PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

Reducing weight precision of convolutional neural networks towards large-scale on-chip image recognition

Ji, Zhengping, Ovsiannikov, Ilia, Wang, Yibing, Shi, Lilong, Zhang, Qiang

Zhengping Ji, Ilia Ovsiannikov, Yibing Wang, Lilong Shi, Qiang Zhang, "Reducing weight precision of convolutional neural networks towards large-scale on-chip image recognition," Proc. SPIE 9496, Independent Component Analyses, Compressive Sampling, Large Data Analyses (LDA), Neural Networks, Biosystems, and Nanoengineering XIII, 94960A (20 May 2015); doi: 10.1117/12.2176598

SPIE.

Event: SPIE Sensing Technology + Applications, 2015, Baltimore, Maryland, United States

Reducing Weight Precision of Convolutional Neural Networks towards Large-scale On-chip Image Recognition

Zhengping Ji*, Ilia Ovsiannikov, Yibing Wang, Lilong Shi and Qiang Zhang
Advanced Image Research Laboratory, Samsung Semiconductor Inc.

ABSTRACT

In this paper, we develop a server-client quantization scheme to reduce bit resolution of deep learning architecture, i.e., Convolutional Neural Networks, for image recognition tasks. Low bit resolution is an important factor in bringing the deep learning neural network into hardware implementation, which directly determines the cost and power consumption. We aim to reduce the bit resolution of the network without sacrificing its performance. To this end, we design a new quantization algorithm called supervised iterative quantization to reduce the bit resolution of learned network weights. In the training stage, the supervised iterative quantization is conducted via two steps on server – apply k-means based adaptive quantization on learned network weights and retrain the network based on quantized weights. These two steps are alternated until the convergence criterion is met. In this testing stage, the network configuration and low-bit weights are loaded to the client hardware device to recognize coming input in real time, where optimized but expensive quantization becomes infeasible. Considering this, we adopt a uniform quantization for the inputs and internal network responses (called feature maps) to maintain low on-chip expenses. The Convolutional Neural Network with reduced weight and input/response precision is demonstrated in recognizing two types of images: one is hand-written digit images and the other is real-life images in office scenarios. Both results show that the new network is able to achieve the performance of the neural network with full bit resolution, even though in the new network the bit resolution of both weight and input are significantly reduced, e.g., from 64 bits to 4-5 bits.

Keywords: Convolutional neural networks, deep learning, image recognition, quantization

1. INTRODUCTION

Deep learning draws considerable research attentions in recent years [1]. Various deep learning architectures, such as deep convolutional neural networks [2], deep belief networks[3] and auto encoder decoders [4] have been developed to provide state-of-the-art solution in various artificial intelligence and computer vision tasks [2-6], including (large-scale) visual object recognition, automatic speech recognition, natural language processing, and music/audio signal processing. Till now, the main efforts of deep learning have been focused on the software implementation, in the aspects of network architectures, learning optimization and demonstration of applications with bench-marking performance. Yet the hardware implementation endowing more powerful on-board behaving is still limited. Transforming the advancement of deep learning from current software implementation to fast and compact chip solution, especially under mobile platform is of great importance for next-generation hand-held products and wearable devices.

Major challenges in achieving deep learning chips lie in the constraints of die size and power consumption. For instance, the software implementation may enjoy a cluster of high performance computers with huge storage space and unlimited power supplies, but this becomes implausible when considering a centimeter size chip running on a standalone battery. One efficient solution to reduce memory footprint and power consumption for calculation on chip is to reduce the precision of model configurations, i.e., the connection weight and internal responses for deep learning architecture. Amount of research work have been conducted in this direction by applying low precision constraints for the existing learning algorithm of neural networks. One early example is called continuous-discrete learning method, which obtained the output error from the discrete network and propagated the error back via the continuous network [7]. A similar approach has been applied to neural networks restricted to single power-of-two weights [8]. Takahashi et al. [9] also used a similar strategy but optimized the continuous-discrete learning using two additional heuristics to avoid local minimum. Later techniques including probabilistic rounding and dynamic rescaling [10, 11, 12] were proposed based on cascade correlation learning algorithm, to allow reliable convergence but with small or gradual reduction of original solution. Choi et al. [13] in the meanwhile introduced a modified random search technique, called improved bidirectional random optimization (IBRO), to improve the search accuracy per iteration and therefore help achieve global minimum

*zhengp.ji@ssi.samsung.com

Independent Component Analyses, Compressive Sampling, Large Data Analyses (LDA), Neural Networks, Biosystems, and Nanoengineering XIII, edited by Harold H. Szu, Liyi Dai, Yufeng Zheng, Proc. of SPIE Vol. 9496, 94960A · © 2015 SPIE · CCC code: 0277-786X/15/$18 · doi: 10.1117/12.2176598

with low-accuracy computational precision. Sakaue et al. [14] further proposed to use a weighted error function (other than the squared error) to alleviate rounding error and eliminate useless weight update, resulting in a reduction of required precision bits for backpropagation.

