Minimally Invasive Surgery for Sparse Neural Networks in Contrastive Manner

Chong Yu[1,2]
[1]NVIDIA [2]Fudan 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 realtime 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 **M**inimally **I**nvasive **S**urgery(**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 **D**istillation (**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, **C**ontrastive **L**earning (**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 **M**inimally **I**nvasive **S**urgery (**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_R 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 *H*ardware *A*cceleration *R*equirements 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 τ, Loss adjustment factors α, β, γ, Overall loss threshold δ
Output: Surgical model M_S

```
Init surgical model M_S by recovered model M_R.
while L_Overall > δ do
    Pruning M_S to meet the hardware acceleration requirement: HAR(M_S).
    if Ground truth label (l_G) exists then
        Surgical prediction loss: 𝕃_P=𝕃(l_G,M_S(x;T = 1))
    else
        Surgical prediction loss: 𝕃_P=𝕃(M_H(x;T = 1),M_S(x;T = 1))
    end if
    Healthy-surgical distillation loss:
            𝕃_Dhs=𝕃(M_H(x;T = τ),M_S(x;T = τ))
    Recovered-surgical distillation loss:
            𝕃_Drs=𝕃(M_R(x;T = τ),M_S(x;T = τ))
    Calculate the overall loss: L_Overall=α*𝕃_P+β*𝕃_Dhs+γ*𝕃_Drs
    Minimize the overall loss: min[L_Overall]
end while
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_i and x_j 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

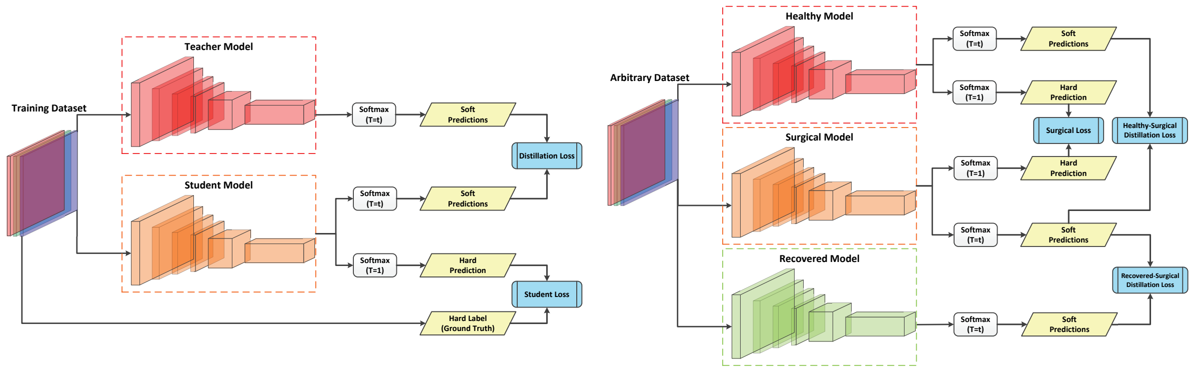

Figure 1. Workflows of vanilla **K**nowledge **D**istillation *(Left)* and **M**inimally **I**nvasive **S**urgery *(Right)*.

models with the training samples from different categories. **K**ullback-**L**eibler (**KL**) divergence is applied to measure the difference between the two representations. Ideally, the target can be denoted with the following formulas.

$$\boldsymbol{KL}(M_H(x_i), M_S(x_i)) \to 0, \qquad \boldsymbol{KL}(M_H(x_j), M_S(x_i)) \to \infty \tag{1}$$

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

$$\begin{cases} \min \sum_{k=1}^{N} \sum_{x_i \in \mathbb{D}(C_k)} [\boldsymbol{KL}(M_H(x_i), M_S(x_i))]^2 \\ \max \sum_{k=1}^{N} \sum_{k'=1}^{N} \sum_{x_i \in \mathbb{D}(C_k)} \sum_{x_j \in \mathbb{D}(C_{k'})}^{k' \neq k} [\boldsymbol{KL}(M_H(x_j), M_S(x_i))]^2 \end{cases} \tag{2}$$

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 $\mathbb{D}$ is defined as $T_{\mathbb{D}}$. 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 φ_H and φ_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 $D^{SH} < D^{SR}$.

