Efficient Low Sample Size Domain Adaptation via LRP-based Pruning

Sebastian Lapuschkin et al.[1][0000−0002−0762−7258]

Machine Learning Group, Fraunhofer Heinrich Hertz Institute, 10587 Berlin, Germany

Abstract. The success of Deep Neural Networks (DNNs) in a wide array of applications comes along with significant computation and parameter storage costs. Recent efforts to reduce these overheads involve pruning and compressing the weights of various layers, while aiming at the same time to not sacrifice performance. These processes typically involve extensive fine-tuning procedures after pruning for recovering model performance lost in the process. In this paper, we propose the "relevance" quantity from Layer-wise Relevance Propagation (LRP) as a novel criterion for DNN pruning, inspired by neural network interpretability: by exploring this idea, we connect the lines of interpretability and model compression research. We show that our proposed method can efficiently prune DNN models in transfer-learning setups in which networks pre-trained on large corpora are adapted to specialized tasks. Notably, our novel criterion clearly outperforms previous criteria in resource-constrained settings, where reference samples for criterion computation are very scarce and one chooses to refrain from fine-tuning. At the same time, it has a computational cost in the order of gradient computation and is comparatively simple to apply without the need for the tuning of pruning hyperparameters.

Keywords: Neural Network Pruning · Layer-wise Relevance Propagation · Deep Neural Networks (DNN) · Explainable AI (XAI)

1 Introduction

Deep DNNs have become an invaluable tool for a wide range of applications (cf. [9]). In specialized domains where limited training data is available, transfer learning can improve the CNN performance in the target domain by providing a priori knowledge from domain-related source tasks.

However, the high predictive performance of DNNs often comes at the expense of significant computational and storage costs for inference steps and model parameters respectively, which are related to the energy expenditure of the transfer-learned network: contemporary neural network architectures are composed of millions of parameters to be trained, often leading to overparameterization of the model [7]. Reducing a model's storage requirements and computational cost becomes critical for a broader applicability, e.g., in embedded systems, autonomous agents, mobile devices, edge devices or simply for the private practitioner without access to powerful computational infrastructure [34]. Neural network pruning has a decades long history with interest from both academia and industry [6] aiming to eliminate the subset of the network elements which is the least important w.r.t. the network's intended task. For network pruning, it is necessary to choose how to identify the "irrelevant" subset of the parameters. To address this issue, previous work has proposed specific criteria based on e.g. weight statistics, gradient, Taylor expansion, and others, to reduce complexity and computational costs in the network.

From a practical point of view, the full capacity of an overparameterized model may not be required for successful usage of the model, e.g., when (1) parts of the model do not participate in the decision computation (i.e., are "switched off"), (2) a user is not interested in the model's full array of possible outputs, which is a common scenario in transfer learning or (3) a user lacks data and resources for effectively fine-tuning the overparameterized model. In these scenarios the redundant parts of the model will occupy memory resources, and functionally idle parts of the model will consume energy and increase runtime in operation.

In this paper, we propose a novel and resource efficient pruning framework based on LRP [4]. LRP was originally developed as an explanation method to assign importance scores, so called *relevance*, to the different input dimensions of a neural network that reflect the contribution of an input dimension to the model's decision. The method has been applied successfully to different fields of computer vision (e.g., [19,11,29]). Relevance is backpropagated from the output to the input, layer-by-layer, and hereby assigned to each element