Another line of research is to quantize the developed network weights to achieve low-bit solution. A series of studies provided theoretical analysis regarding how weight quantization may affect the performance of neural networks [15, 16, 17]. A notable example in practice is the soft weight sharing [18], which treated the weights not as constant numbers but as random variables drawn from a Gaussian mixture distribution, where k-means clustering and uniform quantization were considered as special cases [19, 20]. In general, many of quantization methods can be applied to this line of research to reduce weight precision. However, as far as we know, most of them are still based on the quantization error with respect to the original input signal, e.g., the network weights in our case. How to optimize the quantization scheme with respect to the performance of the network is still an open question.

In this paper, we developed a new quantization algorithm called supervised iterative quantization (SIQ), driven by an objective to minimize the network output error. In the training stage, the proposed SIQ is used to reduce the bit resolution of neural network weights without compromising the performance, which involves intensive data learning process and thus is performed on the sever. In the testing stage, the network configuration and low-bit weights are loaded to the client hardware device to recognize coming input in real-time, where optimized but expensive quantization becomes infeasible. Considering this, we adopted a uniform quantization for the input and internal network responses to maintain low on-chip expenses.

Such a server-client quantization scheme is applied to a popular deep learning architecture – Convolutional Neural Networks (CNN), enabling a deep learning system running with low bit resolution and in the meanwhile maintaining good performance for image recognition task. The Convolutional Neural Network with reduced weight and input/response precision is demonstrated in recognizing two types of images. One is hand-written digit images and the other is real-life images in office scenarios. Both results show that the network is able to maintain the original performance even though the bit resolution of both weights and inputs are largely reduced, e.g., from 64 bits to 4-5 bits.

2. NETWORK ARCHITECTURE

A Convolutional Neural Network (CNN) contains a hierarchy of convolutional layers, which can be interlaced by pooling layers (i.e., subsampling layers) and normalization layers. Each convolutional layer consists of a set of weight kernels (also called filters). In particular to our case, the input is an image and each 2D filter is convolved with the entire image. The convolutional operators are functioned to extract different features of the input. The features are presented as a hierarchical fashion, from simple features (like edges, lines and corners) in lower level to complex features (representing output) in higher level. Responses of each convolving filter pass through an activation function to form the output feature map. Representative activation functions include sigmoid function $f(x)=1/(1+e^{(-kx)})$, hyperbolic tangent $f(x)=\tanh(x)$ and rectified linear unit (ReLU) [21] $f(x)=\max(0,x)$. In this paper, ReLU activation function is used, which shows good performance in recent empirical studies.

In order to reduce variance, a pooling (also called subsampling) layer conducts maximum or average subsampling within an output feature map or across different output feature maps to provide neural responses feeding to the next convolutional layer. This ensures that the same result will be obtained, even when image features have small translations. After that, we enable a local normalization layer [2] to run over pooling outputs in “adjacent” feature maps at the same spatial position, in order to permit the detection of high-frequency features with a big neuron response, while suppressing responses that are uniformly large in a local neighborhood. This sort of response normalization implements a form of lateral inhibition inspired by the type found in real neurons, creating competition for big activities amongst neuronal outputs computed using different feature kernels.

The last few layers of CNN usually consist of fully connected layers, used to map feature representation from the hierarchy of convolutional/pooling/normalization operations to the network output. Since a fully connected layer occupies a large number of the parameters, it is prone to overfitting. The dropout method [22] is used to prevent overfitting in this paper, and also shown to significantly improve the speed of training.

The parameters in Convolutional Neural Network (e.g., weights kernels of convolution layers and weights of fully connected layers) are learned via back propagation algorithm, such that given each input at *t*, the prediction error given ground truth is minimized,

$$\min_{\mathbf{w}_l,\mathbf{w}_{l-1},..\mathbf{w}_1} \sum_t \alpha\left(y^{(t)} - pred\left(F_l\left(\mathbf{w}_l, F_{l-1}\left(\mathbf{w}_{l-1}, F_{l-2}(\mathbf{w}_{l-2},...,F_1(\mathbf{w}_1,\mathbf{x}^{(t)})\right)\right)\right)\right)^2, \quad (1)$$

where $\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_l$ denote the weight kernels from convolutional layer 1 to layer *l*, and $F_1, F_2, ..., F_l$ denote the mapping function with respect to each convolutional layer, interlaced with optional pooling/normalization. *pred* is the mapping function shaped by fully connected layers, acting overall as the classification model.

