Zero-Shot Knowledge Distillation in Deep Networks

Gaurav Kumar Nayak [*1] Konda Reddy Mopuri [*2] Vaisakh Shaj [*3] R. Venkatesh Babu [1]
Anirban Chakraborty [1]

ICML 2019
Cite: 77

2021.06.16
임진혁

0 MIIL Index

임진혁<eeplearning@units.ac.kr>

Limitation of KD(Knowledge Distillation)

KD is very usefully used in many fields
There is a limitation that there is no original data.

In such cases only trained model is available without training data
- Medical diagnosis, that patients' privacy prohibits distribution
- Commercial products with deployed models
- Cost from annotating data
- Proprietary data (JFT-300M, SFC.)

임진혁<eeplearning@units.ac.kr>

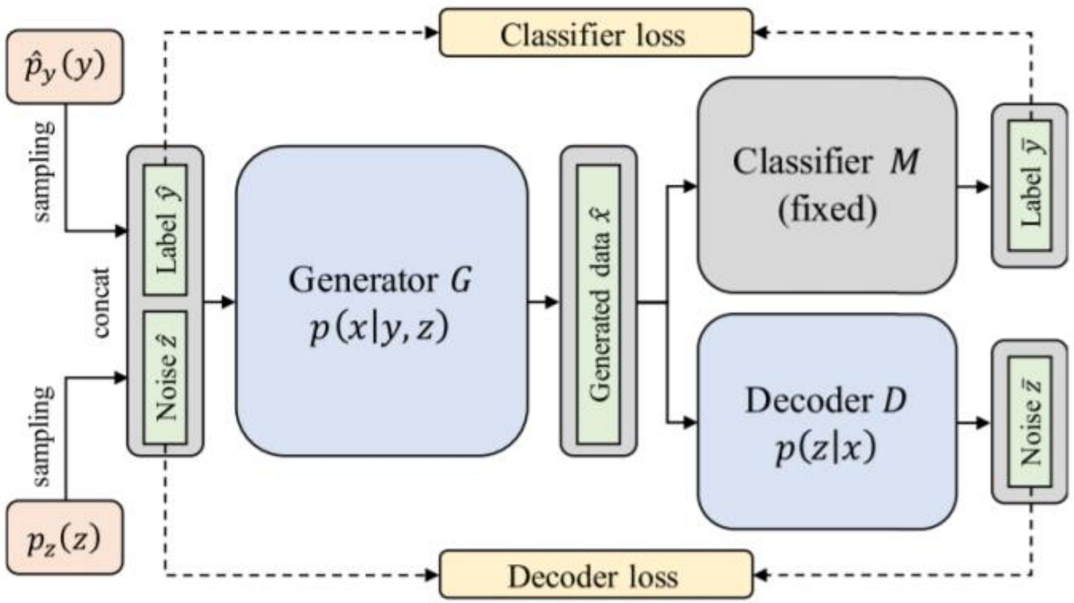
Data-free Knowledge Distillation (Zero Shot)

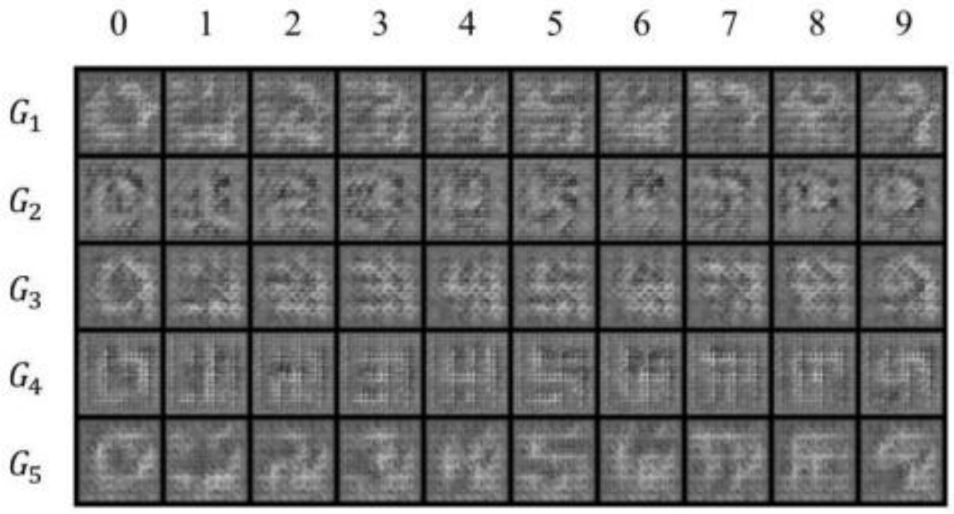
with no data samples and no extracted prior information

So there should be “Transfer Set” instead of “original data”

임진혁<eeplearning@units.ac.kr>

Knowledge Extraction with No Observable Data(Nips 2019)



