ABSTRACT
Deep quantization of neural networks (below eight bits) offers significant promise in reducing their compute and storage cost. Albeit alluring, without special techniques for training and optimization, deep quantization results in significant accuracy loss. To further mitigate this loss, we propose a novel sinusoidal regularization, called SinReQ, for deep quantized training. SinReQ adds a periodic term to the original objective function of the underlying training algorithm. SinReQ exploits the periodicity, differentiability, and the desired convexity profile in sinusoidal functions to automatically propel weights towards values that are inherently closer to quantization levels. Since, this technique does not require invasive changes to the training procedure, SinReQ can harmoniously enhance quantized training algorithms. SinReQ offers generality and flexibility as it is not limited to a certain bitwidth or a uniform assignment of bitwidths across layers. We carry out experimentation using the CIFAR-10, ResNet-20, SVHN, and VGG-11 DNNs with three to five bits for quantization and show the versatility of SinReQ in enhancing multiple quantized training algorithms, DoReFa [1] and WRPN [2]. Averaging across all the bit configurations shows that SinReQ closes the accuracy gap between these two techniques and the full-precision runs by 35.7% and 37.1%, respectively. That is, on average, closing the gap between the quantized network and a full-precision network by 35.7% in the case of DoReFa and 37.1% in the case of WRPN.
bitwidths. They suggest maintaining a high-precision floating point copy of the weights while feeding quantized weights into backprop. WRPN introduces a scheme to train networks from scratch using reduced-precision activations by decreasing the precision of both activations and weights and increasing the number of filter maps in a layer. [21] performs the training phase of the network in full precision, but for inference uses ternary weight assignments. For this assignment, the weights are quantized using two scaling factors which are learned during training phase. SinReQ is a complimentary method that can potentially enhance these algorithms. The paper demonstrates this feature concretely in the context of DoReFa and WRPN training algorithms.

**Binarized and ternarized neural networks.**

Extensive work, [22, 15, 23] focus on binarized neural networks, which impose accuracy loss but reduce the bitwidth to lowest possible level. In BinaryNet [14], an extreme case, a method is proposed for training binarized neural networks which reduce memory size, accesses and computation intensity at the cost of accuracy. XNOR-Net [15] leverages binary operations (such as XNOR) to approximate convolution in binarized neural networks. On the other side, in a weight ternarized network, zero is used as an additional quantized value. [23] introduces ternary-weight networks, in which the weights are quantized to -1, 0, +1 values by minimizing the Euclidian distance between full-precision weights and their ternary assigned values. In [24] different scaling factors are introduced to the ternarized weights. The scaling parameters are learned by gradient descent. None of these techniques propose sinusoidal regularization to make the weight values more quantization friendly as training progresses.

**Loss-aware weight quantization.**

Recent works pursued loss-aware minimization approaches for quantization. [17, 18] developed approximate solutions using proximal Newton algorithm to minimize the loss function directly under the constraints of low bitwidth weights. [19] proposed to learn the quantization of DNNs through regularization by introducing a learnable regularization coefficient to find low bitwidth models efficiently in training. [20] proposed an adaptive technique to jointly train a quantized, bit-operation-compatible DNN and its associated quantizers, as opposed to using fixed, handcrafted quantization schemes such as uniform or logarithmic quantization. Although these techniques use regularization to guide the process of quantized training, they don not explore the use of periodic differentiable trigonometric functions.

## 3. SINUSOIDAL REGULARIZATION FOR AUTOMATIC QUANTIZATION DURING TRAINING

Our proposed method SinReQ exploits weight regularization in order to automatically quantize a neural network while training. To that end, Sections 3.1 to 3.3 describe the role of regularization in neural networks and then Section 3.4 explains SinReQ in more detail.

### 3.1 Loss Landscape of Neural Networks

Neural networks’ loss landscapes are known to be highly non-convex and generally very poorly understood. It has been empirically verified that loss surfaces for large neural networks have many local minima that are essentially equivalent in terms of test error [25], [26]. Moreover, converging to one of the many good local minima proves to be more useful as compared to struggling to find the global minimum of the accuracy loss on the training set (which often leads to overfitting). This opens up and encourages a possibility of adding extra custom objectives to optimize for during the training process, in addition to the original objective (i.e., the accuracy loss). The added custom objective could be with the purpose of increasing generalization performance or imposing some preference on the weights values. Regularization is one of the major techniques that makes use of such facts as discussed in the following subsection.