3. SEVER-CLIENT QUNATIZATION

This section describes the sever-client quantization scheme based on the CNN architecture described above, in order to achieve low precision (in terms of resolution bits) on weight kernels, inputs and feature maps.

3.1 Processing Paradigm

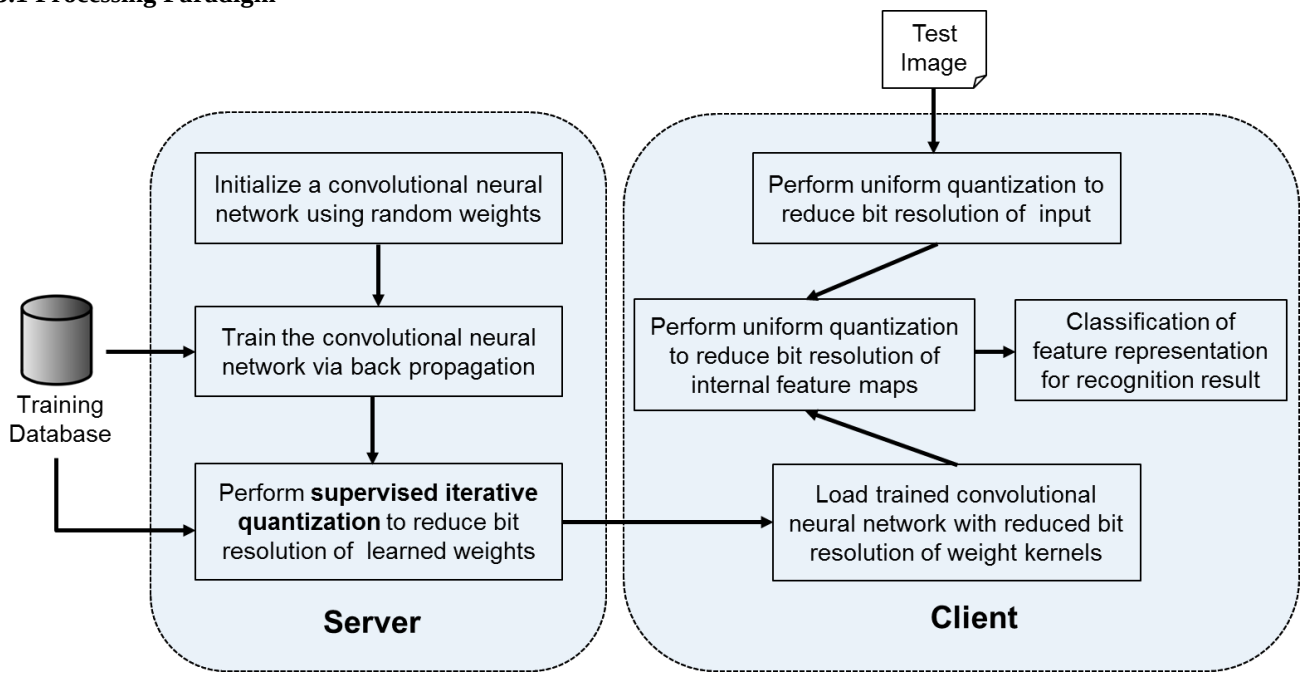

Fig. 1: Block diagram of server-client quantization to reduce precision (in terms of bit resolution) for weight kernels, inputs and internal feature maps in Convolutional Neural Networks.

Fig. 1 describes the proposed server-client quantization scheme for CNN in image recognition tasks. On the server cloud, we first train Convolutional Neural Network with full bit resolution (e.g., double precision float) using back propagation algorithm (Eq. (1))). Then we perform a new quantization method, called supervised iterative quantization (SIQ) to reduce the bit resolution of learned weight kernels. Because SIQ requires expensive computation, we prefer to perform it on server. After that, the network configuration and quantized filters are loaded to the client hardware device to recognize coming input. On the client side, however, we need to deal with input in real time for decision making. Thus, instead of expensive SIQ, we adopt a uniform quantization for the inputs and internal feature maps to guarantee low on-chip computational cost and fast running speed. To this end, the convolutional neural network with reduced weight precision conduct feedforward operation in the client -- receiving quantized image input, generating low-bit feature maps, and feeding final low-bit feature representation to classification model for recognition output.

