



Learning Accurate Low-bit Deep Neural Networks with Stochastic Quantization

Yinpeng Dong¹, Renkun Ni², Jianguo Li³,
Yurong Chen³, Jun Zhu¹, Hang Su¹

¹ Department of CST, Tsinghua University

² University of Virginia

³ Intel Labs China

Deep Learning is Everywhere

Self-Driving



Alpha Go



Machine Translation

Google Translate interface showing a neural machine translation example from Chinese to English:

【谷歌NMT，见证奇迹的时刻】
微信最近疯传人工智能新进展：谷歌翻译实现重大突破！
值得关注和庆贺。mt几乎无限量的自然带标数据在新技术下，似乎开始发力。报道说：

十年前，我们发布了 Google Translate（谷歌翻译），这项服务背后的核⼼算法是基于短语的机器翻译（PBMT:Phrase-Based Machine Translation）。

自那时起，机器智能的快速发展已经给我们的语音识别和图像识别能力带来了巨大的提升，但改进机器翻译仍然是一个高难度的目标。

今天，我们宣布发布谷歌神经机器翻译（GNMT: Google Neural Machine Translation）系统，该系统使用了当前最先进的训练技术，能够实现目前为止机器翻译质量的最大提升。我们的全部研究结果详情请参阅我们的论文（Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine）。

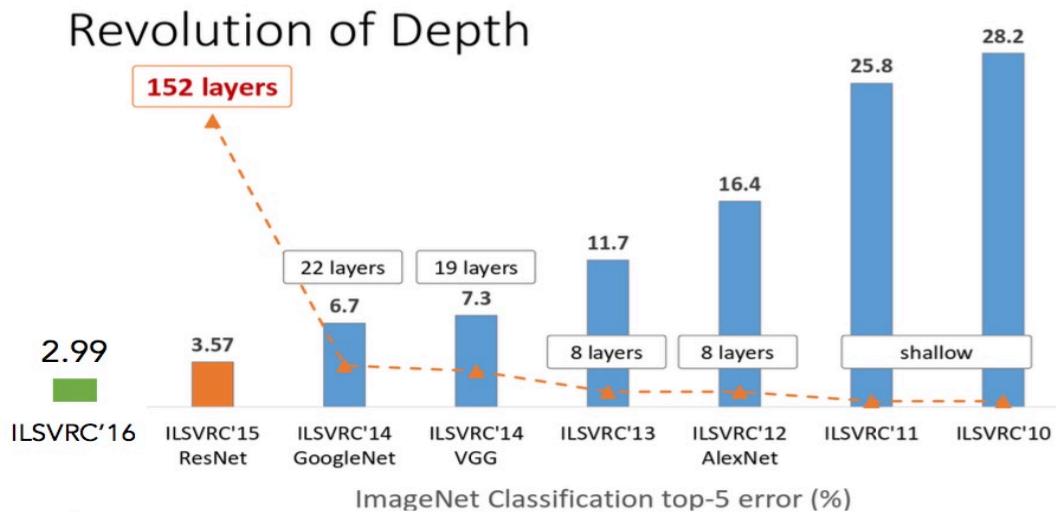
GNMT: Google Neural Machine Translation (GNMT) system, which utilizes state-of-the-art training techniques to maximize the quality of machine translation so far. For a full review of our findings, please see our paper “Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation.”

Dota

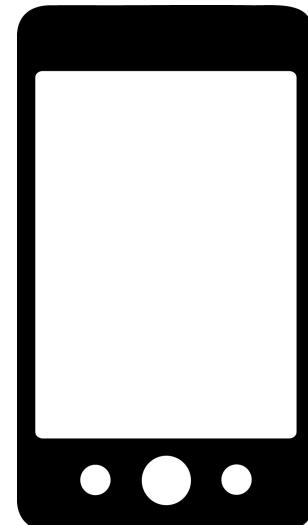


Limitations

- More data + deeper models → more FLOPs + larger memory



- Computation Intensive
- Memory Intensive
- Hard to deploy on mobile devices



Low-bit DNNs for Efficient Inference

- High Redundancy in DNNs;
- Quantize full-precision(32-bits) weights to binary(1 bit) or ternary(2 bits) weights;
- Replace multiplication(convolution) by addition and subtraction;

