Compression Results —Zhou Zhengguang 2017-3-6

Notes:
1. JETC: ACM Journal on Emerging Technologies in Computing Systems
2. SiPS: Signal Processing Systems
3. NC: Neurocomputing-Elsevier
4. JMLR: Journal of Machine Learning Research
5. CoRR: Computing Research Repository
6. ISCA: International Symposium on Computer Architecture
3. CDICS: Computer-Aided Design of Integrated Circuits and Systems
8. PDPS: Parallel and Distributed Processing Symposium
9. FPGA: International Symposium on Field-Programmable Gate Arrays
10. IS: Interspeech
11. ICASSP: Acoustics, Speech and Signal Processing
12. ASR: Automatic Speech Recognition

Notes2:

1. p:performance. 2. b:better. 3. g:GPU. 4. c:CPU. 5: s:speed. 6. c:compress. 7. ps: perform similar 8. is: inference speed 9. -p:-parameter 10. nl:no loss 11. mg: mobile GPU 12. ls: layerwise speed 13. ee: energy efficiency 14. nn: neural network 15. -c:-cost 16. ec: energy consumption 17. co: convolutional operations

References

[1] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[2] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

[3] Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. Harnessing deep neural networks with logic rules. *arXiv preprint arXiv:1603.06318*, 2016.

[4] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *Advances in neural information processing systems*, pages 2654–2662, 2014.

[5] Gregor Urban, Krzysztof J Geras, Samira Ebrahimi Kahou, Ozlem Aslan, Shengjie Wang, Rich Caruana, Abdelrahman Mohamed, Matthai Philipose, and Matt Richardson. Do deep convolutional nets really need to be deep and convolutional? *arXiv preprint arXiv:1603.05691*, 2016.

[6] William Chan, Nan Rosemary Ke, and Ian Lane. Transferring knowledge from a rnn to a dnn. *arXiv preprint arXiv:1504.01483*, 2015.

[7] Ping Luo, Zhenyao Zhu, Ziwei Liu, Xiaogang Wang, Xiaoou Tang, et al. Face model compression by distilling knowledge from neurons. In *AAAI*, pages 3560–3566, 2016.

[8] Yann LeCun, John S Denker, and Sara A Solla. Optimal brain damage. In *Advances in neural information processing systems*, pages 598–605, 1990.

[9] Babak Hassibi, David G Stork, et al. Second order derivatives for network pruning: Optimal brain surgeon. *Advances in neural information processing systems*, pages 164–164, 1993.

[10] Misha Denil, Babak Shakibi, Laurent Dinh, Nando de Freitas, et al. Predicting parameters in deep learning. In *Advances in Neural Information Processing Systems*, pages 2148–2156, 2013.

[11] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems*, pages 1135–1143, 2015.

[12] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[13] Song Han, Jeff Pool, Sharan Narang, Huizi Mao, Enhao Gong, Shijian Tang, Erich Elsen, Peter Vajda, Manohar Paluri, John Tran, et al. Dsd: Dense-sparse-dense training for deep neural networks. 2016.

[14] Suraj Srinivas and R Venkatesh Babu. Data-free parameter pruning for deep neural networks. *arXiv preprint arXiv:1507.06149*, 2015.

[15] Yiwen Guo, Anbang Yao, and Yurong Chen. Dynamic network surgery for efficient dnns. In *Advances In Neural Information Processing Systems*, pages 1379–1387, 2016.

[16] Jongsoo Park, Sheng Li, Wei Wen, Ping Tak Peter Tang, Hai Li, Yiran Chen, and Pradeep Dubey. Faster cnns with direct sparse convolutions and guided pruning. 2016.

[17] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient transfer learning. *arXiv preprint arXiv:1611.06440*, 2016.

[18] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016.

[19] Sajid Anwar, Kyuyeon Hwang, and Wonyong Sung. Structured pruning of deep convolutional neural networks. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 13(3):32, 2017.

[20] Mohammad Babaeizadeh, Paris Smaragdis, and Roy H Campbell. A simple yet effective method to prune dense layers of neural networks. 2016.

