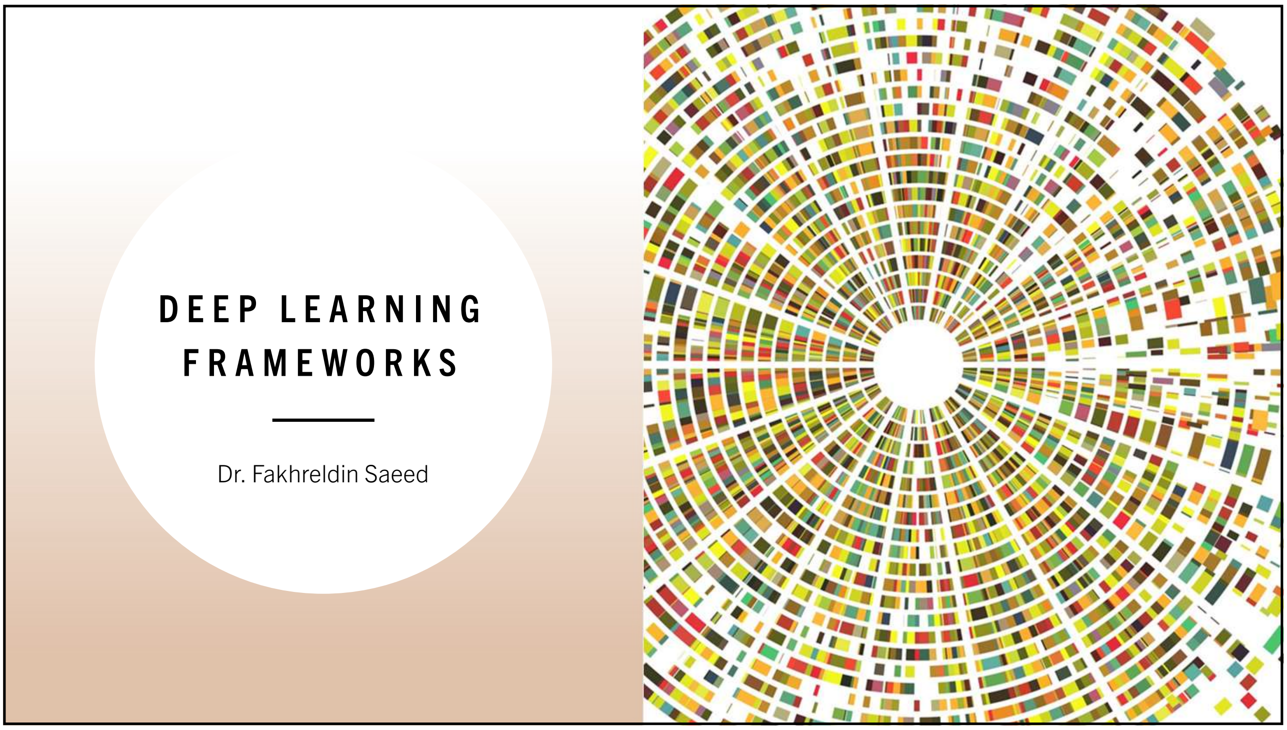
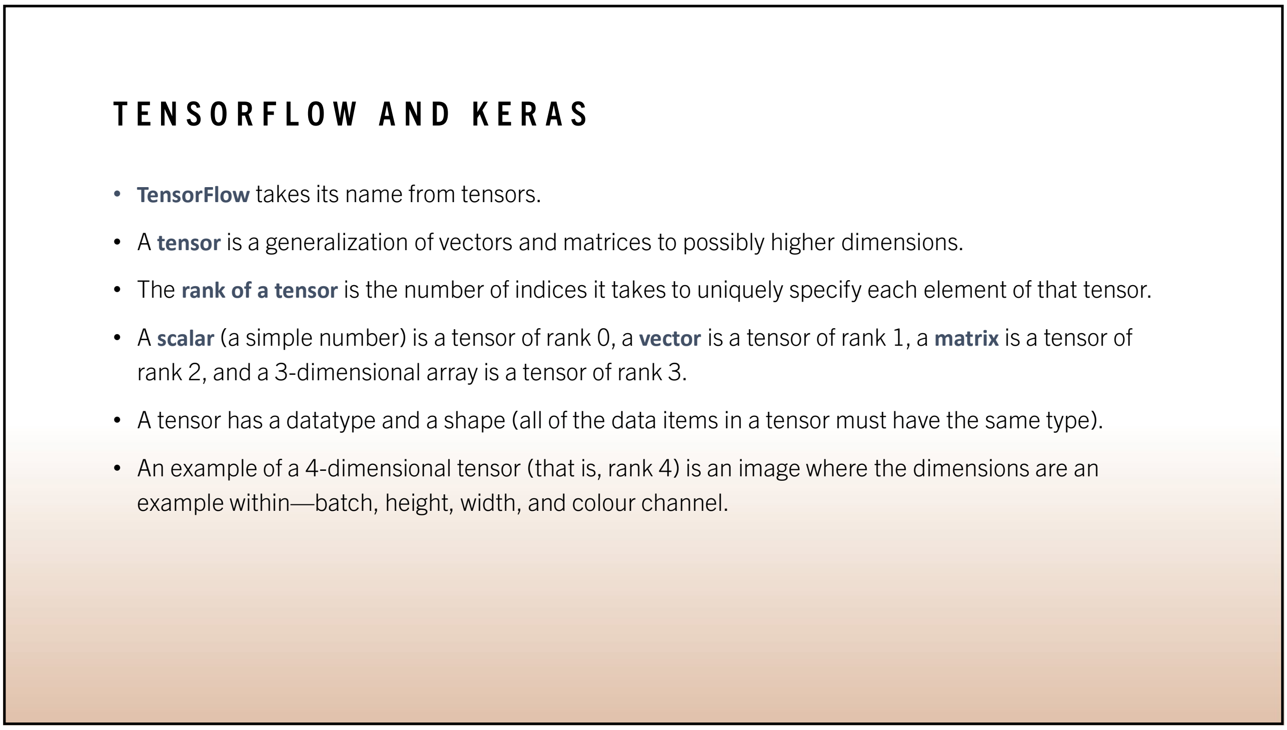


