Welcome to Deep Learning Online Bootcamp

Day 3: Introduction to tf.Keras and Tensors

dφ

Democratizing Data Science Learning

Learning Objectives

TensorFlow, Keras & tf.keras

Setting up TensorFlow

Tensors

TIP

You **DON'T need to memorize** the below jargons!

TensorFlow, Keras, tf.keras

What is TensorFlow?

- As you are aware by now, Tensorflow, is a deep learning library/framework open-sourced by Google.

- It has grown to become one of the most loved and widely adopted ML platforms in the world.

- The TensorFlow community includes:
  - Researchers
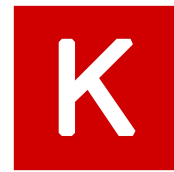 - Developers
  - Companies

What & Why Keras?

K

- Keras is a deep learning API (a tool that enables two applications to exchange data among each other) written in Python.

- It runs on top of TensorFlow.

- It was developed with a focus on enabling fast experimentation. Being able to go from idea to result as fast as possible is the key to doing good research.

What & Why tf.Keras?

tf.keras is a Tensorflow specific implementation of Keras. The name pre-fix "tf" itself says it is specific to TensorFlow framework.

Since tf.keras is a specific implementation of TensorFlow, it has some advantages and support for many TensorFlow specific features.

History/context: Not so long ago, on September 30th, 2019, Tensorflow 2.0 was launched. After the launch, developers started promoting something called **tf.Keras**.

Keras vs tf.Keras

We would suggest using TensorFlow 2.0 and tf.keras for future projects.

Not only you will enjoy the **added speed and optimization** of TensorFlow 2.0, but you'll also receive **new feature updates** — the latest release of the keras package (v2.3.0) will be the last release to support multiple backends and feature updates. Moving forward, the keras package will receive only bug fixes.
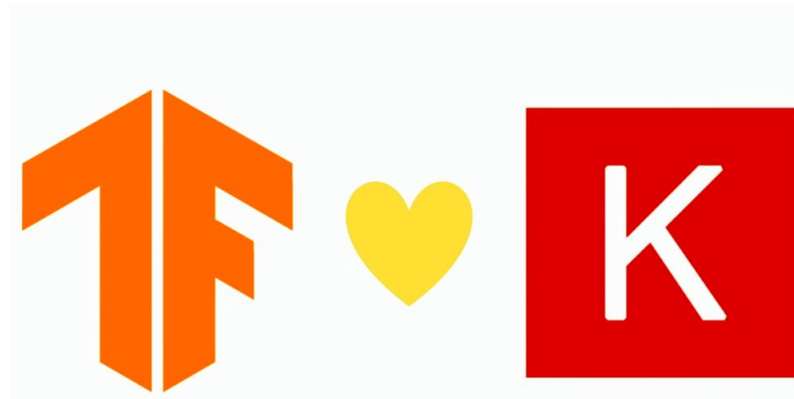
Tensorflow 2.0 Ecosystem & its use cases

Tensorflow 2.0 is not just a library, it's an Ecosystem. Not only you have the ability to train your own models using TensorFlow 2.0 and tf.keras, but you can now use:

- **TensorFlow Lite:** lightweight library for deploying models on mobile and embedded devices.

- **TensorFlow Extended:** end-to-end platform for preparing data,

TensorFlow
Ecosystem

From research

To production

Deployed everywhere

Empowering Responsible AI

Powered by the community

Now that we know why we're using Tensorflow 2.0 and tf.keras, let's get started with it!

Importing Tensorflow

Even if Tensorflow has been installed, we'll always need to import it before use.

Within a Jupyter or Colab Notebook, you can import TensorFlow with a simple import statement. Just how we used np for numpy or pd for Pandas, we'll be using tf for importing Tensorflow as shown below:

```
import tensorflow as tf
```

On running this cell, TensorFlow will be imported and available for your use!

Check the version of Tensorflow

- Open a Jupyter Notebook/Google Colab

- In a cell type and execute the following command:

```
import tensorflow as tf
print(tf.__version__)
```

- Make sure the displayed version starts with 2 i.e ensure Tensorflow 2 has been installed.