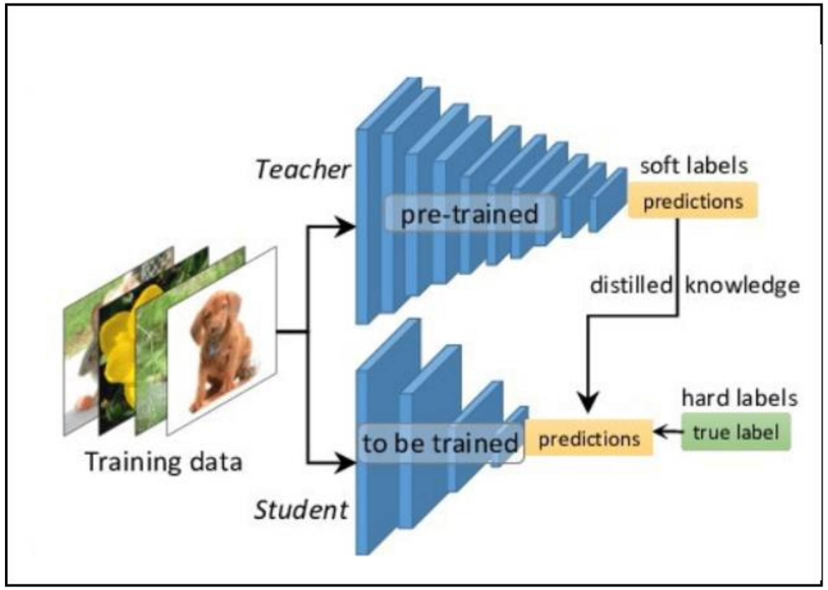


(a) MNIST ($z = \mathbf{0}$).

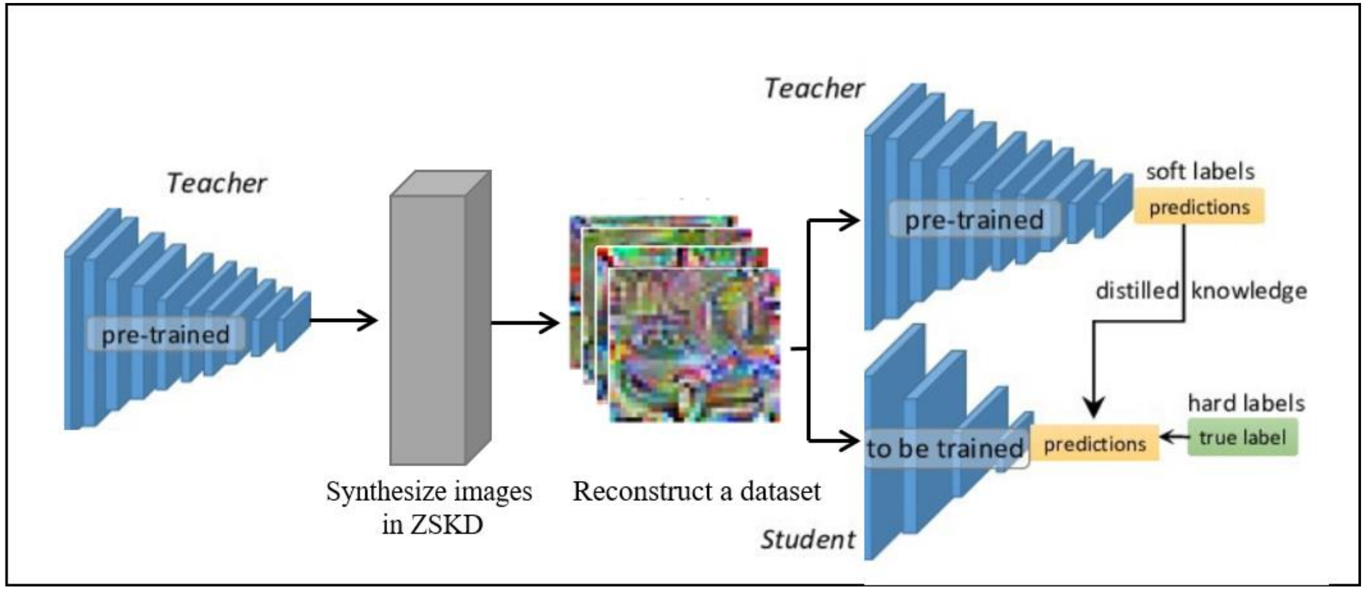
$$\mathcal{D} = \left\{\operatorname{argmax}_{\hat{x}}\, p(\hat{x}|\hat{y},\hat{z}) \;\middle|\; \hat{y} \sim \hat{p}_y(y) \text{ and } \hat{z} \sim p_z(z)\right\}$$

Zero-Shot Knowledge Distillation (ICML 2019)

