Les Cahiers du GERAD ISSN: 0711–2440

Deep LDA-pruned nets and their robustness

Q. Tian,
T. Arbel, J.J. Clark

G–2020–23–EIW04

April 2020

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2020
– Bibliothèque et Archives Canada, 2020

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2020
– Library and Archives Canada, 2020

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Deep LDA-pruned nets and their robustness

Qing Tian
Tal Arbel
James J. Clark

McGill Centre for Intelligent Machines, McGill University, Montréal (Québec), Canada, H3A 2A7

qtian@cim.mcgill.ca
arbel@cim.mcgill.ca
clark@cim.mcgill.ca

April 2020
Les Cahiers du GERAD
G–2020–23–EIW04

Abstract: *Deep neural networks usually have unnecessarily high complexities and possibly many features of low utility, especially for tasks that they are not designed for. In this extended abstract, we present our Deep-LDA-based pruning framework as a solution to such problems. In addition to accuracy-complexity analysis, we investigate our approach's potential in improving networks' robustness against adversarial attacks (e.g. FGSM and NewtonFool Attacks) and noises (e.g. Gaussian, Poisson, Speckle). Experimental results on CIFAR100, Adience, and LFWA illustrate our framework's efficacy. Through pruning, we can derive smaller, but accurate and more robust models suitable for particular tasks.*

1 Introduction

With increasing network depths comes more complexity, which reignited research into network pruning. Approaches that sparsify networks by setting weights to zero include [7, 23, 21, 15, 6, 11, 24]. Compared to individual weights based approaches, filter or neuron pruning has its advantages. Instead of setting zeros in weights matrices, filter pruning removes rows/columns/depths in weight/convolution matrices, leading to direct space and computation savings [26, 17, 19, 9, 20, 27]. However, few if any works have investigated pruning's influence on model robustness. Given the large input-output dimension ratio, input-output correlation could be spurious. In this long abstract, we present our deep LDA based pruning and also analyze its influence on model robustness against adversarial attacks and noises. Through pruning large networks of high memorization capability, our method aims to help over-parameterized nets forget about task-unrelated factors and derive a feature subspace that is more invariant and robust to irrelevant factors and noises.

2 Deep Fisher LDA pruning

In this section, we present our deep Linear Discriminant Analysis (LDA) based pruning approach that pays direct attention to final task utility and its holistic cross-layer dependency. We define and capture task utility by deep LDA and use it to guide the pruning process. The approach is summarized as Algorithm 1.

Algorithm 1: Deep LDA Pruning of NN

Input: basenet, acceptable accuracy t_{acc}
Result: task-desirable pruned models
Pre-train: SGD optimization with cross entropy loss and dropout.
while *accuracy* $\geq t_{acc}$
do
  Step 1 → Pruning
    1. Penultimate LDA Utility Unravelling
    2. Cross-Layer Utility Tracing by Deconv
    3. Pruning as Utility Thresholding
  Step 2 → Re-training
    Similar to the pre-training step.
    Save model if needed.
end

In the rest of the section, we focus on the pruning step. Given the penultimate activation matrix X, our aim is to abandon dimensions of X that possess low or even negative task utility. Inspired by [4, 2, 29, 12, 1], Fisher's LDA is adopted to quantify this utility. Our goal of pruning is to find:

$$W_{opt} = \arg\max_{W} \frac{\mid W^T \Sigma_b W \mid}{\mid W^T \Sigma_w W \mid} \tag{1}$$

where Σ_w, Σ_b, Σ_a are within-class, between-class, and total scatter matrices. Through solving a generalized function with a decorrelated assumption of top layer motifs ([28]), we know that W columns are standard basis vectors. It follows that W columns and some of the original neuron dimensions are aligned. To maximize the class separation during pruning, we can safely discard neurons with small between-class to within-class variance ratios.

After unravelling twisted threads of deep variances and selecting dimensions of high LDA utility, the next step is to trace the utility across all previous layers to guide pruning. Inspired by [8], deconvolution as in [30] is used to reverse an unknown filter's effect and recover corrupted sources. One unit procedure is composed of unpooling, nonlinear rectification, and reversed convolution:

$$U_i = F_i^T Z_i \tag{2}$$

where i indicates the layer, Z_i is converted from feature maps, U_i is reconstructed contributing sources to final utility, F_i^T represents transposed convolution. Figure 1 provides a high level view of deep LDA cross-layer utility tracing. With all neurons'/filters' utility for final discriminability known, pruning simply becomes discarding structures that are less useful to final classification (colored white). Through pruning modular structures like Inception nets, the proposed approach determines how many filters, and of what types, are appropriate in a given layer. The threshold on utility is related to pruning rates. After pruning, retraining with surviving parameters is needed. Since our pruning selects filter dimensions according to task demands, the generated pruned models are more invariant to task-unrelated factors.