DEEP LEARNING FRAMEWORKS

Dr. Fakhreldin Saeed



TENSORFLOW AND KERAS

- **TensorFlow** takes its name from tensors.
- A **tensor** is a generalization of vectors and matrices to possibly higher dimensions.
- The **rank of a tensor** is the number of indices it takes to uniquely specify each element of that tensor.
- A **scalar** (a simple number) is a tensor of rank 0, a **vector** is a tensor of rank 1, a **matrix** is a tensor of rank 2, and a 3-dimensional array is a tensor of rank 3.
- A tensor has a datatype and a shape (all of the data items in a tensor must have the same type).
- An example of a 4-dimensional tensor (that is, rank 4) is an image where the dimensions are an example within—batch, height, width, and colour channel.

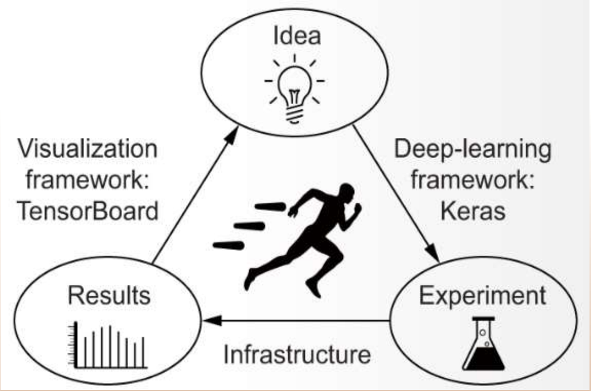
TENSORFLOW ECOSYSTEM

- TensorFlow.js is a collection of APIs that allow you to build and train models using either the low-level JavaScript linear algebra library or the high-level layers API. Hence, models can be trained and run in a browser.
- TensorFlow Lite is a lightweight version of TensorFlow for mobile and embedded devices.
- TensorFlow Hub is a library designed to foster the publication, discovery, and use of reusable modules of machine learning models.
- TensorFlow Extended (TFX) is a TensorFlow-based general-purpose machine learning platform. Libraries released to open source to date include TensorFlow Transform, TensorFlow Model Analysis, and TensorFlow Serving.
- TensorBoard is a suite of visualization tools supporting the understanding, debugging, and optimizing of TensorFlow programs.

