UDC: 004.032.26:004.89:004.93 https://doi.org/10.15407/jai2026.01.059

A. Kozynets[1], L. Demkiv[2]
[1,2]Ivan Franko National University of Lviv, Ukraine
Universytetska St., 1, Lviv, 79000
[1]andrian.kozynets@gmail.com
[2]lidia.demkiv@gmail.com
[1]https://orcid.org/0009-0004-7994-3534
[2]https://orcid.org/0000-0003-4670-5834

RESEARCH ON TWO-STAGE KNOWLEDGE DISTILLATION AND HYBRID COMPRESSION OF CONVOLUTIONAL NEURAL NETWORKS TO REDUCE THEIR COMPUTATIONAL COMPLEXITY

Abstract. The article considers the problem of effective knowledge transfer between deep convolutional neural networks of different capacities for their further deployment on hardware platforms with limited computing resources. The main problem of standard knowledge distillation protocols is the occurrence of "gradient shock" during the initialization of the student model on specific data sets, which leads to the destruction of the feature space and the loss of final accuracy. To overcome this limitation, the Two-Stage Distillation algorithm was developed and implemented. The proposed approach divides the learning process into the classifier stabilization phase and the deep distillation phase. Experimental research was conducted on the ResNet and VGG architectural families. The results obtained confirm that the use of the proposed algorithm allows to increase the accuracy of compact models by 1.5–2.6% compared to standard training. In addition, the work investigated and experimentally confirmed the phenomenon of "distillation recovery" - the ability of the algorithm to restore the accuracy of the model after aggressive structural pruning. It is proven that the use of "soft goals" of the teacher in a narrowed search space allows the sparse ResNet18 model to achieve an accuracy of 77.8%, which exceeds the basic full-size student model.

Keywords: knowledge distillation, neural networks, gradient shock, model compression, ResNet, VGG.

Introduction

Modern convolutional neural networks, such as VGG [1] and ResNet [2], provide high accuracy in computer vision tasks due to the formation of complex multi-layer feature representations. However, their significant over-parameterization creates a critical computational barrier for deployment on edge devices with limited memory [3]. One of the most fundamental approaches to solving this problem is knowledge distillation, which involves training a compact student model to mimic the behavior of a bulky teacher model by using "soft targets" and transferring hidden features.

The development of this direction has led to the emergence of complex optimization methods, such as relational distillation of spatial links [4] and the use of intermediate teacher-assistant models to smooth the architectural gap [5]. However, despite theoretical effectiveness, the practical implementation of these algorithms faces two significant technical problems:

1. *The problem of "Gradient shock" during initialization.* When transferring knowledge from a teacher to a student on new datasets, the student's output classifier is typically initialized randomly. In the initial stages of training, the radical inconsistency between the pre-trained feature extractors in the model's body and the random classifier generates extremely large error gradient values. The backpropagation of these massive gradients destabilizes and destroys the already formed weight coefficients in the deep convolutional layers, which leads to the model getting stuck in suboptimal local minima.

2. *Topology healing efficiency.* As noted in modern studies of architectural compression [6], the aggressive removal of structural elements drastically narrows the parameter space of the model. At the same time, standard fine-tuning using the usual cross-entropy function is often unable to restore lost information links or overestimates the significance of the remaining weights [7]. The use of distillation not only as a "from scratch" training method but also as a healing mechanism in the narrowed search space after pruning is a critically important but insufficiently formalized algorithmic task.

Thus, there is an objective need to develop a specialized two-stage distillation

algorithm that would minimize the destructive impact of initial gradients and ensure the stable convergence of compact and pruned models to metrics comparable to reference architectures.

Problem Statement

Traditional knowledge distillation involves the simultaneous minimization of a combined loss function consisting of cross-entropy and Kullback-Leibler divergence. However, this approach has a critical algorithmic flaw when applied to knowledge transfer tasks to new domains.

The essence of the problem lies in the architectural asymmetry during initialization. The body of the student model may contain partially pre-trained weights or be initialized using complex methods, while the output classifier is always initialized with random values for the new class space. In the initial training iterations, the random classifier generates absolutely incorrect logits. According to studies on the statistical properties of distillation [8], when calculating the error gradient, this initial classifier error is exponentially amplified by the KL divergence component. As a result, backpropagation transmits massive chaotic gradients to the deep convolutional layers, which destroys the spatial hierarchies of features even before the classifier has time to adapt. This negates the benefits of knowledge transfer, as evidenced by convergence problems given a large architectural gap between models [9].

The second, closely related problem is the process of restoring the representative capacity of the network after applying structural pruning methods. The removal of entire filters or channels radically changes the geometry of the parameter space. As the "lottery ticket" hypothesis shows [10], the success of training a pruned network critically depends on the initial weight configuration. Standard fine-tuning of a pruned model using hard labels is often ineffective, since the narrowed architecture loses its generalization ability. In this context, a mechanism is needed that would use the teacher's spatial attention maps [11] as a navigator for the student's optimizer in the limited parameter space.

In view of the above, the scientific and practical problem of this research lies in the need to develop, formalize, and verify a modified knowledge distillation algorithm that would:

– Algorithmically isolate the process of initial classifier initialization from the process of deep feature transfer to eliminate "gradient shock".

