tf.random_gamma

```
random_gamma(
    shape,
    alpha,
    beta=None,
    dtype=tf.float32,
    seed=None,
    name=None
)
```

Defined in `tensorflow/python/ops/random_ops.py`.

See the guide: Constants, Sequences, and Random Values > Random Tensors

Draws `shape` samples from each of the given Gamma distribution(s).

`alpha` is the shape parameter describing the distribution(s), and `beta` is the inverse scale parameter(s).

Example:

samples = tf.random_gamma([10], [0.5, 1.5]) # samples has shape [10, 2], where each slice [:, 0] and [:, 1] represents # the samples drawn from each distribution

samples = tf.random_gamma([7, 5], [0.5, 1.5]) # samples has shape [7, 5, 2], where each slice [:, :, 0] and [:, :, 1] # represents the 7x5 samples drawn from each of the two distributions

samples = tf.random_gamma([30], [[1.],[3.],[5.]], beta=[[3., 4.]]) # samples has shape [30, 3, 2], with 30 samples each of 3x2 distributions.

Note: Because internal calculations are done using `float64` and casting has `floor` semantics, we must manually map zero outcomes to the smallest possible positive floating-point value, i.e., `np.finfo(dtype).tiny`. This means that `np.finfo(dtype).tiny` occurs more frequently than it otherwise should. This bias can only happen for small values of `alpha`, i.e., `alpha << 1` or large values of `beta`, i.e., `beta >> 1`.

Args:

- `shape`: A 1-D integer Tensor or Python array. The shape of the output samples to be drawn per alpha/beta-parameterized distribution.
- `alpha`: A Tensor or Python value or N-D array of type `dtype`. `alpha` provides the shape parameter(s) describing the gamma distribution(s) to sample. Must be broadcastable with `beta`.
- `beta`: A Tensor or Python value or N-D array of type `dtype`. Defaults to 1. `beta` provides the inverse scale parameter(s) of the gamma distribution(s) to sample. Must be broadcastable with `alpha`.
- `dtype`: The type of alpha, beta, and the output: `float16`, `float32`, or `float64`.
- `seed`: A Python integer. Used to create a random seed for the distributions. See `tf.set_random_seed` for behavior.
- `name`: Optional name for the operation.

Returns:

- `samples`: a `Tensor` of shape `tf.concat(shape, tf.shape(alpha + beta))` with values of type `dtype`.

Last updated November 2, 2017.