Minimally Invasive Surgery for Sparse Neural Networks in Contrastive Manner

Chong Yu1,2
1NVIDIA 2Fudan University
chongy@nvidia.com

Abstract

With the development of deep learning, neural networks tend to be deeper and larger to achieve good performance. Trained models are more compute-intensive and memory-intensive, which lead to the big challenges on memory bandwidth, storage, latency, and throughput. In this paper, we propose the neural network compression method named minimally invasive surgery. Different from traditional model compression and knowledge distillation methods, the proposed method refers to the minimally invasive surgery principle. It learns the principal features from a pair of dense and compressed models in a contrastive manner. It also optimizes the neural networks to meet the specific hardware acceleration requirements. Through qualitative, quantitative, and ablation experiments, the proposed method shows a compelling performance, acceleration, and generalization in various tasks.

1. Introduction

Deep learning technologies promote performance in various applications like computer vision, natural language processing, autonomous driving, recommendation system, etc. The promising performance is achieved by deeper and larger neural networks. For example, the classical architectures in convolutional neural networks like VGG-19 [41], ResNeXt-101 [49], SENet-154 [17] has 143.67, 83.46, and 115.09 million parameters, respectively. Google’s neural machine translation model [48] has about 210 million parameters. The popular language understanding model BERT [8] has about 340 million parameters. The deep learning recommendation model (DLRM) [29] has about 540 million parameters.

Neural networks with a huge amount of parameters have some shortcomings [56]. First of all, the large neural network is very compute-intensive. In the network evaluation process, inference costs a lot of time even the network is running on dedicated acceleration hardware like GPU [31] [32] or TPU [40]. We can enlarge the batch size to help improve the throughput of large neural networks. But the latency is still a problem. In fact, whenever we interact with phones or computers, we are very sensitive to the latency of the interaction. We don’t like to wait for an application to launch or for the web-page to load search results. Moreover, we are especially sensitive in real-time interactions such as speech recognition and autonomous driving systems. Secondly, the large neural network is memory-intensive on mobile devices as well as in the server environment. Storage and loading the large neural network to compute inference results consume a large amount of energy. Due to the limitations on application sizes, download time and launch speed, transfer and storage of large models is especially a challenge in the mobile environment.

Compressing the large neural network to a smaller version can bring benefits to more efficient computation, memory, and energy consumptions. But at the meanwhile, how to keep the accuracy of the original neural network during compression needs to be investigated. A common method of neural model compression is network pruning [12]: setting the weights with small magnitude values of a pre-trained network to zero and fine-tuning the remaining weights to try to recover accuracy. For the aggressive network pruning tasks, knowledge distillation [15] is often used as the auxiliary method to improve the accuracy of the pruned network. A complementary method of neural model compression is quantization. Changing fundamental data types adds the ability to accelerate the arithmetic operations, both in training [28] and inference processes [20].

In this work, we explore a neural network compression method based on the knowledge extracted from a pair of dense and compressed models. We named this method as Minimally Invasive Surgery (MIS) because it is inspired by the principle and process of the real minimally invasive surgery. We apply the MIS technique to several networks and tasks to show generality in supervised and unsupervised learning. Our main contributions include:

- We prove that MIS has better performance than knowledge distillation and network compression methods.
- We provide the theoretical demonstration of MIS from information entropy and Bayes perspectives.
- We show that MIS technique can apply to various net-
works and tasks. It can work even without ground truth label info in an unsupervised learning style.

- MIS provides end-to-end compression for neural networks to meet the hardware acceleration requirements.

2. Related work

2.1. Knowledge distillation

Knowledge Distillation (KD) was first proposed by Bucilu et al. [5] and generalized by Hinton et al. [15]. It has become one of the most effective and standard techniques in model compression. KD starts from a large model, named teacher (T), with appealing performance, and then employs a lower-capacity one, named student (S), to learn knowledge from T. In this way, S is supposed to mimic and produce similar results as T but with faster speed and less memory consumption. Take the classification task as an example, instead of just learning from the one-hot representation of the ground-truth label where only the target class is considered in cross-entropy, the student model also learns from the soft labels to represent all probabilities over the whole classes from the teacher model. Hinton et al. [15] proved that the knowledge embedded in soft labels is essential to teach the student more efficiently.

To improve the effectiveness of KD, many methods focused on designing different types of knowledge for the student model. Romero et al. [38] introduced intermediate-level hints from the teacher hidden layers to guide the student. Zagoruyko et al. [51] introduced the attention mechanism in KD. They proved the attention-based feature map has better performance in transferring knowledge to the student than the logits. Ahn et al. [2] improved KD by maximizing the mutual information between the teacher and the student models. In [21], the student learned from several intermediate representative layers in the teacher. They used the teacher’s intermediate representations as input to the student model during training to overcome the lack of useful intermediate representations at the beginning of training.

