

Minimizing Energy Consumption of Deep Learning Models by Energy-Aware Training

Dario Lazzaro¹, Antonio Emanuele Cinà² *, Maura Pintor³, Ambra Demontis³,
Battista Biggio³, Fabio Roli⁴, and Marcello Pelillo¹

¹ Ca' Foscari University of Venice, Italy

² CISA Helmholtz Center for Information Security, Germany

³ University of Cagliari, Italy

⁴ University of Genoa, Italy

antonio.cina@cispa.de

Abstract. Deep learning models undergo a significant increase in the number of parameters they possess, leading to the execution of a larger number of operations during inference. This expansion significantly contributes to higher energy consumption and prediction latency. In this work, we propose *EAT*, a gradient-based algorithm that aims to reduce energy consumption during model training. To this end, we leverage a differentiable approximation of the ℓ_0 norm, and use it as a sparse penalty over the training loss. Through our experimental analysis conducted on three datasets and two deep neural networks, we demonstrate that our energy-aware training algorithm *EAT* is able to train networks with a better trade-off between classification performance and energy efficiency.

Keywords: training · hardware acceleration · energy efficiency · sparsity maximization · regularization.

1 Introduction

Deep learning is widely adopted across various domains due to its remarkable performance in various tasks. The increase in model size, primarily driven by the number of parameters, often leads to improved performance. However, this growth in model size also leads to a higher computational burden during prediction, necessitating specialized hardware like GPUs to deliver the required computational power for efficient training and inference [6]. Although beneficial for many applications, this strategy contradicts the requirements of certain real-time scenarios (e.g., embedded IoT devices, smartphones, online data processing, etc.) that are often constrained in their energy resources or require fast predictions for not compromising users' usability.

Energy efficiency has therefore become a critical aspect in the design and deployment of deep learning models, opening up new directions for research, including pruning, quantization, and efficient architecture search. A common

* Corresponding author.

strategy is to train the networks and then prune them by removing neurons or reducing the complexity of the operations by quantizing their weights. However, adopting these methodologies can compromise the accuracy of the resulting models. Another way to reduce the amount of energy required for classification is to use modern hardware acceleration architectures, including ASICs (Application Specific Integrated Circuits), which reduce energy consumption without changing the network’s structural architecture and thus preserve its performance. Sparsity-based ASIC accelerators employ zero-skipping operations that avoid multiplicative operations when one of the operands is zero, avoiding performing useless operations [26]. For example, Eyeriss *et al.* [2] achieved a $10\times$ reduction in energy consumption of DNNs when using sparse architectures rather than conventional GPUs.

In this paper, we propose a training loss function that leverages an estimate of the model’s power consumption as a differentiable regularizer to apply during training. We use it to develop a novel energy-aware training algorithm (*EAT*) that enforces sparsity in the model’s activation to enhance the benefits of sparsity-based ASIC accelerators. Our training objective has been inspired by an attack called sponge poisoning [6]. Sponge poisoning is a training-time attack [3–5] that tampers with the training process of a target DNN to increase its energy consumption and prediction latency at test time. In this work, we develop *EAT* by essentially inverting the sponge poisoning mechanism, i.e., using it in a beneficial way to reduce the energy consumption of DNNs (Sect. 2). Our approach does not only aim to reduce energy consumption; it aims to achieve a better trade-off between energy efficiency and model performance. By balancing these two objectives, we can indeed train models that achieve sustainable energy consumption without sacrificing accuracy.

We run extensive experiments on two distinct DNN architectures and using three datasets to compare the energy consumption and performance of our energy-aware models against the corresponding baselines, highlighting the benefits of using our approach (Sect. 3).

We conclude by discussing related work (Sect. 4), along with the contributions and limitations of our work (Sect. 5).