Make Train Data Distribution from Teacher's parameter



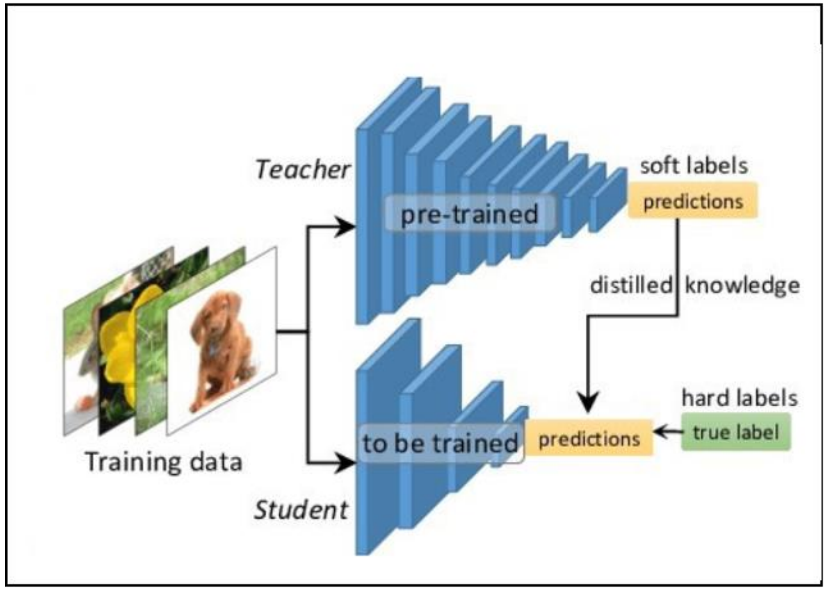
Conventional knowledge distillation



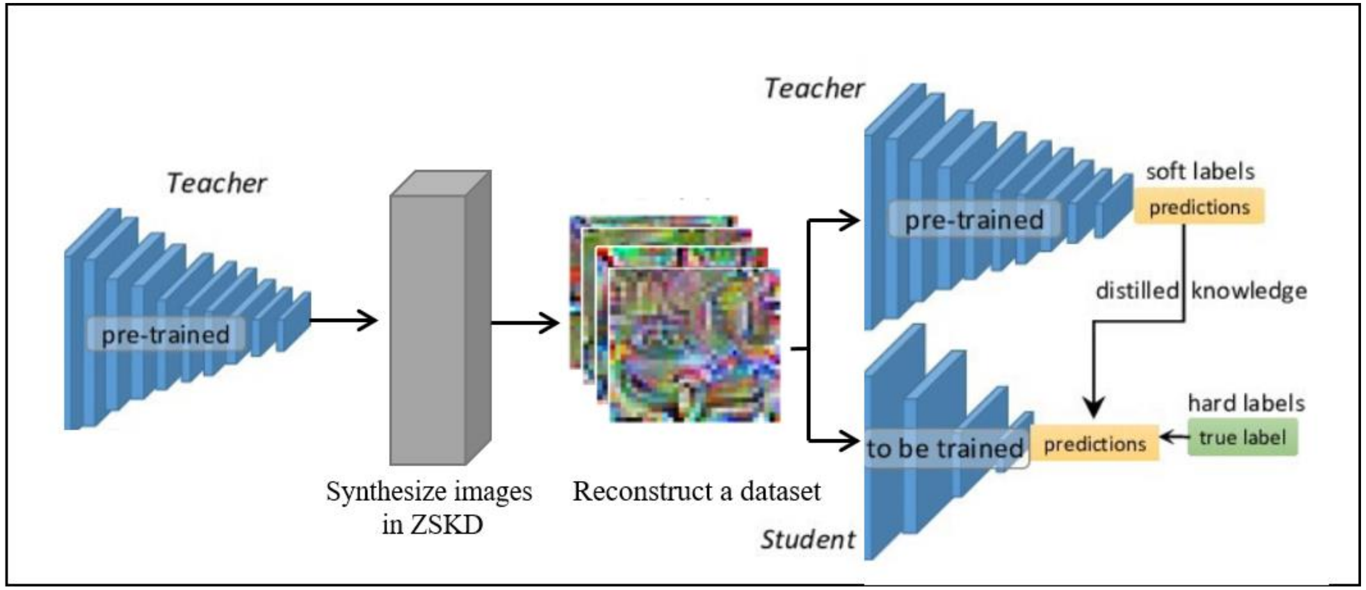
Data-free knowledge distillation
(Zero-shot knowledge distillation)

Zero-Shot Knowledge Distillation (ICML 2019)

Make Train Data Distribution from Teacher's parameter



Conventional knowledge distillation



Data-free knowledge distillation
(Zero-shot knowledge distillation)

Zero-Shot Knowledge Distillation (ICML 2019)

