POLYTECHNIQUE MONTRÉAL

UNIVERSITÉ D'INGÉNIERIE

U-Net Fixed Point Quantization For Medical Image Segmentation

MohammadHossein AskariHemmat

Sina Honari

Lucas Rouhier

Christian S. Perone

Julien Cohen-Adad

Yvon Savaria

Jean-Pierre David

MICCAI2019, October 17th

Presentation outline:

1. Medical Image Segmentation With Deep Neural Networks (DNNs)
2. Quantization in Deep Neural Networks
3. U-Net Fixed-Point Quantization for Medical Image Segmentation
4. Results
5. Conclusion

Medical Image Segmentation With Deep Neural Networks (DNNs)

- Medical Image Segmentation are performed on MRI images to help doctors diagnose diseases.

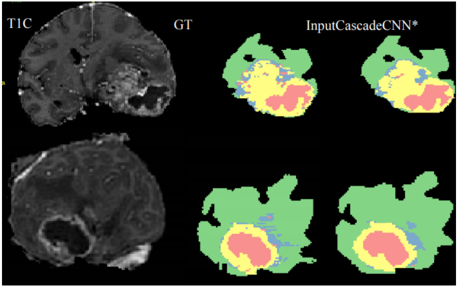


Figure: Segmentation of a brain tumor in an MRI image [M. Havaie et al 2016].

Medical Image Segmentation With Deep Neural Networks (DNNs)

- Medical health centers and hospitals are equipped with pre-trained models used in medical CADs to analyse MRI images.
- However, to perform these tasks with DNNs, a lot of computation needs to be done.
- But how many operations are really needed?

POLYTECHNIQUE MONTRÉAL

Computation Cost in Deep Neural Networks (DNNs)

Training Computation Cost :

Finishing a 90-epoch ImageNet-1k training with ResNet-50 on a NVIDIA M40 GPU takes 14 days. This training requires 10^{18} single precision operations in total [Y. You et al "ImageNet Training in Minutes"].

Inference Computation Cost :

Finishing a full pass of Imagenet with input size of 224x224 with batch size of 128 reuqires 13 GB feature memory and 497 GFLOPs [S. Albanie GitHub:convnet-burden].

Cost and energy consumption source in DNNs:

- Number of parameters in the network (Data Movement).
- Per operation computation cost.

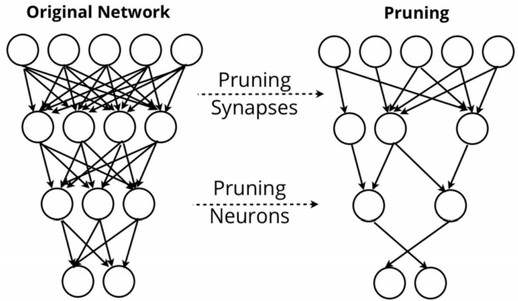


Figure: Deep compression [H. Song, et al., 2015]

Quantization For Accelerating Computation in DNNs :

One way to accelerate computation and save energy in a DNN is to use less precision for computation:

- Less number of bits are required to be stored in memory, which reduces data movement.
- Floating point calculation is costly and slow. Less energy and computation will be spent per operation.

Quantization For Accelerating Computation in DNNs :

What is Quantization in DNN?

Quantization is a technique to reduce memory consumption and the computation time of deep neural networks by lowering the precision of parameters.

Quantization For Accelerating Computation in DNNs :

- Quantization not only uses less memory but it is more energy efficient:

Operation	MUL	ADD
8-bit Integer	0.2pJ	0.03pJ
32-bit Integer	3.1pJ	0.1pJ
16-bit Floating Point	1.1pJ	0.4pJ
32-bit Floating Point	3.7pJ	0.9pJ

Figure: Energy consumption of multiplication and accumulation in a 45nm process (Horowitz, 2014)

- In Intel Core i7 4770 3.40GHz, 32-bit multiplication is more than 3 times faster for fixed point data compared to floating point data. [https://goo.gl/7Y7GWt].

Quantization For Accelerating Computation in DNNs :

Does it Work?

Quantization For Accelerating Computation in DNNs :

- In [M. Courbariaux et al. "BinaryConnect"] it has been shown that for small models (MNIST size) using even 1-bit for weights results in test accuracy drop of only 1%.
- For bigger models (like ImageNet) using 8-bit integers instead of 32-bit floating point shows state-of-the-art performance.
- Pytorch just added quantization option!

