Defining Floating Point Precision

- What is FP64, FP32, FP16?

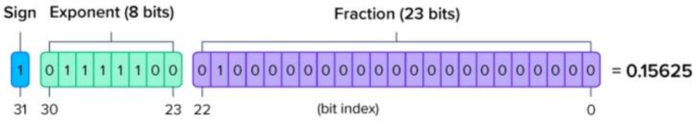
From： https://www.exxactcorp.com/blog/hpc/what-is-fp64-fp32-fp16

Table of Contents

1. What is FP or Floating Point Precision?

Floating Point Precision is a representation of a number through binary. To communicate numbers to our computers, values must be translated to binary 1s and 0s, in which the order and sequence of digits matter. The most common Floating Point Precision formats are Half Precision (FP16), Single Precision (FP32) and Double Precision (FP64), each with their own advantages, disadvantages, and usefulness in specific applications.

Floating Point Precision are structured in a way such that it can define a wide range of values. The first binary digit represents the sign for the number to be positive or negative. The next set of binary digits is the exponent with a base of 2 representing the whole number value. The final set of binary digits represent the significand or mantissa which represents the value after the decimal point. Some even call this the fraction.



Representing number with a long fraction string can be difficult using math operations like exponents and binary. Therefore, there will be some inaccuracies due to rounding errors. for example, in FP32, representing a number with an accuracy within +/- 0.5, the maximum value represented by FP32 must be less than 2^23. Any value larger and distance between values represented by FP32 will be greater than 0.5. On the contrary, representing a number with an accuracy within +/-0.5, the maximum value represented by FP64 can be as large as 2^52. Any value larger and distance between values represented by FP64 will be greater than 0.5. The more accurate, the smaller the range. These inaccuracies can cause errors in applications that need to have fine grain accuracy.

Difference between FP64, FP32, and FP16

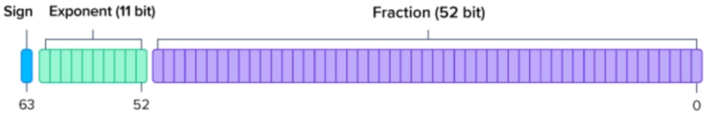
FP64, FP32, and FP16 are the more prevalent floating point precision types. FP32 is the most widely used for its good precision, and reduced size. The larger the floating-point number, the longer it takes to run those highly specific values through calculations. A number of workloads don't benefit from the higher precision, while some workloads are only feasible using lower precision formats. Let's go over the architecture of each floating-point precision format, describe their pros and cons, as well as list some applications these formats best suit.

2. What is FP64?

FP64 or double precision 64-bit floating point precision utilizes 64 bits of binary to represent numbers in calculations performed on your system. This format offers the highest precision among mainstream options making it ideal for application that require precision and accuracy to the tee.

FP64 Double precision uses:

- 1 bit for the positive/negative sign.
- 11 bits for representing the exponent with base 2.
- 52 bits for the fraction/significand/mantissa, i.e. value after the decimal point.



FP64 precision is rarely ever used in Deep Learning and AI workloads as the constant matrix multiplication for updating weights can dramatically increase the training time. FP64 is primarily used for scientific computing with strong precision requirements such as manufacturing product design, mechanical simulation, and fluid dynamics found in Ansys applications.

Only a handful of GPUs can sufficiently compute using FP64 natively such as the NVIDIA GV100 (discontinued), NVIDIA A100, NVIDIA H100, and NVIDIA A800, with other NVIDIA GPUs having a handicapped FP64 performance. Other GPU accelerators that natively support FP64 include the AMD Instinct line such as MI300A and MI300X, all of which we at Exxact stock!

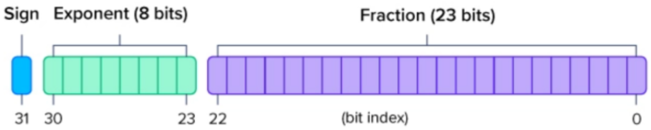
3. What is FP32?

FP32 or single precision 32-bit floating point precision utilizes 32 bits of binary to represent numbers. This format is the most widely used floating point precision format that adequately trades some precision for a lighter weight value represented with less digits. Less digits take up less memory which in turn increases speed.

FP32 Single Precision uses:

- 1 bit for the positive/negative sign
- 8 bits for representing the exponent with base 2.

- 23 bits for the fraction/significand/mantissa, i.e. value after the decimal point.



FP32 is the default for all applications that don't need extreme accuracy, nor does it need to be extremely fast. FP32 is as accurate you can get without going completely overkill. It has been the standard for neural network computations for quite some time and most frameworks and code utilize and run on FP32 by default. However, for those running scientific calculations and simulations, that small inaccuracy in FP32 can throw incorrect calculations due to the accumulation of many small rounding errors.

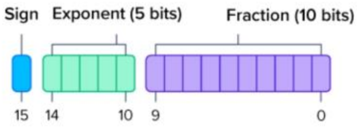
FP32 is used in all workloads including machine learning, rendering, molecular simulations, and more. It is the most balanced floating-point format where speed and precision are necessary.

4. What is FP16?

FP16 or half precision 16-bit floating point precision utilizing only 16-bits of binary. This format is on the upward trend in Deep Learning workloads instead of the traditional FP32. Because of lower precision weights in neural networks don't seem critical nor throw off the performance of a model, the additional precision of FP32 can be traded for speed.

FP16 Half Precision uses:

- 1 bit for the positive/negative sign
- 5 bits for representing the exponent with base 2.
- 10 bits for the fraction/significand/mantissa, i.e. value after the decimal point.



