1070** | **Yes** | **7.93 lbs / 3.6 kg** |
| **Acer Predator 17 G9-793** | **17.3-inch FHD or UHD GSync 60 Hz** | **Skylake Core HQ / max 64 GB RAM** | **GTX 1070** | **Yes** | **9.41 lbs / 4.27 kg** |
| **Alienware 15** | **15.6-inch FHD, QHD GSync 120 Hz or UHD** | **Kaby Lake Core HK / max 32 GB RAM** | **GTX 1070** | **Yes** | **7.69 lbs / 3.49 kg** |
| **Alienware 17** | **17.3-inch FHD, QHD GSync 120 Hz or UHD** | **Kaby Lake Core HK / max 32 GB RAM** | **GTX 1070** | **Yes** | **9.74 lbs / 4.42 kg** |
| **Aorus X7 v6** | **17.3-inch QHD 120 Hz or UHD 60 Hz** | **Kaby Lake Core HK / max 64 GB RAM** | **GTX 1070** | **No** | **6.4 lbs / 2.9 kg** |
| **Aorus X5 v6** | **15.6-inch FHD 120 Hz or WQHD+ 60 Hz** | **Kaby Lake Core HK / max 64 GB RAM** | **GTX 1070** | **No** | **5.5 lbs / 2.5 kg** |
| **Asus ROG G752VS** | **17.3-inch FHD GSync 75/120 Hz or UHD 60 Hz** | **Skylake Core HK / max 64 GB RAM** | **GTX 1070** | **Yes** | **9.5 lbs / 4.3 kg** |
| **Asus ROG GL502VS** | **15.6-inch FHD GSync 60 Hz** | **Kaby Lake Core HQ / max 32 GB RAM** | **GTX 1070** | **No** | **5.7 lbs / 2.6 kg** |
| **Asus ROG GL702VS** | **17.3-inch FHD GSync 75 Hz or 120 Hz** | **Kaby Lake Core HQ / max 32 GB RAM** | **GTX 1070** | **Yes** | **6.4 lbs / 2.9 kg** |

| | | | | | |
|---|---|---|---|---|---|
| EVGA SC17 | 17.3-inch UHD GSync 60 Hz | Skylake Core HK / max 32 GB RAM | GTX 1070 | No | 9.9 lbs / 4.5 kg |
| Gigabyte P57X v6 | 17.3-inch FHD | Kaby Lake Core HQ / max 32 GB RAM | GTX 1070 | No | 6.6 lbs / 3.0 kg |
| Gigabyte P37X v6 | 17.3-inch FHD or UHD | Skylake Core HQ / max 32 GB RAM | GTX 1070 | No | 5.95 lbs / 2.7 kg |
| Gigabyte P35X v6 | 15.6-inch UHD | Kaby Lake Core HQ / max 32 GB RAM | GTX 1070 | No | 5.3 lbs / 2.4 kg |
| Gigabyte P56 | 15.6-inch UHD | Kaby Lake Core HQ / max 32 GB RAM | GTX 1070 | Yes | – |
| HP Omen 17 Gamer | 17.3-inch UHD GSync 60 Hz | Skylake Core HQ / max 32 GB RAM | GTX 1070 | No | 6.3 lbs / 2.8 kg |
| Lenovo IdeaPad Y910 | 17.3-inch FHD | Skylake Core HK / max 64 GB RAM | GTX 1070 | Yes | 10.1 lbs / 4.6 kg |
| MSI GT62VR RE Dominator | 15.6-inch FHD GSync 60 Hz or UHD | Kaby Lake Core HQ / max 64 GB RAM | GTX 1070 | No | 6.4 lbs / 2.9 kg |
| MSI GT72VR RE Dominator | 17.3-inch FHD GSync 60 Hz or UHD | Kaby LakeCore HQ / max 64 GB RAM | GTX 1070 | No | 8.6 lbs / 3.9 kg |
| MSI GT83VR 6RE Titan | 18.4-inch FHD GSync 60 Hz | Skylake Core HK / max 64 GB RAM | SLI GTX 1070 | Yes | 9.9 lbs / 4.5 kg |
| MSI GT73VR 6RE Titan | 17.3-inch FHD GSync 120 Hz or UHD | Skylake Core HQ and HK / max 64 GB RAM | single or SLI GTX 1070 | Yes | 8.4 lbs / 3.8 kg |
| Origin PC EON15-X | 15.6-inch FHD GSync 120 Hz or UHD | Skylake Core K / max 64 GB RAM | GTX 1070 | No | 7.5 lbs / 3.4 kg |
| Sager NP8153-S / Clevo P650RS | 15.6-inch FHD GSync 60 Hz | Kaby LakeCore HQ or HK / max 64 GB RAM | GTX 1070 | No | 5.84 lbs / 2.63 kg |
| Sager NP9152 / Clevo P750DM2 | 15.6-inch FHD GSync 60 Hz | Kaby Lake Core K / max 64 GB RAM | GTX 1070 | Yes | 7.5 lbs / 3.4 kg |

