Reduced-memory training and deployment of deep residual networks by stochastic binary quantization

Mark D. McDonnell[1], Ruchun Wang[2] and André van Schaik[2]

cls-lab.org

[1]Computational Learning Systems Laboratory
School of Information Technology &
Mathematical Sciences
University of South Australia

[2]BENS Laboratory
MARCS Institute,
Western Sydney University, Australia

Motivation and Background

Background

- Deep convolutional neural networks
  - Many parameters
  - Many sequential layers
- Following training:
  - Learnt parameters ~10–100 MB
- During training with BP+SGD:
  - Can easily max the 12 GB of RAM in GPUs
  - Mainly temporary storage from FP for use in BP

Motivation

- How can we minimize MB required *during* training with BP+SGD?
- Different goal to model compression *following* training…
  - but we consider this too
  - model compression methods offer ways to reduce RAM access, if not usage, during BP+SGD
- ”Compressed Learning”

Benefits of reducing RAM use during BP+SGD

- Train larger models on a single GPU
- BP+SGD for <u>large</u> models on mobile devices
- Is it always possible/desirable to train at the data center?
  - Personalized or highly-secure fine-tuning
  - rapid-retraining
  - remote deployment: no comms
  - continuous learning with streaming data…

Low bit-width deep CNNs: Prior results

- Iandola et al., “Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size,” Arxiv:1602.07360, 2016
- Courbariaux, Bengio and David, “Binaryconnect: Training deep neural networks with binary weights during propagations,” Arxiv:1511.00363, 2015.
- Hubara et al., “Quantized neural networks: Training neural networks with low precision weights and activations,” Arxiv:1609.07061.
- Merolla et al., “Deep neural networks are robust to weight binarization and other non-linear distortions,” Arxiv:1606.01981, 2016.
- Rastegari et al., “Xnor-net: Imagenet classification using binary convolutional neural networks,” Arxiv:1603.05279, 2016.
- …

Low bit-width deep CNNs: Prior results

1. Model compression
   - Easy to compress convolution parameters to a single bit following training
   - little accuracy penalty
2. Compressed learning
   - Model compression doesn’t help much: parameters updated using full precision
   - Gradients: need 6-12 bits
   - Activations: Use binary nonlinearity layers instead of ReLUs; incurs an accuracy penalty

Our Approach

Our approach for model compression

- Similar to others
  - use the sign of weights for FP and BP
  - Use full-precision weights for updates
- Different to others
  - we found no need to normalise [Rastegari et al]
  - We use new tricks from full-precision CNN training
  - Net result: large improvements on CIFAR-10

Our approach for model compression

- Our improvements come from:
  - Using *wide ResNets*[1] as a baseline:
  - Using standard “light” data augmentation
  - Using a “warm-restart” learning-rate schedule

[1] S. Zagoruyko and N. Komodakis. Wide residual networks. arXiv:1605.07146, 2016.

Our approach for compressed learning

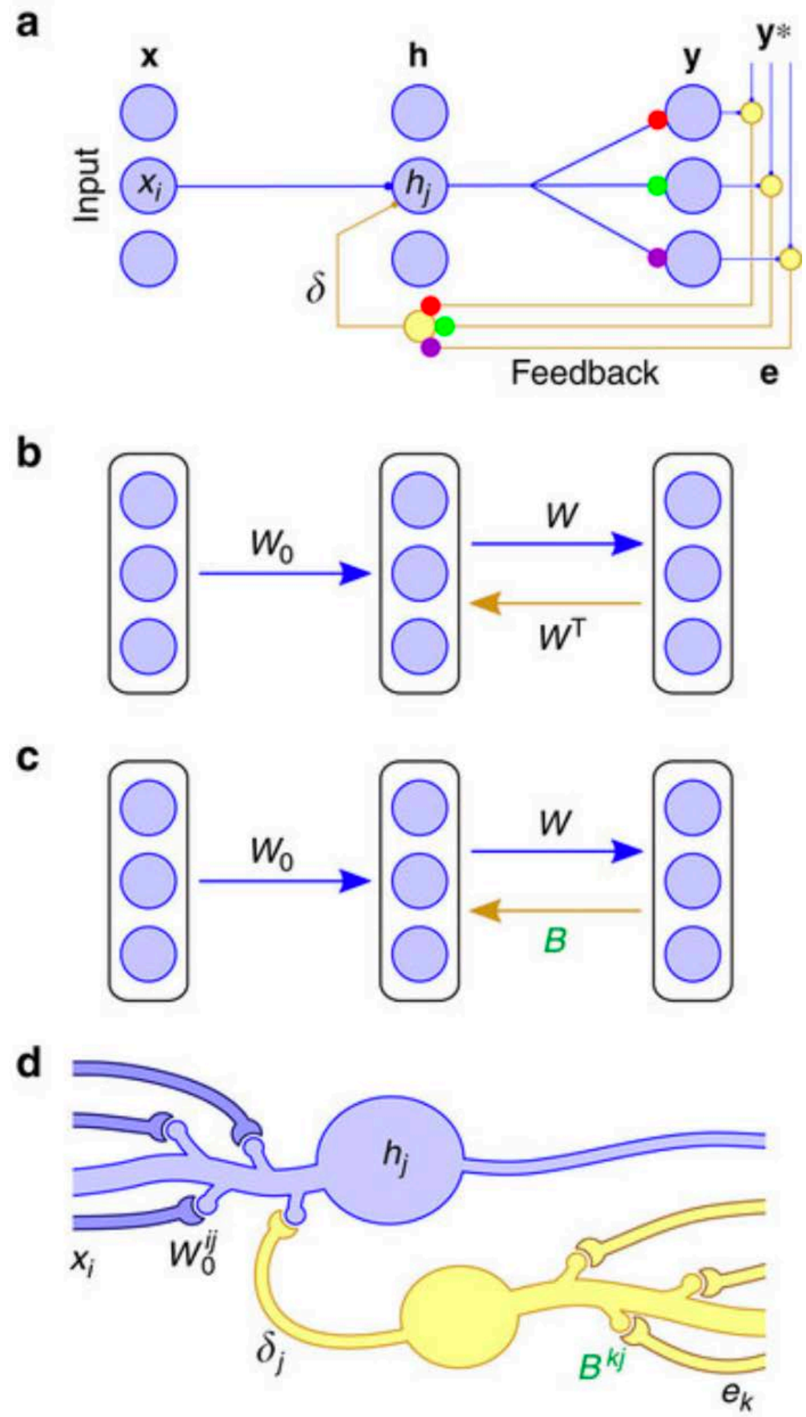
- Inspiration from computational neuroscience: "Feedback alignment"
- Key points:
  - Forward propagation remains unchanged
  - BP with <u>inexact</u> gradient calculations

