Training compact deep learning models for video classification using circulant matrices

The 2nd YouTube-8M Video Understanding Challenge

Alexandre Araujo[1,2], Benjamin Negrevergne[1], Yann Chevaleyre[1] and Jamal Atif[1]

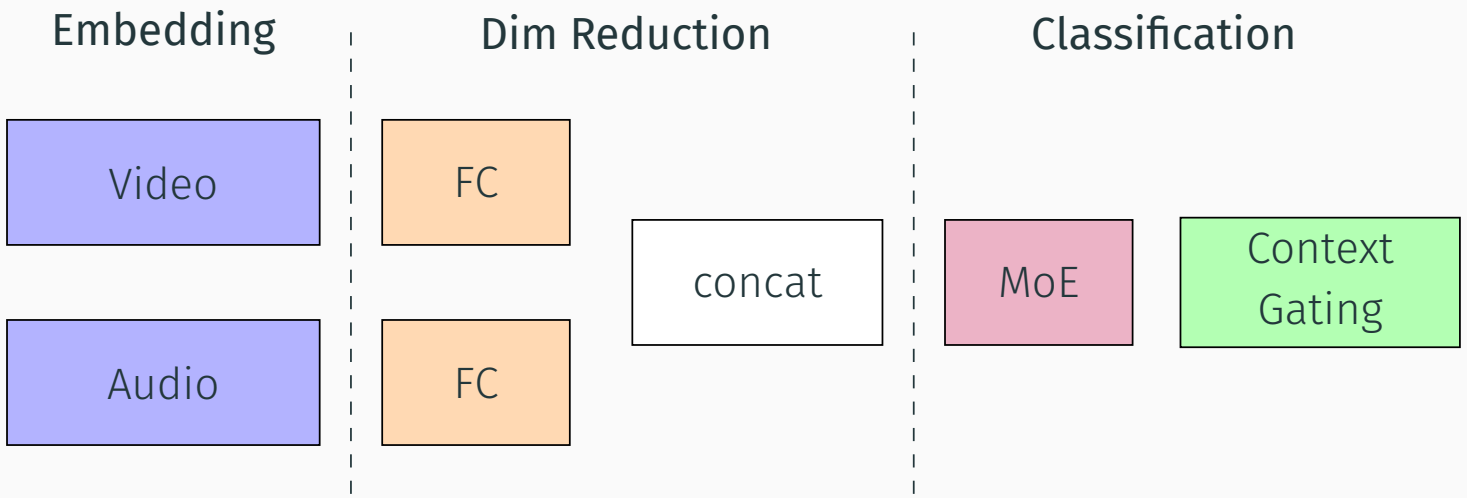
[1]Université Paris-Dauphine, PSL Research University, CNRS, LAMSADE, 75016 Paris, France

[2]Wavestone, Paris, France

Large models for video classification

Video classification requires large models

- model architecture proposed by Miech *et al.* (2017)



