Accelerating Convolutional Neural Networks Using Low Precision Arithmetic

Hiroki Naganuma, Rio Yokota
Tokyo Institute of Technology

東京工業大学 Tokyo Institute of Technology

Background of our research

Convolutional Neural Network(CNN)

1. **Performance overwhelming** traditional methods in the field of **image recognition**
2. Tendency to **multilayer** as accuracy improves
3. Increase in **calculation and data volume** related to training (Training) and inference (Inference)
4. Bottleneck is calculated by **convolution**

Reducing calculation / data volume is a issue

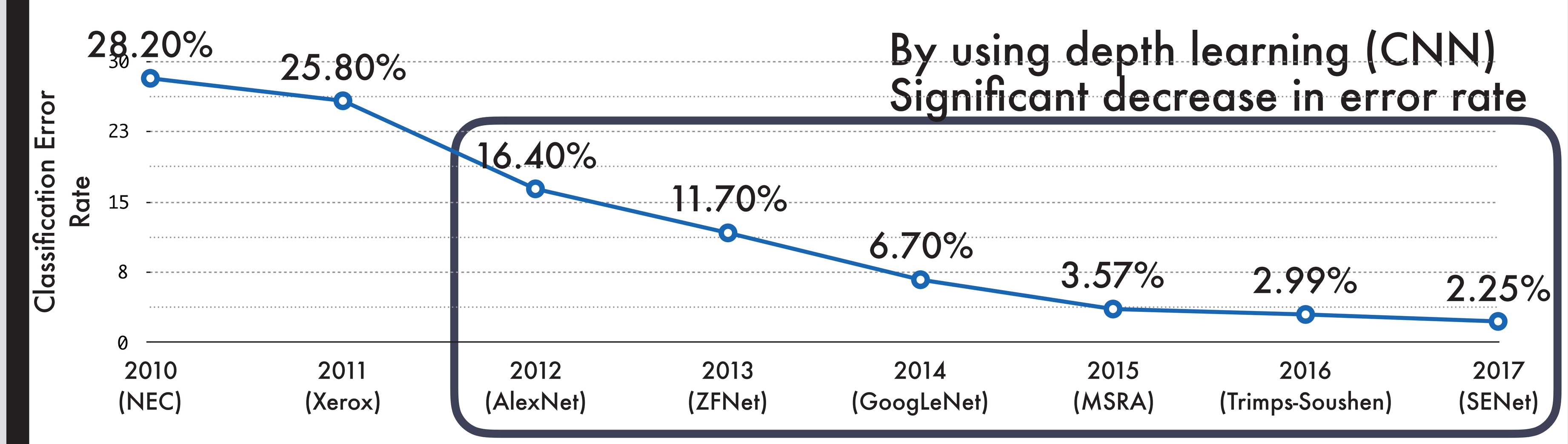


fig1 :1000 object category classification error rate of large scale image recognition contest ILSVRC 2010 - 2015

Research concept

low precision data type and arithmetic

- Verify and evaluate a method to **compress the amount of data** contained in CNN and **speed up**
- Parallelize using SIMD instruction

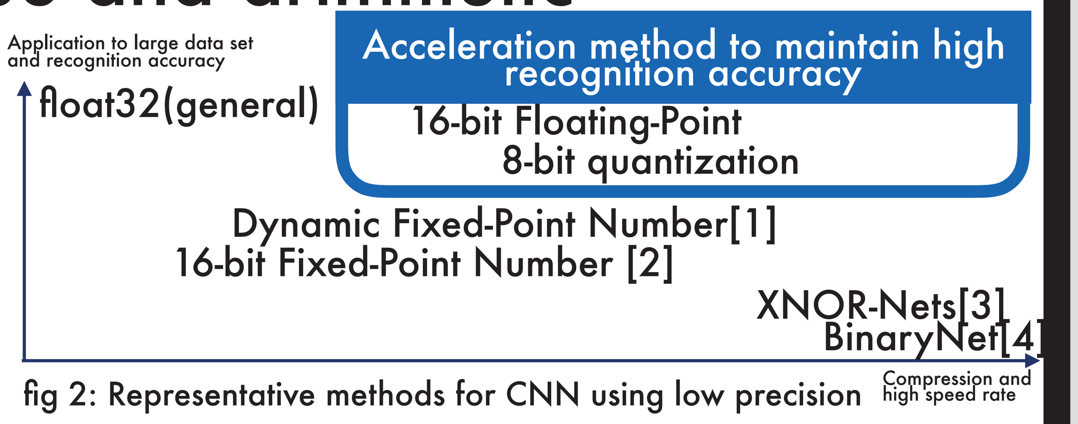


fig 2: Representative methods for CNN using low precision

Compression of model and speedup by SIMD instruction

CNN with low precision arithmetic

float16

- Data type : float16
  -> **Halves memory consumption**
- Computation type : SIMD instruction using float16
  -> **Doubled throughput**

