Model Quantization

Mì AI

To learn the ternary value (codebook), we introduce two quantization factors W_l^p and W_l^n for positive and negative weights in each layer l. During feed-forward, quantized ternary weights w_l^t are calculated as:

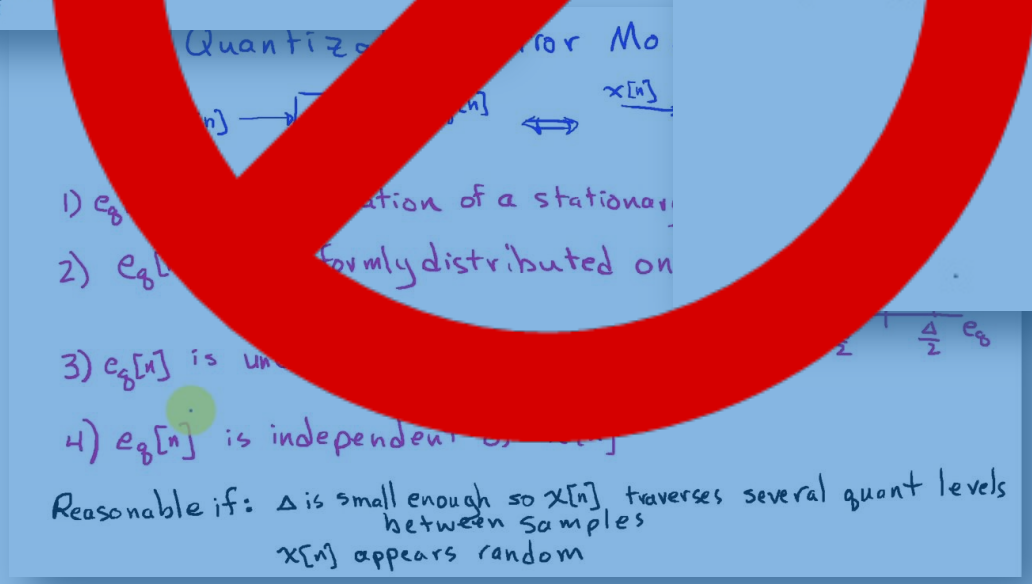
$$w_l^t = \begin{cases} W_l^p : \tilde{w}_l > \Delta_l \\ 0 : |\tilde{w}_l| \leq \Delta_l \\ -W_l^n : \tilde{w}_l < -\Delta_l \end{cases}$$

