Training wide residual networks for deployment using a single bit for each weight

University of South Australia

Mark D. McDonnell

Computational Learning Systems Laboratory, University of South Australia, Australia

CLS LABORATORY

cls-lab.org

1. Motivation: Compression of large trained neural network models

- Today's deep-learning libraries: models are stored using 32 bits per weight.
- Model sizes can be 10s or 100s of MBs: challenging to embed trained networks that run fast enough in custom-hardware designs, e.g. for IoT.
- BUT: deep neural networks don't need many bits of precision; can even work with 1-bit-per-weight and two-level activations (1, 2).
- What is the best way to train a model to achieve 1-bit-per-weight?

2. Strategy for 1-bit-per-weight models with minimal performance loss

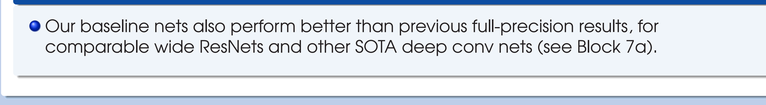
1. State-of-the-art baseline: Wide Residual Networks (3).
2. Make minimal changes when training for 1-bit-per-weight.
3. Simplicity is desirable in custom hardware.

2a. This simple change enables 1-bit-per weight trained models:

- **Key idea 1**: During training, use the sign of weights for forward-prop and back-prop, but use full precision for updates (as proposed previously (1, 2))
- **Key idea 2**: For faster and better convergence, *always* scale binary weights by the *initial* standard deviation of weights in each conv layer. E.g. for He-normal initialization:

$$\mathbf{W}_i = \sqrt{\frac{2}{F_i^2 C_{i-1}}} \text{sign}\left(\mathbf{W}_i\right), \quad i = 1, \ldots, L,$$

where C_i is the number of input channels and F_i is the kernel size in layer i.



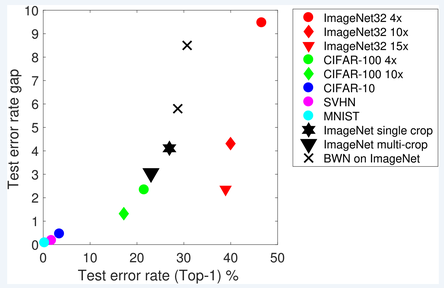
3. Summary of findings

3a. Achievement of 32x model compression with higher accuracy

- We trained wide residual networks with ∼1 to ∼100 Million weights.
- Our results are the best-reported 1-bit-per-weight performance across multiple datasets (see Block 7b), and required far fewer epochs than other methods.
- Model sizes are reduced by a factor of 32 by using 1-bit-per-weight.
- BUT! There is a performance tradeoff:

3b. Mind the gap!

For six datasets, the accuracy gap between 32-bits-per-weight and 1-bit-per-weight decreases with 32-bit accuracy. **Note: we decreased the gap for Imagenet to 3%.**



3c. Improvements in full-precision wide residual networks

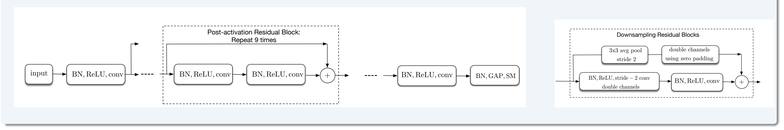
- Our baseline nets also perform better than previous full-precision results, for comparable wide ResNets and other SOTA deep conv nets (see Block 7a).

4. Training and architecture innovations

Best results for *both* full-precision (32 bits) and 1-bit-per-weight were enabled by:

1. Not learning batch-norm scales and offsets (CIFAR-10, CIFAR-100, and SVHN).
2. A 1×1 convolutional layer as the final weight layer, *prior* to global average pooling.
3. A warm-restart learning-rate schedule (4) gives better accuracy and faster convergence.
4. Cutout augmentation (5) (for CIFAR-10 and CIFAR-100).

Wide ResNet architecture used

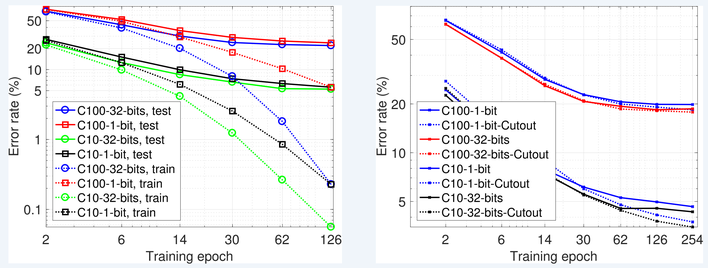


5. Results for CIFAR-10 and CIFAR-100

5a. Test error rates for 32-bits and 1-bit per weight

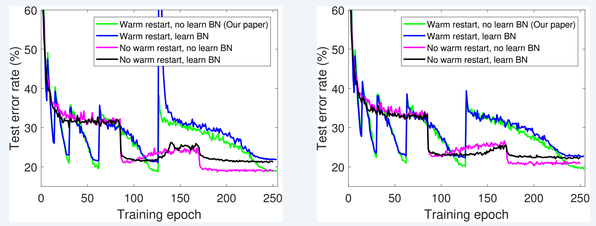
Bits per weight	ResNet	Epochs	Params	C10	C100	C10, cutout	C100, cutout
32-bits	20-4	126	4.3M	5.02	21.53	4.39	20.48
32-bits	20-10	254	26.8M	4.22	18.76	3.46	17.19
32-bits	26-10	254	35.6M	4.23	18.63	3.54	17.22
32-bits	20-20	126	107.0M	-	18.14	-	-
1-bit	20-4	126	4.3M	6.13	23.87	5.34	23.74
1-bit	20-10	254	26.8M	4.72	19.35	3.92	18.51
1-bit	26-10	254	35.6M	4.46	18.94	3.41	18.50
1-bit	20-20	126	107.0M	-	18.81	-	-

5b. Convergence Comparison for 20-4 Wide ResNets



5c. Ablation Analysis for CIFAR-100 and 20-4 Wide ResNets

- These figures illustrate the benefits of warm-restart and not learning BN scale and offset.
- This is for both full precision (left) and, (more prounced) for 1-bit-per-weight (right).