– Provide an effective mechanism for transferring "dark knowledge" for both fully-fledged compact models and pruned topologies, in order to achieve a Pareto-efficient balance between computational complexity and accuracy on devices with limited computational resources.

Analysis of recent research and publications

The basic principles of knowledge transfer between convolutional neural networks have received significant development in the direction of complicating the topology of the transferred features. In particular, work [04] proposes a method of relational distillation, which focuses on transferring structural relationships between objects, and not just individual probabilities. The authors of [11] developed a method for transferring spatial attention maps, which forces the student to activate the same regions of convolutional filters as the teacher.

To solve the problem of a significant architectural gap between models, work [5] introduced the concept of a "teacher assistant", which involves gradual, cascaded training through an intermediate medium-sized model. In the field of reducing model dimensionality, studies [6] systematize structural pruning methods, pointing to an inevitable degradation in accuracy when removing more than 30% of the weights. At the same time, the "lottery ticket" hypothesis, conceptualized in [10], proves that the ability of a pruned model to learn critically depends on its initial initialization.

Research Objective

The objective of this work is the development, theoretical justification, and experimental verification of a two-stage knowledge distillation algorithm to overcome the problem of "gradient shock" during initialization and ensure the effective

restoration of the representative capacity of convolutional neural networks after structural compression.

To achieve the set objective, the following set of scientific and practical tasks has been formulated and solved:

1. *Development and mathematical formalization of the distillation algorithm.* Create a two-phase training protocol that algorithmically isolates the stage of preliminary classifier stabilization using label smoothing from the stage of deep transfer of hidden features through the minimization of Kullback-Leibler divergence.

2. *Analysis of architectural sensitivity to feature transfer.* Experimentally investigate the impact of neural network topology on distillation efficiency by comparing networks with a linear structure and high parametric redundancy with compact networks with residual connections.

3. *Verification of the "Distillation Healing" mechanism.* Investigate and quantitatively evaluate the ability of the developed distillation algorithm to act as a healing mechanism and optimizer navigator in the narrowed parameter space after applying destructive pruning methods.

4. *Hardware-Aware Evaluation.* Perform a cross-platform performance analysis of the developed distilled pipelines on real computing hardware (CPU/GPU) to confirm their suitability in resource-constrained systems.

Presentation of the Main Research Material

1. Mathematical formalization of the "Two-Stage Smart Distillation" algorithm.

One of the key problems during Knowledge Distillation to new domain areas is the phenomenon of "gradient shock". It arises due to the inconsistency of pre-trained feature detectors in the model body and the randomly initialized weights of the new classifier. Large gradient values in the first training iterations lead to the destabilization of weights in deep layers, which reduces the final accuracy.

To solve this problem, the *Two-Stage Distillation* algorithm was developed, which implements a strategy of gradual adaptation:

Stage 1: Stabilization

At this stage, the parameters of the model body θ_{body} are frozen (*requires_grad=False*), and training undergoes exclusively the output layer θ_{head}. The loss function at this stage includes regularization through label smoothing to prevent overfitting:

$$L_{warmup} = (1-\epsilon)H(y,p) + \epsilon H(u,p)$$

where L_{warmup} - is the loss function value at the stabilization stage; $\epsilon = 0.1$ - is the label smoothing coefficient; $H(y,p)$ - is cross-entropy; y - is is the probability distribution predicted by the model; $H(u,p)$ - is the cross-entropy between the uniform distribution and the model's prediction; u - is the uniform distribution. The use of a high learning rate ($lr = 10^{-3}$) for 5 epochs allows for the rapid adaptation of the linear layer to the teacher's feature space.

Stage 2: Deep distillation

After stabilization, the entire network is unfrozen. Training occurs by minimizing the combined loss function, which takes into account the Kullback-Leibler (KL) divergence between the teacher's and student's outputs:

$$L_{total} = (1-\alpha)L_{CE}(y,\sigma(z_s)) + \alpha T^2 L_{KL}(\sigma(\tfrac{z_s}{T}),\sigma(\tfrac{z_t}{T}))$$

where L_{total} - is the total loss function; α - is the balancing coefficient; L_{CE} - is the cross-entropy loss function; y - is the true class label from the training dataset; σ - is the Softmax activation function; z_s - are the student's logits; T - is the distillation temperature; L_{KL} - is the Kullback-Leibler divergence; z_t - are the teacher's logits. Distillation parameters were established empirically: $T = 4.0$ (for maximum distribution entropy and transfer of "dark knowledge") and the balance coefficient $\alpha = 0.7$. Optimization is performed by the AdamW algorithm with cosine learning rate decay, which ensures smooth convergence to the global minimum.

2. Classification of the investigated distillation and compression scenarios

Table 1. List of experimental optimization methods

Method No.	Method Name
00	Baseline (Teacher)
01	Quantization Static (INT8)
02	Pruning Structured(15%)
03	Self-Distillation
04	Distillation + Quantization
05	Baseline Student
06	Two Stage Distillation (Student)
07	Distillation + Prune
08	Distillation + Prune
09	Distillation + Prune + Quantization

Table 1 presents all the methods used in this study:

Methods 00 and 05: Used as benchmarks to measure the architectural gap. The Teacher (Method 00) represents the upper bound of accuracy and is the source of "dark knowledge". The Student (05) is a baseline lightweight architecture, trained in a traditional way (using cross-entropy), which illustrates the problem of insufficient capacity when training "from scratch".

