Mechatronics Engineering and Automation Program
CSE488: Computational Intellegence
Lab #0: Configuration of your Environment

Install TensorFlow with pip

TensorFlow 2 packages are available

- tensorflow —Latest stable release for CPU-only
- tensorflow-gpu —Latest stable release with GPU support (https:/ www.tensorflow.org/install/gpu) (Ubuntu and Windows)
- tf-nightly —Preview build (unstable). Ubuntu and Windows include GPU support (https:/ www.tensorflow.org/install/gpu).

System requirements

- pip 19.0 or later
- Ubuntu 16.04 or later (64-bit)
- macOS 10.12.6 (Sierra) or later (64-bit) (no GPU support)
- Windows 7 or later (64-bit) (Python 3 only)
- Raspbian 9.0 or later.

Hardware requirements

Starting with TensorFlow 1.6, binaries use AVX instructions (https:/ en.wikipedia.org/wiki/Advanced_Vector_Extensions#CPUs_with_AVX) which may not run on older CPUs.

Read the GPU support guide (https:/ www.tensorflow.org/install/gpu) to set up a CUDA®-enabled GPU card on Ubuntu or Windows.

1. Install the Python development environment on your system

For Ubuntu

```
sudo apt update
sudo apt install python3-dev python3-pip
sudo pip3 install -U virtualenv  # system-wide install
```

For windows

Install the *Microsoft Visual C++ 2015 Redistributable Update 3*. This comes with *Visual Studio 2015* but can be installed separately:

1. Go to the Visual Studio downloads,
2. Select *Redistributables and Build Tools*,
3. Download and install the *Microsoft Visual C++ 2015 Redistributable Update 3*.

   Make sure long paths are enabled on Windows.

   Install the *64-bit* Python 3 release for Windows (select `pip` as an optional feature).

```
pip3 install -U pip virtualenv
```

2. Create a virtual environment (recommended)

Python virtual environments are used to isolate package installation from the system.

Create a new virtual environment by choosing a Python interpreter and making a `./venv` directory to hold it:

For Ubuntu

Create a new virtual environment by choosing a Python interpreter and making a `./venv` directory to hold it:

```
virtualenv --system-site-packages -p python3 ./venv
```

Activate the virtual environment using a shell-specific command:

```
source ./venv/bin/activate  # sh, bash, ksh, or zsh
```

When virtualenv is active, your shell prompt is prefixed with `(venv)`. Install packages within a virtual environment without affecting the host system setup. Start by upgrading `pip`:

```
pip install --upgrade pip
pip list  # show packages installed within the virtual environment
```

And to exit virtualenv later:

```
deactivate  # don't exit until you're done using TensorFlow
```

For windows

Create a new virtual environment by choosing a Python interpreter and making a `./venv` directory to hold it: (don't type the C:\> it's the directory from the **cmd**)

```
C:\> virtualenv --system-site-packages -p python3 ./venv
```

Activate the virtual environment using a shell-specific command:

```
C:\> .\venv\Scripts\activate
```

When virtualenv is active, your shell prompt is prefixed with `(venv)`. Install packages within a virtual environment without affecting the host system setup. Start by upgrading `pip`:

```
(venv) C:\> pip install --upgrade pip
(venv) C:\> pip list  # show packages installed within the virtual environment
```

And to exit virtualenv later:

```
(venv) C:\> deactivate  # don't exit until you're done using TensorFlow
```

3. Install the TensorFlow pip package

Choose one of the following TensorFlow packages to install from PyPI:

- `tensorflow` —Latest stable release for CPU-only *(recommended for beginners)*
- `tensorflow-gpu` —Latest stable release with GPU support *(Ubuntu and Windows)*
- `tf-nightly` —Preview build (unstable). Ubuntu and Windows include GPU support

Package location

A few installation mechanisms require the URL of the TensorFlow Python package. The value you specify depends on your Python version.

Version	URL
Linux	
Python 2.7 CPU-only	https://storage.googleapis.com/tensorflow/linux/cpu/tensorflow-2.0.0-cp27-none-linux_x86_64.whl
Python 2.7 GPU support	https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow_gpu-2.0.0-cp27-none-linux_x86_64.whl
Python 3.5 CPU-only	https://storage.googleapis.com/tensorflow/linux/cpu/tensorflow-2.0.0-cp35-cp35m-linux_x86_64.whl
Python 3.5	https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow_gpu-2.0.0-cp35-

GPU support	cp35m-linux_x86_64.whl
Python 3.6 CPU-only	https://storage.googleapis.com/tensorflow/linux/cpu/tensorflow-2.0.0-cp36-cp36m-linux_x86_64.whl
Python 3.6 GPU support	https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow_gpu-2.0.0-cp36-cp36m-linux_x86_64.whl
Python 3.7 CPU-only	https://storage.googleapis.com/tensorflow/linux/cpu/tensorflow-2.0.0-cp37-cp37m-linux_x86_64.whl
Python 3.7 GPU support	https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow_gpu-2.0.0-cp37-cp37m-linux_x86_64.whl
macOS (CPU-only)	
Python 2.7	https://storage.googleapis.com/tensorflow/mac/cpu/tensorflow-2.0.0-py2-none-any.whl
Windows	
Python 3.5 CPU-only	https://storage.googleapis.com/tensorflow/windows/cpu/tensorflow-2.0.0-cp35-cp35m-win_amd64.whl
Python 3.5 GPU support	https://storage.googleapis.com/tensorflow/windows/gpu/tensorflow_gpu-2.0.0-cp35-cp35m-win_amd64.whl
Python 3.6 CPU-only	https://storage.googleapis.com/tensorflow/windows/cpu/tensorflow-2.0.0-cp36-cp36m-win_amd64.whl
Python 3.6 GPU support	https://storage.googleapis.com/tensorflow/windows/gpu/tensorflow_gpu-2.0.0-cp36-cp36m-win_amd64.whl
Raspberry PI (CPU-only)	
There is no TensorFlow 2 support for RPi yet, it is expected in a future release	
Python 2.7, Pi0 or Pi1	https://storage.googleapis.com/tensorflow/raspberrypi/tensorflow-1.14.0-cp27-none-linux_armv6l.whl
Python 2.7, Pi2 or Pi3	https://storage.googleapis.com/tensorflow/raspberrypi/tensorflow-1.14.0-cp27-none-linux_armv7l.whl
Python 3, Pi0 or	https://storage.googleapis.com/tensorflow/raspberrypi/tensorflow-1.14.0-cp34-none-

Pi1	linux_armv6l.whl
Python 3, Pi2 or Pi3	https://storage.googleapis.com/tensorflow/raspberrypi/tensorflow-1.14.0-cp34-none-linux_armv7l.whl