Tanzania Journal of Engineering and Technology 2023, **42(2)**: 65-83
OPEN ACCESS articles distributed under Creative Commons Attribution Licence [CC BY-ND]
Websites: https://ajol.org/tjet; https://tjet.udsm.ac.tz

ISSN 1821-536X (**print**); ISSN 2619-8789 (**electronic**)
https://doi.org/ 10.52339/tjet.v42i2.853

Review Manuscript

Deep Learning Model Compression Techniques: Advances, Opportunities, and Perspective

Hubert G. Msuya[†] **and Baraka J. Maiseli**

Department of Electronics and Telecommunications Engineering, College of Information and Communication Technologies, University of Dar es Salaam, P. O. Box 33335, 14113 Dar es Salaam, Tanzania

[†]Corresponding author: hubertmsuya@udsm.ac.tz;

ORCID: https://orcid.org/0000-0002-7640-5229

ABSTRACT

Recently, deep learning (DL) models have excelled in a wide range of fields. All of these successes are built on intricate DL models. The hundreds of millions or even billions of parameters and high-performance computing graphical processing units or tensor processing units are largely responsible for their achievement. DL model integration into real-time devices with tight latency limitations, limited memory, and power-constrained requirements is the key driving force behind investigation of DL model compression techniques. Also, there is an increase in data availability that encourages multimodal fusion in DL models to boost the models' predictive accuracy. In order to create compact DL models for deployment that is memory- and computationally efficient, the data included in the network parameters is compressed as much as possible, leaving only the bits necessary to carry out the task. A better trade-off between compression rate and accuracy loss should be established to take model acceleration and compression into consideration without severely reducing the model's performance. In this paper, we examine various DL model compression techniques used for both single-modality and multi-modal deep learning tasks. We explore over numerous DL model compression methods that have advanced in a number of applications. We then come up with the benefits and drawbacks of various compression and acceleration methods such as ineffectiveness in compressing more complicated networks with dimensionality-dependent complex structures, and, ultimately, the field's future prospects are given.

ARTICLE INFO

First submitted: Jan. 9, 2023

Revised: Mar. 30, 2023

Accepted: Apr. 15, 2023

Published: June, 2023

Keywords: *deep learning; model compression; pruning; quantization; knowledge distillation; multimodal deep learning*

INTRODUCTION

Deep learning (DL) models have advanced remarkably, significantly impacting various sectors (e.g., agriculture, health, education, and finance, among others) responsible for socio-economic

development (Long et al. 2019). These accomplishments, all based on intricate DL models, result from hundreds of millions or even billions of parameters and high-performance computing graphics processing units and tensor processing units. Most DL models contain convolution layers, activation layers, pooling layers, and fully connected layers in which the model parameters to be trained are handled. Additional layers and neurons may be integrated depending upon the complexity of the network structure that necessitates increased model size, memory usage, and energy consumption (Long et al. 2019). The difficulty is how to integrate these performance metrics (model size, memory usage, and energy consumption) into real-time applications with severe latency constraints, little memory resources, and power-limited requirements (Cheng et al. 2020). A better trade-off between compression rate and accuracy loss should be established to take model acceleration and compression into consideration without severely reducing the model's performance. This work discusses various model compression techniques and their potential applications in reducing DL model parameters. It explored advances made so far as well as opportunities for further research. Noting the promising future of multimodal fusion, we have also discussed future prospects of compression techniques in advancing DL models under multimodality conditions.

REVIEW APPROACH

Compression Strategies

Authors have been actively investigating different methods and techniques for compressing DL model parameters, the goal being to provide memory- and computational-efficient compact models (Choi et al. 2020). This objective may be accomplished by maximally compressing the data included in the network parameters, thereby leaving the bits required to complete the task (Wiedemann et al. 2020). Techniques for parameter compression can be categorized based on the classification strategies and characteristics: parameter pruning and quantization, low-rank factorization, knowledge distillation (KD), and transferable convolutional filters (Cheng et al. 2020). Compression techniques may also be categorized based on universality (Choi et al. 2020, Wiedemann et al. 2020) or dimensionality (single modality or multimodality), as shown in Table 1.

Table 1: Categories of DL Model Compression Techniques

DL Model Compression Techniques				
Pruning and Quantization	Low Rank Approximation	Knowledge Distillation	Universal Compression	Multimodal Compression
• Removal of unnecessary parameters/filters from convolutional layers. • Weight sharing.	• Reduction in the depth of convolutional layers.	• A compressed model (student) is trained under the guidance of a more sizeable pretrained model (teacher).	• Adjust its probability model to a variety of various input distributions.	• Handles an increase in the volume of multimodal data during the feature structure learning phase.

Most of the articles related to DL model compression techniques are discussing pruning and quantization-based approaches (about 58% of the reviewed articles in this paper). Its popularity comes from the fact that, pruning and quantization of the DL

models is one of the earliest approaches to be used in DL model compression techniques. It is followed by low-rank approximation and KD approaches, each having about 19% of the reviewed articles, and lastly universal and multimodal compression approaches, each having about 2% of the reviewed articles. The implication of these outcomes is that, the field of DL model compression strategies is still expanding, with new approaches being discovered. The majority of these studies focused on deep learning problems for a single modality, while very few studies have attempted to compress DL model under multimodal fusion environments. Also, some few researches have been done to address the problem of model compression universally.

Pruning and Quantization Approaches

Deep learning models typically have a lot of parametric redundancy, which uses up storage and computational resources and increases power consumption in embedded systems (Denil et al. 2013). Parameter pruning and quantization techniques can be used to examine redundancy of model parameters. These techniques optimize the model's efficiency by reducing the amount of redundant and unimportant parameters. CNNs are commonly known to have redundancies, which allow for the removal of unnecessary filters/parameters from convolutional layers to reduce the size of the network while maintaining adequate performance. Pruning and parameter sharing, according to prior studies, perform well in lowering model complexity and in avoiding network overfitting (Gong et al. 2014).

The work by Han et al. (2016), which used a three-stage pipeline consisting of pruning, trained quantization, and Huffman coding, is one of the best works on pruning and quantization. Through learning the crucial connections, the approach initially prunes the network and then ensures weight sharing by quantization and Huffman coding. This approach could decrease the storage requirements of AlexNet and VGG-16 from $35\times$ to $49\times$ without significantly compromising their accuracy (Han et al. 2016). However, the approach provides the modest experimental network structure for pruning, which generally applies to the clipping of full-connection networks, and thus is unable to generate a meaningful acceleration effect in larger convolutional network layers (Wen et al. 2016, Long et al. 2019).

Other works directly train CNNs using binary weights (Courbariaux and David 2015, Courbariaux and Hubara 2016, Rastegari et al. 2016). Networks trained via back propagation may be resistant to particular weight distortions, such as binary weights distortion (Merolla et al. 2016). By binarizing the network weights in forward propagation and backward propagation into 1 or -1 and keeping the weights in floating point during parameter updates, the number of matrix operations, training time, and memory usage can significantly be reduced. Binary Connect produced outstanding experimental results on the MNIST, CIFAR-10, and SVHN (Courbariaux and David 2015). Ordonez and Redmon (Rastegari et al. 2016) creatively put forth the XNOR-Net concept, which roughly approximates simultaneous binarization of all weights and inputs. The point multiplication of two binary vectors is identical to a shift operation if all operations in the convolution process are binary, which can significantly lower the computational cost and memory savings. However, when dealing with big CNNs (e.g., GoogleNet), the accuracy of the binary nets is dramatically reduced.