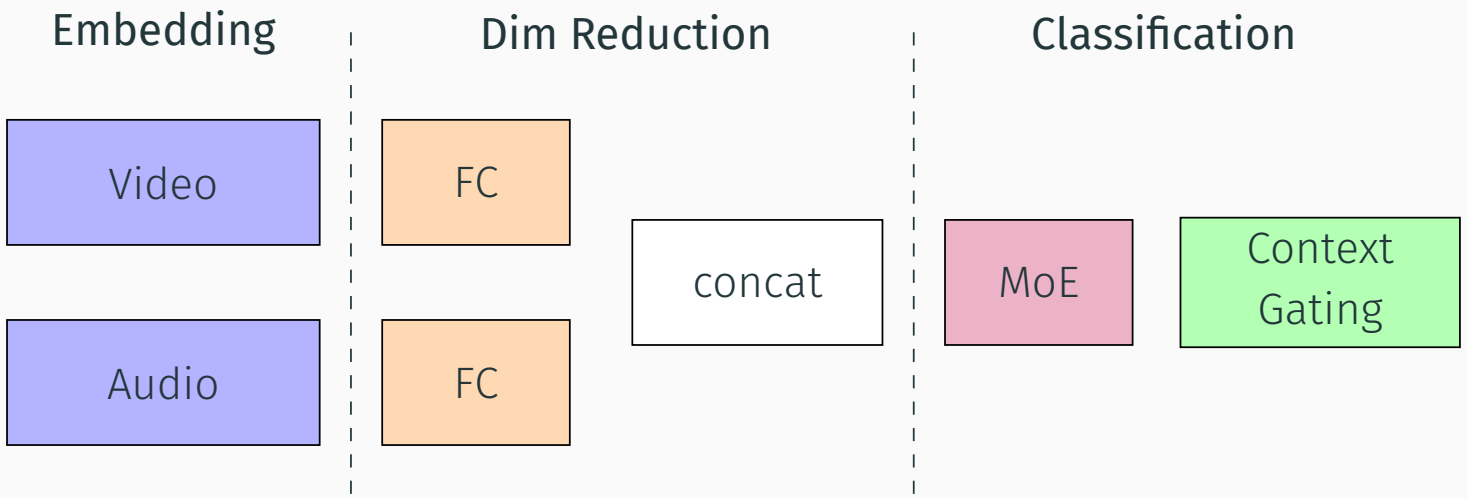
$\sim$ 330M parameters ($\sim$ 1.3 Go)

- winning solutions are large ensemble
  (1st: 7 models, 2nd: 74 models, 3rd: 57 models)

Large models for video classification

Video classification requires large models

- model architecture proposed by Miech *et al.* (2017)



$\sim$ 330M parameters ($\sim$ 1.3 Go)

- winning solutions are large ensemble
  (1st: 7 models, 2nd: 74 models, 3rd: 57 models)

This year, the challenge is focused on learning video representation under budget constraints.

Model compression for Deep Learning

Model compression after training

- Model distillation Hinton *et al.* (2015)
- Pruning Dai *et al.* (2018); Han *et al.* (2016); Lin *et al.* (2017)
- Sparsity regularizer Collins & Kohli (2014); Dai *et al.* (2018); Liu *et al.* (2015)

Model compression for Deep Learning

Model compression after training

- Model distillation Hinton *et al.* (2015)
- Pruning Dai *et al.* (2018); Han *et al.* (2016); Lin *et al.* (2017)
- Sparsity regularizer Collins & Kohli (2014); Dai *et al.* (2018); Liu *et al.* (2015)

Is it possible to devise models which are compact by nature ?

Model compression for Deep Learning

Training of compact model

- constraining the weight representation
  - → floating variable with limited precision Gupta *et al.* (2015)
  - → quantization Courbariaux *et al.* (2015); Mellempudi *et al.* (2017); Rastegari *et al.* (2016))
  - → hashing techniques Chen *et al.* (2015))

Model compression for Deep Learning

Training of compact model

- constraining the weight representation
  - → floating variable with limited precision Gupta *et al.* (2015)
  - → quantization Courbariaux *et al.* (2015); Mellempudi *et al.* (2017); Rastegari *et al.* (2016))
  - → hashing techniques Chen *et al.* (2015))
- matrix factorization Denil *et al.* (2013); Jaderberg *et al.* (2014); Yu *et al.* (2017)
  - → use of low rank matrices and decomposition as weights matrices

Model compression for Deep Learning

Training of compact model

- constraining the weight representation
  - → floating variable with limited precision Gupta *et al.* (2015)
  - → quantization Courbariaux *et al.* (2015); Mellempudi *et al.* (2017); Rastegari *et al.* (2016))
  - → hashing techniques Chen *et al.* (2015))
- matrix factorization Denil *et al.* (2013); Jaderberg *et al.* (2014); Yu *et al.* (2017)
  - → use of low rank matrices and decomposition as weights matrices
- imposing structures on weight matrices
  - → circulant matrices Cheng *et al.* (2015); Sindhwani *et al.* (2015)
  - → vandermonde Sindhwani *et al.* (2015)
  - → fastfood transforms Yang *et al.* (2015)

Circulant matrices for Deep Learning

A n-by-n circulant matrix C is a matrix where each row is a cyclic right shift of the previous one as illustrated below.

