Tensorflow Essentials

AGENDA OF THE VIDEO

- What is tensorflow?
- What can we do with it? Real Life Use Cases of tensorflow..
- Why should i use tensorflow-keras combination instead of Numpy-scikit combination?
- Now, let us start coding basics of tensorflow...
  - Tensors and numpy arrays.
  - Shape, dimension and rank of tensor arrays.
  - Indexing in tensorflow arrays.
  - Reshaping tensors.
- Matrix Manipulation
  - Multiplication
  - Element Wise Operations
  - Transpose
  - Inverse and Determinant
- Task for you....

What is tensorflow?

TensorFlow

Tensorflow is an open source library by google which offers a unique way of solving various tasks in numerical computation and that way is called “data flow".

Now, by default it is not a Machine Learning Library as most of the people assume when they hear about tensorflow. Obviously, Machine Learning is something we can do in tensorflow but if you look at the core of this library, then it is just like “Numpy".

Tensorflow is made up of two words “tensor" and flow.

Tensor = > data (multi dimensional array)
Flow => Method of computation (Graph like flow)

What tensorflow is used for?

Mostly for Advanced ML and DL tasks such as:

1. Detection of objects in a Video.
2. Speech Recognition tasks.
3. Natural Language Processing Tasks.
4. Time series related such as sales forecasting
5. Image Recognition
6. and many more..

What's wrong with using Numpy with scikit learn?

Nothing wrong and in fact we use it a lot when we build models using scikit learn. In fact scikit learn has been built on top of Numpy to offer Machine Learning Tasks. Similarly, Keras is one of the many machine learning libraries which have been built on top of tensorflow.

Tensorflow instead of Numpy

There should be no comparison between these two libraries as each one of these have their own use cases but for larger and complex tasks, tensorflow wins hands down. But why is it?

The first reason is “Because of GPU Support Inbuilt”.

The other reasons that tensorflow is more favorable for doing ranges of tasks is it is really easy to productionize models. They can be executed on across platform like Android, Ios etc.

But there is so such support for these things in Numpy based scikit learn library.

These are some of the very few reasons that we use tensorflow rather than other libraries.

Let's code some tensors

1. What are tensors and difference between arrays and tensors?
2. What is the difference between the shapes, dimension, rank of tensors?
3. Indexing of tensors
4. Reshaping of tensors
5. Linear Algebra with tensors:
   a. Matrix Multiplication
   b. Transpose
   c. Inverse and Determinant

Tensor and numpy arrays

Tensors are Multi dimensional array representation in tensorflow library.

Now, you can think of tensors as just like numpy arrays. The main differences arises in speed of execution.

Shape, dimension, rank

All these terminologies are related.

Scalar	1d array	2d array	3d array
A Single Number	A List of Numbers	A List of List of Numbers	A List of List of List of Numbers
0 axis	1 axis	2 axis	3 axis
0 rank	1 rank	2 rank	3 rank
tensorflow.constant(3)	tensorflow.constant([2,3,4])	tensorflow.constant([[2, 3, 4], [1, 2, 4]])	tensorflow.constant([[[2,3,4], [ ], [ ]], [[ ], [ ], [ ]]])
Shape = ( )	Shape = (3,)	Shape = (2, 3)	Shape = (2, 3, 3)

Task for you..

You have the following set of tasks for you...

1. Make a tensor of shape (3, 4, 5) and call it a.
2. Make another tensor of shape (3,4) and call it b.
3. Now, try to add a and b. What is the output?
4. Make another tensor of shape (4, 10) and call it c.
5. Try to multiple b by c and store the result in d.
6. Now, what is the shape of d.
7. Now, declare a 200 * 200 random matrix of normal random numbers and call this matrix as e
8. Call tf.linalg.det and tf.linalg.inv on the above matrix e.
9. Finally, explore some new functions and linalg API and try to see if you can figure out their functionality.

Thank You