tf.contrib.bayesflow.csiszar_divergence.jensen_shannon

```
jensen_shannon(
    logu,
    self_normalized=False,
    name=None
)
```

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

The Jensen-Shannon Csiszar-function in log-space.

A Csiszar-function is a member of,

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

When `self_normalized = True`, the Jensen-Shannon Csiszar-function is:

```
f(u) = u log(u) - (1 + u) log(1 + u) + (u + 1) log(2)
```

When `self_normalized = False` the `(u + 1) log(2)` term is omitted.

Observe that as an f-Divergence, this Csiszar-function implies:

```
D_f[p, q] = KL[p, m] + KL[q, m]
m(x) = 0.5 p(x) + 0.5 q(x)
```

In a sense, this divergence is the "reverse" of the Arithmetic-Geometric f-Divergence.

This Csiszar-function induces a symmetric f-Divergence, i.e., `D_f[p, q] = D_f[q, p]`.

Warning: this function makes non-log-space calculations and may therefore be numerically unstable for `|logu| >> 0`.

For more information, see: Lin, J. "Divergence measures based on the Shannon entropy." IEEE Trans. Inf. Th., 37, 145-151, 1991.

Args:

- `logu`: `float`-like `Tensor` representing `log(u)` from above.
- `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:

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

Last updated November 2, 2017.