tf.contrib.bayesflow.entropy.renyi_alpha

```
renyi_alpha(
    step,
    decay_time,
    alpha_min,
    alpha_max=0.99999,
    name='renyi_alpha'
)
```

Defined in `tensorflow/contrib/bayesflow/python/ops/entropy_impl.py`.

See the guide: BayesFlow Entropy (contrib) > Ops

Exponentially decaying `Tensor` appropriate for Renyi ratios.

When minimizing the Renyi divergence for `0 <= alpha < 1` (or maximizing the Renyi equivalent of elbo) in high dimensions, it is not uncommon to experience `NaN` and `inf` values when `alpha` is far from `1`.

For that reason, it is often desirable to start the optimization with `alpha` very close to 1, and reduce it to a final `alpha_min` according to some schedule. The user may even want to optimize using `elbo_ratio` for some fixed time before switching to Renyi based methods.

This `Op` returns an `alpha` decaying exponentially with step:

```
s(step) = (exp{step / decay_time} - 1) / (e - 1)
t(s) = max(0, min(s, 1)),  (smooth growth from 0 to 1)
alpha(t) = (1 - t) alpha_min + t alpha_max
```

Args:

- `step`: Non-negative scalar `Tensor`. Typically the global step or an offset version thereof.
- `decay_time`: Positive scalar `Tensor`.
- `alpha_min`: `float` or `double` `Tensor`. The minimal, final value of `alpha`, achieved when `step >= decay_time`
- `alpha_max`: `Tensor` of same `dtype` as `alpha_min`. The maximal, beginning value of `alpha`, achieved when `step == 0`
- `name`: A name to give this `Op`.

Returns:

- `alpha`: A `Tensor` of same `dtype` as `alpha_min`.

Last updated November 2, 2017.