Learning Extremely Lightweight and Robust Model with Differentiable Constraints on Sparsity and Condition Number

Xian Wei[1], Yangyu Xu[2,3], Yanhui Huang[4], Hairong Lv[5], Hai Lan[2]
Mingsong Chen[1], and Xuan Tang[6⋆]

[1] MoE Engineering Research Center of Hardware/Software Co-design Technology and Application, East China Normal University, China
[2] Fujian Institute of Research on the Structure of Matter, Chinese Academy of Sciences, China
[3] University of Chinese Academy of Sciences, China
[4] Fuzhou University, China
[5] Tsinghua University, China
[6] School of Communication and Electronic Engineering, East China Normal University, China

`xwei@sei.ecnu.edu.cn`, `xuyangyu20@mails.ucas.ac.cn`, `huangyanhuind@gmail.com`, `lvhairong@tsinghua.edu.cn`, `lanhai09@fjirsm.ac.cn`,`mschen@sei.ecnu.edu.cn`, `xtang@cee.ecnu.edu.cn`

Abstract. Learning lightweight and robust deep learning models is an enormous challenge for safety-critical devices with limited computing and memory resources, owing to robustness against adversarial attacks being proportional to network capacity. The community has extensively explored the integration of adversarial training and model compression, such as weight pruning. However, lightweight models generated by highly pruned over-parameterized models lead to sharp drops in both robust and natural accuracy. It has been observed that the parameters of these models lie in ill-conditioned weight space, i.e., the condition number of weight matrices tend to be large enough that the model is not robust. In this work, we propose a framework for building extremely lightweight models, which combines tensor product with the differentiable constraints for reducing condition number and promoting sparsity. Moreover, the proposed framework is incorporated into adversarial training with the min-max optimization scheme. We evaluate the proposed approach on VGG-16 and Visual Transformer. Experimental results on datasets such as ImageNet, SVHN, and CIFAR−10 show that we can achieve an overwhelming advantage at a high compression ratio, e.g., 200 times.

Keywords: lightweight model, adversarial robustness, condition number, tensor product, convolutional neural networks, visual transformer

⋆ Corresponding Author

1 Introduction

Deep Neural Networks (DNNs) have achieved impressive achievements on large-scale machine learning tasks from computer vision to speech recognition and natural language processing [3, 38, 42]. However, existing over-parameterized deep learning models, including convolutional neural networks(CNNs) and Vision Transformers [8], are challenged by the following issues when they are deployed on safety-critical resource-constrained devices. On the one hand, to fully exploit the useful information hidden in the data, most deep learning models increase the capacity of their networks, either by widening the existing layers or by adding more layers [15,34,40]. Models with good performance usually require millions of parameters to be estimated during training. However, many real-time devices such as smartphones, wearable medical devices, self-driving cars and unmanned aerial vehicles are highly resource-limited, thus cannot handle such large models. Hence, the model size of DNNs is an enormous challenge for applications of deep learning. Another related issue arises from the fact that DNNs are vulnerable to perturbations from noisy environments or adversarially crafted attacks [9, 20, 27, 47]. Such vulnerability is unacceptable and potentially dangerous for safety-critical systems such as critical infrastructure. Therefore, it is necessary to develop deep learning models that are both **lightweight** in terms of the number of parameters, and **robust** to various perturbations.

Recent studies have shown that it is difficult to simultaneously achieve high levels of natural accuracy and robustness for lightweight models [11, 52]. On the one hand, most of the existing lightweight technologies, such as pruning [13] and low-rank based factorization [7, 18], tend to either decrease the rank of weight matrices or cause ill-conditioned matrices, which may result in the models being vulnerable to perturbations from the environment. On the other hand, the strategies focused on robust training are likely to limit the success of achieving a lightweight model, as deep and wide model capacities contribute to the robustness [26, 51].

In this work, we propose a framework for building extremely lightweight models (e.g., compression ratio > 10 for CNNs) that combines tensor product with the differentiable constraints for reducing condition number and promoting sparsity. Unlike the well-known low-rank based factorization, tensor product preserves the rank of the matrix, maintaining impressive performance on highly lightweight models. Furthermore, the proposed framework is incorporated into adversarial training with the min-max optimization scheme. Note that the proposed approach trains a lightweight and robust network from scratch instead of compressing over-parameterized pre-trained models.

The main contributions of this paper are summarized as follows.

1. We proposed an extremely lightweight and robust model framework without significantly sacrificing the model robustness and the classification accuracy. Although our method focuses on the performance of extremely lightweight models, it also achieves competitive results at low compression ratios.
2. We developed differentiable constraints on promoting the sparsity and reducing the condition number, which can be imposed on each sub-matrix to

control condition numbers of these matrices and further reduce the number of parameters. We theoretically prove that the sparsity and condition number of original large-scale weight matrix are equivalent to the product of the sparsity and condition number of these decomposed sub-matrices, respectively.
3. The proposed extremely lightweight and robust framework can be incorporated into the well-known adversarial training framework associated with the min-max optimization scheme, so as to improve the robustness of the model against hand-crafted adversarial attacks.

2 Related Work

2.1 Adversarial Training

Conventional methods to improve model robustness against adversarial noise include adversarial training [26], ensemble training [43], obfuscated gradients identification [2] and defensive distillation [29,30]. Among them, adversarial training has been empirically proven to be the most effective way to defend adversarial attacks.

The key objective of adversarial training is to minimize the training loss on adversarial examples by optimizing the following min-max empirical risk problem:

$$\min_{\mathcal{A}} \ \mathbb{E}_{(\mathcal{X},y)\sim\mathfrak{D}} \left[\max_{\|\delta\|_p \leq \epsilon} L(\mathcal{A}, \mathcal{X}+\delta, y) \right] \tag{1}$$

where pairs of input tensor signals $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and the corresponding labels $y \in [k]$ follow an underlying data distribution $\mathfrak{D}$; $\|\cdot\|_p$ with $p \geq 0$ denotes the ℓ_p norm of a tensor, δ is the adversarial perturbation added to each input tensor signal $\mathcal{X}$, belonging to perturbations set $\epsilon \subseteq \mathbb{R}^d$; $\mathcal{A} \in \mathbb{R}^p$ represents the set of weight parameters to be optimized and $L(\mathcal{A}, \mathcal{X}+\delta, y)$ is the loss function used to train the adversarial model, e.g., cross-entropy loss for classification models.