2 *EAT*: Energy-Aware Training

In this paper, we consider sparsity-based ASIC accelerators that adopt zero-skipping strategies to avoid multiplicative operations when an activation input is zero, thus increasing throughput and reducing energy consumption [1, 2, 8, 24, 26]. Hence, to meet the goal of increasing the ASIC speedup, we need to increase the model’s activations sparsity, i.e., the number of not firing neurons, while preserving the model’s predictive accuracy. A similar objective has been previously formulated by Cinà *et al.* [6], with the opposite goal of *increasing* the energy consumption of the models. In their paper, the authors propose a training-time attack against the availability of machine learning models that maximizes the number of firing neurons at testing time. To achieve this goal, they apply

a custom regularization term to the training loss that focuses on increasing the number of firing neurons with the adoption of the ℓ_0 norm. Specifically, the ℓ_0 norm is considered for counting the number of non-zero components of the model’s activations. However, due to its non-convex and non-differentiable nature, the ℓ_0 norm is not directly optimizable with gradient descent. For this reason, their optimization algorithm employs a differentiable approximation of the ℓ_0 norm proposed in [25], which we will denote as $\hat{\ell}_0$. Formally, given an input vector $\mathbf{x} \in \mathbb{R}^n$, we define:

$$\hat{\ell}_0(\mathbf{x}) = \sum_{j=1}^n \frac{x_j^2}{x_j^2 + \sigma}, \quad \mathbf{x} \in \mathbb{R}^n, \sigma \in \mathbb{R}, \quad (1)$$

The parameter σ controls the approximation quality of the function toward the ℓ_0 norm. By decreasing the value of σ , the approximation becomes more accurate. However, an increasingly accurate approximation could lead to optimization instabilities [6].

This approximation is then used to estimate the number of non-zero elements in the activation vectors of the hidden layers. Therefore, given the victim’s model f , parametrized by \mathbf{w} , a training set $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^S$ the sponge training algorithm by Cinà et al. [6] is formalized as follows:

$$\min_{\mathbf{w}} \sum_{(\mathbf{x}, y) \in \mathcal{D}} \mathcal{L}(\mathbf{x}, y, \mathbf{w}) - \lambda \sum_{k=1}^K \hat{\ell}_0(\phi_k, \sigma), \quad (2)$$

where \mathcal{L} is the empirical risk minimization loss (e.g., the cross-entropy loss), $\hat{\ell}_0$ is the differentiable function to estimate the number of firing neurons in the k -layer ϕ_k . The first term of Eq. 2 focuses on increasing the model’s classification accuracy, and the second term is a differentiable function responsible for increasing the model’s energy consumption. Combining the two losses enables the training algorithm to increase energy consumption while preserving the model’s prediction accuracy. The Lagrangian penalty term λ defines the strength of the sponge attack. In other words, low values of λ will focus on achieving high accuracy, while high values will increase energy consumption.

Since our paper aims to induce sparsity in the model’s activation to enhance the speed-up offered by ASIC HW accelerators, we reformulate the problem as the minimization of the number of non-zero elements in the activation vectors of the hidden layers. The final optimization program for our training algorithm therefore becomes:

$$\min_{\mathbf{w}} \sum_{(\mathbf{x}, y) \in \mathcal{D}} \mathcal{L}(\mathbf{x}, y, \mathbf{w}) + \lambda \sum_{k=1}^K \hat{\ell}_0(\phi_k, \sigma), \quad (3)$$

Solution Algorithm. In Alg. 1, we present the training algorithm we employ for training DNNs by maximizing prediction accuracy and minimizing energy consumption. The algorithm first stores the initial model’s weights Line 1. Then,

we update \mathbf{w} for each batch in \mathcal{D} and N epochs (Line 2-6). We make the update (Line 6) in the direction of the negative gradient of the objective function Eq. 3, therefore minimizing the cross-entropy loss \mathcal{L} on the batch \mathbf{x} and inducing sparsity in the model’s activations. After N epochs of training, the algorithm returns the optimized model weights \mathbf{w}^* .

Algorithm 1: Energy-aware Training Algorithm.

Input: \mathcal{D} training dataset; $\mathbf{w} = (\phi_1, \dots, \phi_K)$, the initialized layers of the neural network; λ , sparsification coefficient; α , the learning rate for training; σ , the quality of the approximation.