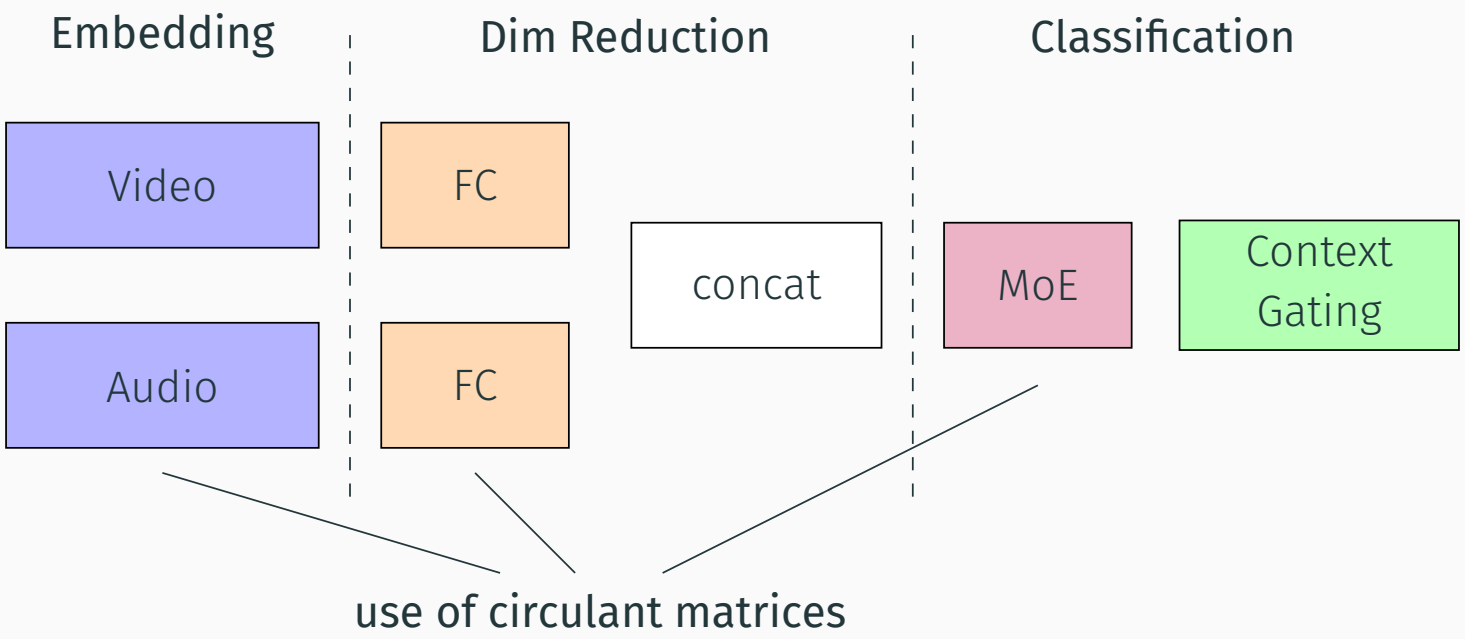
$$C = circ(c) = \begin{pmatrix} c_0 & c_{n-1} & c_{n-2} & \dots & c_1 \\ c_1 & c_0 & c_{n-1} & & c_2 \\ c_2 & c_1 & c_0 & & c_3 \\ \vdots & & & \ddots & \vdots \\ c_{n-1} & c_{n-2} & c_{n-3} & & c_0 \end{pmatrix}$$

Main advantages:

- The circulant matrix $C \in \mathbb{R}^{n \times n}$ can be **compactly represented in memory** using only n real values instead of n^2.
- Multiplying a circulant matrix C by a vector x **can be done efficiently in the Fourier domain**

Architecture used for the experiences

The network samples at random video and audio frames from the input. The sample goes through an embedding layer and is reduced with a Fully Connected layer. The results are then concatenated and classified with a Mixture-of-Experts and Context Gating layer.



