CS6490 Assignment 1

Vishal Vijay Devadiga (CS21BTECH11061)

Question

Understand the **range** and **precision** for **INT32** and single precision **FP32** data types.

Explain how FP32 can support a larger range as well as better precision than INT32, while both of them can represent 2ˆ32 unique numbers.

Illustrate with examples of numbers that can be represented in INT32 but not in FP32 and vice versa.

Answer

The **range** of a data type is the set of values that it can represent.

The **precision** of a data type is the smallest difference between two representable values.

Both **INT32** and **FP32** can represent 2ˆ32 unique numbers, since for each of the 32 bits, there are 2 possibilities (0 or 1), and hence 2ˆ32 possibilities in total.

Range of INT32 and FP32

The **INT32** data type is a 32-bit signed integer, which can represent numbers in the range -2^{31} to $2^{31} - 1$, that is from -2147483648 to 2147483647.

The **FP32** data type is a 32-bit floating point number, which can represent numbers in the range $-3.4028235 \times 10^{38}$ to 3.4028235×10^{38}.

As the numbers show, the range of **FP32** is much larger than that of **INT32**.

Precision of INT32 and FP32

The **INT32** data type has a precision of 1, as the difference between any two consecutive representable numbers is 1.

For **FP32**, the precision depends on the magnitude of the number. For numbers close to 0, that is, the exponent is small/negative, the precision is higher, and for numbers farther from 0, that is, the exponent is large, the precision is lower. This is because of the way floating point numbers are represented, with a sign bit, an exponent and a mantissa. The exponent and mantissa are used to represent the magnitude and precision of the number.

So when the magnitude of the number is small, FP32 can have a high precision, but when the magnitude is large, the precision is lower.

FP32 vs INT32: Illustration

Consider the number 2^{-33}. This number can be represented in **FP32** but not in **INT32**. This is because the range of **FP32** includes numbers as small as 10^{-38}. Even if the decimal point is set at the first bit, the number is still not representable in **INT32**.

Now consider the number 10^{38}. This number can be represented in **FP32** but not in **INT32**. This is because the range of **FP32** includes numbers as large as 3.4028235×10^{38}. However, **INT32** can only represent numbers up to $2^{31} - 1$, which is much smaller than 10^{38}.

Consider the number $2^{31} - 1$. This number can be represented in **INT32** only, and not in **FP32**, even though it is within the range of **FP32**. This is because the mantissa of **FP32** can only represent 23 bits. At $2^{31}, where the exponent is 31, the mantissa can only represent 23 bits, and hence the precision is lost, that is, between each consecutive number, the difference is more than 1 in **FP32**. However, **INT32** can represent this number exactly.