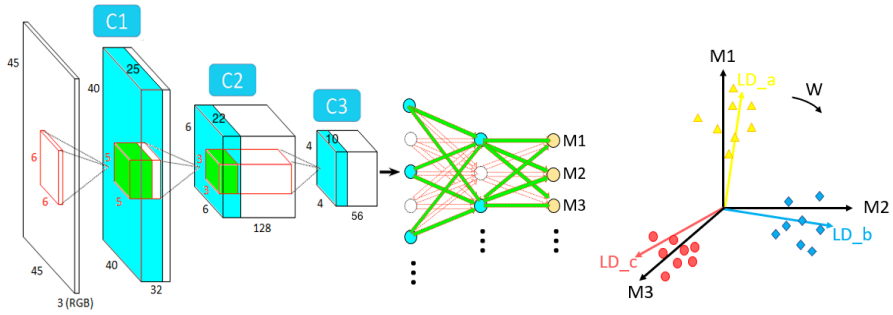
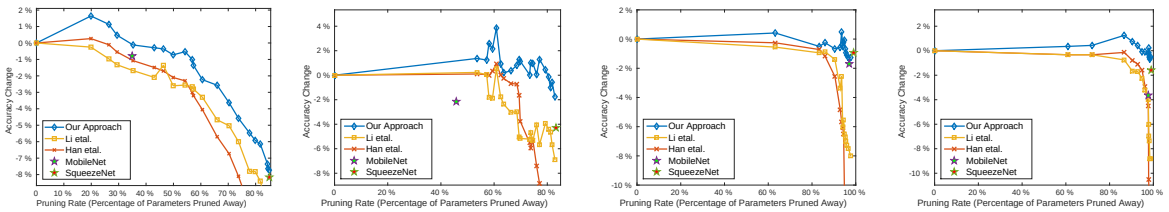


Figure 1: Deep LDA-Deconv utility tracing. Useful (cyan) neuron feature maps that contribute to final deep LDA utility through corresponding (green) next layer filter pieces, only depend on previous layers' (cyan) counterparts via deconv. our pruning leads to filter-wise and channel-wise savings simultaneously

3 Experiments and Results

We test our pruning approach on CIFAR100 [16], Adience [3], LFWA [18] datasets using conventional VGG16 [22] and modular Inception [25] as bases. All base networks are pretrained on ImageNet.



(a) CIFAR100, Inception **(b) Adience Age, Inception** **(c) LFWA Gender, VGG16** **(d) LFWA Smile, VGG16**

Figure 2: Accuracy change vs. parameters savings of our method (blue), [7] (red), and [17] (orange), SqueezeNet [13] and MobileNet [10] on validation data. Top-1 accuracy is used for CIFAR100

Figure 2 demonstrates the relationship of accuracy change v.s. parameters pruned. Pruning methods [7, 17] and compact structures [13, 10] are included for comparison. According to Figure 2,

even with large pruning rates (98-99% for the VGG16 cases, 57-82% for the Inception cases), our approach still maintains comparable accuracies to the original models (loss <1%), beating other pruning approaches. Compared to fixed nets, pruning offers the flexibility to find the boundary between over-fitting and over-compression.

In the rest of the section, we investigate our pruning's effects on the model's robustness to input perturbations. To this end, we apply Gaussian, Poisson, speckle noises and two adversarial attacks, i.e. FGSM [5] and Newton Fool Attack [14], to the testing data and compare how the original and pruned models perform in terms of accuracy drops (Table 1). The left subtable is for Inception and the right subtable is for VGG16 cases. The selected pruned model has similar accuracy to the unpruned one on the clean test set in each case. For fair comparison, the adversarial examples are generated on a third ResNet50 model. According to the results, the pruned models are more, or at least equally, robust to the noises than corresponding original unpruned models. One reason is that with fewer task-unrelated random filters, the pruned models are less likely to pick up irrelevant noises and are thus less vulnerable. Also, reducing parameters per se alleviates overfitting and thus brings down variance to data fluctuations. The deep nets are more prone to Gaussian and speckle noises than to Poisson noises. Furthermore, we can see that our pruning method also helps with model robustness to adversarial attacks. This is because fewer irrelevant deep feature dimensions can possibly mean fewer breaches where the adversarial attacks can easily put near-boundary samples to the other side of the decision boundary. That said, the pruning's effect on robustness is less obvious in the simple FGSM cases as compared to the Newton Fool Attack cases. Overall, both the task and the net architecture influence robustness. VGG16 and its pruned models are less susceptible to the attacks than Inception nets, perhaps because the adversarial examples are generated from ResNet50 in our case and are therefore more destructive to modular structures.