Despite the various progress on KD, this method is still far from perfect. There are two common troubles when applying KD. First, when the capacity difference between the teacher and the student is very large, the effectiveness of KD will decrease. Especially when the student model is compressed with a very high sparse ratio or to a very shallow structure. This is because the inherent discrepancy between the model capacities of the student and the teacher will lead to a much weaker representation ability for the student [9]. Second, it is hard to find a general learning strategy and hyper-parameters in KD [50]. This is because the student has an inherent slower learning speed than the teacher. So this discrepancy between the models prevents the student from fully acquiring knowledge as the teacher in the same training schedule.

2.2. Contrastive learning

In contrast to learning high-level representations from labeled data, Contrastive Learning (CL) means to learn less specialized representations in latent space [30]. By introducing latent classes and hypothesizing that semantically similar points are sampled from the same latent class, CL can leverage unlabeled as well as labeled data. Oord et al. [34] introduced a probabilistic contrastive loss to capture information that is maximally useful to predict future samples in latent space. They proved CL is especially useful to the unsupervised tasks in a wide variety of domains: audio, images, natural language, etc. Arora et al. [3] provided a theoretical analysis of CL which can make provable guarantees on the learning performance.

To solve the aforementioned problems in KD, some works began to borrow the idea from CL for further improvement. Tian et al. [43] changed the typical objective that minimizes the divergence between the probabilistic outputs of the teacher and student networks into a contrastive-based objective. The new objective maximized a lower-bound to the mutual information between the teacher and student representations and provided a better performance on several model compression and knowledge transfer tasks. Gao et al. [9] introduced an assistant in the traditional teacher-student framework in KD to learn the residual error between the teacher and student representations in latent space. They used the lightweight structure for the assistant to ensure the total computational cost has no obvious increase.

2.3. Acceleration of compressed model

The ultimate goal of model compression is to generate the model pattern to save storage, computation, and energy cost. Sparsity has been proven as an effective approach to saving parameters as well as preserving the accuracy of neural models. Han et al. [12] proposed to conduct pruning and retraining alternately, and finally compress a dense model to its sparse form. Guo et al. [10] incorporated network connection splicing into the surgery and dynamically implemented the whole compression process. Zhu et al. [54] proposed a gradual pruning method technique that trained neural models from scratch and gradually pruned the redundant parameters in this process. Lee et al. [23] introduced a saliency criterion that identified connections in the network that were important to the given task in a data-dependent way before training. Given the desired sparsity level, redundant connections were pruned once, and then the sparse pruned network was trained in the standard way.

The sparsity caused by network compression typically resulted in an irregular workload, which was difficult for hardware acceleration. Mao et al. [27] discussed the trade-off among sparse regularity, network accuracy, and acceleration. For the coarse-grained sparsity like filter-sparsity and channel-sparsity, the regular pattern was simple to achieve.
acceleration on general-purpose processors because it was equivalent to obtaining a smaller dense model [47]. For fine-grained sparsity, the acceleration on general-purpose hardware like GPU [31] was very limited. Several custom accelerators [11] [35] have been used to exploit the irregular sparse pattern. With the new generation GPU [32], sparse Tensor Cores can exploit fine-grained structured sparsity to double the compute throughput for neural networks.

3. Minimally invasive surgery

As aforementioned, many works have found the sweet spot between model compression and accuracy retrieval. However, the acceleration of the compressed model is far from being solved. The focus of this research is simultaneously obtaining the high compression ratio, the accuracy retrieval performance, and the acceleration on general-purpose hardware. Our intuition is simple. As we already have various methods to compress a dense model into the irregular sparse pattern without obvious accuracy damage. If we can make tiny changes on the irregular sparse pattern like minimally invasive surgery, and match the tensor acceleration requirements on hardware [32]. Then we can improve the deployment efficiency of the irregularly-compressed models.

The proposed model compression method is named as Minimally Invasive Surgery (MIS) for two reasons. Firstly, as the principle of minimally invasive surgery [45], it encompasses surgical techniques that limit the size of incisions needed and so lessens wound healing time, associated pain and risk of infection. Similarly, when applying MIS to a compressed model, we only make the limited adjustment to ease the influence on accuracy and memory cost. Secondly, the goal of MIS is to heal the injured part to be functional-same as the healthy part. Take the Achilles tendon rupture as an example. After the surgery, we could expect a recovered patient to walk, run, and jump like a normal person. However, for the recovered basketball athlete, it is very hard for him to come back to his peak. Similarly, due to the inherent discrepancy between the dense and compressed models, we could expect they are functional-same, like have similar classification accuracy. It is hard for the heavily compressed model to have exactly the same representation as the dense one. This is also the block in KD.

