ВЫСШАЯ ШКОЛА ЭКОНОМИКИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

Deep neural networks performance optimization in image recognition

Rassadin A. G., Savchenko A. V.

ITNT-2017, Samara

Outline

1. Problem discussion and formulation
2. Overview of performance optimization approaches: categorization and survey
3. Evaluation of existing optimization methods with the respect to the real task
4. Distillation the knowledge as multi-objective optimization

Problem Discussion

The success of deep convolutional neural networks (CNN) has started from the paper *Krizhevsky et al.* [5]. Contemporary CNN architectures are much more accurate when compared with original AlexNet. However, their runtime complexity becomes insufficient for application in several practical tasks, especially with implementation on mobile platforms. Hence, the performance optimization of deep CNN is now considered as one of the most important studies in deep learning.

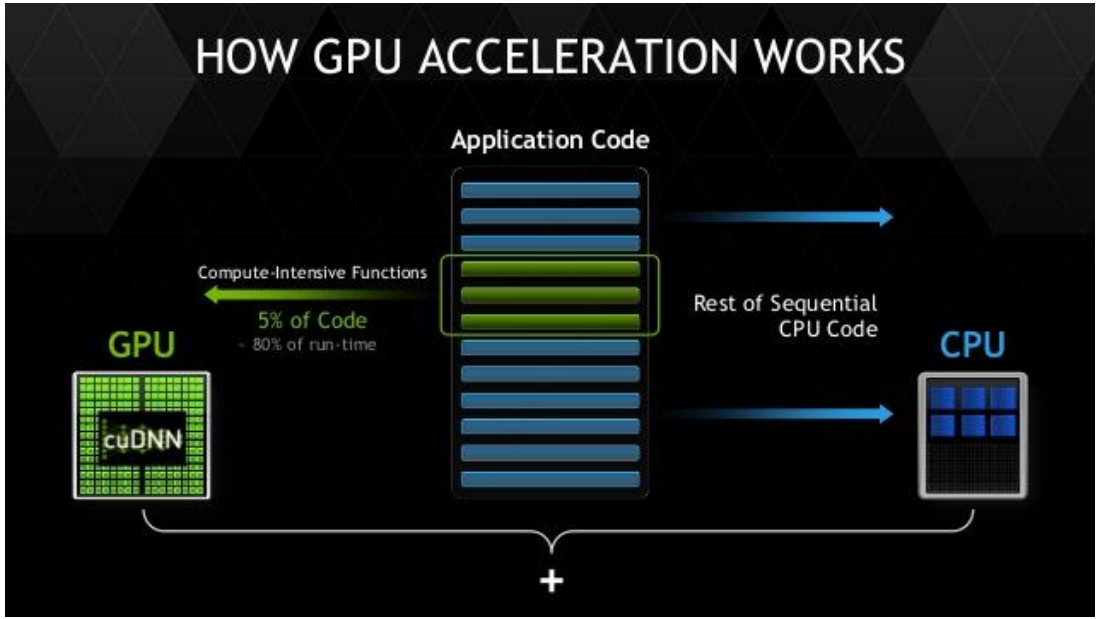


image credits: https://image.slidesharecdn.com/gtc-express-deep-learning-with-cudnn-webinar-150709172741-lva1-app6891/95/gpu-accelerated-deep-learning-for-cudnn-v2-4-638.jpg?cb=1436463102

Problem Formulation

Review and evaluate methods for performance optimization of deep neural networks in application to real image recognition task, namely emotion recognition from images.

We are fixing or don't review:

- hardware / driver optimization

  see, for example, special-purpose processing and memory units (*Google TPU, Nervana Engine, Movidius VPU, SnapDragon 820* etc.)
- optimization-precision trade-offs

  vDNN, *FP16*, *INT8*
- general framework optimizations

Classification of performance optimization methods

By the accuracy loss:

- *lossless*;
- optimization with accuracy loss;
- optimization-accuracy trade-off.

By the approach type:

- architectural;
- operational;
- computational;
- hardware.

By the restrictions:

- architecture-dependent;
- architecture-independent.

