CMSC 5743

Efficient Computing of Deep Neural Networks

Lecture 10: Knowledge Distillation

Bei Yu
CSE Department, CUHK
byu@cse.cuhk.edu.hk

(Latest update: December 3, 2021)

Fall 2021

Introduction

Introduction

Knowledge distillation (KD) is a model compression method in which a small model is trained to mimic a pre-trained, larger model (or ensemble of models).

- The method was first proposed by[1] then generalized by[2].
- This training setting is sometimes referred to as "teacher-student", where the large model is the teacher and the small model is the student.
- In distillation, knowledge is transferred from the teacher model to the student by minimizing a loss function in which the target is the distribution of class probabilities predicted by the teacher model.
- Specifically, KD is accomplished by minimzing the KL divergence between the predictions of teacher and student.

[1] Cristian Buciluă, Rich Caruana, and Alexandru Niculescu-Mizil (2006). "Model compression". In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 535–541.

[2] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean (2015). "Distilling the knowledge in a neural network". In:

KL divergence

Kullback–Leibler divergence (KL divergence), D_{KL}[3], is a measure of how one probability distribution is different from a second

For discrete probability distributions P and Q defined on the same probability space, $\mathcal{X}$, the KL divergence from P to Q is:

$$D_{KL}(P||Q) = \sum_{x \in \mathcal{X}} P(x) \log(\frac{P(x)}{Q(x)}). \tag{1}$$

For distributions P and Q of a continuous random variable, the KL divergence from P to Q is:

$$D_{KL}(P||Q) = \int_{x \in \mathcal{X}} P(x) \log(\frac{P(x)}{Q(x)}) dx. \tag{2}$$

In the context of machine learning, D_{KL} is often called the information gain achieved if P would be used instead of Q which is currently used. By analogy with information theory, it is called the relative entropy of P with respect to Q.

[3] Solomon Kullback and Richard A Leibler (1951). "On information and sufficiency". In: vol. 22. 1. JSTOR, pp. 79–86.

KD Loss

We define a teacher network $T(.)$ and a student network $S(.)$. Typically, networks can produce class probabilities that converts the logits $\boldsymbol{z}_i$, computed for each class into a probability, $\boldsymbol{q}_i$ by using softmax function. We define the class probabilities $\boldsymbol{p}_i$ which is generated by teacher network $T(.)$ and $\boldsymbol{q}_i$ which is generated by student network $S(.)$ correspondingly. Then the KD Loss can be given by:

$$\mathcal{L}_{KD} = -\sum_{i=1}^{N} p(\boldsymbol{x}_i) \log(q(\boldsymbol{x}_i)) \tag{3}$$

where $p(\boldsymbol{x}_i) = \frac{\exp(\boldsymbol{v}_i/\tau)}{\sum_{j=1}^{C-1} \exp(\boldsymbol{v}_j/\tau)}$ and $q(\boldsymbol{x}_i) = \frac{\exp(\boldsymbol{z}_i/\tau)}{\sum_{j=1}^{C-1} \exp(\boldsymbol{z}_j/\tau)}$. Here C denotes the number of classes, τ iss the temperature parameter. $\boldsymbol{v}$ and $\boldsymbol{z}$ indicates the logits generated by teacher network and student network respectively.

Reltation to KL divergence

If we regard the class probablities p as soft label, then the KL Loss can be regarded as a softmax cross entropy. It's not hard to derive the relationships between cross entropy and KL divergence. Given p and q, we define entropy as $H(.)$, we can have:

$$\mathcal{L}_{KD} = -\sum_{i=1}^{N} p(\boldsymbol{x}_i)\log(q(\boldsymbol{x}_i)) = H(p,q). \tag{4}$$

$$\begin{aligned} D_{KL}(p||q) &= \mathbb{E}_p[-\log\frac{q}{p}] \\ &= \mathbb{E}_p[-\log q + \log p] \\ &= \mathbb{E}_p[-\log q] - \mathbb{E}_p[-\log p] \\ &= H(p,q) - H(p) \end{aligned} \tag{5}$$