In the MIS, we refer to the dense baseline model as the healthy model $M_H$, the sparse model as the recovered model $M_R$. $M_R$ is obtained by any model compression method, and often cannot be easily accelerated by the general-purpose hardware due to irregularity. We refer to the target compressed model which satisfies the hardware acceleration restrictions as the surgical model $M_S$. Firstly, after applying MIS, $M_S$ should have the similar accuracy as $M_H$ and $M_R$. Secondly, $M_S$ and $M_R$ have same compression ratio, which means they have similar memory costs. Last but not least, due to the inherent different representation capabilities, the introduction of $M_S$ provides the upper-bound of what we can expect $M_S$ to learn from $M_H$.

Take the image classification task as an example. We use an arbitrary image as input. $M_S$ is initialized by $M_R$ with one-shot magnitude-based pruning to meet the hardware acceleration requirement. Because sometimes we have no access to the original training dataset, we cannot always use the supervised finetuning method to recover the accuracy. Instead of using the ground truth labels in the traditional supervised finetuning method, we use $M_H$ predicted classes as the “fake” labels. The prediction loss is calculated between the “fake” label and the predicted class from $M_S$. Similar to vanilla KD, we use the temperature parameter to control the probability distribution generated by the softmax function. We first calculate the distillation loss between the probability distributions from $M_H$ and $M_S$. Then we calculate the distillation loss between $M_R$ and $M_S$. We emphasize the second distillation loss to mimic the inherent gap for dense and sparse models. The overall loss function is the weighted combination of the prediction loss and the two parts of the distillation loss. We finetune to reduce the overall loss and finally get the desired $M_S$. We illustrate the workflow of MIS in Figure 1. We define the Hardware Acceleration Requirements as an integrated function $HAR(\cdot)$. For example, A100 GPU [32] requires two non-zero values in every four-entry vector to double the math throughput. Then MIS for classification task is summarized in Algorithm 1.

Algorithm 1 Minimally Invasive Surgery (Classification)

| Input: Healthy model $M_H$, Recovered model $M_R$, Training images $x$ |
| Parameter: Distillation temperature $\tau$, Loss adjustment factors $\alpha, \beta, \gamma$, Overall loss threshold $\delta$ |
| Output: Surgical model $M_S$ |

1. Init surgical model $M_S$ by recovered model $M_R$.
2. while $L_{\text{Overall}} > \delta$ do
3. Pruning $M_R$ to meet the hardware acceleration requirement: $HAR(M_S)$.
4. if Ground truth label ($l_G$) exists then
5. Surgical prediction loss: $L_P = L(M_H(x; T = 1), M_S(x; T = 1))$
6. else
7. Surgical prediction loss: $L_P = L(M_H(x; T = 1), M_S(x; T = 1))$
8. end if
9. Healthy-surgical distillation loss: $L_{\text{DSH}} = L(M_H(x; T = \tau), M_S(x; T = \tau))$
10. Recovered surgical distillation loss: $L_{\text{DSR}} = L(M_R(x; T = \tau), M_S(x; T = \tau))$
11. Calculate the overall loss: $L_{\text{Overall}} = \alpha L_P + \beta L_{\text{DSH}} + \gamma L_{\text{DSR}}$
12. Minimize the overall loss: $\min[L_{\text{Overall}}]$
13. end while
14. return Surgical model $M_S$

4. Theoretical demonstration for MIS

The three deep neural networks in the MIS are the healthy model $M_H$, the recovered model $M_R$, and the surgical model $M_S$. Given $x$ as the input of networks, we can denote representations at the penultimate layer before logits as $M_H(x)$, $M_R(x)$ and $M_S(x)$. We use $x_1$ and $x_2$ to represent two training samples from different categories. Our target is to push closer the representations of the healthy model and surgical model with the training samples from the same categories, while to push apart the representations of
models with the training samples from different categories. Kullback-Leibler (KL) divergence is applied to measure the difference between the two representations. Ideally, the target can be denoted with the following optimization problem.

\[
\min \sum_{k=1}^{N} \sum_{x_i \in D(C_k)} |KL(M_H(x_i), M_S(x_i))|^2
\]

If the whole dataset is defined as \(D\), with \(N\) categories, and each category is denoted as \(D(C_k)\). Then the target can be denoted with the following optimization problem.

\[
\max \sum_{k=1}^{N} \sum_{x_i \in D(C_k)} \sum_{x_j \not\in D(C_k)} |KL(M_H(x_i), M_S(x_j))|^2
\]

### 4.1. Information theory perspective

From information theory, there will be an information entropy threshold to measure whether the network can keep the same functionality after compression. Take the classification task as the example, if the entropy \(H(\cdot)\) of the compressed model is higher than the threshold, the classification accuracy keeps the same with the original model, otherwise, the accuracy will drop. The information entropy threshold of dataset \(D\) is defined as \(T_D\). The essence of the success in the model compression method is information redundancy in model representation. And intuitively speaking, the model before compression has higher information redundancy than the compressed model. The information redundancy is defined as \(\varphi_H\) and \(\varphi_R\) for the healthy and recovered models.

