Quantization of Deep Neural Networks for Accurate Edge Computing

WENTAO CHEN*, HAILONG QIU*, and JIAN ZHUANG, Guangdong Academy of Medical Sciences
CHUTONG ZHANG* and YU HU, Easylink Technology Co., Ltd.
QING LU, TIANCHEN WANG, and YIYU SHI, University of Notre Dame
MEIPING HUANG and XIAOWE XU, Guangdong Academy of Medical Sciences

Deep neural networks have demonstrated their great potential in recent years, exceeding the performance of human experts in a wide range of applications. Due to their large sizes, however, compression techniques such as weight quantization and pruning are usually applied before they can be accommodated on the edge. It is generally believed that quantization leads to performance degradation, and plenty of existing works have explored quantization strategies aiming at minimum accuracy loss. In this paper, we argue that quantization, which essentially imposes regularization on weight representations, can sometimes help to improve accuracy. We conduct comprehensive experiments on three widely used applications: fully connected network for biomedical image segmentation, convolutional neural network for image classification on ImageNet, and recurrent neural network for automatic speech recognition, and experimental results show that quantization can improve the accuracy by 1%, 1.95%, 4.23% on the three applications respectively with 3.5x-6.4x memory reduction.

CCS Concepts: • **Computing methodologies** → **Artificial intelligence**; • **Hardware** → **Emerging technologies;**

Additional Key Words and Phrases: Edge computing, deep neural networks, quantization

*Both authors contributed equally to this research. Wentao Chen works at Easylink Technology Co., Ltd, and this work is done when he was an visiting scholar at Guangdong Provincial People's Hospital.

This work was supported by the National Key Research and Development Program of China (2018YFC1002600), the Science and Technology Planning Project of Guangdong Province, China (2017B090904034, 2017B030314109, 2018B090944002, 2019B020230003), Guangdong Peak Project (DFJH201802), and the National Natural Science Foundation of China (62006050).

Authors' addresses: W. Chen, H. Qiu, J. Zhuang, M. Huang (corresponding author), and X. Xu (corresponding author), Guangdong Cardiovascular Institute, Guangdong Provincial Key Laboratory of South China Structural Heart Disease, Guangdong Provincial People's Hospital, Guangdong Academy of Medical Sciences, 106 Zhongshan Second Road, Guangzhou, Guangdong, 510080; emails: chenwentaokl@gmail.com, hailong0518@qq.com, Zhuangjian5413@163.com, huangmeiping@126.com, xiao.wei.xu@foxmail.com; C. Zhang and Y. Hu, Easylink Technology Co., Ltd., Wuhan, Hubei, 43000; Q. Lu, T. Wang, and Y. Shi (corresponding author), Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556; emails: {qlu2, twang9, yshi4}@nd.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2021 Association for Computing Machinery.

1550-4832/2021/06-ART54 $15.00
https://doi.org/10.1145/3451211

ACM Reference format:
Wentao Chen, Hailong Qiu, Jian Zhuang, Chutong Zhang, Yu Hu, Qing Lu, Tianchen Wang, Yiyu Shi, Meiping Huang, and Xiaowe Xu. 2021. Quantization of Deep Neural Networks for Accurate Edge Computing. *J. Emerg. Technol. Comput. Syst.* 17, 4, Article 54 (June 2021), 11 pages.
https://doi.org/10.1145/3451211

1 INTRODUCTION

Deep neural networks (DNNs) have been widely used in various applications and show great potential to tackle complex problems [1–3, 19, 26, 32]. Furthermore, DNNs have been and continue to be instrumental in enabling/advancing breakthroughs in various disciplines, including disease diagnosis, real-time language translation, and autonomous driving, among others [8, 9, 15, 20, 22, 25, 30, 31, 33, 33]. Meanwhile, edge computing for the Internet of Things has been widely studied, which requires DNN models with small memory size and efficient computation [17, 27, 28]. Thus, there is a huge gap between current DNN models and the requirement of edge computing.