Using Class Similarity Matrix

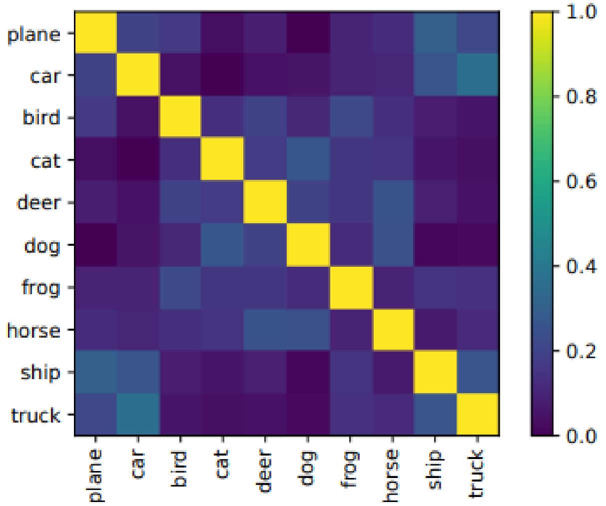
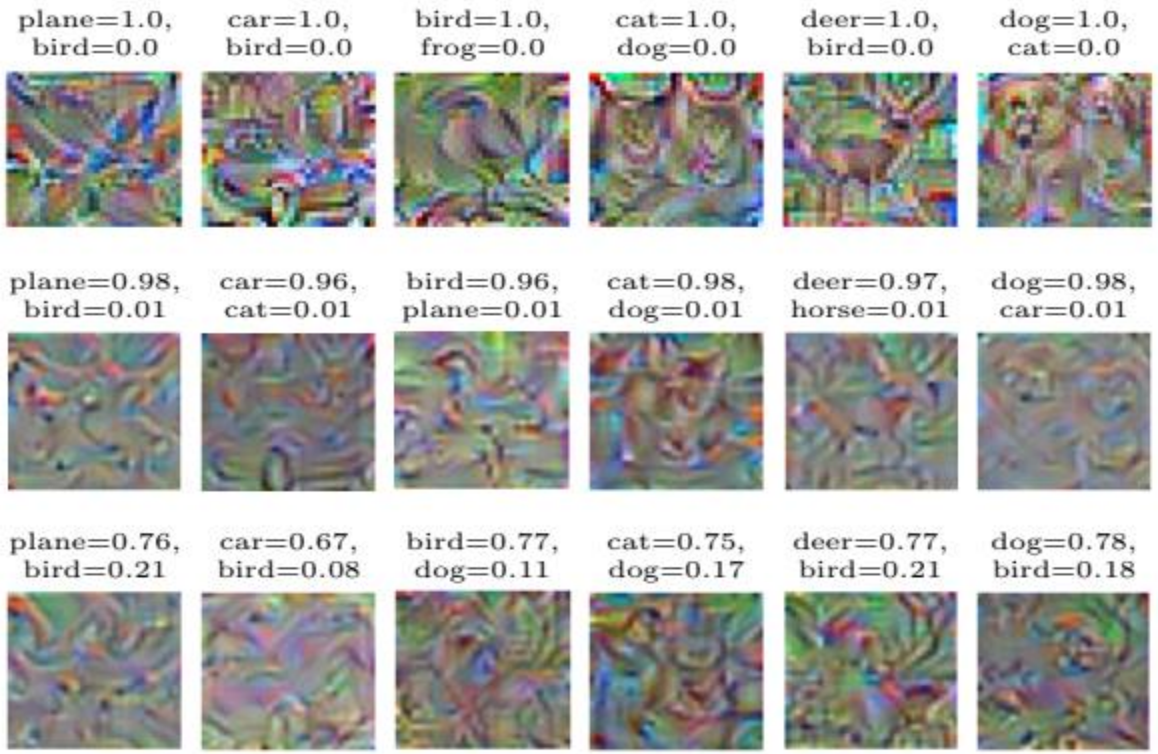
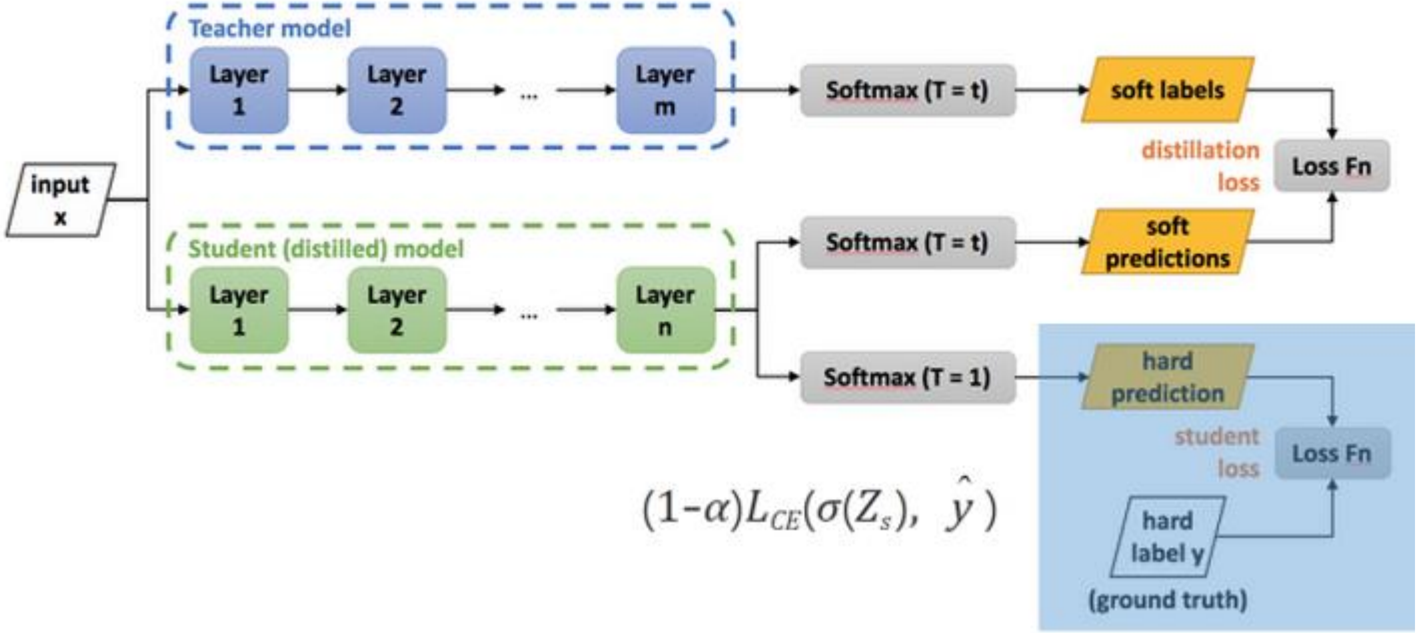


Figure 1. Class similarity matrix computed for the *Teacher* model trained over CIFAR-10 dataset. Note that the class labels are mentioned and the learned similarities are meaningful.