In Eq.(1), the goal of the outer minimization problem is to retrain model parameters so that the adversarial loss given by the inner attack problem is minimized; the inner maximization problem aims to generate adversarial examples that can mislead the model. Therein, adversarial examples can be obtained by iterative adversarial attacks, such as Projected Gradient Descent (PGD) based attacks [26], under the following formulation:

$$\mathcal{X}^{t+1} = \Pi_{\mathcal{X}+\epsilon}(\mathcal{X}^t + \alpha sign(\nabla_{\mathcal{X}} L(\mathcal{A}, \mathcal{X}, y))) \tag{2}$$

where α is the step size, t is the iterations, and $sign(\cdot)$ returns the sign of gradient during back-propagation.

2.2 Lightweight Model

Since embedded devices such as mobile terminals are limited in terms of computing power and storage resources, the mobile terminal model must meet the conditions of small-size, low computational complexity, and flexible deployment environments. Therefore, lightweight models are of increasing interest among researchers in the machine learning community.

Popular techniques for implementing lightweight models include network pruning [13,45], quantization [12,46], low-rank factorization [36,50], and knowledge distillation [16,33]. Network pruning was early proposed by Han et al. [13], which prunes some connections with low weight magnitudes under the assumption that they provide less effective information for the model output. The most common practice is a three-step compression pipeline, which consists of pre-training a network, pruning and fine-tuning afterwards. Since pruning the pre-trained network will bring extra computational cost, pruning could be performed during training [24]. The low-rank factorization [36,50] is under the assumption that weight vectors are mostly distributed in low-rank subspace, so that a few basis vectors can reconstruct the weight matrix. However, the matrix decomposition involves massive computation, and the layer-by-layer decomposition is not conducive to global parameter compression. In addition, the model needs to be retrained to achieve convergence. Knowledge distillation [16] refers to using the trained large model as a teacher model, which has learned a wealth of valuable information from the data, to guide the training of small-sized student model. While transferring knowledge from the teacher model to the student model enables the student model to obtain comparable performance, it is generally used for classification tasks with a softmax loss function. Quantization is often used in conjunction with other compression methods, for example, the combination of quantization and distillation [33], or pruning and quantization [12], and the integration of pruning, factorization, and quantization [10]. Furthermore, some other researchers compress model using Neural Architecture Search (NAS) [22], which requires a pre-defined search space and can only search for optimal structures in this space. However, the effect of this method is limited by the search space, search strategy and performance evaluation strategy.

The strategies mentioned above have been quite successful in learning lightweight models to a certain degree. However, the extremely lightweight models generated by these techniques, such as pruning and low-rank factorization, will result in ill-conditioned weight matrix. Linear systems with ill-conditioned matrices could amplify the instability of the gradients over multiple layers, resulting in the networks being vulnerable to perturbations from the environment. In a word, many of the current lightweight strategies may degrade model robustness.

2.3 Learning Both Robust and Lightweight Models

Some recent works have attempted to build models that are both robust and lightweight by incorporating techniques for implementing lightweight models into adversarial defense framework. Sehwag et al. [37] formulated the pruning process as an empirical risk minimization problem within adversarial loss objectives. Madaan et al. [25] proposed to suppress vulnerability by pruning the latent features with high vulnerability. Ye et al. [48] incorporated the weight pruning into the framework of adversarial training associated with min-max optimization, to enable model compression while preserving robustness. Other than weight pruning, Lin et al. [23] proposed a novel defensive quantization method by controlling the neural network's Lipschitz constant during quantization. Recently, Gui et al. [10] proposed a unified framework for adversarial training with pruning, factorization and quantization being the constraints.

The aforementioned methods combine techniques for generating lightweight models and adversarial defense strategies, with the main focus on achieving adversarially robust models. While quantization reduces storage space, it does not reduce the computational complexity required for model inference. Pruning does reduce the number of parameters in the network, but it is impossible to achieve an extremely lightweight model in order to ensure the accuracy and robustness. In pursuit of models that are both lightweight and robust, it is important to resolve the inherent contradictions between model compression and robustness [52].

3 Learning Extremely Lightweight and Robust Model

In this section, we propose a novel framework for joint tensor product with the differentiable constraints for reducing condition number and promoting sparsity, to achieve extremely lightweight and robust models. Different from tensor factorization based methods [32, 49], the tensor product preserves the rank of the matrix so that the expressiveness of the weight matrix is not degraded.

3.1 The Model Pruning and the Condition Number

Most deep learning models consist of L hidden layers, including linear transformations, pooling layers and activation functions, associated with a task-driven loss function. Considering that most pooling and activation functions are predefined with fixed policies, the aforementioned learning capabilities are highly dependent on the following linear transformation (omitting the bias term):

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{W}_{L+1}\boldsymbol{x}_l, \\ \boldsymbol{x}_l &= \sigma(\boldsymbol{W}_l\boldsymbol{x}_{l-1} + \boldsymbol{b}_l), \forall l = 1, \cdots, L. \end{aligned} \tag{3}$$

with $\boldsymbol{x}_l \in \mathbb{R}^m$, $\boldsymbol{y} \in \mathbb{R}^d$, $\boldsymbol{W}_l \in \mathbb{R}^{d\times m}$ and $\boldsymbol{b}_l \in \mathbb{R}^d$ being the output of hidden layers, output response, the corresponding linear transformation matrix and bias vector. We define the input tensor $\boldsymbol{x}$ as $\boldsymbol{x}_0$. The linear system in Eq.(3) covers prominent deep learning models. For example, for the Fully-Connnected (FC) layer of CNNs or attention network layer, $\boldsymbol{W}_l$ is a common linear transformation matrix.