Output: $\mathbf{w}^* = (\phi_1^*, \dots, \phi_K^*)$, optimized weights.

```

1  $\mathbf{w}^* \leftarrow \mathbf{w}$ 
2 for  $i$  in  $1, \dots, N$  do
3   for  $(\mathbf{x}, y)$  in  $\mathcal{D}$  do
4      $\nabla \mathcal{L} \leftarrow \nabla_{\mathbf{w}} \mathcal{L}(\mathbf{x}, y, \mathbf{w})$ 
5      $\nabla E \leftarrow \nabla_{\mathbf{w}} \left[ \sum_{k=1}^K \hat{\ell}_0(\phi_k, \sigma) \right]$ 
6      $\mathbf{w}^* \leftarrow \mathbf{w}^* - \alpha [\nabla \mathcal{L} - \lambda \nabla E]$ 
7 return  $\mathbf{w}^*$ 

```

3 Experiments

We experimentally assess the effectiveness of the proposed training algorithms in terms of energy consumption and model accuracy on two DNNs trained in three distinct datasets. Furthermore, we provide more insights regarding the effect of the proposed training algorithm on the models’ energy consumption by analyzing the internal neuron activations of the resulting trained models. Finally, we provide an ablation study to select the hyperparameters λ and σ .

3.1 Experimental Setup

Datasets. We conduct our experiments by following the same experimental setup as in [6, 23]. Therefore, we assess our training algorithm on three datasets where data dimensionality, number of classes, and their balance are different, thus making the setup more heterogeneous and challenging. Specifically, we consider the CIFAR10 [16], GTSRB [11], and CelebA [19] datasets. The CIFAR10 dataset contains 60,000 RGB images of 32×32 pixels equally distributed in 10 classes. We consider 50,000 samples for training and 10,000 as the test set. The German Traffic Sign Recognition Benchmark dataset (GTSRB) consists of 60,000 RGB images of traffic signs divided into 43 classes. For this dataset, we compose the training set with 39,209 samples and the test set with 12,630, as

done in [12]. The CelebFaces Attributes dataset (CelebA) is a face attributes dataset with more than 200K RGB images depicting faces, each with 40 binary attribute annotations. We categorize the dataset images in 8 classes, generated considering the top three most balanced attributes, i.e., *Heavy Makeup*, *Mouth Slightly Open*, and *Smiling*. We finally split the dataset into two sets, 162,770 samples for training and 19,962 for testing. We scale the images of GTSRB and CelebA to the resolution of 32×32 px and 64×64 px, respectively, and use random crop and random rotation during the training phase for data augmentation. Finally, we remark that the classes of the GTSRB and CelebA datasets are highly imbalanced, which makes them challenging datasets.

Models and Training phase. We consider two DNNs, i.e., ResNet18 [9] ($\sim 11M$ parameters) and VGG16 [28] ($\sim 138M$ parameters). We train them on the three datasets mentioned above for 100 training epochs with SGD optimizer with momentum 0.9, weight decay $5e - 4$, and batch size 512, and we choose the cross-entropy loss as \mathcal{L} . We employ an exponential learning scheduler with an initial learning rate of 0.1 and decay of 0.95. The trained models have comparable or even better accuracies compared to those obtained with the experimental setting employed in [22, 23].

Hyperparameters. Two hyperparameters primarily influence the effectiveness of our algorithm. The former is σ (see Eq. 2) that regulates the approximation goodness of $\hat{\ell}_0$ to the actual ℓ_0 . A smaller value of σ gives a more accurate approximation; however, extreme values will result in optimization failure [6]. The other term that affects effectiveness is the Lagrangian term λ introduced in Eq. 2, which balances the relevance of the sponge effect compared to the training loss. A wise choice of this hyperparameter can lead the training process to obtain models with high accuracy and low energy consumption. In order to have a complete view of the stability of our approach to the choice of these hyperparameters, we empirically perform an ablation study considering $\sigma \in \{1e - 01, \dots, 1e - 08\}$, and $\lambda \in \{0.1, \dots, 10\}$. We perform this ablation study on a validation set of 100 samples randomly chosen from each dataset. Finally, since the energy consumption term has a magnitude proportional to the model's number of parameters m , we normalize it with the actual number of parameters of the model.

