

INT305 – ASSESSMENT 2

Assessment Number	1
Contribution to Overall Marks	15%
Submission Deadline	12/Dec/2024

Assessment Objective

This assessment aims at evaluating students' ability to exploit the deep learning knowledge, which is accumulated during lectures, and after-class study, to analyze, design, implement, develop, test and document the images classification using CNN framework. The assessment will be based on the Pytorch software.

General Guidelines

1. The descriptions in the **Problem Specifications** are required to be analyzed with **mathematic equations**, combined with the **explanations of all elements in each equation**.
2. The **modified part of the source codes** is required to include in the report.
3. The **final classification performance** that you obtain should be reported in the lab report. Meanwhile, the **screenshot of the final performance results** is also required in the report.
4. For the final performance results that you obtained, the **numeric quantitative results** are required. In addition, is also important to **include some subjective image examples** in the report.
5. Students need to conduct the coding and experiment all by yourself. The obtained results cannot be shared, and each student should analyze the results and write the report individually.
6. Report name: **INT305-Lab-Name-studentID.pdf** (e.g. **INT305-Lab-SanZhang-12345657.pdf**)
7. Late submission by email is **not** acceptable for Assignment 2. Please pay attention to the cut-off deadline. Your assignment will not be graded after the cut-off deadline.
8. For late assignments, a penalty of five points will be deducted for each day past the deadline. If you submitted within 1 hour after the deadline time, it is acceptable for reserved as normal submission. After 1 hour, your assignment will be counted as late submission. (same for CW1)

Handwritten digits (MNIST) recognition

Overall Description:

This lab is to use the Pytorch software and CNN (Convolutional Neural Network) framework for handwritten digits classification. Image classification aims to predict the category of object in an image (one image can only have one object in it). It has attracted much attention within the computer vision community in recent years as an important component for computer vision applications, such as self-driving vehicles, video surveillance and robotics. It is also the foundation of other computer vision research topics, such as object detection and instance segmentation.

CNN is a framework with both feature extraction and classification using deep convolutional neural network. A typical CNN pipeline is shown below.

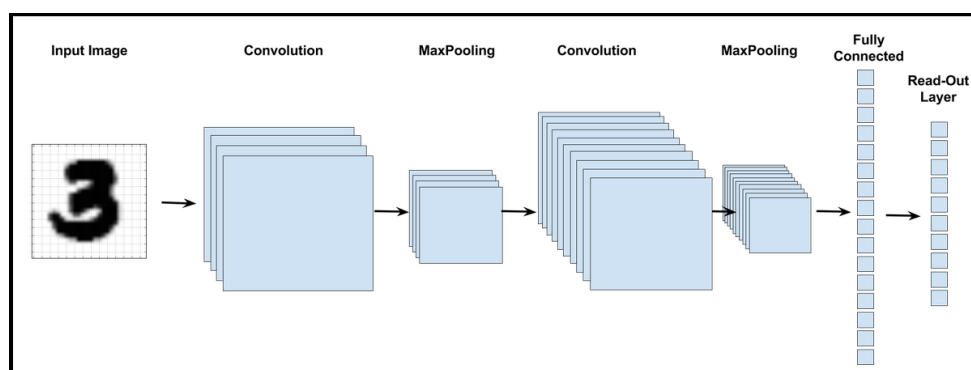


Figure 1. CNN image classification pipeline.

The Dataset we will use is MNIST dataset, it has a **training set of 60,000 examples**, and a **test set of 10,000 examples**. The dataset is originally available on Yann Lecun's website. The followings are examples of handwritten digits in MNIST dataset.

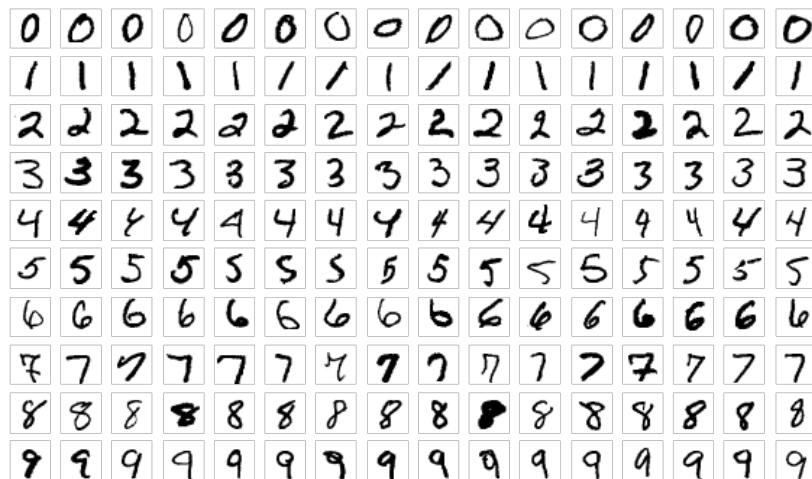


Figure 2. Examples of handwritten digits in MNIST dataset.