3.2 Method

As described in Eq. (1), weight kernels of the Convolutional Neural Network are learned via back propagation algorithm driven by prediction errors. The newly proposed supervised iterative quantization (SIQ) is based on the same goal but in consideration of quantizing weight kernels at each level, forming a new objective function as follows:

$$\min_{Q_l, Q_{l-1}, \ldots, Q_1, \mathbf{w}_l, \mathbf{w}_{l-1}, \ldots, \mathbf{w}_1} \sum_t \left(y^{(t)} - pred\left(F_l\left(Q_l(\mathbf{w}_l), F_{l-1}\left(Q_{l-1}(\mathbf{w}_{l-1}), F_{l-2}(Q_{l-2}(\mathbf{w}_{l-2}), \ldots, F_1(Q_1(\mathbf{w}_1), \mathbf{x}^{(t)}) \right) \right) \right) \right)^2, \quad (2)$$

Solving Eq. (3) via direct gradient descent is hard, suffering the problems of flat-lining. More advanced searching method such as *reweighted least squares* and *stochastic gradient* are shown slow to converge in large problems with many input images. Instead of solving Eq. (2) directly, we used the Split Bregman method [23] to transform the above objective function to an equivalent as below

$$\min_{Q_l, Q_{l-1}, \ldots, Q_1, \mathbf{w}_l, \mathbf{w}_{l-1}, \ldots, \mathbf{w}_1} \sum_t \alpha \left(y^{(t)} - pred\left(F_l\left(\mathbf{w}_l, F_{l-1}\left(\mathbf{w}_{l-1}, F_{l-2}(\mathbf{w}_{l-2}, \ldots, F_1(\mathbf{w}_1, \mathbf{x}^{(t)}) \right) \right) \right) \right)^2 + \sum_{i=1,2,..,l} \beta \left\| Q_i(\mathbf{w}_i) - \mathbf{w}_i \right\|_2^2, \quad (3)$$

where α and β are continuation parameters. From Eq. (3), we can solve the original objective function via two alternating steps: (1) fix $\mathbf{w}$ and solve Q; and then (2) fix Q and solve $\mathbf{w}$, in an iterative fashion. Specifically, we first deploy k-means adaptive quantization [19] on trained network weights, to solve the minimization problem with regards to Q, i.e.

$$\min_{Q_l, Q_{l-1}, \ldots, Q_1} \sum_{i=1,2,..,l} \left\| Q_i(\mathbf{w}_i) - \mathbf{w}_i \right\|_2^2, \quad (4)$$

Then we retrain the network via back propagation (Eq. (1)) to solve the minimization problem with respect to $\mathbf{w}$, where quantized weights are used as initialization. The above two steps are alternated until a certain iteration number is reached.

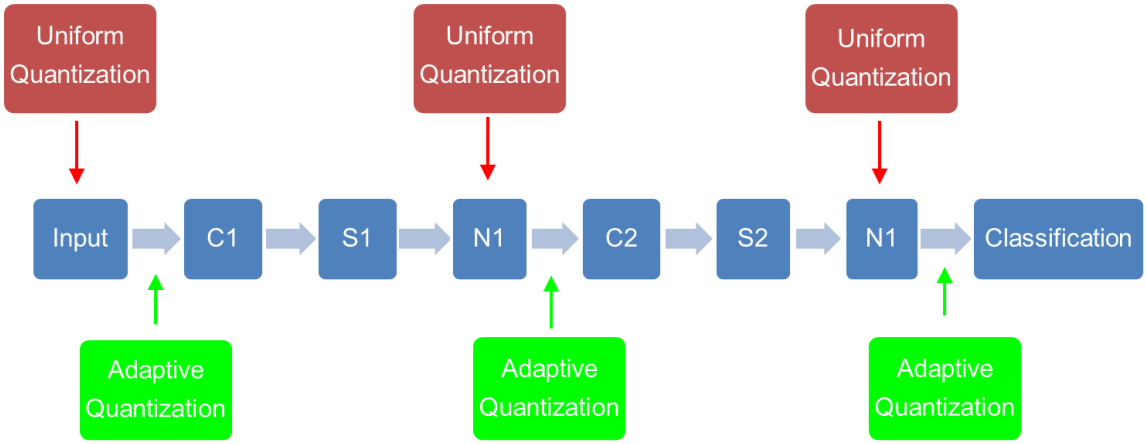

Fig. 2 Schematic figure of a Convolutional Neural Network architecture that contains two convolutional layers, coupled with subsampling and normalization layers **(**best viewed in color).