We can also define the distillation learning effective ratio between the two models to represent how difficult to distill useful information from the original model. The large effective ratio means useful information is easy to be distilled, and the distilled model learns more effectively. We can define the distillation learning effective ratio of the surgical model from the healthy model is \(D^{SH}\), and the ratio of the surgical model from the recovered model is \(D^{SR}\), and the ratio of the recovered model from the healthy model is \(D^{RH}\). Intuitively speaking, if the original model has more parameters, or the parameters amount of two models has a more obvious gap, it is harder for the complete distillation and to mimic the behavior of the original model. So \(D^{SH} < D^{RH}\), and \(D^{SR} < D^{RH}\).

For the vanilla knowledge distillation between the healthy and recovered models, we assume the recovered model can recover to the same accuracy level as the healthy model, then:

\[
H(M_R) = H(M_H) D^{RH} = (T_D + \varphi_H) D^{RH} \geq T_D
\]

For the vanilla knowledge distillation between the healthy and surgical models:

\[
H(M_S) = H(M_H) D^{SH} = (T_D + \varphi_R) D^{SH} < (T_D + \varphi_H) D^{RH}
\]

For the vanilla knowledge distillation between the recovered and surgical models:

\[
H(M_S') = H(M_R) D^{SR} = (T_D + \varphi_R) D^{SR} < (T_D + \varphi_H) D^{SR}
\]

So we cannot make sure the distilled surgical models from the previous two situations still have enough information entropy to exceed the threshold \(T_D\).

According to the proposed MIS method, the surgical is distilled information from both of the healthy and surgical models. We assume the mutual information between the healthy and recovered models will be learned once with higher learning effective ratio, then:

\[
H(M_S'') = H(M_H \cap M_R) D^{SR} + H(M_H) D^{SR} = (\varphi_H - \varphi_R) D^{SR} + (T_D + \varphi_R) D^{SR} = (T_D + \varphi_H) D^{SR} + (T_D + \varphi_R) (D^{SR} - D^{SH})
\]

So we can find \(H(M_S'') > H(M_S')\) and \(H(M_S'') > H(M_S)\) at the same time. It proves why MIS has the better chance to distill more information and achieve better accuracy.

### 4.2. Bayes perspective

Now, suppose the classification accuracy is \(Acc_H\) and \(Acc_R\) for the healthy model \(M_H\) and the recovered model \(M_R\), respectively. The surgical model is initialized by the recovered model, so its classification accuracy \(M_S\) is equal to \(M_R\). We define a latent variable \(C\) which represents whether the classification results provided by the neural models are right \((C = 1)\) or wrong \((C = 0)\). Then the prior probability of the healthy, recovered, and surgical models can be denoted as:

\[
P(C_H = 1) = Acc_H, \quad P(C_H = 0) = 1 - Acc_H
\]

\[
P(C_R = 1) = Acc_R, \quad P(C_R = 0) = 1 - Acc_R
\]

\[
P(C_S = 1) = Acc_R, \quad P(C_S = 0) = 1 - Acc_R
\]

For ease of notation, we define the events \(U\) and \(V\) to denote the model representations between the healthy and surgical.
models, the recovered and surgical models are similar, i.e.,

\[ U \Rightarrow M_U(x) = M_S(x), \quad \overline{U} \Rightarrow M_{\overline{U}}(x) = M_S(x) \]

\[ V \Rightarrow M_V(x) = M_S(x), \quad \overline{V} \Rightarrow M_{\overline{V}}(x) = M_S(x) \]  

(8)

According to the total probability formula, for the vanilla knowledge distillation:

\[ P(C_S = 1) = P(C_S = 1 \mid U)P(U) + P(C_S = 1 \mid \overline{U})P(\overline{U}) \]  

(9)

For the proposed MIS method:

\[ P(C_S = 1) = P(C_S = 1 \mid U, V)P(U, V) + P(C_S = 1 \mid \overline{U}, \overline{V})P(\overline{U}, \overline{V}) \]

\[ + P(C_S = 1 \mid U, \overline{V})P(U, \overline{V}) + P(C_S = 1 \mid \overline{U}, V)P(\overline{U}, V) \]

(10)

Because the surgical model is initialized by the recovered model, so prior probability of event V is:

\[ P(V) = 1, \quad P(\overline{V}) = 0 \]

(11)

The prior total probability formula of MIS method will degrade into the vanilla knowledge distillation form.

Because when the model representations between the healthy and surgical models are similar, the probability of \( P(C_S = 1 \mid U) \) will be very close to the prior probability of the healthy model. So it will not be the problem for the vanilla KD and MIS method.