Note!

Not learning BN parameters is only effective when overfitting is high, such as for CIFAR-10 and CIFAR-100. We did not find this method useful for Imagenet.

6. Results for SVHN, ImageNet32, and ImageNet

Weights	ResNet	Epochs	Params	SVHN	I32	ImageNet
32-bits	20-4	30	4.5M	1.75	46.61 / 22.91	-
32-bits	20-10	30	27.4M	-	39.96 / 17.89	-
32-bits	20-15	30	61.1M	-	38.90 / 17.03	-
32-bits	18-2.5	62	70.0M	-	-	Single crop: 26.92 / 9.20 Multi-crop: 22.99 / 6.91
1-bit	20-4	30	4.5M	1.93	56.08 / 30.88	-
1-bit	20-10	30	27.4M	-	44.27 / 21.09	-
1-bit	26-10	62	∼36M	-	41.36 / 18.93	-
1-bit	20-15	30	61.1M	-	41.26 / 19.08	-
1-bit	18-2.5	62	70.0M	-	-	Single crop: 31.03 / 11.51 Multi-crop: 26.04 / 8.48

7. Comparison with previous results

7a. Test-set error rates for networks with less than 40 Million parameters

Method	# params	C10	C100	I32 Top-1 / Top-5
WRN 22-10 (3)	27M	4.44	20.75	-
1-bit weights WRN 20-10 (This Paper)	27M	4.72	19.35	44.27 / 21.09
WRN 28-10 (6)	37M	-	-	40.97 / 18.87
Full precision WRN 20-10 (This Paper)	27M	4.22	18.76	39.96 / 17.89
1-bit weights WRN 20-10 + cutout (This Paper)	27M	3.92	18.51	-
WRN 28-10 + cutout (5)	34M	3.08	18.41	-
WRN 28-10 + dropout (3)	37M	3.80	18.30	-
ResNeXt-29, 8×64d (7)	36M	3.65	17.77	-
Full precision WRN 20-10 + cutout (This Paper)	27M	3.46	17.19	-
DenseNets (8)	26M	3.46	17.18	-
Shake-shake regularization (9)	26M	2.86	15.97	-
Shake-shake + cutout (5)	26M	2.56	15.20	-

7b. Test error rates using 1-bit-per-weight at test time

Method	C10	C100	SVHN	ImageNet
BC (1)	8.27	-	2.30	-
Weight binarization (10)	8.25	-	-	-
BWN - Googlenet (2)	9.88	-	-	34.5 / 13.9 (full ImageNet)
VGG+HWGQ (11)	7.49	-	-	-
BC with ResNet + ADAM (12)	7.17	35.34	-	52.11 (full ImageNet)
BW with VGG (11)	-	-	-	34.5 (full ImageNet)
This Paper: single center crop	3.41	18.50	1.93	41.26 / 19.08 (ImageNet32) 31.03 / 11.51 (full ImageNet)
This Paper: 5 scales, 5 random crops	-	-	-	26.04 / 8.48 (full ImageNet)

8. Further Work

- Can the accuracy gap be theoretically predicted or bounded?
- Can batch-normalization be removed entirely without accuracy loss?
- Can the method be adapted to recurrent networks?

9. References

(1) M. Courbariaux, Y. Bengio, and J.-P. David. BinaryConnect: Training Deep Neural Networks with binary weights during propagations. Arxiv:1511.00363, 2015.

(2) M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi. XNOR-Net: Imagenet classification using binary convolutional neural networks. Arxiv:1603.05279, 2016.

(3) S. Zagoruyko and N. Komodakis. Wide residual networks. Arxiv:1605.07146, 2016.

(4) I. Loshchilov and F. Hutter. SGDR: stochastic gradient descent with restarts. Arxiv:1608.03983, 2016.

(5) T. Devries and G. W. Taylor. Improved regularization of convolutional neural networks with cutout. Arxiv:1708.04552, 2017.

(6) P. Chrabaszcz, I. Loshchilov, and F. Hutter. A downsampled variant of imagenet as an alternative to the CIFAR datasets. Arxiv:1707.08819, 2017.

(7) S. Xie, R. B. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. Arxiv:1611.05431, 2016.

(8) G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. Arxiv:1608.06993, 2016.

(9) X. Gastaldi. Shake-shake regularization. Arxiv:1705.07485, 2017.

(10) P. Merolla, R. Appuswamy, J. V. Arthur, S. K. Esser, and D. S. Modha. Deep neural networks are robust to weight binarization and other non-linear distortions. Arxiv:1606.01981, 2016.

(11) Z. Cai, X. He, J. Sun, and N. Vasconcelos. Deep learning with low precision by half-wave Gaussian quantization. Arxiv:1702.00953, 2017.

(12) H. Li, S. De, Z. Xu, C. Studer, H. Samet, and T. Goldstein. Training quantized nets: A deeper understanding. Arxiv:1706.02379, 2017.

Funding Acknowledgement

This work was supported by a Discovery Project funded by the Australian Research Council (project number DP170104600).