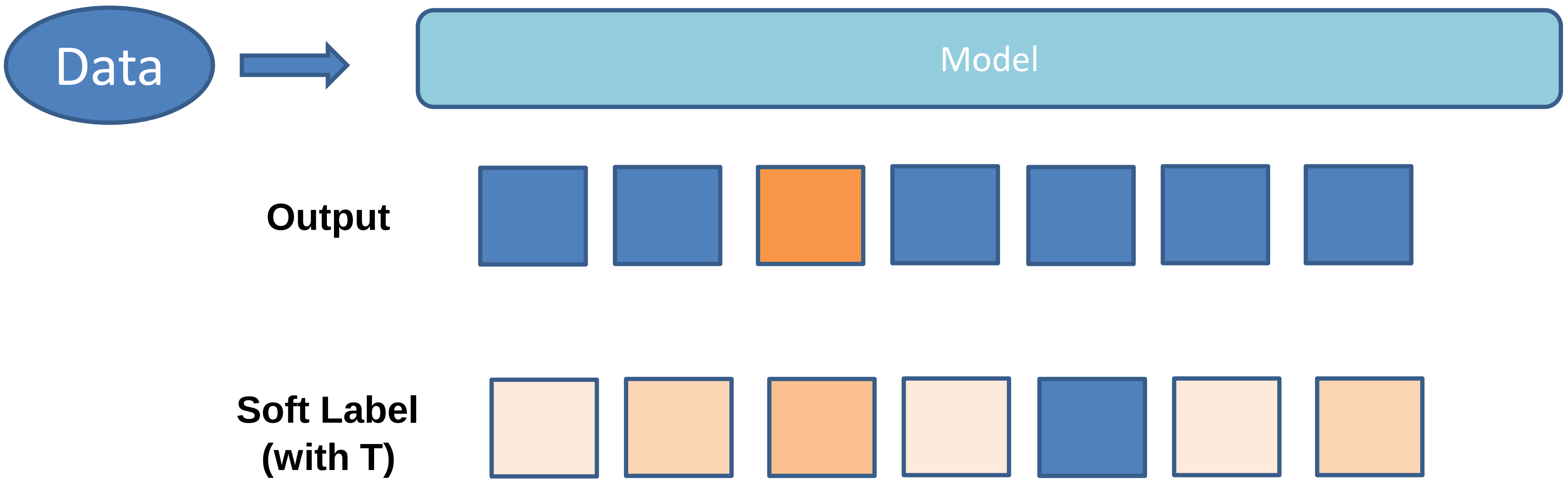
임진혁<eeplearning@units.ac.kr>

Knowledge Distillation (Baseline)