To summarize, the proposed supervised iterative quantization involves operation of bidirectional information flows: (1) feedforward operation for unsupervised adaptive quantization using k-means; (2) supervised feedback operation that tunes quantized weights to minimize the recognition performance error. Optimization on bidirectional information flows alternates to reach a converged solution. This iterative processing requires intensive computation, and thus is suitable to be conducted offline, on a server cloud in our case.

With low-bit weights achieved, we load the parameters to the client hardware device for object recognition. To further reduce computational load on board, we reduce bit resolution for input and internal feature maps. Due to the hardware constraints in power consumption and chip area, we want to minimize the cost for such quantization. Therefore, uniform quantization is adopted. The uniform quantization can be formalized as follows at each network level i ($i=1, 2, \ldots, l$), where we can see it is a close-form solution and no iteration is in need to solve it.

$$Q_i(\mathbf{w}_i) = \Delta \bullet \left(\left\lfloor \frac{\mathbf{w}_i}{\Delta} \right\rfloor + \frac{1}{2} \right) \tag{5}$$

Fig. 2 shows a schematic figure for the overall server-client quantization framework, where supervised iterative question (SIQ) is applied to weight kennels on server cloud and uniform quantization is applied to input and internal feature maps on device. The two together enable low-bit resolution of the convolutional neural network model, resulting in inexpensive computations favorable for object recognition hardware.

4. EXPERIMENTS

Experiments are conducted to evaluate the proposed quantization scheme enabling low-bit CNN to perform object recognition. The CNN with reduced precision is demonstrated in recognizing two types of images. One is hand-written digit images and the other is real-life images in office scenarios. Both results showed that the network is able to maintain the original performance even though the bit resolution of both weight and input are largely reduced.

4.1 MNIST Data

MNIST [24] is a well-known handwritten digit dataset (available at http://yann.lecun.com/exdb/mnist/) composed of 70,000 total images (60,000 training, 10,000 testing) with 10 classes of handwritten digits -- from 0 to 9. Each image is size-normalized to 28×28 = 784 dimensions. All images are translation-normalized, so that each digit resides in the center of the image. Fig. 3 shows examples from MNIST dataset

Fig. 3 Examples of hand written digit images used for simulation of recognition performance with low bit resolution.

We implement a CNN with two convolutional layers, each coupled with one pooling layer and one normalization layer. One fully connected layer (with 100 hidden units) is applied for classification purpose. The weight kernel size at each convolutional layer is 5×5 and pooling size at each pooling layer is 2×2. The number of kernels in the first convolutional is 6 and in the second is 12.

As shown in Fig. 4, we apply three different quantization schemes to quantize the weight kernels of CNN. One is uniform quantization (as defined in Eq. (5)), the other is k-means quantization (also called adaptive quantization) and the third is the proposed supervised iterative quantization. We evaluate the number of bits required for various recognition performances using the above three quantization schemes. Compared to the proposed supervised iterative quantization, the uniform quantization and k-means quantization need more bit resolution in order to reach the same performance. For example, given 4 bit resolution for input and internal feature maps, we only need another 4 bits from supervised iterative quantization (vs. 6 bits from uniform or adaptive quantization) to reach 1.15% recognition error that is originally produced by double precision computation (64 for weight and 64 for input).

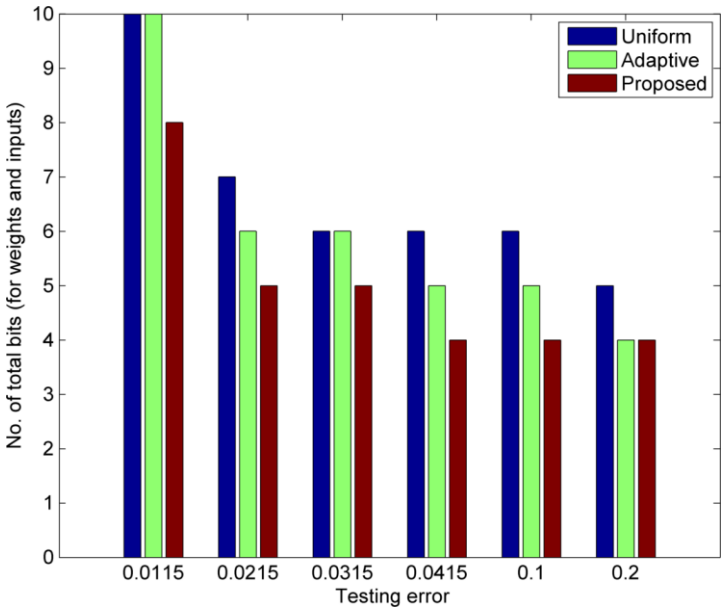

Fig. 4 Number of total bits required for weight and input in order to match/outperform original performance (i.e., 1.15% error with double precision computation) as well as other targeting performance (best viewed in color).