With the definition of the distillation learning effective ratio, then for the vanilla KD method, the probability of whether a similar representation tuple \( (M_H(x), M_S(x)) \) is from the same category \( (C_S = 1) \) or different category \( (C_S = 0) \) is denoted as:

\[ P(U \mid C_S = 1) = D_{SH}^{\beta H}, \quad P(U \mid C_S = 0) = D_{SH}^{\beta H} \]  

(12)

According to the Bayes theorem, the posterior probability for the right classification \( (C_S = 1) \) when the representation-  

s from the healthy model and the surgical model are similar is given by:

\[ P(C_S = 1 \mid U) = \frac{P(U \mid C_S = 1)P(C_S = 1)}{P(U \mid C_S = 1)P(C_S = 1) + P(U \mid C_S = 0)P(C_S = 0)} \]

\[ = \frac{D_{SH}^{\beta H} \cdot Acc_R}{D_{SH}^{\beta H} \cdot Acc_R + D_{SH}^{\beta H} \cdot (1 - Acc_R)} \]  

(13)

Similarly, the posterior probability for the right classification \( (C_S = 1) \) when the representations from the healthy model and the surgical model are different is given by:

\[ P(C_S = 1 \mid \overline{U}) = \frac{P(\overline{U} \mid C_S = 1)P(C_S = 1)}{P(\overline{U} \mid C_S = 1)P(C_S = 1) + P(\overline{U} \mid C_S = 0)P(C_S = 0)} \]

\[ = \frac{D_{SH}^{\beta H} \cdot (1 - Acc_R)}{(1 - D_{SH}^{\beta H}) \cdot Acc_R + (1 - D_{SH}^{\beta H}) \cdot (1 - Acc_R)} \]  

(14)

In the MIS method, with the introduction of the recovered model, the Bayes formulas are as follows:

\[ P(C_S = 1 \mid V) = \frac{P(C_S = 1 \mid V, C_S = 1)P(V \mid C_S = 1)}{P(V \mid C_S = 1)P(C_S = 1)} \]

\[ = \frac{P(C_S = 1 \mid V, C_S = 1)}{P(V \mid C_S = 1)} \]  

(15)

\[ P(C_S = 1 \mid \overline{V}) = \frac{P(C_S = 1 \mid \overline{V}, C_S = 1)P(\overline{V} \mid C_S = 1)}{P(\overline{V} \mid C_S = 1)P(C_S = 1)} \]

\[ = \frac{P(C_S = 1 \mid \overline{V}, C_S = 1)}{P(\overline{V} \mid C_S = 1)} \]  

(16)

In the initialization stage, the values of vanilla KD and MIS method are the same. However, the distillation learning effective ratios of these two methods are different. For vanilla KD, without the help of the recovered model, the learning effective ratio is \( D_{SH}^{\beta H} < D_{SH}^{\beta H} \). What’s worse, this item needs the surgical model to learn when its representation is different from that of the healthy model. Intuitively, the learning effective ratio is even lower as the learning task is more difficult. For the MIS method, the first item in expression (17) is modeling the situation that the representations between the healthy and the surgical models are different, however, the representations between the recovered and the surgical models are similar. This phenomenon often appears because, for the distilled model with a high compression ratio, the expressive capability will reduce. Moreover, learning from the recovered model with similar representation is much easier, leading to a satisfactory learning effective ratio. Although the second item in (17) is difficult to learn, that phenomenon is very rare. We can just ignore it.

In conclusion, the MIS method keeps the same total probability but changes the learning effective ratio and the probability distribution. Because the optimization process cannot guarantee to find the global optimum. So an easier learning target has a higher expectation to achieve during the same learning and optimization process.

5. Experimental results

For the experiments in this section, we choose PyTorch [36] to implement all algorithms. Most of the training and fine-tuning experimental results are obtained with V100 GPU clusters [31]. The acceleration performance results are obtained with A100 GPU clusters [32] to fully utilize its Tensor Core [33] support for fine-grained structured sparsity. Because V100 and A100 GPUs could provide much larger math throughput of FP16 than FP32 data type, we also combine MIS with the mixed-precision training [28] provided by APEX1 to compress the models into a more hardware-efficient format. So all the accuracy results reported by MIS are using FP16 as the default data type. All the reference algorithms use the default data type provided in public repositories. (All use FP32 except where noted.)

And more results with different adjustment parameters \((\alpha, \beta \text{ and } \gamma)\) in sections 5.1 to 5.4 can refer to Appendix.

5.1. Effectiveness experiments for classification task

To evaluate the effectiveness of the MIS on the image classification task, ResNet-50 [14], ResNeXt-101 [49], VGG-19 [41], Inception-V3 [42], DenseNet-161 [18] and MobileNet-V2 [39] from TorchVision2 are chosen as the experiment target models. The original sparse models serve as \( M_R \) are trained with the public Distiller library3 [56].