[21] Tien-Ju Yang, Yu-Hsin Chen, and Vivienne Sze. Designing energy-efficient convolutional neural networks using energy-aware pruning. *arXiv preprint arXiv:1611.05128*, 2016.

[22] Christos Louizos, Karen Ullrich, and Max Welling. Bayesian compression for deep learning. *arXiv preprint arXiv:1705.08665*, 2017.

[23] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

[24] Zelda Mariet and Suvrit Sra. Diversity networks. *arXiv preprint arXiv:1511.05077*, 2015.

[25] Jian-Hao Luo and Jianxin Wu. An entropy-based pruning method for cnn compression. *arXiv preprint arXiv:1706.05791*, 2017.

[26] Kyuyeon Hwang and Wonyong Sung. Fixed-point feedforward deep neural network design using weights+ 1, 0, and- 1. In *Signal Processing Systems (SiPS), 2014 IEEE Workshop on*, pages 1–6. IEEE, 2014.

[27] Darryl Lin, Sachin Talathi, and Sreekanth Annapureddy. Fixed point quantization of deep convolutional networks. In *International Conference on Machine Learning*, pages 2849–2858, 2016.

[28] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. In *Advances in Neural Information Processing Systems*, pages 3123–3131, 2015.

[29] Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830*, 2016.

[30] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Quantized neural networks: Training neural networks with low precision weights and activations. *arXiv preprint arXiv:1609.07061*, 2016.

[31] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European Conference on Computer Vision*, pages 525–542. Springer, 2016.

[32] Minje Kim and Paris Smaragdis. Bitwise neural networks. *arXiv preprint arXiv:1601.06071*, 2016.

[33] Yunchao Gong, Liu Liu, Ming Yang, and Lubomir Bourdev. Compressing deep convolutional networks using vector quantization. *arXiv preprint arXiv:1412.6115*, 2014.

[34] Guillaume Soulié, Vincent Gripon, and Maëlys Robert. Compression of deep neural networks on the fly. In *International Conference on Artificial Neural Networks*, pages 153–160. Springer, 2016.

[35] Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*, 2016.

[36] Fengfu Li, Bo Zhang, and Bin Liu. Ternary weight networks. *arXiv preprint arXiv:1605.04711*, 2016.

[37] Chenzhuo Zhu, Song Han, Huizi Mao, and William J Dally. Trained ternary quantization. *arXiv preprint arXiv:1612.01064*, 2016.

[38] Aojun Zhou, Anbang Yao, Yiwen Guo, Lin Xu, and Yurong Chen. Incremental network quantization: Towards lossless cnns with low-precision weights. *arXiv preprint arXiv:1702.03044*, 2017.

[39] Jiaxiang Wu, Cong Leng, Yuhang Wang, Qinghao Hu, and Jian Cheng. Quantized convolutional neural networks for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4820–4828, 2016.

[40] Wenlin Chen, James Wilson, Stephen Tyree, Kilian Weinberger, and Yixin Chen. Compressing neural networks with the hashing trick. In *International Conference on Machine Learning*, pages 2285–2294, 2015.

[41] Ryan Spring and Anshumali Shrivastava. Scalable and sustainable deep learning via randomized hashing. *arXiv preprint arXiv:1602.08194*, 2016.

[42] Lei Shi, Shikun Feng, et al. Functional hashing for compressing neural networks. *arXiv preprint arXiv:1605.06560*, 2016.

[43] Wenlin Chen, James T Wilson, Stephen Tyree, Kilian Q Weinberger, and Yixin Chen. Compressing convolutional neural networks. *arXiv preprint arXiv:1506.04449*, 2015.

[44] Zhouhan Lin, Matthieu Courbariaux, Roland Memisevic, and Yoshua Bengio. Neural networks with few multiplications. *arXiv preprint arXiv:1510.03009*, 2015.

