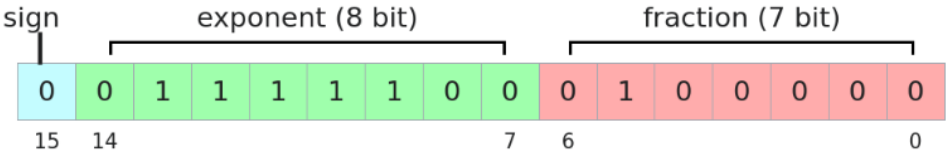
1. The `bfloat16` brain floating-point format consists of a sign bit, an 8-bit exponent followed by a 7-bit mantissa for a total of 16 bits.



https://en.wikipedia.org/wiki/Bfloat16_floating-point_format

The sign bit is set for a negative number, the exponent is stored as a biased integer with the offset being 127 and the mantissa is stored with a leading implicit bit similar to classical floating-point formats.

Find the decimal equivalents of

(i) `0 10000000 1001001`

(ii) `0 01111011 1001101`

Compute the error $\varepsilon = x^* - x$ where x^* is the `bfloat16` closest to

(iii) $x = 2.718$

(iv) $x = 12345.6$

(i) exponent $= 1\cdot 2^7 + 0\cdot 2^6 + 0\cdot 2^5 + \cdots + 0\cdot 2^0 = 128$

$e = 128 - 127 = 1$

$f = \frac{1}{2} + \frac{0}{2^2} + \frac{0}{2^3} + \frac{1}{2^4} + \frac{0}{2^5} + \frac{0}{2^6} + \frac{1}{2^7} = 0.5703125$

$x^* = (1+f)\times 2^e = 1.5703125 \times 2^1 = 3.140625$

(ii) exponent $= 0\cdot 2^7 + 1\cdot 2^6 + 1\cdot 2^5 + 1\cdot 2^4 + 1\cdot 2^3 + 0\cdot 2^2 + 1\cdot 2^1 + 1\cdot 2^0 = 123$

$e = 123 - 127 = -4$

$f = \frac{1}{2} + \frac{0}{2^2} + \frac{0}{2^3} + \frac{1}{2^4} + \frac{1}{2^5} + \frac{0}{2^6} + \frac{1}{2^7} = 0.60156\overline{25}$

$x^* = (1+f)\times 2^e = 1.6015625 \times 2^{-4} = 0.100097656\overline{25}$

(iii) $x = 2.718 = 2 + .718 = 2 + \frac{2(.718)}{2} = 2 + \frac{1.436}{2} = 2 + \frac{1}{2} + \frac{.436}{2}$

$= 1.2 + 0\cdot 2^0 + 1\cdot 2^{-1} + 0.872\cdot 2^{-2}$

$= 1\cdot 2 + 0\cdot 2^0 + 1\cdot 2^{-1} + 0\cdot 2^{-2} + 1.744\cdot 2^{-3}$

$= 1\cdot 2 + 0\cdot 2^0 + 1\cdot 2^{-1} + 0\cdot 2^{-2} + 1\cdot 2^{-3} + 1.488\cdot 2^{-4}$

$= 1\cdot 2 + 0\cdot 2^0 + 1\cdot 2^{-1} + 0\cdot 2^{-2} + 1\cdot 2^{-3} + 1\cdot 2^{-4} + 0.976\cdot 2^{-5}$

$= 1\cdot 2 + 0\cdot 2^0 + 1\cdot 2^{-1} + 0\cdot 2^{-2} + 1\cdot 2^{-3} + 1\cdot 2^{-4} + 0\cdot 2^{-5} + 1.952\cdot 2^{-6}$

7-bits so round 1.952 to 2 to get x^*

$x^* = 1\cdot 2 + 0\cdot 2^0 + 1\cdot 2^{-1} + 0\cdot 2^{-2} + 1\cdot 2^{-3} + 1\cdot 2^{-4} + 0\cdot 2^{-5} + 2\cdot 2^{-6}$

carry as $1\cdot 2^{-5}$

$= 1\cdot 2 + 0\cdot 2^0 + 1\cdot 2^{-1} + 0\cdot 2^{-2} + 1\cdot 2^{-3} + 1\cdot 2^{-4} + 1\cdot 2^{-5}$

$= 2.71875$

$\varepsilon = x^* - x = 2.71875 - 2.718 = 0.00075$

(iv) $x = 12345.6 = 2^{13} + 4153.6 = 2^{13} + 2^{12} + 57.6$

$= 1\cdot 2^{13} + 1\cdot 2^{12} + 0\cdot 2^{11} + 0\cdot 2^{10} + 0\cdot 2^9 + 0\cdot 2^8 + 0\cdot 2^7 + \frac{56}{2^6}\cdot 2^6$