By the optimization type:

- speed;
- memory consumption;
- energy consumption.

By the implementation:

- runtime implementation;
- two-step (training -> optimization);
- sequential (training -> optimization -> re-training).

Optimization building block:

- everything
- convolutional layers
- FC layers

Advanced performance optimization

- **Pruning**

  Han et al. 2016 [9], *Molchanov et al. 2016* [11]

- **Distillation The Knowledge**

  Hinton et al. 2014 [12], *Romero et al. 2014* [13]

- **Weights Hashing / Quantization**

  Chen et al. 2015 [14], *Han et al. 2016* [9]

- **Tensor Decompositions**: TT, CP, Tucker, …

  Lebedev et al. 2015 [15], *Kim et al. 2015* [16], *Novikov et al. 2015* [17], *Garipov et al. 2016* [18]

- **Binarization**

  Courbariaux / Hubara et al. 2016 [19], *Rastegari et al. 2016* [20], *Merolla et al. 2016* [21], *Hou et al. 2016* [32]

- **Architectural tricks** (*simple* but yet *powerful* architecture)

  Hong et al. 2016 [24], *Iandola et al. 2016* [22], *Teerapittayanon et al. 2016* [25]

Comparisons. Scores

	Memory reduction while training	Memory reduction while inference	Inference speedup	Accuracy gain	Baseline model	Dataset
FitNets	-	36	13.36	-1.17	Maxout	CIFAR-10
HashedNets	-	64	?	0,24	*same-size*	MNIST
Deep Compression	-	49 (~4)	?	0.33	VGG-16	ImageNet
CP-Decomposition	-	12	4.5	-1	AlexNet	ImageNet
TensorNet	?	80	?	-1.1	*simple*	CIFAR-10
BinaryNet	~32 (theoretical)		3.4~23	1.53	Maxout	CIFAR-10
Binary-Weight-Network	-	67	58 (CPU)	-8.5	ResNet-18	ImageNet
XNOR-Net	-			-18.1		
SqueezeNet	?	50	1.	0.3	AlexNet	ImageNet
Tiny Darknet	?	60	2.9	1.5		
BranchyNet	-	-	1.9	-1,53	ResNet-110	CIFAR-10

Comparisons. Summary

	Train - Memory	Train - Speed	Inference - Memory	Inference - Speed
HashedNets	**?**	**?**	+	**?**
CP-Decomposition	-	-	+	+
BinaryNet	-	-	+	+
Binary-Weight-Network	-	-	+	+
XNOR-Net	-	+	+	+

Experiments. Formulation

Baseline - visual emotion recognition, Levi et al. 2015 [27]. Unfortunately, original EmotiW 2015 [28] dataset not available and Radboud Faces Database [33] cropped by face was used instead for training and evaluation:

- ±45° of rotation, the same balanced and independent train / test sets for all experiments.

Common experiments settings:

- **SqueezeNet, CP-decomposition**, **HashedNets**, **BWN**, **XNOR-Net**
- author's code (guarantees exact implementation and results reproducibility);
- SGD with momentum equal to 0.9, fixed learning rate equal to 0.001;
- no data augmentation except channel-wise z-score for **SqueezeNet**.

Evaluation settings:

- accuracy metric: test accuracy;
- speedup metrics:
  - epoch time for single forward pass and subsequent gradient update on GPU for mini-batch in one random sample, averaged over 1000 runs
  - GPU inference time for single random sample, averaged over 1000 runs

Experiments. CP-decomposition

The setting:

- SquezeNet 1.1;
- decomposition of last two convolutional layers to rank 192.