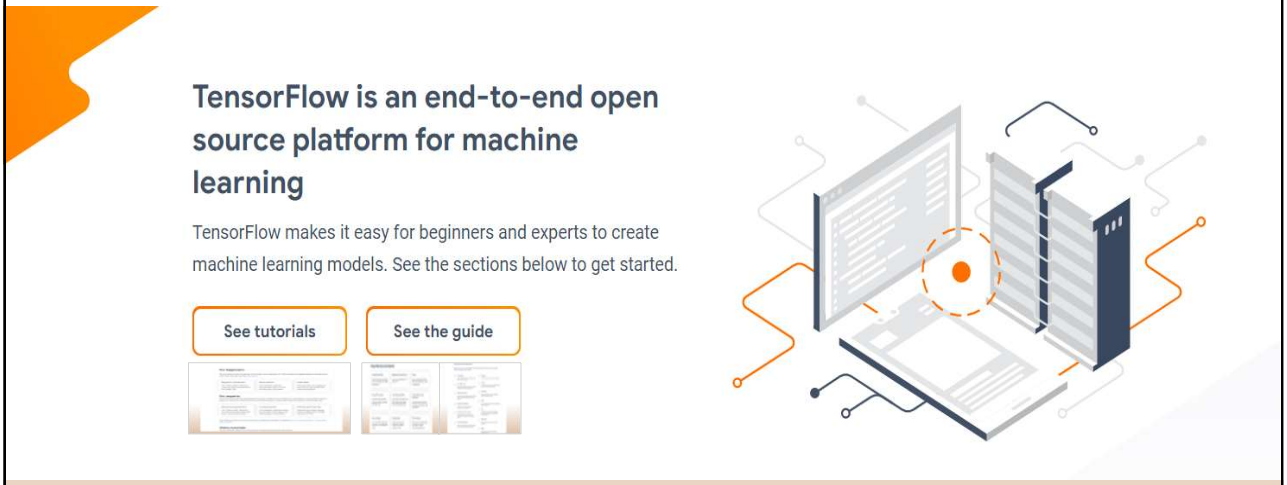
K Keras

Simple. Flexible. Powerful.

- Built on top of TensorFlow 2, Keras is an industry-strength framework that can scale to large clusters of GPUs or an entire TPU pod.


Idea
Deep-learning framework: Keras
Experiment
Infrastructure
Results
Visualization framework: TensorBoard