Then we can have:

$$\mathcal{L}_{KD} = D_{KL} + H(p) \tag{6}$$

Relation to KL divergence

In knowledge distillation, we attempt to optimize the student network that can mimic the teacher network, then we can rewrite the our loss function:

$$\begin{aligned} \mathcal{L}_{KD} &= H(p, q_\theta) \\ &= D_{KL}(p||q_\theta) + H(p) \end{aligned} \tag{7}$$

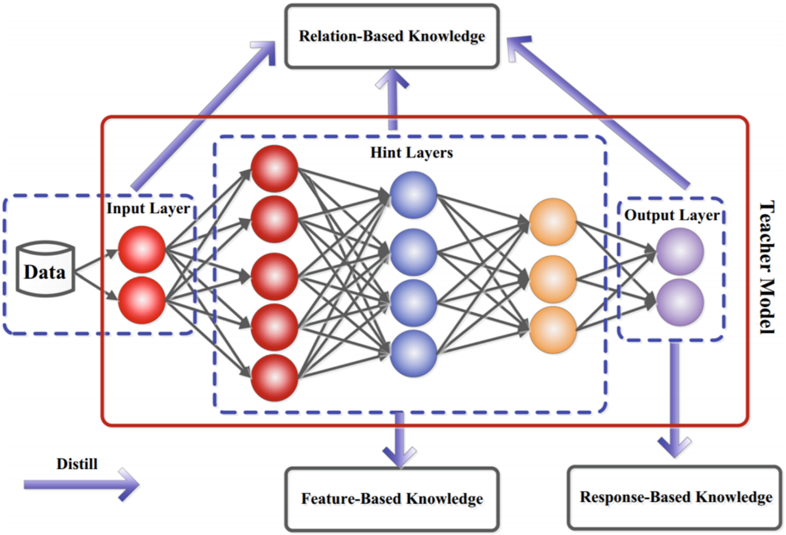
Since $H(p)$ is independent of θ, the optimization goal then becomes:

$$\begin{aligned} \operatorname*{argmin}_\theta \mathcal{L}_{KD} &= \operatorname*{argmin}_\theta H(p, q_\theta) \\ &= \operatorname*{argmin}_\theta D_{KL}(p||q_\theta) \\ &= \operatorname*{argmin}_\theta \mathcal{L}_{KL} \end{aligned} \tag{8}$$

Then we can find that optimizing KD loss is equivalent to optimize KL Loss in the knowledge distillation setting.

Knowledge Modeling

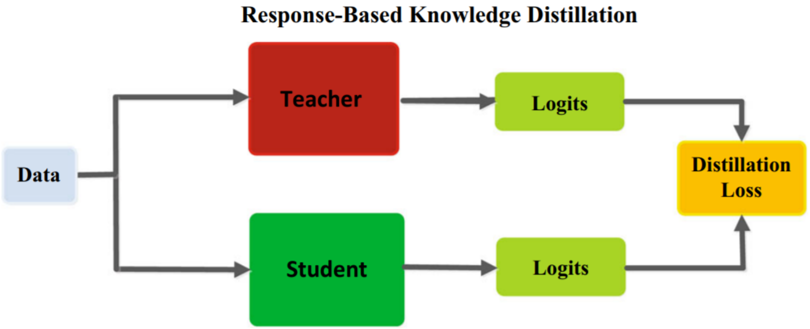
Overview


Relation-Based Knowledge
Hint Layers
Input Layer
Data
Output Layer
Teacher Model
Distill
Feature-Based Knowledge
Response-Based Knowledge

Response-Based Knowledge

Response-Based Knowledge

Response-based knowledge usually refers to the neural response of the last output layer of the teacher model. The main idea is to directly mimic the final prediction of the teacher model.