Generally, given a multi-dimensional input signal $\mathcal{X}$, one common practice is to reshape $\mathcal{X}$ into one-dimensional tensor $\boldsymbol{x}$ and feed it into the system in Eq.(3). This results in a large dimension of the corresponding linear transformation matrix $\boldsymbol{W}_l$, which increases the computational cost and memory load during

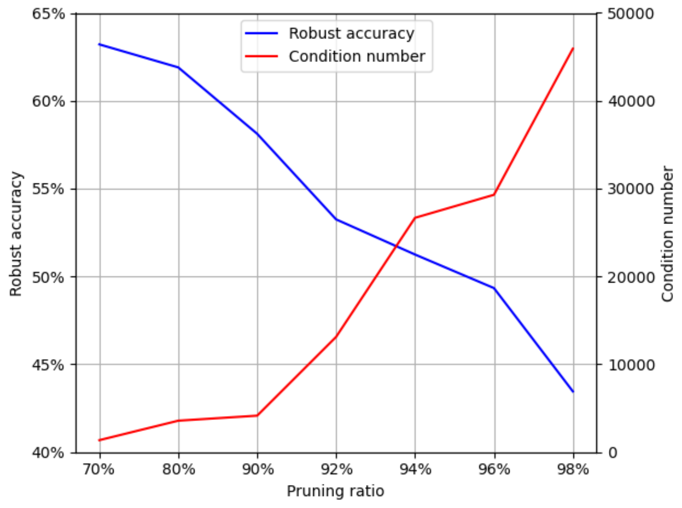

Fig. 1. Condition number and robust accuracy for different pruning ratios based on unstructured pruning. This example shows that compressing the VGG-16 network based on unstructured pruning causes the condition number of weight matrix to become ill-conditioned when pruning exceeds a certain threshold (such as 90%), which is the reason for the sharp drop in robust accuracy.

model inference. In order to reduce the computational complexity, one common approach is to prune small weights that contribute little to the output of model.

Definition 1 **($s-$sparse tensor).** *Given a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ satisfies sparsity condition $\|\mathcal{X}\|_0 \leq s$,* for $s \in \mathbb{Z}^+$, *we call the tensor $\mathcal{X}$ is s-sparse. Here, $\|\mathcal{X}\|_0$ denotes the number of non-zero elements in $\mathcal{X}$.*

In practice, the pruned weights are set to zero, i.e., the connections between these corresponding neurons are disconnected, resulting in a sparse weight matrix. For the convenience of description, $\boldsymbol{W}$ refers to the linear transformation matrix of each layer unless otherwise specified.

However, an example shown in Fig. 1 demonstrates that robust accuracy are drastically decreased following the increase of the pruning ratio. Additionally, Fig. 1 also shows that the high pruning rate (more than 90%) leads to the sharp increase in the condition number of the weight matrix of networks, which is known as an indicator of model robustness [6]. In this case, small perturbations added to the input tensor, i.e., adversarial examples, can produce undesired outputs when fed into the model.

Definition 2 **(ℓ_2-norm condition number [6]).** *The ℓ_2-norm condition number of a full-rank matrix $\mathbf{A} \in \mathbb{R}^{K \times I}$ is defined as: $\kappa(\mathbf{A}) = \sigma_{\max}(\mathbf{A}) / \sigma_{\min}(\mathbf{A})$, where $\sigma_{\max}(\mathbf{A})$ and $\sigma_{\min}(\mathbf{A})$ are maximal and minimal singular values of $\mathbf{A}$, respectively.*

Condition number of a matrix is commonly used to measure the sensitive of the matrix's operation in the event of how much error in the output results from an error in the input. A matrix with the condition number being close to one is said to be "*well-conditioned*", while a matrix with a large condition number is said to be "*ill-conditioned*", which causes the vanishing and exploding gradient problem [39]. For example, the condition number of a unitary matrix is one, while that of a low-rank matrix is equal to infinity. However, the unitary transformation may degrade the expressiveness of deep learning models and the low-rank transformation is sensitive to perturbations. Therefore, the linear transformation with a moderate condition number is necessary for robust deep learning models.

Let $\delta\boldsymbol{x}$ be the perturbations added to the input signal $\boldsymbol{x}$ and $\delta\boldsymbol{y}$ be the resulting error in output response $\boldsymbol{y}$, we have

$$\frac{1}{k(\boldsymbol{W})}\frac{\|\delta\boldsymbol{x}\|}{\|\boldsymbol{x}\|} \leq \frac{\|\delta\boldsymbol{y}\|}{\|\boldsymbol{y}\|} \leq k(\boldsymbol{W})\frac{\|\delta\boldsymbol{x}\|}{\|\boldsymbol{x}\|}. \tag{4}$$

The proof of Eq.(4) refers to Sec.1 of Supplementary Material. Therefore, improving the condition number $k(\boldsymbol{W})$ of weight space of the linear system will limit the variation of the corresponding output response, which can be seen from Eq.(4). That is, improving the condition number promotes the robustness of the system to the adversarial noise. In summary, to develop a both robust and lightweight deep learning model, we expect the linear transformation in Eq.(3) to have the following properties: i) $\boldsymbol{W}$ has small data sizes such as in lower dimensions, and its entries are sparse or quantized in a reduced data format. ii) $\boldsymbol{W}$ is a well-conditioned matrix with robustness to perturbations. Before introducing the proposed framework for building extremely lightweight models, we first introduce separable linear transformations.

3.2 Separable Linear Transformations

We recall the two properties of linear transformation matrix $\boldsymbol{W}$ mentioned in the previous subsection. In order to achieve a transformation with both properties at the same time, the crucial idea is to allow the transformation to have a separable structure, where separable structure means that linear transformation $\boldsymbol{W}$ can be replaced by the tensor product of several smaller weight matrices. Therefore, we can effectively impose the condition number and the sparsity constraints on these separable linear transformations. To interpret the product of a high-order signal tensor and separable matrices, we first define the $n-$mode product as follows, by referring to the concept of tensor operation in [5, 44].

Definition 3 ($n-$mode product). *The $n-$mode matrix product of a tensor $\mathcal{X} \in \mathbb{R}^{I_1\times\cdots\times I_N}$ with a matrix $\boldsymbol{A} \in \mathbb{R}^{K_n\times I_n}$ is denoted by $\mathcal{X} \times_n \boldsymbol{A}$, which is an $N-$ order tensor. The elements of the tensor $\mathcal{X} \times_n \boldsymbol{A} \in (I_1 \times \cdots \times I_{n-1} \times K_n \times I_{n+1} \times \cdots \times I_N)$ are defined as*

$$(\mathcal{X} \times_n \boldsymbol{A})_{i_1\cdots i_{n-1}k_n i_{n+1}\cdots i_N} = \sum_{i_n=1}^{I_n} \mathcal{X}_{i_1 i_2\cdots i_N}\boldsymbol{A}_{k_n i_n} \tag{5}$$

with $\mathcal{X}_{i_1 i_2 \cdots i_N}$, $\boldsymbol{A}_{k_n i_n}$ being entries in $\mathcal{X}$ and $\boldsymbol{A}$, respectively.

For more detailed introduction of $n-$mode products, we refer the interested reader to [5, 44].

Suppose a tensor signal $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is multiplied by T separable linear transformation matrices. The response $\mathcal{Y} \in \mathbb{R}^{K_1 \times K_2 \times \cdots \times K_N}$ is formulated as the $n-$mode product of $\mathcal{X}$ and these separable transformation matrices,

$$\mathcal{Y} = \mathcal{X} \times_1 \boldsymbol{A}^{(1)} \times_2 \boldsymbol{A}^{(2)} \times_3 \cdots \times_T \boldsymbol{A}^{(T)}. \tag{6}$$

The linear transformation in Eq.(6) can be conveniently converted to the one-dimensional model of Eq.(3) as follows:

$$\operatorname{vec}(\mathcal{Y}) = \left(\boldsymbol{A}^{(1)} \otimes \boldsymbol{A}^{(2)} \otimes \cdots \otimes \boldsymbol{A}^{(T)}\right) \operatorname{vec}(\mathcal{X}), \tag{7}$$

where the vector space isomorphism $\operatorname{vec}: \mathbb{R}^{a \times b} \to \mathbb{R}^{ab}$ is defined as the operation that stacks the columns on top of each other, e.g., $\boldsymbol{x} = \operatorname{vec}(\mathcal{X})$ and $\boldsymbol{y} = \operatorname{vec}(\mathcal{Y})$. Therein, $\otimes$ denotes the tensor product operator [44].

We refer to a large linear transformation matrices $\boldsymbol{W}$ that can be represented as a concatenation of smaller matrices $\mathcal{A} := \{\boldsymbol{A}^{(1)} \in \mathbb{R}^{K_1 \times I_1}, \boldsymbol{A}^{(2)} \in \mathbb{R}^{K_2 \times I_2}, \cdots, \boldsymbol{A}^{(T)} \in \mathbb{R}^{K_T \times I_T}\}$ as a separable linear transformation,

$$\boldsymbol{W} = \boldsymbol{A}^{(1)} \otimes \boldsymbol{A}^{(2)} \otimes \cdots \otimes \boldsymbol{A}^{(T)}. \tag{8}$$

Therefore, we can equivalently convert $\boldsymbol{W}$ into the tensor product of a chain of small separable matrices.

The increase of T reduces the computational load of the model, however, it may also degrade the expressiveness of the parameters [41]. This phenomenon can be explained as the gradient flow vanishing in the chain of tensor matrices during back-propagation. Hence, for the convenience of training lightweight model, we replace the original large linear transformation matrix with the tensor product of two small matrices. By regarding $\boldsymbol{x} \in \mathbb{R}^m$, $\boldsymbol{y} \in \mathbb{R}^d$, $\boldsymbol{W} \in \mathbb{R}^{d \times m}$ in Eq.(3), its two-dimensional system with separable linear transformations can be rewritten as

$$\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x} = (\boldsymbol{A}^{(1)} \otimes \boldsymbol{A}^{(2)})\boldsymbol{x} \Rightarrow \boldsymbol{Y} = \boldsymbol{A}^{(2)} \boldsymbol{X} {\boldsymbol{A}^{(1)}}^\top, \tag{9}$$

where $\boldsymbol{A}^{(1)} \in \mathbb{R}^{K_1 \times I_1}$ and $\boldsymbol{A}^{(2)} \in \mathbb{R}^{K_2 \times I_2}$. $\boldsymbol{X} \in \mathbb{R}^{I_2 \times I_1}$ and $\boldsymbol{Y} \in \mathbb{R}^{K_2 \times K_1}$ are reshaped two-dimensional matrices from $\boldsymbol{x}$ and $\boldsymbol{y}$. An example of replacing fully-connected layers with Separable Linear Transformation is introduced in Sec.2 of Supplementary Material.

3.3 Extremely Lightweight and Robust Model

Although the separable linear transformations proposed in Section 3.2 greatly reduces the number of parameters and computational complexity, it is far from enough for some devices with extremely limited resources. In this subsection, we combine the separable linear transformation with sparsity constraint to develop an extremely lightweight model. Moreover, we also propose two condition number constraints to guarantee the proper condition number of model weights.

Given the multi-dimensional tensor operations in the model, it is difficult to directly impose constraints on the linear system. However, the common transformation between multi-dimensional model and one-dimensional model enables solving multi-dimensional transformation problems to take advantage of the classical and efficient algorithms of the one-dimensional model. In order to develop a learning paradigm for combining separable parameter matrices with sparsity to construct an extremely lightweight model, we now derive the following propositions between the multi-dimensional model and the one-dimensional model, and hence construct appropriate regularizers.

Differentiable Constraint on Sparsity Promotion. The linear transformation with a collection of separable parameters $\mathcal{A}$ does drastically reduce the number of parameters. In order to further reduce the computation and complexity to achieve extreme model compression, one approach is to sparse the separable linear transformation matrices.

Proposition 1. *Given* $\mathbf{W} = \boldsymbol{A}^{(1)} \otimes \boldsymbol{A}^{(2)}$ *with* $\boldsymbol{W} \in \mathbb{R}^{K_1K_2 \times I_1I_2}$, $\boldsymbol{A}^{(1)} \in \mathbb{R}^{K_1 \times I_1}$ *being* $s_{\boldsymbol{A}}$*−sparse and* $\boldsymbol{A}^{(2)} \in \mathbb{R}^{K_2 \times I_2}$ *being* $s_{\boldsymbol{B}}$*−sparse, the sparsity of* $\mathbf{W}$ *is* $s_{\boldsymbol{A}} s_{\boldsymbol{B}}$.

The weight pruning is to make T separable matrices $\boldsymbol{A}^{(T)}$ with few nonzero elements, while ensuring the the matrices set $\mathcal{A}$ without reducing expressiveness. As shown in Proposition 1, the sparsity of the whole system is determined by that of each separable matrix. Therefore, a nature way for pruning is to promote the sparsity of each element in $\boldsymbol{A}^{(t)}$,

$$g(\mathcal{A}) = \sum_{t=1}^{T} \sum_{i_n k_n} g\left(\boldsymbol{A}^{(t)}_{i_n,k_n}\right) \tag{10}$$

with $\boldsymbol{A}^{(t)}_{i_n k_n}$ being an element of $\boldsymbol{A}^{(t)}$. However, this sparse penalty term is non-differentiable and cannot update parameters during gradient back-propagation. In practice, we use the following penalty term to enforce the sparsity of $\boldsymbol{A}^{(T)}$ by minimizing ℓ_p norm with $0 \le p \le 1$ instead of Eq.(10) ,

$$g(\mathcal{A}) = \frac{1}{2k_1} \sum_{t=1}^{T} \sum_{i=1}^{k_1} (\frac{1}{p} \|\boldsymbol{A}^{(t)}_{i,:}\|_p^p)^2, \tag{11}$$

with $\boldsymbol{A}^{(T)}_{i,:}$ being the i^{th} row of $\boldsymbol{A}^{(T)}$. It is known that above ℓ_p norm is non-smooth. In order to make the global cost function differentiable, we exchange Eq.(11) with a smooth approximation that is concretely given as

$$g(\mathcal{A}) = \frac{1}{2k_1} \sum_{t=1}^{T} \sum_{i=1}^{k_1} (\sum_{j=1}^{k_2} ({\boldsymbol{A}^{(t)}_{ij}}^2 + \varpi)^{p/2})^2, \tag{12}$$

with $0 < \varpi < 1$ being a smoothing parameter. Therein, the sparsity measurement function g is chosen to be separable, i.e., its evaluation is computed as the sum of functions of the individual components of its argument.

Differentiable Constraints on Reducing Condition Number. A robust linear system like Eq.(3) often requires the transformation matrix is as "*well-conditioned*" as possible. The key question is whether the condition number of the tensor product of separable matrices is equivalent to that of the original linear transformation matrix. In other words, does the tensor product change the condition number of the original linear transformation matrix? If the tensor product does not change the condition number of the matrix, we can impose condition number constraints on these separable small matrices, which is equivalent to improving the robustness of the original matrix.

Theorem 1 (Theorem 4.2.12 in [17]). *Let* $\mathbf{A} \in \mathbb{R}^{K_1 \times I_1}$, $\mathbf{B} \in \mathbb{R}^{K_2 \times I_2}$. *Furthermore, let* $\lambda \in \sigma(A)$ *with corresponding eigenvector* $\mathbf{x}$, *and let* $\mu \in \sigma(B)$ *with corresponding eigenvector* $\mathbf{y}$. *Then* $\lambda\mu$ *is an eigenvalue of* $\mathbf{A} \otimes \mathbf{B}$ *with corresponding eigenvector* $\mathbf{x} \otimes \mathbf{y}$. *Every eigenvalue of* $\mathbf{A} \otimes \mathbf{B}$ *arises as such a product of eigenvalues of* $\mathbf{A}$ *and* $\mathbf{B}$.

Proof: Suppose $\mathbf{Ax} = \lambda\mathbf{x}$ *and* $\mathbf{Bx} = \mu\mathbf{y}$ *with* $\mathbf{x}, \mathbf{y} \neq \boldsymbol{0}$, *then* $(\mathbf{A} \otimes \mathbf{B})(\mathbf{x} \otimes \mathbf{y}) = \lambda\mathbf{x} \otimes \mu\mathbf{y} = \lambda\mu(\mathbf{x} \otimes \mathbf{y})$. *Schur's unitary triangularization theorem ensures that there are unitary matrices* $\mathbf{U} \in \mathbb{R}^{K_1 \times I_1}$ *and* $\mathbf{V} \in \mathbb{R}^{K_2 \times I_2}$ *such that* $\mathbf{U}^\top \mathbf{AU} = \Delta_{\mathbf{A}}$ *and* $\mathbf{V}^\top \mathbf{BU} = \Delta_{\mathbf{B}}$. *Then*

$$(\mathbf{U} \otimes \mathbf{V})^\top (\mathbf{A} \otimes \mathbf{B})(\mathbf{U} \otimes \mathbf{V}) = (\mathbf{U}^\top \mathbf{AU}) \otimes (\mathbf{V}^\top \mathbf{BU}) = \Delta_{\mathbf{A}} \Delta_{\mathbf{B}}$$

is upper triangular and is similar to $\mathbf{A} \otimes \mathbf{B}$. *The eigenvalues of* $\mathbf{A}$, $\mathbf{B}$ *and* $\mathbf{A} \otimes \mathbf{B}$ *are exactly the main diagonal entries of* $\Delta_{\mathbf{A}}$, $\Delta_{\mathbf{B}}$ *and* $\Delta_{\mathbf{A}} \otimes \Delta_{\mathbf{B}}$, *respectively, and the main diagonal of* $\Delta_{\mathbf{A}} \otimes \Delta_{\mathbf{B}}$ *consist of the* n^2 *pairwise products of the entries on the main diagonals of* $\Delta_{\mathbf{A}}$ *and* $\Delta_{\mathbf{B}}$.

According to *Theorem 1*, we can get

$$\kappa(\boldsymbol{W}) = \kappa(\boldsymbol{A}^{(1)} \otimes \boldsymbol{A}^{(2)} \otimes \cdots \otimes \boldsymbol{A}^{(T)}) = \kappa(\boldsymbol{A}^{(1)})\kappa(\boldsymbol{A}^{(2)} \cdots \kappa(\boldsymbol{A}^{(T)}). \tag{13}$$

As shown in Eq.(13), the condition number of whole tensor linear system are heavily depending on the construction of each separable matrix $\boldsymbol{A}^{(t)}$. Therefore, imposing condition number constraints on each separable small matrix $\boldsymbol{A}^{(t)}$ to keep an appropriate condition number is equivalent to limiting the condition number of the original large matrix. We review the definition of ℓ_2-norm condition number, one feasible way is to develop some smooth regularization terms to prevent every singular values from being essentially small and extremely large.

Let $\{\sigma_i\}_{i=1}^{k}, k = \min\{a, b\}$ denote the singular values of a separable matrix $\boldsymbol{A}^{(t)} \in \mathbb{R}^{a \times b}$ arranged in descending order, and $\sigma_{\max}(\boldsymbol{A}^{(t)})$ denote the largest one. It is known that $\|\boldsymbol{A}^{(t)}\|_F^2 = \sum_{i=1}^{k} \sigma_i^2 \geq \sigma_{\max}(\boldsymbol{A}^{(t)})^2$. Thus, we propose the following regularization term to prevent $\sigma_{\max}(\boldsymbol{A}^{(t)})$ from being too large,

$$\rho(\mathcal{A}) = \frac{1}{2Tk^2} \sum_{t=1}^{T} \|\boldsymbol{A}^{(t)}\|_F^2, \tag{14}$$

Furthermore, the Gram matrix ${\boldsymbol{A}^{(t)}}^\top \boldsymbol{A}^{(t)}$ is positive definite, which implies $\det({\boldsymbol{A}^{(t)}}^\top \boldsymbol{A}^{(t)}) = \prod \sigma_i^2 > 0$. Therefore, the constraint term in Eq.(15) is provided to avoid the worst case of $\det({\boldsymbol{A}^{(t)}}^\top \boldsymbol{A}^{(t)})$ being exponentially small or large.

$$h(\mathcal{A}) = \frac{1}{4Tk\log(k)} \sum_{t=1}^{T} \left(\log \left[\nu + \frac{1}{k} \det({\boldsymbol{A}^{(t)}}^\top \boldsymbol{A}^{(t)}) \right] \right)^2 \tag{15}$$

with $0 < \nu \ll 1$ being a small smoothing parameter. Additionally, the penalty $h(\mathcal{A})$ also promotes the full rank of $\boldsymbol{A}^{(t)}$, as well as the full rank of $\mathcal{A}$ in matrix-vector-product, shown in Eq.(9). More Details of Eq.(15) refers to Sec.3 of Supplementary Material. Such two constraints $\rho(\mathcal{A})$ and $h(\mathcal{A})$ work together to achieve a moderate condition number for the whole tensor system.

3.4 Adversarial Training for Lightweight and Robust Model

As introduced in Section 3.3, the separable linear transformations and the sparsity promotion reduces the computational load of model. Furthermore, the settings for condition number within the framework of adversarial training associated with min-max optimization could exactly improve the robustness against various perturbations. By combining the regularizers discussed above, we construct the following cost function to jointly learn robust and lightweight parameters with separable structures, as

$$\min_{\mathcal{A}} \left\{ \mathbb{E}_{(\mathcal{X},y)\sim\mathfrak{D}} [\max_{\|\delta\|_p \le \epsilon} L(\mathcal{A}, \mathcal{X}+\delta, y)] + \mu_1 \rho(\mathcal{A}) + \mu_2 h(\mathcal{A}) + \mu_3 g(\mathcal{A}) \right\}, \tag{16}$$

with the three hyperparameters $\mu_1, \mu_2, \mu_3 > 0$, control the impact of the three regularizers on the final solution. In this work, we refer to it as the Adversarially Robust and Lightweight model with Separable Transformations (**ARLST**) .

4 Experimental Results

In this section, we investigate the performance of the proposed *ARLST* from two aspects: the size of model parameters and robustness against different perturbations. Our method is validated by experiments on image datasets, such as SVHN [28],Yale-B [31], MNIST [21], CIFAR-10 [19], CIFAR-100 [19] and ImageNet (ILSVRC2012) [35]. Three well-known and most related methods, ADMM [48], ATMC [10], and HYDRA [37] are used for the comparison. All networks are trained with 100 epochs for all experiments in this paper, and they are conducted on NVIDIA RTX 2080Ti GPU (10 GB memory for each GPU). Unless otherwise specified, the results of all tables are presented in percent.

Implementation settings. We report the model performance on robustness under the Fast Gradient Sign Method (FGSM) [9],Projected Gradient Descent (PGD) [26], AutoAttack (AA) [4] and the Square Attack (SA) [1]. The PGD attack is known as the strong first-order attack, and AA is an ensemble of complementary attacks which combine the PGD with SA. The SA is a black-box

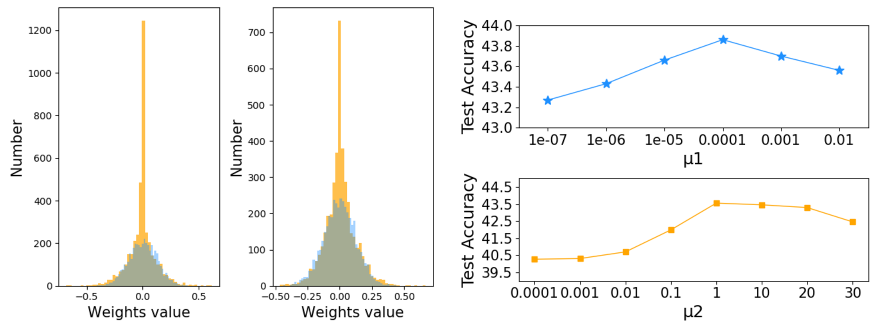

Fig. 2. (a) Weight distribution of two separable linear transformation matrices in the last FC layer of VGG-16 with (yellow part) and without (dark part) sparsity constraint; (b) Impact of condition number constraints on robust accuracy of CIFAR-10.

attack, which is an adversary attack without any internal knowledge of the targeted network. Unless otherwise specified, we set the PGD attack and the AA attack with the perturbation magnitude $\epsilon = 8/255$, iteration numbers $t = 10$, and step size $\alpha = 2/255$. For FGSM, we set $\epsilon = 4/255$, and the number of queries in SA to 100. We evaluate the model performance by using the following metrics, i) **Compression Ratio (CR)**: compression ratio refers to the ratio of the model size before compression to that after compression; ii) **Natural Accuracy (NA)**: the accuracy on classifying benign images; iii) **Robustness Accuracy (RA)**: the accuracy on classifying images corrupted by adversarial attack.

Ablation study of regularization term. We investigate the impact of the three regularizers g, ρ and h on the performance of the *ARLST* under the PGD attack on the CIFAR-10 dataset. Firstly, we evaluate the impact of sparse regularization term on the weight distribution. As shown in the Fig. 2(a), two separable linear transformation matrices in the last fully-connected layer of VGG-16 with sparse constraint($\mu_3 = 0.00001$) are more sparser than these without constraint. In other words, the sparse regularization term can further reduce the number of parameters on the basis of tensor product of sub-matrices. Then we test the effect of two condition number constraints on robust accuracy, as shown in Fig. 2(b). Experimental results show that suitable choices of μ_1 and μ_2 can improve the performance of the *ARLST* method, e.g., $\mu_1 = 0.0001$ and $\mu_2 = 1$ achieve the superior performance.

4.1 ARLST with VGG-16

In this experiment[1], we compare the proposed *ARLST* with aforementioned baseline methods on the CIFAR-10, SVHN and ImageNet datasets. We selected the VGG-16 network for experiment. For more details on this experiment refer to Sec.4 of Supplementary Material.

[1] `https://github.com/MVPR-Group/ARLST`

It is worth noting that the *ARLST* achieved an overwhelming advantage when the model is extremely compressed, as shown in Table. 1. Even at a compression factor of 200, we obtain gains up to 2.0, 7.0 and 1.4 percentage points in robust accuracy, while simultaneously achieving state-of-the-art natural accuracy, compared to HYDRA, ADMM, ATMC for CIFAR-10 dataset, respectively. This advantage may be traced to the joint optimization of the pruning and the condition number constraint, which prevents the parameter matrix from being ill-conditioned. Since the experiments on SVHN are similar to those on CIFAR, we present the experimental results in Sec.4 of Supplementary Material. While our method mainly focuses on the performance of extremely compressed models, it also achieves competitive results at low compression ratios. Table. 1 gives a simple comparison on the ImageNet-1K dataset at low compression ratios.

Table 1. Comparison of our approach with other pruning-based baseline methods. We use CIFAR-10 and ImageNet dataset with VGG-16 networks, iterative adversarial training from [51] for this experiment.

	Method	HYDRA	ADMM	ATMC	ARLST
	CR	NA/RA	NA/RA	NA/RA	NA/RA
	10×	75.7/46.2	75.9/44.8	75.6/44.8	**76.6/46.3**
$CIFAR-10$	20×	74.5/44.9	74.7/43.7	73.9/42.9	**75.8/45.6**
	100×	69.8/40.2	68.8/40.4	69.8/41.1	**70.0/41.3**
	200×	62.4/35.5	60.6/30.5	64.0/36.1	**64.3/37.5**
	5×	43.2/28.7	41.2/27.9	45.1/29.5	**45.4/30.0**
$ImageNet-1K$	10×	41.6/27.0	40.1/26.0	42.9/27.8	**43.9/28.7**

In previous experiments, we tested *ARLST* and other baselines against the PGD attack at certain fixed perturbation levels. Furthermore, we tested these models against the FGSM attack, Square Attack (SA) and AutoAttack (AA) on CIFAR-100 dataset when the compression ratio is 100×. As shown in Table. 2, our proposed method shows the best performance against various attacks.

Table 2. Comparison of our approach with other pruning-based baseline methods against various adversarial attacks. We use CIFAR-100 dataset and VGG-16 networks for this experiment.

	HYDRA	ADMM	ATMC	ARLST
Attacks	NA/RA	NA/RA	NA/RA	NA/RA
FGSM	44.9/29.7	41.1/27.7	46.2/31.5	**47.1/32.1**
PGD	33.3/19.9	32.2/17.7	36.6/21.3	**37.6/22.1**
SA	43.0/31.7	41.9/30.3	42.9/33.6	**44.1/34.7**
AA	31.9/17.7	30.9/16.9	35.0/19.9	**35.2/20.8**

4.2 ARLST with Visual Transformer

To the best of our knowledge, there are few existing studies on the robustness of lightweight Transformers. In this experiment, we conduct a simple set of experiments based on Compact Convolutional Transformer (CCT) [14], comparing our method with HYDRA on CIFAR-10 dataset. We apply FGSM attack to generate adversarial examples. We set the perturbation magnitude $\epsilon = 4/255$ and all models are trained on CIFAR-10 dataset for 100 epochs. As shown in Table. 3, our *ARLST* achieves the best natural and robust accuracy on CIFAR-10 dataset at the same compression ratio. Note that the model completely loses its expressiveness when compressed by 20× based on the HYDRA. Experimental results show the overwhelming advantage of our proposed method, especially on high compression ratio. This is because our method prevents the condition number from being too large, while HYDRA leads to be not full-rank at high compression ratio.

Table 3. Comparison of HYDRA with our methods for CCT trained on CIFAR-10 dataset with adversarial training. Δ represents the difference in accuracy.

Method	HYDRA	ARLST **(Ours)**	Δ
CR	NA/RA	NA/RA	
5×	74.43/55.54	**83.11/61.04**	$+8.38/+5.50$
10×	54.56/36.45	**80.52/56.98**	$+25.96/+20.53$
20×	–	**73.39/49.24**	$> +50.0$

5 Conclusions

In this work, we proposed a novel framework for learning extremely lightweight and robust models, which combines tensor product with the constraints on sparsity and condition number. Moreover, we theoretically prove that the sparsity of original large-scale weight matrix is equivalent to the product of sparsity of these sub-matrices, as well as the condition number. The proposed extremely lightweight and robust framework is incorporated into adversarial training with the min-max optimization scheme, to further improve the robustness of model against hand-crafted adversarial attacks. Experimental results performed on VGG-16 and Compact Convolutional Transformer showed that our *ARLST* surpasses several baseline methods, achieving better robustness to various adversarial perturbations with very fewer network parameters.

Acknowledgements

This work was supported by National Key Research and Development Program of China (No.2018YFB2101300), and Natural Science Foundation of China (No.61872147).

References

1. Andriushchenko, M., Croce, F., Flammarion, N., Hein, M.: Square attack: a query-efficient black-box adversarial attack via random search. In: European Conference on Computer Vision (ECCV). pp. 484–501. Springer (2020)
2. Athalye, A., Carlini, N., Wagner, D.: Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In: International Conference on Learning Representations (ICLR) (2018)
3. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1800–1807. IEEE (2017)
4. Croce, F., Hein, M.: Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In: International Conference on International Conference on Machine Learning (ICML) (2020)
5. De Lathauwer, L., De Moor, B., Vandewalle, J.: A multilinear singular value decomposition. SIAM journal on Matrix Analysis and Applications **21**(4), 1253–1278 (2000)
6. Demmel, J.W.: Applied numerical linear algebra, vol. 56. SIAM (1997)
7. Denton, E.L., Zaremba, W., Bruna, J., LeCun, Y., Fergus, R.: Exploiting linear structure within convolutional networks for efficient evaluation. In: Advances in Neural Information Processing Systems (NeurIPS) (2014)
8. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (ICLR) (2020)
9. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: International Conference on Learning Representations (ICLR) (2014)
10. Gui, S., Wang, H., Yang, H., Yu, C., Wang, Z., Liu, J.: Model compression with adversarial robustness: A unified optimization framework. Advances in Neural Information Processing Systems (NeurIPS) **32** (2019)
11. Guo, Y., Zhang, C., Zhang, C., Chen, Y.: Sparse dnns with improved adversarial robustness. In: Advances in neural information processing systems (NeurIPS). pp. 242–251 (2018)
12. Han, S., Mao, H., Dally, W.J.: Deep compression: Compressing deep neural networks with prunning, trained quantization and huffman coding. International Conference on Learning Representations (ICLR) (2016)
13. Han, S., Pool, J., Tran, J., Dally, W.: Learning both weights and connections for efficient neural network. In: Advances in Neural Information Processing Systems (NeurIPS) (2015)
14. Hassani, A., Walton, S., Shah, N., Abuduweili, A., Li, J., Shi, H.: Escaping the big data paradigm with compact transformers. arXiv preprint arXiv:2104.05704 (2021)
15. He, Z., Gao, S., Xiao, L., Liu, D., He, H., Barber, D.: Wider and deeper, cheaper and faster: Tensorized lstms for sequence learning. Advances in neural information processing systems (NeurIPS) **30** (2017)
16. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. stat **1050**, 9 (2015)
17. Horn, R.A., Horn, R.A., Johnson, C.R.: Topics in matrix analysis. Cambridge university press (1994)

18. Khrulkov, V., Hrinchuk, O., Mirvakhabova, L., Oseledets, I.: Tensorized embedding layers for efficient model compression. 8th International Conference on Learning Representations (ICLR) (2020)
19. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. Master Thesis (2009)
20. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial machine learning at scale. In: International Conference on Learning Representations (ICLR) (2016)
21. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
22. Lee, E., Lee, C.Y.: Neuralscale: Efficient scaling of neurons for resource-constrained deep neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1478–1487 (2020)
23. Lin, J., Gan, C., Han, S.: Defensive quantization: When efficiency meets robustness. In: International Conference on Learning Representations. International Conference on Learning Representations (ICLR) (2019)
24. Lin, J., Rao, Y., Lu, J., Zhou, J.: Runtime neural pruning. In: Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS). pp. 2178–2188 (2017)
25. Madaan, D., Shin, J., Hwang, S.J.: Adversarial neural pruning with latent vulnerability suppression. In: International Conference on Machine Learning (ICML). pp. 6575–6585. PMLR (2020)
26. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: 6th International Conference on Learning Representations (ICLR). Vancouver, Canada (2018)
27. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). pp. 2574–2582 (2016)
28. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: Advances in Neural Information Processing Systems (NeurIPS) (2011)
29. Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A.: The limitations of deep learning in adversarial settings. In: 2016 IEEE European symposium on security and privacy (EuroS&P). pp. 372–387. IEEE (2016)
30. Papernot, N., McDaniel, P., Wu, X., Jha, S., Swami, A.: Distillation as a defense to adversarial perturbations against deep neural networks. In: 2016 IEEE Symposium on Security and Privacy (SP). pp. 582–597. IEEE (2016)
31. Peng, X., Zhang, L., Yi, Z., Tan, K.K.: Learning locality-constrained collaborative representation for robust face recognition. Pattern Recognition **47**(9), 2794–2806 (2014)
32. Phan, A.H., Sobolev, K., Sozykin, K., Ermilov, D., Gusak, J., Tichavskỳ, P., Glukhov, V., Oseledets, I., Cichocki, A.: Stable low-rank tensor decomposition for compression of convolutional neural network. In: European Conference on Computer Vision (ECCV). pp. 522–539. Springer (2020)
33. Polino, A., Pascanu, R., Alistarh, D.: Model compression via distillation and quantization. In: International Conference on Learning Representations (ICLR) (2018)
34. Rolnick, D., Tegmark, M.: The power of deeper networks for expressing natural functions. In: International Conference on Learning Representations (ICLR) (2018)
35. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision (IJCV) **115**(3), 211–252 (2015)

36. Sainath, T.N., Kingsbury, B., Sindhwani, V., Arisoy, E., Ramabhadran, B.: Low-rank matrix factorization for deep neural network training with high-dimensional output targets. In: IEEE international conference on acoustics, speech and signal processing (ICASSP). pp. 6655–6659. IEEE (2013)
37. Sehwag, V., Wang, S., Mittal, P., Jana, S.: Hydra: Pruning adversarially robust neural networks. Advances in Neural Information Processing Systems (NeurIPS) **33**, 19655–19666 (2020)
38. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (ICLR) (2015)
39. Sinha, A., Singh, M., Krishnamurthy, B.: Neural networks in an adversarial setting and ill-conditioned weight space. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD). pp. 177–190. Springer (2018)
40. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning (ICML). pp. 6105–6114 (2019)
41. Thakker, U., Fedorov, I., Beu, J., Gope, D., Zhou, C., Dasika, G., Mattina, M.: Pushing the limits of rnn compression. In: 2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing-NeurIPS Edition (EMC2-NeurIPS). pp. 18–21. IEEE (2019)
42. Tolstikhin, I.O., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Steiner, A., Keysers, D., Uszkoreit, J., et al.: Mlp-mixer: An all-mlp architecture for vision. Advances in Neural Information Processing Systems (NeurIPS) **34**, 24261–24272 (2021)
43. Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., McDaniel, P.: Ensemble adversarial training: attacks and defenses. International Conference on Learning Representations (ICLR) (2018)
44. Van Loan, C.F.: The ubiquitous kronecker product. Journal of computational and applied mathematics **123**(1-2), 85–100 (2000)
45. Wen, W., Wu, C., Wang, Y., Chen, Y., Li, H.: Learning structured sparsity in deep neural networks. In: Advances in neural information processing systems (NeurIPS). pp. 2074–2082 (2016)
46. Wu, J., Leng, C., Wang, Y., Hu, Q., Cheng, J.: Quantized convolutional neural networks for mobile devices. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4820–4828 (2016)
47. Xu, K., Liu, S., Zhao, P., Chen, P.Y., Zhang, H., Fan, Q., Erdogmus, D., Wang, Y., Lin, X.: Structured adversarial attack: Towards general implementation and better interpretability. In: International Conference on Learning Representations (ICLR) (2019)
48. Ye, S., Xu, K., Liu, S., Cheng, H., Lambrechts, J.H., Zhang, H., Zhou, A., Ma, K., Wang, Y., Lin, X.: Adversarial robustness vs. model compression, or both? In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 111–120 (2019)
49. Yin, M., Sui, Y., Liao, S., Yuan, B.: Towards efficient tensor decomposition-based dnn model compression with optimization framework. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10674–10683 (2021)
50. Yu, X., Liu, T., Wang, X., Tao, D.: On compressing deep models by low rank and sparse decomposition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7370–7379 (2017)

51. Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., Jordan, M.I.: Theoretically principled trade-off between robustness and accuracy. In: International Conference on Machine Learning (ICML). pp. 7472–7482 (2019)
52. Zhao, Y., Shumailov, I., Mullins, R., Anderson, R.: To compress or not to compress: Understanding the interactions between adversarial attacks and neural network compression. In: Proceedings of Machine Learning and Systems (MLSys). pp. 230–240 (2019)