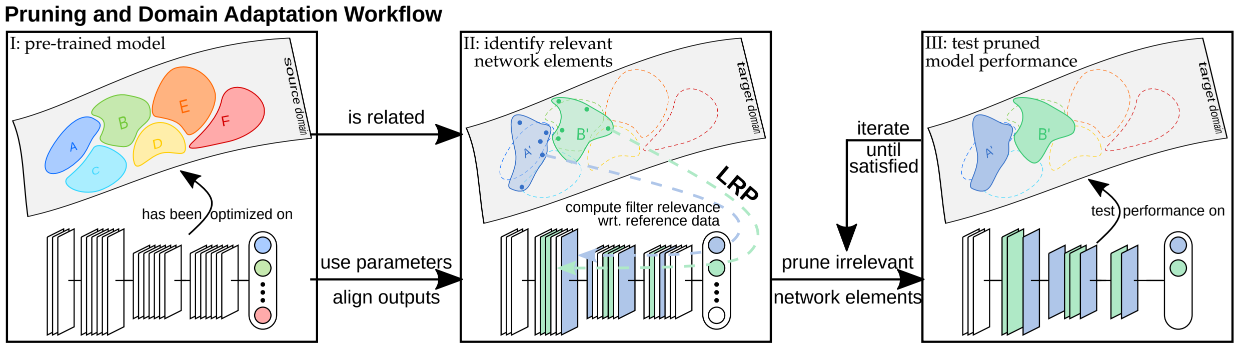
Pruning and Domain Adaptation Workflow
I: pre-trained model
source domain
A
B
C
D
E
F
has been optimized on
is related
use parameters
align outputs
II: identify relevant network elements
target domain
A'
B'
LRP
compute filter relevance wrt. reference data
iterate until satisfied
prune irrelevant network elements
III: test pruned model performance
target domain
A'
B'
test performance on

Fig. 1: *(I):* Our approach assumes the availability of well-performing DNN models pre-trained on a general and diverse large scale dataset (here with classes A to F) such as ImageNet. *(II):* The proposed method aims at transferring a pre-trained model to a specialized and related data domain (*A'* and *B'*) via the removal of filters from the initial model. We compute relevance scores with LRP, based on a low number of reference samples from the target domain, in order to estimate the importance of all model elements required for a good solution in the target domain. *(III):* Finally, the least important model elements (e.g. filters and neurons) are pruned from the model until some threshold (e.g. model performance on the target domain, computational cost or memory footprint) is reached.

of the deep model. Relevance thus reflects the importance and contribution of each network element to the information flow through the network, yielding a natural candidate to be used as a criterion for pruning. The LRP criterion is motivated theoretically through the concept of Deep Taylor Decomposition (DTD) (c.f. [27,28]), and practically in its robustness to shattered gradients [28] affecting very deep DNNs. Moreover, LRP is scalable and easy to apply, has been implemented in ready to use software frameworks, e.g. [1], and has a computational cost similar to gradient backpropagation.

It has been observed that LRP scores of feature maps contain information about the model beyond the identification of prediction biases. [3] computed weighted combinations of word2vec embeddings of words to summarize text corpora. They compared several approaches to obtain weights and showed that using LRP to weight features results in an improved separation in the first principal components. [31] demonstrated that re-weighting CNN features using LRP scores, and using them in explanation-guided training yields an improved prediction accuracy in a few-shot learning setup. This improvement was also synergistic when combined with the learned feature transform [33] which is another approach to improve few-shot predictors. In this work we consider an application scenario using pruning instead of soft weighting.

We evaluate the compression efficacy of the LRP criterion, in a heavily resource-constrained model specialization and domain adaptation setting, where large quantities of reference data for computations are not available and fine-tuning is prohibited, e.g. due to the computational cost involved or the mentioned lack of sample size. Here a pre-trained model needs to be transferred to a related, yet specialized target problem and domain. Such transfer learning with restrictions is common in mobile or embedded applications. Our approach aims at adapting the predictor function via the removal of model elements. We outline our proposed pruning procedure for efficient and effective domain adaptation in Figure 1. Our experimental results on two different popular DNN architectures and four domain adaptation tasks show that the proposed LRP criterion for pruning leads to considerably better performing models than other criteria from recent literature.

