Learned Step Size Quantization

ICLR 2020

Steven K. Esser * , Jeffrey L. McKinstry, Deepika Bablani,
Rathinakumar Appuswamy, Dharmendra S. Modha

Samsung AI Center Cambridge, UK

Part 1.

Background

- **Deep network in deployed systems**
  - Requirement for applications
    - Task performance, throughput, energy-efficiency, and compactness
  - Create networks for low precision hardware
    - Maintain high accuracy while reducing the precision needed to represent their activations and weights

Jason Wei et al. Chain of thought prompting elicits reasoning in large language models. 2022.

Part 1.

Background

- **Low precision networks**
  - Networks can be trained with stochastic gradient descent by updating high precision weights that are quantized, along with activations
    - Mapping for each quantized layer that maximizes task performance
  - Uniform mappings
    - Uniform quantizers: It can be configured by a single step size parameter (the width of a quantization bin)
  - Nonuniform mappings
    - Use backpropagation with stochastic gradient descent to learn a quantizer that minimizes task loss

Part 1.

Background

- **Low precision networks**
  - Fixed mapping
    - Schemes based on user settings
    - No guarantees on optimizing network performance
  - Quantization error minimization
    - Perfectly minimize quantization error
    - Yet still be non optimal if a different quantization mapping actually minimizes task error
    - As the quantizer itself is discontinuous, such an approach requires approximating its gradient
    - Ignore the impact of transitions between quantized states

Jason Wei et al. Chain of thought prompting elicits reasoning in large language models. 2022.

Introduction

- **Learned Step Size Quantization (LSQ)**
  - Simple way to approximate the gradient to the quantizer step size
    - Sensitive to quantized state transitions
    - Finer grained optimization when learning the step size as a model parameter
  - Simple heuristic to bring the magnitude of step size updates into better balance with weight updates
    - Improves convergence
  - Experimental result
    - Significantly better accuracy than prior quantization approaches
    - Milestone of 3-bit quantized networks reaching full precision network accuracy