Problem Specifications:

1. Please describe the 2 key components in the CNN framework: the convolutional kernel and the loss functions used in the framework. (20%)
2. Please train (or fine-tune) and test the framework on MNIST and report the final accuracy performance that you have achieved. Please also report some well classified and misclassified images by including the images and corresponding classification confidence value. (40%).
3. Propose your own method to further improve the classification performance or reduce the model size. You need also compare different methods with the performance you obtained and explain why. The final classification accuracy is not the most important part, you may better refer to some latest published papers and code these state of the art methods to improve the performance. The explanation and analysis of your adopted method is highly related to your final score. (40%)

Environment Preparation:

1 Install Anaconda

1.1 Install Anaconda on Windows

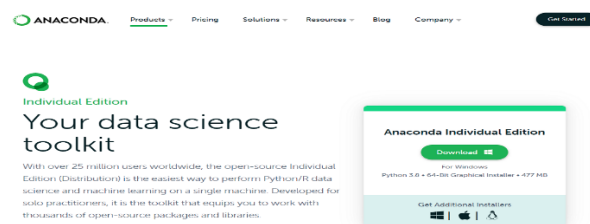
Anaconda is open-source software that contains Jupyter, spyder, etc that is used for large data processing, data analytics, heavy scientific computing.

Conda is a package and environment management system that is available across Windows, Linux, and MacOS, similar to PIP. It helps in the installation of packages and dependencies associated with a specific language like python, C++, Java, Scala, etc. Conda is also an environment manager and helps to switch between different environments with just a few commands.