Methods 01 and 02: Methods demonstrating classical approaches to model reduction without using distillation. Method 02 physically removes 15% of convolutional filters based on their L1-norm, followed by standard fine-tuning. Method 01 converts weights into an 8-bit integer format using calibration on a validation dataset. Both methods usually lead to the degradation of the feature space.

Methods 03 and 04: Implementation of the classical Hinton distillation protocol [12], where a randomly initialized student immediately begins training by minimizing the KL-divergence. These scenarios serve as indicators of the "gradient shock" problem, where the lack of preliminary classifier stabilization leads to the destruction of gradients in deep layers.

Method 06: The core implementation of the Two-Stage Distillation algorithm. It includes isolated classifier stabilization using label smoothing and subsequent temperature distillation of the unfrozen model ($T = 4.0$).

Method 07: A specialized pipeline for high-performance computing devices with tensor core support. Since INT8 dequantization on a GPU causes memory latency, this method converts the distillation-trained model (Method 06) into a half-precision floating-point format (FP16), which guarantees maximum acceleration without losses.

Method 08: Demonstrates the "distillation healing" effect. Instead of standard fine-tuning after filter removal, the pruned architecture is trained using the teacher's "soft targets". This allows the optimizer to find efficient signal routing paths in a limited parameter space.

Method 09: The most aggressive compression chain, aimed at deployment in memory-constrained environments. It combines the advantages of Two-Stage distillation (for high accuracy), structural pruning (for size reduction), and INT8 quantization (to ensure high FPS using the processor's vector instructions).

3. Experimental evaluation: Overcoming gradient shock

The first phase of the experimental study aimed to prove the hypothesis about the destructive impact of random classifier initialization during distillation and to justify the critical need for the developed stabilization stage. To ensure the purity of the experiment, the algorithms were tested on two fundamentally different topologies: a linear deep network (VGG) and a network with residual connections (ResNet).

Analysis of the obtained accuracy metrics allows drawing the following conclusions:

The impact of gradient shock on linear topologies (VGG family):

The impact of gradient shock on linear topologies (VGG family):

The reference teacher model VGG16 (Method 00) achieves an accuracy of 0.684. The attempt to apply the standard Self-Distillation procedure (Method 03) without the preliminary stabilization stage led to a catastrophic degradation of the feature space: the accuracy of the heavy model dropped to *0.621 (-0.063)*. This is a direct empirical proof of "gradient shock": massive errors from the unprepared classifier spread through the linear

structure of VGG (which has no bypass paths for gradients), destroying the already trained convolutional filters.

At the same time, the baseline student model VGG11 (Method 05) achieved an accuracy of *0.616* when trained "from scratch". The application of the developed Two-Stage Distillation algorithm (Method 06) allowed for algorithmically isolating the classifier at the initialization stage. As a result, the distilled student not only avoided gradient destruction but also increased its accuracy to *0.642* (an increase of *+0.026* relative to the baseline student).

Buffering the shock with residual connections (ResNet family):

The study of the ResNet family revealed a different dynamic due to the presence of Skip Connections, which act as highways for the safe passage of gradients. The teacher model ResNet50 (Method 00) demonstrated an accuracy of *0.849*. Due to structural stability, applying standard self-distillation (Method 03) to it did not cause degradation, but instead provided a slight regularization effect, increasing the accuracy to *0.856*.

Despite this architectural advantage of ResNet, the developed two-stage approach remains highly effective for transferring knowledge to smaller models. The baseline student ResNet18 (Method 05) showed an accuracy of *0.770*. Training this student using the Two-Stage Distillation algorithm (Method 06) ensured the safe transfer of "dark knowledge" from the teacher and increased the final accuracy to *0.785* (an increase of *+0.015*).

Experiments indisputably prove that the absence of a stabilization phase makes classical distillation methods extremely unstable and architecture-dependent (in particular, detrimental to VGG). The proposed Two-Stage Distillation algorithm (Method 06) completely eliminates the problem of gradient shock, providing a stable and predictable accuracy increase for both robust (ResNet) and vulnerable (VGG) neural network topologies.

4. Investigation of synergy and the "Distillation Healing" phenomenon.

A key objective of the second phase of the experiments was to investigate the ability of the developed Two-Stage Distillation algorithm to maintain accuracy during the application of destructive compression methods (pruning and quantization) and to prove the existence of the "distillation healing" phenomenon.

Restoration of representativeness after structural pruning.

Structural pruning physically removes convolutional filters, which radically changes the geometry of the parameter space. The isolated use of this method with standard fine-tuning usually does not allow the model to restore the lost connections.