POLYTECHNIQUE MONTRÉAL

Existing works for quantization of medical images:

- FCN Quantization [Xu, X., et al. CVPR 2018]: Applied quantization on Fully Convolutional Networks for biomedical application.
- TernaryNet [Heinrich, M.P et al, IJCARS 2017]: First quantization results for U-Net using ternary net.

Our research objective:

Quantization for Medical Image Segmentation:

In this work, we wanted to know does quantization works for life threatening tasks such as medical imaging?

- We wanted to use a well known model that is widely used for medical imaging.
- We wanted to have only fixed point operations so that it can be used with an integer processors.

U-Net Fixed-Point Quantization for Medical Image Segmentation

- In our work, we transformed all floating point computation to fixed-point computation.

$$x_f = abs(x) - floor(abs(x)), x_i = floor(abs(x)) \tag{1}$$

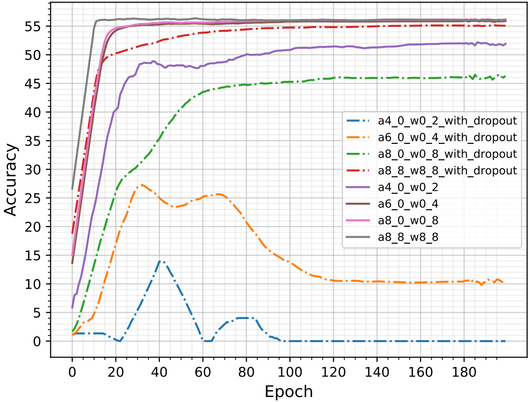
$$quantize(x, n) = (round(clamp(x, n) << n)) >> n \tag{2}$$

$$clamp(x, n) = \begin{cases} 2^n - 1 & \text{when} \quad x \geq 2^n - 1 \\ x & \text{when} \quad 0 < x < 2^n - 1 \\ 0 & \text{when} \quad x \leq 0 \end{cases} \tag{3}$$

- $Q^p i.f$: fixed point quantization of parameter p by using i bits to represent the integer part and f bits to represent the fractional part.

Challenges for Training:

- Dropout and Quantization: we found that when Dropout is applied along with quantization, the accuracy drops a lot:



Challenges for Training:

- Keeping the last layer in full precision has much more impact than keeping the first layer in full precision.
- At inference, we used Pytorch batch-norm folding. Effectively including batch-norm parameters in the quantized model as part of the quantized weights.

Layer (type)	Output Shape	Param #
Conv2d-1	[64 , 200, 200]	640
BatchNorm2d-2	[64 , 200, 200]	128
QuantLayer-3	[64 , 200, 200]	0
Conv2d-4	[64 , 200, 200]	36,928
BatchNorm2d-5	[64 , 200, 200]	128
QuantLayer-6	[64 , 200, 200]	0
DownConv-7	[64 , 200, 200]	0
MaxPool2d-8	[64 , 100, 100]	0

Results For Quantization of U-Net Model for Medical Image:

- We used three different datasets:

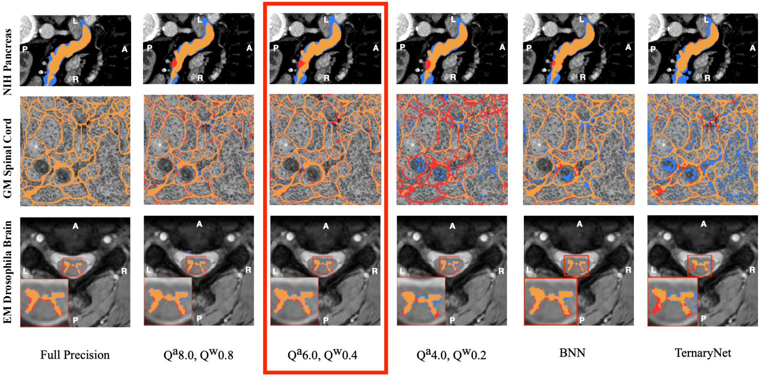


Figure: Segments in blue show false positive, segments in red show false negative and segments in orange show true positive.