Recently, to accommodate DNNs on the edge, DNN quantization has become an active research topic [4, 5, 7, 11, 13, 13, 16, 18, 34–36], which aims to represent DNN weights with less memory (precision) while maintaining acceptable accuracy with efficient memory and computation costs. It has been observed in the literature, however, that sometimes quantization can improve accuracy that can be credited to the reduction of overfitting [23]. Dynamic fixed point [29] can achieve 4x less memory operation cost with only 0.4% to 0.6% top-5 accuracy loss for ImageNet classification [6]. Ternary weight network and BinaryConnect [29] have further reduced the bit-width of weights to 2 bits or even 1 bit with a relatively larger accuracy loss. Recently, their enhanced version, trained ternary training and binary weight network [29] have reduced the accuracy loss to only 0.6% to 0.8%. There also exist some works using non-linear quantization to represent the parameter distribution for better accuracy [29]. Unlike the preceding works, some studies aims to quantize not only the weights but also the activations. Quantized neural networks, binarized neural networks, and XNOR-net [29] reduced the weights to only 1 bit and the activations to 1 to 2 bits, resulting in a large reduction on memory and computation cost yet with significant accuracy loss. Two-step quantization [24] has 2% drop for VGG-16 on the ImageNet-2012 dataset, and Integer-Arithmetic-Only quantization[14] has 3.1% accuracy loss. In some of the preceding works, we notice that quantization can sometimes improve performance [29]; however, there are no comprehensive studies to verify this point.

It is generally believed that quantization of DNN weights lead to performance degradation. For example, two-step quantization [24] has a 2% accuracy drop on the ILSVRC-2012 dataset and Integer-Arithmetic-Only quantization [14] has a 3.1% accuracy loss. Various works in the literature have explored the best ways to quantize the weights with minimum accuracy loss. To some extent, quantization essentially imposes regularization on weight representations. As such, it should also sometimes help to improve accuracy when the overfitting issue presents in large neural networks.

To demonstrate this point, in this article we conduct extensive experiments using incremental quantization on three applications: medical image segmentation, image classification, and automatic speech recognition. For fair comparison, we have re-implemented related methods and performed evaluations under the same hardware, deep learning frameworks, and configurations. In medical image segmentation, our method can achieve 1% accuracy improvement compared with the current state-of-the-art method on the MICCIA Gland dataset. In image classification, extensive experiments are presented on ImageNet-2012 using a widely used **convolutional neural network (CNN)** model VGG-16 [21], and the result shows that our proposed method exceeds the current best performance using VGG-16 by up to 1.95%. In automatic speech recognition, we quantize the Deep Speech [12] network on the TIMIT dataset [10] and improve the accuracy by 4.23%. We also discuss the incremental quantization on the performance of a simplified network with different bit widths, and the experimental results show that incremental quantization can no longer improve the performance of simplified networks with less overfitting. In addition, we get 3.5x to 6.4x memory reduction, which is extremely beneficial in the context of edge computing.

2 PRELIMINARIES

In this section, we briefly review the incremental quantization method used in the experiments. The main goal of incremental quantization is to convert 32-bit-floating-point weights W into low-precision weights $\hat{W}$ either power of 2 or zero with minimum accuracy loss. Each of $\hat{W}$ is chosen from $P_l = \{\pm 2^{n_1}, \ldots, \pm 2^{n_2}\}$, where n_1 and n_2 are two integer numbers determined by the max absolute value of W_l and expected quantized bit-width, and $n_2 \leq n_1$.

The training procedure of incremental quantization [34] consists of three operations: weight partition, group-wise quantization, and re-training. Weights partition is the most crucial part of the whole process. We adopt a pruning-inspired partition strategy to divide weights into two disjoint groups by comparing the absolute value with a layer-wise threshold, which usually achieves better performance than random partition [34].

For the l-th layer, we define weights partition as shown in Equation (1), where $A_l^{(1)}$ represents the first group weights that need to be quantized and $A_l^{(2)}$ denotes the second group with remaining unquantized values.

$$\begin{aligned} A_l^{(1)} \cup A_l^{(2)} &= W_l(i,j) \\ A_l^{(1)} \cap A_l^{(2)} &= \emptyset \end{aligned} \tag{1}$$

The weights in the first group are expected to form a low-precision base for the original model, and thus they are quantized using Equation (2):

$$w^q = \begin{cases} sign(w) \times 2^p & if\ 3 \times 2^{p-2} \leq |w| \leq 3 \times 2^{p-1}; \\ sign(w) \times 2^m & if\ |w| \geq 2^u; \\ 0 & if\ |w| < 2^{-l-1} \end{cases} \tag{2}$$

where w^q are quantized weights, w represents original weights, u and l are upper and lower bounds of the quantized set($l \leq p \leq u$).

By contrast, the second group remains as floating-point values to compensate for the accuracy loss in model and will be re-trained. After one round, these three steps are further adopted only on the second group in the rest of the training process. As a result, all weights are quantized or to zeros, and hence we gain memory reduction with slight accuracy loss.