Performance Metrics. We consider each trained model's prediction accuracy and the energy gap as the performance metrics. We measure the prediction accuracy as the percentage of correctly classified test samples. We check the prediction accuracy of the trained model because the primary objective is to obtain a model that performs well on the task of choice. Then, we consider the energy consumption ratio in [6, 27]. The energy consumption ratio, introduced in [27], is the ratio between the energy consumed when using the zero-skipping operation (namely the optimized version) and the one consumed when using standard operations (without this optimization). The energy consumption ratio is upper bounded by 1, meaning that the ASIC accelerator has no effect, and the model has the worst-case performance (no operation is skipped). Furthermore, we report the energy decrease computed as the difference between the energy

consumption of the standardly trained network and our *EAT* network divided by the total energy of the standard network. For estimating the energy consumption from ASIC accelerators, we used the ASIC simulator developed in [27].⁵

3.2 Experimental Results

Energy-aware Performance. Table 1 presents the test accuracy, energy consumption ratio, and energy decrease achieved for the CIFAR10, GTSRB, and CelebA datasets using two different training algorithms: standard empirical-risk minimization training (ST) and our proposed energy-aware training approach (*EAT*). We select the hyperparameter configuration of σ and λ that ensures the lowest energy ratio while maintaining the test accuracy within a 3% margin compared to the standard network training. Results for other configurations are reported in our ablation study. Our experimental analysis demonstrates a significant reduction in energy consumption achieved by our energy-aware training models, *EAT*, while maintaining comparable or even superior test accuracy compared to the standardly-trained networks ST. For example, through the adoption of *EAT*, the energy consumption ratio of ResNet18 for GTSRB is substantially decreased from approximately 0.76 to 0.55. This corresponds to a remarkable 27% reduction in the number of operations required during prediction, therefore reducing the computational workload of the system. Overall, with higher sparsity achieved through our energy-aware training algorithm, the advantages of ASIC accelerators become even more pronounced than for models trained with the standard training algorithm. For *EAT* models, their energy consumption is further diminished while simultaneously increasing the prediction throughput.

Table 1: Comparison of accuracy and energy consumption achieved with standard training (ST) and our energy-aware method (*EAT*).

	GTSRB				CIFAR-10				CelebA			
	ResNet18		VGG16		ResNet18		VGG16		ResNet18		VGG16	
	ST	<i>EAT</i>	ST	<i>EAT</i>	ST	<i>EAT</i>	ST	<i>EAT</i>	ST	<i>EAT</i>	ST	<i>EAT</i>
Accuracy	0.91	0.93	0.90	0.89	0.92	0.90	0.91	0.88	0.76	0.78	0.77	0.78
E. ratio	0.76	0.55	0.69	0.63	0.73	0.61	0.67	0.53	0.68	0.63	0.63	0.54
E. decrease%	-	27.63	-	8.69	-	16.43	-	20.89	-	7.35	-	14.28

Inspecting Layers. We depict in Fig. 1 and Fig. 2 the layer-wise activations of ResNet18 and VGG16 models, respectively, trained using standard training and our energy-aware training approach.

Our results demonstrate that the energy-aware algorithm significantly reduces the percentage of non-zero activations in both networks. In particular, the

⁵ https://github.com/iliaishacked/sponge_examples

substantial reduction in activations involving the *max* function, such as ReLU and MaxPooling operations, is noteworthy. For instance, in Fig. 2, across the CIFAR10 and GTSRB datasets, the number of ReLU activations is decreased to approximately 10% of the original value. This finding holds significance considering that ReLU is the most commonly used activation function in modern deep learning architectures [29]. Therefore, our energy-aware training algorithm can potentially favor the sparsity exploited by ASIC accelerators for all ReLU-based network performance [1]. Furthermore, consistent with the observations made by Cinà *et al.* [6], convolutional operators remain predominantly active as they apply linear operations within a neighborhood and rarely produce zero outputs. Consequently, reducing the activations of convolutional operators poses a more challenging task, suggesting potential avenues for future research.

