AOC Paper Reading and Review 2
A White Paper on Neural Network Quantization
N26122246 胡家豪

1. Motivation

深度學習的發展，將模型做的更小更快、更低功耗的需求日益增加。為了這個目的，一個最有效的方式就是量化。將原有 32bits 的權重降為 16bits 甚至是 8bits，即可有效降低模型大小，並且降低功耗。

2. Proposed solution

本篇 paper 是一篇對於量化方法的總述，內容講述了現今的神經網路如何進行量化，其將常見的方法分成兩類：不須額外進行訓練的 Post-training quantization (PTQ)與利用訓練尋求更佳效果的 Qunatiztion-aware training (QAT)。

作者大略的總結了現今常見或效果較好的優化方法，

以及也討論了一次量化一整個 layer 的「per-tensor」與一個個將 layer 內不同的 channel 進行量化的「per-channel」之優劣。

並把這些優化與討論的結果總結成了一個標準的流程

一. Quantization

一個 floating-point 利用以下式子可以將其量化成 fixed-point：

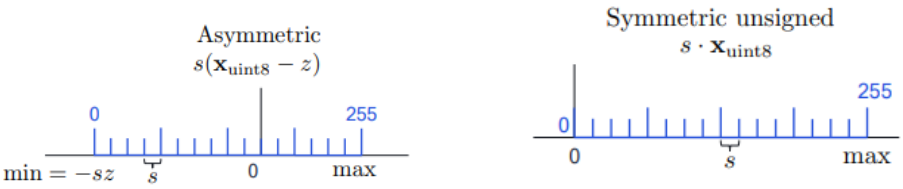
$$X_{int} = clamp\left(\left\lfloor\frac{x}{s}\right\rceil + z\,; 0, 2^b - 1\right)$$

[] ：代表四捨五入

s：計算方式為$\frac{q_{max}-q_{min}}{2^b-1}$ ，代表一個 bit 可以代表多大的間距，q_{max}代表要進行量化的最大值、q_{min}代表要進行量化的最小值

z：代表零點的偏移量

$(0, 2^b - 1)$：代表量化後的範圍

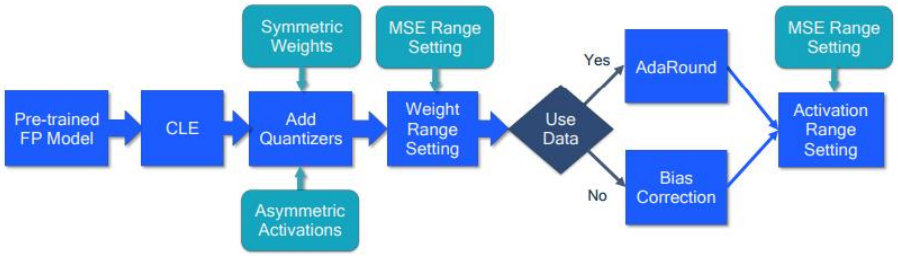


其中上述的也稱為「非對稱量化(Asymmetric)」，當 z 為 0 時稱為「對稱量化(Asymmetric)」，可看成是一種非對稱量化的特別情況。 上左圖是非對稱量化視覺化後的結果，與右側的最大差異就是非對稱量化有 z 的變數會導致量化前後的零點不在相同的位置。

二. PTQ(Post-training quantization)

在已經有訓練好並且精度為 FP32 的網路的情況下，在不使用訓練的前提下將原有的網路轉換成定點數的網路(如 INT8)，使用這種方法有三種需要操作的地方：1.找到一個合適的量化範圍，作者提出了幾種方法，分別是：Min-Max、MSE、Cross entropy、BN based range setting。2.解決原先權重過於離散導致的量化誤差，使用

Cross-Layer Equalization(CLE)。3.解決量化後導致的資料偏移。例如使用 Adaround 等方法。



三. QAT(Qunatiztion-aware training)