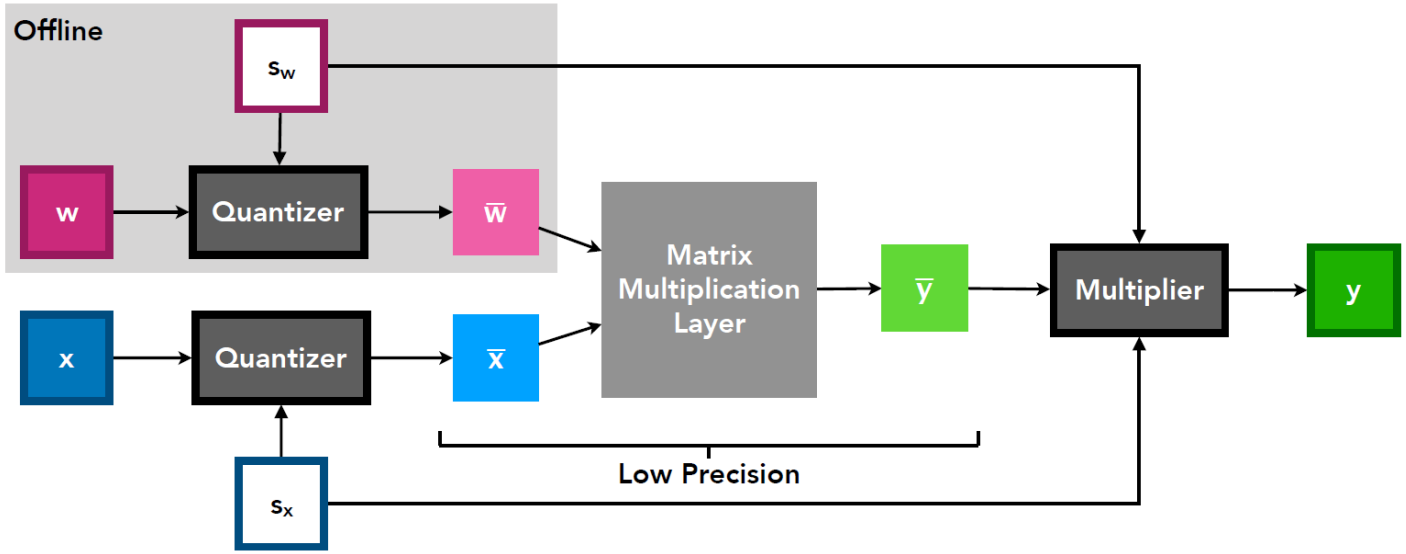
Method

- **Learned Step Size Quantization (LSQ)**
  - At inference time, use low precision integer operations for computations in convolution and fully connected layers
  - Quantizer $\bar{v} = \lfloor clip(v/s, -Q_N, Q_P) \rceil$
  - Quantized representation of the data $\hat{v} = \bar{v} \times s$
  - Quantizer step size s
  - The number of positive and negative quantization levels $Q_P \quad Q_N$
    - Unsigned data (activations) $Q_N = 0 \qquad Q_P = 2^b - 1$
    - Signed data (weights) $Q_N = 2^{b-1} \qquad Q_P = 2^{b-1} - 1$

Part 3.

Method

- **Learned Step Size Quantization (LSQ)**
  - $\bar{w}$ and $\bar{x}$ values
    - Input to low precision integer matrix multiplication units underlying convolution or fully connected layers
    - Output of such layers then rescaled by the step size using a relatively low cost high precision scalar-tensor multiplication
  - Scalar-tensor multiplication
    - Step that can potentially be algebraically merged with other operations such as batch normalization



Computation of a low precision convolution or fully connected layer, as envisioned here

Method

- **Step Size Gradient**
  - Learn s based on the training loss
    - Introducing the following gradient through the quantizer to the step size parameter

$$\frac{\partial \hat{v}}{\partial s} = \begin{cases} -v/s + \lfloor v/s \rceil & \text{if } -Q_N < v/s < Q_P \\ -Q_N & \text{if } v/s \leq -Q_N \\ Q_P & \text{if } v/s \geq Q_P \end{cases}$$

  - Gradient is derived by using the straight through estimator (Bengio et al., 2013)
    - QIL (Jung et al., 2018)
    - PACT (Choi et al., 2018b)
    - LSQ

Yoshua Bengio et al. Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv preprint arXiv:1308.3432. 2013
Sangil Jung et al. Joint training of low-precision neural network with quantization interval parameters. arXiv preprint arXiv:1808.05779. 2018.
Jungwook Choi et al. Pact: Parameterized clipping activation for quantized neural networks. arXiv preprint arXiv:1805.06085. 2018..

Part 3.

Method

- **Step Size Gradient**
  - QIL (Jung et al., 2018)
    - Approximate the gradient through the round function as a pass through operation
  - PACT (Choi et al., 2018b)
    - Estimate the gradient by removing the round operation from the forward equation, algebraically canceling terms
    - Differentiate such that $\partial\hat{v}/\partial s = 0$ where $-Q_N < v/s < Q_P$
  - Both such previous approaches
    - The relative proximity of v to the transition point between quantized states does not impact the gradient to the quantization parameters

$$\frac{\partial \hat{v}}{\partial s} = \begin{cases} -v/s + \lfloor v/s \rceil & \text{if } -Q_N < v/s < Q_P \\ -Q_N & \text{if } v/s \leq -Q_N \\ Q_P & \text{if } v/s \geq Q_P \end{cases}$$

Sangil Jung et al. Joint training of low-precision neural network with quantization interval parameters. arXiv preprint arXiv:1808.05779. 2018.
Jungwook Choi et al. Pact: Parameterized clipping activation for quantized neural networks. arXiv preprint arXiv:1805.06085. 2018..