Typical Low-bit DNNs

- BinaryConnect:

$$B_i = \begin{cases} +1 & \text{with probability } p = \sigma(W_i) \\ -1 & \text{with probability } 1 - p \end{cases}$$

- BWN: minimize $\|W - \alpha B\|$

$$B_i = sign(W_i), \quad \alpha = \frac{\sum_{i=1}^d |W_i|}{d}$$

- TWN: minimize $\|W - \alpha T\|$

$$T_i = \begin{cases} +1 & \text{if } W_i > \Delta \\ 0 & \text{if } |W_i| < \Delta \\ -1 & \text{if } W_i < -\Delta \end{cases}, \quad \alpha = \frac{\sum_{i \in I_\Delta} |W_i|}{|I_\Delta|}$$

$$I_\Delta = \{i \mid |W_i| > \Delta\}, \quad \Delta = \frac{0.7}{d} \sum_{i=1}^d |W_i|$$

Training & Inference of Low-bit DNN

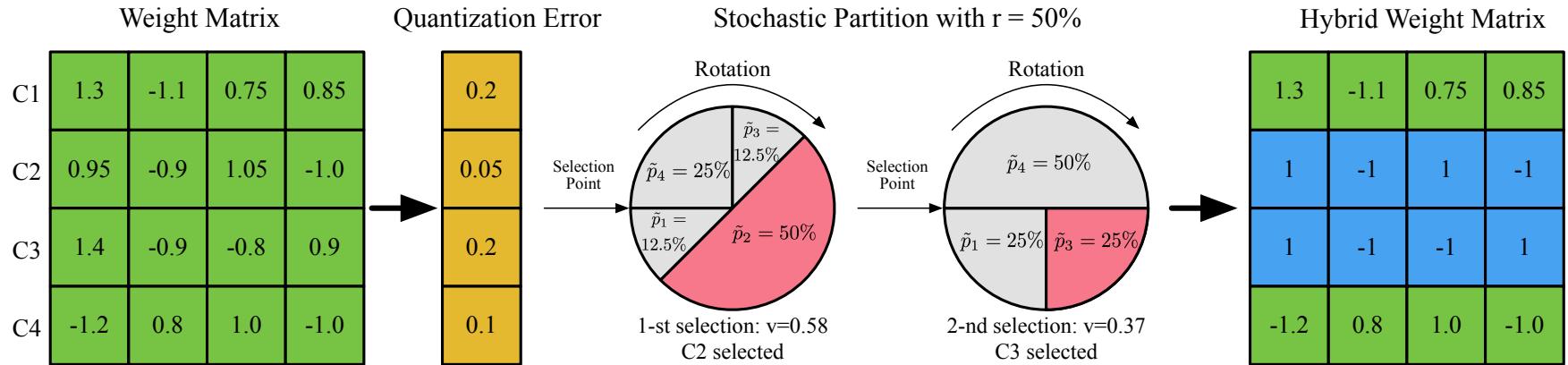
- Let W be the full-precision weights, Q be the low-bit weights ($B, T, \alpha B, \alpha T$).
- Forward propagation: quantize W to Q and perform convolution or multiplication
- Backward propagation: use Q to calculate gradients
- Parameter update: $W^{t+1} = W^t - \eta^t \frac{\partial L}{\partial Q^t}$
- Inference: only need to keep low-bit weights Q

Motivations



- Quantize all weights simultaneously;
 - Quantization error $\|W - Q\|$ may be large for some elements/filters;
 - Induce inappropriate gradient directions.
-
- **Quantize a portion of weights**
 - **Stochastic selection**
 - **Could be applied to any low-bit settings**

Roulette Selection Algorithm



Quantization Error:

$$e_i = \frac{\|W_i - Q_i\|_1}{\|W_i\|_1}$$

Quantization Probability: Larger quantization error

means smaller quantization probability, e.g. $p_i \propto \frac{1}{e_i}$

Quantization Ratio r: Gradually increase to 100%

Training & Inference

- Hybrid weight matrix \tilde{Q}

$$\tilde{Q}_i = \begin{cases} Q_i & \text{if channel } i \text{ being selected} \\ W_i & \text{else} \end{cases}$$