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:

$$H\left(M_R\right) = H\left(M_H\right) D^{RH} = \left(T_{\mathbb{D}} + \varphi_H\right) D^{RH} \geq T_{\mathbb{D}} \tag{3}$$

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

$$H\left(M_S\right) = H\left(M_H\right) D^{SH} = \left(T_{\mathbb{D}} + \varphi_H\right) D^{SH} < \left(T_{\mathbb{D}} + \varphi_H\right) D^{RH} \tag{4}$$

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

$$H\left(M_S{}'\right) = H\left(M_R\right) D^{SR} = \left(T_{\mathbb{D}} + \varphi_R\right) D^{SR} < \left(T_{\mathbb{D}} + \varphi_H\right) D^{SR} \tag{5}$$

So we cannot make sure the distilled surgical models from the previous two situations still have enough information entropy to exceed the threshold $T_{\mathbb{D}}$.

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:

$$\begin{aligned} H\left(M_S{}''\right) &= H\left(M_H \cap \overline{M_R}\right) D^{SH} + H\left(M_R\right) D^{SR} \\ &= \left(\varphi_H - \varphi_R\right) D^{SH} + \left(T_{\mathbb{D}} + \varphi_R\right) D^{SR} \\ &= \left(T_{\mathbb{D}} + \varphi_H\right) D^{SH} + \left(T_{\mathbb{D}} + \varphi_R\right) \left(D^{SR} - D^{SH}\right) \end{aligned} \tag{6}$$

So we can find $H\left(M_S{}''\right) > H\left(M_S{}'\right)$ and $H\left(M_S{}''\right) > H\left(M_S\right)$ 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:

$$\begin{aligned} &P(C_H = 1) = Acc_H, \qquad P(C_H = 0) = 1 - Acc_H \\ &P(C_R = 1) = Acc_R, \qquad P(C_R = 0) = 1 - Acc_R \\ &P(C_S = 1) = Acc_R, \qquad P(C_S = 0) = 1 - Acc_R \end{aligned} \tag{7}$$

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.,

$$\begin{aligned} U &\Rightarrow M_H(x) \doteq M_S(x), \qquad \overline{U} \Rightarrow M_H(x) \neq M_S(x) \\ V &\Rightarrow M_R(x) \doteq M_S(x), \qquad \overline{V} \Rightarrow M_R(x) \neq M_S(x) \end{aligned} \tag{8}$$

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

$$P(C_S = 1) = P(C_S = 1 \mid U)P(U) + P(C_S = 1 \mid \overline{U})P(\overline{U}) \tag{9}$$

For the proposed **MIS** method:

$$\begin{aligned} P(C_S = 1) = {} & P(C_S = 1 \mid U, V)P(U, V) + P(C_S = 1 \mid \overline{U}, V)P(\overline{U}, V) \\ & + P(C_S = 1 \mid U, \overline{V})P(U, \overline{V}) + P(C_S = 1 \mid \overline{U}, \overline{V})P(\overline{U}, \overline{V}) \end{aligned} \tag{10}$$

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

$$P(V) = 1, \qquad P(\overline{V}) = 0 \tag{11}$$

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:

$$P(U \mid C_S = 1) = D_1^{SH}, \qquad P(U \mid C_S = 0) = D_0^{SH} \tag{12}$$

According to the Bayes theorem, the posterior probability for the right classification ($C_S = 1$) when the representations from the healthy model and the surgical model are similar is given by:

$$\begin{aligned} P(C_S = 1 \mid U) &= \frac{P(U \mid C_S = 1)P(C_S = 1)}{P(U \mid C_S = 1)P(C_S = 1) + P(U \mid C_S = 0)P(C_S = 0)} \\ &= \frac{D_1^{SH} Acc_R}{D_1^{SH} Acc_R + D_0^{SH}(1 - Acc_R)} \end{aligned} \tag{13}$$

