# Install Neural Structured Learning

There are several ways to set up your environment to use Neural Structured

Learning (NSL) in TensorFlow:

\* The easiest way to learn and use NSL requires no installation: run the NSL

tutorials directly in your browser using

[Google Colaboratory](https://colab.research.google.com/notebooks/welcome.ipynb).

\* To use NSL on a local machine, install the

[NSL package](#install-neural-structured-learning-using-pip) with Python's

`pip` package manager.

\* If you have a unique machine configuration,

[build NSL](#build-the-neural-structured-learning-pip-package) from source.

Note: NSL requires a TensorFlow version of 1.15 or higher. NSL also supports

TensorFlow 2.x with the exception of v2.1, which contains a bug that is

incompatible with NSL.

## Install Neural Structured Learning using pip

#### 1. Install the Python development environment.

On Ubuntu:

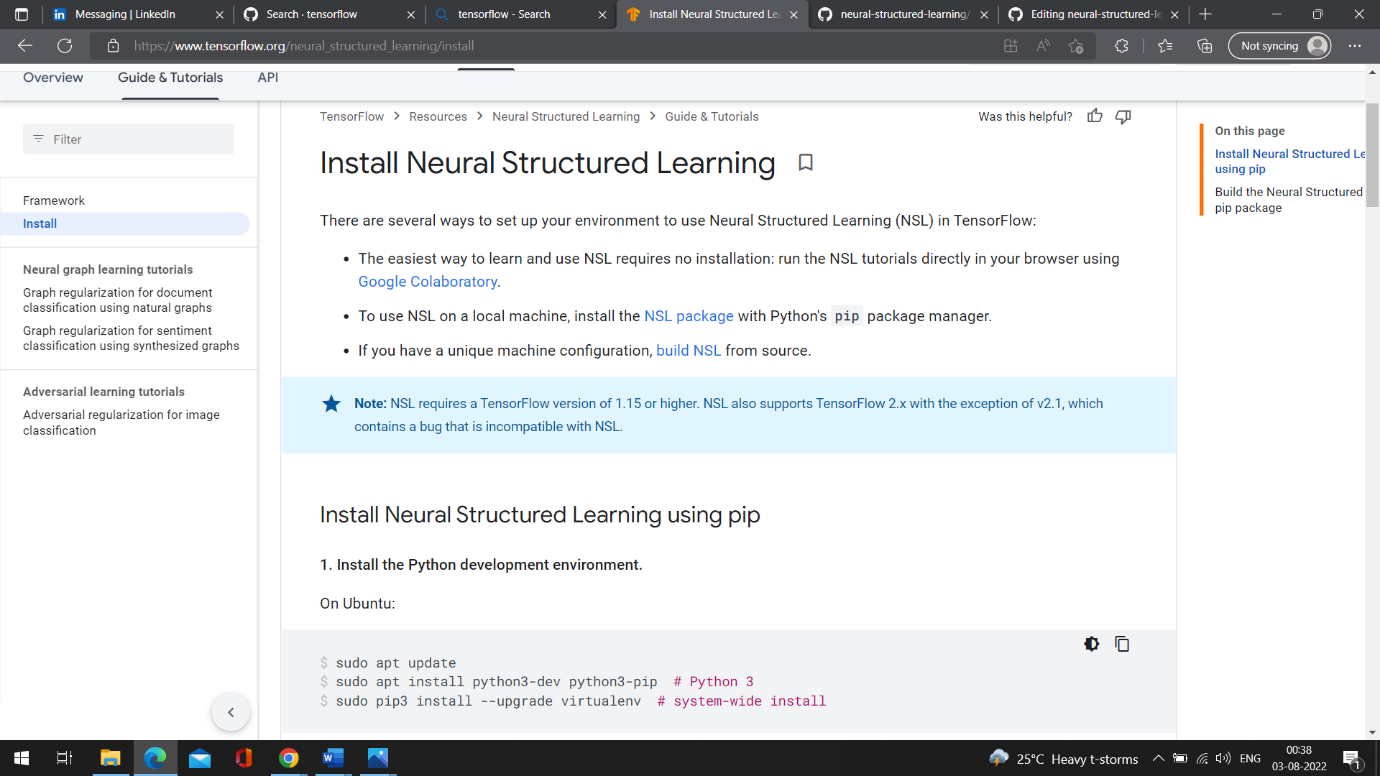
<pre class="prettyprint lang-bsh">

<code class="devsite-terminal">sudo apt update</code>

<code class="devsite-terminal">sudo apt install python3-dev python3-pip # Python 3</code>

