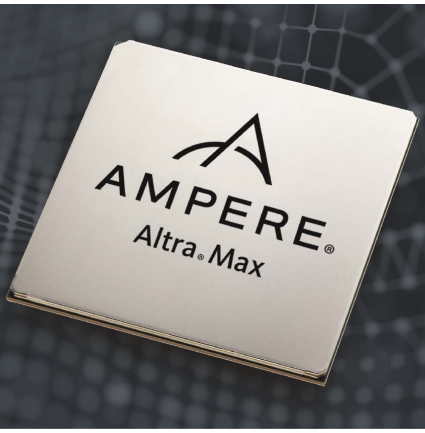
fp16 Data Format Boosts AI Inference Performance in the Cloud



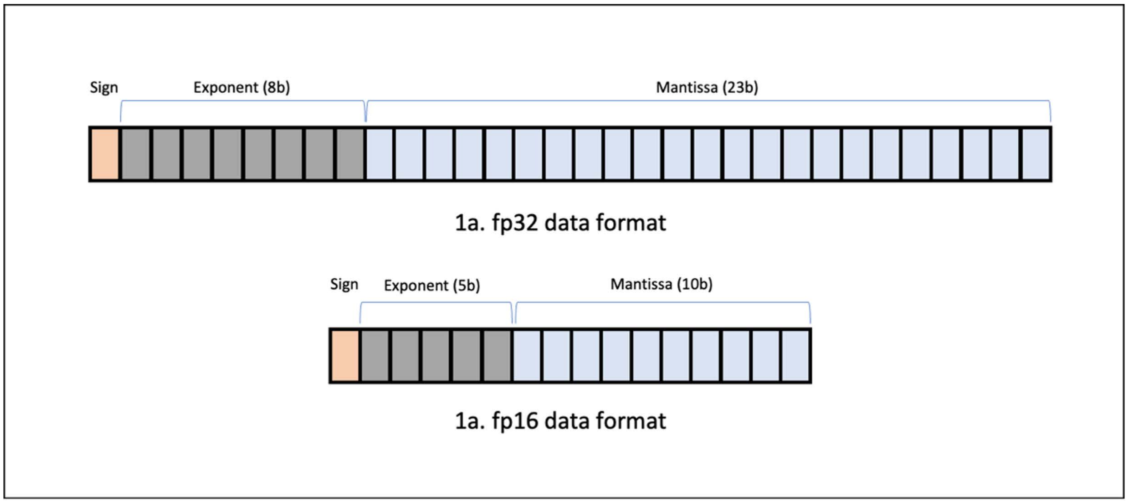
Introduction

Ampere Computing® is currently the only Cloud Native CPU supplier that supports the fp16 data format both in hardware and software with good performance. In running AI inference workloads, the adoption of fp16 instead of the mainstream fp32 offers tremendous advantages in terms of speed-up while reducing power consumption and memory footprint. This advantage comes with virtually no accuracy loss. The switch to fp16 is completely seamless and does not require any major code changes or fine-tuning. Users using Ampere® Altra® and Ampere Altra® Max CPUs will improve their AI inference workload performance instantly.

Overview of Data Formats used in AI

fp32 is the default data format used for training, along with mixed-precision training that uses both fp32 and fp16. fp32 has more than adequate scale and definition to effectively train the most complex neural networks. It also results in large models both in terms of parameter size and complexity of operators as 32 x 32 multiplier accumulators (MACs) are used.

Figure 1: fp32 and fp16 Data Formats



1a. fp32 data format

1a. fp16 data format

fp32 can represent numbers between 10^{-45} and 10^{38}. In most cases, such a wide range is wasteful and does not bring additional precision. The use of fp16 reduces this range to 10^{-8} and 65,504 and cuts in half the memory requirements while also accelerating the training and inference speeds. Make sure to avoid under and overflow situations.

Once the training is completed, one of the most popular ways to improve performance is to quantize the network. A popular data format used in this process, mainly in edge applications is int8 and results in at most a 4x reduction in size with a notable performance improvement. However, quantization into int8 frequently leads to some accuracy loss. Sometimes, the loss is limited to a fraction of a percent but often results in a few percent of degradation, and in many applications, this degradation becomes unacceptable. There are ways to limit accuracy loss by doing quantization-aware training. This consists of introducing the int8 data format selectively and/or progressively during training. It is also possible to apply quantization to the weights while keeping activation functions at fp32 resolution. Though these methods will help limit the accuracy loss, they will not eliminate it altogether.

fp16 is a data format that can be the right solution for preventing accuracy loss while requiring minimal or no conversion effort. Indeed, it has been observed in many benchmarks that the transition from fp32 to fp16 results in no noticeable accuracy without any re-training.

Examples of fp16 Use in Convolutional Neural Networks

In what follows, we provide a few examples that highlight the advantages of using the fp16 data format. Ampere provides many more models on its website (https://github.com/AmpereComputingAI/ampere_model_library) that are already optimized in fp16 format.

SSD ResNet-50 v1.5 (Trained on COCO Dataset for Object Detection)

In this use case, we will look at a ResNet-50 v1.5 SSD object detection using TensorFlow 2.7. Download the model from https://solutions.amperecomputing.com/solutions/ampere-ai.