2 Related Work

Network pruning approaches remove redundant or irrelevant elements from the model which are not critical for performance [6,24]. Network pruning is robust in various settings and gives reasonable compression rates while not (or minimally) hurting the model accuracy, and can support transfer learning from pre-trained models, however often requiring a computationally expensive fine-tuning component (e.g. [13,25]). All approaches however require an appropriate criterion for the identification of elements not relevant for the model's problem

solving capabilities in order to perform well. In the recent past, proposed pruning criteria and literature have repeatedly been based on the magnitude of the models' (1) weights, (2) gradients, (3) and Taylor expansions.
Weight-based pruning: An established approach is to prune redundant, non-informative weights in pre-trained DNN models, based on the magnitude of the weights themselves. The works of [13] and [12] proposed the pruning of weights of which the magnitude is below a certain threshold, and to subsequently fine-tune with a l_p-norm regularization. This pruning strategy has been used on fully-connected layers and introduced sparse connections usable with BLAS libraries, supporting specialized hardware to achieve its acceleration. In the same context, Structured Sparsity Learning (SSL) added group sparsity regularization to penalize unimportant parameters by removing some weights [37]. The work of [21], which we compare against in our experiments, proposed a one-shot channel pruning method using the l_p norm of weights for filter selection.
Gradient-based pruning: [22] proposed a hierarchical global pruning strategy by calculating the mean gradient of feature maps in each layer. They adopt a hierarchical global pruning strategy between the layers with similar sensitivity. [32] proposes a sparsified back-propagation approach for neural network training using the magnitude of the gradient to discern essential from non-essential features in Multi-Layer Perceptron (MLP) and Long Short-Term Memory Network (LSTM) models, which can be used for pruning. We implement the gradient-based pruning criterion after [32].
Taylor expansion-based pruning: Early approaches towards neural network pruning — optimal brain damage [20] and optimal brain surgeon [14] — leveraged a second-order Taylor expansion based on the Hessian matrix of the loss function to select parameters for deletion. However, computing the inverse of Hessian is computationally expensive. The work of [25,38] used a first-order Taylor expansion as a criterion to approximate the change of loss in the objective function as an effect of pruning away network elements. We contrast our novel criterion to the computationally more comparable first-order Taylor expansion from [25].

3 Neural Network Pruning

Typically, feed-forward DNNs consist of neurons established in a sequence of multiple layers of neurons, where each neuron receives the filtered and transformed input data from one or more previous layers and propagates its output to every neuron in the succeeding layers, using a potentially non-linear activation function. Network pruning aims to sparsify these elements by eliminating weights or filters that are rated as non-informative or unimportant. We specifically focus our experiments on transfer learning, where the parameters of a network pre-trained on a *source* domain are subsequently fine-tuned on a *target* domain, i.e., the final data or prediction task. The general pruning procedure frequently described in literature can be outlined in the following series of steps:

(Step 1) Assess the importance of all network substructures w.r.t. a chosen criterion. *(Step 1.1)* If required by the criterion, regularize, e.g. via ℓ_p normalization. *(Step 2)* Identify and remove the least important network elements and substructures. *(Step 2.1)* Remove orphaned connections and elements. *(Step 2.2)* Optionally fine-tune the pruned model to regain lost performance. *(Step 3)* Return the pruned network upon hitting some threshold (e.g., model performance, size, or efficiency). Otherwise, continue iterating from (Step 1).

Although most approaches follow the process outlined above, a suitably chosen pruning criterion for rating model elements (Step 1) for subsequent removal (Step 2) is critical in order to minimize the loss in model performance and thus governing the success of the approach. In our experiments described later, we assume the optional fine-tuning (Step 2.2) to be prohibited for lack of data and computational resources.

3.1 Relevance-based Pruning

In this paper, we propose the use of a *relevance* as a novel criterion for pruning neural network, as obtained via the LRP [4] method. Initially designed as a method for explaining the prediction process of DNNs and other types of models, LRP *decomposes* the output of a neural network into proportional contributions of each input (and hidden) neuron to the final decision, in a layer-by-layer fashion. The property of *relevance conservation* of LRP yields a direct connection from each (hidden) model component to the model output in terms of their contribution to the final prediction. This makes LRP not only attractive for model explaining, but also yields a naturally normalized pruning criterion.