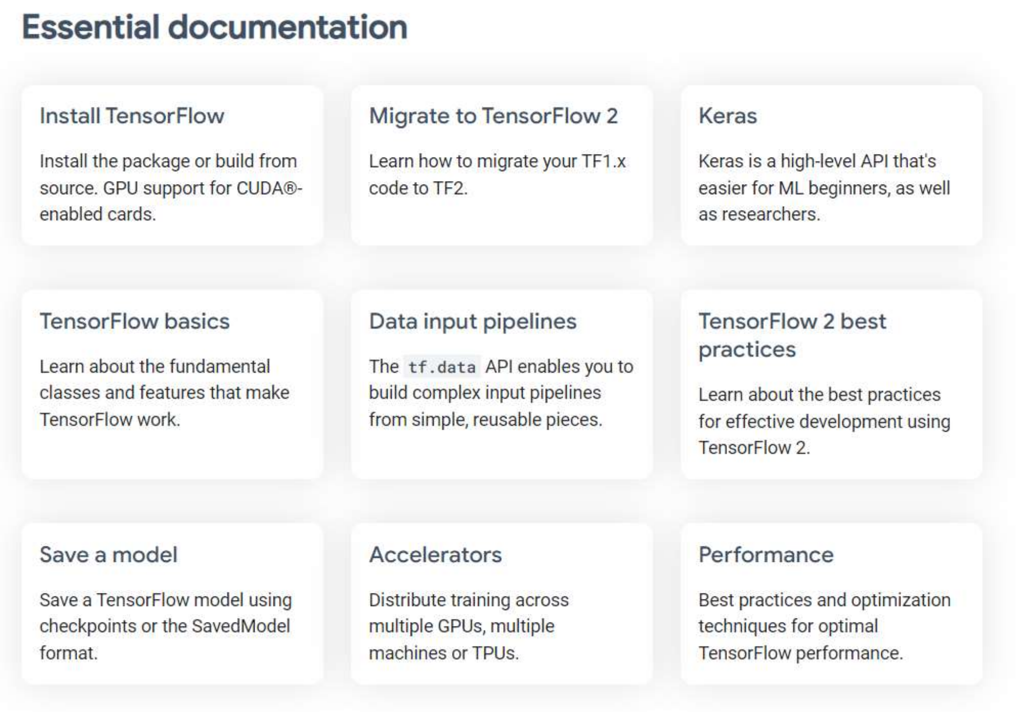
TENSORFLOW CORE



TensorFlow is an end-to-end open source platform for machine learning

TensorFlow makes it easy for beginners and experts to create machine learning models. See the sections below to get started.