Table 2.2.2 NVIDIA GTX 1080 embedded laptops

| Model | Screen | Hardware | Graphics | TB3 | Weight |
|---|---|---|---|---|---|
| Acer Predator 17 X GX-792 | 17.3-inch FHD or UHD GSync 60 Hz | Kaby Lake Core HQ / max 64 GB RAM | GTX 1080 | Yes | 10.03 lbs / 4.54 kg |
| Acer Predator 21 X | 21-inch curved GSync | Kabylake Core HQ / max 64 GB RAM | SLI GTX 1080 | No | 17.8 lbs /8 kg |
| Acer Triton 700 | 15.6-inch GSync | Kaby Lake Core HQ / max 32 GB RAM | GTX 1080 | Yes | 5.73 lbs / 2.6 kg |
| Alienware 17 | 17.3-inch QHD GSync 120 Hz | Kaby Lake Core HK / max 64 GB RAM | GTX 1080 | Yes | 9.74 lbs / 4.42 kg |
| Aorus X7 DT v6 | 17.3-inch QHD 120 Hz or UHD 60 Hz | Kaby LakeCore HK / max 64 GB RAM | GTX 1080 | No | 7.1 lbs / 3.2 kg |
| Asus ROG GX800VH | 18.4-inch FHD or UHD | Skylake Core HK / max 64 GB RAM | SLI GTX 1080 | No | 12.5 lbs / 5.7 kg |
| Asus ROG GX800VI | 18.4-inch FHD or UHD | Skylake Core HK / max 64 GB RAM | GTX 1080 | No | 12.1 lbs / 5.5 kg |
| Asus ROG GX701VI | 17.3-inch FHD GSync 120 Hz | Skylake Core HK / max 64 GB RAM | GTX 1080 | Yes | 8.4 lbs / 3.8 kg |
| MSI GT83VR 6RF Titan Pro | 18.4-inch FHD GSync 60 Hz | Skylake Core HK / max 64 GB RAM | SLI GTX 1080 | Yes | 13.1 lbs / 4.5 kg |
| MSI GT73VR 6RF Titan Pro | 17.3-inch FHD GSync 120 Hz or UHD | Skylake Core HQ and HK / max 64 GB RAM | GTX 1080 | Yes | 8.4 lbs / 3.8 kg |
| Origin PC EON17-X | 17.3-inch FHD GSync 120 Hz or UHD | Skylake Core K / max 64 GB RAM | GTX 1080 | Yes | 8.6 lbs / 3.9 kg |
| Origin PC EON17-SLX | 17.3-inch FHD GSync 120 Hz or UHD | Skylake Core K / max 64 GB RAM | SLI GTX 1080 | Yes | 12 lbs / 5.4 kg |

| | | | | | |
|---|---|---|---|---|---|
| **Razer Blade Pro** | **17.3-inch UHD IGZO GSync touch** | **Skylake Core HQ / max 32 GB RAM** | **GTX 1080** | **Yes** | **7.8 lbs / 3.54 kg** |
| **Schenker XMG U727 / Clevo / Avant P870** | **17.3-inch GSync FHD, QHD or UHD** | **Skylake Core T and K / max 64 GB RAM** | **SLI GTX 1080** | **Yes** | **12.1 lbs / 5.5 kg** |
| **Schenker XMG U716 / Clevo P775 / Sager NP9152** | **17.3-inch GSync FHD or UHD** | **Skylake Core T and K / max 64 GB RAM** | **GTX 1080** | **Yes** | **8.6 lbs / 3.9 kg** |