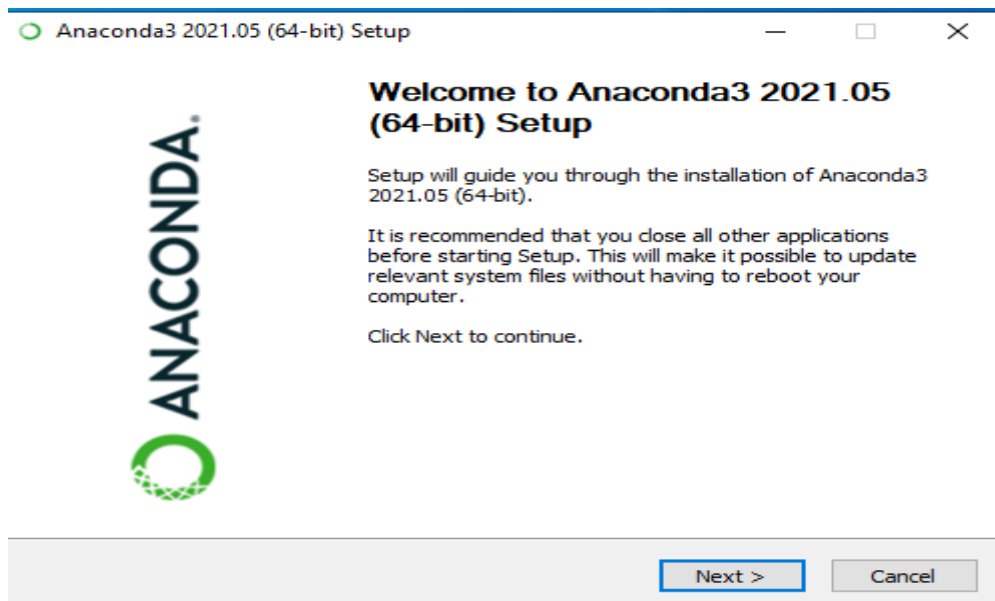
Step 1: Visit this website

<https://www.anaconda.com/products/individual-d>

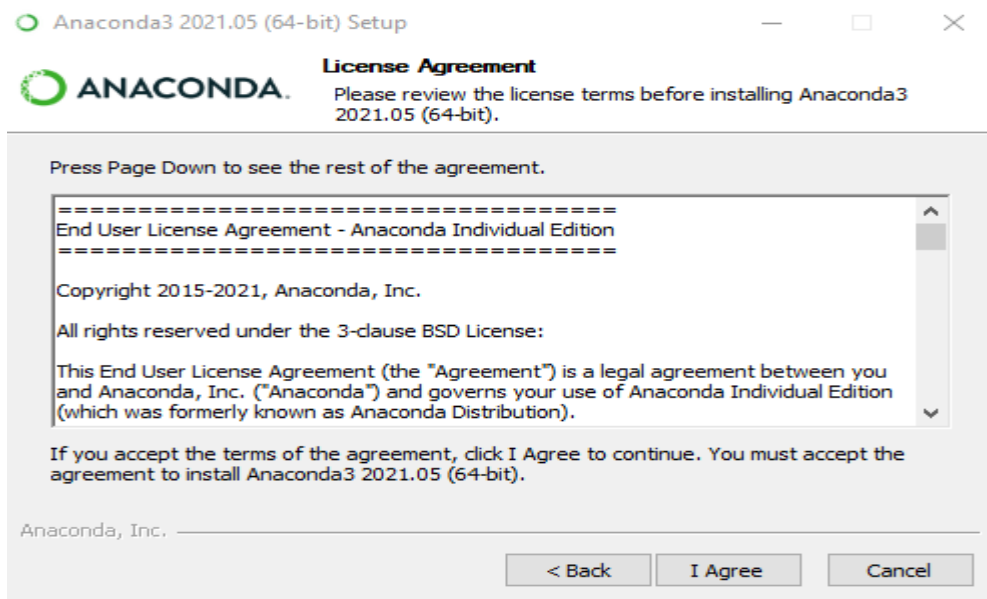
and download the Anaconda installer.



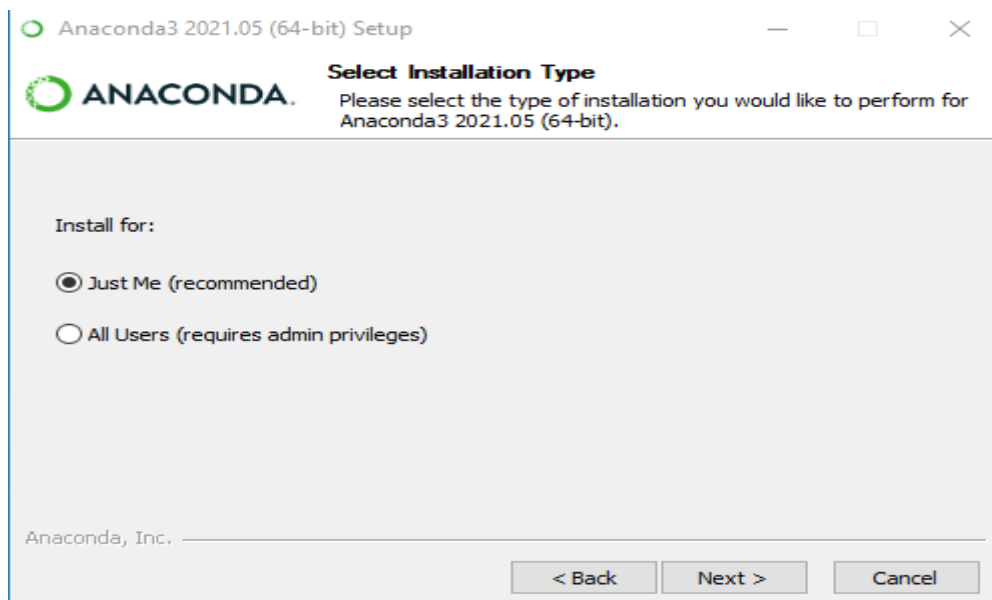
Step 2: Click on the downloaded .exe file and click on Next.