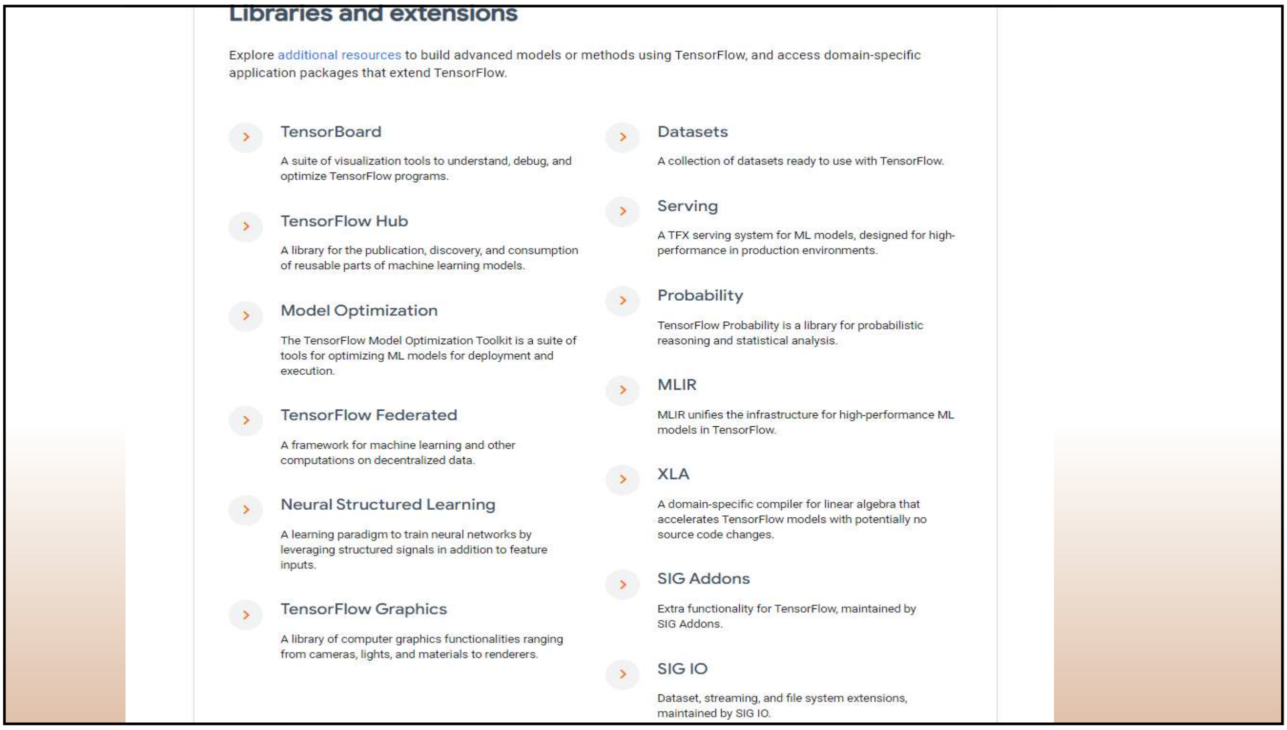
5



Essential documentation

Install TensorFlow Install the package or build from source. GPU support for CUDA®-enabled cards.	**Migrate to TensorFlow 2** Learn how to migrate your TF1.x code to TF2.	**Keras** Keras is a high-level API that's easier for ML beginners, as well as researchers.
TensorFlow basics Learn about the fundamental classes and features that make TensorFlow work.	**Data input pipelines** The `tf.data` API enables you to build complex input pipelines from simple, reusable pieces.	**TensorFlow 2 best practices** Learn about the best practices for effective development using TensorFlow 2.
Save a model Save a TensorFlow model using checkpoints or the SavedModel format.	**Accelerators** Distribute training across multiple GPUs, multiple machines or TPUs.	**Performance** Best practices and optimization techniques for optimal TensorFlow performance.

6



Libraries and extensions

Explore additional resources to build advanced models or methods using TensorFlow, and access domain-specific application packages that extend TensorFlow.

TensorBoard

A suite of visualization tools to understand, debug, and optimize TensorFlow programs.

