Installing TensorFlow through Conda:

CS 510
Supplemental 01
March 25, 2020
Basic Flow

• When in doubt, follow the instructions best suited to your personal working environment.

• These guidelines are mine running OS X Catalina and working toward Jupyter Notebook encapsulated TensorFlow execution.
Start with Instructions Here

Anaconda Documentation

Home
Anaconda Team Edition
Anaconda Enterprise 5
Anaconda Enterprise 4
Anaconda individual Edition

Installation
User guide
Getting started with Anaconda
Anaconda Navigator
Conda
Tasks
Cheat sheet
Frequently asked questions

TensorFlow

Anaconda makes it easy to install TensorFlow, enabling your data science, machine learning, and artificial intelligence workflows.

This page shows how to install TensorFlow with the conda package manager included in Anaconda and Miniconda. TensorFlow with conda is supported on 64-bit Windows 7 or later, 64-bit Ubuntu Linux 14.04 or later, 64-bit CentOS Linux 6 or later, and macOS 10.10 or later.

The instructions are the same for all operating systems. No `apt install` or `yum install` commands are required.

Install TensorFlow

1. Download and install Anaconda or the smaller Miniconda.

2. On Windows open the Start menu and open an Anaconda Command Prompt. On macOS or Linux open a terminal window. Use the default bash shell on macOS or Linux.

3. Choose a name for your TensorFlow environment, such as “tf”.

4. To install the current release of CPU-only TensorFlow recommended for beginners:
Test Your Installation

Missed TensorFlow Dev Summit? Check out the video playlist.

TensorFlow 2 quickstart for beginners

This short introduction uses Keras to:

1. Build a neural network that classifies images.
2. Train this neural network.
Start thinking about where you want to keep your ipynb files.
“Run All” – Your CPU Load May Vary

Note: It is possible to bake this `tf.nn.softmax` in as the activation, which allows the model output more directly interpretable, this approach also ensures a mathematically and numerically stable loss calculation for all models when using our loss.

The `losses.SparseCategoricalCrossentropy` loss takes the log of the loss for each example.

```python
In [34]: loss_fn = tf.keras.losses.SparseCategoricalCrossentropy

This loss is equal to the negative log probability of the true label.

This untrained model gives probabilities close to random (1/10) of
```
tf.log(1/10) ≈ -2.3
```

```python
In [35]: loss_fn(y_train[:1], predictions).numpy()
Out[35]: 2.153421
```

```python
In [36]: model.compile(optimizer='adam',
                    loss=loss_fn,
                    metrics=['accuracy'])
```

<table>
<thead>
<tr>
<th>Process Name</th>
<th>% CPU</th>
<th>CPU Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>python3.7</td>
<td>1,466.1</td>
<td>17:56:02</td>
</tr>
<tr>
<td>CaptchHelper</td>
<td>66.4</td>
<td>36.89</td>
</tr>
<tr>
<td><a href="http://localhost:8888">http://localhost:8888</a></td>
<td>11.4</td>
<td>23.85</td>
</tr>
<tr>
<td>Activity Monitor</td>
<td>10.4</td>
<td>13.79</td>
</tr>
<tr>
<td>python3.7</td>
<td>3.6</td>
<td>3.57</td>
</tr>
<tr>
<td>AXVisualSupportAgent</td>
<td>1.2</td>
<td>6:30:61</td>
</tr>
<tr>
<td>PopClipHelper</td>
<td>1.1</td>
<td>12.65</td>
</tr>
<tr>
<td>com.wacom.IOManager</td>
<td>1.1</td>
<td>4:29:67</td>
</tr>
<tr>
<td>Safari</td>
<td>1.1</td>
<td>54.24</td>
</tr>
<tr>
<td>PopClip</td>
<td>0.6</td>
<td>1:36:57</td>
</tr>
<tr>
<td>Safari Networking</td>
<td>0.5</td>
<td>7.00</td>
</tr>
<tr>
<td>Core Sync</td>
<td>0.4</td>
<td>4:02:97</td>
</tr>
<tr>
<td>Microsoft PowerPoint</td>
<td>0.3</td>
<td>1:00:79</td>
</tr>
<tr>
<td>Adobe Desktop Service</td>
<td>0.3</td>
<td>8:24:65</td>
</tr>
<tr>
<td>Backup and Sync from Google</td>
<td>0.2</td>
<td>6:29:93</td>
</tr>
<tr>
<td>trustd</td>
<td>0.1</td>
<td>2:42:17</td>
</tr>
<tr>
<td>Microsoft Outlook</td>
<td>0.1</td>
<td>42.19</td>
</tr>
<tr>
<td>AdobeIPCBroker</td>
<td>0.1</td>
<td>1:29:25</td>
</tr>
<tr>
<td>SystemUIServer</td>
<td>0.1</td>
<td>1:57:45</td>
</tr>
<tr>
<td>sharingd</td>
<td>0.1</td>
<td>1:57:45</td>
</tr>
<tr>
<td>mdworker_shared</td>
<td>0.1</td>
<td>0.25</td>
</tr>
<tr>
<td>AdobeCRDaemon</td>
<td>0.1</td>
<td>1:11.89</td>
</tr>
<tr>
<td>AdobeCRDaemon</td>
<td>0.0</td>
<td>1:16.53</td>
</tr>
<tr>
<td>AdobeCRDaemon</td>
<td>0.0</td>
<td>1:16.50</td>
</tr>
<tr>
<td>UserEventAgent</td>
<td>0.0</td>
<td>57.28</td>
</tr>
</tbody>
</table>

System: 69.41%
User: 30.59%
Idle: 0.00%
What Next …

• Study to the Tutorial Narrative.
• Make a copy and start adding
  – Look at the shape of terms like x_train
  – Can you display an example training image