CURSOR-BASED ADAPTIVE QUANTIZATION FOR DEEP NEURAL NETWORK

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep neural network (DNN) has rapidly found many applications in different scenarios. However, its large computational cost and memory consumption are barriers to computing restrained applications. DNN model quantization is a widely used method to reduce the DNN storage and computation burden by decreasing the bit width. In this paper, we propose a novel cursor based adaptive quantization method using differentiable architecture search (DAS). The multiple bits' quantization mechanism is formulated as a DAS process with a continuous cursor that represents the possible quantization bit. The cursor-based DAS adaptively searches for the desired quantization bit for each layer. The DAS process can be solved via an alternative approximate optimization process, which is designed for mixed quantization scheme of a DNN model. We further devise a new loss function in the search process to simultaneously optimize accuracy and parameter size of the model. In the quantization step, based on a new strategy, the closest two integers to the cursor are adopted as the bits to quantize the DNN together to reduce the quantization noise and avoid the local convergence problem. Comprehensive experiments on benchmark datasets show that our cursor based adaptive quantization approach achieves the new state-of-the-art for multiple bits' quantization and can efficiently obtain lower size model with comparable or even better classification accuracy.

1 INTRODUCTION

Deep learning (DL) has achieved great successes in varied fields such as gaming, natural language processing, speech recognition, computer vision and so on. However, its huge computational burden and large memory consumption still intimidate many potential applications, especially for mobile devices and embedded systems.

A number of efforts have been devoted to compress the DL model size and accelerate its training and test speed. These efforts can be roughly categorized into four major classes: network pruning (Han et al. (2015); Anwar et al. (2015); Peng et al. (2019); Zhuang et al. (2018)), low rank approximation (Tai et al. (2015); Wang et al. (2018); Hayashi et al. (2019)), knowledge distillation (Hinton et al. (2015); Zagoruyko & Komodakis (2016)), and network quantization (Courbariaux & Bengio (2016); Lin et al. (2015); Wu et al. (2015); Polino et al. (2018); Zhang et al. (2018)). Among them, network quantization methods, jointly optimizing the whole network weights, activations or gradients with low bit (such as 8 bits or even 1 bit), show great potential in compressing model size and accelerating inference time. In addition, quantization based approaches are preferable for mobile devices and embedded systems since these devices are gradually equipped by specifically designed low bit computing hardware. Furthermore, using quantized representations, some operations of the neural network can be implemented by bitwise operations, leading to much less energy consumption.

Although existing quantization based approaches, which mainly use fixed bit to represent the whole DNN model, yields encouraging compression ratio while keeping the model's performance, we argue that simply using only a fixed bit for quantization is not the optimal choice for the tradeoff between a model size and its performance. For example, to run a model on chips with strict memory limitations, 1 bit or 2 bits' quantization suffers from severe accuracy loss (Rastegari et al. (2016)) while 16 bits' or 8 bits' quantization cannot significantly reduce the model size.

To address the above problem, we propose a cursor based adaptive quantization method to derive multiple bits in different layers for DNN model compression, i.e., we search for the best configuration of different bit quantization for different layers in a neural network model. Distinctive from most algorithms aforementioned, our approach is motivated by recent neural architecture search (NAS) that aims to find better performance neural architecture with less calculations or less size automatically. The key in our algorithm is using a continuous cursor that represents the bit quantization scheme for each layer. For different layers, many cursors will be adaptively searched at the same time during the NAS process. Since the cursor itself is continuous and the whole search procedure can be considered as a differentiable architecture search (DAS) process, which can be effectively solved based on an alternative optimization strategy. A novel cost function that considers the model compression and prediction accuracy is also proposed in the DAS process. After the cursor is searched, a quantization process is applied to compress the model size. To reduce the possible quantization noise and local convergence problem, we make use of the closest two integer bits to the cursor to quantize the weights for each layer in the DNN model. We validate our proposed method with image classification tasks on CIFAR10 and ImageNet. Comprehensive experiments on some backbone DNN models show that the proposed cursor based quantization method achieves remarkably better performance of compression ratio with ignorable accuracy drop or even better accuracy.

In summary, the contributions of this work are four-fold:

- We cast the adaptive quantization of neural network as a problem of neural architecture search. A continuous cursor is proposed to represent the possible quantization bit, leading to a more efficient search space.
- A novel regularization function is proposed to optimize model compression in the search process. Thus the search for the cursor position and weights can be efficiently solved in an alternative optimization manner.
- Two nearest neighbor integers to the cursor are adopted with a carefully designed strategy to implement the quantization of the network to reduce the quantization noise and avoid possible local convergence.
- We comprehensively evaluate the proposed adaptive quantization method on some benchmark datasets and achieve new state-of-the-art performance for multiple bit quantization of neural network.

2 Related work

Quantization plays an important part in DNN model compression. Recently, a lot of new quantization approaches have been proposed, enabling the quantized compressed model to compete with their full precision counterparts. In (Gong et al. (2014)), the authors applied vector quantization to compress neural networks by clustering the weights into groups and utilizing the centers of these groups to represent the weights in inference process. Later, clustering method is also applied for their codebook representation(Han et al. (2015)), and then the network is retrained to get better quantized centroids. In (Zhang et al. (2018)), the authors jointly trained a DNN and its associated quantizes to reduce the noticeable predication accuracy gap between the quantized model and its full precision one. A direct differentiable quantization method was introduced in (Louizos et al. (2018)) with promising test accuracy. A new activation quantization method that takes an activation clipping parameter was proposed in (Choi et al. (2018)) to ensure the suitable quantization scale.