Response-Based Knowledge

Formulation

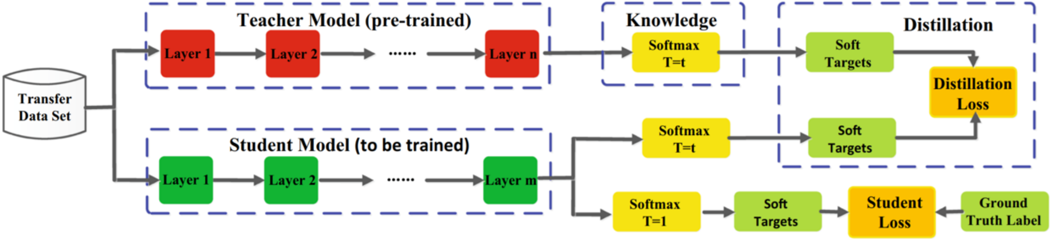
Given a vector of logits z as the outputs of the last fully connected layer of a deep model, the distillation loss for response-based knowledge can be formulated as:

$$L_{ResD}(z_t, z_s) = \mathcal{L}_R(z_t, z_s)$$

where $\mathcal{L}_R(\cdot)$ indicates the divergence loss of logits, and z_t and z_s are logits of teacher and student respectively.

Response-Based Knowledge

Example



Knowledge distillation proposed by[4].

[4] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean (2015). “Distilling the knowledge in a neural network”. In:

Response-Based Knowledge

Example

The most popular response-based knowledge for image classification is known as soft targets[5]. Specifically, soft targets are the probabilities that the input belongs to the classes and can be estimated by a softmax function as

$$p(z_i, T) = \frac{exp(z_i/T)}{\sum_j exp(z_j/T)}$$

where z_i is the logit for the i_{th} class, and a temperature factor T is introduced to control the importance of each soft target.Accordingly, the distillation loss for soft logits can be rewritten as

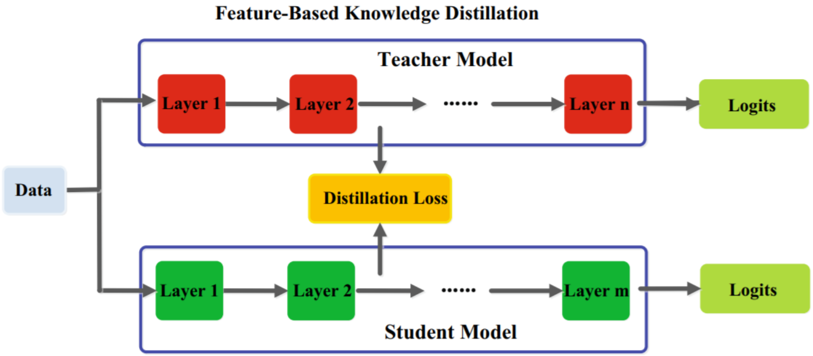
$$L_{ResD}(p(z_t, T), p(z_s, T)) = \mathcal{L}_R(p(z_t, T), p(z_s, T))$$

[5] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean (2015). "Distilling the knowledge in a neural network". In:

Feature-Based Knowledge

Feature-Based Knowledge

The output of intermediate layers, *i.e.*, feature maps, can also be used as the knowledge to supervise the training of the student model, which forged feature-based knowledge distillation. It is the improvement of response-based knowledge distillation.



Feature-Based Knowledge

Formulation

Generally, the distillation loss for feature-based knowledge transfer can be formulated as

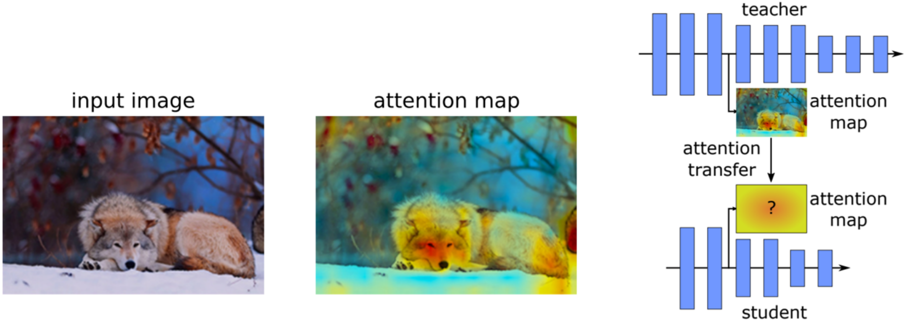
$$L_{FeaD}(f_t(x), f_s(x)) = \mathcal{L}_F(\phi_t(f_t(x)), \phi_s(f_s(x)))$$

where $f_t(x)$ and $f_s(x)$ are the feature maps of the intermediate layers of teacher and student models, respectively. The transformation functions, $\phi_t(f_t(x)$ and $\phi_s(f_s(x))$, are usually applied when the feature maps of teacher and student models are not in the same shape. $\mathcal{L}_F(\cdot)$ indicates the similarity function used to match the feature maps of teacher and student models. $\mathcal{L}_F(\cdot)$ can be $\mathcal{L}_2(\cdot)$, $\mathcal{L}_1(\cdot)$, $\mathcal{L}_{CE}(\cdot)$ and *etc*.

