Block-wise Intermediate Representation Training for Model Compression

Animesh Koratana[*], **Daniel Kang**[*], **Peter Bailis, Matei Zaharia**

Abstract

Knowledge distillation (KD) is a popular method for reducing the computational overhead of deep network inference, in which the output of a teacher model is used to train a smaller, faster student model. Hint training (i.e., FitNets) extends KD by regressing a student model's intermediate representation (IR) to a teacher model's IR. In this work, we introduce b**L**ock-wise **I**ntermediate representation **T**raining (LIT), a novel model compression technique that extends the use of IRs in deep network compression, outperforming KD and hint training. LIT has two key ideas: 1) LIT trains a student of the same width (but shallower depth) as the teacher by directly comparing the IRs, and 2) LIT uses the IR from the previous block in the teacher model as an input to the current student block during training, avoiding unstable IRs in the student network. We show that LIT provides substantial reductions in network depth *without loss in accuracy* — for example, LIT can compress a ResNeXt-110 to a ResNeXt-20 (5.5×) on CIFAR10 and a VDCNN-29 to a VDCNN-9 (3.2×) on Amazon Reviews, outperforming KD and hint training in network size for a given accuracy. Finally, we show that LIT can effectively compress GAN generators, which are not supported in the KD framework because GANs output pixels as opposed to probabilities.

1 Introduction

Modern deep networks have achieved increased accuracy by continuing to introduce more layers [1, 2] at the cost of higher computational overhead. In response, researchers have proposed many techniques to reduce this computational overhead at inference time, which broadly fall under two categories. First, in deep compression [3, 4, 5, 6], parts of a model are removed or quantized to reduce the number of weights and/or the computational footprint.[2] However, deep compression techniques typically require new hardware [7] to take advantage of the resulting model sparsity. Second, in student/teacher methods—introduced in knowledge distillation (KD) [8] and further extended [9, 10, 11]—a smaller student model learns from a large teacher model through distillation loss, wherein the student model attempts to match the logits of the teacher model. As there are no constraints on the teacher and student models, KD can produce hardware-friendly models: the student can be a standard model architecture (e.g., ResNet), optimized for a given hardware substrate.

Hint training (i.e., FitNets [9]) extends KD by using a teacher's intermediate representation (IR, i.e., the output from a hidden layer) to guide the training of the student model. The authors show that hint training with a single IR outperforms KD in compressing teacher networks (e.g., maxout networks [12]) to thinner and deeper student networks.

We ask the natural question: does hint training compress more modern, highly-structured, very deep networks—such as ResNet [2], VDCNN [13], and StarGAN [14]? We find that standard hint training (i.e., with a single hint) and training with multiple hints is not effective for modern deep networks.

[*] Equal Contribution

[2] In this work, we refer to this class of methods as "deep compression," and methods to reduce model size more generally as "model compression."

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

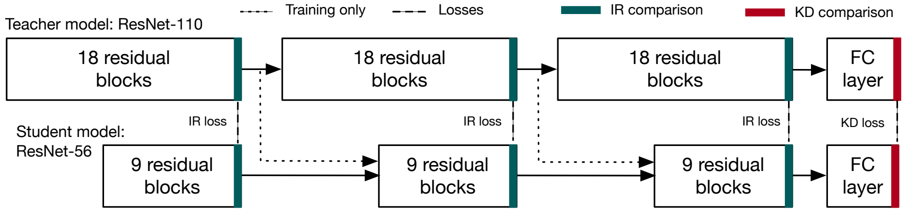


Figure 1: A schematic of LIT. In LIT, the teacher model's blocks are used as input to the student model's blocks during training, except for the first block. Specifically, denoting the blocks S_1,...,S_4 for the student and T_1,...,T_4 for the student, $S_2(T_1)$ is compared against T_2 in training and similarly for deeper parts of the network. S_1 and T_1 are directly compared. LIT additionally compares S and T through the KD loss. The teacher model is not updated in training.

We hypothesize that, for modern deep networks, hint training causes unstable IRs: the deepest network considered in [9] was only 17 layers, achieving 91.61% on CIFAR10; in contrast, a modern 110-layer ResNet achieves 93.68% on CIFAR10. Further experiments validating this hypothesis are given in [15]