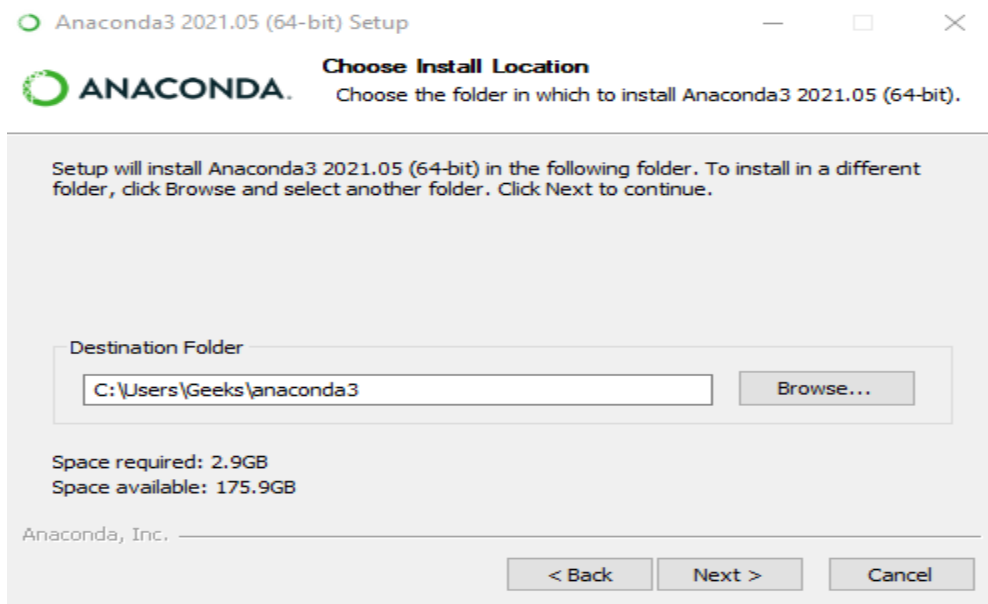
Step 3: Agree to the terms and conditions.



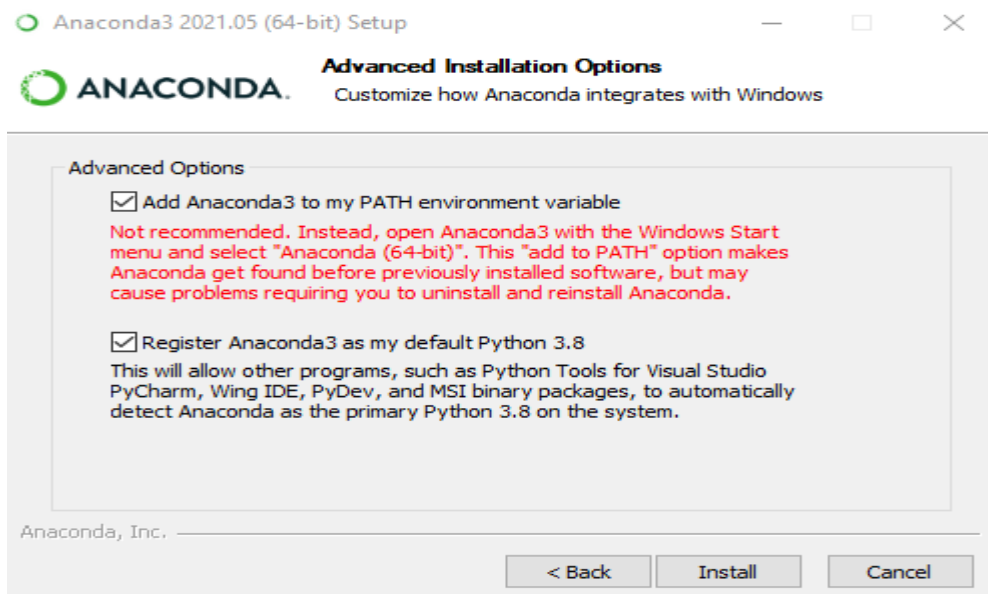
Step 4: Select the installation type.



Step 5: Choose the installation location.

