

Installing TensorFlow through Conda:

CS 510

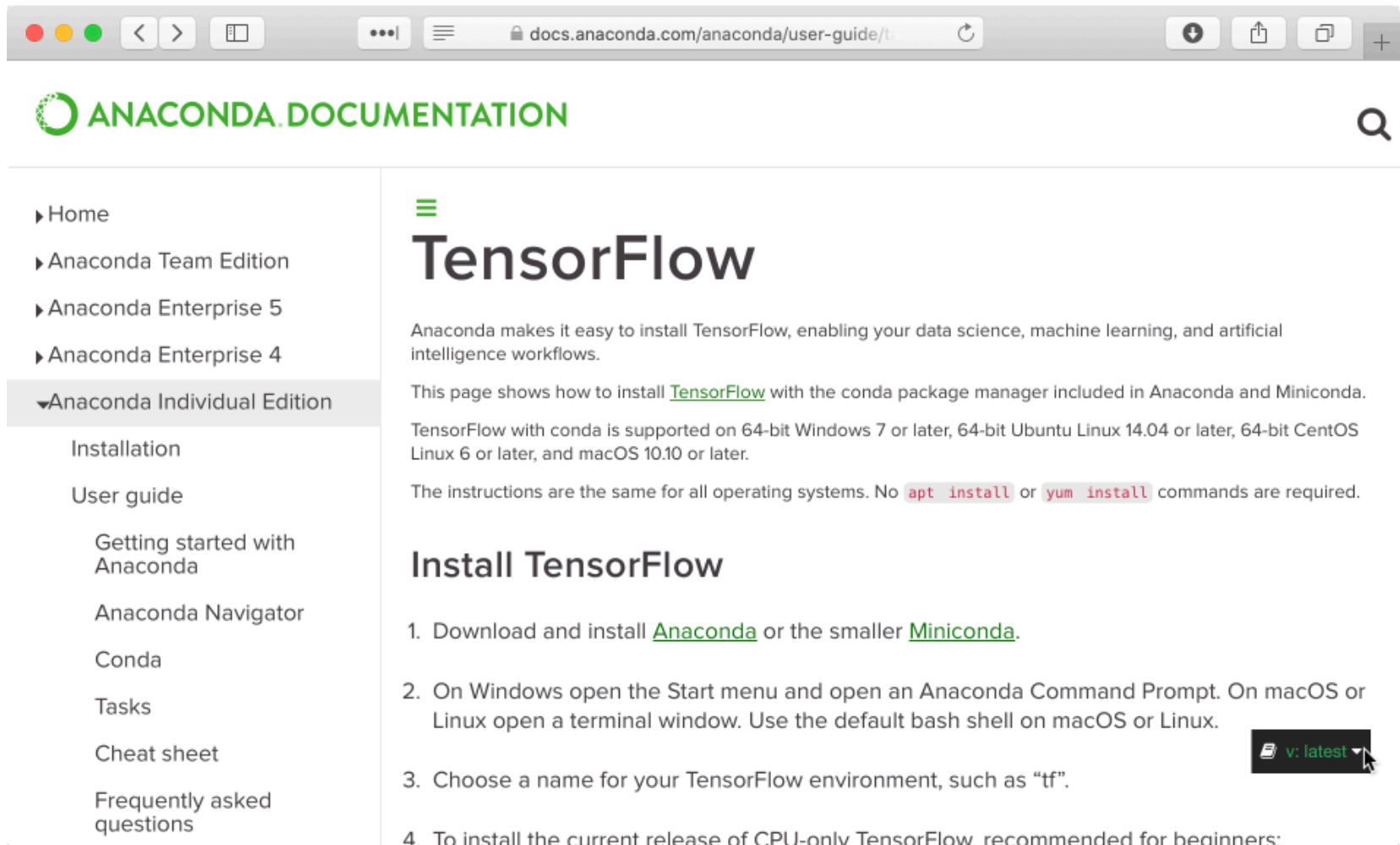
Supplemental 01

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



The screenshot shows a web browser window with the URL `docs.anaconda.com/anaconda/user-guide/t`. The page title is "ANACONDA DOCUMENTATION". The left sidebar contains a navigation menu with the following items: Home, Anaconda Team Edition, Anaconda Enterprise 5, Anaconda Enterprise 4, and Anaconda Individual Edition (expanded). Under "Anaconda Individual Edition", there are sub-items: Installation, User guide, Getting started with Anaconda, Anaconda Navigator, Conda, Tasks, Cheat sheet, and Frequently asked questions. The main content area is titled "TensorFlow" and contains the following text:

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:

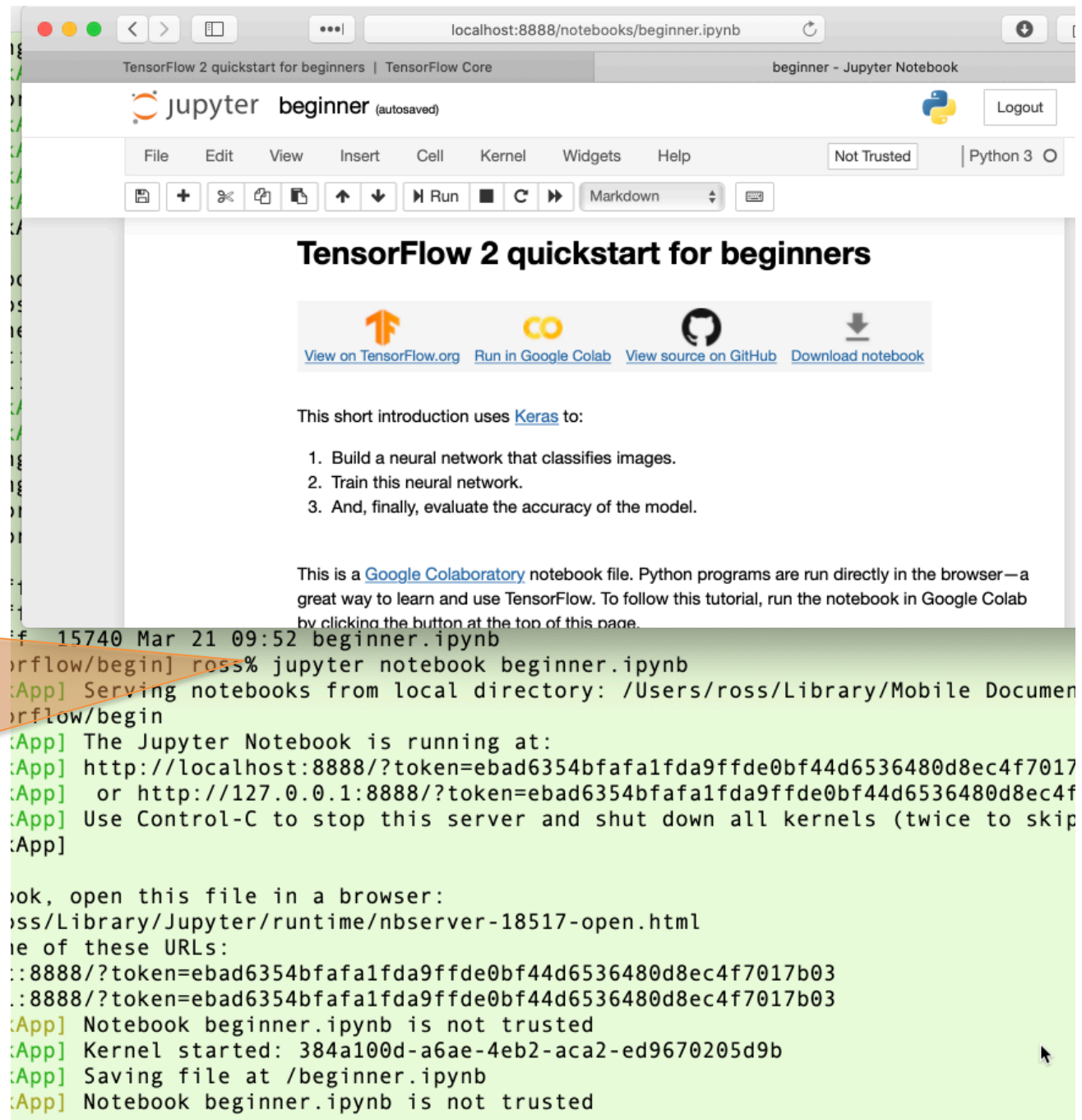
A code snippet is shown in a dark box: `v: latest` with a mouse cursor pointing to it.

Test Your Installation

The screenshot shows a web browser window displaying the TensorFlow website. The URL is www.tensorflow.org/tutorials/quickstart/beginner. The page title is "TensorFlow 2 quickstart for beginners | TensorFlow Core". The navigation bar includes "Install", "Learn", and "More" menus, a search bar, and "Language", "GitHub", and "Sign in" links. The main content area is titled "TensorFlow Core" and has tabs for "Overview", "Tutorials", "Guide", and "TF 1". The "Tutorials" tab is active, showing a list of TensorFlow tutorials on the left sidebar. The "Quickstart for beginners" tutorial is highlighted. The main content area features a dark blue banner with the text "Missed TensorFlow Dev Summit? Check out the video playlist." and a "Watch recordings" button. Below the banner, the breadcrumb "TensorFlow > Learn > TensorFlow Core > Tutorials" is shown, followed by a five-star rating. The main heading is "TensorFlow 2 quickstart for beginners". Below the heading are three buttons: "Run in Google Colab", "View source on GitHub", and "Download notebook". The text "This short introduction uses Keras to:" is followed by a numbered list of two steps: "1. Build a neural network that classifies images." and "2. Train this neural network."

Command Line Launch

Start thinking about where you want to keep your ipynb files



The screenshot shows a web browser window displaying a Jupyter Notebook titled "TensorFlow 2 quickstart for beginners". The notebook content includes a title, navigation links (View on TensorFlow.org, Run in Google Colab, View source on GitHub, Download notebook), and a list of tasks: "1. Build a neural network that classifies images.", "2. Train this neural network.", and "3. And, finally, evaluate the accuracy of the model." Below the list, it states "This is a Google Colaboratory notebook file. Python programs are run directly in the browser—a great way to learn and use TensorFlow. To follow this tutorial, run the notebook in Google Colab by clicking the button at the top of this page."

```
ross% jupyter notebook beginner.ipynb
[App] Serving notebooks from local directory: /Users/ross/Library/Mobile Documents
[App] The Jupyter Notebook is running at:
[App] http://localhost:8888/?token=ebad6354bfafa1fda9ffde0bf44d6536480d8ec4f7017
[App] or http://127.0.0.1:8888/?token=ebad6354bfafa1fda9ffde0bf44d6536480d8ec4f
[App] Use Control-C to stop this server and shut down all kernels (twice to skip
[App]

ok, open this file in a browser:
ross/Library/Jupyter/runtime/nbserver-18517-open.html
one of these URLs:
::8888/?token=ebad6354bfafa1fda9ffde0bf44d6536480d8ec4f7017b03
::8888/?token=ebad6354bfafa1fda9ffde0bf44d6536480d8ec4f7017b03
[App] Notebook beginner.ipynb is not trusted
[App] Kernel started: 384a100d-a6ae-4eb2-aca2-ed9670205d9b
[App] Saving file at /beginner.ipynb
[App] Notebook beginner.ipynb is not trusted
```

“Run All” – Your CPU Load May Vary

localhost:8888/notebooks/beg

Activity Monitor (My Pro

TensorFlow 2 quickstart for beginners | TensorFlow Core

jupyter beginner (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

Code

```
0.11608636, 0.13060607, 0.05098354, 0.
dtype=float32)
```

Note: It is possible to bake this `tf.nn.softmax` in as the a
can make the model output more directly interpretable, this ap
and numerically stable loss calculation for all models when us

The `losses.SparseCategoricalCrossentropy` loss ta
loss for each example.

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

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

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

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

Out[35]: 2.153421

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

Process Name	% CPU	CPU Time
python3.7	1,466.1	17:56.02
CaptoHelper	55.4	36.89
http://localhost:8888	11.4	23.85
Activity Monitor	10.4	13.79
python3.7	3.6	3.57
AXVisualSupportAgent	1.2	5:30.61
PopClipHelper	1.1	12.66
com.wacom.IOManager	1.1	4:29.67
Safari	1.1	54.24
PopClip	0.6	1:35.57
Safari Networking	0.5	7.00
Core Sync	0.4	4:02.97
Microsoft PowerPoint	0.3	1:06.78
Adobe Desktop Service	0.3	8:24.65
Backup and Sync from Google	0.2	6:29.98
trustd	0.1	2:42.17
Microsoft Outlook	0.1	42.19
AdobelPCBroker	0.1	1:29.25
SystemUIServer	0.1	1:57.45
sharingd	0.1	1:57.45
mdworker_shared	0.1	0.25
AdobeCRDaemon	0.1	1:11.89
AdobeCRDaemon	0.0	1:16.53
AdobeCRDaemon	0.0	1:16.50
UserEventAgent	0.0	57.28

System: 69.41%
User: 30.59%
Idle: 0.00%

CPU LOAD

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