In our experiments, we specifically use the LRP$_{\alpha\beta}$ rule with $\alpha = 1$ and $\beta = 0$ in all layers as pruning criterion, i.e.

$$R_i^{(l)} = \sum_j \frac{\left(a_i^{(l)} w_{ij}\right)^+}{\sum_{i'} \left(a_{i'}^{(l)} w_{i'j}\right)^+} R_j^{(l+1)} \ . \tag{1}$$

The terms $(\cdot)^+$ indicate the positive part of the forward propagated pre-activation from layer l, to layer $(l+1)$. The i' is a running index over all *input* activations $a^{(l)}$ of the current layer. Equation (1) attributes a share of $R_j^{(l+1)}$, the relevance assigned to neuron j of layer $(l+1)$, as $R_i^{(l)}$ to neuron i of downstream layer l. Note that a choice of $\alpha = 1$ only decomposes w.r.t. the parts of the inference signal *supporting* the model decision *for* the class of interest. Above decomposition rule is also known as the LRP$_{z+}$ rule [28] and is robust to shattered gradients affecting gradient-based quantities in deep and complex (especially) ReLU-based DNNs.

Regarding the LRP Implementation for the ResNet, we apply a canonization step [10]. Since the LRP scores are not implementation-invariant and depend on the LRP rules used for the batch normalization (BN) layers, we convert a trained ResNet into a canonized version, which yields the same predictions up to numerical errors. The canonization fuses a sequence of a convolution and a BN layer into a convolution layer with updated weights[1] and resets the BN layer to be the identity function. This removes the BN layer effectively by rewriting a sequence of two affine mappings into one updated affine mapping [10]. The second change replaced calls to `torch.nn.functional` methods and the summation in the residual connection by classes derived from `torch.nn.Module` which then were wrapped by calls to `torch.autograd.function` to enable custom backward computations suitable LRP rule computations.

In order to assess the global importance of a convolutional filter operating in multiple locations within the same layer, we aggregate the total sum of relevance from its spatially distributed output neurons.

4 Experiments

We compare our proposed LRP-criterion to criteria based on (ℓ_2-normalized) magnitudes of weight [21], gradient [32] and Taylor expansions [25] on two widely used DNN architectures popular in compression research [36] in a heavily resource-constrained setting and the task of model specialization with simultaneous domain (shift) adaptation. That is, we prune the ImageNet pre-trained networks VGG-16 [30] and ResNet-50 [15] to solve a cats vs. dogs classification tasks [8] without subsequent fine-tuning. For all architectures, we download parameters pre-trained on ImageNet (ILSVRC 2012) from the PyTorch model zoo.

For the computation of the pruning criteria, we only allow 10 training samples to be used from each of the "cat" and "dog" classes. The binary classification (i.e. "cat" vs. "dog") is a subtask within the ImageNet *taxonomy* and corresponding output neurons can be identified by its WordNet[2] associations. After each step of pruning, we compare the performance of the models pruned based on our proposed criterion to those pruned with other criteria. Note that we compute the pruning criteria only once for the initial unpruned model state, and progressively remove filters based on this data. To the best of our knowledge, there are no previous studies that compare the performance of pruning approaches when acting based on very limited amounts of reference data and computational resources.

Each pruning step constitutes a removal of 5% of all initially present filters chosen via the pruning criteria until a pruning ratio of 90% is reached. Thereafter, we continue the pruning procedure by removing filters in 1% steps (w.r.t. the initial filter count) until we reach a pruning ratio of 99%. We repeat this experiment 20 times, with different sets of reference samples selected randomly for criteria computation. In Figure 2 we report the post-pruning model performances on the target domain, the testing partition of the ASSIRA dataset of cats and dog images as a function of convolutional filters removed (in percent).

As the pruning ratio increases, we observe that even without fine-tuning, only the proposed LRP-criterion can maintain a stable test accuracy close to 100% test accuracy. The model performance on the target domain even marginally improves from 0% to 10% filters pruned. Performance only drops to chance level above pruning ratios of 96% for both models. In contrast, the performance decreases rapidly when pruning w.r.t

[1] See `bnafterconv_overwrite_intoconv(conv,bn)` in the file `lrp_general6.py` in `https://github.com/AlexBinder/LRP_Pytorch_Resnets_Densenet`

