

Installing TensorFlow on a Mac with the new M1 or M2 chip

In order to install TensorFlow on an Apple with the new "ARM" chip set (and built-in GPU) you will have to follow instructions specific to this architecture.

Do not attempt to install using the standard

```
conda install tensorflow
```

Verify that you have installed the correct Anaconda !

The Apple M1 or M2 is able to execute code for older Macs (with the Intel 386 chipset) by emulating an Intel 386.

When installing Anaconda: you **must** use the installer with the "M1" in the description.

If you fail to do this, your Anaconda will just emulate an Intel 386

- won't be able to install TensorFlow
- won't be able to use the GPU

Run the following diagnostic in Python to see which Anaconda you have installed

```
import platform  
  
platform.platform()
```

If you have installed the correct Anaconda, the string `arm` will appear as part of the output and you are good to proceed to the next step.

If the output contains the string `i386` (e.g., `macOS-10.16-x86_64-i386-64bit`):
You have installed the wrong Anaconda !

Replacing the wrong Anaconda

You will need to uninstall it, following [these instructions](https://docs.anaconda.com/anaconda/install/uninstall/) (<https://docs.anaconda.com/anaconda/install/uninstall/>).

You then can run the correct Anaconda installer (the one with the "M1" in the title).

Run the diagnostic above to verify that you now have the correct Anaconda.

Environments in Anaconda

Anaconda allows you to have multiple "environments" (distinct collections of Anaconda packages). The default environment (the one you are used to) is called the "base" environment. Environments are distinct: the software installed in one is not related to another.

It is particularly useful to create a new environment when testing out new packages; if the new environment is not successful, you will not affect other environments.

We will create a new environment that we will name `tensorflow` (since we are testing out Tensorflow installation. The name is arbitrary EXCEPT for the fact that we use this name in one of the files named below. The name doesn't matter as long as it is used consistently)

When you have more than one environment (e.g., "base" and "tensorflow") you can switch between them.

To switch to the `tensorflow` environment, use the shell command

`conda activate tensorflow`

All subsequent commands (including ones that install new packages, like `conda install`) will affect ONLY the `tensorflow` environment (and not the base environment). If you think you have installed a package and the system can't find it: it probably means you installed it in a different environment !

To switch back to the "base" environment, use the shell command

`conda deactivate`

Creating the **tensorflow** environment

By creating a separate environment, we prevent any accidents/mistakes from affecting the "base" environment.

As this process is still experimental, it is a good idea to separate the "base" from **tensorflow**.

Make sure that you are in the "base" environment. The following command will return you to the base environment.

```
conda deactivate
```

Create a new environment (called `tensorflow`) with the shell command

```
conda create --name tensorflow
```

This environment will initially be very minimal, providing just basic Python.

Installing TensorFlow for Mac M1/M2 in the tensorflow environment

In the same directory as this notebook, you will find the file `tensorflow-apple-metal-conda.yml`.

This contains a list of the packages that need to be installed.

Install Tensorflow in the new environment by running the shell command

```
conda env update -n tensorflow -f  
tensorflow-apple-metal-conda.yml
```

Note on this installation method

The method using the `tensorflow-apple-metal-conda.yml` file was necessary until sometime in Fall 2023.

- It should still work
- We will continue to use it until such time as we are able to test a simpler, updated method.

Make the **tensorflow** environment visible to Jupyter

You need to make the new environment visible to Jupyter. Switch to the new environment via shell command

```
conda activate tensorflow
```

and then issue the shell command

```
python -m ipykernel install --user --name  
tensorflow --display-name "Python 3.9  
(tensorflow)"
```

The next time you start Jupyter, you will notice a new entry (labeled "Python 3.9 (tensorflow)") in the Change Kernel sub-menu of the Kernel menu entry.

When you want to create a notebook running in the **tensorflow** environment, choose this item from the sub-menu

```
Kernel --> Change Kernel --> Python 3.9  
(tensorflow)
```

If you want to create a notebook running in the base environment, choose Python 3 from the sub-menu

Kernel --> Change Kernel --> Python 3

Having both the old (base) and new (tensorflow) environments is just a safety measure. If the new environment is successful, you may choose to always use it in the future.

Giving a different name to your new environment

You don't need to name your new environment `tensorflow`.

You can choose another name as long as you change the `Name: tensorflow` line in `tensorflow-apple-metal-conda.yml` to match.

You must also use your chosen name in any of the shell commands above that use the name `tensorflow`.

Missing packages

The set of packages we put in the `tensorflow` environment is much smaller than what is available in "base".

It is quite possible that you might find some packages are missing.

If this happens

`conda search XXX` will help you find available packages matching the string `XXX`

Install the missing packages with the shell command

`conda install ...`

Be sure you have activated the `tensorflow` environment before running `conda install` or you may wind up installing the package in the "base" environment and not `tensorflow`.

Credit and References

These instructions were derived from the [video \(https://www.youtube.com/watch?v=5DgWvU0p2bk\)](https://www.youtube.com/watch?v=5DgWvU0p2bk).

- [updated video: January 2023 \(https://www.youtube.com/watch?v=o4-bl_iZKPA&t=0s\)](https://www.youtube.com/watch?v=o4-bl_iZKPA&t=0s)

The video is summarized in [this notebook \(https://github.com/jeffheaton/t81_558_deep_learning/blob/master/install/tensorflow-install-conda-mac-metal-jul-2022.ipynb\)](https://github.com/jeffheaton/t81_558_deep_learning/blob/master/install/tensorflow-install-conda-mac-metal-jul-2022.ipynb).

- [updated notebook: January 2023 \(https://github.com/jeffheaton/t81_558_deep_learning/blob/master/install/tensorflow-install-mac-metal-jan-2023.ipynb\)](https://github.com/jeffheaton/t81_558_deep_learning/blob/master/install/tensorflow-install-mac-metal-jan-2023.ipynb)
-

Verifying your installation

At the end of that notebook is a section "Testing your environment" which you can run to verify the installation.

You should also try running

```
import tensorflow as tf
```

in a Jupyter notebook to verify that you are able to import TensorFlow.

- Remember to do this in a notebook in the tensorflow environment

Alternate method: Miniconda

Miniconda is a very minimal subset of Anaconda.

We will install just enough (hopefully) packages for the course. But should something be missing: see the section on Missing Packages.

Follow the instructions in the notebook (or video) above.


```
In [1]: print("Done")
```

Done