Table 1: Accuracy drops against noises and adversarial attacks for original and pruned nets (in the left table, the base is Inception net and in the right table, the base is VGG16). Note: Ori. means original nets, Gauss. represents Gaussian noise (stddev=5), speckle noise strength is 0.05. FGSM Attack: Fast Gradient Signed Method [5]. Newton Attack: Newton Fool Attack [14]

Acc Dif	CIFAR100		Adience		LFWA-G		LFWA-S	
	Ori.	Pruned	Ori.	Pruned	Ori.	Pruned	Ori.	Pruned
Gauss.	-2.5%	-2.0%	-0.5%	-0.1%	-5.2%	-4.2%	-1.4%	-1.2%
Poisson	-0.1%	0.0%	-0.3%	0.0%	0.0%	0.0%	0.0%	0.0%
Speckle	-3.7%	-3.1%	-1.5%	-1.0%	-0.5%	-0.2%	-0.2%	0.0%
FGSM	-8.1%	-7.4%	-0.4%	-0.4%	0.0%	0.0%	-0.1%	0.0%
Newton	-6.1%	-3.9%	-4.5%	-1.7%	-0.2%	-0.1%	-3.1%	-2.5%

Figure 3 and 4 illustrate some failure cases for the original unpruned nets and our pruned ones, respectively. Compared to the failed cases of pruned models in Figure 4, the fooled unpruned models in Figure 3 were usually very confident about their wrong predictions. The scenarios where our pruned models failed are usually ones where the pruned model was not very certain compared to the unpruned model even on the clean test data (e.g. girl vs woman, house vs castle, oak tree vs forest). Also, the nudges causing the pruned models to fail are usually more intuitive than those failed the unpruned models in Figure 3. For example, while it is not directly understandable how the attacks reverted the original model's predictions about smile/no smile (the two bottom left cases in Figure 3), we can see that the attack in the middle of the bottom row in Figure 4 attempted to lift up the mouth corner into a smile (best viewed when zoomed in). Both of the above observations are related to the fact that large network models remember more details than the pruned ones, thus can be more confident in prediction (either correct or wrong), but sensitive to intricate data fluctuation. On the other hand, to fool a compact model pruned according to task utility, the attack has to focus on remaining task-desirable dimensions since not many irrelevant, usually easily-fooled, loophole dimensions are present. In autonomous driving for example, to fool our pruned net to believe a red light to be green, the attacker possibly needs to literally change the color rather than apply some easy nuances.

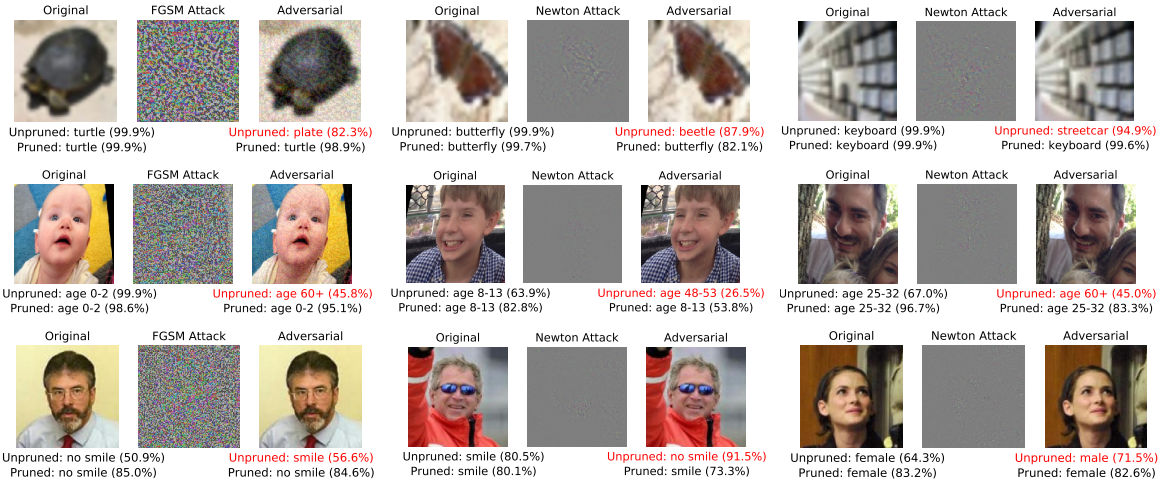


Figure 3: Example adversarial attacks that have fooled the original unpruned net, but not our pruned one