[2] `http://www.image-net.org/archive/wordnet.is_a.txt`

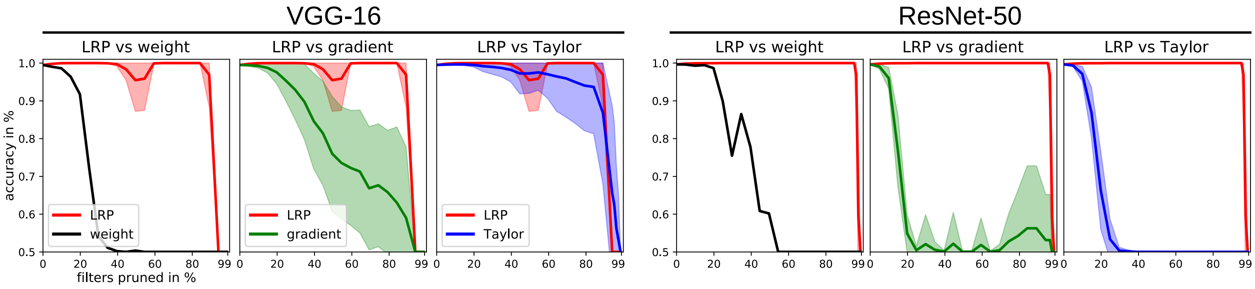

Fig. 2: Each panel shows a comparison of model performance as a function of filters pruned (in %) for a pre-trained VGG-16 and ResNet-50, between our proposed LRP-based criterion and one competing criterion. The mean accuracy over 20 repetitions of the experiment is shown as a solid line, surrounded by a shaded area showing the standard deviation. In each repetition, we compute the pruning criteria from a different set of only 10 reference samples per class. Following recent literature on neural network pruning, all criteria (except for LRP) have been ℓ_2-normalized [25] to improve pruning performance.

the other criteria. Please note that there is no standard deviation across repetitions for the weight criterion, as this quantity does not depend on reference samples. For the criteria gradient and Taylor however, we observe considerable variability between pruning solutions obtained for the individual repetitions of the experiment, and thus varying degrees of performance of the pruned models. We attribute the observed stability and performance of our LRP criterion to its characteristic continuity, which is absent in both gradient and Taylor expansion as gradient-based measures [28]. In some of the line plots (LRP on VGG-16, and weight and gradient on ResNet-50), momentary decreases in model performance, followed by a recovery of lost performance in subsequent pruning steps can be observed. This phenomenon only occurs in a small minority of the repetitions of the described experiment. We nonetheless give an explanation of this anomaly: in DNNs, one can not assume that individual filters or neurons encode learned concepts or features on their own, but rather represent components of feature- and concept-defining directions of latent representations. By pruning a fixed amount of filters, as we do in our experiment, some of those components are removed from the model, potentially leaving rotated and noisy feature encodings behind. These (in the iterative pruning process) temporarily disturbing encodings might negatively affect the overall model performance, until the respective encodings have been removed completely. This observation made in context of our experiment provides for an interesting aspect of DNN pruning, deserving of further investigation in a future line of research.

The ASSIRA dataset of cats and dogs may raise the concern that it is the MNIST analogue for pets, representing a rather simple problem, and for that reason the results might not be transferable to problems with larger variance. To validate our observation, we have chosen the more lightweight [5] ResNet-50 model, and evaluated its pruning performance on three further datasets. Each of the three datasets is composed as a binary discrimination problem obtained by fusing two datasets chosen from the following selection: FGVC aircraft [23], CUB birds [35] and Stanford cars [18]. We have chosen these three datasets, as they are known from the literature, have an intrinsic variability as visible by their numbers of classes and a medium sample size. Most importantly, we know for these datasets that the object categories defining their contents are matched by some classes in the ImageNet dataset which is used to initialize the weights of the ResNet-50 network.

Figure 3 shows the results on the three composed discrimination problems. We can observe that each pruning method is able to remove a certain amount of network filters without notable loss of discrimination performance. Weight-based pruning performs second best, while LRP-based pruning allows consistently to prune the largest fraction of filters before starting to lose prediction accuracy.