雖然 PTQ 在量化上很方便，並不需要重新訓練網路就可以得到不錯的效果，但是在量化成低 bit 數的狀況下，可能因為有較大的誤差而使精確度下降較為劇烈。解決 PTQ 的其中一種方法就是直接利用訓練的方式進行量化。

在這裡最大的挑戰就是需要找出量化函數的梯度，有了這些梯度，就可以將其由 NN 進行訓練。

經過一系列推導，可以找到各個參數的梯度如下：

$$\frac{\partial \hat{x}_i}{\partial \hat{x}_i} = \begin{cases} 1, & if\ q_{min} \le \hat{x}_i \le q_{max} \\ 0, & otherwise \end{cases}, \qquad \frac{\partial \hat{x}_i}{\partial z} = \begin{cases} 0, & q_{min} \le \hat{x}_i \le q_{max} \\ -s, & otherwise \end{cases}$$

$$\frac{\partial \hat{x}_i}{\partial s} = \begin{cases} \frac{-x_i}{s} + \left\lfloor \frac{x_i}{s} \right\rceil, & if\ q_{min} \le \hat{x}_i \le q_{max} \\ n, & if\ x_i < q_{min} \\ p, & if\ x_i > q_{max} \end{cases}$$

，其中 n、p 代表量化範圍 $n = \frac{q_{min}}{s}$ ，$p = \frac{q_{max}}{s}$

3. Evaluation

1. Image classification - PTQ

PASCAL 全稱「Pattern Analysis, Statical Modeling and Computational Learning」，是一個與圖像識

別有關的挑戰賽，其中 VOC(Visual Object Classes)是此挑戰賽的其中一組 dataset

作者利用此組 dataset 對一些比較有名的模型進行量化後，與原本未量化前的版本進行比較。

可以發現，不論是「per-tensor」或「per-channel」的量化方法，都與原有的模型有差不多的精確度。

Models	FP32	Per-tensor		Per-channel	
		W8A8	W4A8	W8A8	W4A8
ResNet18	69.68	69.60	68.62	69.56	68.91
ResNet50	76.07	75.87	75.15	75.88	75.43
MobileNetV2	71.72	70.99	69.21	71.16	69.79

W8A8：weight 與 activation 均量化至 8 bits

W4A8：wegiht 量化至 4bits 而 activation 量化至 8bits

2. Image classification - QAT

同樣使用 PASCAL VOC datset，作者對有名的 model 進行 QAT 量化，多數模型在經過量化後不只模型變小，精確度甚至有所上升

Models	FP32	Per-tensor			Per-channel		
		W8A8	W4A8	W4A4	W8A8	W4A8	W4A4
ResNet18	69.68	70.38	69.76	68.32	70.43	70.01	68.83
ResNet50	76.07	76.21	75.89	75.10	76.58	76.52	75.53
InceptionV3	77.40	78.33	77.84	77.49	78.45	78.12	77.74
MobileNetV2	71.72	71.76	70.17	66.43	71.82	70.48	66.89

W8A8：weight 與 activation 均量化至 8 bits

W4A8：wegiht 量化至 4bits 而 activation 量化至 8bits

W4A4：weight 與 activation 均量化至 4 bits

2. My analysis

我認為量化是一個很神來一筆的概念，通常人們都在盡可能的往更多 bit 數發展(例如 CPU 從 8 bits 到現在 64 bits)，只有量化反其道而行。我覺得可以這樣子做的原因在於人們其實對於神經網路的理解還不夠全面。如果以線性代數而言，或許根本就不需要那麼多參數來表達資訊的組合，而量化恰好將這些多餘的無用資訊剃除；另一方面也加速了硬體執行。

我覺得這篇 paper 未來的研究方向可以朝向硬體邁進，思考如何使用硬體實踐這個方法的加速。或是想辦法找到人類表達訊息所需要的最小單位，如此一來或許可以朝向量化到更小的 bit 但是維持差不多的精度。

最後，這篇 paper 對於如何找到量化的 scale 說明的不太清楚，讓我有些留下疑問。