Feature-Based Knowledge

Example

[6] proposed a feature-based knowledge distillation using attention mechanism. Specifically, the student network learns attention information from teacher network.



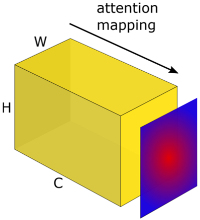
[6] Sergey Zagoruyko and Nikos Komodakis (2016). “Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer”. In: *arXiv preprint arXiv:1612.03928*.

Feature-Based Knowledge

Example

Considering a CNN layer and its corresponding activation tensor $A \in R^{C \times H \times W}$, which consists of C feature planes with spatial dimensions $H \times W$. An activation-based mapping function $\mathcal{F}$ (w.r.t. that layer) takes as input the above 3D tensor A and outputs a **spatial attention map**, i.e., a flattened 2D tensor defined over the spatial dimensions, or

$$\mathcal{F} : R^{C \times H \times W} \longrightarrow R^{H \times W}$$



Feature-Based Knowledge

Example

Specifically, in this work we consider the following activation-based spatial attention maps:

$$\mathcal{F}(A) = \sum_{i=1}^{C} |A_i|$$

Then we can define $\mathcal{I}$ as the indices of all teacher-student activation layer pairs for which we want to transfer attention maps. Also, we define $Q_S^j = \mathcal{F}(A_S^j)$ and $Q_T^j = \mathcal{F}(A_T^j)$ as the j-th ($j \in \mathcal{I}$) pair of student and teacher attention maps.

Feature-Based Knowledge

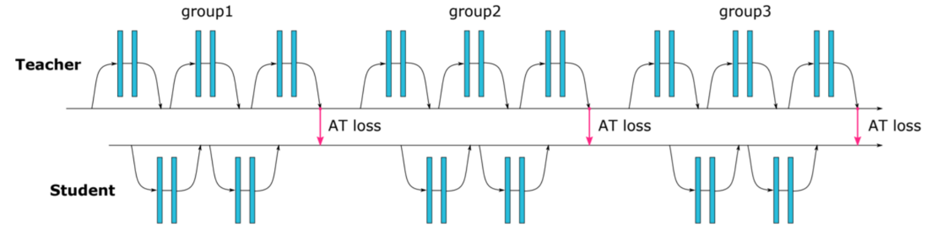
Example

The total distillation loss of[7] is then formulated as:

$$\mathcal{L}_{KD} = \mathcal{L}_{CE} + \mathcal{L}_{AT}$$

$$\mathcal{L}_{AT} = \left|\left|\frac{Q_S^j}{||Q_S^j||_2} - \frac{Q_T^j}{||Q_T^j||_2}\right|\right|_2$$

where $\mathcal{L}_{CE}$ is the cross entropy loss and the pipeline is shown below:

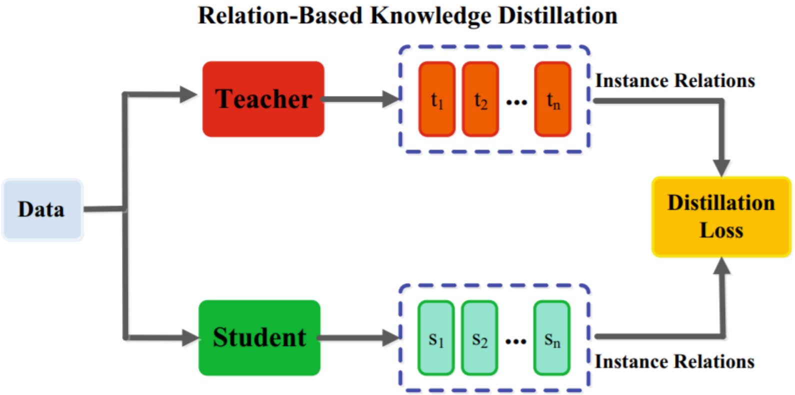


[7] Sergey Zagoruyko and Nikos Komodakis (2016). "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer". In: *arXiv preprint arXiv:1612.03928*.