1 https://github.com/NVIDIA/apex
2 https://github.com/pytorch/vision

3593
The results are shown in Table 1. *FINE* represents the fine-grained sparse model obtained by adopting a gradual pruning technique (AGP) 4 [54], *BLK* represents the block-grained [55] sparse model. *SUR* represents the fine-grained [10] sparse model by applying pruning and splicing in a dynamical manner. *SNIP* represents the single-shot pruned [23] model by analyzing the connection sensitivity. In this experiment, MIS does not use the ground truth label provided by ImageNet [7] dataset. It takes the predicted label from $M_{H}$ to calculate the surgical prediction loss. The loss adjustment parameters among the surgical prediction loss ($\alpha$), the healthy-surgical distillation loss ($\beta$) and the recovered-surgical distillation loss ($\gamma$) apply 1, 10, 50, respectively. (The variance is within $\pm 0.17$ for Top-1, and $\pm 0.15$ for Top-5 accuracy with different random seeds.)

<table>
<thead>
<tr>
<th>Model</th>
<th>Healthy Model Accuracy (%)</th>
<th>Sparsity Ratio</th>
<th>Recovered Model Accuracy (%)</th>
<th>Surgical Model Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Top-1 (%)</td>
<td>Top-5 (%)</td>
<td>Top-1 (%)</td>
<td>Top-5 (%)</td>
</tr>
<tr>
<td>ResNet-50</td>
<td>76.130</td>
<td>92.062</td>
<td>76.460</td>
<td>92.100</td>
</tr>
<tr>
<td></td>
<td>76.150</td>
<td></td>
<td>76.480</td>
<td>92.120</td>
</tr>
<tr>
<td></td>
<td>76.160</td>
<td></td>
<td>76.500</td>
<td>92.140</td>
</tr>
<tr>
<td>ResNet-101</td>
<td>78.188</td>
<td>92.886</td>
<td>78.824</td>
<td>92.984</td>
</tr>
<tr>
<td></td>
<td>78.198</td>
<td></td>
<td>78.844</td>
<td>93.008</td>
</tr>
<tr>
<td></td>
<td>78.208</td>
<td></td>
<td>78.864</td>
<td>93.032</td>
</tr>
<tr>
<td>VGG-19</td>
<td>74.240</td>
<td>90.838</td>
<td>74.600</td>
<td>91.032</td>
</tr>
<tr>
<td></td>
<td>74.250</td>
<td></td>
<td>74.620</td>
<td>91.060</td>
</tr>
<tr>
<td></td>
<td>74.260</td>
<td></td>
<td>74.640</td>
<td>91.080</td>
</tr>
<tr>
<td>Inception-V3</td>
<td>77.368</td>
<td>93.644</td>
<td>77.620</td>
<td>93.960</td>
</tr>
<tr>
<td></td>
<td>77.378</td>
<td></td>
<td>77.640</td>
<td>94.006</td>
</tr>
<tr>
<td></td>
<td>77.388</td>
<td></td>
<td>77.660</td>
<td>94.044</td>
</tr>
<tr>
<td>DenseNet-161</td>
<td>73.114</td>
<td>93.578</td>
<td>73.370</td>
<td>93.872</td>
</tr>
<tr>
<td></td>
<td>73.124</td>
<td></td>
<td>73.390</td>
<td>93.900</td>
</tr>
<tr>
<td></td>
<td>73.134</td>
<td></td>
<td>73.410</td>
<td>93.928</td>
</tr>
<tr>
<td>MobileNet-V2</td>
<td>71.080</td>
<td>80.290</td>
<td>71.320</td>
<td>80.530</td>
</tr>
<tr>
<td></td>
<td>71.090</td>
<td></td>
<td>71.340</td>
<td>80.560</td>
</tr>
<tr>
<td></td>
<td>71.100</td>
<td></td>
<td>71.360</td>
<td>80.590</td>
</tr>
</tbody>
</table>

Table 1. MIS effectiveness on image classification task.

5.2. Effectiveness experiments for detection task

To evaluate the effectiveness of the MIS on the detection task. Faster R-CNN [37], RetinaNet [24], Mask R-CNN [13] from *Detectron* 5, and SSD [26] from *NVIDIA A repository* 6 are chosen as the experiment target models. The original sparse models serve as $M_{P}$ are compressed with AGP method and trained with the Distiller library 7. The results are shown in Table 2. R50, R101 and X101 in the brackets represent the ResNet-50, ResNet-101 and ResNeXt-101 models served as the backbone of the detection networks. $I_{x}$ and $3x$ represent the different learning rate schedulers which are applied when training the backbone models. $AP$ and $AR$ represent the average precision and average recall metrics. In this experiment, MIS use

4Notice some of the sparse ResNet-50 models and all of the sparse ResNeXt-101 models have higher accuracy than the pre-trained dense models provided by TorchVision.