FP16 is used predominantly for deep learning model training and inferencing for fast computations, performing more calculations per unit time. Common AI models that can use less precision is AI trained 3D models where movements don't have to be pin-perfect, and some deviation is allowed. Training these models can help animators ask the model to perform an action to potentially animate into their game or help robotics learn how to walk up a flight of stairs or jump off platforms. However, precise robotics for assembling machinery should benefit from the use of training using higher precision floating point formats.

With the adoption of artificial intelligence in every industry, Exxact is eager to deliver customizable Deep Learning and AI workstations, servers and clusters built ready for any dense workloads. Our close partnerships with hardware companies like NVIDIA, AMD, and Intel enable our systems to be the perfect platform for you to develop revolutionary AI.

Other Floating-Point Precision You May have Encountered

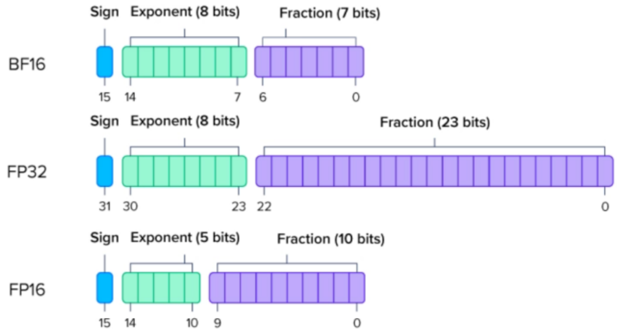
FP64, FP32, and FP16 are the common ones, but there are other floating point precision formats.

5. What is BF16?

BF16 or BFloat16 is a format developed by Google called “Brain Floating Point Format” originating from Google Brain, an AI research group at Google. At Google they saw that FP16 did not have deep learning applications in mind since its range was too limited. While still using 16 bits of binary, they readjusted the allocated bits to have a range that more closely resembles FP32.

BF16 or BFLOAT16 uses:

- 1 bit for positive/negative sign.
- 8 bits for representing exponent with a base of 2.
- 7 bits for the fraction/significand/mantissa, i.e. value after the decimal point.



By using the same number of bits for the exponent as FP32, converting FP32 to BF16 is faster and less complex by ignoring the fraction portion by a number of digits. And while the BF16 format was designed by Google, it is becoming the standard for replacing FP16 when running and training deep neural networks that would’ve used FP32.

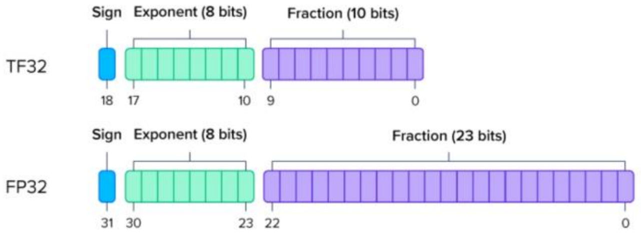
BF16 is also the perfect candidate for mixed precision between FP32 and BF16. Since BF16 is exponentially similar to FP32, it acts as a drop-in replacement for with less decimal digits accuracy while being using half the bit-length. Memory required to store BF16 is lower coupled with the ease of translation, using BF16 can increase speedups for workloads with native FP32 calculations.

6. What is TF32?

TF32 or Tensor-Float32 is an NVIDIA made math mode that represents values from FP32’s 32-bits to 19-bits. Similar to the function of BF16, TF32 uses the same 8-bits for defining exponents to maintain the same range and ease of translation when working with FP32 calculations while using a 10-bit mantissa/fraction from FP16. Because it uses the same 8-bits as FP32, TF32 is another way NVIDIA GPUs can execute FP32 calculations with ease with a shorter fraction value that is rounded from FP32’s 23-bits to TF32’s 10-bits.

TF32 or Tensor-Float32 uses:

- 1 bit for positive/negative sign.
- 8 bits for representing exponent with a base of 2.
- 10 bits for the fraction/significand/mantissa, i.e. value after the decimal point.



Technically speaking, TF32 is 19-bit binary and is a less precise FP32 in the fraction portion of the value represented. NVIDIA Ampere and Later Tensor Cores operate matrix multiplication using translated FP32 inputs to TF32, and outputs matrix multiplied results back in FP32 for further calculations and it works as such:

When calculating matrix operations, FP32 is converted to TF32 by rounding the 23-bit mantissa/fraction to the 10-bits or the 10th decimal place. Next, the FP32 matrix calculations are executed, and then the output TF32 value is defined as an FP32 value. Subsequent non-matrix multiplication operations use FP32.

The reduced precision and less bits to process in matrix operations reduces memory bandwidth required and delivers an increased speedup for these matrix operations found in Deep Learning and HPC.

7. Conclusion

Wrapping things up, floating point precision is a tricky topic that only few will fully grasp. However for those architecting a complex AI model, understanding the right precision to train your model can either increase speedups and reduce total time to completion, or require the utmost accuracy even at the expense of time.

FP64, or dual precision, is the most precise but the most computationally intensive. FP32, or single precision, is the standard with a balance of speed and accuracy. FP16, or half precision, is a reduced precision used for training neural networks. Other formats include BF16 and TF32 which supplement the use of FP32 for increased speedups in select calculations.

If you have any questions as to what kind of hardware suits best for what your workload requires, talk to our engineering team! Whether its deep learning and AI, engineering simulation, or life science/molecular dynamics, Exxact can deliver a system to your needs. Or explore our extensive platforms and build your own custom workstation or server, get a quote.