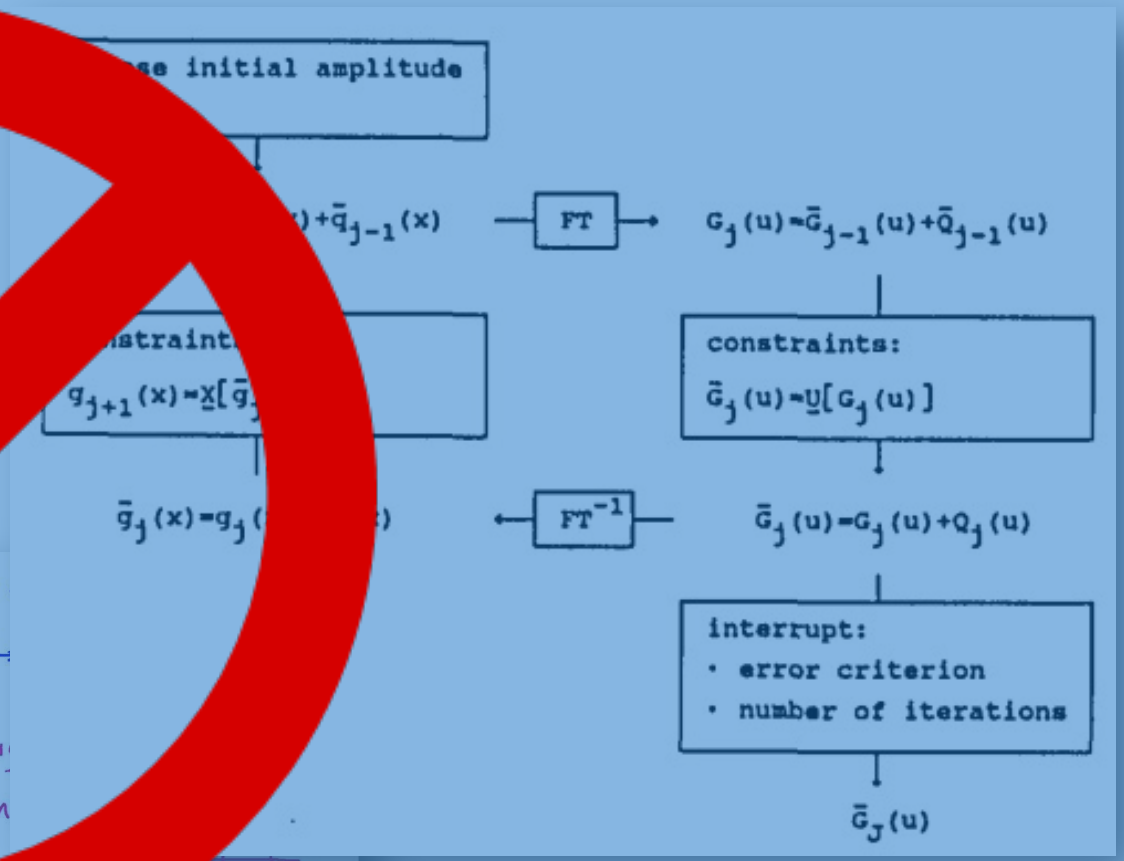
Unlike previous work where quantized weights are calculated from 32-bit weig… cients W_l^p and W_l^n are two independent parameters and are trained togeth… eters. Following the rule of gradient descent, derivatives of W_l^p and W_l^n are ca…

$$\frac{\partial L}{\partial W_l^p} = \sum_{i \in I_l^p} \frac{\partial L}{\partial w_l^t(i)}, \frac{\partial L}{\partial W_l^n} = \sum_{i \in I_l^n} \frac{\partial L}{\partial} \quad (7)$$