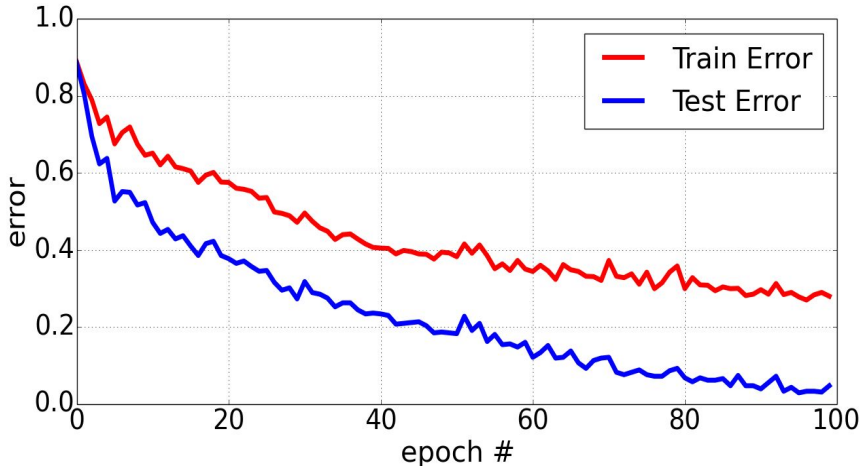
Conclusions:

- suitable only for fully-convolutional architectures;
- good trade-off between compression rate and accuracy loss;
- iterative optimization process can overcome accuracy loss;
- 1.5 time slower possibly due to replacement of the single large convolutional layer to four sequentially connected small layers.

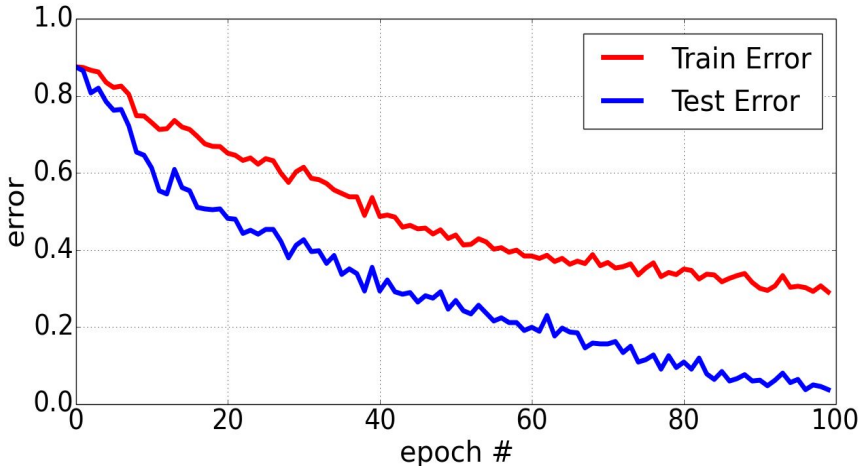
Experiments. HashedNets

The setting:

- compression rate equal to 0.125



Baseline **VGG-S** model
test accuracy: 97.13%



HashedNet model
test accuracy: 96.31%

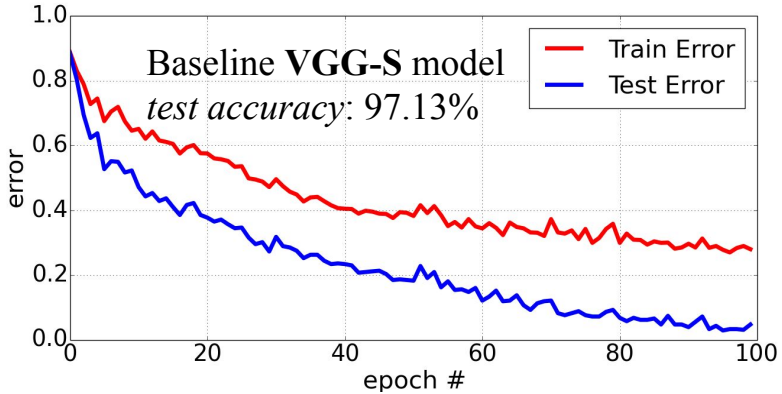
Conclusions:

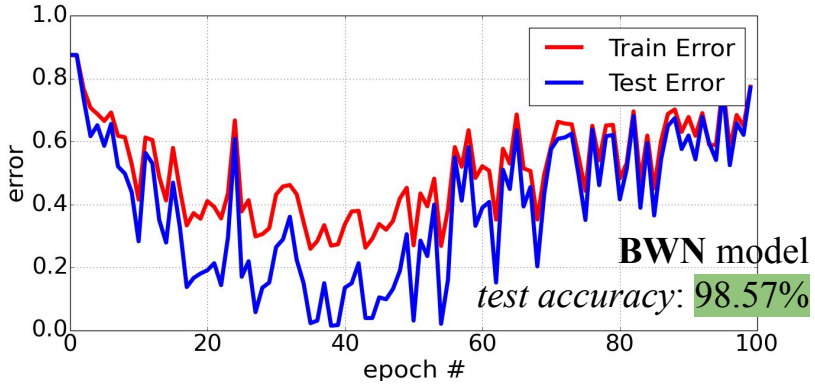
- ultimate compression;
- grate slowdown possibly because of usage `libhashnn`.

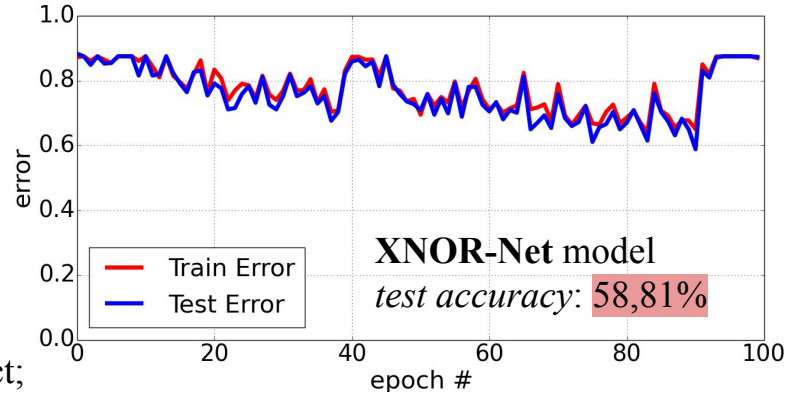
Experiments. XNOR-Nets

The setting:

- XNOR-Net layers (ordering) configuration according the paper (*conv-bn-activation* replaced with the *bn-activation-conv*)







Conclusions:

- BWN converges 2~4 times faster - strong regularization effect;
- XNOR-Net practically useless with current hardware / software stack.

Experiments. Scores

	Epoch time, ms	Inference time, ms	Accuracy, %	Model size, MB
CP-Decomposition	N/A	7.74	87.5	-23.5%
***Baseline* (Caffe)**	**22.94**	**4.94**	**89.14**	**2.8**
HashedNets	294.8	158.2	96.31	-81.64%
Binary-Weight-Network	83.8	33.5	98.57	0%
XNOR-Net	84.3	34.2	58.81	-2.4%
XNOR-Net w/o weights b-n	43.4	34.1	88.32	-2.4%
***Baseline* (Torch)**	**43.7**	**33.4**	**97.13**	**372.2**

Experiments. Summary

	Training speedup	Memory reduction while inference	Inference speedup	Parameters reduction	Accuracy loss
HashedNets	-	?	4.7 times slower	81.64%	slight
CP-Decomposition	N/A	0%	-	23.5%	slight
Binary-Weight-Network	4x (by epochs number)	0%	-	0%	sloght
XNOR-Net	-	2.4%	-	2.4%	huge

Subtotal:

It is clear, that despite the large number of papers devoted to improvements in performance of CNNs, their application in different tasks is very difficult. However, our results demonstrate the advantages of hashing techniques and tensor decomposition. Unfortunately, existing hardware and driver stack unable to exploit the full power of binary calculations.

Problem Rethinking

Say we want to build a powerful model for some task, e.g. *emotion recognition from single image*. However, we also have:

- lack of computational resources.

On the other hand, there can be available pre-trained model though still out of our computational capabilities to use it.

Moreover, we can not reproduce it with more compact architecture, say, due to

- lack of original or suitable dataset.

Possible pipeline will consist of:

- online optimization: transfer the knowledge as a *multi-objective* optimization
- offline optimization: pre-trained model optimization

Experiments. Formulation

Tramèr et al. shows the possibility to reproduce existing model having only *labels* it predicts. Having also *scoring* for each sample (*softmax* output in our case) such reverse engineering becomes more accurate.