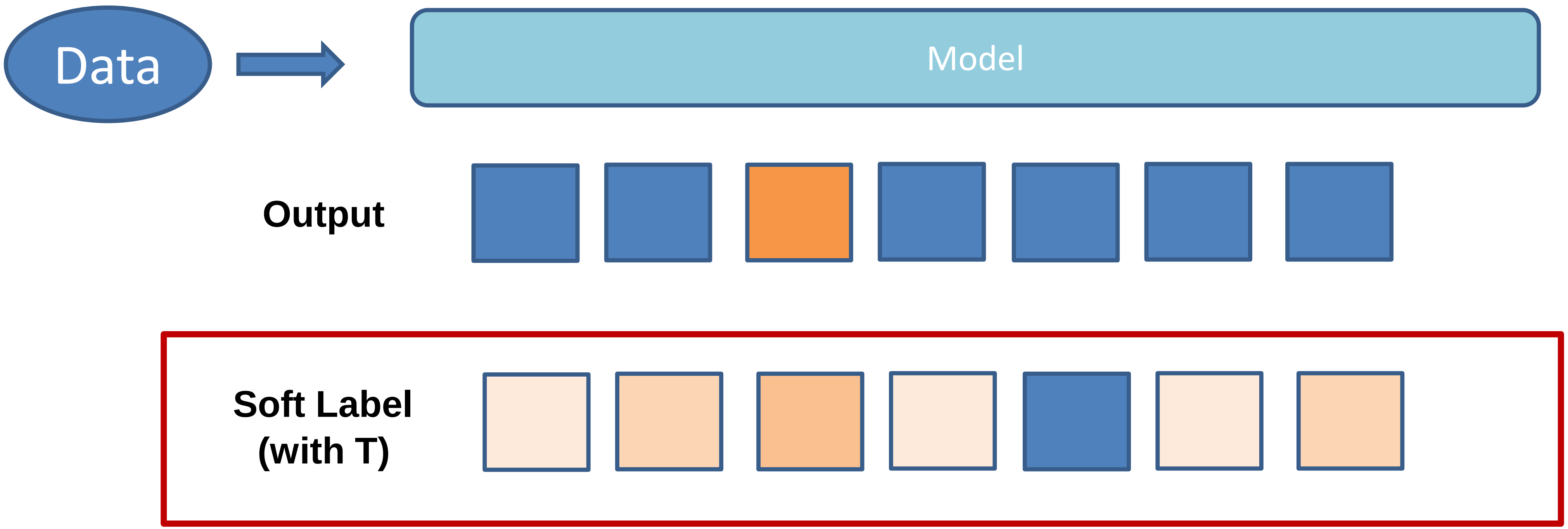


$$L = \sum_{(x,y)\in\mathbb{D}} L_{KD}(S(x,\theta_S,\tau),T(x,\theta_T,\tau))+\lambda L_{CE}(\hat{y}_S,y)$$

Knowledge Distillation (Baseline)

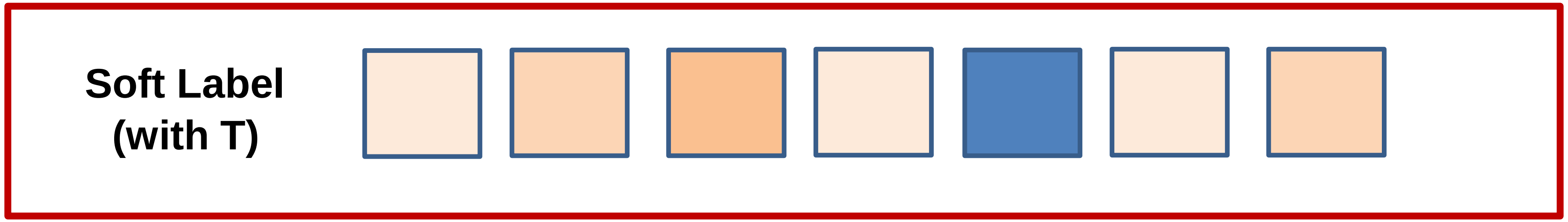


Knowledge Distillation (Baseline)



임진혁<eeplearning@units.ac.kr>

ZSKD

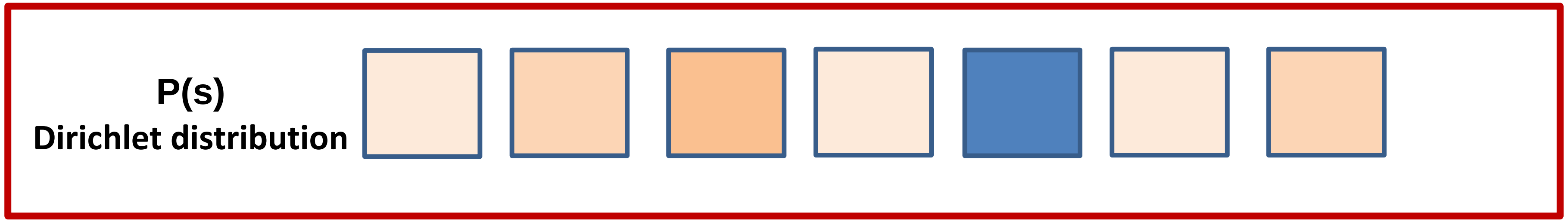


model output space of the *Teacher* model.

- **Model**(동사) **output space of the Teacher model**
- **s~p(s) : Random Vector** that represents the **softmax outputs** of the *Teacher using* **Dirichlet distribution**

임진혁<eeplearning@units.ac.kr>

ZSKD



X 없이 y_pred를 만든다고 생각하면 된다. (이 경우에는 softmax output)
y_pred는**Dirichlet distribution**에서 Sampling한 값이 될 것이다.

$$\bar{x}_i{}^k = \underset{x}{\operatorname{argmin}}\, L_{CE}(\boldsymbol{y}_i^k, T(x, \theta_T, \tau))$$

임진혁<eeplearning@units.ac.kr>

Dirichlet Distribution(디리클레 분포)

model output space of the *Teacher* model.

- 분포를 추정하는 방법은 크게 2가지 (Parametric , Nonparametric)
- Paramaetric
  - 모수를 가정(평균,분산)
  - 데이터가 적어도 모수 분포를 잘 가정하면 좋은 추정이 된다
  - 예시: 가우시안 분포 /이항 분포/ 베타 분포/ 다변량 분포/ 디리클레 분포 (다변량 , 연속형)
- Nonparametric
  - 모수를 가정하지 않음
  - 데이터가 많을 수록 좋은 추정이 가능

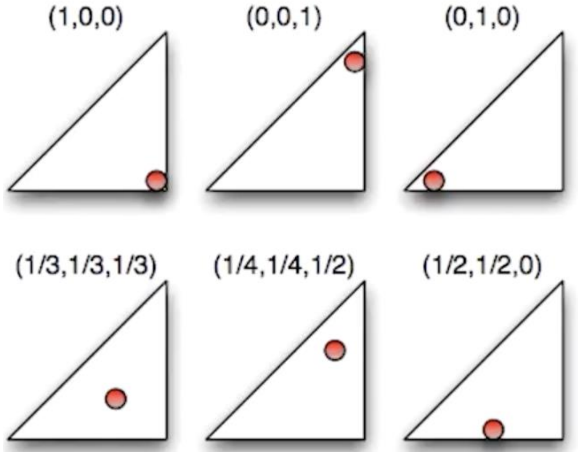
임진혁<eeplearning@units.ac.kr>

Dirichlet Distribution(디리클레 분포)

model output space of the *Teacher* model.