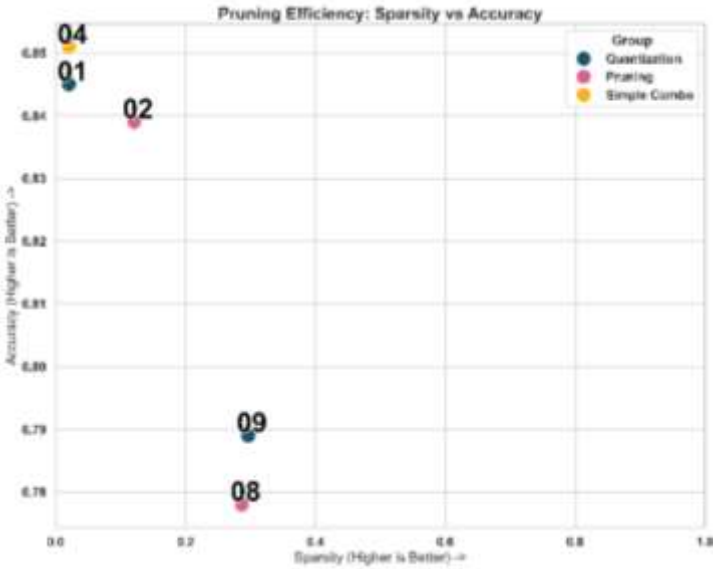

Fig. 1. Impact of distillation on robustness to structural pruning (ResNet)

The data in Figure 1 confirm that the application of standard pruning (Method 02) reduces the network accuracy to *0.839*. However, the "Distillation Healing" phenomenon is clearly manifested in Method 08. After removing almost 28% of the weights in the ResNet18 student, the use of "soft targets" using the Two-Stage Distillation algorithm allows the pruned model to achieve an accuracy of *0.778*. A counterintuitive paradox arises: the pruned distilled model works more accurately than the fully-fledged baseline student model (Method 05, accuracy *0.770*). This indicates that distillation acts as an ideal spatial navigator for the optimizer in a narrowed parameter space.

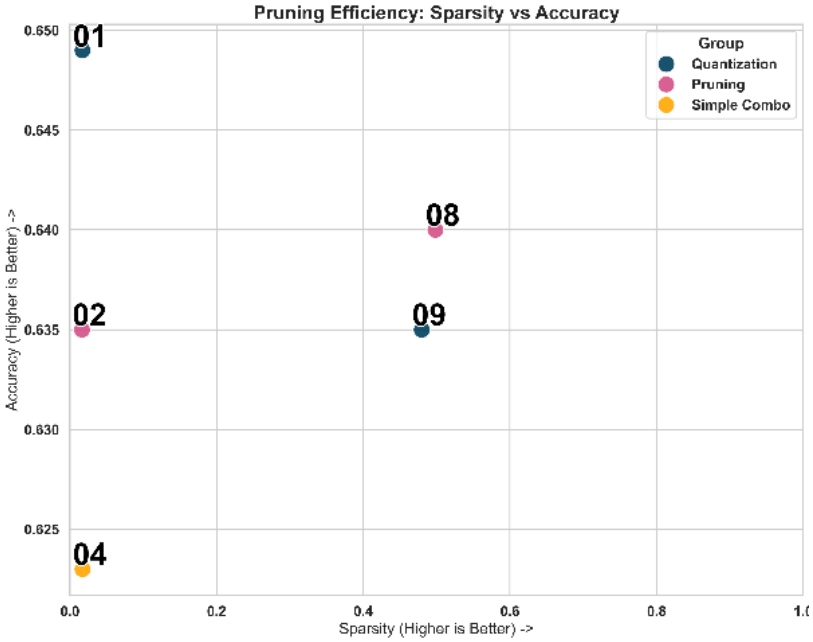

Fig. 2. Impact of distillation on robustness to structural pruning (VGG)

Figure 2 shows a similar "healing" effect for linear topologies. Standard pruning (Method 02) drops the accuracy to *0.635*. But the application of the hybrid Method 08 allows for partial restoration of the feature space, reaching an accuracy of *0.640* and outperforming the baseline VGG11 student (Method 05, accuracy *0.616*).

Deep compression and memory footprint reduction.

For the deployment of models on edge devices, the physical size of the model in memory is a critical metric.

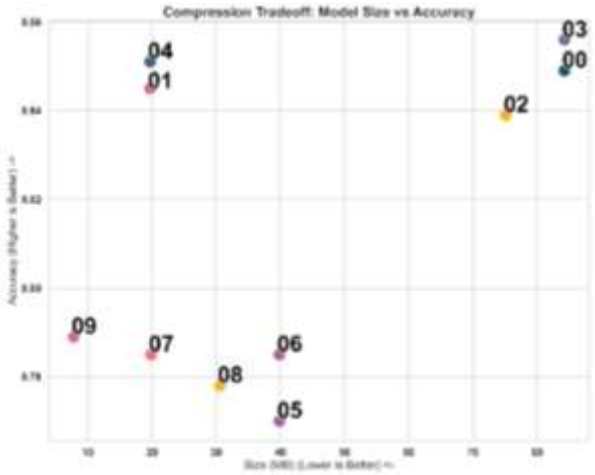

Fig. 3. Trade-off between model size and classification accuracy (ResNet)

As can be seen from Figure 3, the most impressive result of extreme compression is demonstrated by the hybrid Method 09. The sequential application of Two-Stage distillation, pruning, and INT8 quantization reduces the memory footprint for ResNet from *84.2 MB* (teacher, Method 00) to an unprecedented *7.7 MB* (compression ratio *10.9x*). At the same time, the preserved accuracy is *0.789*, which outperforms all other configurations of compact models.