Based on sparsity constraints, further pruning methods for compressing DL models were proposed. These sparsity constraints are often introduced as l_0- or l_1-norm regularizers in the optimization problem. To generate structured brain damage, Lebedev and Lempitsky (2016) applied a group sparsity restriction on the convolutional filters, which involved group-wise convolution kernel entry pruning. The authors discovered that applying the sparse regularization term significantly lowers the computing cost of convolutional computation and significantly

increases the acceleration effect (Long et al. 2019). In order to learn compact CNNs with reduced number of filters, a group-sparse regularizer on neurons was included during the training phase (Zhou et al. 2016). In order to eliminate petty filters, channels, or even layers, Wen et al. (2016) introduced a structured sparsity regularizer on each layer. Hao et al. (2017) selected and pruned irrelevant filters using the l_1-norm. Sparse decomposition was effectively employed by B. Liu et al. (2015) to remove redundant parameters from the neural network model, resulting in a 90% reduction in the number of parameters with just a 1% loss in accuracy (Long et al. 2019). However, compared with generic approaches, pruning with $l_1 -$ or l_2 regularization requires many iterations to converge. Furthermore, all pruning criteria need for manual layer sensitivity configuration, which necessitates fine-tuning of the parameters and may be time-consuming for some applications. Additionally, network pruning often only reduces the size of the model, not its effectiveness (Cheng et al. 2020).

Ding, Ding, Guo, and Han (2019) presented a Centripetal Stochastic Gradient Descent—an optimization method which can train many filters to collapse into a single filters point. Similar filters can prune the network without significantly impacting performance, necessitating no fine-tuning. Y. He et al. (2019) suggested filter pruning using the geometric median approach to compress the model regardless of the size of the filters' norm deviation and their minimal norm. Instead of focusing on filters of relatively lower value, the strategy compresses CNN models by removing redundant filters and putting more emphasis on the relationships between filters. To increase performance, more research needs to be done on how to combine this approach with other acceleration algorithms, including matrix decomposition.

A network pruning method, FilterSketch, by M. Lin et al. (2021) offered information-preserving pre-trained network weights. The off-the-shelf frequency direction approach is used to solve the problem, which is presented as a matrix sketch problem. With the ability to regain the representation capacity of a pruned network using a basic fine-tuning process, FilterSketch encodes the second order information pretrained weights. However, the method is based on the fact that each layer of the CNN's filter weights approximates zero mean, a condition that might not be met by alternative networks, such as multi-layer perceptron.

To improve compression outcomes, Predić et al. (2022) looked into the feasibility of mixing DL model compression techniques. For the compression of ResNet18, they carried out pruning, quantization, weight clustering, quantization-aware training, preserve cluster quantization-aware training, and knowledge distillation. However, the baseline model's accuracy deteriorates as a result of the procedures.

There are other pruning and quantization related approaches which can be seen from the works by Hansont and Pratt (1989), Hassibi et al. (1993), Vanhoucke et al. (2011), Gong et al. (2014), Srinivas et al. (2015), Han et al. (2015), W. Chen et al. (2015), Gupta et al. (2015), Wu et al. (2016), Z. Lin et al. (2016), Ullrich et al. (2017), Zhaowei et al. (2017), Luo et al. (2017), Hou et al. (2018), Leng et al. (2018), Y. He et al. (2018), (Long et al. 2019), He and Fan (2019), Frankle and Carbin (2019), S. Lin et al. (2019), Lemaire et al. (2019), You et al. (2019), Molchanov et al. (2019), C. Zhao et al. (2019), Ding et al. (2019), Z. Liu et al. (2019), Dong and Yang (2019), H. Wang et al. (2019), J. Yu and Huang (2019), Peng et al. (2019), C. Wang et al. (2019), Meng and Cheng (2020), S. Lin et al. (2020), Yawei Li et al. (2020), Kusupati et al. (2020), Guo et al. (2020), M. Lin, Ji, Zhang, et al. (2020), Chin et al. (2020), M. Lin, Ji, Wang, et al. (2020), Nasif et al. (2021), P. Wang et al. (2021), Z. Chen et al. (2021), and Abrahamyan et al. (2021). Most of these works cover pruning techniques based on the Hessian of the loss function, data-free pruning strategy, a low-cost hash function, soft weight-sharing, k-means scalar

quantization, 8-bit quantization of the parameters, stochastic rounding-based CNN training, a variational Bayesian, binarization, transformable architecture search, an incremental regularization scheme, binary search, ternary values, group sparsity, structured sparsity, generative adversarial learning, artificial bee colony algorithm, and the likes. The details of these techniques can be seen from the referred papers.

Pruning necessitates numerous iterations before converging, which takes a lot of training time. It makes parameter fine-tuning exceedingly laborious and increases calculation complexity. Weight sharing would also somewhat lower the model's training accuracy (Long et al. 2019). However, pruning and quantization techniques typically offer a respectable compression rate without significantly degrading accuracy. These techniques are therefore preferable for applications that demand steady model performance (Cheng et al. 2020).

The main goal of pruning strategies is to cut out links within neurons, which immediately shrinks the model and narrows the feature map. Channel removal may significantly alter the input of the subsequent layer. Additionally, pruning techniques are effective on basic networks, including VGG and AlexNet, but are ineffective for compressing more complicated networks (e.g., ResNets) with dimensionality-dependent complex structures. Dimensionality dependencies cause ResNets' structure to be broken by filter pruning, making the network untrainable.

Low Rank Approximation

The reduction of the convolution layer would boost the rate of parameter compression and the overall speed, which serve as the inspiration for the low-rank factorization-based technique (Cheng et al. 2020). Matrix decomposition is used to estimate the informative parameters of the DL model, and layer-by-layer low rank approximation is carried out. Reduction in the depth of convolutional layers indicates that the whole model is compacted because the convolution process dominates the computational load of DL models. Any weight vector in the convolutional layer can essentially be thought of as a four-dimensional tensor, and the existence of a significant amount of redundant information in the four-dimensional tensors allows for the low-rank estimation.

A learned full rank filter bank can be approximated as combinations of a rank-1 filter basis to take advantage of the redundant representation in CNN convolutional layers. By utilizing cross-channel or filter redundancy to build a low rank basis of filters that are rank-1 in the spatial domain, Jaderberg et al. (2014) proposed strategies for accelerating convolutional layers. They used combinations of a rank-1 filter basis to mimic a learned full rank filter bank. The low-rank tensor decomposition approach devised by Tai et al. (2016) determines the precise global optimizer of the decomposition and removes redundant convolution kernels. Additionally, they introduced a technique for creating low-rank constrained CNN models from scratch, which outperformed non-constrained versions, on the challenges of overfitting and local minima. In several instances, the constrained model had a higher training error but performed better generalization, which indicated space for advancement in both numerical techniques and CNN model regularizations.

Y. D. Kim et al. (2016) introduced a one-shot compression approach using a single generic low-rank approximation method and a global rank selection strategy. The system comprised of Tucker decomposition on the kernel tensor, rank selection with variational Bayesian matrix factorization, and fine-tuning to regain aggregated performance loss. However, the method was not thoroughly examined to see whether the chosen rank is truly the best or not, hence opportunity was left for further research.