### 3.2 Regularization in Neural Networks

Neural networks often suffer from redundancy of parameterization and consequently they commonly tend to overfit. Regularization is one of the commonly used techniques to enhance generalization performance of neural networks. Regularization effectively constrains weight parameters by adding a term (regularizer) to the objective function that captures the desired constraint in a soft way. This is achieved by imposing some sort of preference on weight updates during the optimization process. As a result, regularization seamlessly leads to unconditionally constrained optimization problem instead of explicitly constrained which, in most cases, is much more difficult to solve.

### 3.3 Classical regularization: weight decay.

The most commonly used regularization technique is known as weight decay, which aims to reduce the network complexity by limiting the growth of the weights. It is realized by adding a term to the objective function that penalizes large weight values

$$E(w) = E_o(w) + \frac{1}{2} \lambda \sum_i w_i^2$$  \hspace{1cm} (1)

where $E_o$ is the original loss measure, and $\lambda$ is a parameter governing how strongly large weights are penalized. $w_i$ is a vector of all weights for layer $i$ of the network while summation of $i$ is over all the layers in the network.

### 3.4 Periodic Regularization: SinReQ.

In this work, we propose a new type of regularization that is friendly to quantization. The proposed regularization is based on a periodic function (sinusoidal) that provides a smooth and differentiable loss to the original objective. Figure 1 (a). The periodic regularizer has a periodic pattern of minima that correspond to the desired quantization levels. Such correspondence is achieved by matching the period to the quantization step based on a particular number of bits for a given layer.

$$E(w) = E_o(w) + \frac{1}{2} \lambda \sum_i w_i^2 + \frac{1}{2} \lambda_q \sum_i \sin^2 \left( \frac{\pi w_i}{2^{q_{bits} - 1}} \right)$$  \hspace{1cm} (2)

where $E_o$ is the original loss measure, and $\lambda_q$ is SinReQ regularization strength that is a parameter governing how strongly weight quantization errors are penalized. For the sake of simplicity and clarity, Figure 1 (b) and (c) depict a geometrical
4. Evaluation: SinReQ in action

To demonstrate the effectiveness of our proposed sinusoidal regularization, we evaluated it on three neural networks (CIFAR10, SVHN, and ResNet-20) Here, we focus on fine-tuning from a pretrained models as compared to training from scratch.

4.1 Experimental Setup.

We implemented our technique inside Distiller [27], an open source framework for compression by Intel Nervana. The reported accuracies for DoReFa and WRPN are with the built-in implementations in Distiller, which may not exactly match the accuracies reported in their respective papers. However, an independent implementation from a major company provides an unbiased foundation for the comparisons.

4.2 Semi-quantized weight distributions.

Figure 2 shows the evolution of weights distributions over fine-tuning epochs for different layers of (a) CIFAR10 and (b) SVHN networks at different bitwidths (3, 4, and 5 bits). The high-precision weights form clusters and gradually converge.
around the quantization centroids (i.e., levels) as regularization loss is minimized along with the main accuracy loss. The rate of convergence to the target quantization levels depends on (i) the number of fine-tuning epochs, (2) the regularization strength ($\lambda_q$). It is worth noting that $\lambda_q$ is a hyper-parameter that controls the tradeoff between the accuracy loss and the regularization loss. Fixed value can be presumed ahead of training or fine-tuning, however careful setting of such parameter can yield optimum results. [19] considers the regularization coefficient as a learnable parameter.

### 4.3 Arbitrary-bitwidth quantization.

Considering the following sinusoidal regularizer, with $step_q$ denoting the quantization step, and $\Delta$ is an offset.

$$R(W) = \lambda_q \sum_l \sin^2 \left( \frac{\pi w_l + \Delta}{step_q} \right)$$

(3)

SinReQ provides generality in two aspects. First, the flexibility to adapt for arbitrary number of bits. The parameter $step_q$ controls the periodicity of the sinusoidal function. Thus, for any arbitrary bitwidth ($q$bits), $step_q$ can be tuned to match the respective quantization step. For uniform quantization:

$$step_q = 2^{-qbits} - 1$$

The second aspect of generality is the seamless accommodation for different quantization styles. There are two styles of uniform quantization: mid-tread and mid-rise. In mid-tread, zero is considered as a quantization level, while in mid-rise, quantization levels are shifted by half a step such that zero is not included. Ternary quantization, using $\{-1,0,1\}$, is an example of the former, while binary quantization, using $\{-1,1\}$, is an example of the latter. Figure 2 (a) shows the second conv