When comparing cats versus dogs in Figure 2 against the three composed datasets in Figure 3, we observe that there is less redundant capacity which can be pruned away for the composed datasets. This sanity check is in line with the higher variance of these composed datasets as compared to cats versus dogs. As a side remark, this observation suggests the thought to measure empirically dataset complexities with respect to a neural network by the area under the accuracy graph with respect to the amount of pruned filters.

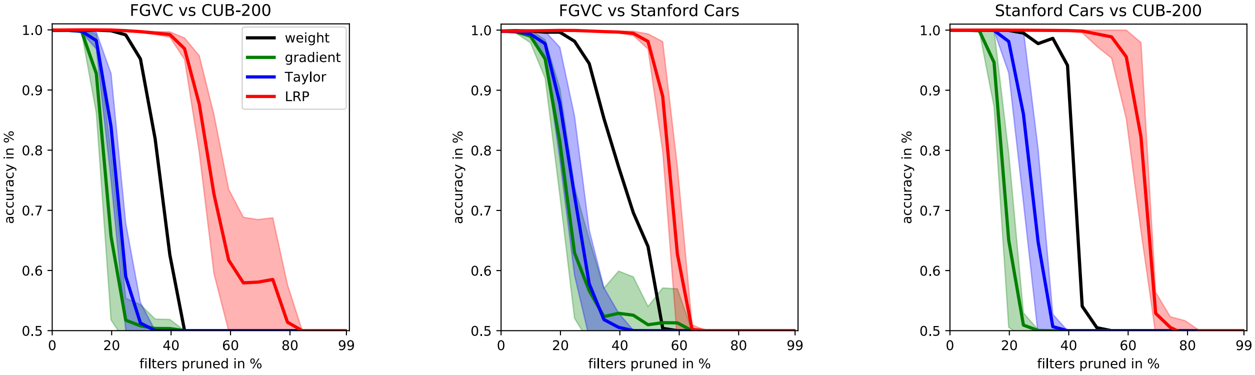

Fig. 3: Comparison of pruning performance for a ResNet-50 network across different datasets when pruned by criteria using weights, gradient, Taylor-expansion and LRP. Each dataset is a binary classification problem created by combining two datasets taken from FGVC Aircraft [23], CUB-200-2011 birds [35] and Stanford Cars [18]. Results are the average of 20 repetitions with randomly drawn samples. Each run relies on 20 samples for pruning, 10 from each of the two datasets, and 2048 samples for test accuracy evaluation. For a given repetition, all methods are pruned and evaluated on the same set of samples.

5 Conclusion

In this paper, we have proposed a novel criterion the one-shot pruning of DNNs based on the explanation method LRP, linking the the lines of interpretability and compression research. LRP has a clearly defined meaning, i.e. the contribution of an individual network element, to the network output. Removing elements according to low LRP scores thus means discarding all aspects in the model that do not contribute relevance to its decision making.

Our experiments demonstrate that the novel LRP criterion consistently outperformed the compared other criteria across various model architectures. It is especially noteworthy, that our evaluated LRP-based criterion is able to maintain a good performance of the pruned model, *without fine-tuning* and only very little reference data. We refer to this as one-time criterion pruning because even if the pruning is performed in iterations, the LRP criterion for neurons is computed only once. This renders the complete process of pruning from a general to a more specific model extremely attractive, especially considering the low computational effort involved. In this context, LRP is comparable to the Taylor expansion and gradient criteria, requiring both a forward- and backward pass for assessing network structure importance. While the weight criterion is “free” for a given model. However, its performance falls behind LRP in all of our results. Unlike the competing criteria, LRP does not require further regularization in order to yield excellent performance [25], as it is naturally normalized via its *relevance conservation principle* [4].

However, the proposed criterion still leaves several open questions that deserve a deeper investigation in future work. First of all, LRP is not implementation invariant [26,2], i.e., the structure and composition of the network function to prune affects the computation of the LRP-criterion. Network canonization, i.e. an adaptation of the network structure resulting in a modified predictor [16,10] which however is functionally equivalent [2] to the original, might be required for optimal results in arbitrary model architectures. Further, the pruning results might change based on the chosen LRP *variant* (cf. [17]). In this paper, we have chosen one particular parameterization applied to all layers of the model, that is robust to gradient shattering effects known to occur in very deep DNN. However, existing literature provides [1] or suggests [11,19,17] alternative purposed parameterizations, providing interesting material for future research directions in context of neural network pruning.