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:

$$\begin{aligned} P(C_S = 1 \mid \overline{U}) &= \frac{P(\overline{U} \mid C_S = 1)P(C_S = 1)}{P(\overline{U} \mid C_S = 1)P(C_S = 1) + P(\overline{U} \mid C_S = 0)P(C_S = 0)} \\ &= \frac{(1 - D_1^{SH}) Acc_R}{(1 - D_1^{SH}) Acc_R + (1 - D_0^{SH})(1 - Acc_R)} \end{aligned} \tag{14}$$

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

$$\begin{aligned} P(C_S = 1 \mid \overline{U}, V) &= \frac{P(C_S = 1)P(\overline{U} \mid C_S = 1)P(V \mid C_S = 1, \overline{U})}{P(\overline{U})P(V \mid \overline{U})} \\ &= P(C_S = 1 \mid \overline{U}) \frac{P(V \mid C_S = 1, \overline{U})}{P(V \mid \overline{U})} \end{aligned} \tag{15}$$

$$\begin{aligned} P(C_S = 1 \mid \overline{U}, \overline{V}) &= \frac{P(C_S = 1)P(\overline{U} \mid C_S = 1)P(\overline{V} \mid C_S = 1, \overline{U})}{P(\overline{U})P(\overline{V} \mid \overline{U})} \\ &= P(C_S = 1 \mid \overline{U}) \frac{P(\overline{V} \mid C_S = 1, \overline{U})}{P(\overline{V} \mid \overline{U})} \end{aligned} \tag{16}$$

Compare the total probability formulas (9) and (10), **MIS** method divide the last item of vanilla **KD** into two parts.

$$\begin{aligned} &\textbf{KD}: \quad P(C_S = 1 \mid \overline{U})P(\overline{U}) \\ &\textbf{MIS}: \quad P(C_S = 1 \mid \overline{U}, V)P(\overline{U}, V) + P(C_S = 1 \mid \overline{U}, \overline{V})P(\overline{U}, \overline{V}) \end{aligned} \tag{17}$$

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 $D^{SH} < D^{SR}$. What is 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 ***APEX***[1] 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 (α, β and γ) 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 ***TorchVision***[2] are chosen as the experiment target models. The original sparse models serve as M_R are trained with the public ***Distiller*** library[3] [56].

[1] https://github.com/NVIDIA/apex.
[2] https://github.com/pytorch/vision.
[3] https://github.com/NervanaSystems/distiller.

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 (α), the healthy-surgical distillation loss (β) and the recovered-surgical distillation loss (γ) apply 1, 10, 50, respectively. (The variance is within ± 0.17 for Top-1, and ± 0.15 for Top-5 accuracy with different random seeds.)

Model	Healthy Model Accuracy		Sparsity Ratio	Recovered Model Accuracy		Surgical Model Accuracy	
	Top-1 (%)	Top-5 (%)		Top-1 (%)	Top-5 (%)	Top-1 (%)	Top-5 (%)
ResNet-50	76.130	92.862	70%-FINE	76.496	93.080	75.910	92.650
			85%-FINE	75.670	92.682	75.198	92.280
			90%-FINE	74.680	92.298	74.156	91.874
			95%-FINE	71.830	90.646	71.414	90.288
			70%-BLK	76.452	92.990	76.224	92.852
			80%-SUR	75.538	92.670	75.162	92.390
ResNeXt-101	78.188	93.886	75%-FINE	79.078	94.468	**79.254**	**94.544**