Circulant matrices for Deep Learning

Base on the work of Müller-Quade *et al.* (1998); Schmid *et al.* (2000); Huhtanen & Perämäki (2015). Any n-by-n matrix A can be decomposed into the product of diagonal and circulant matrices as follows:

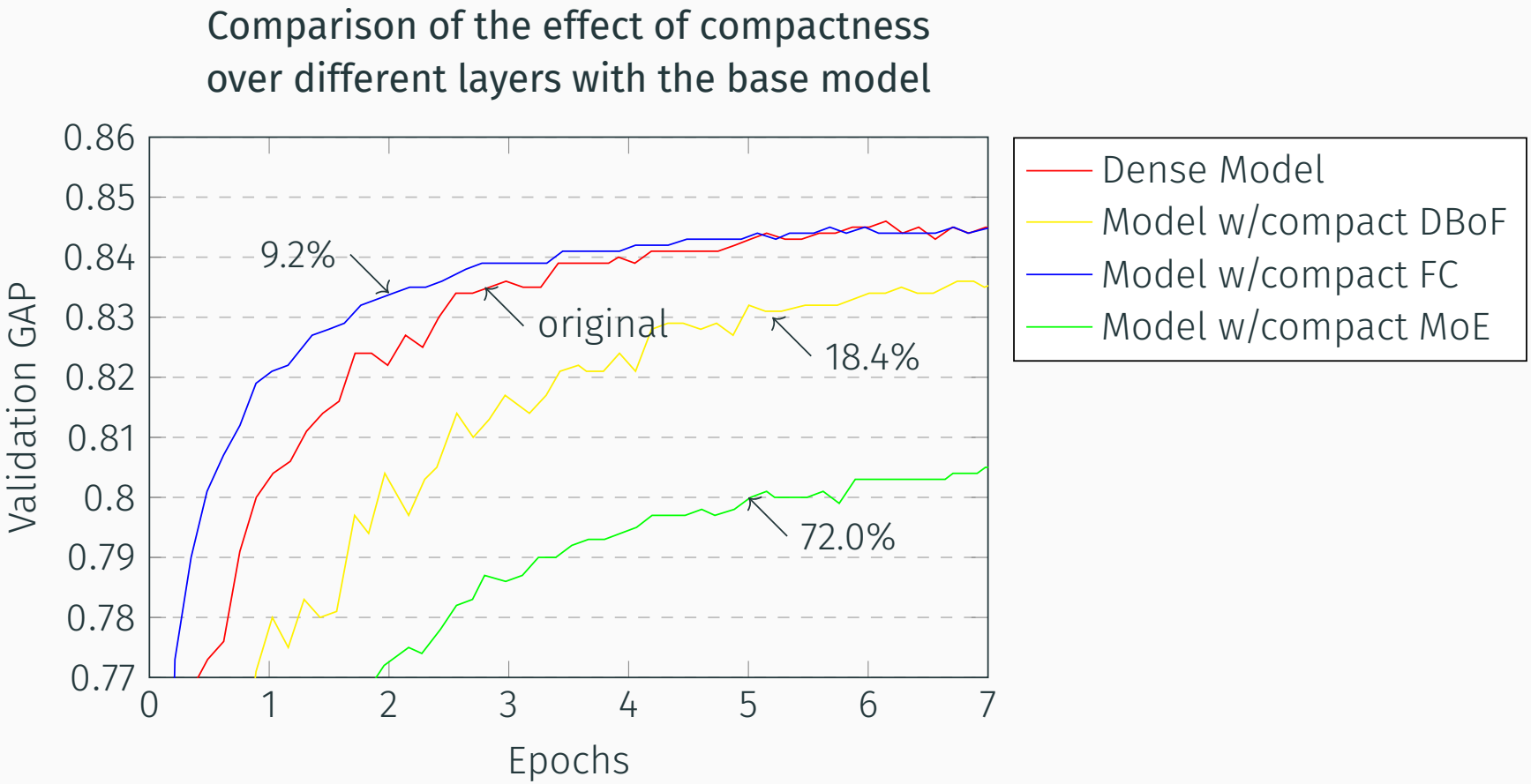
$$A = D^{(1)}C^{(1)}D^{(2)}C^{(2)}\dots D^{n}C^{n} = \prod_{i=1}^{n} D^{(i)}C^{(i)} \qquad \approx \prod_{i=1}^{k<n} D^{(i)}C^{(i)}$$

The fully connected layers are then represented as follows:

$$h(x) = \phi\left(\left[\prod_{i=1}^{k} D^{(i)}C^{(i)}\right]x + b\right)$$

where the parameters of each matrix $D^{(i)}$ and $C^{(i)}$ are trained using a gradient based optimization algorithm, and k defines the number of factors we choose for the training.

