002 TensorFlow intro

TensorFlow 2.0 Introduction

In this notebook you will be given an interactive introduction to TensorFlow 2.0. We will walk through the following topics within the TensorFlow module:

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

If you'd like to follow along without installing TensorFlow on your machine you can use **Google Collaboratory**. Collaboratory is a free Jupyter notebook environment that requires no setup and runs entirely in the cloud.

Installing TensorFlow

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

```
pip install tensorflow
Copy
```

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](https://www.tensorflow.org/install/gpu

```
pip install tensorflow-gpu
Copy
```

Importing TensorFlow

The first step here is going to be to select the correct version of TensorFlow from within collabratory!

```
%tensorflow_version 2.x  # this line is not required unless you are in a notebook
Copy
```

```
import tensorflow as tf  # now import the tensorflow module
print(tf.version)  # make sure the version is 2.x
Copy
```

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)

It should't surprise you that tensors are a fundemental apsect of TensorFlow. They are the main objects that are passed around and manipluated throughout the program. Each tensor represents a partialy 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.

In the next sections we will discuss some different properties of tensors. This is to make you more familiar with how tensorflow represnts data and how you can manipulate this data.

Creating Tensors

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

You simply define the value of the tensor and the datatype and you are good to go! It's worth mentioning that 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

string = tf.Variable("this is a string", tf.string)
number = tf.Variable(324, tf.int16)
floating = tf.Variable(3.567, tf.float64)

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.

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.

tf.rank(rank2_tensor)

<tf.Tensor: shape=(), dtype=int32, numpy=2>

The rank of a tensor is direclty 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.

Shape of Tensors

Now that we've talked about the rank of tensors it's time to talk about the shape. 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.

rank2_tensor.shape

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 convient to be able to change the shape of a tensor.

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

tensor1 = tf.ones([1,2,3]) # tf.ones() creates a shape [1,2,3] tensor full of ones
tensor2 = tf.reshape(tensor1, [2,3,1]) # reshape existing data to shape [2,3,1]
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 numer of elements in the reshaped tensor MUST match the number in the original

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

print(tensor1)
print(tensor2)
print(tensor3)
# Notice the changes in shape

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 refrences a different dimension of the tensor.

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

I've included a few examples that will hopefully help illustrate how we can manipulate tensors with the slice operator.

```
# 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)

# 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

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

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

row_2_and_4 = tensor12  # selects second and fourth row
print(row2and4)

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

Types of Tensors

Before we go to far, I will mention that there are diffent types of tensors. These are the most used and we will talk more in depth about each as they are used.

- Variable
- Constant
- Placeholder
- SparseTensor

With the execption of `Variable` all these tensors are immuttable, 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 direclty from the TensorFlow website which can be found below.

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