[45] Zhiyong Cheng, Daniel Soudry, Zexi Mao, and Zhenzhong Lan. Training binary multilayer neural networks for image classification using expectation backpropagation. *arXiv preprint arXiv:1503.03562*, 2015.

[46] Vincent Vanhoucke, Andrew Senior, and Mark Z Mao. Improving the speed of neural networks on cpus. In *Proc. Deep Learning and Unsupervised Feature Learning NIPS Workshop*, volume 1, page 4, 2011.

[47] Karen Ullrich, Edward Meeds, and Max Welling. Soft weight-sharing for neural network compression. *arXiv preprint arXiv:1702.04008*, 2017.

[48] Yoojin Choi, Mostafa El-Khamy, and Jungwon Lee. Towards the limit of network quantization. *arXiv preprint arXiv:1612.01543*, 2016.

[49] Alexander Novikov, Dmitrii Podoprikhin, Anton Osokin, and Dmitry P Vetrov. Tensorizing neural networks. In *Advances in Neural Information Processing Systems*, pages 442–450, 2015.

[50] Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. Deep learning with limited numerical precision. In *ICML*, pages 1737–1746, 2015.

[51] Jordan L Holi and J-N Hwang. Finite precision error analysis of neural network hardware implementations. *IEEE Transactions on Computers*, 42(3):281–290, 1993.

[52] Zhaowei Cai, Xiaodong He, Jian Sun, and Nuno Vasconcelos. Deep learning with low precision by half-wave gaussian quantization. *arXiv preprint arXiv:1702.00953*, 2017.

[53] Zhaofan Qiu, Ting Yao, and Tao Mei. Deep quantization: Encoding convolutional activations with deep generative model. *arXiv preprint arXiv:1611.09502*, 2016.

[54] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[55] Jonghoon Jin, Aysegul Dundar, and Eugenio Culurciello. Flattened convolutional neural networks for feedforward acceleration. *arXiv preprint arXiv:1412.5474*, 2014.

[56] Roberto Rigamonti, Amos Sironi, Vincent Lepetit, and Pascal Fua. Learning separable filters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2754–2761, 2013.

[57] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Speeding up convolutional neural networks with low rank expansions. *arXiv preprint arXiv:1405.3866*, 2014.

[58] Emily L Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. In *Advances in Neural Information Processing Systems*, pages 1269–1277, 2014.

[59] Vadim Lebedev, Yaroslav Ganin, Maksim Rakhuba, Ivan Oseledets, and Victor Lempitsky. Speeding-up convolutional neural networks using fine-tuned cp-decomposition. *arXiv preprint arXiv:1412.6553*, 2014.

[60] Xiangyu Zhang, Jianhua Zou, Xiang Ming, Kaiming He, and Jian Sun. Efficient and accurate approximations of nonlinear convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1984–1992, 2015.

[61] Shuchang Zhou and Jia-Nan Wu. Compression of fully-connected layer in neural network by kronecker product. *arXiv preprint arXiv:1507.05775*, 2015.

[62] Jian Xue, Jinyu Li, and Yifan Gong. Restructuring of deep neural network acoustic models with singular value decomposition. In *Interspeech*, pages 2365–2369, 2013.

[63] Yu Cheng, Felix X Yu, Rogerio S Feris, Sanjiv Kumar, Alok Choudhary, and Shi-Fu Chang. An exploration of parameter redundancy in deep networks with circulant projections. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2857–2865, 2015.

[64] Cheng Tai, Tong Xiao, Yi Zhang, Xiaogang Wang, et al. Convolutional neural networks with low-rank regularization. *arXiv preprint arXiv:1511.06067*, 2015.

[65] Zichao Yang, Marcin Moczulski, Misha Denil, Nando de Freitas, Alex Smola, Le Song, and Ziyu Wang. Deep fried convnets. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1476–1483, 2015.

[66] Yani Ioannou, Duncan Robertson, Jamie Shotton, Roberto Cipolla, and Antonio Criminisi. Training cnns with low-rank filters for efficient image classification. *arXiv preprint arXiv:1511.06744*, 2015.