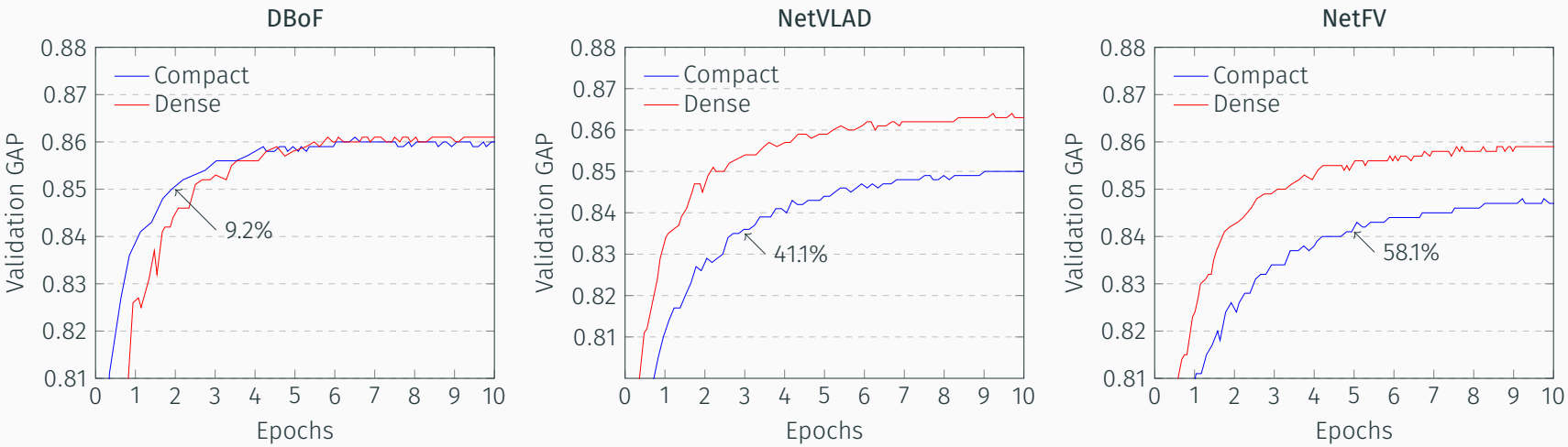
Effect of circulant matrices over different layers



Validation GAP according to the number of epochs for different compact models.

Effect of circulant matrices with different embeddings

The figures bellow show the validation GAP of compact and *Dense* fully connected layer with different embeddings according to the number of epochs.



Conclusion

- We propose the use of a matrix decomposition into diagonal and circulant matrices in Deep Learning settings
- We apply this decomposition on several layers with different embeddings
- we showed that this method allow a good compression rate without a big loss in accuracy.

Questions?

References

Chen, Wenlin, Wilson, James T., Tyree, Stephen, Weinberger, Kilian Q., & Chen, Yixin. 2015. Compressing Neural Networks with the Hashing Trick. *Pages 2285–2294 of: Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*. ICML'15. JMLR.org.

Cheng, Y., Yu, F. X., Feris, R. S., Kumar, S., Choudhary, A., & Chang, S. F. 2015 (Dec). An Exploration of Parameter Redundancy in Deep Networks with Circulant Projections. *Pages 2857–2865 of: 2015 IEEE International Conference on Computer Vision (ICCV)*.

Collins, Maxwell D., & Kohli, Pushmeet. 2014. Memory Bounded Deep Convolutional Networks. *CoRR*, **abs/1412.1442**.

References ii

Courbariaux, Matthieu, Bengio, Yoshua, & David, Jean-Pierre. 2015. BinaryConnect: Training Deep Neural Networks with Binary Weights During Propagations. *Pages 3123–3131 of: Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*. NIPS'15. Cambridge, MA, USA: MIT Press.