“Feedback alignment”

Lillicrap et al. “Random synaptic feedback weights support error backpropagation for deep learning,” *Nature Communications*, vol. 7, p. 13276, 2016.

“CINE: Computation-inspired neurobiological elements!”

Thought-provoking 2016 Hinton talk:
“Can the brain do backpropagation?”



Our approach for compressed learning

- Key points we borrow from feedback alignment:
  - Forward propagation remains unchanged
  - BP with inexact gradient calculations
- Different to others:
  - We keep ReLU activations, A, for forward pass
  - We convert to a single bit, A_q only for use in the backward pass
- Our single-bit quantization of activations is stochastic:

$$A_q = \mathrm{I}(A + \text{noise} > 1)$$

Our approach for compressed learning

- Benefits E.g. 20 layer resnet on imagenet
  - 32 bit precision: BP+SGD needs 1.8GB
  - 1 bit precision: 1.8 GB → 56 MB

Our Results

Our Results: Model Compression for CIFAR (single-bit weights following training)

Method	Depth	Width	#params	CIFAR-10	CIFAR-100
32-bit Wide ResNet	28	10	36.5M	4.00%	19.25%
Binary connect (VGG net)[1]	9	8	10.3M	8.27%	N/A
Weight binarization[2] (VGG net)	8	8	11.7M	8.25%	N/A
BWN (VGG net)[3]	8	8	11.7M	9.88%	N/A
Our Wide Resnet	**20**	**4**	**4.3M**	**6.34%**	**23.79%**
Our Wide Resnet	**20**	**10**	**26.8M**	**4.48%**	**22.28%**

We used only 63 epochs for width=4 and 127 for width=10

[1]Courbariaux et al., "Binaryconnect: Training deep neural networks with binary weights during propagations," Arxiv:1511.00363, 2015.
[2]Hubara et al., "Quantized neural networks: Training neural networks with low precision weights and activations," Arxiv:1609.07061.
[3]Rastegari et al., "Xnor-net: Imagenet classification using binary convolutional neural networks," Arxiv:1603.05279, 2016.

Our Results: Model Compression for CIFAR (single-bit weights following training)

Method	Depth	Width	#params	Top-1	Top-5
32-bit ResNet	20	1	11.5M	30.70%	10.80%
BNN (googlenet)[1]	13	-		52.9%	30.90%
BWN (ResNet)[2]	20	1	11.5M	39.2%	17.0%
Our Resnet	**20**	**1**	**11.5M**	**44.48%**	**20.9%**

We need to train for longer…

[1]Hubara et al., “Quantized neural networks: Training neural networks with low precision weights and activations,” Arxiv:1609.07061.
[2]Rastegari et al., “Xnor-net: Imagenet classification using binary convolutional neural networks,” Arxiv:1603.05279, 2016.

Our Results: Compressed Learning for CIFAR

Method	Depth	Width	#params	CIFAR-10	CIFAR-100
32-bit Wide ResNet	28	10	36.5M	4.00%	19.25%
BNN (GoogleMet)[1]	9	8	10.3M	10.15%	N/A
Xnor-net (ResNet)[2]	8	8	11.7M	10.17%	N/A
Our Wide Resnet	**20**	**4**	**4.3M**	**6.86%**	**25.93%**
Our Wide Resnet	**20**	**10**	**26.8M**	**5.43%**	**23.01%**
Our Wide Resnet + model compression	**20**	**10**	**26.8M**	**5.55%**	**23.7%**

[1]Hubara et al., “Quantized neural networks: Training neural networks with low precision weights and activations,” Arxiv:1609.07061.

[2]Rastegari et al., “Xnor-net: Imagenet classification using binary convolutional neural networks,” Arxiv:1603.05279, 2016.

Summary

Model compression

- We achieved SOTA error rates on CIFAR-10 when using 1-bit weights at test time
- Same as error rates for full-precision!
- Achieved using far fewer training epochs

Learning compression

- 32 x reduced memory during BP+SGD
- Error rates fell by only ~1% (absolute)
- Drawback: cannot use xnor approache
- Advantage: better and faster learning

Next steps

- More training on Imagenet
- Faster BP+SGD using improved methods of feedback alignment
- Theory for why our approach works
- Add low bit-width gradients and updates
- Ultimately: low-power hardware BP+SGD
- Applications: not just supervised classifiers!

Thanks for your attention!

mark.mcdonnell@unisa.edu.au

cls-lab.org

Mark D. McDonnell[1], Ruchun Wang[2] and André van Schaik[2]