TensorFlow Hub

A library for the publication, discovery, and consumption of reusable parts of machine learning models.

Model Optimization

The TensorFlow Model Optimization Toolkit is a suite of tools for optimizing ML models for deployment and execution.

TensorFlow Federated

A framework for machine learning and other computations on decentralized data.

Neural Structured Learning

A learning paradigm to train neural networks by leveraging structured signals in addition to feature inputs.

TensorFlow Graphics

A library of computer graphics functionalities ranging from cameras, lights, and materials to renderers.

Datasets

A collection of datasets ready to use with TensorFlow.

Serving

A TFX serving system for ML models, designed for high-performance in production environments.

Probability

TensorFlow Probability is a library for probabilistic reasoning and statistical analysis.

MLIR

MLIR unifies the infrastructure for high-performance ML models in TensorFlow.

XLA

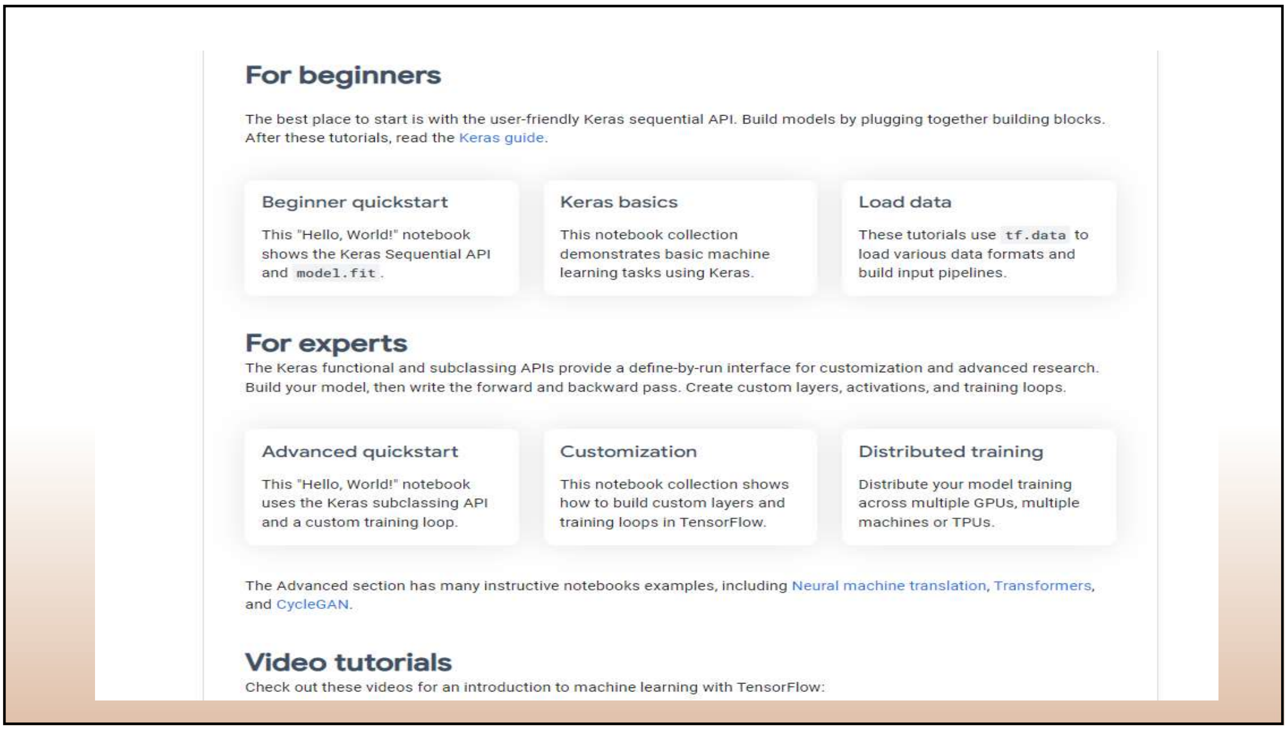
A domain-specific compiler for linear algebra that accelerates TensorFlow models with potentially no source code changes.