Some efforts have also been taken on quantization of the neural network with multiple bits for different layers. In (Lin et al. (2015)), signal-to-quantization-noise ratio (SQNR) is applied on layer weight to evaluate the effects of quantization error. Based on SQNR, different bits were used for quantization of each layer, yielding about 20% model size reduction without accuracy loss in their tests. The authors (Wang et al. (2018)) presented an automated mixed precision quantization scheme by applying reinforcement learning (RL) technique to obtain better latency for different hardware such as edge and cloud, they claimed that their actor-critic model produced efficient actions that result in better latency and less energy consumption with negligible loss of accuracy. In (Zhou et al. (2017)), the authors theoretically studied the relations between quantization choice of different layer and the whole DNN performance. A novel model that describes the effects of parameter quantization error on the whole prediction accuracy was also designed.

In the past few years, a new trend has been witnessed for network design, i.e., neural architecture search (NAS). RL based approaches are first utilized to generate network with high accuracy (Zoph & Le (2016)), and they also build a strong basis for the following recent works such as (Gao et al. (2019); Guo et al. (2018)). Then, evolution based approach(Liang et al. (2018)) is further applied to obtain the possible optimal solution in the large search space. Both of these two category approaches tend to yield large amount of computational burden because NAS is treated as a blackbox optimization problem in a discrete domain, yielding a large number of architecture evaluations, and thus run very slow even on the most advanced GPU machine. To alleviate this problem, in 2018, the authors (Liu et al. (2018)) proposed a differentiable approach to accelerate the search of a good neural network by relaxation of the possible operation on the cell level structure. Wu et al. recently proposed a new approach to find the mixed bits for different layers by applying differentiable NAS (DNAS) method based on a model of super net (Wu et al. (2018)), which is a kind of directed acyclic graph. They considered the quantization as a problem of sampling on a stochastic super net. And a Gumbel softmax function is applied to relax the sampling process differentiable.

We cast the multiple bits' quantization for DNN as a cursor based adaptive architecture search problem, and it is different from the traditional direct quantization works and the learning based mixed bits' quantization approaches mentioned above. Moreover, it is also distinctive from DARTs and DNAS in the methodology itself. Our designed cursor based search is mainly motivated by NAS, but it does NOT need extra relaxation as in DARTs and DNAS. More details about the distinctions from DARTs and DNAS will be emphasized in the next Section.

3 Cursor-based adaptive quantization

In this section, we first briefly present some basic concepts in NAS, followed by the introduction of search space for quantization problem if we reflect on quantization with different bits from the angle of NAS. Differentiable cursor search method for adaptive quantization is proposed afterwards. Finally, quantized network training with two closest integers is presented.

3.1 Neural architecture search

It is well known that DNN model needs much time to design its structure and fine tune all the related key parameters such as learning rate, momentum, weigh decay and so on. As such, there is a strong need for neural network architecture engineering. Neural architecture search (NAS) recently emerged as a popular methodology to overcome the above problem. It automatically designs the optimal architecture of a neural network by considering all possible factors such as number of layers, width of each layer, different operators in each layer and so on. Two key concepts are directly related to a NAS process, i.e., search space and search strategy. All the possible combinations of the major factors that influence the network structure constitute the search space, which may affect the possible performance limitation of a DNN. Generally speaking, the search space of a DNN is very large, leading to a huge computational task even for one typical network such as Resnet20. As such, the previous NAS works instead design normal and reduction cell (Pham et al. (2018)) first, where the NAS is applied for the different possible operation types to find the best one. And then this kind of motif is repeated to build the final neural network. Another definition is about search strategy, that is, how to transverse in such a large search space. With each searched network structure, the performance of it will be evaluated. A typical search method is random search, however, its efficiency is not ideal. Therefore, most recent works (Cai et al. (2018); Liu et al. (2018))have been proposed along this big direction to improve the search efficiency as much as possible.

3.2 Search space for quantization problem

Quantization has also been a very hot research topic in the past few years. Rounding function, vector quantization or stochastic function are typically applied to implement quantization to compact the model size while maintaining equivalent performance or acceptable loss. Some other approaches also use stochastic or probabilistic methods to quantize the neural network. Most previous methods simply apply one kind of bit quantization to the whole network due to the simplicity of implementation. A few recent works begin to utilize different bit quantization scheme to further improve the compression ratio and prediction accuracy. If we consider quantization choice as a part of the neural architecture, we can estimate its corresponding search space. Let us take Resent20 as an example and if we decide to quantize the neural network with the possible bit width of 1, 2, 4, 8, 16, 32, then all the possible quantization choices for ResNet20 will be 6^{20}. In the context of NAS, this is a very large number for the search space. Hence, evaluation of so many designs one by one seems infeasible right now. How to devise an efficient search method remains very challenging in fact.

3.3 Differentiable cursor search for adaptive quantization

The discrete search space of the above quantization scheme is so large. If we further consider the possible bit for each layer as a continuous cursor in the range of [0, 32], the cursors then become significant parts of the architecture for a neural network model, and the search of cursors is adaptive. If we assume a DNN has N layers, each layer has a different value of cursor, $c_1, c_2, ..., c_N$, together with their weights of W_c, our goal evolves to how to find a good combination of $c_1, c_2, ..., c_N$in terms of prediction accuracy and compression rate. As such, for the whole neural network it can be described as an optimization problem that minimizes the loss on the validation data after training through the minimization of the loss on the training data as follows:

$$\min E(\mathrm{x}^{'}, y^{'})_{\tilde{D}_v}(Loss(C, W_C)) \quad \text{s.t. } \mathrm{W}_{c^*} = argmin\mathrm{E}(x^{'},\mathrm{y}^{'})_{\tilde{D}_T}(Loss(C, W_C)) \tag{1}$$

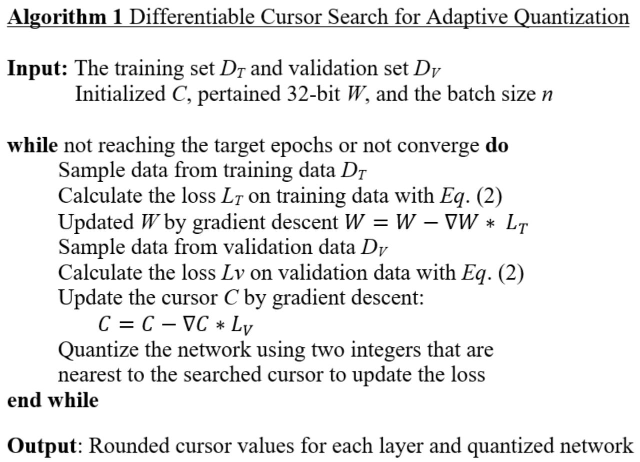
where C represents the cursor vector,W_{c^*} is the best weights of C and obtains the best training accuracy, $Loss(C, Wc)$ is the loss function based on the cursors and the weights with the condition of C, D_T and D_V represents the training and validation dataset respectively, (x, y) and $(x^{'}$,y$^{'})$ means data from the training and validation dataset. To consider both the prediction accuracy and model size, we design the loss function as a combination of cross entropy and parameter compression as follows:

$$\mathrm{Loss(C,W}_c) = CrossEntropy(C, W_c) + \lambda \mathrm{Loss}_Q \tag{2}$$

where $CrossEntropy(C, W_c)$ is the cross entropy function widely used in the field of machine learning, encoding the prediction accuracy of the model. The reason why we add a regularization item to the loss function is because the regularization can prevent overfitting to some extent. In addition, it might also speed up the convergence process. Concerning the loss related to $Loss_Q$, we focus on the compression of the weight size after quantization and before quantization. So we conceive it in the form of Eq.(3), and λ is a regularization coefficient that controls the trade-off of accuracy and compression.

The above process is a bi-level optimization problem, which may require to deduce higher order derivatives and is hard to obtain an exact solution. An approximated iterative solution can be applied instead, so we alternatively take the optimization strategy in weight and cursor space to update C based on the validation losses from D_V and update W based on the training losses from D_T. By solving this bi-level optimization problem using alternative approximation approach, the cursors can be efficiently searched by gradient based optimization approach such as Adam. Our later experimental results also show that the alternative optimization method may yield a good solution with high compression ratio and accuracy. Compared to the original discrete search space, this search method is more efficient because the design of continuous cursor and the direct gradient based optimization approach. The whole differentiable cursor search for adaptive quantization based on the alternative optimization of W and C is illustrated in Figure 1. The last step in the loop about quantization using the nearest two integers to the cursor will be introduced in details in the subsequent section. The outputs of the whole algorithm are the rounded cursor values for each layer together with its quantized model.

It should be emphasized that our proposed cursor based differentiable search is different from DARTs (Liu et al. (2018)) in the following three aspects. First, DARTs method considers the possible operation in each layer as a mixture of primitive operations. We directly make use of cursor to represent the quantization bit for each layer, no mixture operation exists in the whole search algorithm. Second, in DARTs, each primitive operation is assigned with a probability through a softmax function. Cursor based search is optimized directly without probability. Third, DARTs approach concentrates on the cell structure, but we apply the DAS directly on the whole network. Compared to DNAS (Wu et al. (2018)), our approach is also distinctive. For DNAS, the authors build a



```
Algorithm 1 Differentiable Cursor Search for Adaptive Quantization

Input: The training set $D_T$ and validation set $D_V$
       Initialized $C$, pertained 32-bit $W$, and the batch size $n$

while not reaching the target epochs or not converge do
    Sample data from training data $D_T$
    Calculate the loss $L_T$ on training data with $Eq.$ (2)
    Updated $W$ by gradient descent $W = W - \nabla W * L_T$
    Sample data from validation data $D_V$
    Calculate the loss $Lv$ on validation data with $Eq.$ (2)
    Update the cursor $C$ by gradient descent:
        $C = C - \nabla C * L_V$
    Quantize the network using two integers that are
    nearest to the searched cursor to update the loss
end while

Output: Rounded cursor values for each layer and quantized network
```

Figure 1: Overview of the proposed algorithm

stochastic super net first to describe all the possible quantization choices, then a sampling step with a Gumbel softmax function that enables the discrete distribution to be continuous and differentiable is applied in each layer of the super net. Our cursor based differentiable search has no super net or sampling process in the pipeline. Hence, the subsequent solutions to the optimization problem is also completely different. In short, the proposed method requires no relaxation anymore as in both DARTs and DNAS approach.