In this work, we extend hint training's ability to transfer intermediate knowledge from teacher to student to reduce the depth of modern, highly-structured architectures (e.g., compressing a standard ResNeXt-110 to a standard ResNeXt-20 with no loss in accuracy). We do this via a novel method called b**L**ock-wise **I**ntermediate representation **T**raining (LIT), a student/teacher compression technique that outperforms training student networks from scratch, hint training, and KD. LIT targets highly structured, modern networks that consist of repetitive blocks (i.e., groups of layers) that can be scaled up/down for accuracy/speed trade-offs; for example, ResNets have standard configurations from 20 to hundreds of layers. LIT leverages two key ideas to reduce unstable IRs in deep networks. First, LIT directly trains student networks of the *same width* as the teacher model (as opposed to using a single, thinner hint as in hint training). Second, LIT avoids unstable student IRs deep in the network by using the IR from the *previous* block in the teacher model as input to the current student block during training; each student block is effectively trained in isolation to match the corresponding (deeper) block in the teacher. We show that LIT's block-wise training improves accuracy, allows for copying parts of the teacher model directly to the student model, and permits selective compression of networks (e.g., compressing one out of three blocks in a network and copying the rest). For example, consider compressing a ResNet-56 from a ResNet-110 (Figure 1), each of which have four sections. The IR loss is applied to the output of each block, and the teacher model's IRs are used as input to the student blocks.

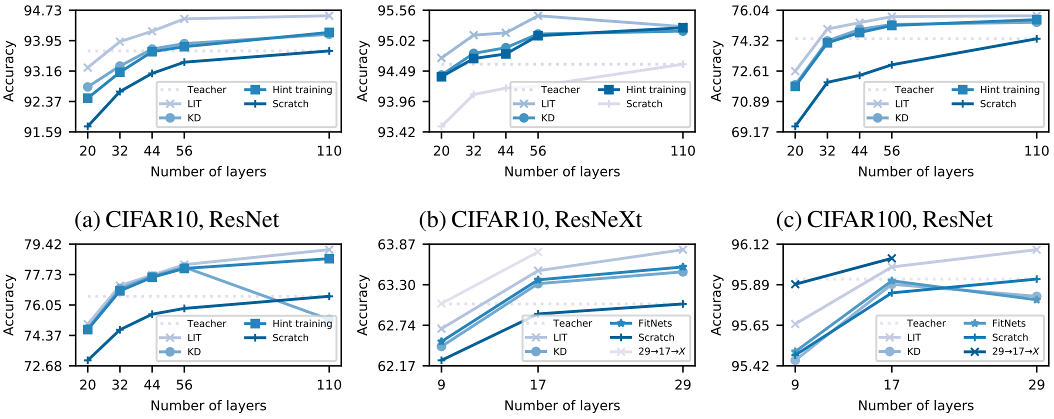
Because it is possible to transfer IRs directly, LIT is, to our knowledge, the first student/teacher compression method that works for GAN generators. LIT can compress GAN generators by only compressing the repetitive blocks present in certain GANs [14]. In contrast, KD does not apply directly as the KL divergence in KD loss operates on probabilities but not the pixels output by GAN generators. LIT can compress GANs by leveraging LIT's key property that, by matching the teacher IR dimensions, parts of the teacher network can be directly copied to the student network.

2 Experiments

We evaluate LIT's efficacy at compressing models on a range of tasks and models, including image classification, sentiment analysis, and image-to-image translation (GAN). Throughout, we use student and teacher networks with the same broad architecture (e.g., ResNet to ResNet). We consider ResNet [2], ResNeXt [16], VDCNN [13], and StarGAN [14]. We use standard architecture depths, widths, and learning rate schedules; details are given in [15].

LIT is effective at compressing a range of datasets and models. We ran LIT on a variety of models and datasets for image classification and sentiment analysis (CIFAR10, CIFAR100, Amazon Reviews). We additionally performed KD and hint training on these datasets and models.

Figure 2 shows the results for ResNet and ResNeXt for CIFAR10 and CIFAR100, and VDCNN on Amazon Reviews (full, polarity). LIT can compress models by up to 5.5× (CIFAR10, ResNeXt 110 to 20) on image classification and up to 3.2× on sentiment analysis (Amazon Reviews, VDCNN 29 to 9), with no loss in accuracy. LIT outperforms KD and hint training on all settings. We have found that, in some cases, training sequences of models using LIT results in higher performance. Thus, for VDCNN, we additionally compressed using LIT a VDCNN-29 to a VDCNN-17, and using this



(a) CIFAR10, ResNet

(b) CIFAR10, ResNeXt

(c) CIFAR100, ResNet

(d) CIFAR100, ResNeXt

(e) Amazon full, VDCNN

(f) Amazon polarity, VDCNN

Figure 2: The accuracy of ResNet and ResNeXt on CIFAR10/100 and VDCNN on Amazon Reviews when trained from scratch, trained via KD, and trained via LIT. The teacher models were ResNet-110, ResNeXt-110, and VDCNN-29/17. As shown, LIT outperforms KD for every student model. In some cases, KD can reduce the accuracy of the student model, as reported in [17].