$= 1\cdot 2^{13} + 1\cdot 2^{12} + 0\cdot 2^{11} + 0\cdot 2^{10} + 0\cdot 2^9 + 0\cdot 2^8 + 0\cdot 2^7 + 0.875\cdot 2^6$

7-bits so round 0.875 to 1 to get x^*

$x^* = 1\cdot 2^{13} + 1\cdot 2^{12} + 0\cdot 2^{11} + 0\cdot 2^{10} + 0\cdot 2^9 + 0\cdot 2^8 + 0\cdot 2^7 + 1\cdot 2^6$

$= 12352$

$\varepsilon = x^* - x = 12352 - 12345.6 = 6.4$

2. The Julia function

```
function f(x,y)
    return (abs(x)^3+abs(y)^3)^(1/3)
end
```

implements the calculation of

$$f(x,y) = \left(|x|^3 + |y|^3\right)^{1/3}.$$

Rewrite this function in a way that avoids overflow. Use your routine to compute $f(x,y)$ when

(i) $x = 5.0 \times 10^{120}$ and $y = 3.0 \times 10^{130}$

(ii) $x = 1.0$ and $y = 3.0 \times 10^{130}$

(iii) $x = 5.0 \times 10^{120}$ and $y = 1.0$

(iv) $x = 0.0$ and $y = 0.0$

Note the number 5.0×10^{120} may be expressed in Julia as `5.0e120`.

Case $|x| > |y|$ then compute

$$f(x,y) = |x| \left(1 + \left|\frac{y}{x}\right|^3\right)^{1/3}$$

Case $|x| < |y|$ then compute

$$f(x,y) = |y| \left(\left|\frac{x}{y}\right|^3 + 1\right)^{1/3}$$

Case $|x| = |y|$ then compute

$$f(x,y) = 2^{1/3} |x| .$$

Call the rescaled computation $g(x,y)$. Then

```
julia> function g(x,y)
           xabs=abs(x); yabs=abs(y)
           if xabs>yabs
               return xabs*(1+(yabs/xabs)^3)^(1/3)
           elseif xabs<yabs
               return yabs*((xabs/yabs)^3+1)^(1/3)
           end
           return 2^(1/3)*xabs
       end
g (generic function with 1 method)
```

(i)
```
julia> g(5e120,3e130)
3.0e130
```

(ii)
```
julia> g(1e0,3e130)
3.0e130
```

(iii)
```
julia> g(5e120,1e0)
5.0e120
```

(iv)
```
julia> g(0.0,0.0)
0.0
```

3. Perform a forward and backward error analysis for $f(x) = \sqrt[3]{x}$. You should find the relative error is reduced by a third when applying f.

Forward error analysis.

Suppose the relative error in the approximation x^* of x is δ. Thus $x^* = x(1+\delta)$. It follows for δ small that

$$f(x^*) = f(x(1+\delta)) \approx f(x) + \delta x f'(x)$$

$$= x^{1/3} + \delta x \, \frac{1}{3} x^{-2/3} = x^{1/3} + \frac{\delta}{3} x^{1/3} = f(x)\left(1 + \frac{\delta}{3}\right)$$

Therefore the relative error in $f(x^*)$ is $\frac{\delta}{3}$.

In other words

$$\frac{f(x^*) - f(x)}{f(x)} = \frac{f(x)\left(1 + \frac{\delta}{3}\right) - f(x)}{f(x)} = \frac{\delta}{3} = \frac{1}{3}\delta$$

Therefore the initial relative error of δ has been decreased by a factor of $1/3$.

Backwards error analysis

Let y^* be an approximation of $y = f(x)$. We want to compute the the relative error

$$\frac{y^* - y}{y}$$

Let $x^* = (y^*)^3$. Then $y^* = f(x^*)$ and $\frac{x^* - x}{x}$ may be identified as the relative residual error.

The condition number K_f is given as

$$K_f(x) = \frac{x f'(x)}{f(x)} = \frac{x \frac{1}{3} x^{-2/3}}{x^{1/3}} = \frac{1}{3}$$

Then

$$\frac{y^* - y}{y} = \frac{f(x^*) - f(x)}{f(x)} \approx K_f(x) \frac{x^* - x}{x} = \frac{1}{3} \frac{x^* - x}{x}$$

shows the relative error is y^* is $\frac{1}{3}$ of the relative residual error.