What if we can distill the knowledge to the lightweight one learning from *labels* and *scoring* of the heavy baseline?

Experiments settings:

- pre-trained on EmotiW 2015 [28] Levi et al. 2015 [27] model and similar one trained on RaFD [33]
- Pubfig83 [31] as *training* dataset
- RaFD [33] as *evaluation* dataset

Distillation the knowledge as a multi-objective optimization

	VGG(EmotiW) -> SqueezeNet(Pubfig83) -> SqueezeNet(RaFD)		VGG(RaFD) -> SqueezeNet(Pubfig83) -> SqueezeNet(RaFD)	
	labels only	***labels + softmax***	***labels only***	***labels + softmax***
baseline accuracy	>40%		>80%	
distilled accuracy	66.5%	73%	75.5%	77%
new data accuracy	12.3%	18.6 (23.8)%	40.9%	46.9%

Subtotal:

- **labels + softmax** accuracy always better than **labels only**
- **weaker** initial model tends to **weaker** result on new data
- straightforward implementation still suffer in accuracy

Conclusions

- reviewed several modern approaches to optimize performance of deep neural networks
- emphasized the obvious trends in this field, namely, efficient tensor decomposition techniques, lower precision calculations and more accurate network binarization
- performed an evaluation of the state-of-the-art techniques in application to visual emotion recognition based on facial expressions
- studied a possibility to build a maximal effective pipeline with limited both data and computational resources

The main direction for the further research will be concentrated on combining of the most successful reviewed techniques for implementation on mobile platforms.

References

[1] Savchenko, A.V. Search Techniques in Intelligent Classification Systems. / A.V. Savchenko // Springer International Publishing – 2016

[2] Savchenko, A.V. Maximum-Likelihood Approximate Nearest Neighbor Method in Real-time Image Recognition / A.V. Savchenko // Pattern Recognition – 2017 – Vol. 61 – p. 459-469

[3] Choi, K. Automatic Tagging using Deep Convolutional Neural Networks / K. Choi, G. Fazekas, M. Sandler // arXiv preprint arXiv:1606.00298 - 2016

[4] Choi, K. Convolutional Recurrent Neural Networks for Music Classification / K. Choi, G. Fazekas, M. Sandler, K. Cho // arXiv preprint arXiv:1609.04243 - 2016

[5] Krizhevsky A. ImageNet Classification with Deep Convolutional Neural Networks / A. Krizhevsky, I. Sutskever, G. E. Hinton // Advances in Neural Information Processing Systems – 2012 – Vol. 25 – p. 1106-1114

[6] Lin, M. Network In Network / M. Lin, Q. Chen, S. Yan // arXiv preprint arXiv:1312.4400 - 2013

[7] Szegedy, C. Going Deeper with Convolutions / C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich // arXiv preprint arXiv:1409.4842 - 2014

[8] He, K. Deep Residual Learning for Image Recognition / K. He, X. Zhang, S. Ren, J. Sun // arXiv preprint arXiv:1512.03385 - 2015

[9] Han, S. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding / S. Han, H. Mao, W. J. Dally // arXiv preprint arXiv:1510.00149 - 2015

[10] ARM Community [Electronic resource]: https://community.arm.com/iot/embedded/f/discussions/166/fast-deep-learning-inference-on-qualcomm-snapdragon-achieving-super-fast-inference-times-by-tweaking-models-and-codes

[11] Molchanov, P. Pruning Convolutional Neural Networks for Resource Efficient Transfer Learning / P. Molchanov, S. Tyree, T. Karras, T. Aila, J. Kautz // arXiv preprint arXiv:1611.06440 - 2016

[12] Hinton, G. Distilling the Knowledge in a Neural Network / G. Hinton, O. Vinyals, J. Dean // arXiv preprint arXiv:1503.02531 - 2015

References

[13] Romero, A. FitNets: Hints for Thin Deep Nets / A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, Y. Bengio // arXiv preprint arXiv:1412.6550 - 2014