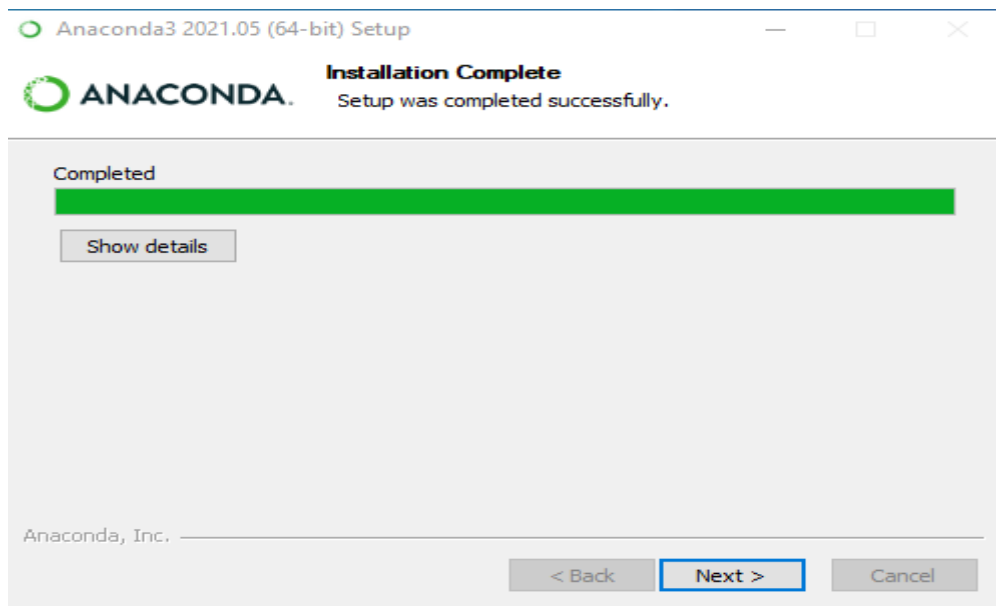


Step 6: Now check the checkbox to add Anaconda to your environment Path and click Install.

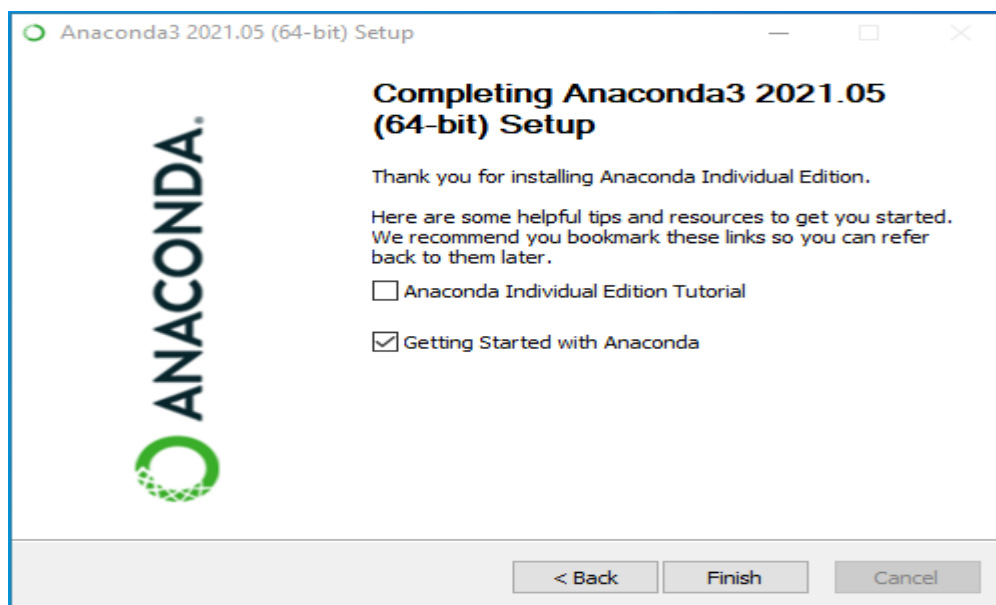


This will start the installation.

Step 7: After the installation is complete you'll get the following message, here click on Next.



Step 8: You'll get the following screen once the installation is ready to be used. Here click on Finish.

