tf.contrib.bayesflow.csiszar_divergence.amari_alpha

```
amari_alpha(
    logu,
    alpha=1.0,
    self_normalized=False,
    name=None
)
```

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

The Amari-alpha Csiszar-function in log-space.

A Csiszar-function is a member of,

```
F = { f:R_+ to R : f convex }.
```

When `self_normalized = True`, the Amari-alpha Csiszar-function is:

```
f(u) = { -log(u) + (u - 1),      alpha = 0
       { u log(u) - (u - 1),     alpha = 1
       { [(u**alpha - 1) - alpha (u - 1)] / (alpha (alpha - 1)),    otherwise
```

When `self_normalized = False` the `(u - 1)` terms are omitted.

> **Warning:** when `alpha != 0` and/or `self_normalized = True` this function makes non-log-space calculations and may therefore be numerically unstable for `|logu| >> 0`.

For more information, see: A. Cichocki and S. Amari. "Families of Alpha-Beta-and GammaDivergences: Flexible and Robust Measures of Similarities." Entropy, vol. 12, no. 6, pp. 1532-1568, 2010.

Args:

- `logu`: `float`-like `Tensor` representing `log(u)` from above.
- `alpha`: `float`-like Python scalar. (See Mathematical Details for meaning.)
- `self_normalized`: Python `bool` indicating whether `f'(u=1)=0`. When `f'(u=1)=0` the implied Csiszar f-Divergence remains non-negative even when `p, q` are unnormalized measures.
- `name`: Python `str` name prefixed to Ops created by this function.

Returns:

- `amari_alpha_of_u`: `float`-like `Tensor` of the Csiszar-function evaluated at `u = exp(logu)`.

Raises:

- `TypeError`: if `alpha` is `None` or a `Tensor`.
- `TypeError`: if `self_normalized` is `None` or a `Tensor`.

Last updated November 2, 2017.