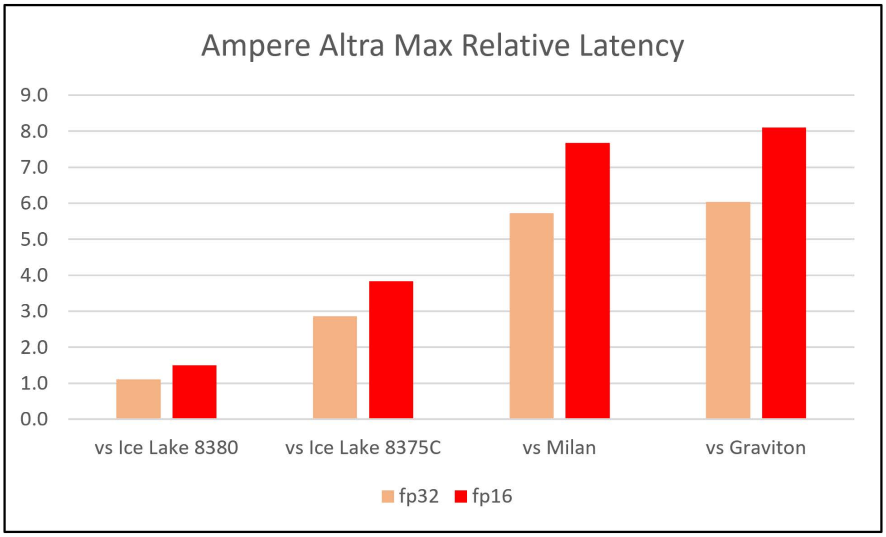
The network is benchmarked with the devices listed in **Table 1**.

Table 1: CPU Devices used in the Benchmark

CPU	Cores	Threads
Ampere Altra Max	128	128
Intel Ice Lake 8375C	32	64
Intel Ice Lake 8380	40	80
AMD Milan 7763	64	128
AWS Graviton 2	64	64

Figure 2 shows the benchmark results relative to fp32 performance on competitive CPUs for single-stream latency.

Figure 2: Object Detection Single-Stream Latency Advantage



Ampere Altra Max in fp32 is up to 6x faster than its competitors except for Ice Lake, which has a 10% advantage. Using fp16 nearly doubles Ampere Altra Max's performance while there is no accuracy loss, as seen in Table 2.

Table 2: Top 1% Accuracy for ResNet-50 v1.5

FP32 Accuracy	FP16 Accuracy
75.1%	75.3%

Support for fp16 confers a unique performance advantage to Ampere Altra Max.

Inception v2 (Trained on ImageNet for Image Classification)

In this example, we will look at Ampere's throughput advantage against the same competing CPU devices under the same benchmarking conditions.

Figure 3 summarizes Ampere's normalized throughput advantage against its competitors. As in the case of the latency evaluation, Ampere Altra Max already has a significant performance advantage in fp32. This gap is almost doubled by moving to fp16 again with no accuracy loss, as seen in Table 3.

Figure 3: Inception v2 Throughput Benchmark (1.0 Indicates Equal Performance)

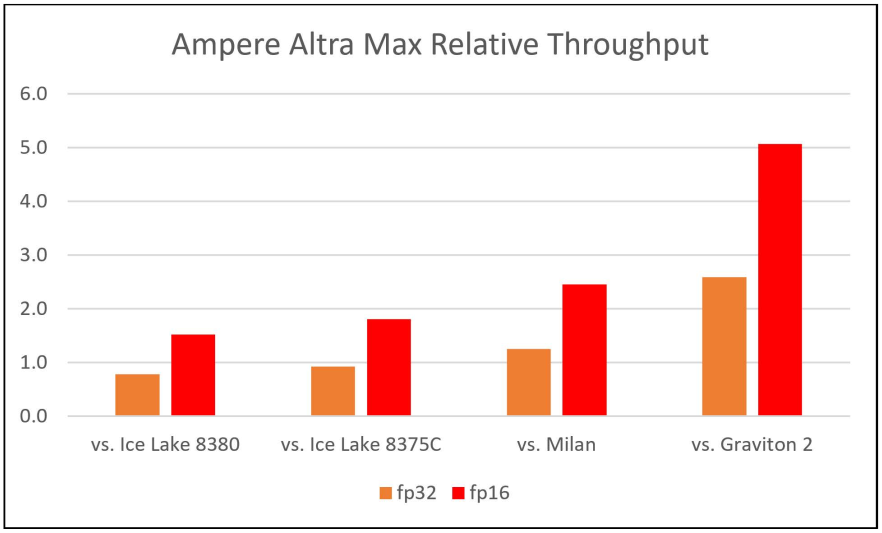


Table 3: Top 1% Accuracy for Inception v2

FP32 Accuracy	FP16 Accuracy
73.0%	72.8%

Conclusions

Ampere Altra and Ampere Altra Max are the only broadly available cloud CPUs that natively support the fp16 data format. They can be used in training with NVIDIA GPUs. More importantly, deployed in inference to double inference speeds while reducing the memory footprint and power consumption. If the original model was not trained using fp16, its conversion to fp16 is extremely easy and does not require re-training or code changes. It is also shown that the switch to fp16 led to no visible accuracy loss in most cases.

Users can download models of interest from Ampere's website and benefit from an instant performance boost.