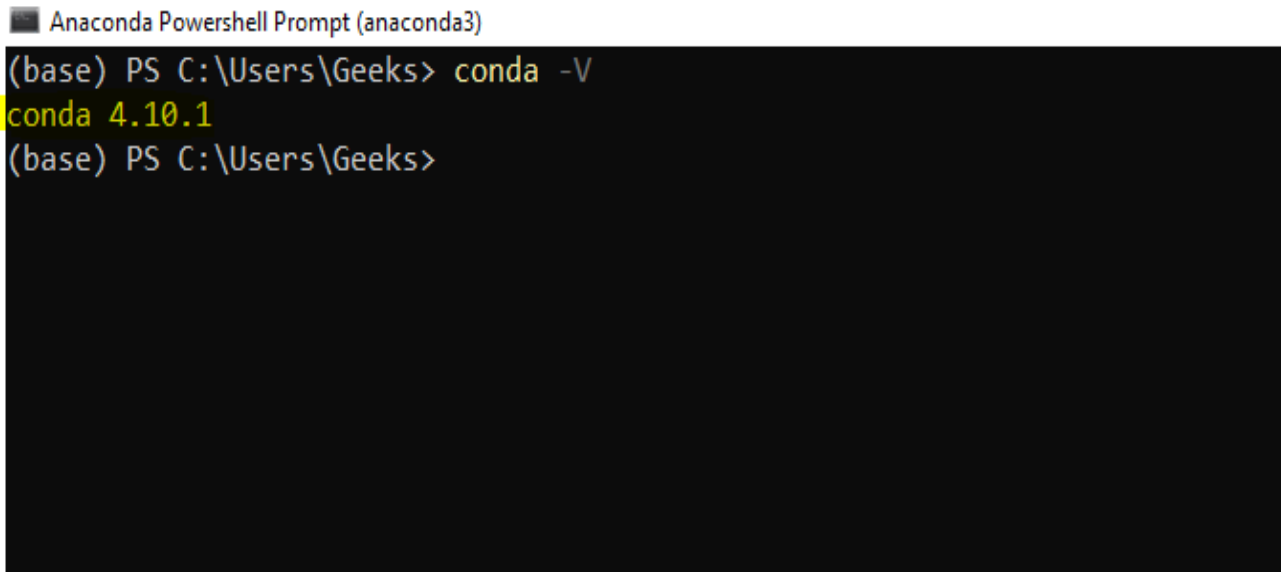


Verifying the installation:

Now open up the Anaconda Power Shell prompt and use the below command to check the conda version:

`conda -V`

If conda is installed successfully, you will get a message as shown below:

A screenshot of an Anaconda Powershell Prompt terminal window. The title bar reads "Anaconda Powershell Prompt (anaconda3)". The terminal shows a PowerShell prompt "(base) PS C:\Users\Geeks>" followed by the command "conda -V". The output is "conda 4.10.1". The prompt then returns to "(base) PS C:\Users\Geeks>".

```
Anaconda Powershell Prompt (anaconda3)
```

```
(base) PS C:\Users\Geeks> conda -V
conda 4.10.1
(base) PS C:\Users\Geeks>
```

1.2 Install Anaconda on Linux

Prerequisites

Firstly, open terminal on your Ubuntu system and execute the command mentioned below to update packages repository:

```
sudo apt update
```

Then install the curl package, which is further required for the downloading the installation script.

```
sudo apt install curl -y
```

Step 1 – Prepare the Anaconda Installer

Now I will go to the /tmp directory and for this purpose we will use cd command.

```
cd /tmp
```

Next, use the curl command line utility to download the Anaconda installer script from the official site. Visit the Anaconda installer script download page to check for the latest versions.

Then, download the script as below:

```
curl --output anaconda.sh https://repo.anaconda.com/archive/Anaconda3-2021.05-Linux-x86_64.sh
```

To check the script SHA-256 checksum, I will use this command with the file name, though this step is optional:

```
sha256sum anaconda.sh
```

Output:

```
25e3ebae8905450ddac0f5c93f89c467  anaconda.sh
```


Check if the hash code is matching with code showing on download page.

Step 2 – Installing Anaconda on Ubuntu

Your system is ready to install Anaconda. Let's move to the text step and execute the Anaconda installer script as below:

```
bash anaconda.sh
```

Follow the wizard instructions to complete Anaconda installation process. You need to provide inputs during installation process as described below:

01. Use above command to run the downloaded installer script with the **bash** shell.

```
tecadmin@ubuntu2004:~$ bash anaconda.sh

Welcome to Anaconda3 2021.05

In order to continue the installation process, please review the license
agreement.
Please, press ENTER to continue
>>>
```

02. Type **“yes”** to accept the Anaconda license agreement to continue.

```
Do you accept the license terms? [yes|no]
[no] >>> yes
```

03. Verify the directory location for Anaconda installation on Ubuntu 20.04 system. Just hit **Enter** to continue installer to that directory.

```
Anaconda3 will now be installed into this location:
/home/tecadmin/anaconda3

- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below

[/home/tecadmin/anaconda3] >>>
```

04. Type **“yes”** to initialize the Anaconda installer on your system.

```
Preparing transaction: done
Executing transaction: done
installation finished.
Do you wish the installer to initialize Anaconda3
by running conda init? [yes|no]
[no] >>> yes
```