3.4 Quantized network training

Aiming for DNN quantization, we should apply the cursor that represents the bit to quantize the weight layers. Unfortunately, the cursor obtained during the search is a fractional number, which cannot be directly used for quantization. One choice is to round the cursor to its neighbor integers at a distance, but it may cause rather large quantization error if we choose the rather distant bits. On the other hand, if we directly round the cursor to its nearest integer, it cannot efficiently represent the variation of cursor. For example, if cursor1 and cursor2 for different epochs in the same layer is 2.6 and 2.8 respectively, they will be rounded to the same integer 3, yielding no change in the weight size for this layer when implementing quantization. In addition, in the whole search process, such one integer choice may result in local convergence because the iteration process of one integer quantization may get stuck in a local minimum region for the cursor search. To alleviate the above two problems, we propose instead to make use of the nearest lower and upper integer bound at the same time in the search training process. Compared to directly using the nearest one neighbor to quantize, the lower and upper integer bounds can approximate the cursor real position through a weighted summary of these two bounds, leading to a better representation of the continuous cursor. Moreover, they may produce more variations in the loss function that describes the compression effects, yielding effective gradient changes to update the cursors more efficiently. Subsequent experiments also demonstrate that this design can obtain better quantization performance compared to simply applying rounding function on the searched cursor. As such, the loss function in the cursor based differentiable search is designed as follows:

$$Loss_Q = (\sum_{layer=1}^{n} \frac{parameter\ size\ after\ quantization\ in\ one\ layer}{parameter\ size\ before\ quantization\ in\ one\ layer})^{\gamma} \tag{3}$$

where γ is a coefficient that determines the importance of the whole regularizer, and we summarize all the layers' size change to describe the whole compression effects. When implementing the quantization for each layer, we utilize the DoReFa-Net (Zhou et al. (2016)) quantization for each layer based on the obtained two integers:

$$w_k = 2Q_k(\frac{tanh(w)}{2max(|tanh(w)|)} + 0.5) - 1 \tag{4}$$

where w_k represents the full precision weight of a model and $Q_k(*)$ is the k-bit quantization function that transforms a continuous value $x \in [0, 1]$ to a k-bit output $y \in [0, 1]$ as follows:

$$y = \frac{1}{2^k - 1} round(2^k - 1) * x \tag{5}$$

where *round* function is the typical rounding operation used in quantization. In other words, in the process of quantization, after searching the possible quantization bit of c in each layer, its corresponding two nearest neighbor integers a_1 and a_2 will be applied to Eq.(5) and Eq.(4) to quantize the network to obtain the size change in Eq.(3) . In the neural network forward process, based on the cursor's lower and upper bound integer a_1 and a_2 , we can define two coefficients d_1 and d_2 as below:

$$d_1 = 1 - (c_i - a_1); \quad d_2 = 1 - (a_2 - c_i) \tag{6}$$

where c_i represents the cursor searched by our adaptive method in the i^{th} layer of the DNN. Then, the output in the forward process can be described with the following equation:

$$f = d_1 * (Conv(X, W_1) + d_2 * Conv(X * W_2)) \tag{7}$$

where W_1 and W_2 are the weights after quantization using a_1 and a_2, $Conv$ is the convolution operation. While for the computation of size loss in Eq.(3), we have similar one as below:

$$layer\_size = layer\_base\_size * (d_1 * a_1 + d_2 * a_2) \tag{8}$$

where the layer_base_size is defined as the size of a layer when it is represented by 1 bit for parameter size calculation, and the total loss can be obtained by a summary of the loss in each layer. In the backward process, we apply the gradient on its full precision counterpart. After the approximate alternative optimization approach converges or reaches the target epoch number, the final quantization bit in each layer can be obtained by applying rounding operation on each cursor for inference. And the final quantized model should also be retrained based on the quantization bit. It should be noted that there are still some quantization errors in the whole procedure, this is also why we may need to fine tune the weights of the quantized DNN after the search process.

4 Experiments

Our codes are implemented with PyTorch and we make use of NVidia TiTanX to carry out all the tests. Since it has been mentioned that with 16 bits, the neural network's performance almost has no loss (Elthakeb et al. (2018)), we concentrate on possible application of multiple lower bits in all our experiments, that is, we only apply 1, 2, 3, 4, 5, 6, 7, 8 bit(s) as the possible quantization bit width for each layer to test our scheme's feasibility. As a matter of fact, there is a new trend that investigates the possible quantization with bit that is not power of 2 (Elthakeb et al. (2018); Park et al. (2018b;a); Wang et al. (2018)). In addition, some hardware such as FPGA also gradually supports efficient quantization using such bits (Wang et al. (2018); Wei et al. (2019)). Currently, we only apply quantization on the weights and use full precision activations. In addition, we also follow the traditions in the domain of DNN quantization to avoid the quantization of the first and last layer in a model.