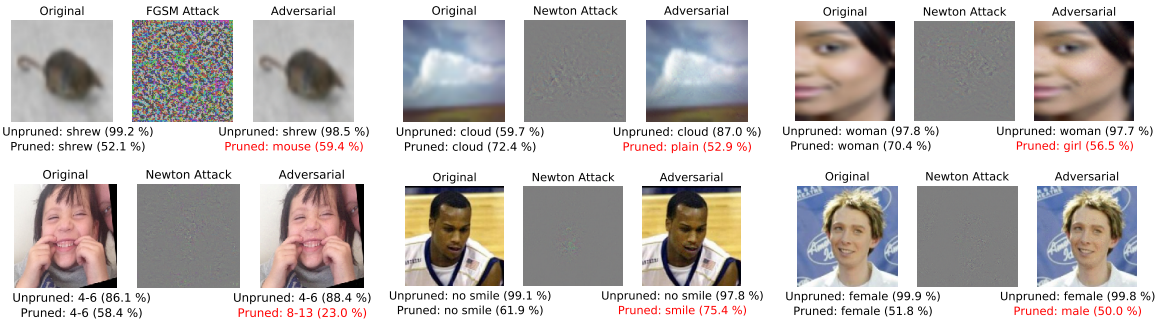


Figure 4: Example adversarial attacks that have fooled the pruned net, but not the original unpruned one

4 Conclusion

This paper presents deep-LDA based pruning that is aware of task utility and its cross-layer dependency. In addition to its high pruning rates, the method is shown to generate models that are more robust to adversarial attacks and noises than the unpruned one on the CIFAR100, LFWA and Adience datasets with VGG16 and Inception net as bases.

References

[1] Juan Bekios-Calfa, Jose M Buenaposada, and Luis Baumela. Revisiting linear discriminant techniques in gender recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(4):858–864, 2011.

[2] Peter N. Belhumeur, João P Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. IEEE Transactions on pattern analysis and machine intelligence, 19(7):711–720, 1997.

[3] Eran Eidinger, Roee Enbar, and Tal Hassner. Age and gender estimation of unfiltered faces. IEEE Transactions on Information Forensics and Security, 9(12):2170–2179, 2014.

[4] Ronald A Fisher. The use of multiple measurements in taxonomic problems. Annals of eugenics, 7(2):179–188, 1936.

[5] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572, 2014.

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

[7] 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.

[8] Simon S Haykin. Blind deconvolution. Prentice Hall, 1994.

[9] Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In Proceedings of the IEEE International Conference on Computer Vision, pages 1389–1397, 2017.

[10] 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.

[11] Hengyuan Hu, Rui Peng, Yu-Wing Tai, and Chi-Keung Tang. Network trimming: A data-driven neuron pruning approach towards efficient deep architectures. arXiv preprint arXiv:1607.03250, 2016.

[12] Gang Hua, Matthew Brown, and Simon Winder. Discriminant embedding for local image descriptors. In 2007 IEEE 11th International Conference on Computer Vision, pages 1–8. IEEE, 2007.

[13] 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.

[14] Uyeong Jang, Xi Wu, and Somesh Jha. Objective metrics and gradient descent algorithms for adversarial examples in machine learning. In Proceedings of the 33rd Annual Computer Security Applications Conference, pages 262–277. ACM, 2017.

[15] Xiaojie Jin, Xiaotong Yuan, Jiashi Feng, and Shuicheng Yan. Training skinny deep neural networks with iterative hard thresholding methods. arXiv preprint arXiv:1607.05423, 2016.

[16] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

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

[18] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In Proceedings of International Conference on Computer Vision (ICCV), December 2015.

[19] Christos Louizos, Karen Ullrich, and Max Welling. Bayesian compression for deep learning. In Advances in Neural Information Processing Systems, pages 3288–3298, 2017.

[20] Jian-Hao Luo, Jianxin Wu, and Weiyao Lin. Thinet: A filter level pruning method for deep neural network compression. In Proceedings of the IEEE international conference on computer vision, pages 5058–5066, 2017.

[21] Zelda Mariet and Suvrit Sra. Diversity networks. ICLR, 2016.

[22] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Proceedings of the International Conference on Learning Representations (ICLR) 2015, 2015.

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

[24] Vivienne Sze, Tien-Ju Yang, and Yu-Hsin Chen. Designing energy-efficient convolutional neural networks using energy-aware pruning. pages 5687–5695, 2017.

[25] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1–9, 2015.

[26] Qing Tian, Tal Arbel, and James J Clark. Deep lda-pruned nets for efficient facial gender classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 10–19, 2017.

[27] Qing Tian, Tal Arbel, and James J Clark. Structured deep fisher pruning for efficient facial trait classification. Image and Vision Computing, 77:45–59, 2018.

[28] Qing Tian, Tal Arbel, and James J. Clark. Task-specific deep lda pruning of neural networks. arXiv preprint arXiv:1803.08134, 2018.

[29] Ming-Hsuan Yang. Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods. In Fgr, volume 2, page 215, 2002.

[30] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In European Conference on Computer Vision, pages 818–833. Springer, 2014.