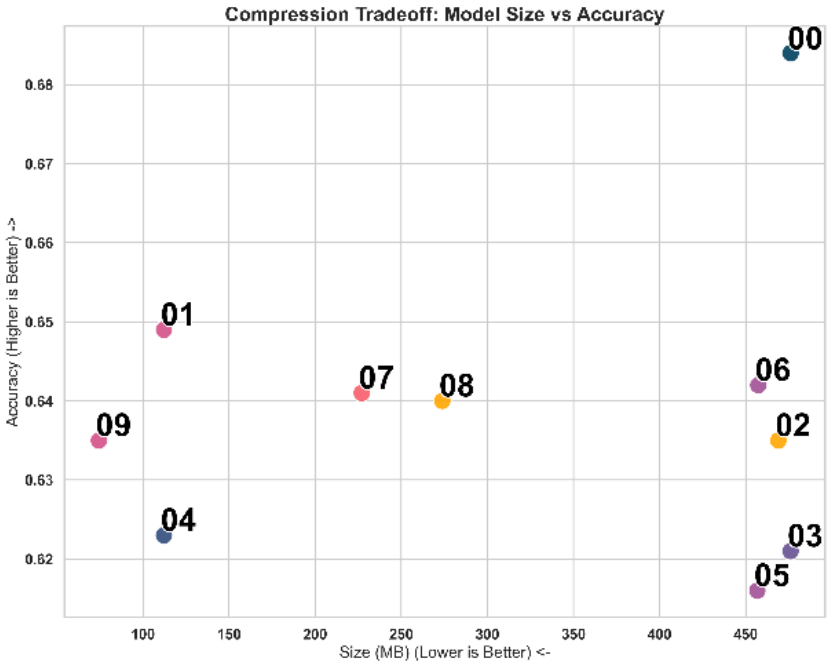

Fig. 4. Trade-off between model size and classification accuracy (VGG)

In Figure 4, it can be seen that the massive VGG architecture compresses even more effectively. Method 09 allows reducing the VGG size from *513.7 MB* (Method 00) to *123.4 MB* (compression ratio *4.16x*), maintaining accuracy at the *0.635* level. Attempts to achieve such compression without our distillation algorithm (for example, Method 04 with standard distillation) yield a significantly worse accuracy result (*0.623*).

Pareto efficiency and cross-platform analysis

The hardware architecture of the computing device dictates the choice of the optimal compression method. For a comprehensive evaluation of the methods, Pareto fronts were constructed, illustrating the trade-off between inference speed and accuracy on different hardware platforms.

The massive VGG architecture traditionally suffers from low speed due to a large number of parameters in the fully connected layers.

From Figure 5, it can be seen that the absolute leader on the Pareto front for systems with limited power consumption is the hybrid Method 09. Thanks to the synergy of distillation and converting weights into an integer format, the processor can effectively use vector instructions. The inference speed increases from the baseline 9.09 FPS (teacher, Method 00) to 35.07 FPS, providing a 3.85x acceleration at an accuracy of 0.635. Isolated static quantization (Method 01) provides only 18.9 FPS, which proves the need for a comprehensive approach with pruning.

In Figure 6, it can be seen that methods with INT8 quantization (01, 04, 09) are absent, as they demonstrate a critical performance drop (0.0 FPS). This is explained by the fact that modern GPUs are architecturally optimized for floating-point operations, and the software dequantization of integer tensors creates a memory "bottleneck". Therefore, the optimal point on the GPU front is Method 07. Converting the distilled student to a half-precision format allows the tensor cores to

process the network at a speed of 601.78 FPS, which is 2.37x times higher than the metric of the original teacher (253.6 FPS), while maintaining high accuracy (0.641).

ResNet family networks, due to residual connections and the absence of massive FC layers, are initially more optimized, however, the Two-Stage Distillation algorithm allows to further maximize their hardware efficiency.