In all the experiments, we take ResNet18, ResNet20 (He et al. (2015)) or MobileNetV2(Sandler et al. (2018)) as the backbone model. It should be noted that these models should be pretrained to obtain the floating point models first. For the initialization of the cursor in each layer, all of them are set with 4 bits for the convenience of iteration. When the cursors are obtained by our method, the model may be further trained from scratch to get its final accuracy, which is a practical tradition in the fields of NAS and quantization.

As for the parameter λ in Eq.(2) and γ in the loss of quantization in Eq.(3), a rather optimal set of them is chosen as (0.25, 0.3) after trials. We also dedicate to study the influence of λ in the experiments to show that in most cases, the cursor based adaptive quantization scheme is robust to its change if λ is at a larger interval of $\lambda \geq 0.1$. Concerning the learning rate schedule of weight and cursor, we apply cosine annealing method to adjust them. The minimum learning rate for them is 0.001 and 0.0001 respectively.

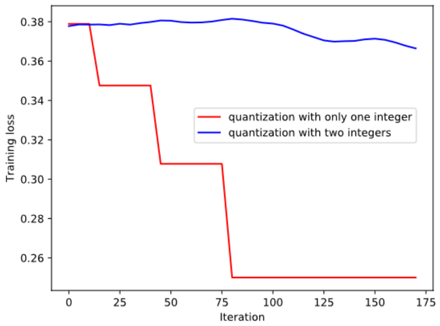


Figure 2: Loss change for different quantization scheme

4.1 Training loss comparison

To show the validity of quantization approach using two integer bounds nearest to the cursor, we first implement the search process by comparing it to using only one nearest integer of the cursor. We analyze their losses to show the great distinction in the training process.

Here we apply ResNet20 on CIFAR10 dataset to demonstrate the optimization process. For illustrative purpose, we only draw the loss change in part of the iteration process in one epoch. As shown in Figure 2, the red curve represents the training loss using one nearest integer to implement quantization, while the blue one denotes the loss obtained by using two neighbor integers nearest to the curser searched by the proposed scheme. The major differences in these two tests lie at the quantization choices. In fact, we also tried some other parameters and random initialization for one integer quantization scheme, and similar curves can be found. Obviously, the blue one looks more smooth and natural for a convergence process. The red loss may lead to a strong possibility that the cursors are stuck in a local minimum region instead. In fact, the cursor values obtained by the one neighbor scheme tend to be 1 bit for all layers after some epochs' iteration. The reason why the one integer quantization scheme fails may be because, in most cases, the weights in one layer span a rather small range, one lower integer quantization may lead to the same quantization results on the weights in the training process. Such same quantization results further yield almost no change in the backward gradient process, which is not beneficial for the optimal cursor search. The designed two integers' quantization process, on the other hand, can map the cursor to two different integer values, leading to efficient change in the loss even for the weights in rather a small value range.

4.2 Search process analysis

To get some insights of our adaptive cursor search algorithm, we investigate its iteration process in this subsection. For illustration only, we take MobileNetV2 on CIFAR10 as an example. Its search process is depicted in Figure 3 with the quantization bits ignored due to space limitation. Here the abscissa and vertical coordinate respectively represents the compression ratio and prediction accuracy. It should be noted that here our proposed algorithm runs 10 epochs only to clearly show the variation of performance. In addition, because of the cosine annealing scheduler, such an iteration process may also be representative. From Figure 3, we see that for the proposed adaptive cursor search scheme, it first begins at the lower left region (lower accuracy and compression) and then gradually assembles to the upper right region (higher accuracy and compression). Meanwhile, there is some small vibrations in the whole process, for example, from epoch 8 to epoch 9, there is some increase in accuracy as well as compression ratio, but from epoch 9 to epoch 10, there is a slight reduction in both measures. It can be noticed that the search process is rather stable and gathers to the final upper right region with better accuracy and compression ratio. We also observed similar pattern for ResNet20 on CIFAR10, but we ignored the picture of it because of space limitation. The reason why the search process of our method can reach to a region with high prediction accuracy and compression ratio may be due to the alternative optimization approach to solve this bi-level problem with two goals. In addition, the regularization item may also play a positive role in this process.

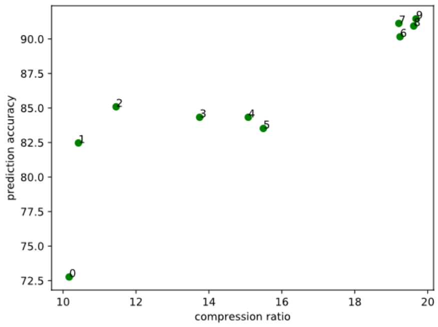


Figure 3: Adaptive search process for MobileNetV2 on CIFAR10, the number beside the points represent the epoch

Table 1: Performance of ResNet20 on CIFAR10 with different λ

λ	0.9	0.7	0.5	0.25	0.1	0.05	0.01
Accuracy(%)	90.10	90.14	90.18	91.79	91.03	91.16	91.58
Compression ratio	30.37	30.00	30.09	28.33	28.43	19.46	13.95

4.3 Impact of regularization coefficient

The regularization coefficient λ in Eq.(2) controls the balance between the model precision and size. In this part, we carry out some experiments to analyze the influence of it on the whole performance. We choose λ = 0.9, 0.7, 0.5, 0.25, 0.1, 0.05, 0.01, and we test its effects on the quantized model. For the purpose of illustration, we test ResNet20 on CIFAR10. To directly show the effects of our cursor based differentiable search, we do NOT implement finetune step for all these results after finishing the cursor search. The results of the quantized ResNet20 on CIFAR10 is demonstrated in Table 1, and all the results are obtained by implementing the search with 200 epochs.