5https://github.com/facebookresearch/detectron2.


<table>
<thead>
<tr>
<th>Model</th>
<th>Healthy Model BLEU Score</th>
<th>Sparsity Ratio</th>
<th>Recovered Model BLEU Score</th>
<th>Surgical Model BLEU Score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Top-1 (%)</td>
<td>Top-5 (%)</td>
<td>Top-1 (%)</td>
<td>Top-5 (%)</td>
</tr>
<tr>
<td>GNMITBMW16 En-Ge</td>
<td>24.37((±0.20)</td>
<td>75%</td>
<td>24.67((±0.14)</td>
<td>75%</td>
</tr>
<tr>
<td>Transformer(En-Ge)</td>
<td>28.65((±0.10)</td>
<td>75%</td>
<td>28.79((±0.09)</td>
<td>75%</td>
</tr>
<tr>
<td>Transformer(En-Ge)</td>
<td>27.93((±0.13)</td>
<td>75%</td>
<td>27.99((±0.13)</td>
<td>75%</td>
</tr>
</tbody>
</table>

Table 3. MIS effectiveness on translation task.

5.3. Effectiveness experiments for translation task

To evaluate the effectiveness of the MIS on the translation task, we take the GNMT [48] from *NVIDIA repository* 7 and Transformer [44] from *Fairseq* 8 as the experiment target models. The original sparse models serve as $M_{R}$ are compressed with the pruning method [6]. The results are shown in Table 3. WMT14 En-Ge and WMT16 En-Ge in the brackets represent the WMT14 and WMT16 English-German dataset 8, respectively. In this experiment, MIS use the ground truth info provided by WMT datasets. The loss adjustment parameters among the surgical prediction loss ($\alpha$), the healthy-surgical distillation loss ($\beta$) and the recovered-surgical distillation loss ($\gamma$) apply 1, 2, 5.

5.4. Effectiveness experiments for super resolution

To evaluate the effectiveness of the MIS on the super resolution task, we take the SRResNet 9 [22] as the experiment target model. The original sparse models serve as $M_{R}$ are compressed with the pruning method [16]. SRResNet is trained on the DIV2K dataset [1]. The DIV2K validation images, as well as Set5 [4] and Set14 [52] datasets are

7https://github.com/pytorch/fairseq.


Table 2. Effectiveness on detection task.
used to report deployment quality. In the super resolution task, image quality is often evaluated by two metrics: Peak Signal-to-Noise Ratio (PSNR) [19] and Structural Similarity (SSIM) [46]. The results are shown in Table 4, and a representative output is shown in Figure 2. The loss adjustment parameters among the surgical prediction loss (α), the healthy-surgical distillation loss (β) and the recovered-surgical distillation loss (γ) apply 1, 1.5, 3, respectively.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Healthy Model</th>
<th>Residual Error</th>
<th>Recovered Model</th>
<th>Surgical Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set5</td>
<td>31.803</td>
<td>0.863</td>
<td>50%</td>
<td>31.234 0.870</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>75%</td>
<td>31.145 0.862</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>90%</td>
<td>30.989 0.854</td>
</tr>
<tr>
<td>Set14</td>
<td>28.643</td>
<td>0.726</td>
<td>50%</td>
<td>28.315 0.755</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>75%</td>
<td>28.275 0.750</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>90%</td>
<td>28.012 0.743</td>
</tr>
<tr>
<td>DIV2K</td>
<td>29.256</td>
<td>0.788</td>
<td>50%</td>
<td>28.926 0.811</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>75%</td>
<td>28.795 0.793</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>90%</td>
<td>28.423 0.735</td>
</tr>
</tbody>
</table>

Table 4. MIS effectiveness on super resolution task.

Figure 2. Representative super resolution results with enlargements of boxed areas (The Recovered Model and Surgical Model are compressed to 50% sparse level).

### 5.5. Ablation experiments and insights

In this experiment, we want to check the contribution of each component in MIS to the final model compression effect. Then we can have a deep insight into why MIS can outperform state-of-the-art methods. Apart from AGP and KD methods we have discussed, we also involve the Residual Knowledge Distillation [9] (RKD) and Contrastive Representation Distillation [43] (CRD) methods in the comparison. The results are shown in Table 5. More results with different sparsity ratio can refer to Appendix. Unsupervised and Supervised in the brackets represent MIS does not use and use the ground truth info provided by ImageNet, respectively.