3 CASE STUDIES

3.1 Biomedical Image Segmentation

3.1.1 Network Quantization. As shown in Figure 1, the network quantization for biomedical image segmentation has two steps: **suggestive annotation (SA)** with quantization and network training (NT) with quantization. In the first step, we add a quantization module to suggestive **fully connected networks (FCNs)** for high uncertainty. In the second step, quantization of segmentation FCNs are performed with the suggestive training samples for higher accuracy. The FCN is a 34-layer network, and more details are presented in the work of Yang et al. [32]. To obtain high representativeness, each FCN in suggestive FCNs should be diverse for high uncertainty with

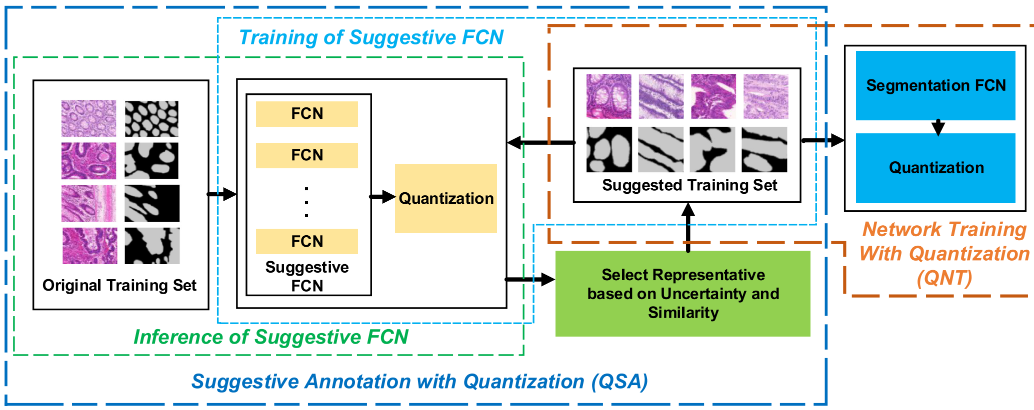

Fig. 1. Illustration of quantization framework based on the SA framework. In SA with quantization, better training samples (suggestive training set) can be extracted from the original training set. In network training with quantization, better performance can be achieved by reducing overfitting.

acceptable accuracy. However, usually DNNs including FCNs are over-parameterized, and a large portion of the parameters is redundant. Thus, multiple suggestive FCNs will have very small variance of the final prediction but with different weight initialization. The adopted regularization techniques including weight decay and dropout scheme further make the multiple suggestive FCNs almost the same. By adding quantization to SA, the preceding requirement can be satisfied. Although it may be a little offensive since most of the time it will degrade accuracy, it is particularly appreciated by suggestive FCNs that focus on uncertainty. In particular, quantization transforms the originally continuous weight space, where the weights of several networks can be arbitrarily close, into a sparse and discrete one, thus increasing the distances between the trained networks and accordingly the diversity of the outputs. Note that accuracy should be also considered, and too offensive quantization methods should be avoided.

3.1.2 Experimental Setup. We adopt the 2015 MICCAI Gland Challenge dataset [22], which has 85 training images (Part A: 37 normal glands; Part B: 48 abnormal glands) and 80 testing images (Part A: 60 normal glands; Part B: 20 abnormal glands). In SA [32], 16 images with the highest uncertainty scores are extracted first, and then 8 images are collected based on their representativeness using similarity, which are added to the suggested training set in each iteration. In total, there are 120 iterations in SA, and a total of 960 suggested training samples are produced. We quantize the weights from 2 bits to 9 bits. For the ensembling method, we test from two FCNs to seven FCNs. We also discuss the overfitting problem with a simplified version of FCN by reducing the filter numbers by half. All experiments are evaluated considering detection (F1 score) and segmentation (dice score) [29], and their averaged results are reported.

The training parameters are as follows. We adopt a simple learning rate scaling strategy: set the learning rate to 5×10^{-4} in the initial stage and to 5×10^{-5} when the iteration time reaches a threshold. All configurations are repeated four times, and the one with the optimal performance is selected for comparison.

3.1.3 Results and Analysis. As there are two networks in SA, we discuss both of them with incremental quantization. As shown in Table 1, we first analyze the performance of quantization methods in SA with quantization and NT with quantization. We can notice that compared with

Table 1. Segmentation Accuracy (Averaged Dice Score and F1 Score in %) Using Different Quantization Bit-Width on the MICCAI Gland Dataset

Configurations	Bit-Width								
	Original	2 Bits	3 Bits	4 Bits	5 Bits	6 Bits	7 Bits	8 Bits	9 Bits
Float SA + Quantized NT	86.25	65.55	79.37	85.87	86.12	86.33	**86.51**	85.93	85.26
Quantized SA + Float NT	86.25	62.98	85.39	86.17	86.52	**86.66**	86.51	86.22	86.12

The number in bold represents the best performance.