From Table 1, we can observe that for $\lambda >= 0.1$, the whole performance of the proposed quantization method is rather steady, that is, the accuracy and compression ratio of the quantized model maintain at a concentrated region with the accuracy about 90% while the compression ratio about 29.00. When $\lambda < 0.1$, the cursor based adaptive quantization approach may still have a good performance of prediction but gradually loses its effects on model compression. This can be explained that when the regularization becomes gradually weak, it does NOT exert its compression effects so well as when the coefficient is large. This further validates the effectiveness of the regularization function proposed in this paper.

4.4 CIFAR10 results

We demonstrate our cursor based adaptive quantization algorithm on CIFAR10 benchmark dataset with ResNet20 and MobileNetV2. The obtained mixed bits in ResNet20 for all the layers except the first and last one is shown with a tuple as [1,1,1,1,1,1,1,3,5,1,1,1,1,1,8,1,1,1]. It is interesting to see that most of the bits in the final quantization scheme is 1, contributing much to the impressive compression ratio (28.3). This also shows that there is a lot of redundancy among the neural network layers. In addition, the compressed model achieves a prediction accuracy of 92.27% on the test set, which is a little bit better than the original full precision model (92.06%) based on our own implementation. It should be pointed out that we do NOT spend much time fine tuning the original model to obtain the best performance in the literature. Our focus is on the quantization scheme proposed in this work, and the target of our tests is to validate that this cursor based adaptive quantization scheme can efficiently reduce the model size with comparable or better prediction accuracy.

Table 2: Performance comparison with other works on CIFAR10

	Accuracy(%)	Compression Ratio
Ours	92.27	28.3
DNAS(most efficient)	92.00	16.6
DNAS(most accurate)	92.72	11.6
LQE(2 bit)	91.80	16.0
TTQ(2 bit)	91.13	16.0
PACT(2 bit)	89.70	16.0
Baseline(32 bit)	92.06	1.0

We further compare the accuracy and compression ratio of the proposed approach to some related or similar works such as DNAS (Wu et al. (2018)), TTQ (Zhu et al. (2016)), PACT (Choi et al. (2018)) and LQE (Zhang et al. (2018)) with Resnet20 on CIFAR-10, and the details of accuracy and compression ratio are shown in Table 2. It can be noticed that, compared to the other related works, our method achieves much better compression ratio while achieving comparable or better classification accuracy on CIFAR10 dataset. The reason why the proposed approach is better than the quantization methods such as LQE, TTQ and PACT may be due to the adaptive cursor based search mechanism. By considering both the model accuracy and compression ratio, the cursor based approach can effectively search different quantization bit for each layer as a whole, leading to better compression ratio with better accuracy. Compared to DNAS, the reason for our better performance in terms of CR is partially due to that the two closest integers' quantization scheme produces less quantization error in each layer. In addition, it may be also because of our multiple lower bits' design in the search process.

We also apply the proposed approach to MobilenetV2, which is a typical DL model for mobile devices and embedded systems. To save space, we show the searched bit results for different layers using a tuple as below: [6, 6, 2, 1, 4, 5, 4, 6, 3, 5, 4, 6, 5, 3, 7, 7, 5, 6, 5, 7, 4, 6, 4, 6, 5, 3, 2, 5, 3, 2, 5, 6, 5, 5, 2, 2, 4, 3, 3, 6, 5, 2, 5, 2, 4, 2, 1, 4, 4, 1, 2, 2]. For MobileNetV2 on CIFAR10, the bit distribution is more diversified compared to that of ResNet20, this may be attributed to the inherent structure difference of them. The accuracy and compression ratio of our algorithm on MobilenetV2 is 93.28% and an impressive compression ratio of 12.4, while the original 32 bit model has an accuracy of 92.39%.

4.5 ImageNet results

In this subsection, we apply ResNet18 and MobileNetV2 to ImageNet dataset, which is a much larger dataset compared to CIFAR10. Here, as in (Han et al. (2015); Wang et al. (2018)), we present two sets of our results, i.e., the most efficient result and most accurate one to compare more conveniently. We mainly demonstrate the whole performance of the proposed scheme, ignoring the searched bit distributions in this subsection due to space limitation.