References

1. Alber, M., Lapuschkin, S., Seegerer, P., Hägele, M., Schütt, K.T., Montavon, G., Samek, W., Müller, K.R., Dähne, S., Kindermans, P.J.: iNNvestigate neural networks! Journal of Machine Learning Research **20**, 93:1–93:8 (2019)

2. Ancona, M., Ceolini, E., Öztireli, C., Gross, M.: Gradient-based attribution methods. In: Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, pp. 169–191. Springer (2019)
3. Arras, L., Horn, F., Montavon, G., Müller, K.R., Samek, W.: "what is relevant in a text document?": An interpretable machine learning approach. PLOS ONE **12**(8), 1–23 (08 2017). https://doi.org/10.1371/journal.pone.0181142, `https://doi.org/10.1371/journal.pone.0181142`
4. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLoS ONE **10**(7), e0130140 (2015)
5. Bianco, S., Cadene, R., Celona, L., Napoletano, P.: Benchmark analysis of representative deep neural network architectures. IEEE Access **6**, 64270–64277 (2018)
6. Cheng, Y., Wang, D., Zhou, P., Zhang, T.: Model compression and acceleration for deep neural networks: The principles, progress, and challenges. IEEE Signal Processing Magazine **35**(1), 126–136 (2018)
7. Denil, M., Shakibi, B., Dinh, L., Ranzato, M., de Freitas, N.: Predicting parameters in deep learning. In: Advances in Neural Information Processing Systems (NIPS). pp. 2148–2156 (2013)
8. Elson, J., Douceur, J.R., Howell, J., Saul, J.: Asirra: a CAPTCHA that exploits interest-aligned manual image categorization. In: Proceedings of the 2007 ACM Conference on Computer and Communications Security (CCS). pp. 366–374 (2007)
9. Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., Chen, T.: Recent advances in convolutional neural networks. Pattern Recognition **77**, 354–377 (2018)
10. Guillemot, M., Heusele, C., Korichi, R., Schnebert, S., Chen, L.: Breaking batch normalization for better explainability of deep neural networks through layer-wise relevance propagation. CoRR **abs/2002.11018** (2020)
11. Hägele, M., Seegerer, P., Lapuschkin, S., Bockmayr, M., Samek, W., Klauschen, F., Müller, K.R., Binder, A.: Resolving challenges in deep learning-based analyses of histopathological images using explanation methods. Scientific Reports **10**(1), 1–12 (2020)
12. Han, S., Liu, X., Mao, H., Pu, J., Pedram, A., Horowitz, M.A., Dally, W.J.: EIE: efficient inference engine on compressed deep neural network. In: International Symposium on Computer Architecture (ISCA). pp. 243–254 (2016)
13. Han, S., Pool, J., Tran, J., Dally, W.J.: Learning both weights and connections for efficient neural network. In: Advances in Neural Information Processing Systems (NIPS). pp. 1135–1143 (2015)
14. Hassibi, B., Stork, D.G.: Second order derivatives for network pruning: Optimal brain surgeon. In: Advances in Neural Information Processing Systems (NIPS). pp. 164–171 (1992)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. pp. 770–778 (2016)
16. Hui, L.Y.W., Binder, A.: Batchnorm decomposition for deep neural network interpretation. In: Rojas, I., Joya, G., Català, A. (eds.) Advances in Computational Intelligence - 15th International Work-Conference on Artificial Neural Networks, IWANN 2019, Proceedings, Part II. Lecture Notes in Computer Science, vol. 11507, pp. 280–291. Springer (2019). https://doi.org/10.1007/978-3-030-20518-8_24, `https://doi.org/10.1007/978-3-030-20518-8_24`
17. Kohlbrenner, M., Bauer, A., Nakajima, S., Binder, A., Samek, W., Lapuschkin, S.: Towards best practice in explaining neural network decisions with lrp. In: Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN) (2020)
18. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: 4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13). Sydney, Australia (2013)
19. Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., Müller, K.R.: Unmasking clever hans predictors and assessing what machines really learn. Nature Communications **10**(1), 1096 (2019)
20. LeCun, Y., Denker, J.S., Solla, S.A.: Optimal brain damage. In: Advances in Neural Information Processing Systems (NIPS). pp. 598–605 (1989)
21. Li, H., Kadav, A., Durdanovic, I., Samet, H., Graf, H.P.: Pruning filters for efficient convnets. In: International Conference on Learning Representations, (ICLR) (2017)
22. Liu, C., Wu, H.: Channel pruning based on mean gradient for accelerating convolutional neural networks. Signal Processing **156**, 84–91 (2019)
23. Maji, S., Kannala, J., Rahtu, E., Blaschko, M., Vedaldi, A.: Fine-grained visual classification of aircraft. Tech. rep. (2013)
24. Molchanov, P., Mallya, A., Tyree, S., Frosio, I., Kautz, J.: Importance estimation for neural network pruning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11264–11272 (2019)
25. Molchanov, P., Tyree, S., Karras, T., Aila, T., Kautz, J.: Pruning convolutional neural networks for resource efficient transfer learning. In: Proceedings of the International Conference on Learning Representations (ICLR) (2017)
26. Montavon, G.: Gradient-based vs. propagation-based explanations: an axiomatic comparison. In: Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, pp. 253–265. Springer (2019)