- Parameter update

$$W^{t+1} = W^t - \eta^t \frac{\partial L}{\partial \tilde{Q}^t}$$

- Inference: all weights are quantized; use Q to perform inference

Ablation Studies

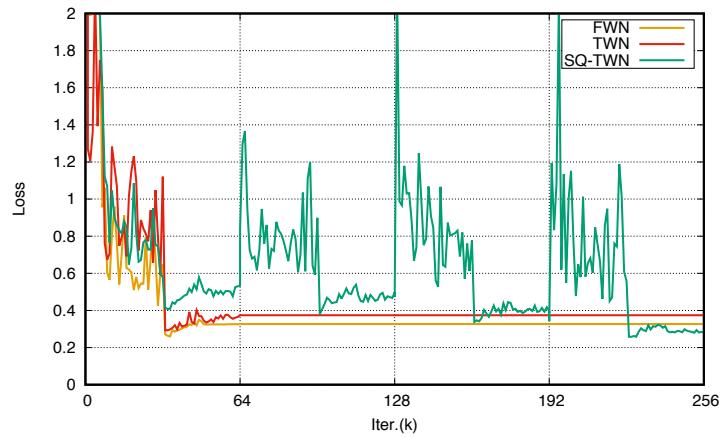
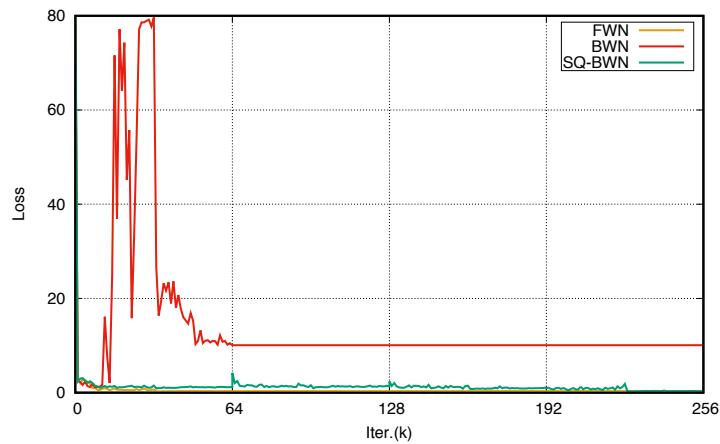


- Selection Granularity:
 - **Filter-level** > Element-level
- Selection/partition algorithms
 - **Stochastic** (roulette) > deterministic (sorting) ~ fixed
(selection only at first iteration)
- Quantization functions
 - **Linear** > Sigmoid > Constant ~ Softmax
 - $p_i = \exp(f_i)/\sum \exp(f_i)$, where $f = \frac{1}{e}$
- Quantization Ratio Update Scheme
 - **Exponential** > Fine-tune > Uniformly
 - 50% → 75% → 87.5% → 100%

Results -- CIFAR



	Bits	CIFAR-10		CIFAR-100	
		VGG-9	ResNet-56	VGG-9	ResNet-56
FWN	32	9.00	6.69	30.68	29.49
BWN	1	10.67	16.42	37.68	35.01
SQ-BWN	1	9.40	7.15	35.25	31.56
TWN	2	9.87	7.51	34.80	32.09
SQ-TWN	2	8.37	6.20	34.24	28.90



Results -- ImageNet

	Bits	AlexNet-BN		ResNet-18	
		top-1	top-5	top-1	top-5
FWN	32	44.18	20.83	34.80	13.60
BWN	1	51.22	27.18	45.20	21.08
SQ-BWN	1	48.78	24.86	41.64	18.35
TWN	2	47.54	23.81	39.83	17.02
SQ-TWN	2	44.70	21.40	36.18	14.26

Conclusions

- We propose a stochastic quantization algorithm for Low-bit DNN training
- Our algorithm can be flexibly applied to all low-bit settings;
- Our algorithm help to consistently improve the performance;
- We release our codes to public for future development
 - <https://github.com/dongyp13/Stochastic-Quantization>

Q & A