			85%-FINE	78.764	94.368	**78.880**	**94.398**
			90%-FINE	78.530	94.110	**78.584**	**94.154**
			95%-FINE	76.922	93.574	**77.058**	**93.596**
			75%-BLK	79.063	94.404	**79.173**	**94.471**
			80%-SUR	78.631	94.356	**78.845**	**94.502**
VGG-19	74.246	91.838	50%-FINE	75.578	92.694	75.552	**92.732**
			75%-FINE	73.716	91.499	**73.724**	**91.513**
			90%-FINE	73.435	91.358	**73.437**	**91.361**
			75%-BLK	73.689	91.443	**73.721**	**91.493**
			80%-SUR	73.523	91.406	**73.620**	**91.469**
Inception-V3	77.568	93.644	50%-FINE	78.204	93.998	78.092	**94.014**
			75%-FINE	77.832	93.762	**77.904**	**93.794**
			90%-FINE	77.335	93.601	**77.453**	**93.604**
			75%-BLK	77.689	93.599	**77.717**	**93.620**
			80%-SUR	77.495	93.622	**77.518**	**93.639**
DenseNet-161	77.114	93.578	50%-FINE	78.564	94.280	78.422	94.176
			75%-FINE	77.745	93.835	**77.750**	**93.912**
			90%-FINE	77.201	93.576	**77.310**	**93.611**
			75%-BLK	77.668	93.697	**77.676**	**93.701**
			80%-SUR	77.504	93.601	**77.515**	**93.650**
MobileNet-V2	71.880	90.290	50%-FINE	69.023	88.765	**70.804**	**88.918**
			75%-FINE	68.371	88.303	**68.500**	**88.412**

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 repository***[6] are chosen as the experiment target models. The original sparse models serve as M_R are compressed with **AGP** method and trained with the ***Distiller*** library[3]. 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. ***1x*** 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 the ground truth info provided by **COCO** [25] dataset. 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, 10, 15.

Model	Healthy Model		Sparsity Ratio	Recovered Model		Surgical Model	
	Box AP	Box AR		Box AP	Box AR	Box AP	Box AR
Faster R-CNN(R50-1x)	37.65(±0.12)	52.14(±0.17)	50%	38.58(±0.11)	53.04(±0.16)	38.76(±0.14)	53.05(±0.17)
			75%	36.67(±0.14)	51.31(±0.21)	36.57(±0.13)	51.42(±0.19)
Faster R-CNN(R50-3x)	39.79(±0.14)	52.14(±0.17)	50%	39.96(±0.13)	53.97(±0.15)	39.89(±0.12)	53.92(±0.14)
			75%	38.85(±0.16)	52.92(±0.16)	38.94(±0.15)	53.21(±0.18)
Faster R-CNN(R101-3x)	41.92(±0.16)	55.55(±0.11)	50%	42.03(±0.14)	55.53(±0.19)	42.01(±0.11)	55.65(±0.18)
			75%	41.12(±0.18)	55.11(±0.22)	41.11(±0.15)	55.23(±0.20)
Faster R-CNN(X101-3x)	43.08(±0.12)	55.63(±0.09)	50%	42.59(±0.15)	55.74(±0.17)	42.68(±0.14)	55.83(±0.18)
			75%	42.52(±0.18)	55.63(±0.21)	42.63(±0.19)	55.74(±0.19)
RetinaNet(R50-1x)	36.45(±0.15)	53.36(±0.18)	50%	37.43(±0.16)	53.82(±0.14)	37.42(±0.17)	54.11(±0.12)
			75%	34.85(±0.15)	51.84(±0.19)	34.81(±0.16)	51.93(±0.18)
RetinaNet(R50-3x)	38.45(±0.14)	54.34(±0.16)	50%	37.44(±0.17)	53.71(±0.14)	37.55(±0.16)	53.81(±0.19)
			75%	37.40(±0.18)	53.33(±0.20)	37.43(±0.15)	53.28(±0.15)
RetinaNet(R101-3x)	40.04(±0.13)	55.61(±0.15)	50%	39.33(±0.14)	55.22(±0.19)	39.27(±0.15)	55.07(±0.18)
			75%	39.22(±0.15)	54.32(±0.22)	39.06(±0.17)	54.33(±0.18)