[67] Min Wang, Baoyuan Liu, and Hassan Foroosh. Factorized convolutional neural networks. *arXiv preprint arXiv:1608.04337*, 2016.

[68] Yong-Deok Kim, Eunhyeok Park, Sungjoo Yoo, Taelim Choi, Lu Yang, and Dongjun Shin. Compression of deep convolutional neural networks for fast and low power mobile applications. *arXiv preprint arXiv:1511.06530*, 2015.

[69] Peisong Wang and Jian Cheng. Accelerating convolutional neural networks for mobile applications. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 541–545. ACM, 2016.

[70] Vikas Sindhwani, Tara Sainath, and Sanjiv Kumar. Structured transforms for small-footprint deep learning. In *Advances in Neural Information Processing Systems*, pages 3088–3096, 2015.

[71] Min Wang, Baoyuan Liu, and Hassan Foroosh. Design of efficient convolutional layers using single intra-channel convolution, topological subdivisioning and spatial bottleneck structure.

[72] Tara N Sainath, Brian Kingsbury, Vikas Sindhwani, Ebru Arisoy, and Bhuvana Ramabhadran. Low-rank matrix factorization for deep neural network training with high-dimensional output targets. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6655–6659. IEEE, 2013.

[73] Sek Chai, Aswin Raghavan, David Zhang, Mohamed Amer, and Tim Shields. Low precision neural networks using subband decomposition. *arXiv preprint arXiv:1703.08595*, 2017.

[74] Baoyuan Liu, Min Wang, Hassan Foroosh, Marshall Tappen, and Marianna Pensky. Sparse convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 806–814, 2015.

[75] Simone Scardapane, Danilo Comminiello, Amir Hussain, and Aurelio Uncini. Group sparse regularization for deep neural networks. *Neurocomputing*, 241:81–89, 2017.

[76] Soravit Changpinyo, Mark Sandler, and Andrey Zhmoginov. The power of sparsity in convolutional neural networks. *arXiv preprint arXiv:1702.06257*, 2017.

[77] Benjamin Graham. Spatially-sparse convolutional neural networks. *arXiv preprint arXiv:1409.6070*, 2014.

[78] Guoliang Kang, Jun Li, and Dacheng Tao. Shakeout: A new approach to regularized deep neural network training. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[79] Markus Thom and Günther Palm. Sparse activity and sparse connectivity in supervised learning. *Journal of Machine Learning Research*, 14(Apr):1091–1143, 2013.

[80] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. In *Advances in Neural Information Processing Systems*, pages 2074–2082, 2016.

[81] Mikhail Figurnov, Aizhan Ibraimova, Dmitry P Vetrov, and Pushmeet Kohli. Perforatedcnns: Acceleration through elimination of redundant convolutions. In *Advances in Neural Information Processing Systems*, pages 947–955, 2016.

[82] Shengjie Wang, Haoran Cai, Jeff Bilmes, and William Noble. Training compressed fully-connected networks with a density-diversity penalty. 2016.

[83] Arash Ardakani, Carlo Condo, and Warren J Gross. Sparsely-connected neural networks: Towards efficient vlsi implementation of deep neural networks. *arXiv preprint arXiv:1611.01427*, 2016.

[84] Mohammad Javad Shafiee, Parthipan Siva, and Alexander Wong. Stochasticnet: Forming deep neural networks via stochastic connectivity. *IEEE Access*, 4:1915–1924, 2016.

[85] Yani Ioannou, Duncan Robertson, Roberto Cipolla, and Antonio Criminisi. Deep roots: Improving cnn efficiency with hierarchical filter groups. *arXiv preprint arXiv:1605.06489*, 2016.

[86] Hao Zhou, Jose M Alvarez, and Fatih Porikli. Less is more: Towards compact cnns. In *European Conference on Computer Vision*, pages 662–677. Springer, 2016.

[87] Xuanyi Dong, Junshi Huang, Yi Yang, and Shuicheng Yan. More is less: A more complicated network with less inference complexity. *arXiv preprint arXiv:1703.08651*, 2017.