From the results, we can see the gradual pruning technique (AGP) during finetuning can get a fine-grained sparse model with even higher accuracy than the dense healthy model. However, the compressed model has an irregular sparse pattern. So this model can hardly get acceleration on general-purpose processors. This is the same situation for the models compressed with BLK and KD. When both methods use the ground truth info, the accuracy of the compressed model by KD is obviously lower than applying MIS. It proves the introduction of the recovered model is essential to improving the final accuracy. RKD also introduces an assisted model. The assistant is to learn the residual error between the feature maps of the student and teacher in KD. We can regard it as an improved strategy than Kullback-Liebler (KL) divergence in KD. But when we also need to consider the hardware acceleration restrictions in RKD, the accuracy is even lower than KD. Different from RKD, CRD does not introduce another network. It improves the KL by distilling the knowledge from the representation differences of the student and teacher in the “latent space”. CRD outperforms KD in some tasks. However, the accuracy of CRD is still lower than MIS. The results of RKD and CRD prove that the inherent success of MIS does not only rely on introducing a recovered compressed model but also on what should be learned from this recovered model. MIS introduces two distillation loss items to learn the inherent discrepancy between the representation capacities of the dense and the compressed model, and the discrepancy introduced by hardware acceleration restrictions between two compressed models. So all of these key differences from KD, RKD and CRD contribute to the good effectiveness of MIS.

We apply the Class Activation Mapping (CAM) tool [53] to the healthy model $M_H$, the recovered model $M_R$ and the surgical model $M_S$ for ResNet-50. CAM can highlight the importance of the image region to the final prediction. The visualization results are shown in Figure 3.

For CAM, the red color highlight the “attention” area of each model. Though the surgical model is restricted by the hardware acceleration requirements, the CAMs of $M_H$, $M_R$ and $M_S$ respectively.
and $M_S$ all focus on the inherent features of the Malinois in ground truth image, which leading to the right classification.

We can also find even without the ground truth info from the training set, MIS can still achieve satisfactory accuracy. We show the accuracy curve in Figure 4. MIS in unsupervised training will obviously lower the accuracy of the training dataset. However, the accuracy during testing has less influence. The distillation between the different representation capacities of the dense and the irregular-compressed model helps MIS to improve the generalization without ground truth.

Figure 4. Accuracy change trends during MIS process.

We change the healthy model with a more accurate one to verify whether it can further improve the effect of MIS. We use the pre-trained ResNeXt-101 from TorchVision as the healthy model. The results are shown in Table 6.

Table 6. MIS with more accurate healthy model.

<table>
<thead>
<tr>
<th>Model</th>
<th>Sparsity Ratio</th>
<th>Recovered Model Accuracy</th>
<th>Surgical Model Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Top-1 (%)</td>
<td>Top-5 (%)</td>
</tr>
<tr>
<td>ResNet-50</td>
<td>0%</td>
<td>76.110</td>
<td>92.862</td>
</tr>
<tr>
<td></td>
<td>70%-FINE</td>
<td>76.496</td>
<td>93.080</td>
</tr>
<tr>
<td></td>
<td>85%-FINE</td>
<td>75.670</td>
<td>92.682</td>
</tr>
<tr>
<td></td>
<td>90%-FINE</td>
<td>74.680</td>
<td>92.298</td>
</tr>
<tr>
<td></td>
<td>95%-FINE</td>
<td>71.830</td>
<td>90.646</td>
</tr>
<tr>
<td></td>
<td>70%-BLK</td>
<td>76.452</td>
<td>92.990</td>
</tr>
<tr>
<td></td>
<td>80%-SUR</td>
<td>75.518</td>
<td>92.670</td>
</tr>
</tbody>
</table>

Table 7. MIS acceleration on GPUs.

From the results, we can conclude the compressed models by MIS can get a considerable acceleration effect on V100 and A100 GPUs. The acceleration in FP32 data type mainly comes from the reduction of memory bus utilization and memory access latency. The extra acceleration in FP16 data type on general-purpose GPUs comes from the full utilization of new sparse Tensor Core. The extra acceleration in FP16 data type on A100 GPU comes from the utilization of FP16 Tensor Core for irregular sparse pattern acceleration.

6. Conclusion

In this paper, we analyze the potential problems in knowledge distillation. Inspired by the principle of minimally invasive surgery, we propose a brand-new model compression method, MIS introduces an intermediate model as the “bridge”. We prove MIS changes the learning effective ratio and the probability distribution between easy and hard learning objects from information entropy and Bayes perspectives. With the comparison and ablation experiments, we show the success of MIS relies on learning the inherent discrepancy between the representation capacities of the dense and compressed model, and the discrepancy introduced by hardware acceleration restrictions between two compressed models. With MIS, we can change the irregular-compressed models into efficient forms and can get considerable acceleration in general-purpose GPUs.

For the open-source community, our experimental observations and proposed compression technique could be inspiring to the model compression field. Our study also provides good guidance for people who want to try the latest features for the newly announced A100 GPU.
References


[28] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael


