Teaching TensorFlow

## Getting Started

Direct your students to: <https://www.tensorflow.org/get_started/>

Both undergraduate and graduate students should focus on tf.keras and eager execution. As a more intuitive programming paradigm, Tf.keras is the recommended API for students. In cases involving very large data sets, students can take advantage of distributed training by converting a tf.keras model to an Estimator, as described in the tf.keras guide: <https://www.tensorflow.org/programmers_guide/keras>

For students, the best place to start is with the sequence of introductory keras notebooks on the getting started page. After that, have them try the eager notebooks on that same page.

## Assessing Student Learning

We recommend assigning project based homework. Minor TensorFlow API updates are released every couple of months, so we don't recommend focusing too heavily on testing syntax.

**Example assignments:**

* Undergrad students: train an image classifier on Imagenet, train a text classifier
* Graduate students: write a translation based model: <https://twitter.com/TensorFlow/status/1009579409781772295>

|  |  |
| --- | --- |
| **Some fun ML interactions:**   * Games on js.tensorflow.org * magenta.tensorflow.org/demos - Sketch * medium.com/tensorflow posenet | **Keep in Mind**  Deep Learning is a relatively new area, the best way to write code is still being discovered. To help stay up to date check out:   * TF blog: medium.com/tensorflow * TF twitter: twitter.com/tensorflow |

**TF is not just Python: js.tensorflow.org**

|  |
| --- |
| **Additional Resources** |
| <https://www.tensorflow.org/versions/r1.9/programmers_guide/keras> |
| For the latest way of doing things: <https://www.tensorflow.org/programmers_guide/eager> |
| <https://www.tensorflow.org/dev-summit/> |
| Slides you may use or repurpose: <https://goo.gl/uF9A1A> |