[88] Maxwell D Collins and Pushmeet Kohli. Memory bounded deep convolutional networks. *arXiv preprint arXiv:1412.1442*, 2014.

[89] Hessam Bagherinezhad, Mohammad Rastegari, and Ali Farhadi. Lcnn: Lookup-based convolutional neural network. *arXiv preprint arXiv:1611.06473*, 2016.

[90] Felix Juefei-Xu, Vishnu Naresh Boddeti, and Marios Savvides. Local binary convolutional neural networks. *arXiv preprint arXiv:1608.06049*, 2016.

[91] Kaiming He and Jian Sun. Convolutional neural networks at constrained time cost. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5353–5360, 2015.

[92] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[93] Chunpeng Wu, Wei Wen, Tariq Afzal, Yongmei Zhang, Yiran Chen, and Hai Li. A compact dnn: Approaching googlenet-level accuracy of classification and domain adaptation. *arXiv preprint arXiv:1703.04071*, 2017.

[94] Nadav Cohen, Or Sharir, and Amnon Shashua. Deep simnets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4782–4791, 2016.

[95] Michael Mathieu, Mikael Henaff, and Yann LeCun. Fast training of convolutional networks through ffts. *arXiv preprint arXiv:1312.5851*, 2013.

[96] Nicolas Vasilache, Jeff Johnson, Michael Mathieu, Soumith Chintala, Serkan Piantino, and Yann LeCun. Fast convolutional nets with fbfft: A gpu performance evaluation. *arXiv preprint arXiv:1412.7580*, 2014.

[97] Song Han, Xingyu Liu, Huizi Mao, Jing Pu, Ardavan Pedram, Mark A Horowitz, and William J Dally. Eie: efficient inference engine on compressed deep neural network. In *Proceedings of the 43rd International Symposium on Computer Architecture*, pages 243–254. IEEE Press, 2016.

[98] Renzo Andri, Lukas Cavigelli, Davide Rossi, and Luca Benini. Yodann: An architecture for ultra-low power binary-weight cnn acceleration. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2017.

[99] Aleksandar Zlateski, Kisuk Lee, and H Sebastian Seung. Znn–a fast and scalable algorithm for training 3d convolutional networks on multi-core and many-core shared memory machines. In *Parallel and Distributed Processing Symposium, 2016 IEEE International*, pages 801–811. IEEE, 2016.

[100] Yaman Umuroglu, Nicholas J Fraser, Giulio Gambardella, Michaela Blott, Philip Leong, Magnus Jahre, and Kees Vissers. Finn: A framework for fast, scalable binarized neural network inference. In *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pages 65–74. ACM, 2017.