int8 quantization

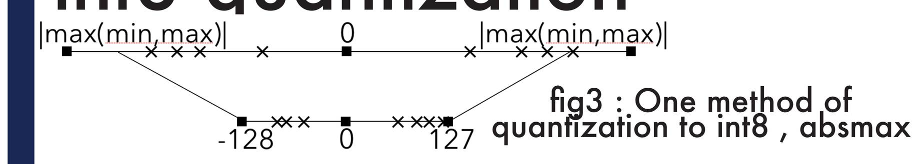


fig3 : One method of quantization to int8 , absmax

- Reduce data volume to 1/4
- Verify the influence of CNN of multiple quantization methods(minmax, absmax)on recognition accuracy respectively

Experiment

Speed comparison of matrix multiply-add

product-sum calculation of the matrix

- Execution time verification of **multiply-accumulate operation** of fp32, fp16, int8
- fp16 : NVIDIA Tesla P100
- fp32, int8 : NVIDIA GTX1080TI

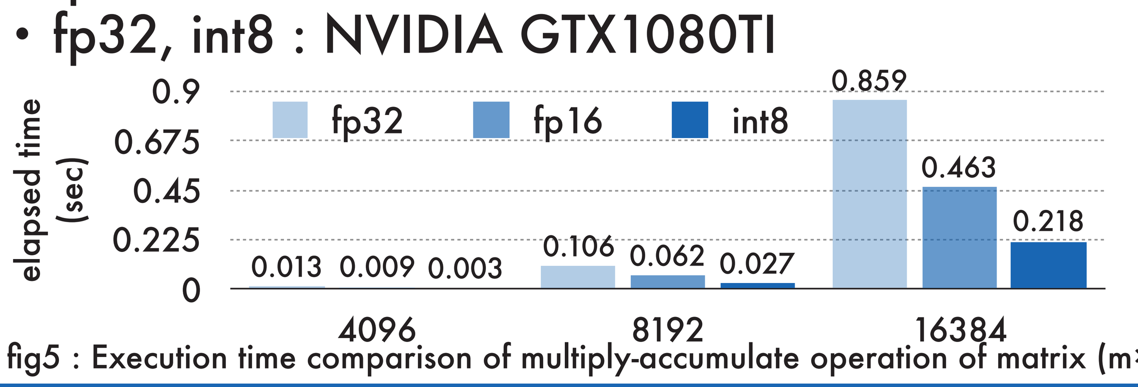


fig5 : Execution time comparison of multiply-accumulate operation of matrix (m×n×k matrixsize=m=n=k)

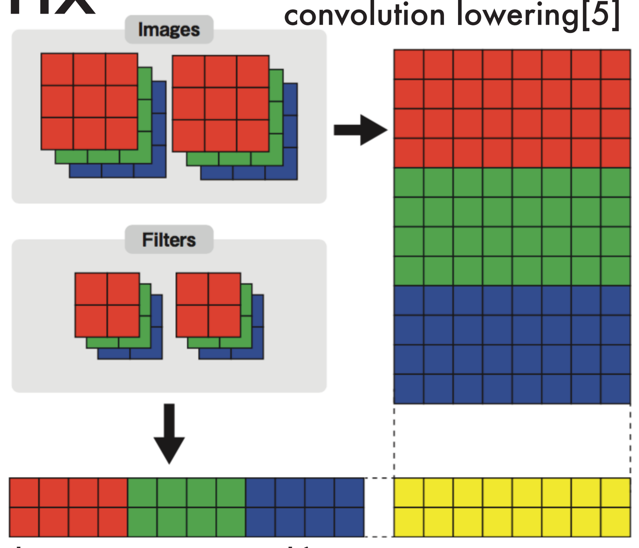
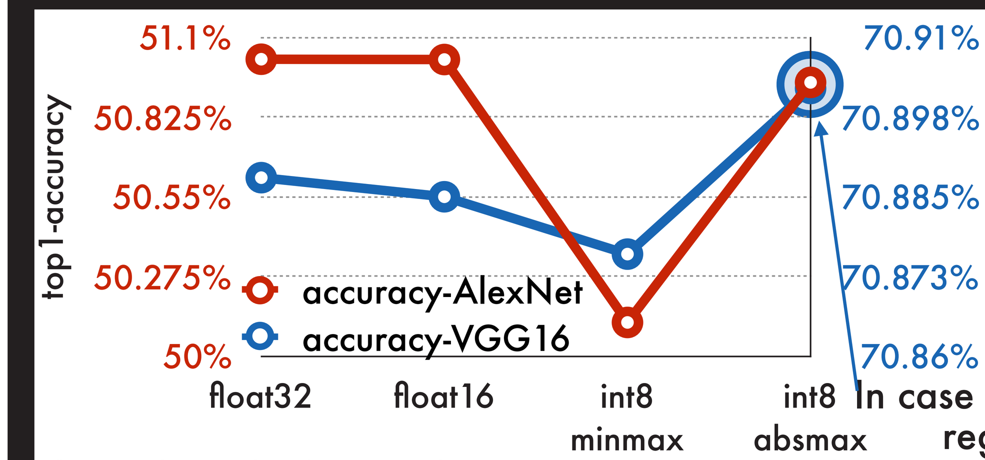