SSD(R50)	25.11(±0.08)	36.13(±0.11)	50%	25.83(±0.17)	36.91(±0.20)	25.72(±0.16)	36.80(±0.19)
			75%	24.90(±0.22)	35.88(±0.24)	24.86(±0.20)	35.93(±0.21)
Mask R-CNN(R50-1x)	39.91(±0.23)	54.42(±0.11)	50%	39.79(±0.17)	53.92(±0.18)	40.21(±0.15)	54.62(±0.16)
			75%	37.27(±0.16)	52.01(±0.20)	37.41(±0.16)	52.13(±0.15)
Mask R-CNN(R50-3x)	40.62(±0.19)	54.53(±0.12)	50%	40.70(±0.14)	54.63(±0.17)	40.84(±0.16)	54.50(±0.18)
			75%	39.90(±0.14)	54.24(±0.18)	39.75(±0.17)	54.22(±0.15)
Mask R-CNN(R101-3x)	42.92(±0.17)	56.51(±0.11)	50%	43.21(±0.19)	56.83(±0.13)	43.01(±0.16)	56.55(±0.17)
			75%	42.04(±0.16)	56.01(±0.18)	42.16(±0.15)	56.03(±0.20)
Mask R-CNN(X101-3x)	44.13(±0.14)	56.92(±0.12)	50%	43.95(±0.20)	55.81(±0.24)	43.89(±0.18)	55.74(±0.19)
			75%	43.62(±0.19)	56.32(±0.21)	43.80(±0.17)	56.29(±0.17)

Table 2. **MIS** effectiveness on detection 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***[6] and Transformer [44] from ***Fairseq***[7] 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 (α), the healthy-surgical distillation loss (β) and the recovered-surgical distillation loss (γ) apply 1, 2, 5.

Model	Healthy Model	Sparsity Ratio	Recovered Model	Surgical Model
	BLEU Score		BLEU Score	BLEU Score
GNMT(WMT16 En-Ge)	24.37(±0.20)	50%	24.77(±0.16)	24.73(±0.15)
		75%	24.67(±0.14)	24.69(±0.12)
		90%	24.30(±0.13)	24.31(±0.11)
Transformer(WMT14 En-Ge)	28.65(±0.10)	50%	28.89(±0.11)	28.91(±0.12)
		75%	28.79(±0.09)	28.77(±0.10)
		90%	28.15(±0.12)	28.21(±0.11)
Transformer(WMT16 En-Ge)	27.79(±0.13)	50%	28.01(±0.14)	28.03(±0.13)
		75%	27.99(±0.13)	27.97(±0.13)
		90%	27.65(±0.11)	27.70(±0.10)

Table 3. **MIS** effectiveness on translation task.

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

[4]Notice 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***.

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

[6]https://github.com/NVIDIA/DeepLearningExamples.

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

[8]http://www.statmt.org/wmt16/translation-task.html.

[9]https://github.com/twtygqyy/pytorch-SRResNet.

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.

Dataset	Healthy Model		Sparsity Ratio	Recovered Model		Surgical Model	
	PSNR	SSIM		PSNR	SSIM	PSNR	SSIM
Set5	31.803	0.863	50%	31.234	0.870	**31.484**	**0.872**
			75%	31.145	0.862	**31.301**	0.861
			90%	30.989	0.854	**31.004**	**0.856**
Set14	28.643	0.726	50%	28.315	0.755	**28.417**	0.754
			75%	28.275	0.750	**28.369**	**0.753**
			90%	28.012	0.743	**28.134**	**0.747**
DIV2K	29.256	0.788	50%	28.926	0.811	**29.025**	0.810
			75%	28.795	0.793	**28.918**	**0.798**
			90%	28.423	0.735	**28.506**	**0.740**

Table 4. **MIS** effectiveness on super resolution task.

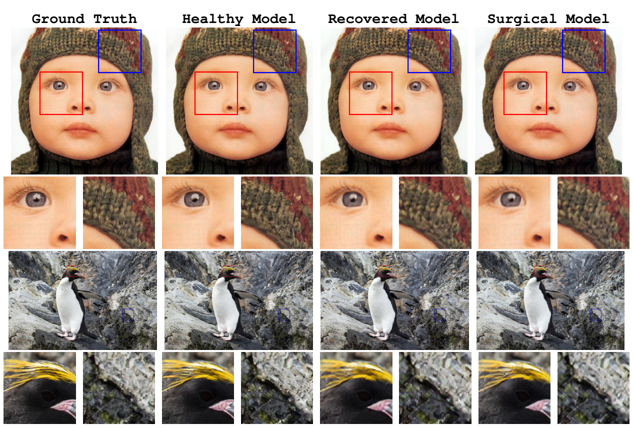

Figure 2. Representative super resolution results with enlargements of boxed areas (The Recovered Model and Surgical Model are compressed to 50% sparse level).