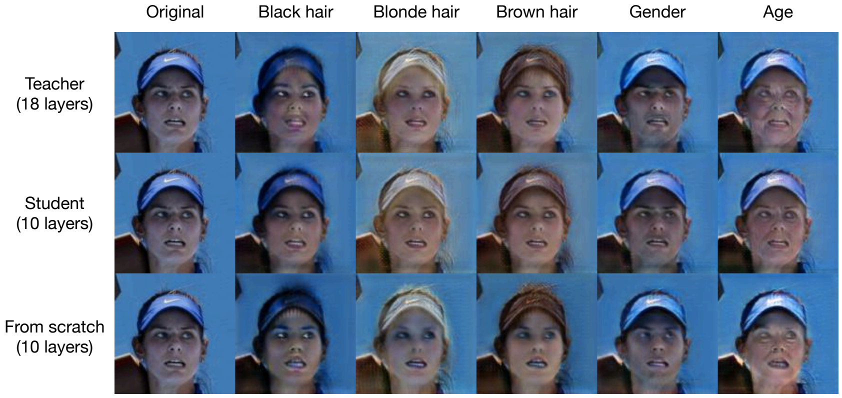


Figure 3: Selected images from the teacher (six residual blocks), student (two residual blocks), and trained from scratch (two residual blocks) StarGANs. As shown (column two, four), LIT can appear to improve GAN performance while significantly compressing models. Best viewed in color.

Model	Inception score (higher is better)	FID score (lower is better)
Teacher (18 layers)	3.49	6.43
LIT student (10 layers)	**3.56**	**5.84**
Trained from scratch (10 layers)	3.37	6.56

Table 1: Inception and FID scores for different versions of StarGAN. Despite having fewer layers than the teacher, the LIT student model achieves the best scores.

VDCNN-17, we trained a VDCNN-9 and VDCNN-17. We also found that in some cases, KD degrades the accuracy of student models when the teacher model is the same architecture (ResNeXt-110 on CIFAR100, VDCNN-29 on Amazon Reviews polarity). This corroborates prior observations in [17].

LIT can compress GANs. We compressed StarGAN's generator [14] using the LIT procedure with $\beta = 0$ (i.e., only using the intermediate representation loss). The original StarGAN has 18 total convolutional layers (including transposed convolutional layers), with 12 of the layers in the residual blocks (for a total of six residual blocks). We compressed the six residual blocks to two residual blocks (i.e., 12 to four layers) while keeping the rest of the layers fixed. The remaining layers for the teacher model were copied to the student model and fine-tuned. The discriminator remained fixed.

As shown in Table 1, LIT outperforms all baselines in inception and FID score. Additionally, as shown in Figure 3, the student model appears to perceptually outperform both the teacher model and equivalent model trained from scratch, suggesting LIT can serve as a form of regularization.

Acknowledgments

This research was supported in part by affiliate members and other supporters of the Stanford DAWN project—Ant Financial, Facebook, Google, Intel, Microsoft, NEC, Teradata, SAP, and VMware—as well as DARPA under No. FA8750-17-2-0095 (D3M), industrial gifts and support from Toyota Research Institute, Keysight Technologies, Hitachi, Northrop Grumman, NetApp, and the NSF under grants DGE-1656518 and CNS-1651570.

References

[1] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[2] 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.

[3] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[4] Chenzhuo Zhu, Song Han, Huizi Mao, and William J Dally. Trained ternary quantization. *arXiv preprint arXiv:1612.01064*, 2016.

[5] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016.

[6] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Quantized neural networks: Training neural networks with low precision weights and activations. *Journal of Machine Learning Research*, 18:187–1, 2017.

[7] 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. In *Computer Architecture (ISCA), 2016 ACM/IEEE 43rd Annual International Symposium on*, pages 243–254. IEEE, 2016.

[8] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *NIPS Deep Learning Workshop*, 2014.

[9] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *ICLR*, 2015.

[10] Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. *arXiv preprint arXiv:1606.07947*, 2016.

[11] Tommaso Furlanello, Zachary C Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. *arXiv preprint arXiv:1805.04770*, 2018.

[12] Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. *arXiv preprint arXiv:1302.4389*, 2013.

[13] Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. Very deep convolutional networks for text classification. *arXiv preprint arXiv:1606.01781*, 2016.

[14] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. *arXiv preprint*, 1711, 2017.

[15] Animesh Koratana, Daniel Kang, Peter Bailis, and Matei Zaharia. Lit: Block-wise intermediate representation training for model compression. *arXiv preprint arXiv:1810.01937*, 2018.

[16] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 5987–5995. IEEE, 2017.

[17] Asit Mishra and Debbie Marr. Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy. *arXiv preprint arXiv:1711.05852*, 2017.