SIG Addons

Extra functionality for TensorFlow, maintained by SIG Addons.

SIG IO

Dataset, streaming, and file system extensions, maintained by SIG IO.



For beginners

The best place to start is with the user-friendly Keras sequential API. Build models by plugging together building blocks. After these tutorials, read the Keras guide.

Beginner quickstart

This "Hello, World!" notebook shows the Keras Sequential API and `model.fit`.

Keras basics

This notebook collection demonstrates basic machine learning tasks using Keras.

Load data

These tutorials use `tf.data` to load various data formats and build input pipelines.

For experts

The Keras functional and subclassing APIs provide a define-by-run interface for customization and advanced research. Build your model, then write the forward and backward pass. Create custom layers, activations, and training loops.

Advanced quickstart

This "Hello, World!" notebook uses the Keras subclassing API and a custom training loop.

Customization

This notebook collection shows how to build custom layers and training loops in TensorFlow.

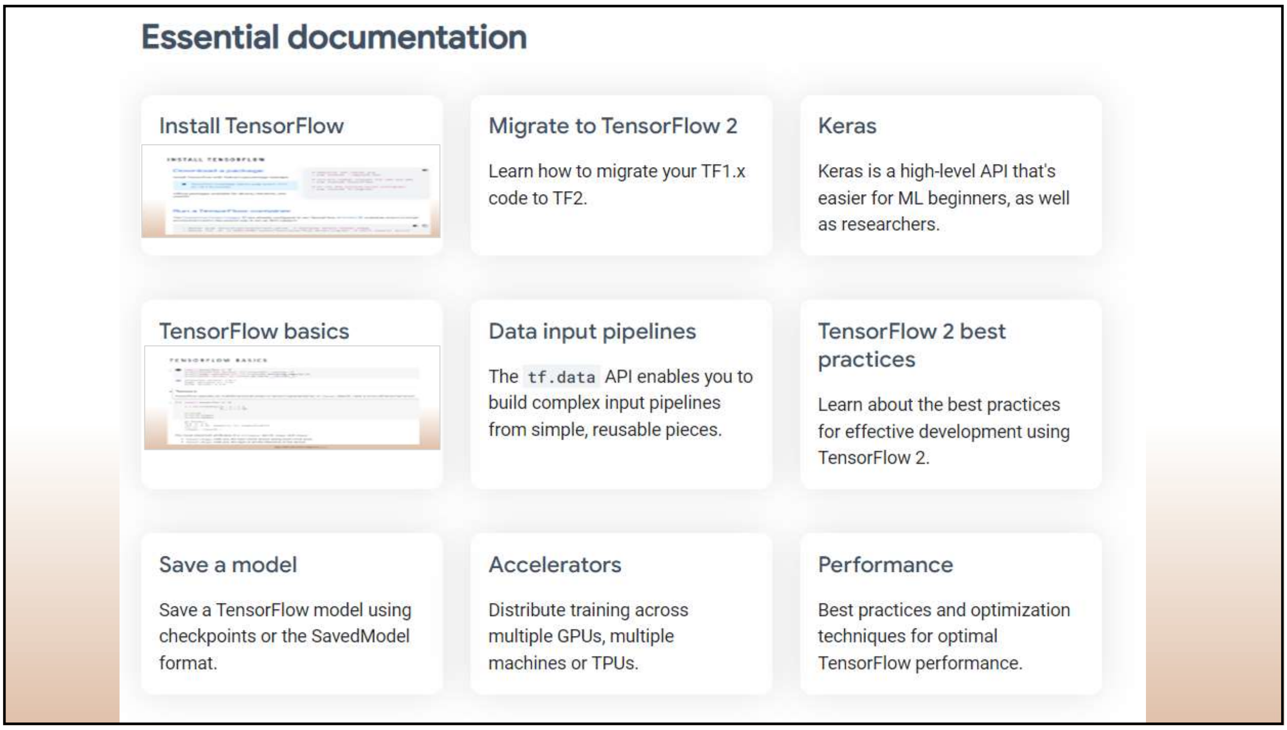
Distributed training

Distribute your model training across multiple GPUs, multiple machines or TPUs.

The Advanced section has many instructive notebooks examples, including Neural machine translation, Transformers, and CycleGAN.

Video tutorials

Check out these videos for an introduction to machine learning with TensorFlow:



Essential documentation

Install TensorFlow

Migrate to TensorFlow 2

Learn how to migrate your TF1.x code to TF2.

Keras

Keras is a high-level API that's easier for ML beginners, as well as researchers.

TensorFlow basics

Data input pipelines

The `tf.data` API enables you to build complex input pipelines from simple, reusable pieces.

TensorFlow 2 best practices

Learn about the best practices for effective development using TensorFlow 2.

Save a model

Save a TensorFlow model using checkpoints or the SavedModel format.

Accelerators

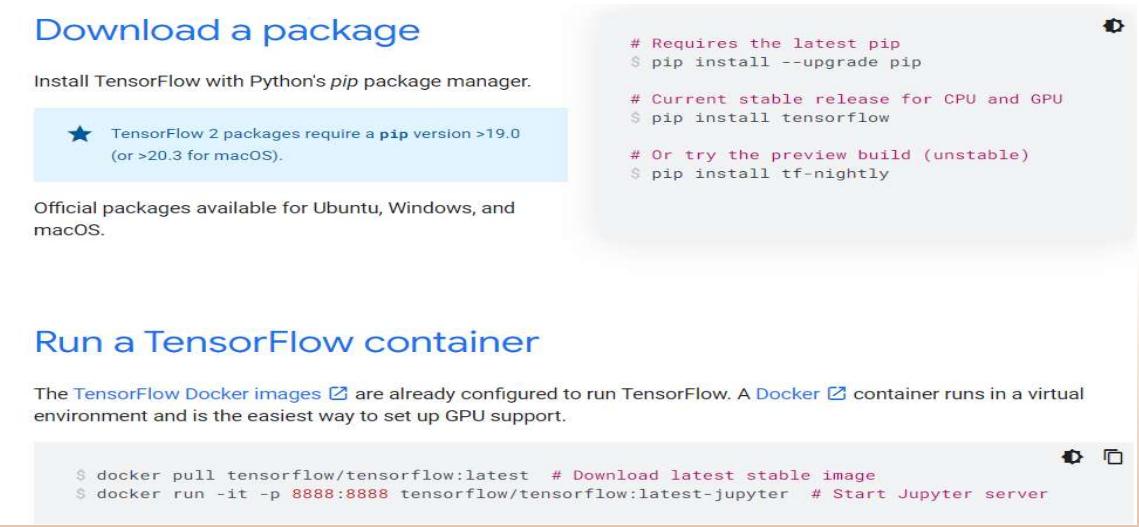
Distribute training across multiple GPUs, multiple machines or TPUs.