- αk에 대하여 k개의 연속형 확률변수에 대응되는 k개의 continous values을 사용하여 분포 표현
- 확률 정의에 따라 해당 continuous random variables은 0보다는 크고 합하면 1이 된다.
- LDA에 쓰이는 바로 그 분포
- k=3일 때, 2차원으로 시각화

  값이 1보다 클수록
  다양한 차원으로 퍼진다.



Dirichlet Distribution(디리클레 분포)

Dirichlet distribtutions

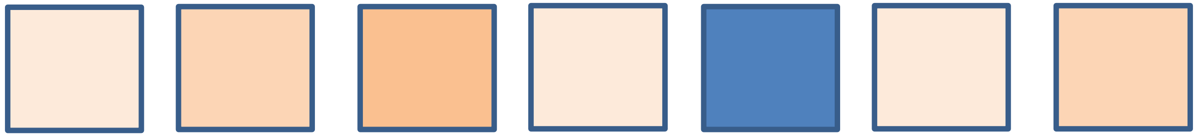
$$f(x_1, \cdots, x_k; \alpha_1, \cdots, \alpha_k) = \frac{1}{\mathrm{B}(\alpha)} \prod_{i=1}^{k} x_i^{\alpha_i - 1}$$

$$\mathrm{B}(\alpha) = \frac{\prod_{i=1}^{k} \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{k} \alpha_i\right)} \ (\Gamma \text{is gamma function})$$

영제야 이게
베타 분포의 관점에서 볼 때, 다항분포를 Control 하는 효과를 가진
분포라는데 무슨 뜻인지 모르겠어 **시바**

ZSKD

P(s)
Dirichlet distribution



Softmax output s^k of class k

$Dir(K, \alpha^k)$,

$k \in 1 \cdots K$ **is the class index,**

K **is the dimension of the output probability vector**

α^k **is the concentration parameter of the distribution modelling class** k

$\boldsymbol{\alpha}^k = [\alpha_1^k, \alpha_2^k, \ldots, \alpha_K^k]$

It should reflect the similarities across softmax vector

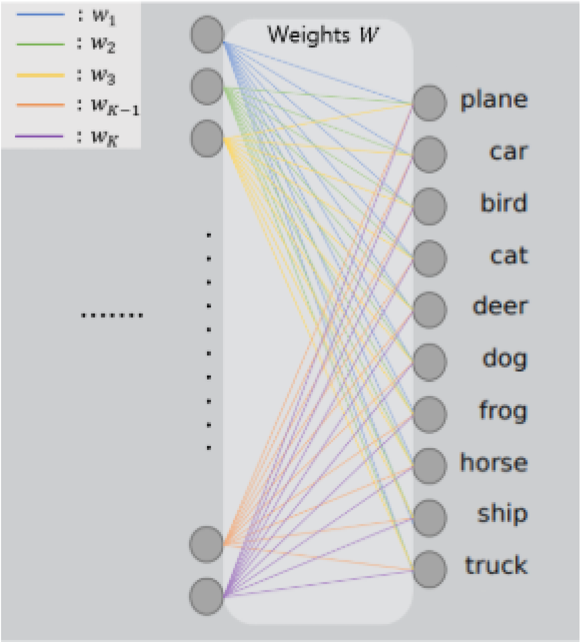
임진혁<eeplearning@units.ac.kr>

ZSKD- Class similarity Matrix

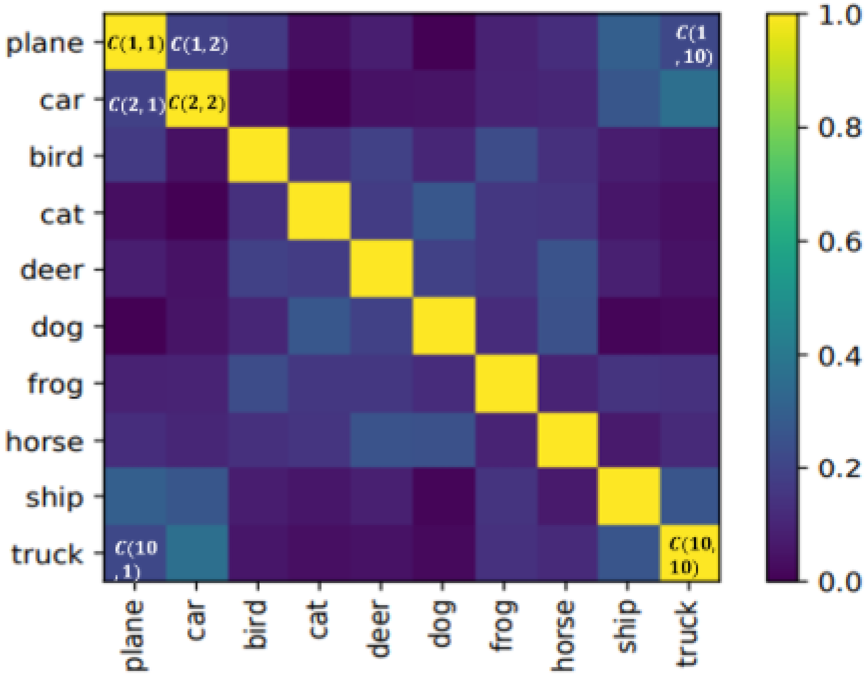
Matrix is consist of the weights connecting the final (softmax) W