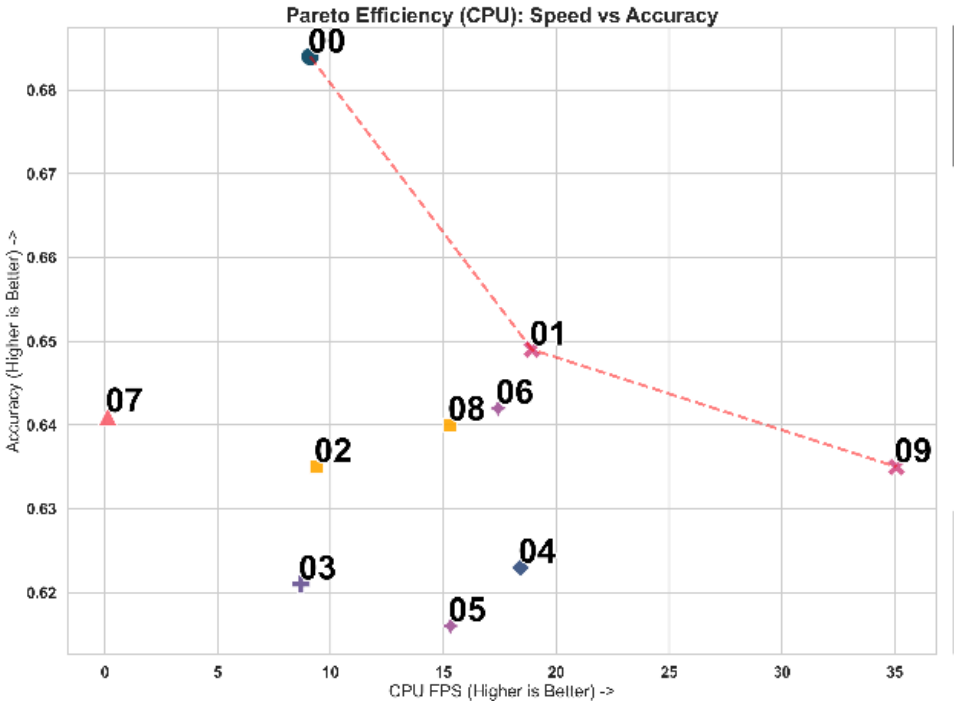

Fig. 5. Pareto front of compression methods efficiency for VGG on CPU

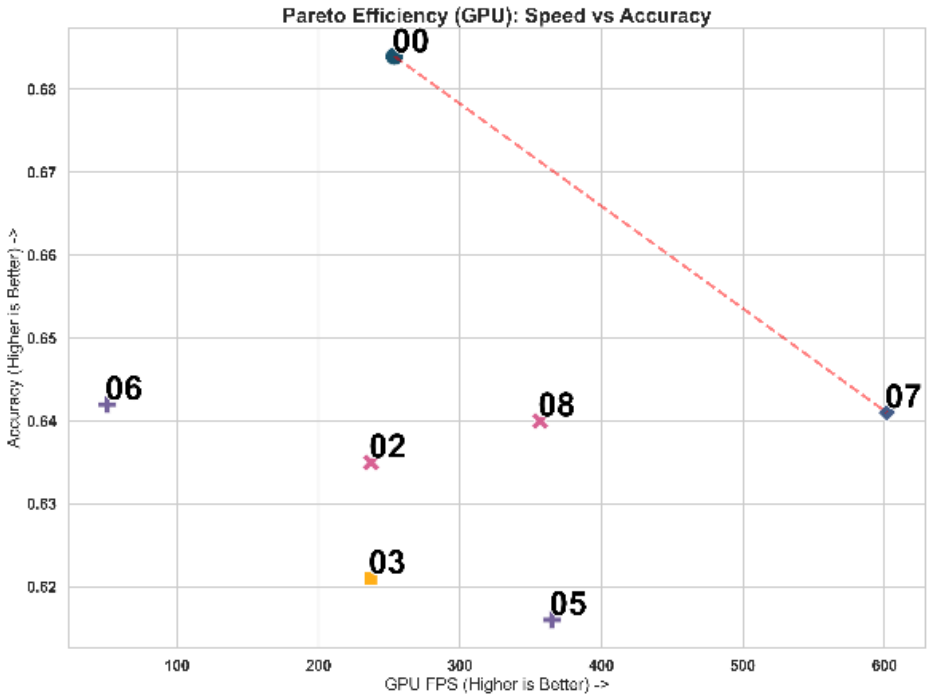

Fig. 6. Pareto front of compression methods efficiency for VGG on GPU

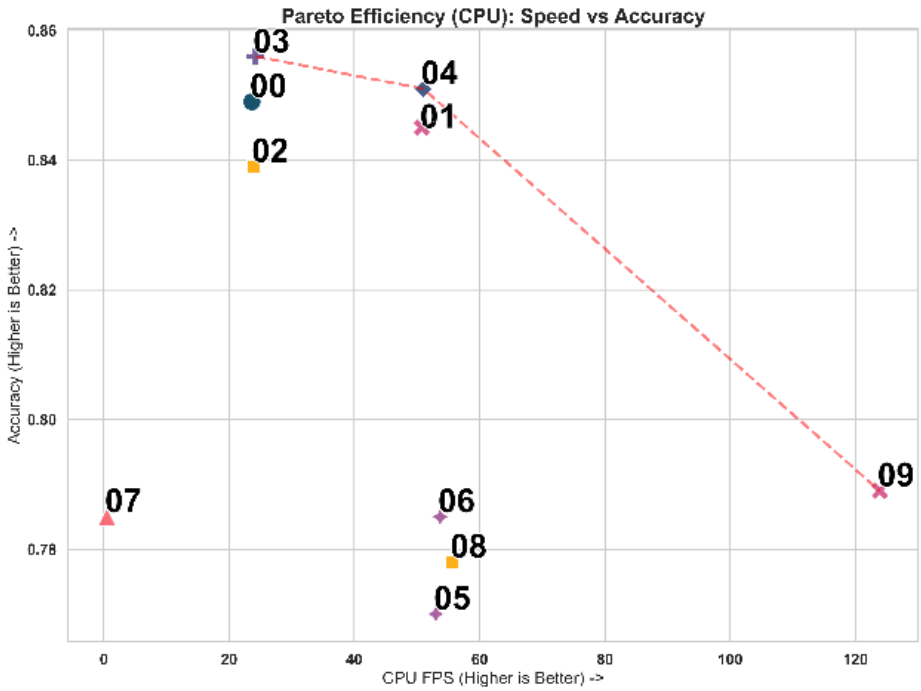

Fig. 7. Pareto front of compression methods efficiency for ResNet on CPU

From Figure 7, it is evident that, as in the case of VGG, Method 09 forms the optimal Pareto front. Processing speed on the central processing unit increases from 23.6 FPS (teacher, Method 00) to a phenomenal 123.8 FPS (a 5.24x acceleration). At the same time, the preserved accuracy is 0.789, making this pipeline an ideal candidate for the deployment of real-time computer vision systems on embedded IoT devices.