Fig. 5 further shows the average recognition error over low bit resolutions (i.e., 1-4 bits for input and internal feature maps, combined with 1-4 bits for weight kernels; 16 combination in total). For different quantization schemes, as we can see, the supervised iterative method leads to smaller recognition error compared to the other two.

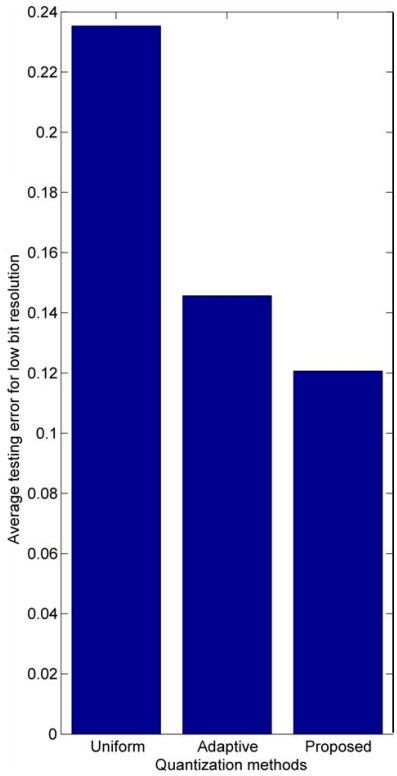

Fig. 5. Average recognition error of different quantization methods for MNSIT data.

4.2 Real-life Images

In this experiment, we evaluate the low-bit CNN for real-life images selected from ImageNet database [25]. ImageNet is a publicly available image database containing 15 million labeled images belonging to 22 thousand object categories, which are organized according to the WordNet hierarchy of meaningful concepts. Among the 22 thousand categories, we select 20 categories and their associated images in office scenarios to conduct object recognition using the low-bit CNN. Fig. 6 shows examples from 7 categories. We first convert all the images to gray scale, and then resize the smaller dimension of the image to be 32 pixels, with image aspect ratio retained. A set of 32×32 subimages (obtained by shifting one pixel along the longer dimension of the resized image) is extracted, and assigned with the same label. In testing, we average the predictions made by the network on the 32×32 subimages. This data augmentation process helps avoid overfitting problem of CNN.

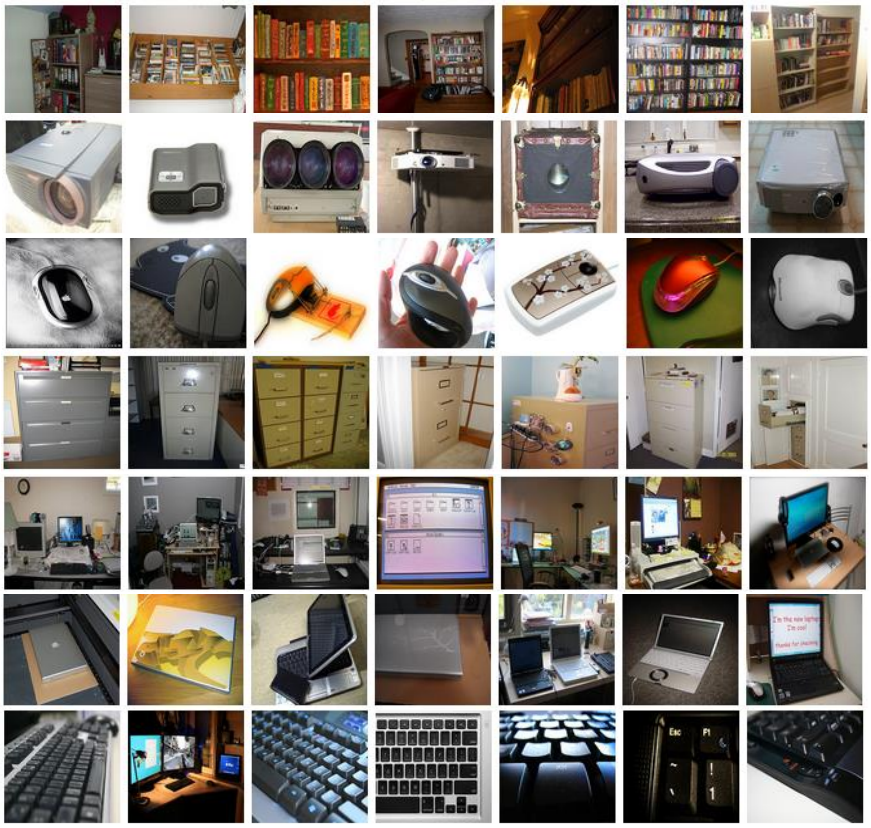

Fig. 6. Examples of selected images under office scenario. Images in each row belong to one of 7 classes, i.e., from top to bottom, bookcase, projector, mouse, computer desk, laptop and mouse.