- Google Colab uses TensorFlow 2 by default.
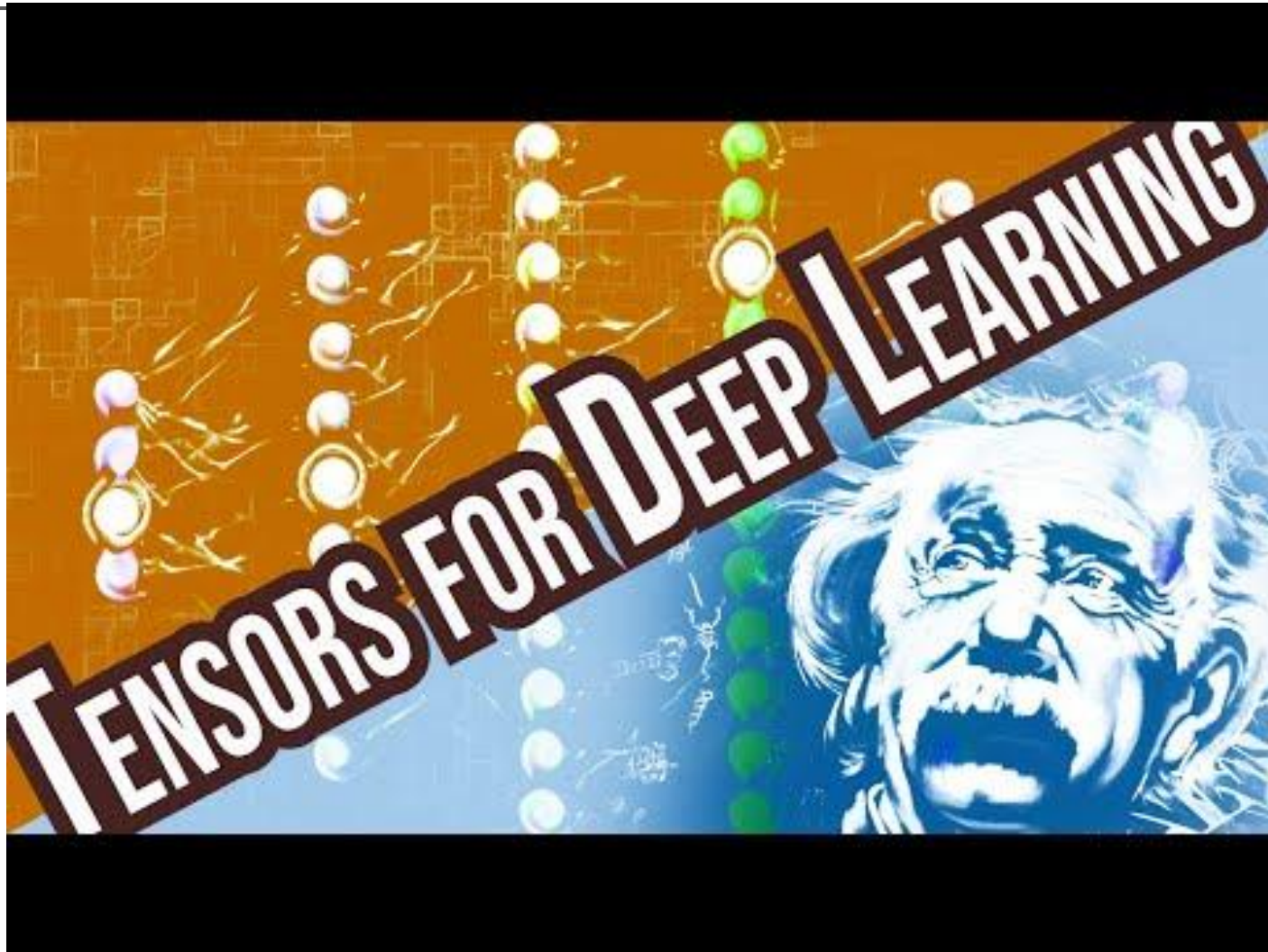
dφ
Democratizing Data Science Learning

Tensors

Why should we study about Tensors?

Tensors are the primary data structures used by neural networks (the building blocks of Deep Learning).
And they are rather fascinating as well!

In neural networks transformations, input, output etc are performed via tensors.

What are Tensors?

What are Tensors?

Tensors are multi-dimensional arrays with a uniform type (called a dtype).

If you're familiar with NumPy, tensors are (kind of) like np.arrays.

