**Model optimization**

Inference efficiency is particularly important for edge devices, such as mobile and Internet of Things (IoT). Such devices have many restrictions on processing, memory, power-consumption, and storage for models. Furthermore, model optimization unlocks the processing power of fixed-point hardware and next generation hardware accelerators.

Model quantization

Quantizing deep neural networks uses techniques that allow for reduced precision representations of weights and, optionally, activations for both storage and computation. Quantization provides several benefits:

1.Support on existing CPU platforms.

2.Quantization of activations reduces memory access costs for reading and storing intermediate activations.

3.Many CPU and hardware accelerator implementations provide SIMD instruction capabilities, which are especially beneficial for quantization.

There are two kinds of Quantizations:

1.Post-training quantization quantizes weights and activations post training easily.

2.Quantization-aware training allows for training of networks that can be quantized with minimal accuracy drop; this is only available for a subset of convolutional neural network architectures.

Representation for quantized tensors

8-bit quantization approximates floating point values using the following formula. real\_value = (int8\_value - zero\_point) \* scale.

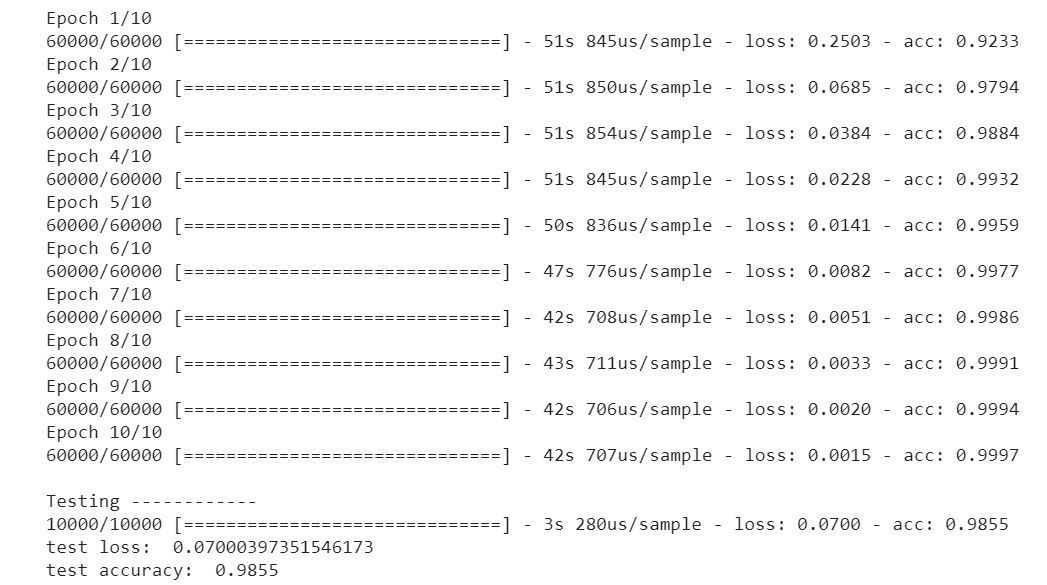
The representation has two main parts:

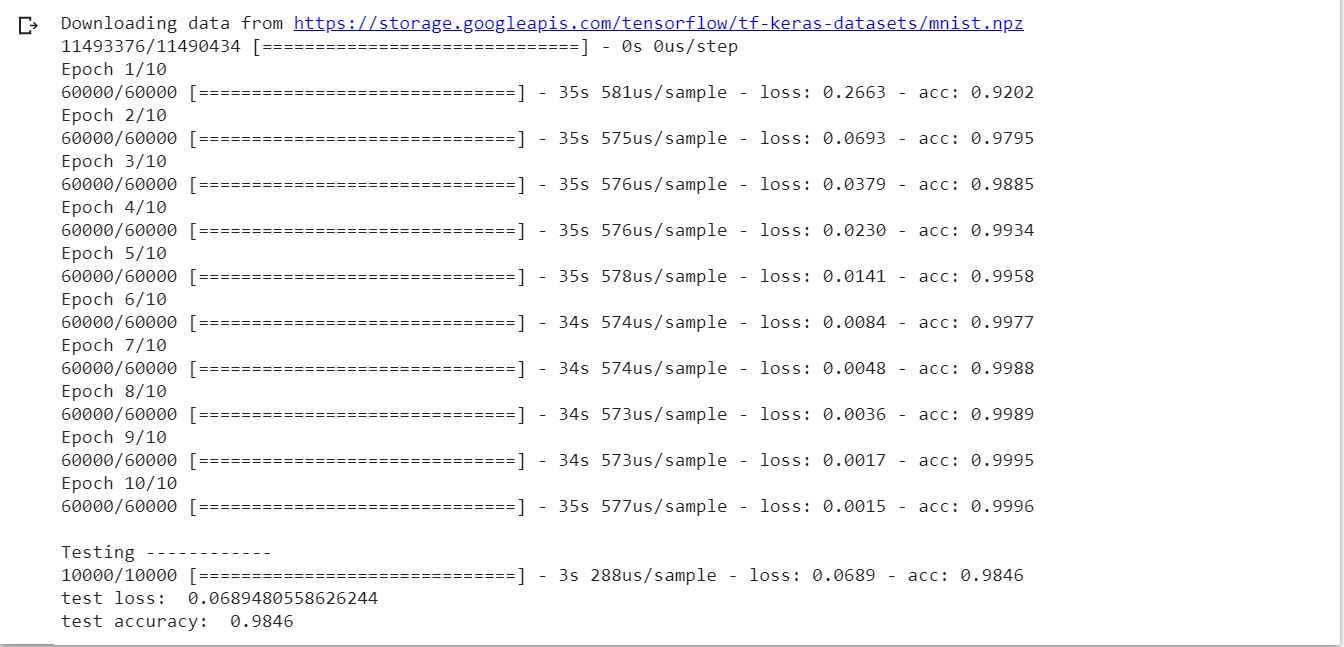
1.Per-axis (aka per-channel) or per-tensor weights represented by int8 two’s complement values in the range [-127, 127] with zero-point equal to 0.

2.Per-tensor activations/inputs represented by int8 two’s complement values in the range [-128, 127], with a zero-point in range [-128, 127].

Below are two training instances where the first one is a vanilla keras model trained on the MNIST dataset with a simple CNN architecture and the second instance is where the same model is trained under the quantization aware training schema

Both instances show little to no deviation in accuracy, although deviation is to be expected when training on more complex datasets. We can clearly see here that the time taken to train reduces significantly when doing quantization aware training.





The second example here is the result of post training quantization predictions

Although the prediction times vary and in most of the cases the tflite model will produce faster results, there is a slight dropoff in accuracy