We record the performance of the proposed method with ResNet18 on ImageNet in Table 3 as well as some comparisons to LQE (Zhang et al. (2018)), TTQ (Zhu et al. (2016)), PACT (Choi et al. (2018)) methods. From Table 3, it can be noticed that, compared to the original 32 bit model, the most accurate result of our algorithm achieves a promising compression rate of 13.9 with a slight accuracy drop of 0.15%, and for the most efficient one, our algorithm achieves an accuracy of 68.80% and an impressive compression ratio of 18.1. The most accurate result of our algorithm shows much better accuracy over LQE, TTQ and PACT methods although the compression ratio is a little bit smaller. As for the most efficient one, both the accuracy and compression ratio are better than those of LQE, TTQ and PACT, validating the effectiveness of the proposed scheme.

The performance of MobileNetV2 on ImageNet is illustrated in Table 4 together with comparisons to some related works such as HAQ (Wang et al. (2018)) and deep compression (Han et al. (2015)). In Table 4, we notice that, for the most accurate result, the quantized MobileNetV2 model using our approach shows slight accuracy loss (71.65% vs 72.19% of the original 32 bit model) while achieves an encouraging compression ratio of 9.1. It may also be observed that the accuracy of our most accurate one is a little bit higher than the corresponding most accurate results of HAQ and

Table 3: Performance comparison with other works using ResNet18 on ImageNet

	Accuracy(%)	Compression Ratio
Baseline(32 bit)	69.75	1.0
Ours(most accurate)	69.60	13.9
Ours(most efficient)	68.80	18.1
LQE(2 bit)	68.00	16.0
TTQ(2 bit)	66.60	16.0
PACT(2 bit)	64.40	16.0

Table 4: Performance comparison with other works using MobileNetV2 on ImageNet

	Accuracy(%)	Compression Ratio
Baseline(32 bit)	72.19	1.0
Ours(most accurate)	71.65	9.1
Ours(most efficient)	70.59	14.3
HAQ(most accurate)	71.47	7.5
HAQ(most efficient)	66.75	13.9
Deep compression(most accurate)	71.24	8.0
Deep compression(most efficient)	58.07	16.0

deep compression together with a better compression ratio over those of HAQ and deep compression. While for the most efficient one, our algorithm shows a compression ratio of 14.3, it is better than that of HAQ, but smaller than that of deep compression. However, the accuracy of it demonstrates a drastically better accuracy of 70.59% compared to the corresponding 66.75% of HAQ and 58.07% of deep compression, also illustrating the validity of our algorithm.

5 Conclusions

In this paper, we have proposed a novel cursor based DAS algorithm for obtaining the mixed precision DNN model. Different from most of the traditional approaches, which choose quantization configuration using heuristics or learning based rules, we adaptively choose the quantization bit for each layer in the DNN model from the perspective of NAS. A cursor based search algorithm with alternative manner is applied for efficient optimization. The nearest two neighbor integers to the cursor are used to implement the quantization in the training process to reduce the quantization noise and avoid local convergence. The result of our algorithm is the adaptive bit width choice for different layers as a whole. Extensive experiments with some typical models demonstrate that the proposed approach provides dramatic compression capability with accuracy on par with or better than the state-of-the-art of methods on benchmark datasets. In the near future, we may apply the proposed approach to activation layers to further improve the compression ability of it.

References

Sajid Anwar, Kyuyeon Hwang, and Wonyong Sung. Structured pruning of deep convolutional neural networks. *CoRR*, abs/1512.08571, 2015. URL http://arxiv.org/abs/1512.08571.

Han Cai, Jiacheng Yang, Weinan Zhang, Song Han, and Yong Yu. Path-level network transformation for efficient architecture search. *CoRR*, abs/1806.02639, 2018. URL http://arxiv.org/abs/1806.02639.

Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. PACT: parameterized clipping activation for quantized neural networks. *CoRR*, abs/1805.06085, 2018. URL http://arxiv.org/abs/1805.06085.

Matthieu Courbariaux and Yoshua Bengio. Binarynet: Training deep neural networks with weights and activations constrained to +1 or -1. *CoRR*, abs/1602.02830, 2016. URL http://arxiv.org/abs/1602.02830.

Ahmed T. Elthakeb, Prannoy Pilligundla, Amir Yazdanbakhsh, Sean Kinzer, and Hadi Esmaeilzadeh. Releq: A reinforcement learning approach for deep quantization of neural networks. *CoRR*, abs/1811.01704, 2018. URL http://arxiv.org/abs/1811.01704.

Yang Gao, Hong Yang, Peng Zhang, Chuan Zhou, and Yue Hu. Graphnas: Graph neural architecture search with reinforcement learning. *CoRR*, abs/1904.09981, 2019. URL http://arxiv.org/abs/1904.09981.

Yunchao Gong, Liu Liu, Ming Yang, and Lubomir D. Bourdev. Compressing deep convolutional networks using vector quantization. *CoRR*, abs/1412.6115, 2014. URL http://arxiv.org/abs/1412.6115.

Minghao Guo, Zhao Zhong, Wei Wu, Dahua Lin, and Junjie Yan. IRLAS: inverse reinforcement learning for architecture search. *CoRR*, abs/1812.05285, 2018. URL http://arxiv.org/abs/1812.05285.

Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. *CoRR*, abs/1510.00149, 2015.

Kohei Hayashi, Taiki Yamaguchi, Yohei Sugawara, and Shin ichi Maeda. Einconv: Exploring unexplored tensor decompositions for convolutional neural networks. *ArXiv*, abs/1908.04471, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. *arXiv e-prints*, art. arXiv:1503.02531, Mar 2015.