Diverging components are frequently found when training the convolutional tensors using mathematical optimization strategies. In order to achieve effective compression while

maintaining the functionality of the neural networks, Phan et al. (2020) studied degeneracy in the tensor decomposition of convolutional kernels, and offered a strategy that might stabilize the low-rank estimate of convolutional kernels. Swaminathan et al. (2020) proposed a sparse low rank approach, which sparsifies singular value decomposition matrices to obtain greater compression rate by keeping lower rank for insignificant neurons. Based on the factors, such as absolute weight, activations, or cost change, neurons were chosen for sparsification. Moreover, there was still an opportunity for further research on effective ways to combine numerous neuron selection criteria and calculate distinct sparsity rates for input and output neurons.

Yin et al. (2022) introduced a budget-aware Tucker decomposition-based compression method that effectively determines optimal tensor ranks through one-shot training. They give the deconstructed DL models the ability to automatically learn the appropriate rank from data by incorporating the rank selection into the training process. A low-rank compression technique based on tensor-train decomposition on permuted kernel weight tensor with autonomous rank determination was proposed by Gabor and Zdunek (2022). Rather than starting from scratch, the approach enables the fine-tuning of neural networks using the deconstructed variables. To expand the current method to higher order CNNs and study the compression of bigger CNNs on the ImageNet dataset, more research is required.

There other low rank approximation related approaches which can be seen from the works by X. Yu et al. (2017), Astrid and Lee (2017), H. Kim et al. (2018), Ma et al. (2019), T. Kim et al. (2020), H. Yang et al. (2020), Ruan et al. (2021), Chu and Lee (2021), Liebenwein et al. (2021), F. Yang et al. (2021), Yuchao Li et al. (2021), and Zhang et al. (2022). These works cover low rank approximation techniques based on CANDECOMP/PARAFAC (CP)-decomposition and the tensor power, low-rank kernel decomposition, channel grouping and decomposition, Bayesian optimization, global compression rate optimization, and the likes. The details of these techniques can be seen from the referred papers.

Some low-rank approximation approaches compress each layer separately rather than all at once, which deforms each layer to a variable degree (Long et al. 2019). Therefore, to find a compromise between inference precision and training pace, especially when compression ratio is high, the approaches necessitate continual fine-tuning and cross-validation. The lengthy training process and need for a large training set make fine tuning challenging.

When end-to-end solutions to a problem are required, low-rank approximation and transferable convolutional filters techniques are typically taken into account. However, it is important to remember that because these strategies are orthogonal, they can be coupled to increase the gain. For instance, in models with both convolutional and fully connected layers, we can prune the fully connected layers and compress the convolutional layers using low-rank based methods, respectively (Cheng et al. 2020).

Knowledge Distillation Approaches

A technique called knowledge distillation (KD) involves training one classifier using the results from another classifier, that is, a compressed model (student) is trained under the guidance of a more sizeable pretrained model (teacher). This is widely regarded as an effective model compression technique. By learning the class distributions generated via softmax, the major goal of KD-based techniques is to transfer knowledge from a big teacher model into a small one, that is, to duplicate the performance of a larger model using a smaller neural network with fewer parameters (Cheng et al. 2020, Frosst and Hinton 2018, Hinton et al. 2015, Jung et al. 2019). KD extends much beyond model compression, and it can be viewed as a general-purpose training approach that, in comparison to the conventional training method, is more robust to typical problems in real-world datasets (Sarfraz et al. 2020).

A Patient Knowledge Distillation technique was suggested by Sun et al. (2019) to compress an original huge BERT model (teacher) into a similarly potent lightweight shallow network (student). Through a multi-layer distillation process, the technique encouraged the student model to gradually learn from and emulate the teacher model while enabling the extraction of rich information in the teacher's hidden layers. A problem with initialization mismatch did exist, though, and it was believed that pretraining BERT from scratch would fix it.

Relative knowledge distillation, which considers the geometry of the relevant latent spaces and enables the transfer of knowledge regardless of dimension, was the focus of Lassance et al. (2020). They specifically presented a method of relative knowledge distillation that is graph-based and uses graphs to represent the geometry of latent spaces. However, more research needs to be done on finding more suitable graph distances, thoroughly examining how to grow the student network, and studying how to train a teacher network layer-by-layer.

A parallel block-wise distillation approach was put forth by Blakeney et al. (2021) to quicken the distillation of complex DL models. The approach used depth-wise separable layers as the effective replacement block architecture, took advantage of local information to perform independent block-wise distillation, and attempted to solve parallelism-limiting variables, including dependency, synchronization, and load balancing. For time series regression problems when the student and teacher are employing distinct architectures, Xu et al. (2022) presented a contrastive adversarial knowledge distillation. To automatically align the global feature distribution between student and teacher networks, they initially suggested adversarial adaptation. Then, taking care of the fine-grained features, they used a contrastive loss for instance-wise alignment between the student and teacher. Multi-staged knowledge distillation was a strategy that J. Kim et al. (2021) devised for condensing deep graph convolution networks (GCNs) to single-layered GCNs. While maintaining the multi-hop feature grouping of deep GCNs by a single effective layer, it distilled the knowledge of the aggregation from several GCN layers along with task prediction. Future research may include extending the method to take feature semantics into account.

Weight pruning and knowledge distillation were integrated by Aghli and Ribeiro (2021) for CNN compression that operated on ResNet-based regression and classification networks. To prevent damaging the ResNet architecture's network structure, they only applied the weight pruning technique to a specific number of layers. After that, they added a loss function and a knowledge distillation architecture to condense the trimmed layers during the pruning. Future research is needed because the method has not yet been used on networks trained on larger datasets such as ImageNet or on architectures other than ResNet.

In order to solve the problem of weight allocation in the knowledge distillation process, M. Zhao et al. (2022) brought knowledge distillation and weight quantization into the pruning process. The fully connected layer of the model, however, was not pruned to expedite processing, and owing to the experimental constraints, the approach had not been tested on a bigger dataset. Ji et al. (2022) suggested a KD and parameter quantization-based neural network compression technique for the detection of bearing faults. They also highlighted the challenge in the intricate selection and design of the student network's structure. Future study will be needed to determine the best way to choose and construct the structure of student networks automatically.

There other knowledge distillation related approaches are based on sequence-level KD (Y. Kim and Rush (2016)), data-free knowledge distillation which required a little amount of additional metadata to be included (Lopes et al. (2017), utilization of KD and hint learning (G. Chen et al. (2017), condensation and passing of the knowledge from a pretrained DL model (Yim (2017),

maintenance of the pairwise similarities in its own representation space (Tung and Mori (2019)), examination of the unique instance of linear and deep linear classifier (Phuong and Lampert (2019)), soft target probabilities of the training model itself (Hahn and Choi (2019), acquired ensemble information to every compressed student model (Walawalkar et al. (2020), few-sample knowledge distillation (T. Li et al. (2020), learning-during-teaching-based knowledge distillation (Xu et al. (2021)), and frequency domain learning and optimal transport theory (Binh and Woo (2022)). The details of these techniques can be seen from the referred papers.