fig6 : im2col / convolution lowering[5]

Accelerator speeds up matrix multiplication by SIMD instruction

Influence of low-precision data-type on CNN recognition performance



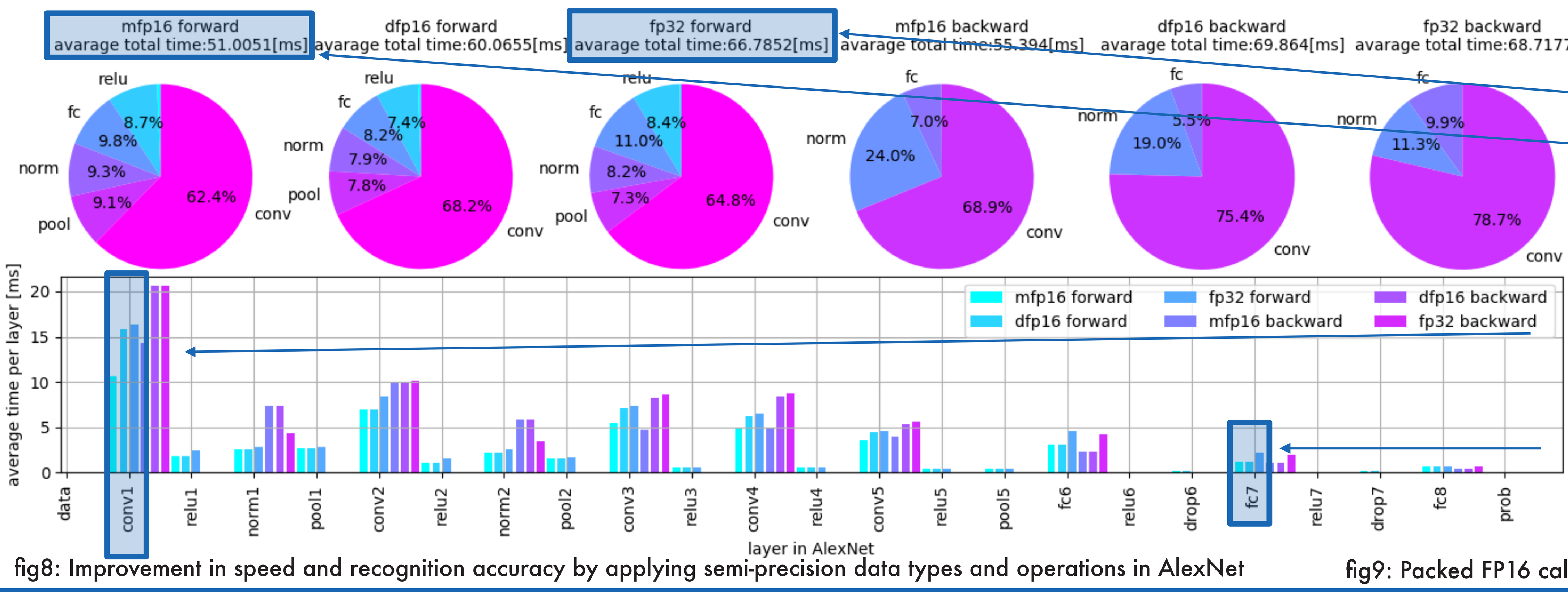
Precision	Description
float32	Data type : fp32(single precision) Computation type : fp32(single precision)
float16	Data type : float16 Computation type : SIMD instruction using float16
int8	Data type : int8 quantization (minmax / absmax) Computation type : float32 (SIMD instruction using float16 int8 / future work)

In case of VGG16, it is considered to be the effect of noise regularization by application of low precision type[6]

fig7: Relationship between low accuracy operation and recognition accuracy in AlexNet, VGG16

Recognition performance of CNN using low precision data type and low precision arithmetic of float16 does not degrade greatly

impact and speedup of half-precision computation / data-type application on CNN



Analysis by layer

baseline (inference)