Performance

Best practices and optimization techniques for optimal TensorFlow performance.

9

INSTALL TENSORFLOW



Download a package

Install TensorFlow with Python's *pip* package manager.

★ TensorFlow 2 packages require a **pip** version >19.0 (or >20.3 for macOS).

Official packages available for Ubuntu, Windows, and macOS.

```
# Requires the latest pip
$ pip install --upgrade pip

# Current stable release for CPU and GPU
$ pip install tensorflow

# Or try the preview build (unstable)
$ pip install tf-nightly
```

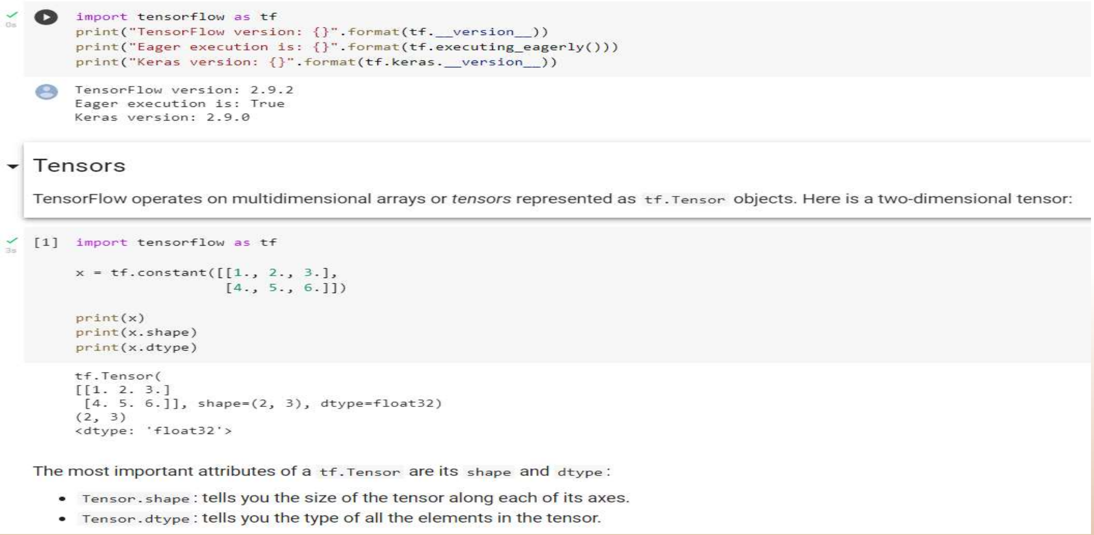
Run a TensorFlow container

The TensorFlow Docker images are already configured to run TensorFlow. A Docker container runs in a virtual environment and is the easiest way to set up GPU support.

```
$ docker pull tensorflow/tensorflow:latest  # Download latest stable image
$ docker run -it -p 8888:8888 tensorflow/tensorflow:latest-jupyter  # Start Jupyter server
```

10

TENSORFLOW BASICS



```
import tensorflow as tf
print("TensorFlow version: {}".format(tf.__version__))
print("Eager execution is: {}".format(tf.executing_eagerly()))
print("Keras version: {}".format(tf.keras.__version__))
```

```
TensorFlow version: 2.9.2
Eager execution is: True
Keras version: 2.9.0
```

Tensors

TensorFlow operates on multidimensional arrays or *tensors* represented as `tf.Tensor` objects. Here is a two-dimensional tensor:

```
[1] import tensorflow as tf

    x = tf.constant([[1., 2., 3.],
                     [4., 5., 6.]])

    print(x)
    print(x.shape)
    print(x.dtype)
```

```
tf.Tensor(
[[1. 2. 3.]
 [4. 5. 6.]], shape=(2, 3), dtype=float32)
(2, 3)
<dtype: 'float32'>
```

The most important attributes of a `tf.Tensor` are its `shape` and `dtype`:

- `Tensor.shape`: tells you the size of the tensor along each of its axes.
- `Tensor.dtype`: tells you the type of all the elements in the tensor.

https://www.tensorflow.org/guide/basics