Table 2. Segmentation Accuracy (Averaged Dice Score and F1 Score in %) Using Different Parallel FCNs on the MICCAI Gland Dataset

Configurations	Parallel number						
	Original	2	3	4	5	6	7
Float SA + Float NT	86.25	86.23	85.93	85.81	86.24	**87.25**	86.32
Float SA + 7 bits NT	86.25	86.2	85.39	85.32	**86.84**	86.08	85.74

The number in bold represents the best performance.

Table 3. Segmentation Accuracy (Averaged Dice Score and F1 Score in %) Using Different Bit-Width and Five Parallel FCNs on the MICCAI Gland Dataset

Configurations	Bit-Width								
	Original	2 Bits	3 Bits	4 Bits	5 Bits	6 Bits	7 Bits	8 Bits	9 Bits
Float SA + 5 FCNs	87.85	64.40	85.94	88.20	88.51	88.77	**89.12**	87.86	87.55
7 bits SA + 5 FCNs	87.55	71.88	80.13	87.55	88.63	88.73	**89.2**	88.67	87.53

The number in bold represents the best performance.

Table 4. Segmentation Accuracy (Averaged Dice Score and F1 Score in %) Using Quantization with 4 Bits and 8 Bits with Small Models

Network	Float	4 Bit	8 Bit
Simplified (Quantized SA + Float NT)	**82.53**	78.14	82.14
Simplified (Float SA + Quantized NT)	**82.53**	76.12	81.27

The number in bold represents the best performance.

the original setting, quantization can improve accuracy by 0.26% and 0.31% for quantized NT and quantized SA, respectively.

As shown in Table 2, we can discover that the ensembling method can improve segmentation accuracy most of the time. Similar to Table 1, the optimal accuracy is achieved in some median parallel number (e.g., 6 for float SA + 7 bits NT).

As shown in Table 3, we further present quantization on both SA and NT with the optimal parallel number. The trend is the same as that in Table 1, and the median bit width (7 bits) obtains the best performance. With both the quantization and the ensemble methods, we can achieve a promising performance on the MICCAI 2015 Gland dataset, which outperforms the state-of-the-art method by 1%. In addition, our method can also obtain 4.6x and 6.4x reduction on memory usage for incremental quantization with 7 bits and 5 bits, respectively. Note that as activations are in floating-point representation, the runtime is not affected.

As shown in Table 4, we also present quantization with simplified networks, such as small models. We can notice that when the network model is smaller, the segmentation accuracy with 4 bits and 8 bits is no longer improved; however, there is significant accuracy degradation. This is due to the fact that a smaller network with less parameters has less overfitting, and quantization further reduces the network's representation capability, resulting in degraded performance.

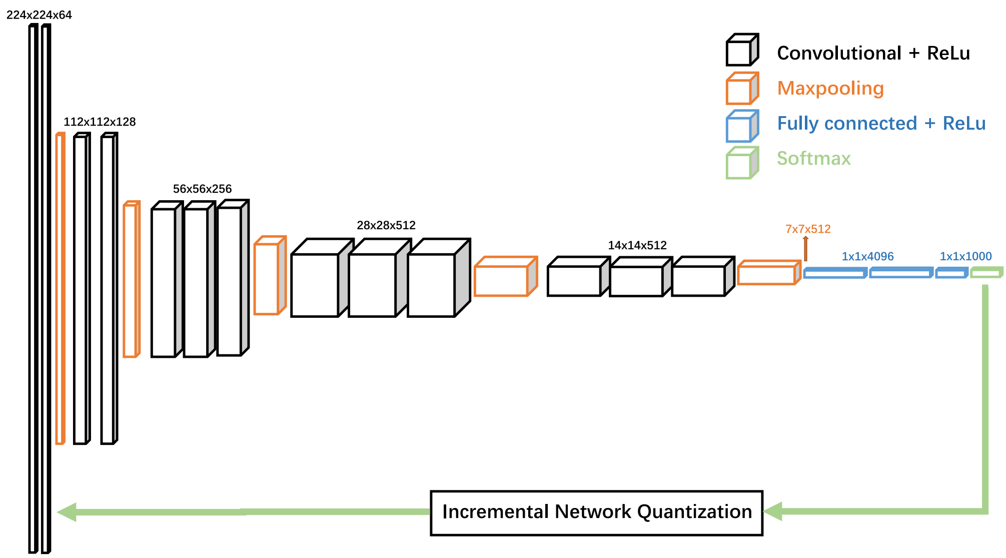

Fig. 2. Illustration of a quantization framework on the VGG-16 model. In small VGG models, we remove the first layers (with a feature map size of $112 \times 112 \times 128$) and halve the channels of the rest of the layers except for the first two input layers. In small VGG-16 models with five layers, we remove the third and fourth layers (with feature map sizes of $28 \times 28 \times 512$ and $14 \times 14 \times 512$, respectively). In small VGG-16 models with eight layers, we only remove the fourth layer.

