

1. Inroduction to TensorFlow By Mukesh Kumar Singh

December 17, 2024

1 TensorFlow 2.0 Introduction

- TensorFlow Install and Setup
- Representing Tensors
- Tensor Shape and Rank
- Types of Tensors

1.1 Installing TensorFlow

To install TensorFlow on your local machine you can use pip.

```
conda install tensorflow
```

If you have a CUDA enabled GPU you can install the GPU version of TensorFlow. You will also need to install some other software which can be found here: <https://www.tensorflow.org/install/gpu>

```
conda install tensorflow-gpu
```

1.2 Importing TensorFlow

```
[7]: import tensorflow as tf # now import the tensorflow module
      print(tf.version) # make sure the version is 2.x
```

```
<module 'tensorflow._api.v2.version' from
'/home/test/anaconda3/envs/DNN/lib/python3.12/site-
packages/tensorflow/_api/v2/version/__init__.py'>
```

1.2.1 Tensors

“A tensor is a generalization of vectors and matrices to potentially higher dimensions. Internally, TensorFlow represents tensors as n-dimensional arrays of base datatypes.” (<https://www.tensorflow.org/guide/tensor>)

Tensors are a fundemental apsect of TensorFlow. They are the main objects that are passed around and manipulated throughout the program. Each tensor represents a partially defined computation that will eventually produce a value. TensorFlow programs work by building a graph of Tensor objects that details how tensors are related. Running different parts of the graph allow results to be generated.

Each tensor has a data type and a shape.

Data Types Include: float32, int32, string and others.

Shape: Represents the dimension of data.

Just like vectors and matrices tensors can have operations applied to them like addition, subtraction, dot product, cross product etc.

1.2.2 Creating Tensors

Below is an example of how to create some different tensors.

Simply define the value of the tensor and the datatype and you are good to go! Usually we deal with tensors of numeric data, it is quite rare to see string tensors.

For a full list of datatypes please refer to the following guide.

https://www.tensorflow.org/api_docs/python/tf/dtypes/DType?version=stable

```
[6]: string = tf.Variable("this is a string", tf.string)
      number = tf.Variable(324, tf.int16)
      floating = tf.Variable(3.567, tf.float64)
```

```
2024-12-17 17:30:53.121195: E
external/local_xla/xla/stream_executor/cuda/cuda_driver.cc:152] failed call to
cuInit: INTERNAL: CUDA error: Failed call to cuInit: UNKNOWN ERROR (303)
```

1.2.3 Rank/Degree of Tensors

Another word for rank is degree, these terms simply mean the number of dimensions involved in the tensor. What we created above is a *tensor of rank 0*, also known as a scalar.

Now we'll create some tensors of higher degrees/ranks.

```
[8]: rank1_tensor = tf.Variable(["Test"], tf.string)
      rank2_tensor = tf.Variable(["test", "ok"], ["test", "yes"], tf.string)
```

To determine the rank of a tensor we can call the following method.

```
[9]: tf.rank(rank2_tensor)
```

```
[9]: <tf.Tensor: shape=(), dtype=int32, numpy=2>
```

The rank of a tensor is directly related to the deepest level of nested lists. You can see in the first example `["Test"]` is a rank 1 tensor as the deepest level of nesting is 1. Where in the second example `[["test", "ok"], ["test", "yes"]]` is a rank 2 tensor as the deepest level of nesting is 2.

1.2.4 Shape of Tensors

The shape of a tensor is simply the number of elements that exist in each dimension. TensorFlow will try to determine the shape of a tensor but sometimes it may be unknown.

To get the shape of a tensor we use the shape attribute.

```
[10]: rank2_tensor.shape
```

```
[10]: TensorShape([2, 2])
```

1.2.5 Changing Shape

The number of elements of a tensor is the product of the sizes of all its shapes. There are often many shapes that have the same number of elements, making it convenient to be able to change the shape of a tensor.

The example below shows how to change the shape of a tensor.

```
[21]: tensor1 = tf.ones([1,2,3]) # tf.ones() creates a shape [1,2,3] tensor full of 1s
      ↪ ones
      # The number of elements in the reshaped tensor MUST match the number in the
      ↪ original
      print(tensor1)
```

```
tf.Tensor(
[[[1. 1. 1.]
  [1. 1. 1.]]], shape=(1, 2, 3), dtype=float32)
```

Now let's have a look at our different tensors.

```
[20]: tensor2 = tf.reshape(tensor1, [2,3,1]) # reshape existing data to shape [2,3,1]
      # The number of elements in the reshaped tensor MUST match the number in the
      ↪ original
      print(tensor2)
```

```
tf.Tensor(
[[[1.]
  [1.]
  [1.]]

 [[1.]
  [1.]
  [1.]]], shape=(2, 3, 1), dtype=float32)
```

```
[22]: tensor3 = tf.reshape(tensor2, [3, -1]) # -1 tells the tensor to calculate the
      ↪ size of the dimension in that place
      # this will reshape the tensor to [3,3]
      # The number of elements in the reshaped tensor MUST match the number in the
      ↪ original
      print(tensor3)
```

```
tf.Tensor(
[[1. 1.]
 [1. 1.]
 [1. 1.]], shape=(3, 2), dtype=float32)
```

1.2.6 Slicing Tensors

You may be familiar with the term “slice” in python and its use on lists, tuples etc. Well the slice operator can be used on tensors to select specific axes or elements.

When we slice or select elements from a tensor, we can use comma seperated values inside the set of square brackets. Each subsequent value references a different dimension of the tensor.

Ex: `tensor[dim1, dim2, dim3]`

```
[25]: # Creating a 2D tensor
matrix = [[1,2,3,4,5],
          [6,7,8,9,10],
          [11,12,13,14,15],
          [16,17,18,19,20]]

tensor = tf.Variable(matrix, dtype=tf.int32)
print(tf.rank(tensor))
print(tensor.shape)

tf.Tensor(2, shape=(), dtype=int32)
(4, 5)
```

```
[27]: # Now lets select some different rows and columns from our tensor

three = tensor[0,2] # selects the 3rd element from the 1st row
print(three) # -> 3

tf.Tensor(3, shape=(), dtype=int32)
```

```
[28]: row1 = tensor[0] # selects the first row
print(row1)

tf.Tensor([1 2 3 4 5], shape=(5,), dtype=int32)
```

```
[29]: column1 = tensor[:, 0] # selects the first column
print(column1)

tf.Tensor([ 1  6 11 16], shape=(4,), dtype=int32)
```

```
[32]: row2and4 = tensor[1::2] # selects second and fourth row
print(row2and4)

tf.Tensor(
[[ 6  7  8  9 10]
 [16 17 18 19 20]], shape=(2, 5), dtype=int32)
```

```
[31]: column_1_in_row_2_and_3 = tensor[1:3, 0]
print(column_1_in_row_2_and_3)

tf.Tensor([ 6 11], shape=(2,), dtype=int32)
```

1.2.7 Types of Tensors

There are different types of tensors.

- Variable
- Constant
- Placeholder
- SparseTensor

With the exception of `Variable` all these tensors are immutable, meaning their value may not change during execution.

For now, it is enough to understand that we use the `Variable` tensor when we want to potentially change the value of our tensor.

#Sources Most of the information is taken directly from the TensorFlow website which can be found below.

<https://www.tensorflow.org/guide/tensor>