Method

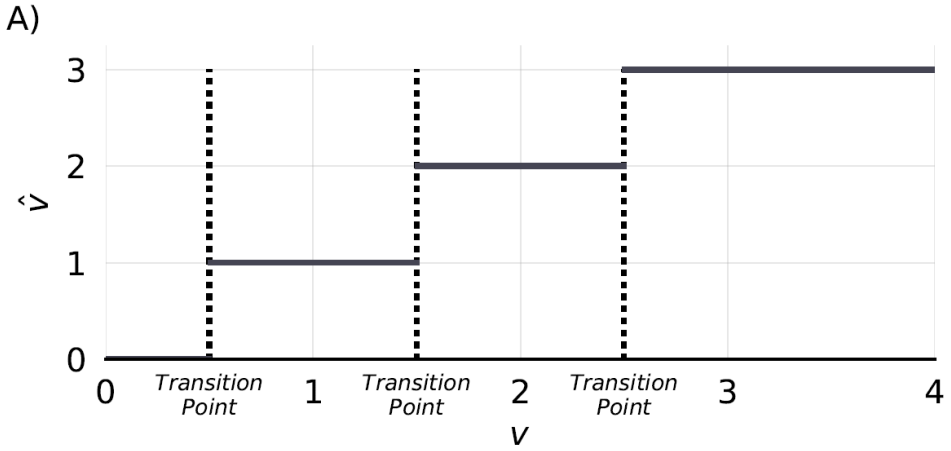
- **Step Size Gradient**
  - Relationship
    - The closer a given v is to a quantization transition point, the more likely it is to change its quantization bin ($\overline{v}$) as a result of a learned update to s
    - Expect $\partial \hat{v} / \partial s$ to increase as the distance from v to a transition point decreases
  - Observe this relationship in the LSQ gradient
    - Gradient naturally falls out of our simple quantizer formulation
    - Use of the straight through estimator for the round function
  - Observe this relationship in the LSQ gradient
    - Each layer of weights and each layer of activations has a distinct step size
    - Represent as an fp32 value
    - Initialize to $2\langle |v| \rangle / \sqrt{Q_P}$
    - Compute on either the initial weights values or the first batch of activations, respectively

Part 3.

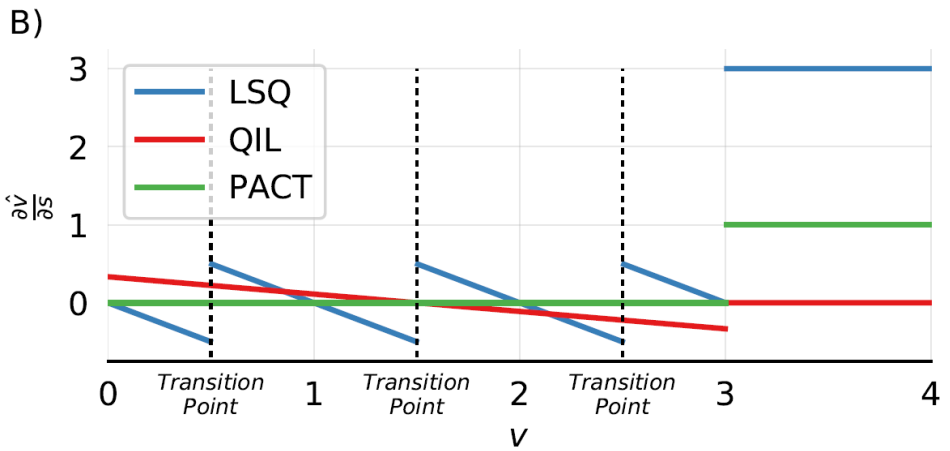
Method

- **Step Size Gradient**
  - Given $s = 1, Q_N = 0, Q_P = 3$, networks with the LSQ gradient reach higher accuracy
  - QIL (Jung et al., 2018)
    - Gradient is sensitive only to the distance from quantizer clip points
  - PACT (Choi et al., 2018b)
    - Gradient is zero everywhere below the clip point
  - LSQ
    - Gradient is sensitive to the distance between v and each transition point

Quantizer output



Gradients of the quantizer output



Yoshua Bengio et al. Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv preprint arXiv:1308.3432. 2013.

Part 3.

Method

- **Step Size Gradient Scale**
  - Reasoning
    - Good convergence is achieved during training where the ratio of average update magnitude to average parameter magnitude is approximately the same for all weight layers in a network (You et al., 2017)
    - Once learning rate has been properly set, this helps to ensure that all updates are neither so large as to lead to repeated overshooting of local minima, nor so small as to lead to unnecessarily long convergence time
  - Conclusion
    - Each step size should also have its update magnitude to parameter magnitude proportioned similarly to that of weights

Ratio

$$R = \frac{\nabla_s L}{s} \Big/ \frac{\|\nabla_w L\|}{\|w\|}$$

Loss Function

$$L$$

L2-Norm

$$\|z\|$$

Yang You et al. Large batch training of convolutional networks. arXiv preprint arXiv:1708.03888, 2017.