05. You will see the below message on successful Anaconda installation on Ubuntu 20.04 system.

```

Thank you for installing Anaconda3!

=====

Working with Python and Jupyter notebooks is a breeze with PyCharm Pro,
designed to be used with Anaconda. Download now and have the best data
tools at your fingertips.

PyCharm Pro for Anaconda is available at: https://www.anaconda.com/pycharm

tecadmin@ubuntu2004:~$

```

The Anaconda Installation Completed Successfully on your Ubuntu system. Installer added the environment settings in .bashrc file. Now, activate the installation using following command:

```
source ~/.bashrc
```

Now we are in the default base of the programming environment. To verify the installation we will open conda list.

```
conda list
```

Output:

```

# packages in environment at /home/tecadmin/anaconda3:
#
# Name                                Version                                Build Channel
_ipyw_jlab_nb_ext_conf               0.1.0                                py38_0
_libgcc_mutex                         0.1                                  main
alabaster                             0.7.12                             pyhd3eb1b0_0
anaconda                             2021.05                             py38_0
anaconda-client                       1.7.2                                py38_0
anaconda-navigator                    2.0.3                                py38_0
anaconda-project                      0.9.1                             pyhd3eb1b0_1
anyio                                 2.2.0                             py38h06a4308_1
appdirs                              1.4.4                                py_0

```

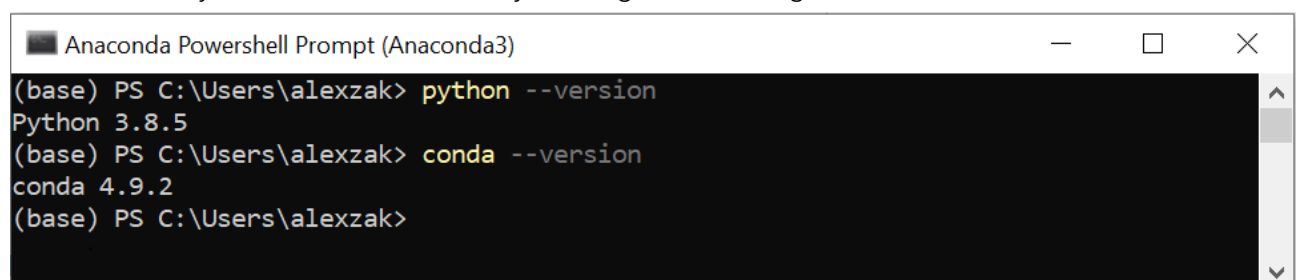
2 Install and configure PyTorch on your machine.

First, you'll need to setup a Python environment.

Open Anaconda manager via Start - Anaconda3 - Anaconda PowerShell Prompt and test your versions:

You can check your Python version by running the following command: `python --version`

You can check your Anaconda version by running the following command: `conda --version`



```

Anaconda Powershell Prompt (Anaconda3)
(base) PS C:\Users\alexzak> python --version
Python 3.8.5
(base) PS C:\Users\alexzak> conda --version
conda 4.9.2
(base) PS C:\Users\alexzak>

```

Now, you can install PyTorch package from binaries via Conda.

- 1 Navigate to <https://pytorch.org/>.

Select the relevant PyTorch installation details:

- PyTorch build – stable.
- Your OS
- Package – Conda
- Language – Python
- Compute Platform – CPU.

PyTorch Build	Stable (1.8.1)		Preview (Nightly)	
Your OS	Linux	Mac	Windows	
Package	Conda	Pip	LibTorch	Source
Language	Python		C++ / Java	
Compute Platform	CUDA 10.2	CUDA 11.1	ROCm 4.0 (beta)	CPU
Run this Command:	<pre>conda install pytorch torchvision torchaudio cpuonly -c pytorch</pre>			