Relation-Based Knowledge

Relation-Based Knowledge

Both response-based and feature-based knowledge use the outputs of specific layers in the teacher model. Relationbased knowledge further explores the relationships between different layers or data samples.



Relation-Based Knowledge

Formulation

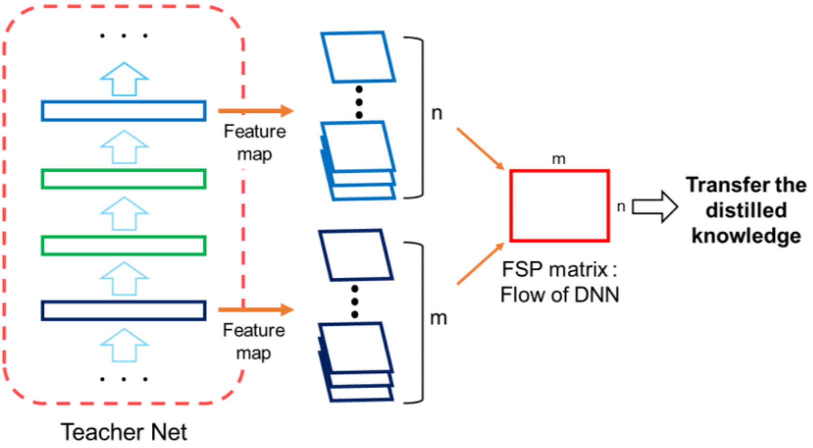
In general, the distillation loss of relation-based knowledge based on the relations of feature maps can be formulated as

$$L_{RelD}(f_t, f_s) = \mathcal{L}_R(\Phi_t(\hat{f}_t, \check{f}_t), \Phi_s(\hat{f}_s, \check{f}_s))$$

where f_t and f_t are the feature maps of teacher and student models, respectively. Pairs of feature maps are chosen from the teacher model, $\hat{f}_t$ and $\check{f}_t$, and from the student model, $\hat{f}_s$ and $\check{f}_s$. $\Phi_t(\cdot)$ and $\Phi_s(\cdot)$ are the similarity functions for pairs of feature maps from the teacher and student models. $\mathcal{L}_R(\cdot)$ indicates the correlation function between the teacher and student feature maps.

Relation-Based Knowledge

[8] proposed a typical example of relation-based knowledge distillation. Firstly, we learns the concept of FSP (flow of the solution procedure) matrix.



[8] Junho Yim et al. (2017). "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4133–4141.

Relation-Based Knowledge

Example

The FSP matrix $G \in R^{m \times n}$ is generated by the features from two layers. Let one of the selected layers generate the feature map $F^1 \in R^{h \times w \times m}$, where h,w,and m represent the height, width, and number of channels,respectively. The other selected layer generates the featuremap $F^2 \in R^{h \times w \times n}$. Then, the FSP matrix $G \in R^{m \times n}$ is calculated by

$$G_{i,j}(x; W) = \sum_{s=1}^{h} \sum_{t=1}^{w} \frac{F^1_{s,t,i}(x; W) \times F^2_{s,t,j}(x; W)}{h \times w}$$

where x and W represent the input image and the weights of the DNN, respectively.

