Tutorial 2 solutions

(1.) • From lecture, we know that a number, x, can be represented as follows in a floating point system:

$$x = \pm \sum_{k=0}^{p-1} \left(\frac{d_k}{\beta^k}\right) \beta^E$$

• Consider a floating point system with $\beta = 10$, $p = 4$, and $U = 3$

The largest number we can represent is 9.999×10^3
(we want to make each digit in the mantissa as large as possible, and make the exponent as large as possible too)

• Therefore, for a general floating point system, the largest number, x_*, must satisfy the following:

1) $d_k = \beta - 1 \quad \forall k$ (so each digit in the mantissa is as large as possible
2) sign $= +$
3) $E = U$ (ie. exponent is as large as possible)

Substituting these properties into our expression for x, we get:

$$X_* = \sum_{k=0}^{p-1} \left(\frac{\beta - 1}{\beta^k}\right) \beta^U$$

• Now we can use the following equality: $\sum_{n=0}^{N} \frac{a-1}{a^n} = a - a^{-N}$, where $N = p-1$ $n = k$ $a = \beta$

$$\therefore X_* = \left(\beta - \beta^{-(p-1)}\right) \beta^U$$

$$= \left(\beta - \beta^{-p+1}\right) \beta^U$$

$$= \left(\beta - \beta^{-p}\beta\right) \beta^U$$

$$= \beta\left(1 - \beta^{-p}\right) \beta^U$$

$\therefore$ the largest number we can represent is $X_* = (1 - \beta^{-p}) \beta^{U+1}$

(2.) Recall that an n-bit binary variable can take on 2^n unique values.

Now, if our floating point system can represent m unique values for the mantissa, and e unique values for the exponent, the total number of unique floating point numbers it can represent is:

$$m \cdot e \qquad \text{(ignoring the sign)}$$

If we use 8 bits to store the exponent, we can have 2^8 unique values for the exponent. Using 23 bits for the mantissa means it can take on 2^{23} unique values.

$\therefore$ we can represent $2^8 \cdot 2^{23} = 2^{31}$ unique floating point numbers

If we use 9 bits for the exponent and 22 for the mantissa, we can represent $2^9 \cdot 2^{22} = 2^{31}$ unique floating point numbers.

This means, both systems can represent the same number of floating point numbers. However, the question asks about normalized floats.

Recall that if the exponent is all zeroes, the number is de-normalized. Therefore, we should exclude this exponent from our calculations

$\therefore$ 8-bit exponent & 23-bit mantissa can represent $(2^8-1)(2^{23}) = 2^{31}-2^{23}$ unique normalized floats

9-bit exponent & 22-bit mantissa can represent $(2^9-1)(2^{22}) = 2^{31}-2^{22}$ normalized floats

$\therefore$ this system has more normalized floating point numbers

$a = 5.659 \times 10^4$ $\quad c = 9.337 \times 10^2$ $\quad B = 10 \quad p = 4 \quad L = -8 \quad U = 8$

$b = 5.629 \times 10^4$ $\quad d = 7.529 \times 10^{-1}$

3.

Using "round to even"

1. $a + b = 11.288 \times 10^4$ which gets rounded to 1.129×10^5

2. $a - b = 0.03 \times 10^4$ which is represented as 3.000×10^2

3. $c/d = 1240.138133 \xrightarrow{\text{rounds to}} 1.240 \times 10^3$

4. $b \times d = 42380.741 \longrightarrow 4.238 \times 10^3$

5. $a + c = 57523.7 \longrightarrow 57520$

$b + d = 56290.7529 \longrightarrow 56290$

$(a+c) - (b+d) = 57520 - 56290 = 1230 \rightarrow 1.230 \times 10^3$

Better way of doing calculation:

$(a-b) + (c-d)$

$a - b \longrightarrow 3.000 \times 10^2$ (see part 2)

$b + d = 932.9471 \longrightarrow 9.329 \times 10^2$

$\therefore (a-b) + (c-d) = 1232.9 \rightarrow 1.233 \times 10^3$

notice that this is closer to the true value of 1.2329471×10^3 than this difference of squares

(4.) Rather than doing $x^2 - y^2$, do $(x+y)(x-y)$

Why is this better?

Consider the following case, where x and y are very similar in size:

$y = x + \varepsilon$, where $\varepsilon << x$, but there is still enough precision to represent y exactly

ie. the difference between x and y (ε) is much smaller than x or y, but the floating point system has just enough precision to represent y exactly (so $x + \varepsilon \neq x$)

If we calculate y^2, we get $y^2 = (x+\varepsilon)^2 = x^2 + 2x\varepsilon + \varepsilon^2$

since $\varepsilon << x$, $\varepsilon^2 <<< x^2$ so there is not enough precision to include it, so y^2 gets represented as $x^2 + 2x\varepsilon$

Aside: (For example, if $x = 1.0$, $\varepsilon = 1.0 \times 10^{-1}$ and $p = 2$:
$x + \varepsilon = 1.1$ which can be represented exactly
$x^2 + \varepsilon^2 = 1.01$ which is represented as 1.0 since $p = 2$)

Back to our question:
y^2 is represented as $x^2 + 2x\varepsilon$, so $x^2 - y^2 = -2x\varepsilon$

Now let's do $(x+y)(x-y)$ instead:

$x + y = x + (x+\varepsilon) = 2x + \varepsilon$

$x - y = x - (x+\varepsilon) = -\varepsilon$

$\therefore (x+y)(x-y) = -2x\varepsilon - \varepsilon^2$

ε^2 and $2x\varepsilon$ are much closer in size than ε^2 and x^2, so there is (hopefully) enough precision to retain the ε^2 term

ex. going back to our example here:

$2x\varepsilon = 2.0 \times 10^{-1}$

$\varepsilon^2 = 1.0 \times 10^{-2}$

$\therefore 2x\varepsilon + \varepsilon^2 = 2.1 \times 10^{-1}$ which can be represented exactly so the ε^2 information is not lost