<code class="devsite-terminal">sudo pip3 install --upgrade virtualenv # system-wide install</code>

</pre>



On macOS:

<pre class="prettyprint lang-bsh">

<code class="devsite-terminal">/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"</code>

<code class="devsite-terminal">export PATH="/usr/local/bin:/usr/local/sbin:$PATH"</code>

<code class="devsite-terminal">brew update</code>

<code class="devsite-terminal">brew install python # Python 3</code>

<code class="devsite-terminal">sudo pip3 install --upgrade virtualenv # system-wide install</code>

</pre>

#### 2. Create a virtual environment.

<pre class="prettyprint lang-bsh">

<code class="devsite-terminal">virtualenv --python python3 "./venv"</code>

<code class="devsite-terminal">source "./venv/bin/activate"</code>

<code class="devsite-terminal tfo-terminal-venv">pip install --upgrade pip</code>

</pre>

Note: To exit the virtual environment, run `deactivate`.

#### 3. Install TensorFlow

CPU support:

<pre class="prettyprint lang-bsh">

<code class="devsite-terminal tfo-terminal-venv">pip install 'tensorflow>=1.15.0'</code>

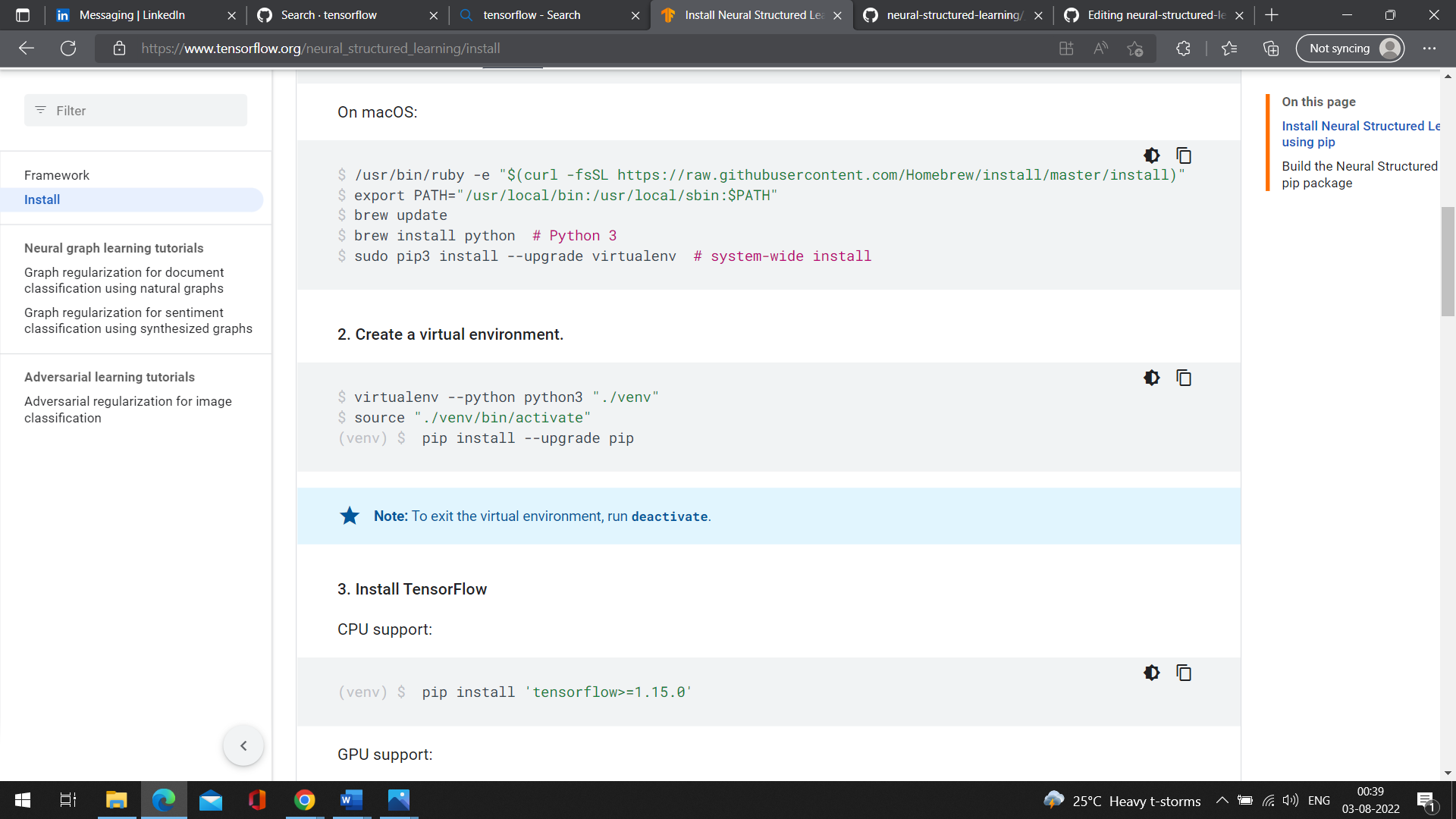
</pre>

GPU support:

<pre class="prettyprint lang-bsh">

<code class="devsite-terminal tfo-terminal-venv">pip install 'tensorflow-gpu>=1.15.0'</code>

</pre>



#### 4. Install the Neural Structured Learning `pip` package.

<pre class="prettyprint lang-bsh">

<code class="devsite-terminal tfo-terminal-venv">pip install --upgrade neural\_structured\_learning</code>

</pre>

#### 5. (Optional) Test Neural Structured Learning.

<pre class="prettyprint lang-bsh">

<code class="devsite-terminal tfo-terminal-venv">python -c "import neural\_structured\_learning as nsl"</code>

</pre>

Success: Neural Structured Learning is now installed.

## Build the Neural Structured Learning pip package

#### 1. Install the Python development environment.

On Ubuntu:

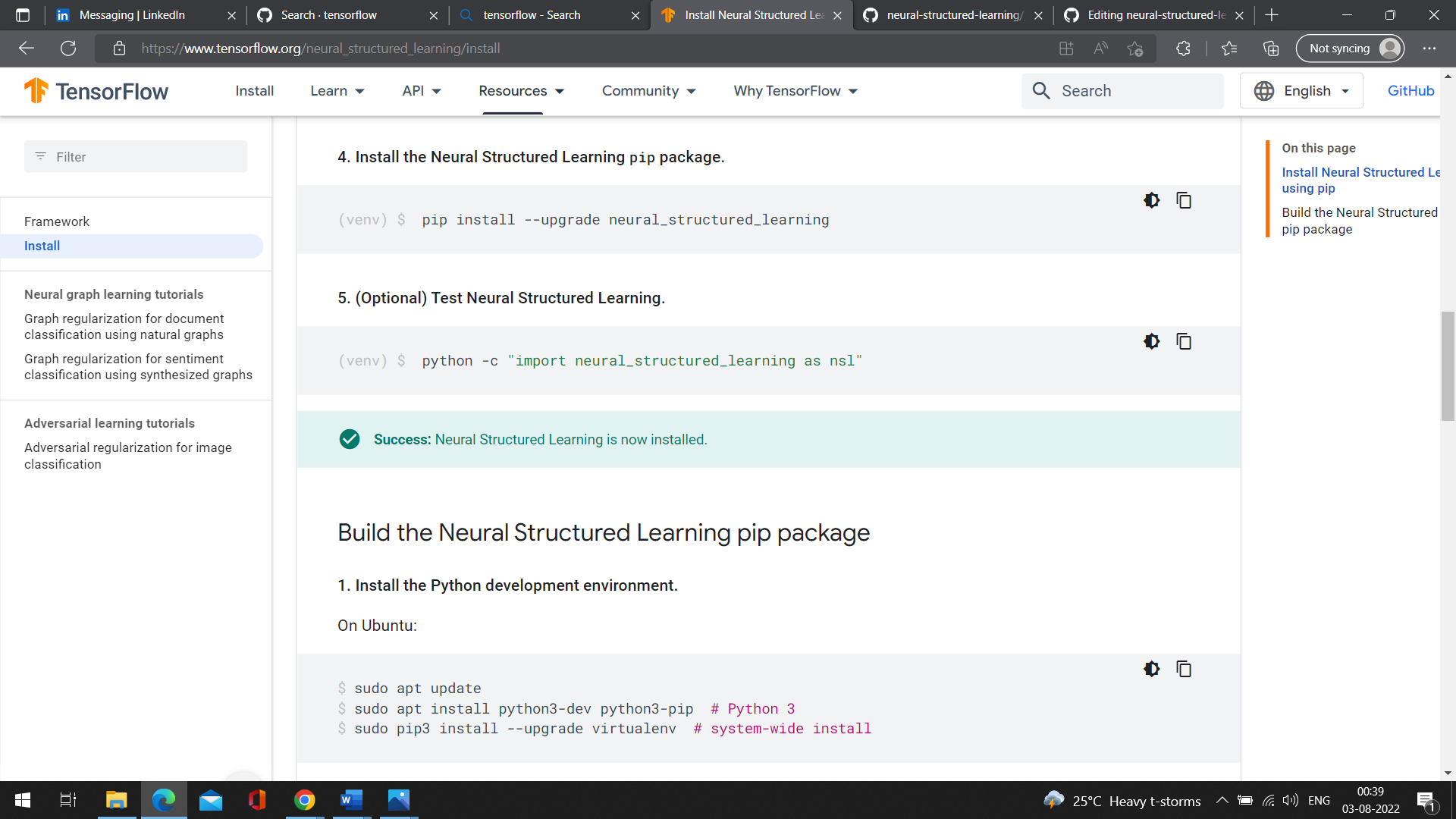
<pre class="prettyprint lang-bsh">

<code class="devsite-terminal">sudo apt update</code>

<code class="devsite-terminal">sudo apt install python3-dev python3-pip # Python 3</code>

<code class="devsite-terminal">sudo pip3 install --upgrade virtualenv # system-wide install</code>

</pre>



On macOS:

<pre class="prettyprint lang-bsh">

<code class="devsite-terminal">/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"</code>

<code class="devsite-terminal">export PATH="/usr/local/bin:/usr/local/sbin:$PATH"</code>

<code class="devsite-terminal">brew update</code>

<code class="devsite-terminal">brew install python # Python 3</code>

<code class="devsite-terminal">sudo pip3 install --upgrade virtualenv # system-wide install</code>

</pre>

#### 2. Install Bazel.

[Install Bazel](https://docs.bazel.build/versions/master/install.html), the

build tool used to compile Neural Structured Learning.

#### 3. Clone the Neural Structured Learning repository.

<pre class="prettyprint lang-bsh">

<code class="devsite-terminal">git clone https://github.com/tensorflow/neural-structured-learning.git</code>

</pre>

#### 4. Create a virtual environment.

<pre class="prettyprint lang-bsh">

<code class="devsite-terminal">virtualenv --python python3 "./venv"</code>

<code class="devsite-terminal">source "./venv/bin/activate"</code>

<code class="devsite-terminal tfo-terminal-venv">pip install --upgrade pip</code>

</pre>

Note: To exit the virtual environment, run `deactivate`.

#### 5. Install Tensorflow

Note that NSL requires a TensorFlow version of 1.15 or higher. NSL also supports

TensorFlow 2.0.

CPU support:

<pre class="prettyprint lang-bsh">

<code class="devsite-terminal tfo-terminal-venv">pip install 'tensorflow>=1.15.0'</code>