3.2 Image Classification

3.2.1 Network Quantization. We use incremental quantization [34] on the VGG-16 model for image classification. The original VGG-16 has 16 layers (13 conv-layers and 3 FC layers) as shown in Figure 2. Besides that, we focus on the parameter-redundancy problem on VGG-16. We re-train two small VGG-16 models with the first 5 convolutional layers of the original 8 layers) and the first 8 convolutional layers of the original 11 layers), respectively, for overfitting analysis.

3.2.2 Experimental Setup. We adopt the ILSVRC-2012 dataset [6], which contains more than 1.2 million training images and more than 50,000 testing images. VGG-16 [21] is adopted for classification. We quantize the weights from 2 bits to 7 bits for discussion. Additionally, we test two VGG models and nine VGG models to further clarify the compensation of ensemble learning in terms of accuracy loss brought by weight quantization. For small VGG-16 models, we compare 4 bits quantization and 8 bits quantization with floating weights. The learning rate is set to 1×10^{-4} with a learning rate decay set as 1×10^{-6}. Stochastic gradient descent function is selected to be the optimizer and Momentun of 0.9 is used to accelerate training. To unify the variables and discuss how the bit-width impacts the network performance, the max value of the quantized weights is set to 4.0, and it takes four iterations from full floating-point weights to quantized weights, and the accumulated portions of quantized weights at four iterative steps are {50%, 75%, 87.5%, 100%}.

3.2.3 Results and Analysis. We compare our result with the original and small VGG-16 models. As shown in Table 5, the configuration with 8 bits obtains optimal performance, which gets a 1.1% improvement. We set a variety of ensemble experiments as the control group to discuss the effect of ensemble methods. The result of six configurations from two to seven parallel models using 32-bit floating-point weights are shown in Table 6. With three parallel models, we can get a 0.81% performance improvement. For the VGG-16 model with 8 bits representation and parallel

Table 5. Image Classification Performance (Top-1 Error in %) of the VGG-16 Model [14] with Weights from 2 Bits to 9 Bits on the ImageNet Dataset

Configurations	Quantization Bit-Width								
	Original	2 Bits	3 Bits	4 Bits	5 Bits	6 Bits	7 Bits	8 Bits	9 Bits
VGG-16	69.29	42.61	63.72	67.70	68.64	69.68	70.36	**70.48**	70.28

Table 6. Image Classification Performance (Top-1 Error in %) of the VGG-16 Model [14] with Parallel Numbers from 2 to 7 on the ImageNet Dataset

Configurations	Parallel Numbers						
	Original	2	3	4	5	6	7
VGG-16	69.29	70.18	**70.10**	70.38	69.31	69.82	68.58
VGG-16 with 8 bits	69.29	70.10	**71.24**	71.03	69.80	69.17	69.85

Table 7. Classification Performance (Top-1 Error in %) Comparison Between Floating Points, 4 Bits and 8 Bits Representations Using Small VGG-16 Models on ImageNet

Network	Float	4 Bit	8 Bit
VGG-16 with 5 conv-layers	**64.14**	64.12	64.05
VGG-16 with 8 conv-layers	**66.04**	66.00	66.02

numbers from two to seven, we get a 1.95% improvement with two parallel models compared with the original model. In addition, 4x memory reduction is obtained with quantization.

As for overfitting, we discuss the performance of 4 bits and 8 bits quantization on small VGG-16 models. We shorten the original VGG-16 model to contain 5/8 conv-layers with three FC layers. As shown in Table 7, we get 0.02% precision loss after quantization on both of the two simplified VGG-16 models. Comparing to the results in Table 5, it can be concluded that quantization can benefit the over-parameterized models to some extent. However, if the original model is already simplified, the inhibiting effect of quantization will be faded.

3.3 Automatic Speech Recognition

3.3.1 Network Quantization. We use incremental quantization [34] on Deep Speech for the speech recognition task. As shown in Figure 3, the original Deep Speech has five hidden layers, composed of three time-distributed FC layers, one bi-directional **recurrent neural network (RNN)** layer, and one time-distributed FC layer followed by one softmax layer as output. We remove two time-distributed FC layers before the RNN layer to simplify the model and re-train the small model on the DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus dataset (TIMIT).

3.3.2 Experiment Setup. We adopt TIMIT [10], which contains 6,300 sentences read by 630 different Americans for the Deep Speech network [12]. We quantize the weights to 2 bits to 7 bits, and we test two to nine models in the ensemble method. For training parameters, we set the learning rate to 1×10^{-4} with a learning rate decay of 1×10^{-6}. The predicted error rate will be evaluated by the word error rate.