C(i,j) denotes
the visual similarity between
the categories **i** and **j** in [0,1].

$$C(i,j) = \frac{\boldsymbol{w}_i^T \boldsymbol{w}_j}{\|\boldsymbol{w}_i\| \|\boldsymbol{w}_j\|}$$



(a) Representing weights
in the final and pre-final layers



(b) Class similarity matrix computed for the
Teacher model trained over CIFAR-10 dataset

ZSKD Process

$$\bar{x}_i{}^k = \underset{x}{\text{argmin}}\, L_{CE}(\boldsymbol{y}_i^k, T(x, \theta_T, \tau))$$

N softmax vectors corresponding to class k sampled from **Dir(K, αk) distribution.**

Knoledge Distill with S

$$\theta_S = \underset{\theta_S}{\text{argmin}} \sum_{\bar{x} \in \bar{X}} L_{KD}(T(\bar{x}, \theta_T, \tau), S(\bar{x}, \theta_S, \tau))$$

Algorithm 1 Zero-Shot Knowledge Distillation

Input : *Teacher* model T
N: number of DIs crafted per category,
$[\beta_1, \beta_2, ..., \beta_B]$: B scaling factors,
τ: Temperature for distillation
Output : Learned *Student* model $S(\theta_S)$,
$\bar{X}$: *Data Impressions*

```
Obtain K: number of categories from T
Compute the class similarity matrix
  C = [c_1^T, c_2^T, ..., c_K^T] as in eq. (2)
X̄ ← ∅
for k=1:K do
    Set the concentration parameter α^k = c_k
    for b=1:B do
        for n=1:⌊N/B⌋ do
            Sample y_n^k ~ Dir(K, β_b × α^k)
            Initialize x̄_n^k to random noise and craft x̄_n^k =
              argmin_x L_CE(y_n^k, T(x, θ_T, τ))
            X̄ ← X̄ ∪ x̄_n^k
        end
    end
end
Transfer the Teacher's knowledge to Student using the DIs
  via θ_S = argmin_{θ_S} Σ_{x̄∈X̄} L_KD(T(x̄, θ_T, τ), S(x̄, θ_S, τ))
```


임진혁<eeplearning@units.ac.kr>

ZSKD Process

- Step 1 : Train the Teacher network with cifar 10
- Step 2 : Extract final layer weights from the Pretrained Teacher Network
- Step 3 : Compute and save the Class Similarity for scales of 1.0 and 0.1
- Step 4 : Generate the Data Impressions (DI's)
- Step 5 : Train the Student network with generated DI's

Algorithm 1 Zero-Shot Knowledge Distillation

Input : *Teacher* model T
N: number of DIs crafted per category,
$[\beta_1, \beta_2, ..., \beta_B]$: B scaling factors,
τ: Temperature for distillation

Output : Learned *Student* model $S(\theta_S)$,
$\bar{X}$: *Data Impressions*

Obtain K: number of categories from T
Compute the class similarity matrix
$C = [\mathbf{c}_1^T, \mathbf{c}_2^T, \ldots, \mathbf{c}_K^T]$ as in eq. (2)
$\bar{X} \leftarrow \emptyset$
for *k=1:K* **do**
  Set the concentration parameter $\boldsymbol{\alpha}^k = \mathbf{c}_k$
  for *b=1:B* **do**
    for *n=1:*$\lfloor N/B \rfloor$ **do**
      Sample $\boldsymbol{y}_n^k \sim Dir(K, \beta_b \times \boldsymbol{\alpha}^k)$
      Initialize $\bar{x}_n^k$ to random noise and craft $\bar{x}_n^k = \underset{x}{\operatorname{argmin}}\, L_{CE}(\boldsymbol{y}_n^k, T(x, \theta_T, \tau))$
      $\bar{X} \leftarrow \bar{X} \cup \bar{x}_n^k$
    end
  end
end
Transfer the *Teacher*'s knowledge to *Student* using the *DIs* via $\theta_S = \underset{\theta_S}{\operatorname{argmin}} \sum_{\bar{x} \in \bar{X}} L_{KD}(T(\bar{x}, \theta_T, \tau), S(\bar{x}, \theta_S, \tau))$

임진혁<eeplearning@units.ac.kr>

3 Experiments

Experiement.pdf

감사합니다
죄송합니다.

임진혁<eeplearning@unist.ac.kr>