- 2 Open Anaconda manager and run the command as it specified in the installation instructions.
`conda install pytorch torchvision torchaudio cpuonly -c pytorch`

```

Anaconda Powershell Prompt (Anaconda3)
(base) PS C:\Users\alexzak> conda install pytorch torchvision torchaudio cpuonly -c pytorch
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\alexzak\Anaconda3

  added / updated specs:
    - cpuonly
    - pytorch
    - torchaudio
    - torchvision

The following packages will be downloaded:

  package | build | size |
  -----|-----|-----|
  conda-4.9.2 | py37haa95532_0 | 2.9 MB |
  cpuonly-1.0 | 0 | 2 KB | pytorch
  libuv-1.40.0 | he774522_0 | 255 KB |
  ninja-1.10.2 | py37h6d14046_0 | 246 KB |
  pytorch-1.7.1 | py3.7_cpu_0 | 157.0 MB | pytorch
  torchaudio-0.7.2 | py37 | 2.7 MB | pytorch
  torchvision-0.8.2 | py37_cpu | 6.6 MB | pytorch
  typing_extensions-3.7.4.3 | py_0 | 28 KB |
  -----|-----|-----|
  Total: | 169.7 MB |

The following NEW packages will be INSTALLED:

  cpuonly | pytorch/noarch::cpuonly-1.0-0
  libuv | pkgs/main/win-64::libuv-1.40.0-he774522_0
  ninja | pkgs/main/win-64::ninja-1.10.2-py37h6d14046_0
  pytorch | pytorch/win-64::pytorch-1.7.1-py3.7_cpu_0
  torchaudio | pytorch/win-64::torchaudio-0.7.2-py37
  torchvision | pytorch/win-64::torchvision-0.8.2-py37_cpu
  typing_extensions | pkgs/main/noarch::typing_extensions-3.7.4.3-py_0

The following packages will be UPDATED:

  conda | 4.8.2-py37_0 --> 4.9.2-py37haa95532_0

Proceed ([y]/n)?

```

3 Confirm and complete the extraction of the required packages.

```

Anaconda Powershell Prompt (Anaconda3)

Proceed ([y]/n)? y

Downloading and Extracting Packages
pytorch-1.7.1 | 1007.0 MB | #####
##### | 100%
torchaudio-0.7.2 | 2.7 MB | #####
##### | 100%
libuv-1.40.0 | 255 KB | #####
##### | 100%
torchvision-0.8.2 | 7.3 MB | #####
##### | 100%
ninja-1.10.2 | 247 KB | #####
##### | 100%
cudatoolkit-11.0.221 | 627.0 MB | #####
##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
(base) PS C:\Users\alexzak>

```

Let's verify PyTorch installation by running sample PyTorch code to construct a randomly initialized tensor.

- 4 Open the Anaconda PowerShell Prompt and run the following command.

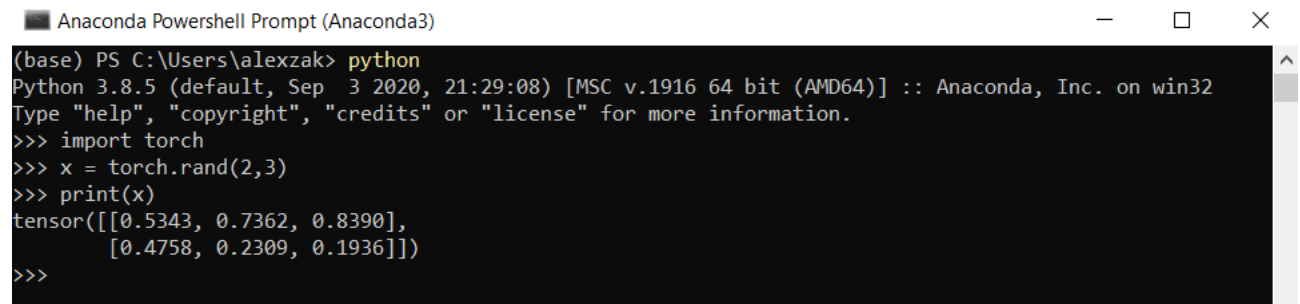
```
python
```

```
import torch
```

```
x = torch.rand(2, 3)
```

```
print(x)
```

The output should be a random 5x3 tensor. The numbers will be different, but it should look similar to the below.

A screenshot of the Anaconda PowerShell Prompt window. The title bar reads "Anaconda Powershell Prompt (Anaconda3)". The command prompt shows the following sequence of commands and output:

```
(base) PS C:\Users\alexzak> python
Python 3.8.5 (default, Sep 3 2020, 21:29:08) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import torch
>>> x = torch.rand(2,3)
>>> print(x)
tensor([[0.5343, 0.7362, 0.8390],
        [0.4758, 0.2309, 0.1936]])
>>>
```

References

[1] <http://yann.lecun.com/exdb/mnist/>

[2] https://github.com/Eng-Abdelrahman-M/handwritten_digit_recognition/blob/main/handwritten_digit_recognition.ipynb