3.3.3 Results and Analysis. As shown in Table 8, the quantized model with 9 bits obtains optimal performance, which is 3.43% higher than the model with floating-point representation. As shown in Table 9, with three parallel networks, we can get an improvement of 4.2%. Using two parallel models, our method obtains optimal performance among all configurations, which is 4.23%

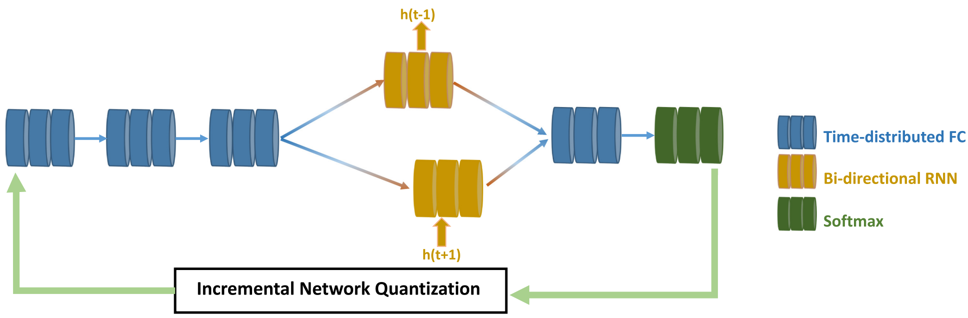

Fig. 3. Illustration of the quantization framework on Deep Speech. For small Deep Speech models, we remove the second and third FC layers.

Table 8. Automatic Speech Recognition Performance (Classification Error Rate in %) on Deep Speech [34] with Weights from 2 Bits to 9 Bits

Configurations	Quantization Bit-Width								
	Original	2 Bits	3 Bits	4 Bits	5 Bits	6 Bits	7 Bits	8 Bits	9 Bits
Deep Speech	48.13	60.33	64.30	69.53	52.93	47.20	49.50	52.40	**44.70**

Table 9. Automatic Speech Recognition Performance (Classification Error Rate in %) on Deep Speech [34] with Parallel Numbers from 2 to 7

Configurations	Parallel Numbers						
	Original	2	3	4	5	6	7
Deep Speech	48.13	45.37	47.30	**43.93**	42.83	46.23	43.30
Deep Speech with 9 bits	48.13	**43.90**	46.17	48.07	48.83	53.90	45.53

Table 10. Automatic Speech Recognition Performance (Classification Error Rate in %) Comparison Between Floating Points, 4 Bits and 8 Bits Representations Using Small Deep Speech Models

Network	Float	4 Bit	8 Bit
Small Deep Speech	50.04	**47.05**	47.08

higher than the original method, which is 0.03% better than the ensemble model. In addition, 3.6× reduction on memory usage can be achieved.

In Table 10, we present the quantization effectiveness on small Deep Speech models using 4 bits and 8 bits. Our model can get an improvement of 3% compared with the original model.

3.4 Discussion

As for accuracy, we can notice that incremental quantization can improve the segmentation performance by around 1% on the MICCIA Gland dataset. For image classification, we ended up using the 8-bit fixed-point VGG-16 model for two parallel models to achieve a 1.95% improvement compared with the original model. For speech recognition, the optimal performance obtained an improvement of 4.23% over the original Deep Speech model. In addition, our method has up to 3.5× to 6.4× reduction on memory usage. By comparing network training with and without quantization in Table 1, Table 5, and Table 8, we can observe that network training with quantization will not always improve accuracy, especially in extremely situations with 2 to 4 bits representation. In addition, too large bit widths do not improve performance, and optimal performance is achieved with some median bit width (e.g., 6 bits and 7 bits). Thus, with proper incremental quantization, accuracy can usually be improved with memory reduction.

As for the impact on the ensemble method, six parallel FCNs, two parallel CNNs, and two parallel RNNs with quantization reach optimal performance on segmentation, classification, and automatic speech recognition, respectively. In addition, optimal accuracy is achieved in some median parallel number. Thus, a proper parallel number can improve accuracy by scarifying some memory operation and computation cost.

For overfitting, we discuss the corresponding small models in the three applications with quantization. As shown in Table 4, Table 7, and Table 10, we can notice that when the model is much smaller than the original networks, accuracy degrades seriously, such as the results in medical image segmentation and image classification. When the model is only simplified a bit, such as the small model in automatic speech recognition, accuracy is still improved. This is possibly due to the fact that large models usually have more overfitting than small ones. Thus, when quantization is applied, the overfitting in large models is reduced, resulting in improved performance. However, for small models with less overfitting, the overfitting is reduced and the representation capability is also degraded, resulting in accuracy loss. Therefore, incremental quantization may be used as a regulation method to reduce overfitting.