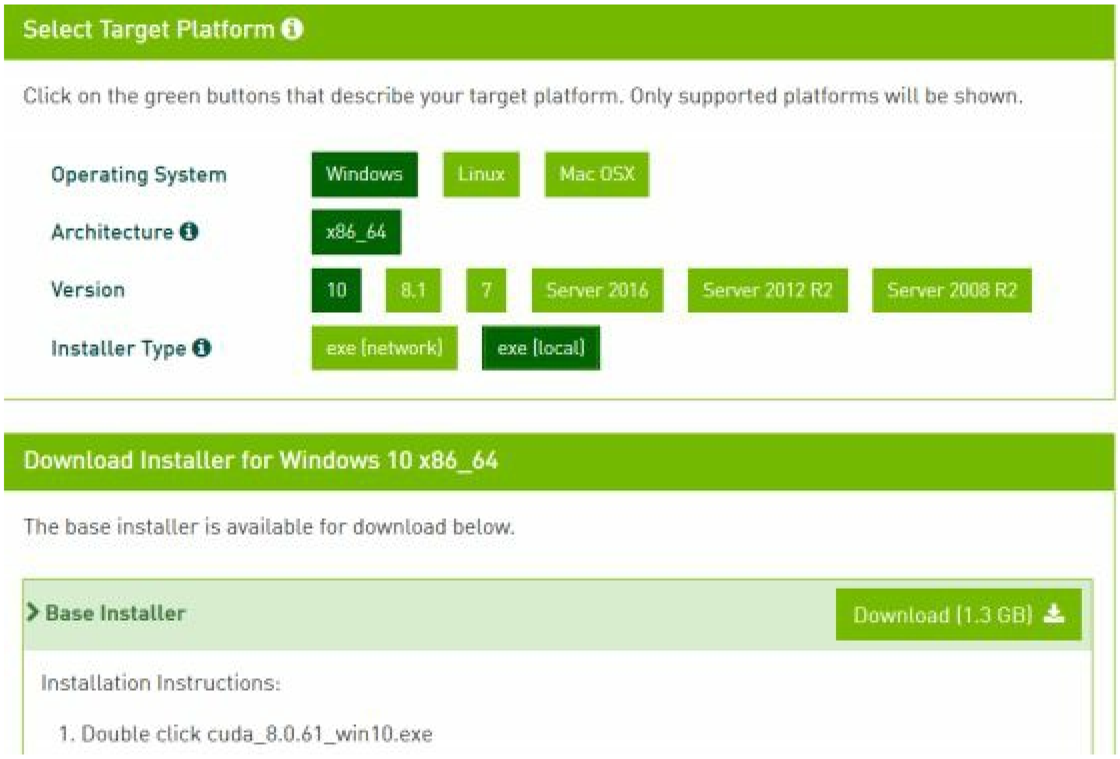
In this Chapter2.2, ASUS ROG GL502VS is used in this book.

## cygwin installation

On Windows 10 using GeForce GTX 1070 Ti (core 1920), first we need to install cygwin on Windows 10. In order to install cygwin, download the file setup-x86_64.exe from the following site:
https://cygwin.com/setup-x86_64.exe
In cygwin installation, you should install openssh, unzip, tar, wget, vim.

## cuda installation

Download cuda_8.0.61_win10.exe from
https://developer.nvidia.com/cuda-downloads


Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown.

Operating System: [✓ Windows] Linux Mac OSX
Architecture: [✓ x86_64]
Version: [✓ 10] 8.1 7 Server 2016 Server 2012 R2 Server 2008 R2
Installer Type: exe (network) [✓ exe (local)]

Download Installer for Windows 10 x86_64

The base installer is available for download below.

Base Installer — Download (1.3 GB)

Installation Instructions:

1. Double click cuda_8.0.61_win10.exe

Fig. 2.2.1 how to download cuda file
1. Double-click cuda_8.0.61_win10.exe file for installation.
2. Download python3.5.2 from the following site:
https://www.python.org/ftp/python/3.5.2/python-3.5.2-amd64.exe
3. Double-click python-3.5.2-amd64.exe file for installation.

## virtualenv installation

4. In order to install virtualenv, run the following command

```
$ sudo pip install -U pip
$ sudo pip install virtualenv
```

5. In order to make python3.5 environment, run the following command:

```
$ virtualenv p35 -p $(which python3.5)
```

or

```
$ virtualenv p35
```

6. In order to activate p35 environment, run the following command:

```
$ source p35/Scripts/activate
(p35)$
```

7. Download binary files (scipy, numpy) from:

http://www.lfd.uci.edu/~gohlke/pythonlibs/

scipy-0.19.0-cp35-cp35m-win_amd64.whl

numpy-1.13.0+mkl-cp35-cp35m-win_amd64.whl

## keras installation

8. Install scipy, numpy, keras respectively by the following commands:

```
(p35) $ pip install -U pip
(p35) $ pip install scipy*.whl
(p35) $ pip install numpy*.whl
(p35) $ pip install -U keras
```

9. Download tensorflow-gpu file from the following site using wget command:

https://storage.googleapis.com/tensorflow/windows/gpu/tensorflow_gpu-1.2.0rc1-cp35-cp35m-win_amd64.whl

```
(p35) $ wget https://storage.googleapis.com/tensorflow/windows/gpu/tensorflow_gpu-1.2.0rc1-cp35-cp35m-win_amd64.whl
```

10. Install tensorflow-gpu by pip command:

```
(p35) $ pip install tensorflow_gpu-1.2.0rc1-cp35-cp35m-win_amd64.whl
```

or

```
(p35) $ pip install --ignore-installed --upgrade tensorflow_gpu-1.2.0rc1-cp35-cp35m-win_amd64.whl
```

11. Download mnist_cnn.py for benchmarking GPU using wget command

```
(p35) $ wget https://raw.githubusercontent.com/fchollet/keras/master/examples/mnist_cnn.py
```

12. Run mnist_cnn.py using keras framework:

```
(p35) $ python mnist_cnn.py
```