We implement a CNN with three convolutional layers, each coupled with one pooling layer and one normalization layer. One fully connected (with 300 hidden units) is applied for classification purpose. The weight kernel size at each convolutional layer is 5×5 and pooling size at each pooling layer is 2×2. The number of kernels in the first convolutional is 12, in the second is 24 and in third is 32.

Table 1 shows recognition performance using different combination of weight bits and input bits. Given a fixed bit resolution for input and feature maps, again, we apply different quantization schemes to quantize the weight kernels. One is uniform quantization (whose recognition error is shown in "red" row of the table), the other is k-means quantization (whose recognition error is shown in "blue" row of the table) and the third is the proposed supervised iterative quantization (whose recognition error is shown in "green" row of the table). As we can see, the proposed

supervised iterative quantization constantly performs better (with less recognition error) than the uniform quantization and adaptive (k-means) quantization, based the same bit resolution for input/feature maps and weight kernels.

Table 1. Recognition error of office images at different set of resolution bits for input and weight (best viewed in color).

		Input resolution bit								
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**64**
Weight resolution bit	**1**	0.7650	0.6822	0.6264	0.6368	0.6411	0.6466	0.6436	0.6515	0.6577
		0.6828	0.4779	0.3757	0.2782	0.2653	0.2680	0.2671	0.2653	0.2625
		0.6699	0.4583	0.2650	0.1988	0.1896	0.1963	0.1969	0.1963	0.1933
	2	0.7000	0.5000	0.4368	0.3239	0.3607	0.3552	0.3466	0.3417	0.3442
		0.6613	0.4301	0.2485	0.1834	0.1748	0.1767	0.1761	0.1748	0.1730
		0.6472	0.3613	0.1742	0.1337	0.1239	0.1227	0.1221	0.1215	0.1233
	3	0.6712	0.4712	0.3049	0.1847	0.1706	0.1528	0.1479	0.1405	0.1436
		0.6552	0.3331	0.1528	0.1172	0.1172	0.1172	0.1166	0.1166	0.1160
		0.6374	0.3399	0.1497	0.1172	0.1061	0.1055	0.1031	0.1025	0.1031
	4	0.6693	0.4460	0.2638	0.1558	0.1405	0.1147	0.1147	0.1110	0.1104
		0.6540	0.3374	0.1497	0.1135	0.1110	0.1184	0.1172	0.1184	0.1172
		0.6374	0.3288	0.1442	0.1135	0.1025	0.0963	0.0982	0.0969	0.0969

5. CONCLUSION

In this paper, we proposed a server-client quantization scheme to reduce bit resolution of deep learning architecture, e.g. convolutional neural networks, where uniform quantization is applied to the input and internal feature maps, and supervised iterative quantization is applied to weight kernels. Rather than minimizing reconstructive error to original signal (i.e., the weigh kernels in CNN), the supervised iterative quantization aims at maximizing the performance, i.e., accuracy of the whole network. The performance driven regularity enables the supervised iterative quantization to develop low bit weight kernels without compromising its performance in visual object recognition tasks. As the proposed supervised iterative quantization involves alternating the minimization of concurrent bidirectional information flows, the iteration is necessary to reach a converged solution. Such expensive computation is not applicable to perform on client but more preferably to perform on server cloud as offline training. The future work will lie in optimizing current supervised iterative quantization to enable online training, and in meanwhile exploring efficient and also more effective quantization schemes (other than uniform quantization) for input and internal feature maps. The quantization of fully connected weights is not discussed in this paper, where we fix 8 bit resolution throughout the experiments. Although the current supervised iterative quantization is applicable to the fully connected layer, a more dynamic control and detailed tuning in contrast to convolutional layers is required. This is under investigation and will be presented in future work.

REFERENCES

[1] Y. Bengio, “Learning Deep Architectures for AI”, Foundations and Trends in Machine Learning, 2009

[2] A. Krizhevsky, I. Sutskever, G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks”, NIPS 2012.