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 **R**esidual **K**nowledge **D**istillation [9] (**RKD**) and **C**ontrastive **R**epresentation **D**istillation [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	Algorithm	Sparsity Ratio	Model Accuracy	
			Top-1 (%)	Top-5 (%)
ResNet-50	Baseline	0%	76.130	92.862
	BLK	70%	76.452	92.990
	AGP	70%	76.496	93.080
	KD	70%	75.950	92.710
	RKD	70%	75.474	93.124
	CRD	70%	76.432	93.190
	MIS(Unsupervised)	70%	75.910	92.650
	MIS(Supervised)	70%	**76.558**	**93.188**
ResNeXt-101	Baseline	0%	78.188	93.886
	BLK	75%	79.063	94.404
	AGP	75%	79.078	94.468
	KD	75%	79.114	94.466
	RKD	75%	78.954	94.482
	CRD	75%	78.958	94.462
	MIS(Unsupervised)	75%	79.254	94.544
	MIS(Supervised)	75%	**79.348**	**94.682**

Table 5. Ablation experiment on image classification task.

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 **CAM**s of M_H, M_R

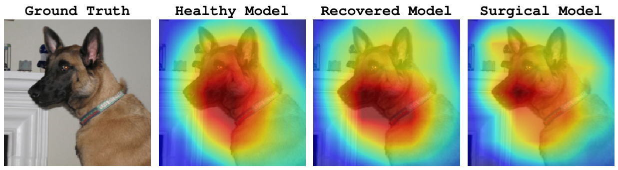

Figure 3. Class activation mapping visualization. (The Recovered Model and Surgical Model are compressed to 80% sparse level).

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.

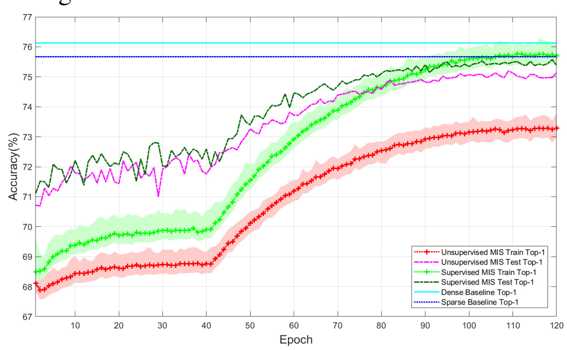

Figure 4. Accuracy change trends during **MIS** process.

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***[2] as the healthy model. The results are shown in Table 6.

Model	Sparsity Ratio	Recovered Model Accuracy		Surgical Model Accuracy	
		Top-1 (%)	Top-5 (%)	Top-1 (%)	Top-5 (%)
ResNet-50	0%	76.130	92.862	N/A	N/A
ResNeXt-101	0%	78.188	93.886	N/A	N/A
ResNet-50	70%-FINE	76.496	93.080	**77.038**	**93.240**
	85%-FINE	75.670	92.682	**75.836**	**92.704**
	90%-FINE	74.680	92.298	**74.796**	92.208
	95%-FINE	71.830	90.646	**71.964**	90.638
	70%-BLK	76.452	92.990	**77.112**	**93.304**
	80%-SUR	75.538	92.670	**75.820**	**92.738**

Table 6. **MIS** with more accurate healthy model.

From the results, we can conclude a more accurate healthy model can bring extra benefit to accuracy. It also proves that MIS can be used when dense and compressed models have different structures. This is not realizable for the model compression methods which rely on distillation from pure feature maps, like **LIT** [21].