Deeper models may become shallower through the use of knowledge distillation-based methods, which also aid in lowering computational costs. However, given that the capacities of the teacher and student models may vary significantly, their presumptions are overly rigid (Cheng et al. 2020). KD can be used to enhance efficiency in applications with small or medium-sized datasets, but not with larger datasets. Additionally, knowledge distillation techniques can only be used for problems with a softmax loss function, which restricts their application.

Universal Compression Approaches

The joint probability distribution of the input source was presumed to be known by the decoder in many researches that addressed the non-universal DL model compression problem, which is not always the case in real-world situations. As a result, universal DL model compression was developed, and its codes feature a mechanism that enables it to adjust its probability model to a variety of various input distributions. Any form of neural network should be compatible with the strategy; thus, it is not necessary to determine their distribution beforehand.

By using universal vector quantization and universal source coding, Choi et al. (2020) established universal DL model compression. They looked into universal randomized lattice quantization of DL models, which uniformly randomizes DL model weights prior lattice quantization and may operate nearly optimally on any source without requiring knowledge of its probability distribution. They also provided a technique for recovering the performance loss following quantization by fine-tuning vector quantized DL models. Theoretically, vector quantization offers a superior rate-distortion trade-off. In reality, the benefit of vector quantization is constrained for compression of a finite amount of data by the codebook overhead, which grows significantly as dimension rises and eventually takes over as the main factor degrading the compression ratio.

Wiedemann et al. (2020) proposed another universal DL model compression strategy, DeepCABAC—a general-purpose compression approach. The strategy is based on applying context-based adaptive binary arithmetic coder (CABAC) to the DL model parameters. Initially developed for the H.264/AVC video coding standard, CABAC has become the industry standard for efficient video compression. DeepCABAC used a quantization method that considered the effects of quantization on DL model performance while simultaneously minimizing a rate-distortion function.

Multimodal Compression Approaches

Deep learning-based artificial intelligence must gather and assess multimodal input to develop in comprehending the world around it. On the one hand, multimodal deep learning must provide models that can correlate and evaluate data from several modalities. On the other hand, the growth of multimodal data outpaces that of computing device speed. As a result, multimodal data fusion deep learning models trained on existing architectural devices may not be able to properly handle an increase in the volume of multimodal data during the feature structure learning phase, necessitating the use of multimodal compression techniques. The majority of studies on compression methods had focused on deep learning problems for a single modality, while very few studies have

attempted to compress DL models in multimodal fusion environments.

Using the deep autoencoder architecture, Ben Said et al. (2017) reported a hybrid compression and classification strategy of electroencephalogram and electromyography signals. The encoder-decoder layers in the architecture were created to retrieve discriminant features from the multimodal data representation and to recover the data from the latent representation. In order to accommodate multimodal data at the encoder layer, reconstruction and retrieval were included to the autoencoder. However, for low compression ratios, the strategy is less effective.

The merging of multimodal inputs was presented as a compression task by Sahu and Vechtomova (2021), where the goal was to preserve as much data from the various modalities as feasible. They put forth two adaptive methods, Auto-Fusion and GAN-Fusion, that are designed to combine multimodal inputs successfully while minimizing shallowness and computing overhead issues. To come to a firm conclusion, nevertheless, more extensive experimentation is needed to investigate the consequence of adding more modalities and reliability of multimodal features.

PERSPECTIVES, CHALLENGES, AND OPPORTUNITIES

Perspective

In DL model compression techniques, researchers have classified the methods in different perspective. There are those who classified the compression strategies base on the characteristics or nature of the strategies. For instance, parameter pruning based method, low-rank factorization, knowledge distillation (KD), and transferable convolutional filters (Cheng et al. 2020). Some have classified the strategies based on whether they are application specific or universal DL model compression techniques. Others have categorized the strategies in terms of the dimensionality, that is, whether they are for single modal or multimodal DL model compression techniques. In this work, we have reviewed the works from all those viewpoints of categorization to provide a broader exploration of available opportunities.

Multimodal data fusion is perceived to be very crucial for robust prediction and compensation in case of missing data from one of the modalities. They also provide a deeper comprehension of the underlying behavior. For instance, the healthcare provider typically uses several physiological parameters to make a precise diagnosis, and in studies of emotional computing, they combine physiological (heat rate, temperature, and the like) and physical (facial expression, voice, and the likes) modalities to come up with a good prediction of somebody's emotional state. These applications necessitate multimodal data processing, which lead to a significant increase in storage and processing power. For these applications to be used while performing deep learning network miniaturization, DL model compression is quite important.

Pruning and quantization techniques typically offer a respectable compression rate without significantly degrading accuracy. These techniques are therefore preferable for applications that demand steady model performance (Cheng et al. 2020). It is also important to remember that because these strategies are orthogonal, they can be coupled to increase the gain. For instance, in models with both convolutional and fully connected layers, we can prune the fully connected layers and compress the convolutional layers using low-rank based methods, respectively.

Challenges

DL models support many real-world applications, including scene monitoring, the plethora of biological sensors used in medical diagnostics, mobile phones and apps, robotic devices, self-driving vehicles, and similar technologies. However, there are stringent restrictions for physical size and energy consumption for these applications (Long et al. 2019). Regarding energy use and

bandwidth utilization, data delivery should be as effective and optimized as feasible. Deep networks' learning capabilities are significantly impacted by storage and computational costs. Hardware, limitations in many small platforms such as mobile, robotic, and self-driving cars, continue to be a significant obstacle to the expansion of deep CNNs.

Even though the data have diverse properties, using multiple modalities can provide a deeper comprehension of the underlying behavior. Its applications can be made in a variety of contexts, including the health sector, affective computing, and robotics. For instance, the healthcare provider typically uses several physiological parameters to make a precise diagnosis. There are other uses as well, such as emotional computing, which combines physiological and physical modalities and necessitates multimodal data processing, requiring a significant increase in storage and processing power, because these applications produce data volumes that are increasing faster than the speed of computing hardware.

Compression strategies involving layer compression poses additional difficulties that result in varying degrees of deformation in each layer, making training challenging and decreasing accuracy. Most of the cutting-edge methods currently in use are based on carefully thought-out CNN models, which have limited flexibility when it comes to changing configurations such as network architectures and hyper-parameters.

Even though compression techniques have achieved significant success, the main obstacle to adoption is still the black box mechanism. For instance, it is unclear why particular neurons or connections are pruned. Additionally, there are currently a variety of evaluation techniques used for evaluating the compression and acceleration of deep network models, and no standard measurement technique exists (Long et al. 2019).

In case of pruning techniques, they are effective on basic networks, including VGG and AlexNet, but are ineffective for compressing more complicated networks (e.g., ResNets) with dimensionality-dependent complex structures. Dimensionality dependencies cause ResNets' structure to be broken by filter pruning, making the network untrainable.

In case of low-rank approximation approaches, some of them compress each layer separately rather than all at once, which deforms each layer to a variable degree (Long et al. 2019). Therefore, to find a compromise between inference precision and training pace, especially when compression ratio is high, the approaches necessitate continual fine-tuning and cross-validation. The lengthy training process and need for a large training set make fine tuning challenging.

Other challenge can be pointed out when using KD compression approach. Deeper models may become shallower through the use of KD-based methods, which also aid in lowering computational costs. However, given that the capacities of the teacher and student models may vary significantly, their presumptions are overly rigid (Cheng et al. 2020). KD can be used to enhance efficiency in applications with small or medium-sized datasets, but not with larger datasets. Additionally, knowledge distillation techniques can only be used for problems with a softmax loss function, which restricts their application.

