Tensorflow documentation

What is Tensorflow??

Tensorflow is an end-to-end platform for Machine Learning Models. It can be run on GPU and TCP. Using Tensorflow we will be able to access the pre-built deep learning models. This give the whole stack

- *Preprocess data*
- *Model data*
- *Deploy data*

Why Tensorflow??

Easy model building

Easily build and deploy machine(Robust ML production anywhere)

Powerful experimentation for research

Virtually anyone can solve in a with no initial investment, problems that would have required an engineering team working for a quarter and 20k dollars in hardware in 2014.

What is a GPU/TPU:

GPU- *Graphical Processing Unit*

TPU- *Tensor Processing Unit*

Tensor Processing Unit (**TPU**) is an AI accelerator application-specific integrated circuit (ASIC) developed by Google for neural network machine learning, using Google's own TensorFlow software.

What is a tensor?
Input which can be audio, video or image..
That is converted to a numerical encoding.
To convert this we create a tensor. This tensor is passed to the neural network and it outputs another tensor and then converted to something that is understandable to a human sensation.

On elaborating what a tensor is…
Vector we have learnt that is a physical quantity that has both the direction and the magnitude. The vector can also be represented as an area.
For a vector there is only one basis vector that can be called a tensor of a 1 length.
And scalars can be called the tensors of length 0.

Random tensors are tensors of some arbitrary size which contain random number

Tensorflow workflow
Get the data ready → Build or pick a pretrained model → Fit the model to the data and predict → Evaluate the model → Improve through experimentation → Save and reload your trained model

Functions I know n tensorflow:
tf.__version__ -> for checking the version of tensorflow using

tf.constant(value, dtype=None, shape=None, name='Const') ->
Creates a constant tensor from a tensor-like object.
.https://www.tensorflow.org/api_docs/python/tf/constant

<variable>.ndim -> checking the dimension of the tensor.

tf.Variable(
  initial_value=None,
  trainable=None,
  validate_shape=True,
  caching_device=None,
  name=None,
  variable_def=None,
  dtype=None,
  import_scope=None,
  constraint=None,
  synchronization=`tf.VariableSynchronization.AUTO`,
  aggregation=`tf.compat.v1.VariableAggregation.NONE`,
  shape=None,
  experimental_enable_variable_lifting=True
) -> https://www.tensorflow.org/api_docs/python/tf/Variable

<tensor variable>.assign():
<tensor variable>.assign(self, value, use_locking=False, name=None, read_value=True)->
Assigns a new value to the variable.
  This is essentially a shortcut for `assign(self, value)`.
  Args:
    value: A `Tensor`. The new value for this variable.
    use_locking: If `True`, use locking during the assignment.
    name: The name of the operation to be created
    read_value: if True, will return something which evaluates to the new
      value of the variable; if False will return the assign op.
  Returns:
    The updated variable. If `read_value` is false, instead returns None in
    Eager mode and the assign op in graph mode.

Random tensors:
Random tensors are tensors of some arbitrary size which contain a random number.
It is necessary because normally the tensors are initialized to random tensors.

Normal distribution:
Normal distribution, also known as the Gaussian distribution, is a probability distribution that is symmetric about the mean, showing that data near the mean are more frequent in occurrence than data far from the mean.

Uniform Distribution:
uniform distribution refers to a type of probability distribution in which all outcomes are equally likely.

```
from tensorflow._api.v2 import random
```
->import syntax for importing random module
```
tf.random.Generator(
    copy_from=None, state=None, alg=None
)->
```
https://www.tensorflow.org/api_docs/python/tf/random/normal

```
tf.random.normal(
    shape,
    mean=0.0,
    stddev=1.0,
    dtype=tf.dtypes.float32,
    seed=None,
    name=None
) -> Outputs random values from a normal distribution.
```
https://www.tensorflow.org/api_docs/python/tf/random/normal

random.set_seed(seed):
tf.random.set_seed(
   seed
)

https://www.tensorflow.org/api_docs/python/tf/random/set_seed
Rules:
Its interactions with operation-level seeds is as follows:

If neither the global seed nor the operation seed is set: A randomly picked seed is used for this op.
If the global seed is set, but the operation seed is not: The system deterministically picks an operation seed in conjunction with the global seed so that it gets a unique random sequence. Within the same version of tensorflow and user code, this sequence is deterministic. However across different versions, this sequence might change. If the code depends on particular seeds to work, specify both global and operation-level seeds explicitly.
If the operation seed is set, but the global seed is not set: A default global seed and the specified operation seed are used to determine the random sequence.
If both the global and the operation seed are set: Both seeds are used in conjunction to determine the random sequence.

It looks like if we want our shuffled tensors to be in the same order, we've got to use the global level random seed as well as operation level random seed:
>Rule 4:"If both the global and operation seed are set: Both seeds are used in conjunction to determine the random sequence."