Method

- **Step Size Gradient Scale**
  - Step size parameter is smaller as precision increases
    - The data is quantized more finely
  - Step size updates is larger as the number of quantized items increases
    - More items are summed across when computing its gradient
  - Correct the ratio
    - Multiply the step size loss by a gradient scale
    - Weight step size
    - Activation step size
    - The number of features in a layer

g

$g = 1/\sqrt{N_W Q_P}$

$g = 1/\sqrt{N_F Q_P}$

N_f

Ratio

$$R = \frac{\nabla_s L}{s} \Big/ \frac{\|\nabla_w L\|}{\|w\|}$$

Loss Function

L

L2-Norm

$\|z\|$

Yang You et al. Large batch training of convolutional networks. arXiv preprint arXiv:1708.03888, 2017.

Part 3.

Method

- **Training**
  - Common means of training quantized networks (Courbariaux et al., 2015)
    - Full precision weights are stored and updated
    - Quantized weights and activations are used for forward and backward passes
  - Gradient through the quantizer round function is computed using the straight through estimator (Bengio et al., 2013)

$$\frac{\partial \hat{v}}{\partial v} = \begin{cases} 1 & \text{if } -Q_N < v/s < Q_P \\ 0 & \text{otherwise,} \end{cases}$$

  - Use as input $\hat{v}$ to matrix multiplication layers
  - Dataset
    - ImageNet, ResNet, VGG with batch norm, SqueezeNext

Matthieu Courbariaux et al. Binaryconnect: Training deep neural networks with binary weights during propagations. NeurIPS. 2015.
Yoshua Bengio et al. Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv preprint arXiv:1308.3432. 2013

Experiment

- **Weight Decay**
  - Lower precision networks reached higher accuracy with less weight decay
    - Reduce weight decay by half for the 3-bit netw
    - Reduce weight decay by a quarter for the 2-bit network ork

ResNet-18 top-1 accuracy for various weight decay values

Weight Decay	2-bit	3-bit	4-bit	8-bit
10^{-4}	66.9	70.1	**71.0**	**71.1**
0.5×10^{-4}	67.3	**70.2**	70.9	71.1
0.25×10^{-4}	**67.6**	70.0	70.9	71.0
0.125×10^{-4}	67.4	66.9	70.8	71.0

Experiment

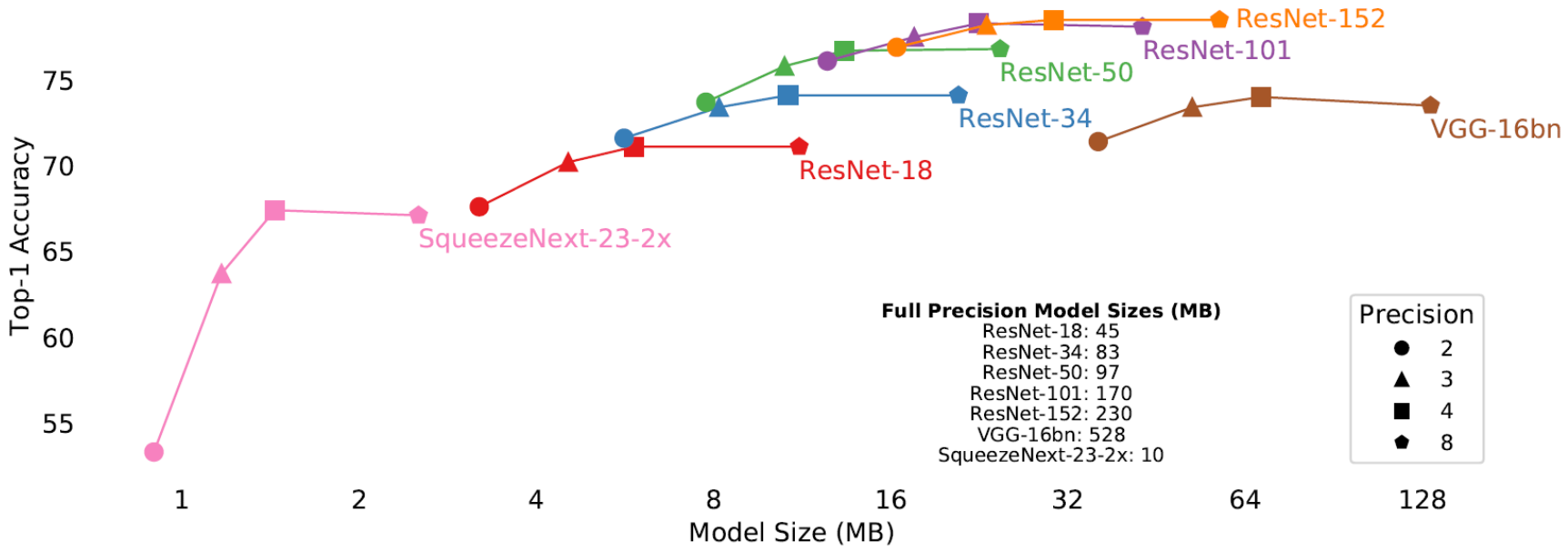
- **Comparison with other approaches**
  - LSQ Achievement
    - Higher top-1 accuracy & top-5 accuracy
  - SqueezeNext performance drop
    - Maximize performance using as few parameters as possible, which may have placed it at a design point extremely sensitive to reductions in precision