Opportunities

In supporting many real-world applications using DL models, in small platforms with hardware limitations there are still future research opportunities available. There are issues that need to be solved, such as how to utilize the limited computational source to its greatest potential and how to create unique compression techniques for such systems (Cheng et al. 2020). The DL models can be compressed and included in medical diagnostic tools, mobile phones and apps, robotic devices, self-driving vehicles, and the likes.

Multimodal data fusion can provide a deeper comprehension of the underlaying behaviour or scenario at hand. They can also produce

data volumes that are increasing faster than the speed of computing hardware, which necessitate multimodal DL model compression. However, there hasn't been much research on the issue of multimodal DL model compression in the context of health/mHealth, affective computing, and similar topics. These kinds of applications have a significant demand for multimodal compression techniques in order to speed up network computation and make it easier to use compact platforms. There is an opportunity for researches to modify successful single-modality compression techniques to fit into these multimodal data environments.

In the case of layer compression, we have seen that it poses additional difficulties that result in varying degrees of deformation in each layer, which makes training challenging and decreasing accuracy. With this challenge, there is a research opportunity to find an adaptable compression technique depending on the various layer situations.

Most of cutting-edge methods currently in use are based on carefully thought-out CNN models. However, CNN models have limited flexibility when it comes to changing configurations such as network architectures and hyper-parameters. Future work should offer more logical configuration options for the compressed models to perform more challenging tasks.

There is also an opportunity of exploring the interpretability of knowledge behind significant success of DL model compression. It is still the black box mechanism, for example, it is unclear why pruning particular connections or neurons (Cheng et al. 2020). Additionally, future studies must put out a single evaluation standard that can be used for various models of various data sets.

CONCLUSION

In this paper, we examine various DL model compression techniques with a goal of exploring over numerous DL model compression methods that have advanced in a number of applications. We used the articles for the review based on foundational studies for DL model compression techniques, including early development and ground-breaking DL model compression algorithms, as well as the recent studies relevant to the subject. Techniques for data compression can be categorized based on their characteristics: parameter pruning and quantization, low-rank factorization, knowledge distillation (KD), universal compression, and multimodal compression-based approaches. Most of the articles related to DL model compression techniques are discussing pruning and quantization-based approaches (about 58% of the reviewed articles in this paper). Its popularity comes from the fact that, pruning and quantization of the DL models is one of the earliest approaches to be used in DL model compression techniques. However, it is worth noting that, the majority of these techniques operate independently, but may generate outstanding results if integrated.

We have discussed the challenges, opportunities, and developments in DL model compression strategies, and realized that the field is still expanding, particularly in applications involving multimodal fusion. Multimodal deep learning is generating amounts of data that are growing faster than the speed of computing hardware. As a result, multimodal data fusion deep learning models trained on current architectural devices may not be able to effectively handle an increase in the volume of multimodal data throughout the feature structure learning process. Multimodal DL model compression techniques are, thus, one of viable areas for further study in addressing the problem.

REFERENCES

Abrahamyan L, Chen Y, Bekoulis G, and Deligiannis N 2021 Learned Gradient Compression for Distributed Deep Learning. *IEEE Transactions on Neural Networks and Learning Systems*. 1–1, DOI: 10.1109/TNNLS.2021.3084806.

Aghli N and Ribeiro E 2021 Combining weight pruning and knowledge

distillation for CNN compression. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. 3185–3192.

Astrid M and Lee SI 2017 CP-decomposition with Tensor Power Method for Convolutional Neural Networks compression. *2017 IEEE International Conference on Big Data and Smart Computing, BigComp 2017*. (1): 115–118, DOI: 10.1109/BIGCOMP.2017.7881725.

Ben Said A, Mohamed A, Elfouly T, Harras K, and Wang ZJ 2017 Multimodal deep learning approach for Joint EEG-EMG Data compression and classification. *IEEE Wireless Communications and Networking Conference, WCNC*, DOI: 10.1109/WCNC.2017.792570 9.

Binh LM and Woo S 2022 ADD: Frequency Attention and Multi-View Based Knowledge Distillation to Detect Low-Quality Compressed Deepfake Images. *Proceedings of the AAAI Conference on Artificial Intelligence*. 36(1): 122–130, https://doi.org/10.1609/aaai.v36i1.19886.

Blakeney C, Li X, Yan Y, and Zong Z 2021 Parallel Blockwise Knowledge Distillation for Deep Neural Network Compression. *IEEE Transactions on Parallel and Distributed Systems*. 32(7): 1765–1776, DOI: 10.1109/TPDS.2020.3047003.

Chen G, Choi W, Yu X, Han T, and Chandraker M 2017 Learning efficient object detection models with knowledge distillation. *Advances in Neural Information Processing Systems*. 2017-Decem(Nips): 743–752.

Chen W, Edu JWW, Cse C, and Edu W 2015 Compressing Neural Networks with the Hashing Trick. *Proceedings of the 32nd International Conference on Machine Learning, PMLR*. 37: 2285–2294.

Chen Z, Xu TB, Du C, Liu CL, and He H 2021 Dynamical Channel Pruning by Conditional Accuracy Change for Deep Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*. 32(2): 799–813, DOI: 10.1109/TNNLS.2020.2979517.

Cheng Y, Wang D, Zhou P, and Zhang T 2020 A Survey of Model Compression and Acceleration for Deep Neural Networks. *ArXiv*. 1–10. https://doi.org/10.48550/arXiv.1710.09282

Chin TW, Ding R, Zhang C, and Marculescu D 2020 Towards efficient model compression via learned global ranking. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 1515–1525, https://doi.org/10.48550/arXiv.1904.12368.

Choi Y, El-Khamy M, and Lee J 2020 Universal Deep Neural Network Compression. *IEEE Journal on Selected Topics in Signal Processing*. 14(4): 715–726, DOI: 10.1109/JSTSP.2020.2975903.

Chu B-S and Lee C-R 2021 Low-rank Tensor Decomposition for Compression of Convolutional Neural Networks Using Funnel Regularization. *Association for the Advancement of Artificial Intelligence*. https://doi.org/10.48550/arXiv.2112.03690.

Courbariaux M and David J 2015 BinaryConnect : Training Deep Neural Networks with binary weights during propagations. In *Advances in Neural Information Processing Systems* 3123–3131. Montreal, Quebec, Canada.

Courbariaux M and Hubara I 2016 Binarized Neural Networks: Training Neural Networks with Weights and Activations Constrained to +1 or − 1. *CoRR*. abs/1602.0, https://doi.org/10.48550/arXiv.1602.02830.

Denil M, Shakibi B, Dinh L, Ranzato M, and De Freitas N 2013 Predicting parameters in deep learning. *Advances in Neural Information Processing Systems*. 1–9.

Ding X, Ding G, Guo Y, and Han J 2019 Centripetal SGD for pruning very deep convolutional networks with complicated structure. *Proceedings of the IEEE Computer Society*

Conference on Computer Vision and Pattern Recognition. 2019-June: 4938–4948.

Ding X, Ding G, Guo Y, Han J, and Yan C 2019 Approximated oracle filter pruning for destructive CNN width optimization. *36th International Conference on Machine Learning, ICML 2019*. 2019-June: 2899–2909, http://prceedings.mlr.press/v97/ding19a.html.