Class		Result
Transfer	[1]	Distilling: MNIST, ASR,JFT dataset,Specialist Models,generalist model
	[2]	FitNets: CIFAR-10(10c+pb,13gs+36c+1%pl),-100,SVHN,MNIST,AFLW
	[3]	FOL: SST2,MR,CR,NER task,CNN+pb,RNN+pb
	[4]	Do?: TIMIT,CIFAR-10,student+ps,much faster
	[5]	Yes! they do.
	[6]	RNN2CNN: Wall Street Journal (WSJ),3.93 WER-4.54 WER
	[7]	FaceModel: 51.6c+90is+pb
Pruning	[8]	OBD: MNIST(-30%p)
	[9]	OBS: MONK's -62%p
	[10]	Predict: predict more than 95%p
	[11]	Both: AlexNet(9c+nl),VGG-16(13c+nl)
	[12]	DeepC: AlexNet(35c+nl),VGG-16(49c+nl),cgmg(3 4ls+3 7ee)
	[13]	DSD: CNN,RNN,LSTM,VGG-16(4.3%pb),ResNet-50(1.1%pb),DeepSpeech(2%pb)
	[14]	Data-free: MNIST-nn(-85%p),AlexNet(-35%p+ps)
	[15]	DNS: LeNet-5(108c+nl), AlexNet(17.7c+nl),less epochs
	[16]	GuidedP: AlexNet(3.1 7.3s)
	[17]	for-transf: Caltech-UCSD Birds 200-2011,Oxford Flowers
	[18]	Prune-filter: VGG-16(-34%c),ResNet-110(-38%c)+ps on CIFAR10
	[19]	Struct-Prune: CIFAR10,MNIST(-60%p in a layer)
	[20]	NoiseOut: LeNet5(-95%p+nl)
	[21]	Energy-Aware: AlexNet(3.7ec+1%pl),GoogLeNet(1.6%ec+1%pl)
	[22]	Bayesian: DC,DNS,SWS,LeNet5(108c+8gs+3ec),VGG16(51gs),VGG16(95c+ps)
	[23]	Uncertainty: comparable performance to dropout on MNIST classification
	[24]	DivNet: superior to random pruning, importance pruning
	[25]	Entropy-based: VGG16(3.3s+16.64c),ResNet50(1.54s+1.47c) +1%top5-pl
Quantization	[26]	fixed-qcnn: MNIST,TIMIT,ternary weight+ps
	[27]	fixed-qcnn: CIFAR10(-20%c+pb)
	[28]	BinaryConnect: MNIST,CIFAR10,SVHN +ps, binary w during fw and bp
	[29]	Binarized-NN: MNIST(7gs),CIFAR10,SVHN +ps, binary w and a at run and bp
	[30]	Quantized-NN: MNIST(7gs),CIFAR10,SVHN +ps,1bit w+2bit a–AlexNet p:51%
	[31]	Xnor-net: 58co+32c,ImageNet,binary filters and input
	[32]	Bitwise-NN: MNIST+ps
	[33]	VQ-nn: ImageNet,ZF-net(16 24c+1%pl)
	[34]	fly: MNIST,CIFAR10,larger compression
	[35]	Dorefa-net: SVHN,ImageNet,AlexNet(1bit w+2bit a+6bit g+p:46.1%)
	[36]	TWN: MNIST,CIFAR10,ImageNet, +16 32c+ps
	[37]	TTQ: +16c,ResNet-32,44,56 on CIFAR10,AlexNet on ImageNet +pb3%
	[38]	INQ: ResNet-18(4-bit+pb)
	[39]	Q-CNN: ImageNet(4 6s+15 20c+1%pl)
	[40]	HashNet: MNIST,CONVEX,RECT, super to RER,LRD,NN,DK
	[41]	LSH-nn: use 5% multip +1%pl,MNIST,NORB,CONVEX,Rectangle
	[42]	FunHashNN: MNIST,CONVEX, super to HashNet and NN
	[43]	FreshNets: MNIST,CIFAR10-100,SVHN,super to LRD,HashNet,DropFilt,DropFreq
	[44]	few-multip: MNIST,CIFAR10,SVHN,+pb,binary w and q-represent during bp
	[45]	BMNN-EBP: MNIST+ps
	[46]	improve-cpu: 3s,HMM/NN(10s)
	[47]	w-sharing: LeNet300-100(64c+ps),LeNet5(162c+ps),ResNet on CIFAR10(45c)
	[48]	ECSQ: 51.25,22.17,40.65c for LeNet,ResNet and AlexNet+ps
	[49]	Tensorizing-NN: VGG(7c, fc-200000c),CIFAR10,ImageNet
	[50]	Limited: train 16bit NN+nl
	[51]	Finite Precision Error Analysis of Neural Network Hardware Implementations
	[52]	HWGQ-Net: AlexNet,ResNet,GoogLeNet and VGG-Net(1bit w+ 2bit a+ps)
	[53]	DeepQ: UCF101,ActivityNet,CUB-200-2011+pb