Relation-Based Knowledge

Example

Suppose the FSP matrices of teacher network and student network are defined as $G^T(x; W_t)$ and $G^S(x; W_s)$, the knowledge distillation loss is then calculated as:

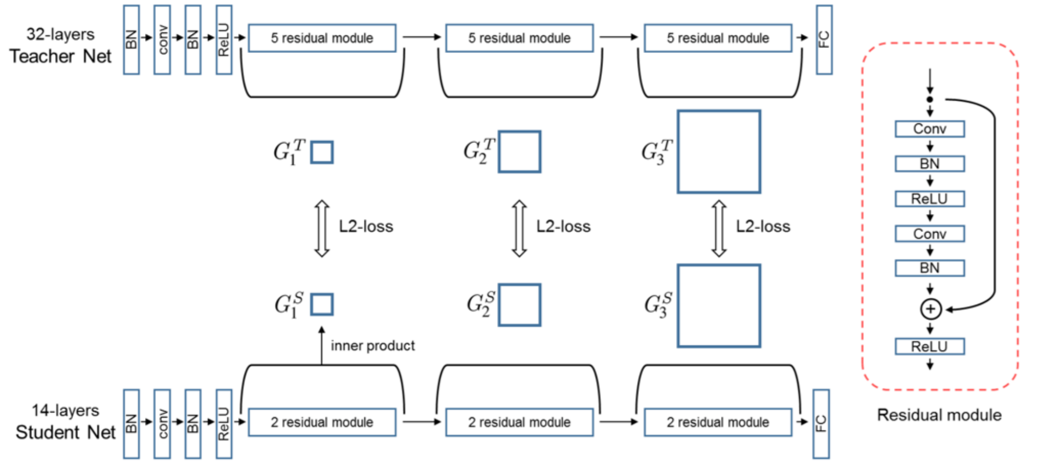
$$\mathcal{L}_{KD}(W_t, W_s) = \frac{1}{N} \sum_x \sum_{i=1}^{n} \lambda_i \times ||G_i^T(x; W_t) - G_i^S(x; W_s)||_2^2$$

where λ_i and N represent the weight for each loss term and the number of data points, respectively.

Relation-Based Knowledge

Example

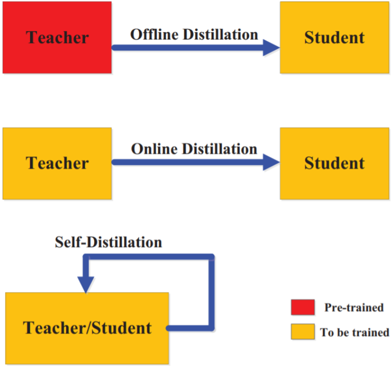
The network pipeline is shown below:



Distillation Method

Offline Distillation

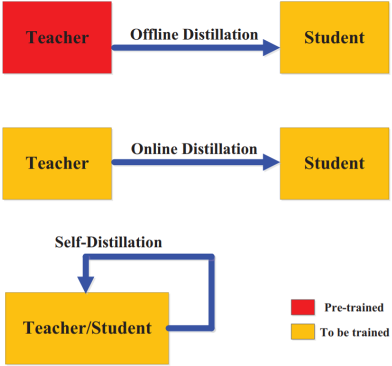
Most of previous knowledge distillation methods work offline. In offline knowledge distillation, the knowledge is transferred from a pre-trained teacher model into a student model.



Offline Distillation

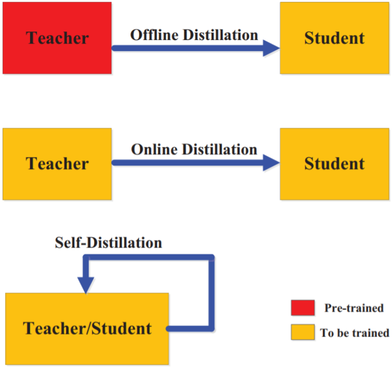
Therefore, the whole training process has two stages:

- The large teacher model is first trained on a set of training samples before distillation.
- The teacher model is used to extract the knowledge in the forms of logits or the intermediate features, which are then used to guide the training of the student model during distillation.



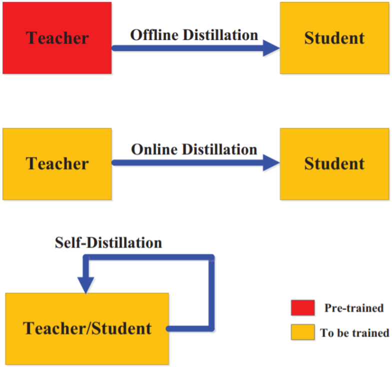
Online Distillation

In online distillation, both the teacher model and the student model are updated and the whole knowledge distillation framework is end-to-end trainable.