Dong X and Yang Y 2019 Network pruning via transformable architecture search. *Advances in Neural Information Processing Systems*. 32(NeurIPS): 1–12.

Frankle J and Carbin M 2019 The lottery ticket hypothesis: Finding sparse, trainable neural networks. *7th International Conference on Learning Representations, ICLR 2019*. 1–42, https://doi.org/10.48550/arXiv.1803.03635.

Frosst N and Hinton G rey 2018 Distilling a neural network into a soft decision tree. *CEUR Workshop Proceedings*. 2071. https://doi.org/10.48550/arXiv.1711.09784

Gabor M and Zdunek R 2022 Convolutional Neural Network Compression via Tensor-Train Decomposition on Permuted Weight Tensor with Automatic Rank Determination. *Computational Science – ICCS 2022. ICCS 2022. Lecture Notes in Computer Science*. 13352 LNCS: 654–667, https://doi.org/10.1007/978-3-031-08757-8_54.

Gong Y, Liu L, Yang M, and Bourdev L 2014 Compressing Deep Convolutional Networks using Vector Quantization 1–10. Retrieved from https://doi.org/10.48550/arXiv.1412.6115

Guo S, Wang Y, Li Q, and Yan J 2020 DMCP: Differentiable markov channel pruning for neural networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 1536–1544, https://doi.org/10.48550/arXiv.2005.03354.

Gupta S, Agrawal A, Gopalakrishnan K, Heights Y, Narayanan P, and Jose S 2015 Deep Learning with Limited Numerical Precision. *Proceedings of the 32nd International Conference on Machine Learning*. 37: 1737–1746. Retrieved from https://proceedings.mlr.press/v37/gupta15.html

Hahn S and Choi H 2019 Self-knowledge distillation in natural language processing. *International Conference Recent Advances in Natural Language Processing, RANLP*. 2019-Septe: 423–430, https://doi.org/10.48550/arXiv.1908.01851.

Han S, Mao H, and Dally WJ 2016 Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*. 1–14, https://doi.org/10.48550/arXiv.1510.00149.

Han S, Pool J, Tran J, and Dally WJ 2015 Learning both Weights and Connections for Efficient Neural Networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, 1–9, https://proceedings.neurips.cc/paper_files/paper/2015/file/ae0eb3eed39d2bcef4622b2499a05fe6-Paper.pdf.

Hansont SJ and Pratt LY 1989 Comparing Biases for Minimal Network Construction with Back-Propagation. *Advances in Neural Information Processing Systems*. 177–185, https://proceedings.neurips.cc/paper_files/paper/1988/file/1c9ac0159c94d8d0cbedc973445af2da-Paper.pdf.

Hao L, Asim K, Durdanovic I, Hanan S, and Graf HP 2017 Pruning filters for efficient convnets. In *ICLR 2017* 1–13, https://doi.org/10.48550/arXiv.1608.08710.

Hassibi B, Stork DG, Road SH, and Park M 1993 Second order derivatives for network pruning : Optimal Brain Surgeon. *Advances in Neural Information Processing Systems 5*. 164–171, Retrieved from https://proceedings.neurips.cc/paper_

files/paper/1992/file/303ed4c69846ab36c2904d3ba8573050-Paper.pdf.

He Y, Kang G, Dong X, Fu Y, and Yang Y 2018 Soft filter pruning for accelerating deep convolutional neural networks. *IJCAI International Joint Conference on Artificial Intelligence*. 2018-July: 2234–2240, https://doi.org/10.48550/arXiv.1808.06866.

He Y, Liu P, Wang Z, Hu Z, and Yang Y 2019 Filter pruning via geometric median for deep convolutional neural networks acceleration. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2019-June: 4335–4344.

He Z and Fan D 2019 Simultaneously optimizing weight and quantizer of ternary neural network using truncated gaussian approximation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2019-June: 11430–11438.

Hinton G, Vinyals O, and Dean J 2015 Distilling the Knowledge in a Neural Network 1–9. Retrieved from http://doi.org/10.48550/arXiv.1503.02531

Hou L, Yao Q, and Kwok JT 2018 Loss-aware binarization of deep networks. *CoRR*. 1–11. Retrieved from https://doi.org/10.48550/arXiv.1611.01600

Jaderberg M, Vedaldi A, and Zisserman A 2014 Speeding up convolutional neural networks with low rank expansions. *BMVC 2014 - Proceedings of the British Machine Vision Conference 2014*, https://doi.org/10.48550/arXiv.1405.3866.

Ji M, Peng G, Li S, Cheng F, Chen Z, Li Z, and Du H 2022 A neural network compression method based on knowledge-distillation and parameter quantization for the bearing fault diagnosis. *Applied Soft Computing*. 127: 109331. Retrieved from https://doi.org/10.1016/j.asoc.2022.109331

Jung J, Heo H, Shim H, and Yu H-J 2019 Distilling the Knowledge of Specialist Deep Neural Networks in Acoustic Scene Classification 1(October): 114–118, https://doi.org/10.33682/gqpj-ac63.

Kim H, Karim MU, and Kyung C-M 2018 Efficient Neural Network Compression. *CVPR*. 12569–12577. Retrieved from https://doi.org/10.48550/arXiv.1811.12781

Kim J, Jung J, and Kang U 2021 Compressing deep graph convolution network with multi-staged knowledge distillation. *PLoS ONE*. 16(8 August): 1–18. Retrieved from http://dx.doi.org/10.1371/journal.pone.0256187

Kim T, Lee J, and Choe Y 2020 Bayesian optimization-based global optimal rank selection for compression of convolutional neural networks. *IEEE Access*. 8: 17605–17618, DOI: 10.1109/ACCESS.2020.2968357.

Kim Y and Rush AM 2016 Sequence-level knowledge distillation. *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*. 1317–1327, https://doi.org/10.48550/arXiv.1606.07947.

Kim YD, Park E, Yoo S, Choi T, Yang L, and Shin D 2016 Compression of deep convolutional neural networks for fast and low power mobile applications. *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*. 1–16, https://doi.org/10.48550/arXiv.1511.06530.

Kusupati A, Ramanujan V, Somani R, Wortsman M, Jain P, Kakade S, and Farhadi A 2020 Soft threshold weight reparameterization for learnable sparsity. *37th International Conference on Machine Learning, ICML 2020*. PartF16814: 5500–5511, http://proceedings.mlr.press/v119/kusupati20a/kusupati20a.pdf.

Lassance C, Bontonou M, Hacene GB, Gripon V, Tang J, and Ortega A 2020 Deep Geometric Knowledge Distillation with Graphs. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal*

Processing - Proceedings. 2020-May: 8484–8488, DOI: 10.1109/ICASSP40776.2020.9053986.

Lebedev V and Lempitsky V 2016 Fast ConvNets Using Group-wise Brain Damage. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016* 2554–2564.

Lemaire C, Achkar A, and Jodoin PM 2019 Structured pruning of neural networks with budget-aware regularization. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2019-June: 9100–9108.

Leng C, Dou Z, Li H, Zhu S, and Jin R 2018 Extremely low bit neural network: Squeeze the last bit out with ADMM. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*. 3466–3473, https://doi.org/10.1609/aaai.v32i1.11713.

Li T, Li J, Liu Z, and Zhang C 2020 Few sample knowledge distillation for efficient network compression. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 14627–14635, https://doi.org/10.48550/arXiv.1812.01839.