5.6. Acceleration performance

We measure the acceleration performance of the compressed models by **MIS** on general-purposed hardware. In this experiment, we choose the V100 [31] and A100 [32] GPUs which can access from the cloud service as the testing platforms. We measure the performance in FP32 and FP16 data types, respectively. The acceleration results are shown in Table 7. The performance reported in Table 7 is the acceleration ratio. The baseline (1.0**X**) means the real performance of the dense healthy models with FP32 data type on V100 GPU.

Task	Model	V100 GPU		A100 GPU	
		FP32	FP16	FP32	FP16
Classification	ResNet-50	1.12 ~ 1.22**X**	8.77 ~ 9.50**X**	1.37 ~ 1.49**X**	21.47 ~ 23.44**X**
	ResNeXt-101	1.07 ~ 1.18**X**	8.21 ~ 9.09**X**	1.29 ~ 1.41**X**	20.77 ~ 22.81**X**
	VGG-19	1.15 ~ 1.27**X**	8.89 ~ 9.63**X**	1.37 ~ 1.53**X**	22.35 ~ 24.76**X**
	Inception-V3	1.14 ~ 1.24**X**	8.84 ~ 9.62**X**	1.37 ~ 1.51**X**	22.19 ~ 24.63**X**
	DenseNet-161	1.17 ~ 1.26**X**	9.16 ~ 9.84**X**	1.43 ~ 1.54**X**	23.05 ~ 24.84**X**
	MobileNet-V2	1.04 ~ 1.16**X**	8.11 ~ 9.01**X**	1.23 ~ 1.37**X**	19.56 ~ 21.49**X**
Detection	Faster R-CNN	1.15 ~ 1.23**X**	9.02 ~ 9.54**X**	1.37 ~ 1.47**X**	22.33 ~ 24.15**X**
	RetinaNet	1.15 ~ 1.25**X**	9.07 ~ 9.73**X**	1.35 ~ 1.46**X**	22.36 ~ 24.20**X**
	SSD	1.08 ~ 1.19**X**	8.57 ~ 9.45**X**	1.33 ~ 1.47**X**	21.28 ~ 23.47**X**
	Mask R-CNN	1.13 ~ 1.24**X**	8.98 ~ 9.86**X**	1.39 ~ 1.53**X**	22.32 ~ 24.54**X**
Translation	GNMT	1.15 ~ 1.26**X**	9.11 ~ 9.94**X**	1.41 ~ 1.54**X**	22.63 ~ 24.68**X**
	Transformer	1.25 ~ 1.35**X**	9.88 ~ 10.68**X**	1.53 ~ 1.66**X**	24.54 ~ 26.53**X**
Super Resolution	SRResNet	1.13 ~ 1.24**X**	8.94 ~ 9.81**X**	1.39 ~ 1.52**X**	22.28 ~ 24.38**X**

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 V100 GPU comes from the utilization of FP16 Tensor Core. The extra acceleration in FP16 data type on A100 GPU comes from the full utilization of new sparse 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

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 126–135, 2017. 6

[2] Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9163–9171, 2019. 2

[3] Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning. *arXiv preprint arXiv:1902.09229*, 2019. 2

[4] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. 2012. 6

[5] Cristian Bucilu, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541, 2006. 2

[6] Robin Cheong and Robel Daniel. transformers. zip: Compressing transformers with pruning and quantization. Technical report, tech. rep., Stanford University, Stanford, California, 2019. 6

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1

[9] Mengya Gao, Yujun Shen, Quanquan Li, and Chen Change Loy. Residual knowledge distillation. *arXiv preprint arXiv:2002.09168*, 2020. 2, 7

[10] Yiwen Guo, Anbang Yao, and Yurong Chen. Dynamic network surgery for efficient dnns. In *Advances in neural information processing systems*, pages 1379–1387, 2016. 2, 6

[11] Song Han, Xingyu Liu, Huizi Mao, Jing Pu, Ardavan Pedram, Mark A Horowitz, and William J Dally. Eie: efficient inference engine on compressed deep neural network. *ACM SIGARCH Computer Architecture News*, 44(3):243–254, 2016. 3

[12] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Advances in neural information processing systems*, pages 1135–1143, 2015. 1, 2

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 6

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 1, 2

[16] Zejiang Hou and Sun-Yuan Kung. Efficient image super resolution via channel discriminative deep neural network pruning. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3647–3651. IEEE, 2020. 6

[17] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 1

[18] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 5

[19] Quan Huynh-Thu and Mohammed Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics letters*, 44(13):800–801, 2008. 7