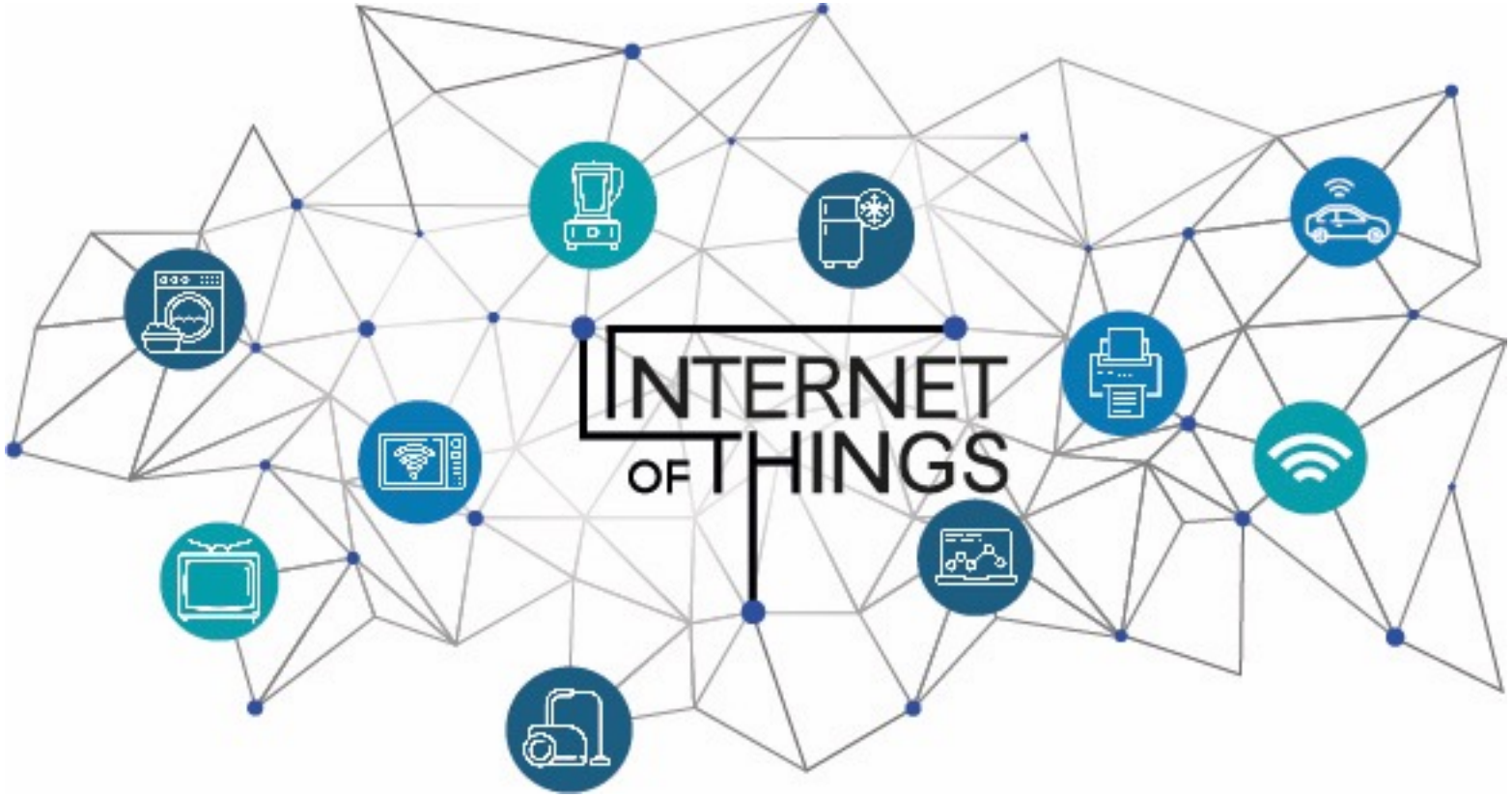
Here $I_l^p = \{i|\tilde{w}_l(i) > \Delta_l\}$ and $I_l^n = \{i|(i)\tilde{w}_l < -\Delta_l\}$. Fur… because of the existence of two scaling factors, gradients of latent full precision wei… longer be calculated by Equation 2. We use scaled gradients for 32-bit weights:

$$\frac{\partial L}{\partial \tilde{w}_l} = \begin{cases} W_l^p \times \frac{\partial L}{\partial w_l^t} : \tilde{w}_l \\ 1 \times \frac{\partial L}{\partial w_l^t} : |\tilde{w} \\ W_l^n \times \frac{\partial L}{\partial w_l^t} : \tilde{w}_l \end{cases} \quad (8)$$





IoT Concept



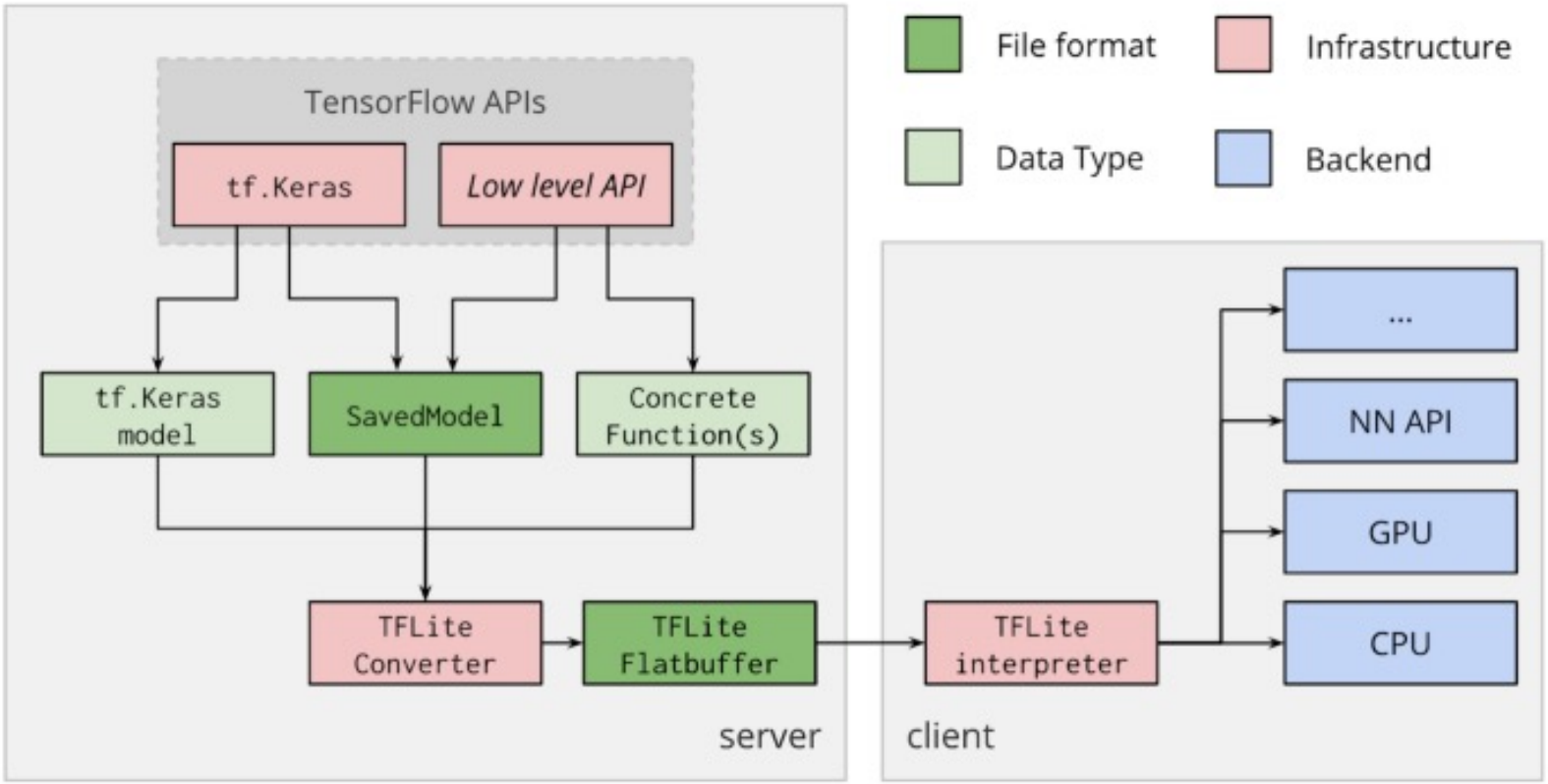
IoT Requirements

- Small size to save memory
- Use little energy to save battery life and
- Low latency or, in other words, high inference time so that a user feels that the model is realtime

TFLite

- From Google to meet all above requirements
- Targeted especially to mobile, edge or IoT devices.
- It also has support for GPU-based model inference via GPU delegates. GPU delegates will work with the native libraries for GPU acceleration via their APIs

TFLite

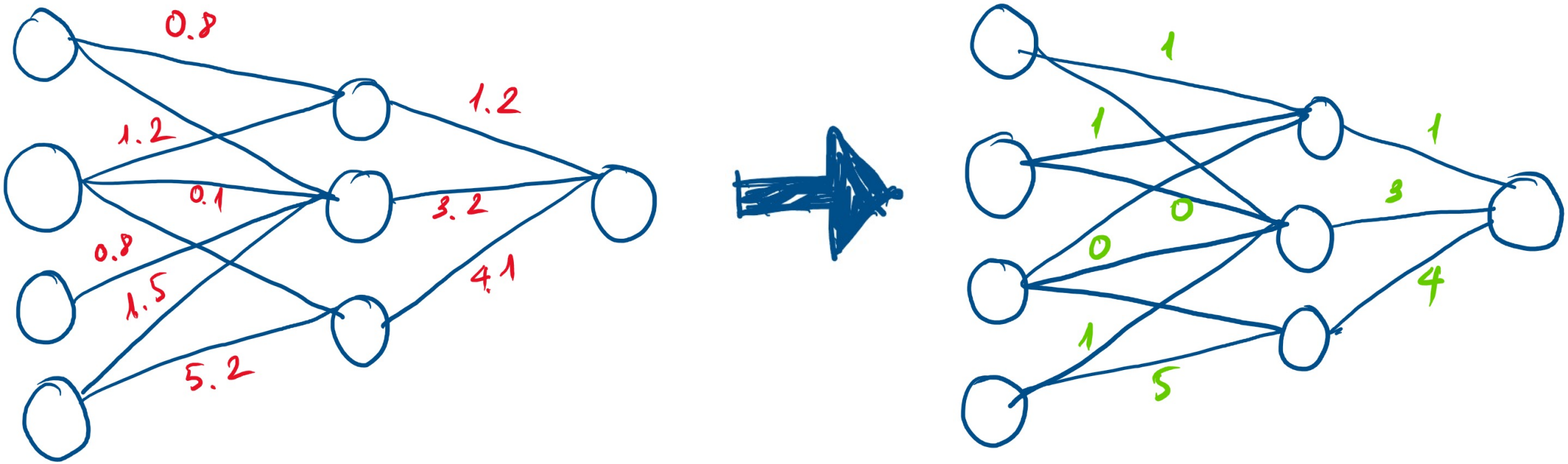


- TFLite interpreter comes with a very light interpreter (with a size of less than 300 KB)
- TFLite converter converts the TensorFlow or Keras model (.pb or .h5) into TFLite model (.tflite)

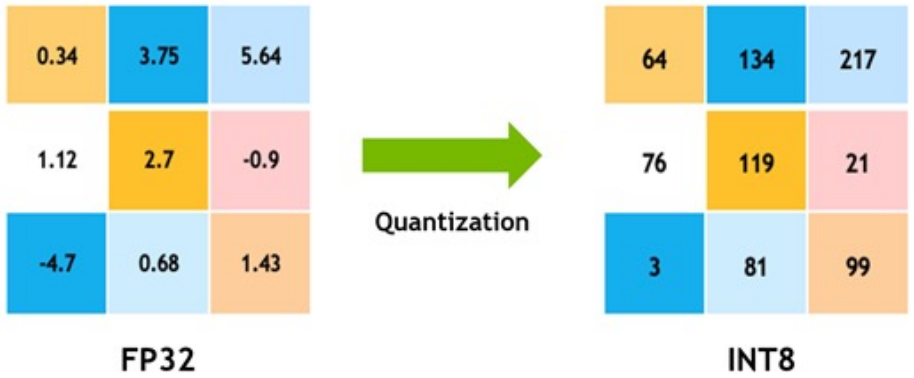
TFLite

https://www.youtube.com/watch?v=Jr7PL-uHUiY

Quantization



Quantization



Quantization

Pros:

Faster, Less Memory

Cons:

Accuracy reduce

Quantization

Cons:
Accuracy
reduce

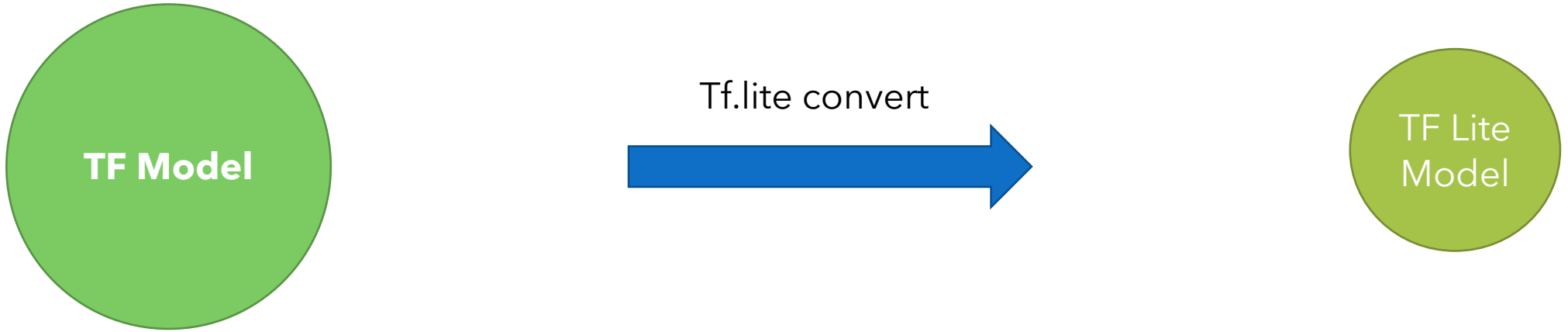
Quantization

Post-training quantization
Entails quantization of the parameters after the model is trained

Quantization-aware training
Entails quantizing the model during the training time

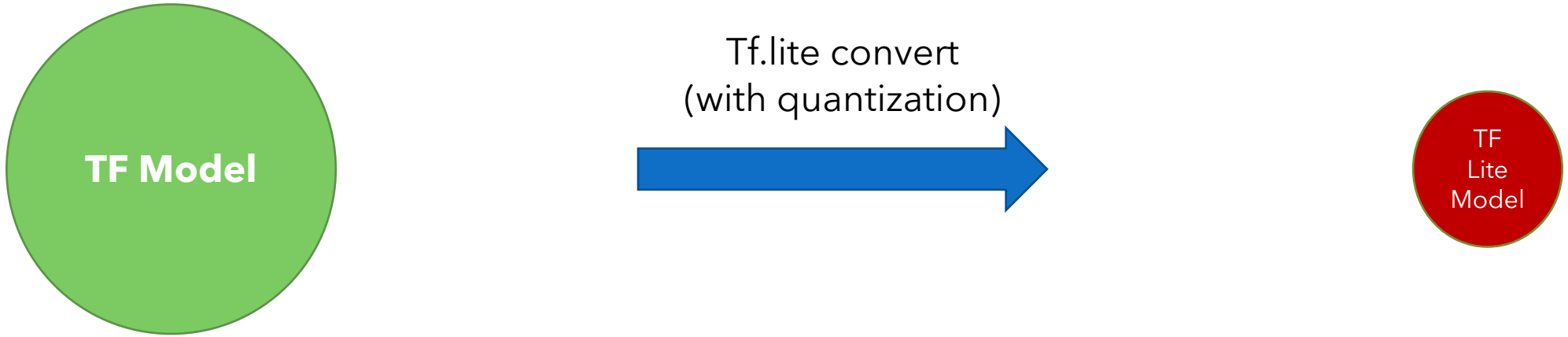
Quantization

Post-training quantization
Entails quantization of the parameters after the model is trained



Quantization

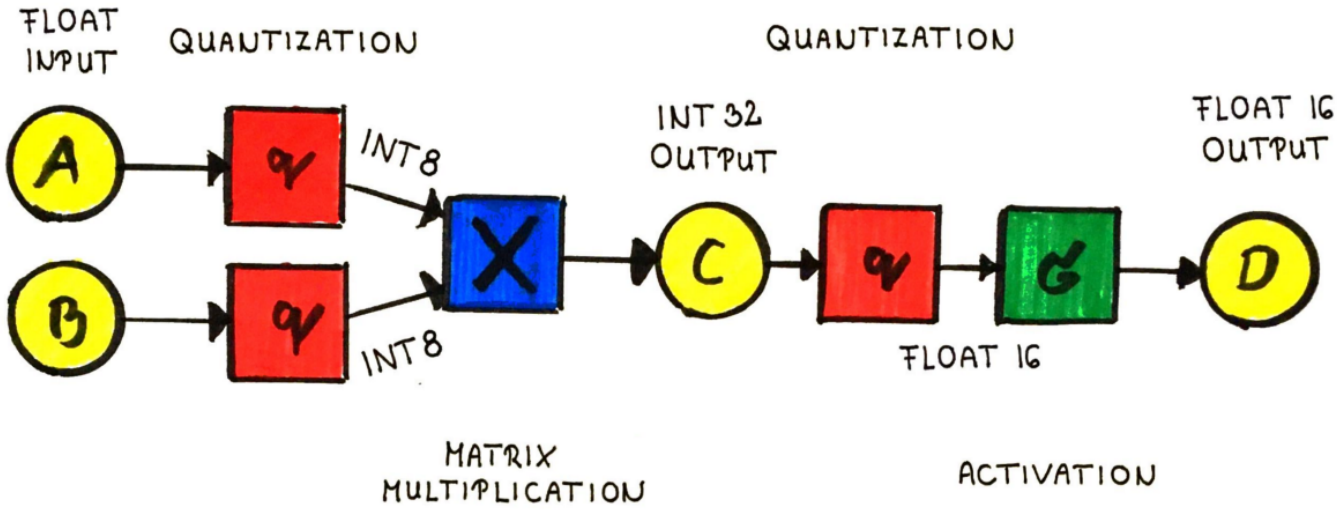
Post-training quantization
Entails quantization of the parameters after the model is trained



Quantization

Post-training quantization
Entails quantization of the parameters after the model is trained

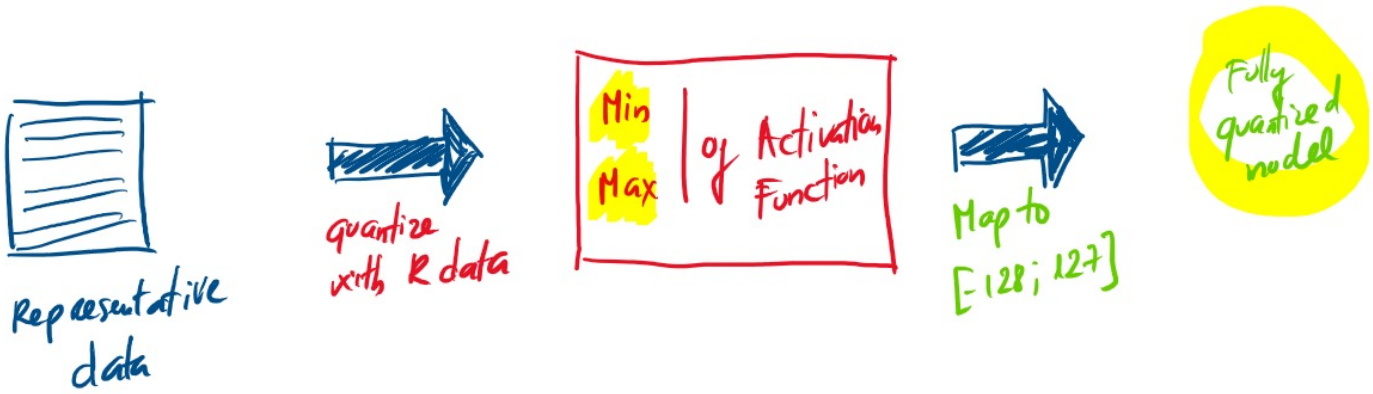
- 2 types:
  - Weight/Hybrid Quantization



Quantization

Post-training quantization
Entails quantization of the parameters after the model is trained

- Full quantization
  - Fully quantize the trained model, i.e., quantization of both weight and activation values is performed
  - The output of an activation function is mapped to, for example [-128, 127] signed INT values
  - Requires a calibration step to determine the scaling parameters. These parameters are computed by running several examples of your representative dataset.



Quantization

Quantization-aware training
Entails quantizing the model during the training time