Class		Result
Decomposition and Low-Rank	[54]	MobileNets: VGG16(32c+27s+ps),AlexNet(45c+9.4s+4%pb),face,detection
	[55]	flattened: 2s+significant-c+pb,MNIST,CIFAR10-100
	[56]	Learning Separable Filters
	[57]	LRD: scene text character recognition,cnn(2.5s+nl,4.5s+1%pl)
	[58]	Biclustering: 2cs,gs+1%pl
	[59]	CP-decomposition: 8.5cs+1%pl,AlexNet(4s+1%pl-top5)
	[60]	app-nonlinear: ImageNet(4s+0.9%pl-top5),super AlexNet and SPP-net
	[61]	Kronecker: SVHN,scene text,ImageNet(10c+1%pl)
	[62]	SVD: -80%p+nl
	[63]	Circulant: CIFAR10(4c+1.2s+1%pl),ImageNet,
	[64]	low-rank regul: CIFAR10+pb,AlexNet,NIN,VGG(2s+ps)
	[65]	Fried-CNN: MNIST(11c),ImageNet
	[66]	low-rank filter: VGG11(-41%compute-76%p+ps),CIFAR(-46%comp-55%p)
	[67]	Factorized-CNN: GoogLeNet+3.4s+pb
	[68]	Tucker-decomposition: AlexNet(5.46c+2.67s),GoogLeNet(1.28c+2.06s)
	[69]	BTD: VGG16(6.6s+1%pl-top5)
	[70]	Structured Transforms: MNIST(3.5c+ps),super to RER,LRD,NN,DK,HashNet
	[71]	intra-channel: VGG,ResNet-50,ResNet-101+42s,4.5s,6.5s
	[72]	matrix-f: LVCSR tasks-30% 50%p
	[73]	Subband Decomposition: DNN+17c+stable learning
Sparse	[74]	SCNN: ImageNet(-90%p+1%pl)
	[75]	Group Sparse: DIGITS dataset,MNIST,SSD,+ps
	[76]	power sparsity: MNIST(1000c+1%pl),CIFAR10,VGG16(7c+ps)
	[77]	Spatially-sparse: CASIA-OLHWDB1.1,MNIST,CIFAR10-100,+pb
	[78]	Shakeout: MNIST,CIFAR-10,ImageNet,superior to Dropout
	[79]	sparse activity: MNIST+pb
	[80]	SSL: AlexNet(5.1cs,3.1gs),improve accuracy on CIFAR10
	[81]	PerforatedCNNs: CIFAR10,ImageNet, AlexNet,VGG16,+2 4s
	[82]	Density-Diversity: LeNet300-100,LeNet5,MNIST,TIMIT
	[83]	Sparsely-connect: MNIST,CIFAR10,SVHN,-90%p+pb
	[84]	StochasticNet: CIFAR-10,MNIST,SVHN,STL-10,2c+ps
	[85]	Deep Roots: ImageNet,ResNet50(-40%p-45%flop+31%cs),GoogLeNet(-7%p+16%gs)
	[86]	Less is more: LeNet,CIFAR10,AlexNet,VGG,only 30%neurons in fc+nl
	[87]	More is less: CIFAR10-100,ImageNet,32%s+ps
	[88]	Memory Bounded: MNIST,CIFAR10,ImageNet,AlexNet(4c+ps)
Design	[89]	LCNN: ImageNet,AlexNet(3.2s+p:55.1%top1,37.6s+p:44.3%top1), few-shot learning
	[90]	LBCNN: 9 169c,MNIST,SVHN,CIFAR10,ImageNet
	[91]	Constrained Time: ImageNet,AlexNet(20%s)
	[92]	SqueezeNet: AlexNet(50c+ps, 510c)
	[93]	Conv-M: DNN(4.1M=59%GoogN+GoogLeNet p and DA)
	[94]	Deep SimNets: CIFAR10-100,SVHN,+2s+pb
	[95]	FFTs: fast training
	[96]	fbfft: CNN(1.5gs)
	[97]	EIE: 189cs,13gs,24000ee to CPU,3400ee to GPU
	[98]	YodaNN: 61.2TOp/s/W
	[99]	ZNN: 90cs
	[100]	FINN: FPGA accelerators, MNIST,CIFAR10,SVHN,fastest classification rates