All tensors are immutable like Python numbers and strings i.e you can never update the contents of a tensor, only create a new one.
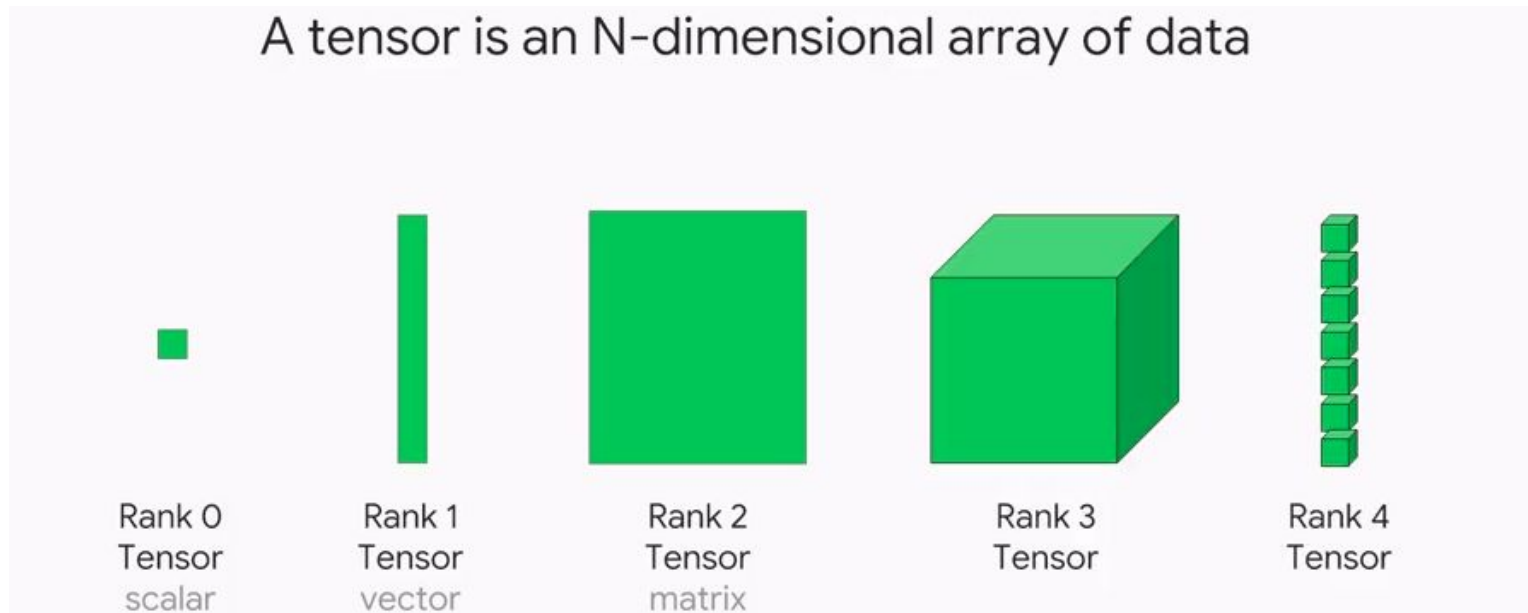
What are Tensors?

As you can see in the figure, the first 3 ranks of Tensors have special names that you might have encountered in the past.

- Rank 0 tensor - **Scalar**
- Rank 1 tensor - **Vector**
- Rank 2 tensor - **Matrix**

The tensors with ranks more than 2 are simply called ' Tensor '.

A tensor is an N-dimensional array of data

Rank 0 Tensor	Rank 1 Tensor	Rank 2 Tensor	Rank 3 Tensor	Rank 4 Tensor
scalar	vector	matrix		

Understanding Tensor

```
Tensor("Const:0", shape=(), dtype=string)
```

Name

Shape

Data Type

To define Tensor

- Name to Tensor
- Define shape like x axis, y axis...
- Data type

How is the tensor shape important?

Our networks operate on tensors, after all, and this is why understanding a tensor's shape and the available reshaping operations are super important.

Let's learn about the shape and the methods of reshaping tensors in the **next 2 videos.**

Note: These videos involve some **syntax of PyTorch but not to worry**, the **concepts remain the same** regardless of the framework! We have added a practical implementation of it in tensorflow provided in the github repository.

Rank, Axes and Shape

Reshape, Squeeze and Flatten

Additional Resources

On going through the below short resources from Tensorflow's official documentation, you'll notice the similarity between what you saw in the videos and what is implemented.

- Tensorflow Reshape:
  https://docs.w3cub.com/tensorflow~python/tf/reshape/

- Tensorflow Squeeze:
  https://docs.w3cub.com/tensorflow~python/tf/squeeze/

dφ

Democratizing Data Science Learning

Let's Practice

Tensorflow Operations Notebook:

https://github.com/dphi-official/Deep_Learning_Bootcamp/tree/master/Tensor_Operations

Slides Download Link

You can download this module from the below link:

https://docs.google.com/presentation/d/1aFh443T4jAy6LVUv5BwPFhlig7gAyq0IahAb4r-kd5Y/edit?usp=sharing

That's it for the day. Thank you!

Feel free to post any queries in the #help channel on Slack