[20] Sambhav R Jain, Albert Gural, Michael Wu, and Chris Dick. Trained uniform quantization for accurate and efficient neural network inference on fixed-point hardware. *arXiv preprint arXiv:1903.08066*, 2019. 1

[21] Animesh Koratana, Daniel Kang, Peter Bailis, and Matei Zaharia. Lit: Learned intermediate representation training for model compression. In *International Conference on Machine Learning*, pages 3509–3518, 2019. 2, 8

[22] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017. 6

[23] Namhoon Lee, Thalaiyasingam Ajanthan, and Philip HS Torr. Snip: Single-shot network pruning based on connection sensitivity. *arXiv preprint arXiv:1810.02340*, 2018. 2, 6

[24] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 6

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 6

[26] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 6

[27] Huizi Mao, Song Han, Jeff Pool, Wenshuo Li, Xingyu Liu, Yu Wang, and William J Dally. Exploring the granularity of sparsity in convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 13–20, 2017. 2

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

Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training. *arXiv preprint arXiv:1710.03740*, 2017. 1, 5

[29] Maxim Naumov, Dheevatsa Mudigere, Hao-Jun Michael Shi, Jianyu Huang, Narayanan Sundaraman, Jongsoo Park, Xiaodong Wang, Udit Gupta, Carole-Jean Wu, Alisson G Azzolini, et al. Deep learning recommendation model for personalization and recommendation systems. *arXiv preprint arXiv:1906.00091*, 2019. 1

[30] Anh Nguyen, Jeff Clune, Yoshua Bengio, Alexey Dosovitskiy, and Jason Yosinski. Plug & play generative networks: Conditional iterative generation of images in latent space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4467–4477, 2017. 2

[31] NVIDIA. NVIDIA Tesla V100 GPU Architecture, 2017. 1, 3, 5, 8

[32] NVIDIA. NVIDIA A100 Tensor Core GPU Architecture, 2020. 1, 3, 5, 8

[33] NVIDIA. NVIDIA Tensor Core, 2020. 5

[34] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2

[35] Angshuman Parashar, Minsoo Rhu, Anurag Mukkara, Antonio Puglielli, Rangharajan Venkatesan, Brucek Khailany, Joel Emer, Stephen W Keckler, and William J Dally. Scnn: An accelerator for compressed-sparse convolutional neural networks. *ACM SIGARCH Computer Architecture News*, 45(2):27–40, 2017. 3

[36] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017. 5

[37] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 6

[38] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014. 2

[39] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 5

[40] K Sato. An in-depth look at google's first tensor processing unit (tpu). *Google Cloud Platform*, 2017. 1

[41] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1, 5

[42] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 5

[43] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. *arXiv preprint arXiv:1910.10699*, 2019. 2, 7

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 6

[45] Mark Vierra, MD. Minimally invasive surgery. *Annual review of medicine*, 46(1):147–158, 1995. 3

[46] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 7

[47] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. In *Advances in neural information processing systems*, pages 2074–2082, 2016. 3

[48] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016. 1, 6

[49] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 1, 5

[50] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4133–4141, 2017. 2

[51] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016. 2

[52] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *International conference on curves and surfaces*, pages 711–730. Springer, 2010. 6

[53] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 7

[54] Michael Zhu and Suyog Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression. In *International Conference on Learning Representations*, 2018. 2, 6

[55] Zmora. Block pruning using L1-norm ranking and AGP, 2019. 6

[56] Neta Zmora, Guy Jacob, and Gal Novik. Neural network distiller. *URL https://zenodo. org/record/1297430*, 2018. 1, 5