Li Yawei, Gu S, Mayer C, Van Gool L, and Timofte R 2020 Group Sparsity: The Hinge between Filter Pruning and Decomposition for Network Compression. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 8015–8024, https://doi.org/10.48550/arXiv.2003.08935.

Li Yuchao, Lin S, Liu J, Ye Q, Wang M, Chao F, Yang F, Ma J, Tian Q, and Ji R 2021 Towards Compact CNNs via Collaborative Compression Multi-step Heuristic. *CVPR*. 6438–6447, https://doi.org/10.48550/arXiv.2105.11228.

Liebenwein L, Maalouf A, Gal O, Feldman D, and Rus D 2021 Compressing Neural Networks: Towards Determining the Optimal Layer-wise Decomposition. *Advances in Neural Information Processing Systems*. 7(NeurIPS): 5328–5344, Retrieved from https://proceedings.neurips.cc/paper_files/paper/2021/file/2adcfc3929e7c03fac3100d3ad51da26-Paper.pdf.

Lin M, Cao L, Li S, Ye Q, Tian Y, Liu J, Tian Q, and Ji R 2021 Filter Sketch for Network Pruning. *IEEE Transactions on Neural Networks and Learning Systems*. 1–10, DOI: 10.1109/TNNLS.2021.3084206.

Lin M, Ji R, Wang Y, Zhang Y, Zhang B, Tian Y, and Shao L 2020 Hrank: Filter pruning using high-Rank feature map. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 1526–1535, https://doi.org/10.48550/arXiv.2002.10179.

Lin M, Ji R, Zhang Y, Zhang B, Wu Y, and Tian Y 2020 Channel pruning via automatic structure search. *IJCAI International Joint Conference on Artificial Intelligence*. 2021-Janua: 673–679, https://doi.org/10.48550/arXiv.2001.08565.

Lin S, Ji R, Li Y, Deng C, and Li X 2020 Toward Compact ConvNets via Structure-Sparsity Regularized Filter Pruning. *IEEE Transactions on Neural Networks and Learning Systems*. 31(2): 574–588, DOI: 10.1109/TNNLS.2019.2906563.

Lin S, Ji R, Yan C, Zhang B, Cao L, Ye Q, Huang F, and Doermann D 2019 Towards optimal structured CNN pruning via generative adversarial learning. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2019-June: 2785–2794.

Lin Z, Memisevic R, and Courbariaux M 2016 Neural networks with few multiplications. *CoRR*. 1–9. https://doi.org/10.48550/arXiv.1510.03009.

Liu B, Wang M, Foroosh H, Tappen M, and Penksy M 2015 Sparse Convolutional Neural Networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 07-12-June: 806–814.

Liu Z, Mu H, Zhang X, Guo Z, Yang X, Cheng KT, and Sun J 2019 MetaPruning: Meta learning for automatic neural network channel pruning. *Proceedings of the IEEE International Conference on Computer Vision*. 2019-Octob: 3295–3304.

Long X, Ben Z, and Liu Y 2019 A Survey of Related Research on Compression and Acceleration of Deep Neural Networks. *Journal of Physics: Conference Series*. 1213(5): 1–8, DOI 10.1088/1742-6596/1213/5/052003.

Lopes RG, Fenu S, and Starner T 2017 Data-Free Knowledge Distillation for Deep Neural Networks. *NIPS 2017*. http://doi.org/10.48550/arXiv.1710.07535

Luo J-H, Wu J, and Lin and W 2017 ThiNet: A Filter Level Pruning Method for Deep Neural Network Compression. *IEEE International Conference on Computer Vision (ICCV), 2017*. 5058–5066, https://doi.org/10.48550/arXiv.1707.06342.

Ma Y, Chen R, Li W, Shang F, Yu W, Cho M, and Yu B 2019 A Unified Approximation Framework for Compressing and Accelerating Deep Neural Networks. *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, DOI: 10.1109/ICTAI.2019.00060. Retrieved from http://arxiv.org/abs/1807.10119

Meng F and Cheng H 2020 Pruning Filter in Filter. In *34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.* 1–12.

Merolla P, Appuswamy R, Arthur J, Esser SK, and Modha D 2016 Deep neural networks are robust to weight binarization and other non-linear distortions. *Neural and Evolutionary Computing*. https://doi.org/10.48550/arXiv.1606.01981.

Molchanov P, Mallya A, Tyree S, Frosio I, and Kautz J 2019 Importance estimation for neural network pruning. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2019-June(11264): 11256–11264.

Nasif A, Othman ZA, and Sani NS 2021 The deep learning solutions on lossless compression methods for alleviating data load on IoT nodes in smart cities. *Sensors*. 21(12), https://doi.org/10.3390/s21124223.

Peng H, Wu J, Chen S, and Huang J 2019 Collaborative channel pruning for deep networks. *36th International Conference on Machine Learning, ICML 2019*. 2019-June: 8947–8957. Retrieved from https://proceedings.mlr.press/v97/peng19c.html

Phan AH, Sobolev K, Sozykin K, Ermilov D, Gusak J, Tichavský P, Glukhov V, Oseledets I, and Cichocki A 2020 Stable Low-Rank Tensor Decomposition for Compression of Convolutional Neural Network. *Computer Vision – ECCV 2020. ECCV 2020. Lecture Notes in Computer Science()*. 12374 LNCS: 522–539, DOI: 10.1007/978-3-030-58526-6_31.

Phuong M and Lampert CH 2019 Towards understanding knowledge distillation. *36th International Conference on Machine Learning, ICML 2019*. 2019-June (2014): 8993–9007. Retrieved from https://proceedings.mlr.press/v97/phuong19a.html

Predić B, Vukić U, Saračević M, Karabašević D, and Stanujkić D 2022 The Possibility of Combining and Implementing Deep Neural Network Compression Methods. *Axioms*. 11(5), https://doi.org/10.3390/axioms11050229.

Rastegari M, Ordonez V, and Redmon J 2016 XNOR-Net : ImageNet Classification Using Binary Convolutional Neural Networks 1: 525–542, https://doi.org/10.1007/978-3-319-46493-0_32.

Ruan X, Liu Y, Yuan C, Li B, Hu W, Li Y, and Maybank S 2021 EDP: An Efficient Decomposition and Pruning Scheme for Convolutional Neural Network Compression. *IEEE Transactions on Neural Networks*

and Learning Systems. 32(10): 4499–4513, DOI: 10.1109/TNNLS.2020.3018177.

Sahu G and Vechtomova O 2021 Adaptive fusion techniques for multimodal data. *EACL 2021 - 16th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference*. 3156–3166, https://doi.org/10.48550/arXiv.1911.03821.

Sarfraz F, Arani E, and Zonooz B 2020 Knowledge distillation beyond model compression. *Proceedings - International Conference on Pattern Recognition*. 6181–6188, DOI: 10.1109/ICPR48806.2021.9413016.

Srinivas S, Babu RV, and Education S 2015 Data-free Parameter Pruning for Deep Neural Networks. In *Proceedings of the British Machine Vision Conference 2015, BMVC 2015, Swansea, UK, September 7-10, 2015* 1–12, https://doi.org/10.48550/arXiv.1507.06149.