Jason Zhi Liang, Elliot Meyerson, and Risto Miikkulainen. Evolutionary architecture search for deep multitask networks. *CoRR*, abs/1803.03745, 2018. URL http://arxiv.org/abs/1803.03745.

Darryl Dexu Lin, Sachin S. Talathi, and V. Sreekanth Annapureddy. Fixed point quantization of deep convolutional networks. *CoRR*, abs/1511.06393, 2015. URL http://arxiv.org/abs/1511.06393.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: differentiable architecture search. *CoRR*, abs/1806.09055, 2018. URL http://arxiv.org/abs/1806.09055.

Christos Louizos, Matthias Reisser, Tijmen Blankevoort, Efstratios Gavves, and Max Welling. Relaxed quantization for discretized neural networks. *CoRR*, abs/1810.01875, 2018. URL http://arxiv.org/abs/1810.01875.

Eunhyeok Park, Dongyoung Kim, Sungjoo Yoo, and Peter Vajda. Precision highway for ultra low-precision quantization. *CoRR*, abs/1812.09818, 2018a. URL http://arxiv.org/abs/1812.09818.

Eunhyeok Park, Sungjoo Yoo, and Peter Vajda. Value-aware quantization for training and inference of neural networks. *CoRR*, abs/1804.07802, 2018b. URL http://arxiv.org/abs/1804.07802.

Hanyu Peng, Jiaxiang Wu, Shifeng Chen, and Junzhou Huang. Collaborative channel pruning for deep networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5113–5122, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/peng19c.html.

Hieu Pham, Melody Y. Guan, Barret Zoph, Quoc V. Le, and Jeff Dean. Efficient neural architecture search via parameter sharing. *CoRR*, abs/1802.03268, 2018. URL http://arxiv.org/abs/1802.03268.

Antonio Polino, Razvan Pascanu, and Dan Alistarh. Model compression via distillation and quantization. *CoRR*, abs/1802.05668, 2018. URL http://arxiv.org/abs/1802.05668.

Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. *CoRR*, abs/1603.05279, 2016. URL http://arxiv.org/abs/1603.05279.

Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. *CoRR*, abs/1801.04381, 2018. URL http://arxiv.org/abs/1801.04381.

Cheng Tai, Tong Xiao, Yi Zhang, Xiaogang Wang, and Weinan E. Convolutional neural networks with low-rank regularization. *arXiv e-prints*, art. arXiv:1511.06067, Nov 2015.

Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. HAQ: hardware-aware automated quantization. *CoRR*, abs/1811.08886, 2018. URL http://arxiv.org/abs/1811.08886.

Xin Wei, Wenchao Liu, Lei Chen, Long Ma, He Chen, and Yin Zhuang. Fpga-based hybrid-type implementation of quantized neural networks for remote sensing applications. *Sensors*, 19(4), 2019. ISSN 1424-8220. doi: 10.3390/s19040924. URL https://www.mdpi.com/1424-8220/19/4/924.

Bichen Wu, Yanghan Wang, Peizhao Zhang, Yuandong Tian, Peter Vajda, and Kurt Keutzer. Mixed precision quantization of convnets via differentiable neural architecture search. *CoRR*, abs/1812.00090, 2018. URL http://arxiv.org/abs/1812.00090.

Jiaxiang Wu, Cong Leng, Yuhang Wang, Qinghao Hu, and Jian Cheng. Quantized convolutional neural networks for mobile devices. *CoRR*, abs/1512.06473, 2015. URL http://arxiv.org/abs/1512.06473.

Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *CoRR*, abs/1612.03928, 2016. URL http://arxiv.org/abs/1612.03928.

Dongqing Zhang, Jiaolong Yang, Dongqiangzi Ye, and Gang Hua. Lq-nets: Learned quantization for highly accurate and compact deep neural networks. *CoRR*, abs/1807.10029, 2018. URL http://arxiv.org/abs/1807.10029.

Shuchang Zhou, Zekun Ni, Xinyu Zhou, He Wen, Yuxin Wu, and Yuheng Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *CoRR*, abs/1606.06160, 2016. URL http://arxiv.org/abs/1606.06160.

Yiren Zhou, Seyed-Mohsen Moosavi-Dezfooli, Ngai-Man Cheung, and Pascal Frossard. Adaptive quantization for deep neural network. *CoRR*, abs/1712.01048, 2017. URL http://arxiv.org/abs/1712.01048.

Chenzhuo Zhu, Song Han, Huizi Mao, and William J. Dally. Trained ternary quantization. *CoRR*, abs/1612.01064, 2016. URL http://arxiv.org/abs/1612.01064.

Zhuangwei Zhuang, Mingkui Tan, Bohan Zhuang, Jing Liu, Yong Guo, Qingyao Wu, Junzhou Huang, and Jin-Hui Zhu. Discrimination-aware channel pruning for deep neural networks. *CoRR*, abs/1810.11809, 2018. URL http://arxiv.org/abs/1810.11809.

Barret Zoph and Quoc V. Le. Neural architecture search with reinforcement learning. *CoRR*, abs/1611.01578, 2016. URL http://arxiv.org/abs/1611.01578.