[3] G. E. Hinton, S. Osindero, and Y. Teh, “A Fast Learning Algorithm for Deep Belief Nets”, Neural Computation, vol 18, 2006

[4] P. Vincent, H. Larochelle Y. Bengio, I. Lajoie and P. A. Manzagol, “Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion.” The Journal of Machine Learning Research 11 (2010): 3371-3408.

[5] M. D. Zeiler and R. Fergus, “Visualizing and Understanding Convolutional Networks”, ECCV 2014.

[6] Z. Ji, “Decoupling Sparse Coding with Fusion of Fisher Vectors and Scalable SVMs for Large-Scale Visual Recognition”, Computer Vision and Pattern Recognition Workshops (CVPRW), 2013.
[7] F. Choudry, E. Fiesler, A. Choudry and H. J. Caulfield, “A Weight Discretization Paradigm for Optical Neural Networks”, Proceedings of the International Congress on Optical Science and Engineering, 1990.
[8] C. Z. Tang and H. K. Kwan, “Multilayer Feedforward Neural Networks with Single Power-of-Two Weights”, IEEE Transactions on Signal Processing, vol. 41, no. 8, pp. 2724-2727, August 1993.
[9] M. Takahashi, M. Oita, S. Tai, K. Kojima, and K. Kyuma, “A Quantized Back Propagation Learning Rule and its Application to Optical Neural Networks”, Optical Computing and Processing; The Science and Technology of Optics in Computing, Communications, Switching and Information Processing, vol. 1, no. 2, pp. 175-182, 1991.
[10] M. Hoehfeld and S. E. Fahlman, “Learning with Limited Numerical Precision Using the Cascade-Correlation Algorithm”, IEEE Transactions On Neural Networks, Vol. 3, No. 4, July 1992.
[11] J. Vincent and D. Myers, “Weight Dithering and Wordlength Selection for Digital Backpropagation Networks”, BT Technology Journal, 124–133
[12] Y. Xie and M. A. Jabri, “Training Algorithms for Limited Precision Feedforward Neural Networks”, SEDAL Technical Report No. 1991-8-3, 1991
[13] J. J. Choi, S. Oh and R. J. Marks II, “Training Layered Perceptrons Using Low Accuracy Computation”, IJCNN 1991.
[14] S. Sakaue, T. Kohda, H. Yamamato, S. Maruno, and Y. Shimeki, “Reduction of Required Precision Bits for Back Propagation Applied to Pattern Recognition,” IEEE Transactions on Neural Networks, Vol. 4, pp. 270–275, 1993.
[15] Y. Xie, M. A. Jabri, “Analysis Of The Effects of Quantization in Multilayer Neural Networks Using A Statistical Model”, IEEE Trans. Neural Networks, 3, pp. 334-338, 1992.
[16] D. Wu and J. N. Gowdy, “Error Analysis of Quantized Weights For Feedforward Neural Networks (FNN)”, SOUTHEASTCON, 1994
[17] G. Dundar and K. Rose, “The Effects of Quantization on Multilayer Neural Networks”, IEEE Trans. Neural Networks, 6, pp. 1446-1451, 1995.
[18] F. Köksal, E. Alpaydin, G. Dündar, “Weight Quantization for Multi-layer Perceptrons Using Soft Weight Sharing”, ICANN 2001
[19] E. W. Forgy, “Cluster Analysis of Multivariate Data: Efficiency versus Interpretability of Classifications", Biometrics 21: 768–769, 1965.
[20] Allen Gersho and Robert M. Gray, “Vector Quantization and Signal Compression”, Springer, ISBN 978-0-7923-9181-4, 1991.
[21] V. Nair and G. E. Hinton, “Rectified Linear Units Improve Restricted Boltzmann Machines”, ICML 2010.
[22] N. Srivastava, C. Nitish, H. Geoffrey Hinton, A. Krizhevskyy, I. Sutskever and R. Salakhutdinov, “Dropout: A Simple Way To Prevent Neural Networks From Overfitting”, Journal of Machine Learning Research 15 (1): 1929–1958, 2014
[23] T. Goldstein and S. Osher, “The Split Bregman Method for L1-Regularized Problems” SIAM J. Imaging Sci., 2(2), 323–343, 2009
[24] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-Based Learning Applied to Document Recognition”, Proceedings of IEEE, 86(11):2278–2324, 1998.
[25] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li and L. Fei-Fei, “ ImageNet: A Large-Scale Hierarchical Image Database”, CVPR 2009