Sun S, Cheng Y, Gan Z, and Liu J 2019 Patient knowledge distillation for BERT model compression. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*. 4323–4332, https://doi.org/10.48550/arXiv.1908.09355.

Swaminathan S, Garg D, Kannan R, and Andres F 2020 Sparse low rank factorization for deep neural network compression. *Neurocomputing*. 398: 185–196. Retrieved from https://doi.org/10.1016/j.neucom.2020.02.035

Taco SC and Max W 2016 Group Equivariant Convolutional Networks. *Proceedings of The 33rd International Conference on Machine Learning, PMLR 48 New York, NY, USA, 2016*. 48, Retrieved from https://proceedings.mlr.press/v48/cohenc16.html.

Tai C, Xiao T, Zhang Y, Wang X, and Weinan E 2016 Convolutional neural networks with low-rank regularization. *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*. 1(2014): 1–11, https://doi.org/10.48550/arXiv.1511.16067.

Tung F and Mori G 2019 Similarity-Preserving Knowledge Distillation. In *IEEE/CVF International Conference on Computer Vision (ICCV)* 1365–1374. Retrieved from https://openaccess.thecvf.com/content_ICCV_2019/papers/Tung_Similarity-Preserving_Knowledge_Distillation_ICCV_2019_paper.pdf

Ullrich K, Welling M, and Meeds E 2017 Soft weight-sharing for neural network compression. *ICLR2017*. abs/1702.0: 1–16, https://doi.org/10.48550/arXiv.1702.04008.

Vanhoucke V, Senior A, and Mao MZ 2011 Improving the speed of neural networks on CPUs. *Deep Learning and Unsupervised Feature Learning Workshop, NIPS 2011*. 1–8. Retrieved from https://storage.googleapis.com/pub-tools-public-publication-data/pdf/37631.pdf.

Walawalkar D, Shen Z, and Savvides M 2020 Online Ensemble Model Compression Using Knowledge Distillation. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 12364 LNCS: 18–35, DOI: 10.1007/978-3-030-58529-7_2.

Wang C, Grosse R, Fidler S, and Zhang G 2019 Eigendamage: Structured pruning in the Kronecker-factored eigenbasis. *36th International Conference on Machine Learning, ICML 2019*. 2019-June: 11398–11407, Available from https://proceedings.mlr.press/v97/wang19g.html.

Wang H, Zhang Q, Wang Y, Yu L, and Hu H 2019 Structured Pruning for

Efficient ConvNets via Incremental Regularization. *Proceedings of the International Joint Conference on Neural Networks*. 2019-July, DOI: 10.1109/IJCNN.2019.8852463.

Wang P, He X, Chen Q, Cheng A, Liu Q, and Cheng J 2021 Unsupervised Network Quantization via Fixed-Point Factorization. *IEEE Transactions on Neural Networks and Learning Systems*. 32(6): 2706–2720, DOI: 10.1109/TNNLS.2020.3007749.

Wen W, Wu C, Wang Y, Chen Y, and Hai L 2016 Learning Structured Sparsity in Deep Neural Network. *Advances in Neural Information Processing Systems 29 (NIPS 2016)*. (Nips), Retrieved from https://proceedings.neurips.cc/paper/2016/hash/41bfd20a38bb1b0bec75acf0845530a7-Abstract.html.

Wiedemann S, Kirchhoffer H, Matlage S, Haase P, Marban A, Marinč T, Neumann D, Nguyen T, Schwarz H, Wiegand T, et al. 2020 DeepCABAC: A Universal Compression Algorithm for Deep Neural Networks. *IEEE Journal on Selected Topics in Signal Processing*. 14(4): 700–714, DOI: 10.1109/JSTSP.2020.2969554.

Wu J, Leng C, Wang Y, Hu Q, and Cheng J 2016 Quantized Convolutional Neural Networks for Mobile Devices. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4820–4828.

Xu Q, Chen Z, Ragab M, Wang C, Wu M, and Li X 2022 Contrastive adversarial knowledge distillation for deep model compression in time-series regression tasks. *Neurocomputing*. 485(Xiaoli Li): 242–251, https://doi.org/10.1016/j.neucom.2021.04.139.

Xu Q, Chen Z, Wu K, Wang C, Wu M, and Li X 2021 KDnet-RUL: A Knowledge Distillation Framework to Compress Deep Neural Networks for Machine Remaining Useful Life Prediction. *IEEE Transactions on Industrial Electronics*. (June), DOI: 10.1109/TIE.2021.3057030.

Yang F, Liu W, Liu J, Liu C, Mi Y, and Song H 2021 Iterative low-rank approximation based on the redundancy of each network layer. In *Proc. SPIE 11720, Twelfth International Conference on Graphics and Image Processing (ICGIP 2020)*, https://doi.org/10.1117/12.2589425

Yang H, Tang M, Wen W, Yan F, Hu D, Li A, Li H, and Chen Y 2020 Learning low-rank deep neural networks via singular vector orthogonality regularization and singular value sparsification. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. 2020-June: 2899–2908.

Yim J 2017 A Gift from Knowledge Distillation. *CVPR*. 4133–4141. Retrieved from https://openaccess.thecvf.com/content_cvpr_2017/papers/Yim_A_Gift_From_CVPR_2017_paper.pdf

Yin M, Phan H, Zang X, Liao S, and Yuan B 2022 BATUDE: Budget-Aware Neural Network Compression Based on Tucker Decomposition. *Proceedings of the AAAI Conference on Artificial Intelligence*. 36(8): 8874–8882, https://doi.org/10.1609/aaai.v36i8.20869.

You Z, Yan K, Ye J, Ma M, and Wang P 2019 Gate decorator: Global filter pruning method for accelerating deep convolutional neural networks. *Advances in Neural Information Processing Systems*. 32(NeurIPS): 1–12.

Yu J and Huang T 2019 AutoSlim: Towards One-Shot Architecture Search for Channel Numbers. *Computer Vision and Pattern Recognition (CVPR)*. Retrieved from https://doi.org/10.48550/arXiv.1903.11728

Yu X, Liu T, Wang X, and Tao D 2017 On compressing deep models by low rank and sparse decomposition. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*. 2017-Janua: 67–76.

Zhang H, Liu L, Zhou H, Sun H, and Zheng N 2022 CMD: controllable matrix decomposition with global optimization for deep neural network compression. *Machine Learning*.

111(3): 831–851. Retrieved from https://doi.org/10.1007/s10994-021-06077-5

Zhao C, Ni B, Zhang J, Zhao Q, Zhang W, and Tian Q 2019 Variational convolutional neural network pruning. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2019-June: 2775–2784.

Zhao M, Li M, Peng SL, and Li J 2022 A Novel Deep Learning Model Compression Algorithm. *Electronics (Switzerland)*. 11(7): 1–12, https://doi.org/10.3390/electronics11071066.

Zhaowei Cai, Xiaodong He, Jian Sun NV 2017 Deep Learning with Low Precision by Half-wave Gaussian Quantization. In *Computer Vision and Pattern Recognition* (*CVPR*) 5918–5926, https://doi.org/10.48550/arXiv.1702.00953.

Zhou H, Alvarez JM, and Porikli F 2016 Less Is More: Towards Compact CNNs. In *European Conference on Computer Vision* Vol. 1, 662–677. Amsterdam, the Netherlands, https://doi.org/10.1007/978-3-319-46493-0_40.