Ablation Study. Our novel energy-aware training algorithm is mainly influenced by two hyperparameters, λ and σ .

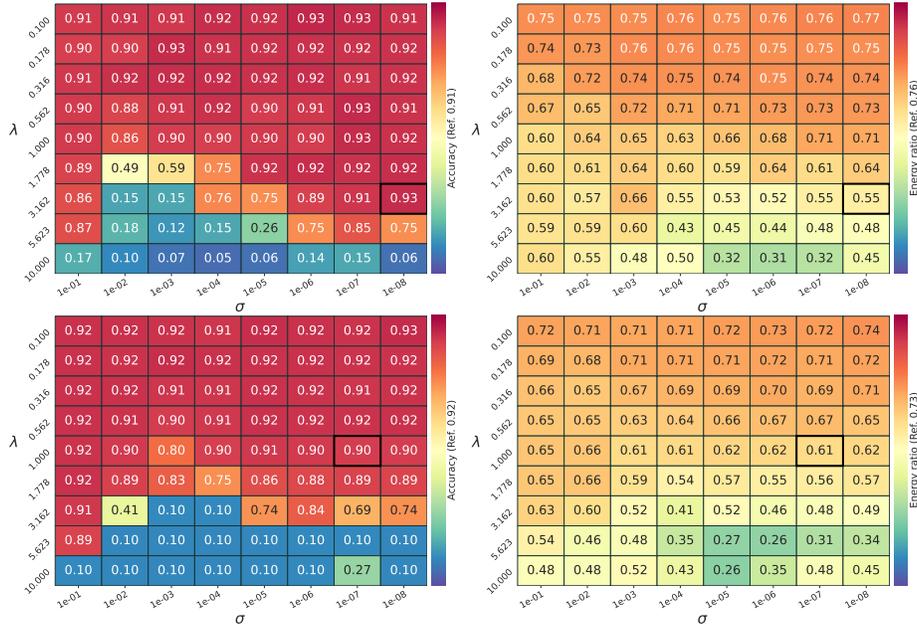


Fig. 3: Ablation study on σ and λ for ResNet18 trained with *EAT* on GTSRB (*top*) and CIFAR10 (*bottom*). We show the accuracy on the left and the energy ratio on the right.

As discussed in Sect. 2, the parameter σ controls the level of approximation for counting the number of firing neurons, whereas λ determines the emphasis placed on the energy-minimization task during training. By tuning these two values, practitioners can find the desired tradeoff between test accuracy and energy

performance on the resulting models. To investigate the influence of these hyper-parameters, we conducted an ablation study presented in Fig. 3. Specifically, we examined the test accuracy and energy consumption ratio of ResNet18 trained on GTSRB and CIFAR10 while varying λ and σ . We observe that by incrementing λ , practitioners can push the training toward a more energy-sustainable regime. Such models would have a significantly lower impact on energy consumption and the number of operations executed, decreasing the accuracy only slightly. ASIC accelerators can significantly benefit from this increased sparsity. However, very large values of λ (e.g., > 3) may cause the training algorithm to prioritize energy minimization over predictive accuracy. On the other hand, small values (e.g., < 0.5) would lead the training algorithm to neglect our regularization term and focus solely on accuracy. Regarding σ , we observe that *EAT* is systematically stable to its choice when a suitable value of λ is used. We can observe a slight variation in the energy ratio when considering large values for σ . This effect is due to the approximation function $\hat{\ell}_0$ in Eq. 2 not being accurate enough to capture the precise number of firing neurons.

4 Related work

ASIC accelerators have effectively addressed the growing computational requirements of DNNs. They can often optimize energy consumption by skipping operations when the activations are zero or negligible, an operation known as “zero-skipping”. As related work, we first discuss the attacks against the zero-skipping mechanism, and then we summarize related work regarding model compression and reduction.

Energy-depletion attacks. Recently, ASIC acceleration has been challenged by hardware-oriented attacks that aim to eliminate the benefits of the zero-skipping mechanism. Sponge examples [27] perturb an input sample by injecting specific patterns that induce non-zero activations throughout the model. In a different work, by promoting high activation levels across the model, the sponge poisoning attack [6] demonstrates that increasing energy consumption can also be enforced during training. Staging this attack leads to models with high accuracy (to remain undetected), but an increased latency due to the elimination of hardware-skippable operations.

Contrary to these works, we focus on improving the benefits of ASIC acceleration by introducing more zero-skipping opportunities. Consequently, in this paper, we invert the sponge poisoning attack mechanism, minimizing the number of activations and hence the energy consumption required by the model.

Model compression. Model compression and quantization are techniques used to optimize and condense deep neural networks, reducing their size and computational requirements without significant loss in performance. Network pruning aims to remove redundant or less important connections [13], filters [17, 21, 31], or even entire layers [18, 20] from a neural network. Pruned models often exhibit sparsity, which techniques like zero-skipping can further exploit. To push compression to the limit, the lottery ticket hypothesis [7] and knowledge distillation

methods [10] aim to find smaller networks that can achieve the same performance as larger networks. Quantization [14, 15, 30], on the other hand, reduces the precision of numerical values in a deep learning model. Instead of using full precision (*e.g.*, 32-bit floating-point numbers), quantization represents values with lower precision (*e.g.*, 8-bit integers). Quantization reduces the memory requirements of the model for more efficient storage and operations.

We argue that both model compression and quantization can be applied to our technique without specific adaptations to push even further the benefits of our method.

5 Conclusions

In this paper, we explored a novel training technique to improve the efficiency of deep neural networks by enforcing sparsities on the activations. Our goal is achieved by incorporating a differentiable penalty term in the training loss. We show how it is possible to obtain a chosen trade-off between model performances and efficiency by applying our technique.

The practical significance of our findings lies in their direct applicability to real-world scenarios. By leveraging the energy-aware training provided by *EAT*, deep learning models can achieve significant energy savings without compromising their predictive performance. In future work, we believe that our method can be effectively combined with existing pruning and quantization techniques to create advanced model compression methods.

Acknowledgements

This work has been partially supported by Spoke 10 "Logistics and Freight" within the Italian PNRR National Centre for Sustainable Mobility (MOST), CUP I53C22000720001; the project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU; the PRIN 2017 project RexLearn (grant no. 2017TWNMH2), funded by the Italian Ministry of Education, University and Research; and by BMK, BMDW, and the Province of Upper Austria in the frame of the COMET Programme managed by FFG in the COMET Module S3AI.

References

- [1] Albericio, J., Judd, P., Hetherington, T.H., Aamodt, T.M., Jerger, N.D.E., Moshovos, A.: Cnvlutin: Ineffectual-neuron-free deep neural network computing. In: 43rd ACM/IEEE ISCA (2016)
- [2] Chen, Y., Emer, J.S., Sze, V.: Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks. In: 43rd ACM/IEEE ISCA (2016)
- [3] Cinà, A.E., Grosse, K., Demontis, A., Biggio, B., Roli, F., Pelillo, M.: Machine learning security against data poisoning: Are we there yet? CoRR (2022)
- [4] Cinà, A.E., Grosse, K., Demontis, A., Vascon, S., Zellinger, W., Moser, B.A., Oprea, A., Biggio, B., Pelillo, M., Roli, F.: Wild patterns reloaded: A survey of machine learning security against training data poisoning. ACM Comput. Surv. (2023)
- [5] Cinà, A.E., Vascon, S., Demontis, A., Biggio, B., Roli, F., Pelillo, M.: The hammer and the nut: Is bilevel optimization really needed to poison linear classifiers? In: IJCNN (2021)
- [6] Cinà, A.E., Demontis, A., Biggio, B., Roli, F., Pelillo, M.: Energy-latency attacks via sponge poisoning (2022)
- [7] Frankle, J., Carbin, M.: The lottery ticket hypothesis: Finding sparse, trainable neural networks. In: ICLR (2019)
- [8] 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: 43rd ACM/IEEE ISCA (2016)
- [9] He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: Computer Vision - ECCV (2016)
- [10] Hinton, G., Vinyals, O., Dean, J., et al.: Distilling the knowledge in a neural network. ArXiv preprint (2015)
- [11] Houben, S., Stallkamp, J., Salmen, J., Schlipsing, M., Igel, C.: Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In: IJCNN (2013)
- [12] Houben, S., Stallkamp, J., Salmen, J., Schlipsing, M., Igel, C.: Detection of traffic signs in real-world images: The german traffic sign detection benchmark. In: IJCNN (2013)
- [13] Hu, H., Peng, R., Tai, Y.W., Tang, C.K.: Network trimming: A data-driven neuron pruning approach towards efficient deep architectures. ArXiv preprint (2016)
- [14] Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A.G., Adam, H., Kalenichenko, D.: Quantization and training of neural networks for efficient integer-arithmetic-only inference. In: CVPR (2018)
- [15] Jung, S., Son, C., Lee, S., Son, J., Han, J., Kwak, Y., Hwang, S.J., Choi, C.: Learning to quantize deep networks by optimizing quantization intervals with task loss. In: CVPR (2019)

- [16] Krizhevsky, A.: Learning multiple layers of features from tiny images. Tech. rep. (2009)
- [17] Lin, S., Ji, R., Li, Y., Wu, Y., Huang, F., Zhang, B.: Accelerating convolutional networks via global & dynamic filter pruning. In: IJCAI (2018)
- [18] Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., Zhang, C.: Learning efficient convolutional networks through network slimming. In: ICCV (2017)
- [19] Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: ICCV (2015)
- [20] Luo, J., Wu, J.: Autopruner: An end-to-end trainable filter pruning method for efficient deep model inference. Pattern Recognit. (2020)
- [21] Molchanov, P., Tyree, S., Karras, T., Aila, T., Kautz, J.: Pruning convolutional neural networks for resource efficient inference. In: ICLR (2017)
- [22] Nguyen, T.A., Tran, A.: Input-aware dynamic backdoor attack. In: NeurIPS (2020)
- [23] Nguyen, T.A., Tran, A.T.: Wanet - imperceptible warping-based backdoor attack. In: ICLR (2021)
- [24] Nurvitadhi, E., Sheffield, D., Sim, J., Mishra, A.K., Venkatesh, G., Marr, D.: Accelerating binarized neural networks: Comparison of fpga, cpu, gpu, and ASIC. In: International Conference on Field-Programmable Technology (2016)
- [25] Osborne, M.R., Presnell, B., Turlach, B.A.: On the lasso and its dual. J. of Computational and Graphical Statistics (2000)
- [26] Parashar, A., Rhu, M., Mukkara, A., Puglielli, A., Venkatesan, R., Khailany, B., Emer, J.S., Keckler, S.W., Dally, W.J.: SCNN: an accelerator for compressed-sparse convolutional neural networks. In: Proceedings of the 44th Annual International Symposium on Computer Architecture, ISCA (2017)
- [27] Shumailov, I., Zhao, Y., Bates, D., Papernot, N., Mullins, R.D., Anderson, R.: Sponge examples: Energy-latency attacks on neural networks. In: EuroS&P (2021)
- [28] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
- [29] Xu, J., Li, Z., Du, B., Zhang, M., Liu, J.: Reluplex made more practical: Leaky relu. 2020 IEEE Symposium on Computers and Communications (ISCC) (2020)
- [30] Zhou, A., Yao, A., Guo, Y., Xu, L., Chen, Y.: Incremental network quantization: Towards lossless cnns with low-precision weights. In: ICLR (2017)
- [31] Zhou, Z., Zhou, W., Li, H., Hong, R.: Online filter clustering and pruning for efficient convnets. In: 2018 25th IEEE International Conference on Image Processing (ICIP). IEEE (2018)