[14] Chen, W. Compressing Neural Networks with the Hashing Trick / W. Chen, J. T. Wilson, S. Tyree, K. Q. Weinberger, Y. Chen // arXiv preprint arXiv: 1504.04788 - 2015

[15] Lebedev, V. Speeding-up Convolutional Neural Networks Using Fine-tuned CP-Decomposition / V. Lebedev, Y. Ganin, M. Rakhuba, I. Oseledets, V. Lempitsky // arXiv preprint arXiv:1412.6553 - 2014

[16] Kim, Y.-D. Compression of Deep Convolutional Neural Networks for Fast and Low Power Mobile Applications / Y.-D. Kim, E. Park, S. Yoo, T. Choi, L. Yang, D. Shin // arXiv preprint arXiv:1511.06530 - 2015

[17] Novikov, A. Tensorizing Neural Networks / A. Novikov, D. Podoprikhin, A. Osokin, D. Vetrov // arXiv preprint arXiv:1509.06569 - 2015

[18] Garipov, T. Ultimate tensorization: compressing convolutional and FC layers alike / T. Garipov, D. Podoprikhin, A. Novikov, D. Vetrov // arXiv preprint arXiv:1611.03214 – 2016

[19] Courbariaux, M. Binarized Neural Networks: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1 / M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, Y. Bengio // arXiv preprint arXiv:1602.02830 - 2016

[20] Rastegari, M. XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks / M. Rastegari, V. Ordonez, J. Redmon, A. Farhadi // arXiv preprint arXiv:1603.05279 - 2016

[21] Merolla, P. Deep neural networks are robust to weight binarization and other non-linear distortions / P. Merolla, R. Appuswamy, J. Arthur, S. K. Esser, D. Modha // arXiv preprint arXiv:1606.01981 - 2016

[22] Iandola, F. N. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size / F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, K. Keutzer // arXiv preprint arXiv:1602.07360 - 2016

[23] The Darknet project web site [Electronic resource]: http://pjreddie.com/darknet/tiny-darknet/

[24] Hong, S. PVANet: Lightweight Deep Neural Networks for Real-time Object Detection / S. Hong, B. Roh, K.-H. Kim, Y. Cheon, M. Park // arXiv preprint arXiv:1608.08021 - 2016

References

[25] Teerapittayanon, S. BranchyNet: Fast Inference via Early Exiting from Deep Neural Networks / S. Teerapittayanon, B. McDanel, H. T. Kung // ICPR – 2016

[26] Langner, O. Presentation and validation of the Radboud Faces Database / O. Langner, R. Dotsch, G. Bijlstra, D. H. J. Wigboldus, S.T. Hawk, A. van Knippenberg // Cognition & Emotion - 2010 - 24(8), 1377—1388. DOI: 10.1080/02699930903485076

[27] Levi, G. Emotion Recognition in the Wild via Convolutional Neural Networks and Mapped Binary Patterns / G. Levi, T. Hassner // ICMI '15 Proceedings of the 2015 ACM on International Conference on Multimodal Interaction – 2015 – p. 503-510

[28] The Third Emotion Recognition in the Wild Challenge (EmotiW 2015) [Electronic resource]: https://cs.anu.edu.au/few/emotiw2015.html

[29] Emotion Recognition in the Wild via Convolutional Neural Networks and Mapped Binary Patterns web site [Electronic resource]: http://www.openu.ac.il/home/hassner/projects/cnn_emotions/

[30] Tramèr, F. Stealing Machine Learning Models via Prediction APIs / F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, T. Ristenpart // arXiv preprint arXiv:1609.02943 - 2016

[31] Pinto, N. Scaling Up Biologically-Inspired Computer Vision: A Case Study in Unconstrained Face Recognition on Facebook / N. Pinto, Z. Stone, T. Zickler, D. D. Cox // Proc. Workshop on Biologically Consistent Vision (in conjunction with CVPR) - 2011

[32] Hou, L. Loss-aware Binarization of Deep Networks / L. Hou, Q. Yao, J. T. Kwok // arXiv preprint arXiv:1611.01600 - 2016

[33] Radboud Faces Database website [Electronic resource]: http://www.socsci.ru.nl:8180/RaFD2/RaFD