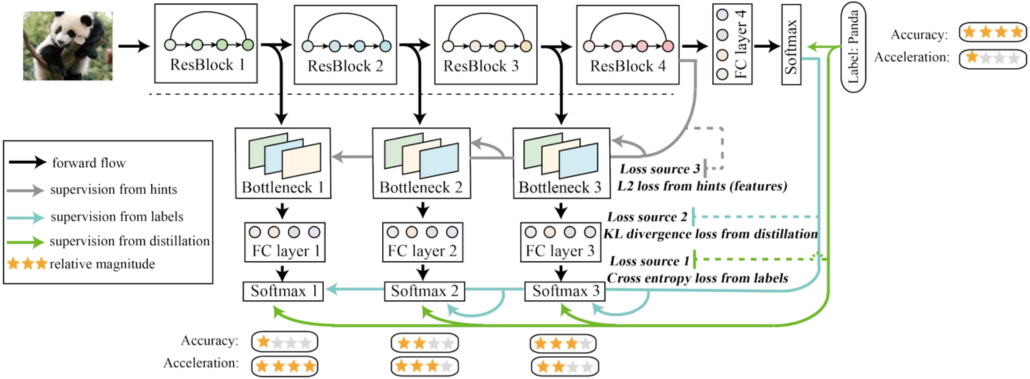
Self Distillation

In self-distillation, the same networks are used for the teacher and the student models. This can be regarded as a special case of online distillation.



Example

[9] proposed a new self-distillation method, in which knowledge from the deeper sections of the network is distilled into its shallow sections.



[9] Linfeng Zhang et al. (2019). "Be your own teacher: Improve the performance of convolutional neural networks via self distillation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3713–3722.

Example

The distillation loss of[10] is designed as:

- α: Hyper-parameter
- i: i_{th} sub-network
- C: Number of sub-network
- q^i: Logits of i_{th} sub-network
- F_i: Features of i_{th} sub-network
- $\mathcal{L}_{CE}$: Cross entropy loss
- $\mathcal{L}_{KL}$: KL divergence loss
- y: Ground truth

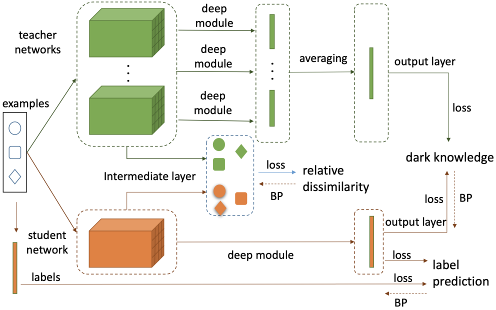
$$\mathcal{L}_R = \sum_{i}^{C}((1-\alpha)\cdot\mathcal{L}_{CE}(q^i, y) + \alpha\cdot\mathcal{L}_{KL}(q^i, q^C) + \lambda\cdot||F_i - F_C||_2^2)$$

[10] Linfeng Zhang et al. (2019). “Be your own teacher: Improve the performance of convolutional neural networks via self distillation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3713–3722.

KD Scenarios

Ensemble KD[11]

Combine KD and Ensemble learning. A graphical diagram for the proposed method to train a new thin deep student network by incorporating multiple comparable teacher networks. The method consists of three losses, including label prediction loss, dark knowledge loss and the relative similarity loss. The incorporation of multiple teacher networks exists in two places. One is in the output layers via averaging the softened output targets; the other lies in the intermediate layer by determining the best triplet ordering relationships.



[11] Shan You et al. (2017). "Learning from multiple teacher networks". In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1285–1294.

Ensemble KD[11]

Here, the label prediction loss is a simple softmax cross entropy loss. Relative similarity loss is triplet loss, dark knowledge loss is ensemble KD loss. Given m teacher networks $\mathcal{N}_{T_1}, \mathcal{N}_{T_2}, \cdot, \mathcal{N}_{T_m}$ and one student network $\mathcal{N}_S$, we can have.