Results For Quantization of U-Net Model for Medical Image:

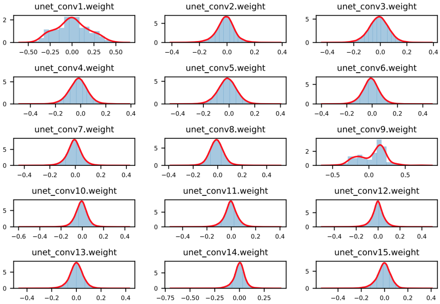
- $Q^a 6.0, Q^w 0.4$ compared to other methods:

Quantization			EM Dataset		GM Dataset		NIH Panceas
Activation	Weight	Parameter Size	Dice Score ReLU	Dice Score Tanh	Dice Score ReLU	Dice Score Tanh	Dice Score
Full Precision		18.48 MBytes	94.05	93.02	56.32	56.26	75.69
Q8.8	Q8.8	9.23 MBytes	92.02	91.08	56.11	56.01	74.61
Q8.0	Q0.8	4.61 MBytes	92.21	88.42	56.10	53.78	73.05
Q6.0	Q0.4	2.31 MBytes	91.03	90.93	55.85	52.34	73.48
Q4.0	Q0.2	1.15 MBytes	79.80	54.23	51.80	48.23	71.77
BNN [18]		0.56 MBytes	78.53	-	31.44	-	72.56
TernaryNet [20]		1.15 MBytes	-	82.66	-	43.02	73.9

Figure: shows best score overall and shows best score between three quantiation methods.

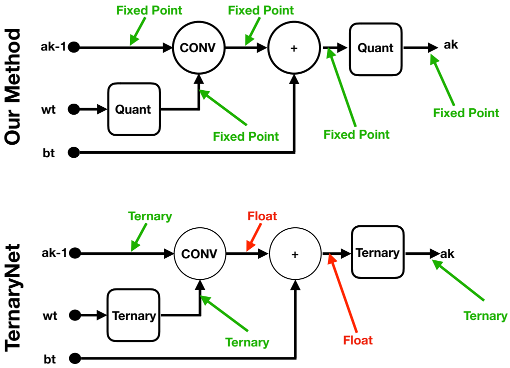
Results and Discussion:

- We found out that instead of using 32-bit floating point values, we can use 4 bits for weights and 6 bits for activations.



Results and Discussion:

- Fully fixed point data path:



Results and Discussion

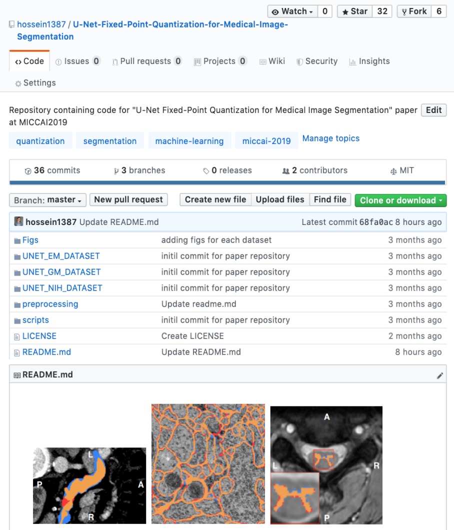
- Relu Vs Tanh performance:

Layer Type	Instruction Type	Execution time in μs **Tanh**	Execution time in μs **ReLU**	Performance Gain of using ReLU over Tanh	Tensor Dimension
Activation	jit_avx2_FP32	30	5	6	[100, 100]
FullyConnected	gemm_blas_FP32	20	19	-	-
FullyConnected	gemm_blas_FP32	860	527	-	-
Activation	jit_avx2_FP32	77	9	8.6	[100, 300]

Figure: Comparing ReLU and Tanh run time using Intel's OpenVino [Deanne. D rt. al Release notes for intel® 2019]

Our Code is on GitHub:

https://github.com/hossein1387/U-Net-Fixed-Point-Quantization-for-Medical-Image-Segmentation.



Our Contribution in this work:

- We report the first **fixed point quantization** results on the U-Net architecture for the medical image segmentation task and show that the current quantization methods available for U-Net are not efficient for the hardware commonly available in the industry.
- We quantify the impact of fixed point quantization on the performance of the U-Net model using three different medical imaging datasets.
- We report results comparable to a full precision segmentation model by using only 6 bits for activation and 4 bits for weights, effectively reducing the weights size by a factor of 8x and the activation size by a factor of 5x.

Thank you for your attention!