Incremental quantization not only reduces memory consumption by 3.5× to 6.4× but also speeds up the processing. As the optimization mainly comes from transforming multiplication to addition in the convolution calculation, specific hardware modules are required for speedup. Thus, the speedup depends on specific hardware, such as the CPU, GPU, and FPGAs. If no specific hardware module is implemented for such transforming, no speedup is obtained (e.g., on existing CPUs and GPUs). If we implement a specific hardware module for 2D convolution operation on FPGAs, the speedup can be 1.7× to 7.8× according to Xu et al. [28]. Note that the speedup depends on a variety of factors, such as hardware platforms and network structures.

4 CONCLUSION

Among DNNs, compression techniques such as weight quantization and pruning are usually applied before they can be accommodated on the edge. It is generally believed that quantization leads to performance degradation, and plenty of existing works have explored quantization strategies aiming at minimum accuracy loss. In this article, we show that quantization can sometimes help improve accuracy by imposing regularization on weight representations. We conduct comprehensive experiments on three widely used applications: FCN for biomedical image segmentation, CNN for image classification, and RNN for automatic speech recognition, and experimental results show that incremental quantization can improve accuracy by 1%, 1.95%, and 4.23% on the three applications, respectively, with 3.5x to 6.4x memory reduction. As a case in compression techniques, incremental quantization shows great potential to reduce overfitting, and there may exist some general rules and strategies to enable other compression techniques to have the capability of reducing overfitting. We encourage related researchers to explore this interesting topic, which is also our future work.

REFERENCES

[1] Beatriz Blanco-Filgueira, Daniel García-Lesta, Mauro Fernández-Sanjurjo, Víctor Manuel Brea, and Paula López. 2019. Deep learning-based multiple object visual tracking on embedded system for IoT and mobile edge computing applications. *IEEE Internet of Things Journal* 6, 3 (2019), 5423–5431.

[2] Hao Chen, Xiaojuan Qi, Jie-Zhi Cheng, and Pheng-Ann Heng. 2016. Deep contextual networks for neuronal structure segmentation. In *Proceedings of the 30th AAAI Conferrence on Artificial Intelligence (AAAI'16).* 1167–1173.

[3] Hao Chen, Xiaojuan Qi, Lequan Yu, and Pheng-Ann Heng. 2016. DCAN: Deep contour-aware networks for accurate gland segmentation. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16).* 2487–2496.

[4] M. Courbariaux and Y. Bengio. 2016. BinaryNet: Training deep neural networks with weights and activations constrained to+ 1 or -1. arXiv:1602.02830.

[5] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. 2015. BinaryConnect: Training deep neural networks with binary weights during propagations. In *Proceedings of Neural Information Processing Systems.* 3123–3131.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *Proceedings of CVPR'09.* IEEE, Los Alamitos, CA, 248–255.

[7] Yukun Ding, Jinglan Liu, and Yiyu Shi. 2018. On the universal approximability of quantized ReLU neural networks. arXiv:1802.03646.

[8] Marc Egger and Detlef Schoder. 2017. Consumer-oriented tech mining: Integrating the consumer perspective into organizational technology intelligence—The case of autonomous driving. In *Proceedings of the 50th Hawaii International Conference on System Sciences.*

[9] Ge Gao, Chengyan Wang, Xiaodong Zhang, Juan Hu, Xuedong Yang, He Wang, Jue Zhang, and Xiaoying Wang. 2017. Quantitative analysis of diffusion-weighted magnetic resonance images: Differentiation between prostate cancer and normal tissue based on a computer-aided diagnosis system. *Science China Life Sciences* 60, 1 (2017), 37–43.

[10] John S. Garofolo. 1993. TIMIT acoustic-phonetic continuous speech corpus. Web download. Linguistic Data Consortium, Philadelphia, PA.

[11] Song Han, Huizi Mao, and William J. Dally. 2015. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. arXiv:1510.00149.

[12] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, et al. 2014. Deep speech: Scaling up end-to-end speech recognition. arXiv:1412.5567.

[13] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. 2016. Quantized neural networks: Training neural networks with low precision weights and activations. arXiv:1609.07061.

[14] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. 2018. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2704–2713.

[15] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 2012. 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 1 (2012), 221–231.

[16] Fengfu Li, Bo Zhang, and Bin Liu. 2016. Ternary weight networks. arXiv:1605.04711.