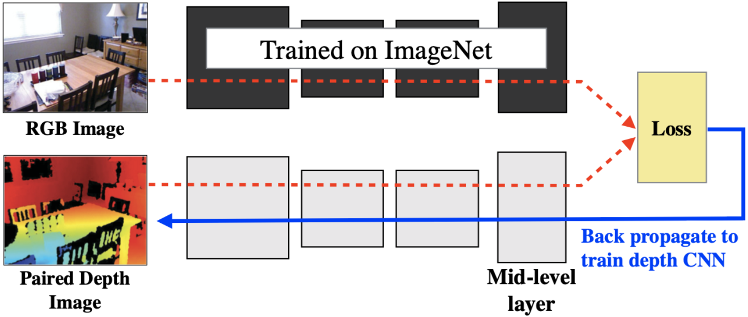
$$\mathcal{L}_{final} = \sum [\mathcal{H}(\boldsymbol{y}_i, \mathcal{N}(\boldsymbol{x}_i)) + \alpha \mathcal{H}(\frac{1}{m} \sum_{t=1}^{m} \mathcal{N}_{T_t}^{\tau}(\boldsymbol{x}_i), \mathcal{N}_S^{\tau}(\boldsymbol{x}_i))] + \beta \mathcal{L}_{RD}(w_s; \boldsymbol{x}_i, \boldsymbol{x}_i^+, \boldsymbol{x}_i^-), \quad (9)$$

where $\mathcal{H}$ means the entropy function, w_s indicates the parameters of feature extractor, $\boldsymbol{x}_i, \boldsymbol{x}_i^+, \boldsymbol{x}_i^-$ means the triplet pairs.

[11] Shan You et al. (2017). "Learning from multiple teacher networks". In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1285–1294.

Cross-Modal KD[12]

Combine KD and Cross Modal Learning. Architecture: We train a CNN model for a new image modality (like depth images), by teaching the network to reproduce the mid-level semantic repre- sentations learned from a well labeled image modality (such as RGB images) for modalities for which there are paired images.



[12] Saurabh Gupta, Judy Hoffman, and Jitendra Malik (2016). "Cross modal distillation for supervision transfer". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2827–2836.

Cross-Modal KD[12]

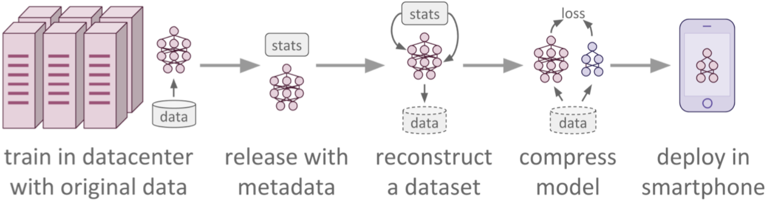
The proposed scheme for learning rich representations for images of modality $\mathcal{M}_d$. t indicates the functions that maps the $\left(\psi^{L}_{\mathcal{M}_d}(I_d)\right)$ to the same dimension with $\phi^{i^*}_{\mathcal{M}_s,D_s}(I_s)$. for some chosen and fixed layer $i^* \in [1 \ldots K]$, we measure the similarity between the representations using an appropriate loss function f (for example, euclidean loss).

$$\mathcal{L}_{final} = \sum_{(I_s,I_d)\in U_{s,d}} f\left(t\left(\psi^{L}_{\mathcal{M}_d}(I_d)\right),\ \phi^{i^*}_{\mathcal{M}_s,D_s}(I_s)\right) \tag{10}$$

[12] Saurabh Gupta, Judy Hoffman, and Jitendra Malik (2016). "Cross modal distillation for supervision transfer". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2827–2836.

Data-free KD[13]

Combine KD and Data Free Compression. The proposed model compression pipeline: a model is trained in a datacenter and released along with some metadata. Then, another entity uses that metadata to reconstruct a dataset, which is then used to compress the model with Knowledge Distillation. Finally, the model is deployed in a smartphone.



[13] Raphael Gontijo Lopes, Stefano Fenu, and Thad Starner (2017). "Data-free knowledge distillation for deep neural networks". In: *arXiv preprint arXiv:1710.07535*.