Specification

- Softmax(input,axis)= Exp(input) / ReduceSum(Exp(Input), axis=axis, keepdims=1)

$$s(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

- Runtime errors Operator SoftMax may overflow (resp. underflow) for large (resp. small) input input values. In that case, the operator will return a nan.

  Every implementation should provide a correct (or correctly approximated) result without any error if $\forall\ 1 \leq j \leq K.\ z_j \leq 80$ for $K \leq 4000$.

  Better: K*exp(max(z_i)) < FLT_MAX
  too much connected to a naïve implementation in float32?

- A non-naïve implementation should provide wider conditions for which it delivers a correctly approximated result. It should also provide the list of possible errors as well as conditions that produce elements or subsets of this list.
- Any implementation may return an error or a correct result outside the case K*exp(max(z_i)) < FLT_MAX

Robust implementation answer

The implementation that computes

- $z_{max} = max_{1 \le j \le K}(z_j)$
- s(z_i) as $$\text{SoftMax}(x_i) = \frac{e^{x_i - max(x)}}{\sum_{j=1}^{K} e^{x_j - max(x)}}$$

- provides a correctly approximated result if all z_j are finite.

Naïve implementation answer in double

- provides a correctly approximated result if $\forall\ 1 \leq j \leq K . z_j \leq 700$ for $K\ \ \leq 16000$, which contains the specification requirements.

The naive implementation produces results in [0, 1] or it raises a NaN - Infinity results are not possible -. More precisely,

- If $\forall\ 1 \leq j \leq K . z_j \leq 700$ and $K\ \ \leq 16000$, s(z_i) ∊ [0, 1] and it produces a correctly approximated result (an implementation error formula could be produced);
- If $z_i\ \geq 710$, result is NaN;
- If $(\exists\ 1 \leq j \leq K\ . j \neq i\ and\ z_j \geq 710)$ and if $z_i \leq 700$, result is 0 (underflow);
- In any other cases, result is in [0, 1] or NaN (partial specification)