[17] Zihao Liu, Xiaowei Xu, Tao Liu, Qi Liu, Yanzhi Wang, Yiyu Shi, Wujie Wen, Meiping Huang, Haiyun Yuan, and Jian Zhuang. 2019. Machine vision guided 3D medical image compression for efficient transmission and accurate segmentation in the clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 12687–12696.

[18] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. 2016. XNOR-Net: ImageNet classification using binary convolutional neural networks. In *Proceedings of the European Conference on Computer Vision.* 525–542.

[19] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015.* Lecture Notes in Computer Science, Vol. 9351. Springer, 234–241.

[20] Chuck Rosenberg. 2013. Improving photo search: A step across the semantic gap. *Google Research Blog* 12 (2013).

[21] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556.

[22] Korsuk Sirinukunwattana, Josien P. W. Pluim, Hao Chen, Xiaojuan Qi, Pheng-Ann Heng, Yun Bo Guo, Li Yang Wang, et al. 2017. Gland segmentation in colon histology images: The glas challenge contest. *Medical Image Analysis* 35 (2017), 489–502.

[23] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.

[24] Peisong Wang, Qinghao Hu, Yifan Zhang, Chunjie Zhang, Yang Liu, and Jian Cheng. 2018. Two-step quantization for low-bit neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 4376–4384.

[25] Tianchen Wang, Xiaowei Xu, Jinjun Xiong, Qianjun Jia, Haiyun Yuan, Meiping Huang, Jian Zhuang, and Yiyu Shi. 2020. ICA-UNet: ICA inspired statistical UNet for real-time 3D cardiac cine MRI segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention.* 447–457.

[26] Xiaowei Xu, Yukun Ding, Sharon Xiaobo Hu, Michael Niemier, Jason Cong, Yu Hu, and Yiyu Shi. 2018. Scaling for edge inference of deep neural networks. *Nature Electronics* 1, 4 (2018), 216–222.
[27] Xiaowei Xu, Qing Lu, Tianchen Wang, Yu Hu, Chen Zhuo, Jinglan Liu, and Yiyu Shi. 2018. Efficient hardware implementation of cellular neural networks with incremental quantization and early exit. *ACM Journal on Emerging Technologies in Computing Systems* 14, 4 (2018), 1–20.
[28] Xiaowei Xu, Qing Lu, Tianchen Wang, Jinglan Liu, Cheng Zhuo, Xiaobo Sharon Hu, and Yiyu Shi. 2017. Edge segmentation: Empowering mobile telemedicine with compressed cellular neural networks. In *Proceedings of the 2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD'17)*. IEEE, Los Alamitos, CA, 880–887.
[29] Xiaowei Xu, Qing Lu, Lin Yang, Sharon Hu, Danny Chen, Yu Hu, and Yiyu Shi. 2018. Quantization of fully convolutional networks for accurate biomedical image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8300–8308.
[30] Xiaowei Xu, Tianchen Wang, Yiyu Shi, Haiyun Yuan, Qianjun Jia, Meiping Huang, and Jian Zhuang. 2019. Whole heart and great vessel segmentation in congenital heart disease using deep neural networks and graph matching. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*. 477–485.
[31] Xiaowei Xu, Tianchen Wang, Jian Zhuang, Haiyun Yuan, Meiping Huang, Jianzheng Cen, Qianjun Jia, Yuhao Dong, and Yiyu Shi. 2020. ImageCHD: A 3D computed tomography image dataset for classification of congenital heart disease. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*. 77–87.
[32] Lin Yang, Yizhe Zhang, Jianxu Chen, Siyuan Zhang, and Danny Z. Chen. 2017. Suggestive annotation: A deep active learning framework for biomedical image segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*. 399–407.
[33] Li Zhang, Haixin Ai, Wen Chen, Zimo Yin, Huan Hu, Junfeng Zhu, Jian Zhao, Qi Zhao, and Hongsheng Liu. 2017. CarcinoPred-EL: Novel models for predicting the carcinogenicity of chemicals using molecular fingerprints and ensemble learning methods. *Scientific Reports* 7, 1 (2017), 1–14.
[34] Aojun Zhou, Anbang Yao, Yiwen Guo, Lin Xu, and Yurong Chen. 2017. Incremental network quantization: Towards lossless CNNs with low-precision weights. arXiv:1702.03044.
[35] Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. 2016. DoReFa-Net: Training low bitwidth convolutional neural networks with low bitwidth gradients. arXiv:1606.06160.
[36] Chenzhuo Zhu, Song Han, Huizi Mao, and William J. Dally. 2016. Trained ternary quantization. arXiv:1612.01064.

Received May 2020; revised January 2021; accepted February 2021