![Figure 2](image)

**Figure 2:** Evolution of weight distributions over training epochs (with the proposed regularization) at different layers and bitwidths for CIFAR10 and SVHN. (a) CIFAR10, second convolution layer with 3 bits, top row: mid-rise type of quantization (shifting by half a step to exclude zero as a quantization level); bottom row: mid-tread type of quantization (zero is included as a quantization level). (b) SVHN, top row: first convolution layer with 4 bits; bottom row: mid-tread type of quantization (shifting by half a step to exclude zero as a quantization level); bottom row: mid-tread type of quantization.

As different layers have different levels of sensitivity to the quantization bitwidth [28], enabling layer-wise quantization opens the possibility for heterogenous bitwidth quantization and consequently more optimized quantized networks. This can be achieved by adding a custom regularizer (as shown in equation 3) for each layer and sum over all layers. Then, we add the regularization losses of all layers to the main accuracy loss and pass the entire collective loss to the gradient-descent optimizer.

### 4.4 Layer-wise optimization.

As different layers have different levels of sensitivity to the quantization bitwidth [28], enabling layer-wise quantization opens the possibility for heterogenous bitwidth quantization and consequently more optimized quantized networks. This can be achieved by adding a custom regularizer (as shown in equation 3) for each layer and sum over all layers. Then, we add the regularization losses of all layers to the main accuracy loss and pass the entire collective loss to the gradient-descent optimizer.

### 4.5 Comparison to existing methods.

We assess the efficacy of SinReQ on boosting the performance of existing methods for training quantized networks, DoReFa, and WRPN. Table 1 summarizes the accuracies obtained by DoReFa, and WRPN with and without SinReQ. Results show that integrating SinReQ within the training algorithm achieves up to 5.3%, and 2.6% accuracy improvements to DoReFa, and WRPN methods respectively. That is averaging to around 2% and 1.6% respectively.

### 4.6 Convergence analysis.

Figure 3 (a) and (b) show the convergence behavior of SinReQ by visualizing both accuracy and regularization loss over fine-tuning epochs for two networks: CIFAR10 and SVHN. As can be seen, the regularization loss (SinReQ Loss) is minimized across the fine-tuning epochs while the accuracy is maximized. This demonstrates a validity for the proposed regularization being able to optimize the two objectives simultaneously. Figure 3 (c) and (d) contrasts the convergence behavior with and without SinReQ for the case of training from scratch for VGG-11. It can be noticed that, at the onset of training, the accuracy in the presence of SinReQ is behind that without SinReQ. This can be explained as a result of optimizing for an extra objective in case of with SinReQ as compared to without. Shortly thereafter, the regularization effect kicks in and eventually achieves ~6% accuracy improvement. The convergence behavior, however, is primarily controlled by the regularization strength ($\lambda_q$). As briefly mentioned in section

![Figure 3](image)

**Figure 3:** Convergence behavior: accuracy and SinReQ regularization loss over fine-tuning epochs for (a) CIFAR10, (b) SVHN. Comparing convergence behavior with and without SinReQ during training from scratch (a) accuracy, (b) training loss.
3.4. $\lambda_q \in [0, \infty)$ is a hyperparameter that weights the relative contribution of the proposed regularization objective to the standard accuracy objective. In the context of neural networks, it is sometimes desirable to use a separate setting of $\lambda_q$ for each layer of the network. Throughout our experiments, $\lambda_q$ is set the same across all layers and in the range of 0.5 – 10. We reckon that careful setting of $\lambda_q$ across the layers and during the training epochs is essential for optimum results [19].

5. CONCLUSION

Deep quantization of DNNs promises to be a powerful technique in reducing their complexity. However, it comes with the vice of loss in accuracy that needs to be remedied. This paper provided a new approach in using sinusoidal regularizations terms to push the weight values closer to the quantized levels. This mathematical approach is versatile and augments other quantized training algorithms by improving the quality of the network they train. While this technique consistently improves the accuracy, SinReQ does not require changes to the base training algorithm or the neural network topology.

6. REFERENCES