Dai, Bin, Zhu, Chen, Guo, Baining, & Wipf, David. 2018. Compressing Neural Networks using the Variational Information Bottleneck. *Pages 1143–1152 of:* Dy, Jennifer, & Krause, Andreas (eds), *Proceedings of the 35th International Conference on Machine Learning*. Proceedings of Machine Learning Research, vol. 80. Stockholmsmässan, Stockholm Sweden: PMLR.

References iii

Denil, Misha, Shakibi, Babak, Dinh, Laurent, Ranzato, Marc' Aurelio, & de Freitas, Nando. 2013. Predicting Parameters in Deep Learning. *Pages 2148–2156 of:* Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., & Weinberger, K. Q. (eds), *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc.

Gupta, Suyog, Agrawal, Ankur, Gopalakrishnan, Kailash, & Narayanan, Pritish. 2015. Deep Learning with Limited Numerical Precision. *Pages 1737–1746 of: Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*. ICML'15. JMLR.org.

Han, Song, Mao, Huizi, & Dally, William J. 2016. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. *International Conference on Learning Representations (ICLR)*.

References iv

Hinton, Geoffrey, Vinyals, Oriol, & Dean, Jeffrey. 2015. Distilling the Knowledge in a Neural Network. *In: NIPS Deep Learning and Representation Learning Workshop*.

Huhtanen, Marko, & Perämäki, Allan. 2015. Factoring Matrices into the Product of Circulant and Diagonal Matrices. *Journal of Fourier Analysis and Applications*, **21**(5), 1018–1033.

Jaderberg, Max, Vedaldi, Andrea, & Zisserman, Andrew. 2014. Speeding up Convolutional Neural Networks with Low Rank Expansions. *CoRR*, **abs/1405.3866**.

Lin, Ji, Rao, Yongming, Lu, Jiwen, & Zhou, Jie. 2017. Runtime Neural Pruning. *Pages 2181–2191 of:* Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., & Garnett, R. (eds), *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc.

References v

Liu, Baoyuan, Wang, Min, Foroosh, H., Tappen, M., & Penksy, M. 2015 (June). Sparse Convolutional Neural Networks. *Pages 806–814 of: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Mellempudi, Naveen, Kundu, Abhisek, Mudigere, Dheevatsa, Das, Dipankar, Kaul, Bharat, & Dubey, Pradeep. 2017. Ternary Neural Networks with Fine-Grained Quantization. *CoRR*, **abs/1705.01462**.

Miech, Antoine, Laptev, Ivan, & Sivic, Josef. 2017. Learnable pooling with Context Gating for video classification. *CoRR*, **abs/1706.06905**.

Müller-Quade, Jörn, Aagedal, Harald, Beth, Th, & Schmid, Michael. 1998. Algorithmic design of diffractive optical systems for information processing. *Physica D: Nonlinear Phenomena*, **120**(1-2), 196–205.

References vi

Rastegari, Mohammad, Ordonez, Vicente, Redmon, Joseph, & Farhadi, Ali. 2016. XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks. *In: ECCV*.

Schmid, Michael, Steinwandt, Rainer, Müller-Quade, Jörn, Rötteler, Martin, & Beth, Thomas. 2000. Decomposing a matrix into circulant and diagonal factors. *Linear Algebra and its Applications*, **306**(1-3), 131–143.

Sindhwani, Vikas, Sainath, Tara, & Kumar, Sanjiv. 2015. Structured Transforms for Small-Footprint Deep Learning. *Pages 3088–3096 of:* Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., & Garnett, R. (eds), *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc.

References vii

Yang, Z., Moczulski, M., Denil, M., d. Freitas, N., Smola, A., Song, L., & Wang, Z. 2015 (Dec). Deep Fried Convnets. *Pages 1476–1483 of: 2015 IEEE International Conference on Computer Vision (ICCV).*

Yu, X., Liu, T., Wang, X., & Tao, D. 2017 (July). On Compressing Deep Models by Low Rank and Sparse Decomposition. *Pages 67–76 of: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*