Heterogeneous Knowledge Distillation using Information Flow Modeling

Supplementary Material

N. Passalis, M. Tzelepi and A. Tefas
Department of Informatics, Aristotle University of Thessaloniki, Greece
{passalis, mtzelepi, tefas}@csd.auth.gr

1. Datasets and Evaluation Setups

The proposed method was evaluated using four different datasets: the CIFAR-10 [5] dataset, the STL-10 [1] dataset, the CUB-200 [8] dataset and the SUN Attribute [7] dataset. For the CIFAR-10, the training split was used for training and transferring the knowledge to the student models, while for the retrieval evaluation the training split was also used to compile the database. Then, the test set was used to query the database and measure the retrieval performance of various representations. For the STL-10 dataset we followed the same setup as for the CIFAR-10, but we also used the provided unlabeded training split for transferring the knowledge to the student models. For the CUB-200 we also followed the same setup, however the experiments were conducted using the first 30 classes of the data, due to the significantly restricted learning capacity of the employed student models (recall that among the objectives of the paper is to evaluate the performance of KD approaches for ultra-lightweight network architectures and heterogeneous KD setups). Finally, images from the eight most common categories (for which at least 40 images exist) were used for training and evaluating the methods when the SUN Attribute dataset was employed, since a very small number of images exist for the rest of the categories. The 80% of the extracted images was used for training the networks and building the database, while the rest 20% was used to query the database. The evaluation process was repeated 5 times and the mean and standard deviation of the evaluated metrics are reported. For the SUN attribute dataset, the knowledge was distilled from a 2×2 HoG features.

For the CIFAR-10 and STL dataset we used the supplied images without performing any resizing (the original 32×32 images were used). However, the training dataset was augmented by randomly performing horizontal flipping and randomly cropping the images using padding of 4 pixels. A similar augmentation protocol was used for the CUB-200 dataset. However, the images of the CUB-200 dataset were first resized into 256×256 pixels and then a random crop of 224×224 pixels was used (a center crop of the same size was used during the evaluation process). Also, random rotation up to 20° was used when training the models. Finally, the images of the SUN attribute dataset were resized into 128×128 pixels, before feeding them into the network, following the protocol used in [6].

Figure 1. Network architectures used for the conducted experiments. The green model was used as the student for the conducted experiments (unless otherwise stated), while the red model was used as the auxiliary teacher. For experiments involving classification, an additional fully connected layer with N_C (number of classes) neurons was added.

2. Network Architectures

The network architectures used for the conducted experiments are shown in Fig. 1. The CNN-1 family was used for the experiments conducted using the CIFAR-10 and STL dataset, the CNN-2 family was used for the experiments conducted using the CUB-200 dataset, while the CNN-3 family was used for the SUN Attribute dataset. The suffix "-A" is used to denote the model that was used as the auxiliary teacher. The auxiliary teacher was trained using the PKT method [6], by transferring the knowledge from the penultimate layer of a ResNet-18 teacher (for the CIFAR-10, STL and CUB-200 datasets) or from handcrafted features (for the SUN Attribute dataset). The ReLU activation function was used for all the layers [2], while the batch normalization was used after each convolutional layer [3].

3. Training Hyper-parameters

For all the conducted experiments we used the Adam optimizer [4], with the default training hyper-parameters. For the experiments conducted using the CIFAR-10 dataset the optimization ran for 50 training epochs with a learning rate of 0.001 (batches of 128 samples were used) for all the evaluated methods. For the ablation results reported in Fig. 2 of the main manuscript the optimization ran for 20 epochs. For the STL dataset the optimization ran for 30 training epochs with a learning rate of 0.001 and batch size equal to 128. For the CUB-200 dataset the optimization ran for 100 training epochs, using a learning rate of 0.001 for the first 50 training epochs and 0.0001 for the subsequent 50 training epochs. Also, for the SUN Attribute dataset the optimization ran for 20 training epochs. Furthermore, the decay factor γ was set to 0.6 for this dataset, due to the smaller number of training epochs. Finally, note that for the experiments conducted with the contrastive supervision (CIFAR-10) we employed the contrastive loss with the margin set to 1 and the loss was combined with the KD loss after weighting it with 0.1. Also, for the classification experiments reported in Table 2, all the methods were also trained using a supervised classification term (cross-entropy loss). Finally, for all the experiments conducted using the distillation loss, a temperature of $T = 2$ was used.

References

[1] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the Conference on Artificial Intelligence and Statistics*, pages 215–223, 2011. 1

[2] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 315–323, 2011. 2

[3] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning*, pages 448–456, 2015. 2

[4] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*, pages 315–323, 2015. 2

[5] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Technical Report*, 2009. 1

[6] Nikolaos Passalis and Anastasios Tefas. Learning deep representations with probabilistic knowledge transfer. In *Proceedings of the European Conference on Computer Vision*, pages 268–284, 2018. 1, 2

[7] Genevieve Patterson and James Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2751–2758, 2012. 1

[8] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. 2010. 1