人工智慧晶片設計與應用

AI-ON-CHIP FOR MACHINE LEARNING AND INFERENCE

Lab 2

E14096724

鄭喆嚴

1. Practice to implement quantization function 50% (每少一張截圖-5%)
   - Question 1. (15%)

```
##########################################################

scale = ( fp_max - fp_min ) / ( q_max - q_min )
zero_point = q_min - fp_min / scale

##########################################################
```

   - Question 2. (15%)

```
##########################################################

q_tensor = torch.round( fp32_tensor / scale ) + zero_point

##########################################################
```

   - Question 3. (20%)

```
################################################

M = input_scale * weight_scale / output_scale
output = torch.nn.functional.linear((input - input_zero_point ), (weights - weight_zero_point ))
output *= M
output += output_zero_point

################################################
```

2. Problem 50%
   - What is the size of the model after int8 quantization if its original size is 50MB? Please write down your calculation process. (Assume the original resolution is 32 bits) (15%)

     進行 int8 量化後精度從 32 位減少到 8 位，可以計算出模型大小下降了 32 / 8 = 4 倍，int8 量化後的模型大小為 50MB / 4 = 12.5MB。

   - If M = 0.2, determine values for M0 and n such that the equation on page 11 is true. (10%)

     根據第 11 頁的公式$M = 2^{-n}M_0$而 M0 = [0.5, 1)，所以當 M = 0.2 時只有一個答案 M0 = 0.8、n = 2。

   - 閱讀 “Quantization and Training of Neural Networks for Efficient Integer Arithmetic-Only Inference”．並根據這篇論文的理論闡述，說明在軟硬體實作上，要怎麼將其理論做實際的應用?(僅說明理論不會有分數，理論說明請用自己的話闡述). 例如: M 在硬體上如何近似處理及如何和其他 post-processing 的步驟搭配，Batch normalization 在軟體上可以怎麼實現 folding，軟體上怎麼實現 fuse layer 等等 .....其他不同的面向 (20%)

在硬體上乘上 M 的作法是如下， M 可以表示為 M=2^(-n)*M0，M0 位於[0.5,1)且 n 是非負整數，所以在乘上 M 時可以先向右位移 n bit，再跟 M0 做 fixed-point multiplication 來達到乘 M 的效果。在 32-bit 中 integer part bit 應該要是 16-bit 以上來確保不會發生 overflow，因為 q1*q2 是 uint8*uint8 還需要進行累加，所以需要 16-bit 以上像是 20-bit 或 24-bit 的 integer part bit 來避免 overflow，如果是使用 32-bit 的累加器 fractional part bit 就是 32-bit 減掉 integer part bit 剩下的 bit，如果需要提升精度的話需要將 fractional part bit 的數量提升，但同時就會提升發生溢出的機率。溢出發生時是採用 down-scaling 的作法，也就是如果 overflow 時使用最大的可用值代替，如果 underflow 時使用最小的可用值代替。

Batch normalization 在軟體上實現 folding 的幫法是將 convolution 和 batch norm 進行合併，在很大的 batch sizes 時大量減少了記憶體的使用。在軟體上實現 fuse layer 的方法是在 forward pass 過程中，執行正常的 convolution 和 batch norm，但只保存 convolution 的輸入；在 backward pass 過程中，則需要重新計算 convolution 的 forward pass 來獲取 batch norm 的輸入，才能進行 backward pass。

- Share your thoughts on this lab, any advice or improvement on codes, tutorials, or other ideas about quantization. (5%) 有認真表達心得一律滿分

這次的 Lab 讓我更加了解了有關 quantization 的做法，像是公式是如何運用在程式中，還有實際計算了 quantization 過後模型會縮小多少，另外還有閱讀了有關 quantization 的論文和 Batch normalization convolution fusion 的相關文章，學習到了有關 quantization 在硬體上面的使用情形和 Batch normalization convolution fusion 在軟體上面是如何執行的，這些地方之前都沒有學過讓我覺得有些困難，但在我了解之後都讓我很有收穫。