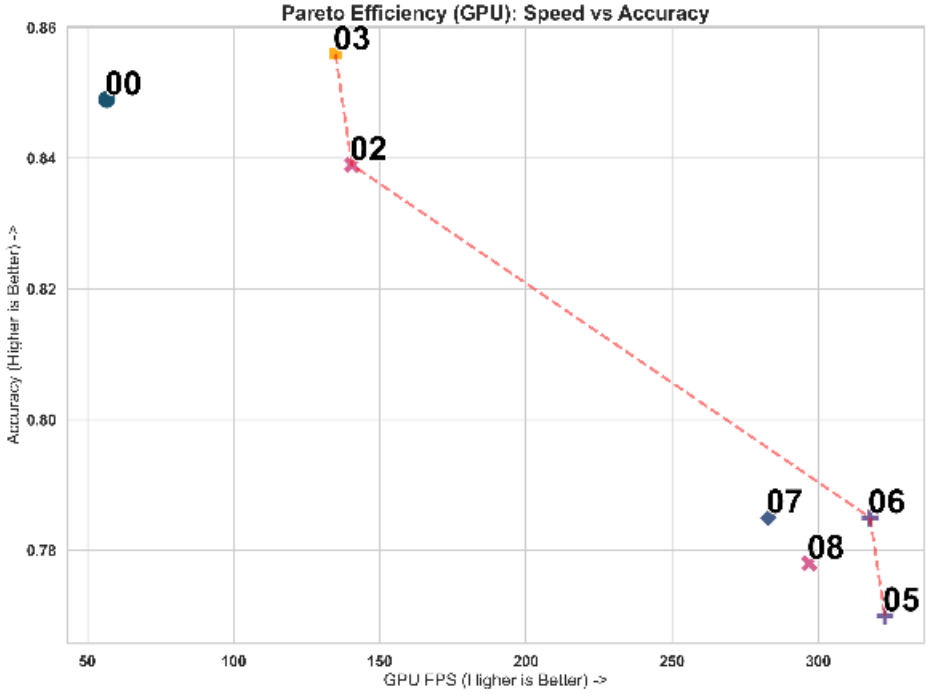

Fig. 8. Pareto front of compression methods efficiency for ResNet on GPU

From Figure 8, it can be observed that on a CUDA-supported platform, Method 07 indisputably takes the lead. While the baseline ResNet50 model (Method 00) provides 56.6 FPS, the distilled student in FP16 format accelerates inference to 282.8 FPS (an exact 5.0x acceleration). The accuracy remains at 0.785, which is a statistically significant metric for the compact ResNet18 model. Similarly to the VGG family, the application of INT8 methods on a GPU for ResNet leads to zero efficiency.

Discussion of the results

1. Elimination of gradient shock and architectural dependence.

The conducted comparison of standard distillation (Method 03) and the proposed two-stage algorithm (Method 06) revealed a critical vulnerability of linear topologies (VGG) to initial initialization. Since linear networks lack bypass paths (skip connections), chaotic gradients from an unprepared classifier irreversibly destroy spatial filters. The introduction of a stabilization stage using label smoothing algorithmically isolates this destructive impact. For networks with residual connections (ResNet), this problem is less pronounced due to their structural stability; however, Two-Stage Distillation still provides a more stable feature transfer and higher final accuracy.

2. The "Distillation Healing" mechanism as a parameter space navigator.

The removal of the network's structural elements (pruning, Method 02) sharply narrows the model's capacity, making the standard cross-entropy function ineffective for fine-tuning. It is proven that the use of the teacher's "soft targets" (Method 08) acts as a powerful spatial navigator. Distillation transforms the destructive pruning process into an effective method for finding the optimal sub-architecture (the "lottery ticket"), allowing the pruned network to outperform its baseline full-size counterparts in accuracy.

3. Cross-platform adaptability of pipelines.

The isolated application of compression methods (for example, solely quantization, Method 01) is unable to provide a Pareto-efficient trade-off, as it leads either to a significant loss of accuracy or to hardware incompatibility.

4. Summary.

It is proven that the proposed comprehensive approach based on Two-Stage Distillation is capable of generating highly efficient hardware-specific solutions. For systems with limited power consumption on a CPU, the path of deep compression is without alternative: distillation followed by structural pruning and static INT8 quantization (Method 09). This ensures maximum acceleration due to vector instructions and minimal memory footprint. Whereas for cloud inference servers (high-performance GPUs), the use of integer arithmetic creates memory latency, therefore, the maximum Pareto efficiency here is provided by combining smart distillation and the FP16 format (Method 07).