Network	Method	Top-1 Accuracy @ Precision 2	3	4	8	Top-5 Accuracy @ Precision 2	3	4	8
ResNet-18		*Full precision: 70.5*				*Full precision: 89.6*			
	LSQ (Ours)	**67.6**	**70.2**	**71.1**	**71.1**	**87.6**	**89.4**	**90.0**	**90.1**
	QIL	65.7	69.2	70.1					
	FAQ			69.8	70.0			89.1	89.3
	LQ-Nets	64.9	68.2	69.3		85.9	87.9	88.8	
	PACT	64.4	68.1	69.2		85.6	88.2	89.0	
	NICE		67.7	69.8			87.9	89.21	
	Regularization	61.7		67.3	68.1	84.4		87.9	88.2
ResNet-152		*Full precision: 78.9*				*Full precision: 94.3*			
	LSQ (Ours)	**76.9**	**78.2**	**78.5**	**78.5**	**93.2**	**93.9**	**94.1**	**94.2**
	FAQ			78.4	**78.5**			**94.1**	94.1
Squeeze Next-23-2x		*Full precision: 67.3*				*Full precision: 87.8*			
	LSQ (Ours)	**53.3**	**63.7**	**67.4**	**67.0**	**77.5**	**85.4**	**87.8**	**87.7**

Experiment

- **Comparison with other approaches**
  - 2-bit ResNet-34 and ResNet-50 networks
    - Offer an absolute advantage over using a smaller network, but with higher precision
  - VGG-16bn
    - Network was developed prior to a number of recent innovations in achieving higher performance with fewer parameters.

Accuracy vs. model size for the networks



Part 4.

Experiment

Ratio

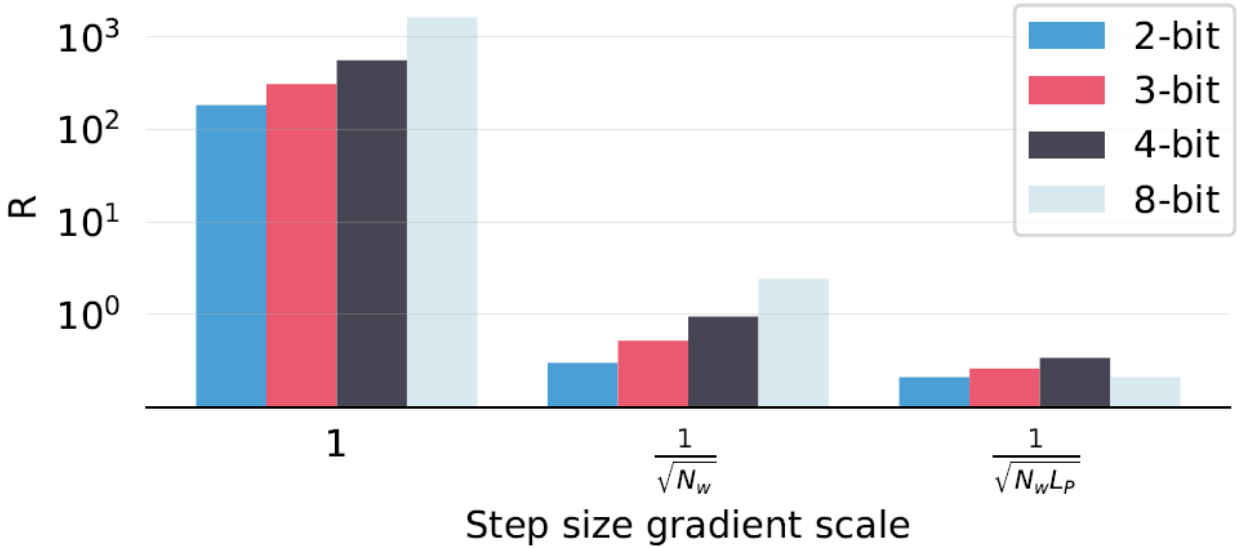
$$R = \frac{\nabla_s L}{s} \Big/ \frac{\|\nabla_w L\|}{\|w\|}$$