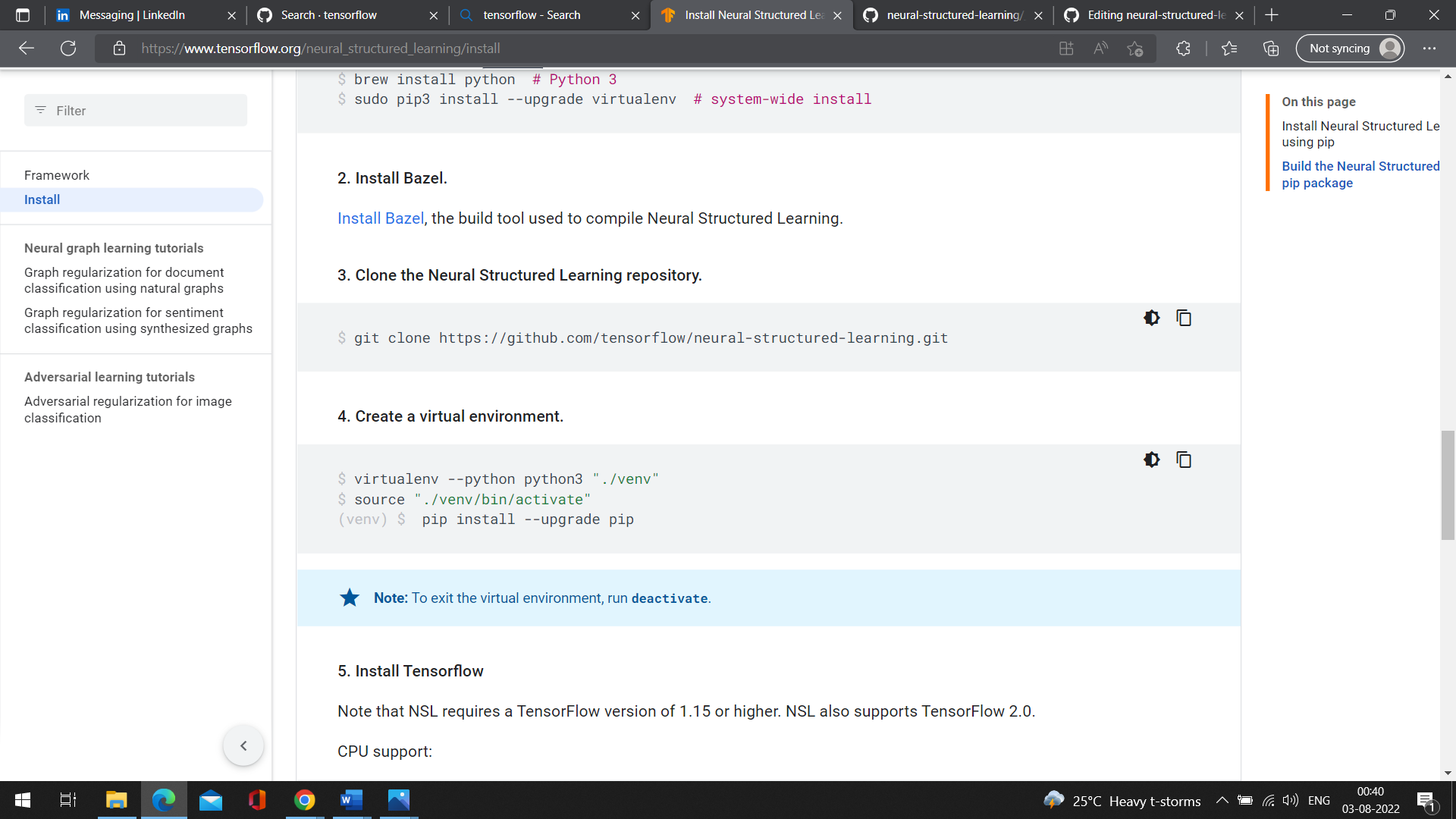
</pre>

GPU support:

<pre class="prettyprint lang-bsh">

<code class="devsite-terminal tfo-terminal-venv">pip install 'tensorflow-gpu>=1.15.0'</code>

</pre>



#### 6. Install Neural Structured Learning dependencies.

<pre class="prettyprint lang-bsh">

<code class="devsite-terminal">cd neural-structured-learning</code>

<code class="devsite-terminal tfo-terminal-venv">pip install --requirement neural\_structured\_learning/requirements.txt</code>

</pre>

#### 7. (Optional) Unit Test Neural Structured Learning.

<pre class="prettyprint lang-bsh">

<code class="devsite-terminal tfo-terminal-venv">bazel test //neural\_structured\_learning/...</code>

</pre>

#### 8. Build the pip package.

<pre class="prettyprint lang-bsh">

<code class="devsite-terminal tfo-terminal-venv">python setup.py bdist\_wheel --universal --dist-dir="./wheel"</code>

</pre>

#### 9. Install the pip package.

<pre class="prettyprint lang-bsh">

<code class="devsite-terminal tfo-terminal-venv">pip install --upgrade ./wheel/neural\_structured\_learning\*.whl</code>

</pre>

#### 10. Test Neural Structured Learning.

<pre class="prettyprint lang-bsh">

<code class="devsite-terminal tfo-terminal-venv">python -c "import neural\_structured\_learning as nsl"</code>

</pre>

Success: The Neural Structured Learning package is built.