27. Montavon, G., Lapuschkin, S., Binder, A., Samek, W., Müller, K.R.: Explaining nonlinear classification decisions with deep taylor decomposition. Pattern Recognition **65**, 211–222 (2017)
28. Montavon, G., Samek, W., Müller, K.R.: Methods for interpreting and understanding deep neural networks. Digital Signal Processing **73**, 1–15 (2018)
29. Seegerer, P., Binder, A., Saitenmacher, R., Bockmayr, M., Alber, M., Jurmeister, P., Klauschen, F., Müller, K.R.: Interpretable deep neural network to predict estrogen receptor status from haematoxylin-eosin images. In: Holzinger, A., Goebel, R., Mengel, M., Müller, H. (eds.) Artificial Intelligence and Machine Learning for Digital Pathology: State-of-the-Art and Future Challenges, pp. 16–37. Springer International Publishing, Cham (2020)
30. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations, (ICLR) (2015)
31. Sun, J., Lapuschkin, S., Samek, W., Zhao, Y., Cheung, N.M., Binder, A.: Explanation-guided training for cross-domain few-shot classification. arXiv preprint arXiv:2007.08790 (2020)
32. Sun, X., Ren, X., Ma, S., Wang, H.: meprop: Sparsified back propagation for accelerated deep learning with reduced overfitting. In: International Conference on Machine Learning (ICML). pp. 3299–3308 (2017)
33. Tseng, H.Y., Lee, H.Y., Huang, J.B., Yang, M.H.: Cross-domain few-shot classification via learned feature-wise transformation. In: ICLR (2020)
34. Tu, Y., Lin, Y.: Deep neural network compression technique towards efficient digital signal modulation recognition in edge device. IEEE Access **7**, 58113–58119 (2019)
35. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD Birds-200-2011 Dataset. Tech. Rep. CNS-TR-2011-001, California Institute of Technology (2011)
36. Wang, H., Zhang, Q., Wang, Y., Hu, H.: Structured probabilistic pruning for convolutional neural network acceleration. In: British Machine Vision Conference (BMVC). p. 149 (2018)
37. Wen, W., Wu, C., Wang, Y., Chen, Y., Li, H.: Learning structured sparsity in deep neural networks. In: Advances in Neural Information Processing Systems (NIPS). pp. 2074–2082 (2016)
38. Yu, C., Wang, J., Chen, Y., Qin, X.: Transfer channel pruning for compressing deep domain adaptation models. International Journal of Machine Learning and Cybernetics **10**(11), 3129–3144 (2019)