tf.train.exponential_decay

```
exponential_decay(
    learning_rate,
    global_step,
    decay_steps,
    decay_rate,
    staircase=False,
    name=None
)
```

Defined in `tensorflow/python/training/learning_rate_decay.py`.

See the guide: Training > Decaying the learning rate

Applies exponential decay to the learning rate.

When training a model, it is often recommended to lower the learning rate as the training progresses. This function applies an exponential decay function to a provided initial learning rate. It requires a `global_step` value to compute the decayed learning rate. You can just pass a TensorFlow variable that you increment at each training step.

The function returns the decayed learning rate. It is computed as:

```
decayed_learning_rate = learning_rate *
                        decay_rate ^ (global_step / decay_steps)
```

If the argument `staircase` is `True`, then `global_step / decay_steps` is an integer division and the decayed learning rate follows a staircase function.

Example: decay every 100000 steps with a base of 0.96:

```
...
global_step = tf.Variable(0, trainable=False)
starter_learning_rate = 0.1
learning_rate = tf.train.exponential_decay(starter_learning_rate, global_step,
                                           100000, 0.96, staircase=True)
# Passing global_step to minimize() will increment it at each step.
learning_step = (
    tf.train.GradientDescentOptimizer(learning_rate)
    .minimize(...my loss..., global_step=global_step)
)
```

Args:

- `learning_rate`: A scalar `float32` or `float64` `Tensor` or a Python number. The initial learning rate.
- `global_step`: A scalar `int32` or `int64` `Tensor` or a Python number. Global step to use for the decay computation. Must not be negative.
- `decay_steps`: A scalar `int32` or `int64` `Tensor` or a Python number. Must be positive. See the decay computation above.
- `decay_rate`: A scalar `float32` or `float64` `Tensor` or a Python number. The decay rate.
- `staircase`: Boolean. If `True` decay the learning rate at discrete intervals

- `name`: String. Optional name of the operation. Defaults to 'ExponentialDecay'.

Returns:

A scalar `Tensor` of the same type as `learning_rate`. The decayed learning rate.

Raises:

- `ValueError`: if `global_step` is not supplied.