Loss Function

$$L$$

- **Step Size Gradient Scale Impact**
  - Relative parameter update magnitudes given different step size gradient scales
    - Gradient scale of $1/\sqrt{N_W Q_P}$ better balances relative step size & weight gradient magnitudes
  - Top-1 accuracy for various gradient scale values
    - Using the full gradient scaling with an additional ten-fold increase or decrease also reduced top-1 accuracy

Relative parameter update magnitudes



Top-1 accuracy for 2-bit ResNet-18

Gradient scale	Learning Rate	Accuracy
$1/\sqrt{NQ_P}$	**0.01**	**67.6**
$1/\sqrt{N}$	0.01	67.3
1 (No gradient scale)	0.01	Did not converge
1 (No gradient scale)	0.0001	64.2
$10/\sqrt{NQ_P}$	0.01	67.4
$1/10\sqrt{NQ_P}$	0.01	67.3

Experiment

- **Cosine Learning Rate Decay Impact**
  - Remove the need to select learning rate schedule hyperparameters
  - Available in most training frameworks
  - Does not increase training time

  - Step-based learning rate decay
    - Use an initial learning rate of 0.01, which was multiplied by 0:1 every 20 epochs

  - Performance
    - Reach top-1 accuracy of 67:2
    - Reach reduction of 0.4 from the equivalent model trained with cosine learning rate decay
    - Mark improvement of 1:5

Experiment

- **Quantization Error**
  - Quantization error
    - Distance between $\hat{v}$ and v on some metric $-\mathrm{E}[\log(q(\hat{v}(s)))]$
  - Final step size learned by LSQ $\hat{s}$
    - Let S be the set of discrete values $\{0.01\hat{s}, 0.02\hat{s}, \ldots, 20.00\hat{s}\}$
  - On a single batch of test data $s \in S$
  - The percent absolute difference between $\hat{s}$ and s

Error Type	Equation	Activation layers	Weight layers
Mean absolute error	$\langle \lvert(\hat{v}(s) - v)\rvert \rangle$	50%	47%
Mean square error	$\langle (\hat{v}(s) - v)^2 \rangle$	63%	28%
Kullback-Leibler divergence	$\int p(v) \log p(v) - \int p(v) \log q(\hat{v}(s))$	64%	46%
Ignore the first term of KL divergence	$-\mathrm{E}[\log(q(\hat{v}(s)))]$		

Part 4.

Experiment

- **Improvement with knowledge distillation**
  - Use the distillation loss function of Hinton et al. (2015) with temperature of 1
  - Teacher network
    - Trained full precision model with frozen weights
    - Same architecture as the low precision network trained
  - Knowledge-distillation can help low precision networks catch up to full precision performance (Mishra & Marr, 2017)

Accuracy for low precision networks trained with LSQ and knowledge distillation

	Top-1 Accuracy @ Precision					Top-5 Accuracy @ Precision				
Network	2	3	4	8	*32*	2	3	4	8	*32*
ResNet-18	67.9	70.6	71.2	71.1	*70.5*	88.1	89.7	90.1	90.1	*89.6*
ResNet-34	72.4	74.3	74.8	74.1	*74.1*	90.8	91.8	92.1	91.7	*91.8*
ResNet-50	74.6	76.9	77.6	76.8	*76.9*	92.1	93.4	93.7	93.3	*93.4*

Geoffrey Hinton et al. Distilling the knowledge in a neural network. NeurIPS. 2014
Asit Mishra and Debbie Marr. Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy. arXiv preprint arXiv:1711.05852, 2017.

Part 5.

Conclusion

- **Learned Step Size Quantization (LSQ)**
  - Method
    - Reduce the number of bits of precision necessary to achieve good performance across a range of network architectures on ImageNet
  - Experimental Result
    - Find best performance when rescaling the quantizer step size loss gradient based on layer size and precision
  - Simple architecture
    - Does not appear to minimize quantization error, requiring only a single additional parameter per weight or activation layer
  - Limitation
    - Not yet clear whether this goal is achievable for 2-bit networks
    - For example, with an 8MB model size limit, a 2-bit ResNet-50 was better than a 4-bit ResNet-34