In case the data type and operation type are set to float16

Inference time of CNN as a whole
1.309x speed up

Conv1 layer (computation bound)
1.548x speed up

Fc7 layer (memory bound)
1.789x speed up

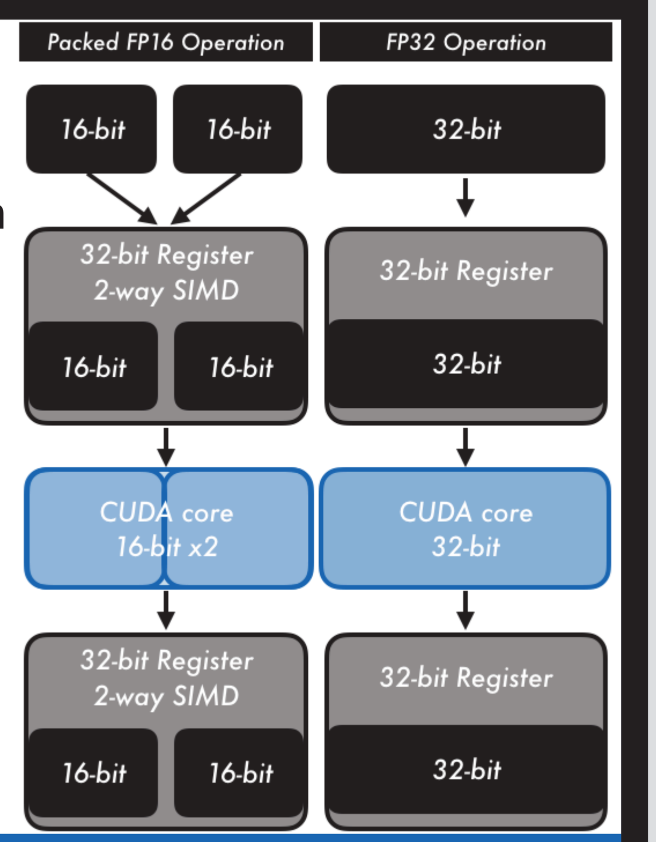


fig8: Improvement in speed and recognition accuracy by applying semi-precision data types and operations in AlexNet

fig9: Packed FP16 calculation in NVIDIA Pascal generation GPU (right figure)

Half-precision arithmetic has different effects depending on each layer (memory bound: memory compression, computerion bound: SIMD instruction)

Conclusion

purpose of this research

Validation and evaluation of a method for **compressing** the amount of **data** contained in convolution neural network using **low precision data type** and **low precision arithmetic** and **speeding up** by **SIMD instruction**

conclusion

- Application of half-precision arithmetic is sufficiently effective even in the data type in the layer which is memory bound (**speed up of about 1.78x, memory compression ~2x**)
- Speeding up in the SIMD instruction of float16 is effective for the computation bound layer, but it is about **1.58x** at maximum
- The **accuracy of recognition does NOT degrade** depending on CNN and the method of inference using int 8 data type

Computation Precision	Inference Time	GPU UseRate	Memory Usage	top-1 acc	top-5 acc
fp32	66.785ms	99%	12845MiB	0.56828	0.79950
Dfp16Mfp32	60.065ms	99%	7475MiB	0.56813	0.79962
Dfp16Mfp16	51.005ms	99%	7475MiB	0.56821	0.79944

fig10: Acceleration of AlexNet by half precision data type and application of half precision arithmetic and its effect on recognition accuracy

Future work

Future work

- Verification of half precision arithmetic using NVIDIA TESLA V100
- GPU implementation of int8 arithmetic type corresponding to 8-bit quantization
- Verification of convergence of learning by 8-bit quantization
- Validation of effective use of weights of compressed models such as Deep Compression[7]

Reference

[1] S. Gupta, A. Agrawal, K. Gopalakrishnan, P. Narayanan, Deep Learning with Limited Numerical Precision. , International Conference on Machine Learning, 2015
[2] M. Courbariaux, Y. Bengio, J. David, Training deep neural networks with low precision multiplications., Advances in Neural Information Processing Systems 28, 2015
[3] M. Rastegari, V. Ordonez, J. Redmon, A. Farhadi, XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks., European Conference on Computer Vision, 2016
[4] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, Y. Bengio, BinaryNet: Training Deep Neural Networks with Weights and Activations Constrained to +1 or −1., Computer Vision and Pattern Recognition, 2016
[5] K. Chellapilla,S. Puri, P. Simard, High Performance Convolutional Neural Networks for Document Processing, Tenth International Workshop on Frontiers in Handwriting Recognition, pp. ~386-408, 2006
[6] Y Luo, F Yang, Deep Learning With Noise, http://www.andrew.cmu.edu/user/fanyang1/deep-learning-with-noise.pdf, 2014
[7] S. Han, H. Mao, W. Dally, Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding, International Conference on Learning Representation, pp. ~74-76, 2016