Conclusions

As a result of the conducted research, the scientific and practical task of increasing the efficiency of knowledge transfer between convolutional neural networks for their deployment on resource-constrained edge devices has been solved. Based on theoretical analysis and experimental modeling, the following conclusions were drawn:

1. Overcoming the phenomenon of gradient shock.

It has been proven that standard knowledge distillation protocols (without a stabilization phase) cause degradation of the feature space upon random initialization of the classifier, which is particularly critical for linear topologies (VGG accuracy drops by *6.3%*). The developed and implemented Two-Stage Distillation algorithm with a Warmup stage (using label smoothing) algorithmically isolates destructive gradients. This allowed for a stable increase in the accuracy of compact student models: for VGG11 by *+2.6%* (to *0.642*), and for ResNet18 by *+1.5%* (to *0.785*) compared to standard training.

2. Confirmation of the "Distillation Healing" mechanism.

The ability of the developed distillation algorithm to act as a healing mechanism after destructive structural pruning has been experimentally proven. The use of temperature-smoothed teacher's "soft targets"

as a spatial navigator allows pruned architectures (Method 08) to outperform their full-size counterparts in accuracy, negating the losses from removing up to *28%* of the weight coefficients.

3. Selection of the optimal method for computing with limited resources.

It was established that for Internet of Things systems, the comprehensive approach combining Two-Stage distillation, structural pruning, and static INT8 quantization (Method 09) is without alternative. The developed chain provided a compression ratio of the physical memory footprint of *10.9x* (from *84.2 MB* to *7.7 MB* for ResNet) and an inference acceleration of *5.24x* on the central processing unit (up to *123.8 FPS*) while maintaining a high generalization capacity (accuracy *0.789*).

4. Hardware-dependent optimization for high-performance computing.

It has been proven that the application of integer quantization is inefficient on modern graphics accelerators due to memory dequantization latency. For cloud architectures, the optimal solution is a combination of Two-Stage distillation and converting weights into a half-precision format (Method 07). This approach allows maximizing the utilization of tensor cores, achieving an inference speed of *282.8 FPS* for the ResNet family and *601.7 FPS* for the VGG family without any loss in classification accuracy.

References

1. Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations (ICLR)*. [Online]. Available: http://arxiv.org/abs/1409.1556
2. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. doi: 10.1109/CVPR.2016.90. [Online]. Available: https://doi.org/10.1109/CVPR.2016.90
3. Deng, L., Li, G., Han, S., Shi, L., & Xie, Y. (2020). Model Compression and Hardware Acceleration for Neural Networks: A Comprehensive Survey. *Proceedings of the IEEE*, 108(4), 485–532. doi: 10.1109/JPROC.2020.2976475. [Online]. Available: https://doi.org/10.1109/JPROC.2020.2976475
4. Park, W., Kim, D., Lu, Y., & Cho, M. (2019). Relational Knowledge Distillation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3967–3976. doi: 10.1109/CVPR.2019.00409. [Online]. Available: https://doi.org/10.1109/CVPR.2019.00409
5. Mirzadeh, S. I., Farajtabar, M., Li, A., & Ghasemzadeh, H. (2020). Improved Knowledge Distillation via Teacher Assistant. *AAAI Conference on Artificial Intelligence*, 34(04), 5191–5198. doi: 10.1609/aaai.v34i04.5963. [Online]. Available: https://doi.org/10.1609/aaai.v34i04.5963
6. Blalock, D., Gonzalez Ortiz, J. J., Frankle, J., & Guttag, J. (2020). What is the State of Neural Network Pruning? *Proceedings of Machine Learning and Systems (MLSys)*, 2, 129–146. [Online]. Available: https://proceedings.mlsys.org/paper/2020/file/d2ddea18f00665ce8623e36bd4e3c7c5-Paper.pdf
7. Liu, Z., Sun, M., Zhou, T., Huang, G., & Darrell, T. (2021). Rethinking the Value of Network Pruning. *International Conference on Learning Representations (ICLR)*. [Online]. Available: https://openreview.net/forum?id=rJlnB3C5Ym
8. Menon, A. K., Rawat, A. S., Reddi, S. J., & Kumar, S. (2021). Why Distillation Helps: A Statistical Perspective. *International Conference on Machine Learning (ICML)*, 139, 7651–7662. [Online]. Available: http://proceedings.mlr.press/v139/menon21a.html
9. Stanton, S., Izmailov, P., Kirichenko, P., Alemi, A. A., & Wilson, A. G. (2021). Does Knowledge Distillation Really Work? *Advances in Neural Information Processing Systems (NeurIPS)*, 34, 6906–6919. [Online]. Available: https://doi.org/10.48550/arXiv.2106.05945
10. Frankle, J., & Carbin, M. (2019). The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. *International Conference on Learning Representations (ICLR)*. [Online]. Available: https://doi.org/10.48550/arXiv.1803.03635
11. Zagoruyko, S., & Komodakis, N. (2017). Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer. *International Conference on Learning Representations (ICLR)*. [Online]. Available: https://arxiv.org/abs/1612.03928
12. Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the Knowledge in a Neural Network. *NIPS Deep Learning Workshop*. [Online]. Available: http://arxiv.org/abs/1503.02531

The article has been sent to the editors 26.01.26.
After processing 14.02.26.
Submitted for printing 30.03.26.