#hutom_ai

Knowledge Distillation

2020-01-06 | JiHyun Lee

hutom

Content

1. Introduction of Knowledge Distillation
    1) Distilling Ensemble; Single Model

1. Distilling the Knowledge in a Neural Network (2014)
    1) Knowledge
    2) Distillation

1. Relational Knowledge Distillation (2019)

Introduction of Knowledge Distillation

- Ensemble
  - To improve generalization performance Ensemble could be used.
  - Pros
    - Good performance
  - Cons
    - 저장 공간이 많이 듦.
    - 계산 시간이 많이 걸림.
    - 병렬 처리를 한다고 하더라도, 앙상블 개수가 많을수록 시간이 많이 걸림.

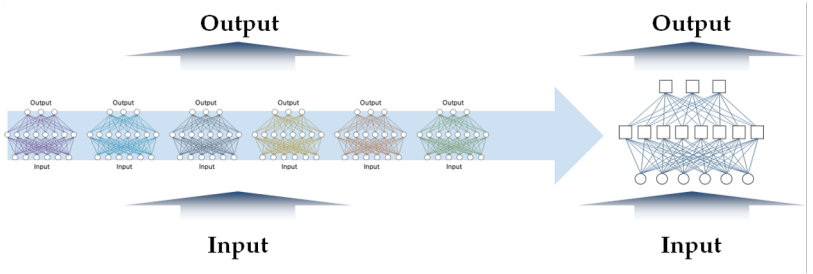
- Distilling Ensemble; we want to get a single and shallow model
  - Good performance (앙상블만큼 좋은 성능을 보이며)
  - Low computation (계산 시간, 저장 공간을 적게 차지하는)
  - single shallow model
    - 이를 위해서

      1) 앙상블 모델을 만든 후,

      2) 앙상블의 정보를 single shallow model에 이전시킴.

      3) 3가지 방법

      - class
      - logit
      - distilling knowledge



- **Class**
  - **If we use many observations, Generalization property will be good.**

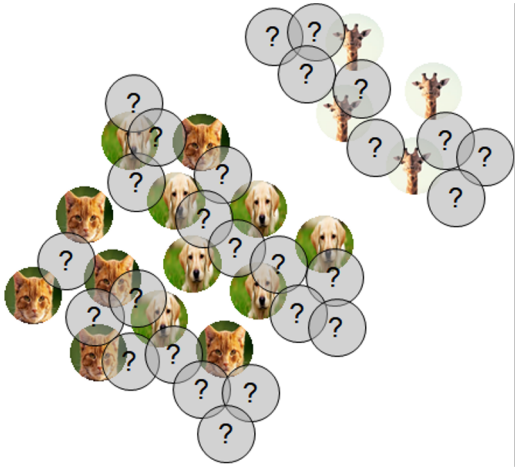
  - 앙상블의 경우, training data가 많지 않더라도 Generalization의 성능이 높도록 학습.
  - single model의 경우 training data가 많지 않다면 overfitting이 되는 경향이 있음.
  - **기존에 학습된 앙상블 모델을 토대로 예측을 해서 oversampling된 데이터에 class (label)가 붙음.**
  - **해당 데이터로 single shallow net을 학습시킴.**
    즉, 앙상블의 정보가 이전되어 (distilled knowledge), 일반화 성능이 높은 single shallow net이 생성

weight	color	length	...	Y
10	red	80		
30	yellow	201		
15	white	100		
6	gray	50		
5	gray	40		

over sampling

weight	color	length	...	Y
10	red	80		
30	yellow	201		
15	white	100		
6	gray	50		
5	gray	40		
20	gray	80		?
32	yellow	205		?
10	white	102		?
8	gray	52		?
9	white	42		?
12	gray	45		?

over sampling



Buciluǎ, C., Caruana, R., & Niculescu-Mizil, A. (2006, August). In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 535-541). ACM.
https://dl.acm.org/doi/10.1145/1150402.1150464

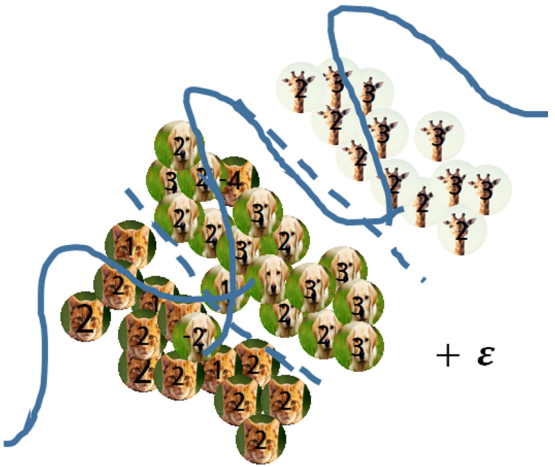
Introduction of Knowledge Distillation; Single Model 2

- **logit**
  - **we can use 'logit'** instead of class
  - logit could be understood as score of class (class의 확률분포)
  - 앙상블의 logit 값을 knowledge로 사용.
  - 이렇게 하면, 단순히 class 정보를 준 것보다, 더 많은 class 정보 (데이터 분포 정도) 들어가 있음.

- **logit + noise**
  - Also, can add 'noise' to 'logit', which acts as regularizer.
  - noise가 regularizer의 역할을 해서 성능이 더 좋아진다고 함.

weight	color	length	...	Y
10	red	80		2
30	yellow	201		1
15	white	100		3
6	gray	50		3
5	gray	40		2
20	gray	80		-2
32	yellow	205		3
10	white	102		1
8	gray	52		3
9	white	42		-2
12	gray	45		1

weight	color	length	...	Y
10	red	80		ε +2
30	yellow	201		ε +1
15	white	100		ε +3
6	gray	50		ε +3
5	gray	40		ε +2
20	gray	80		ε +-2
32	yellow	205		ε +3
10	white	102		ε +1
8	gray	52		ε +3
9	white	42		ε +-2
12	gray	45		ε +1



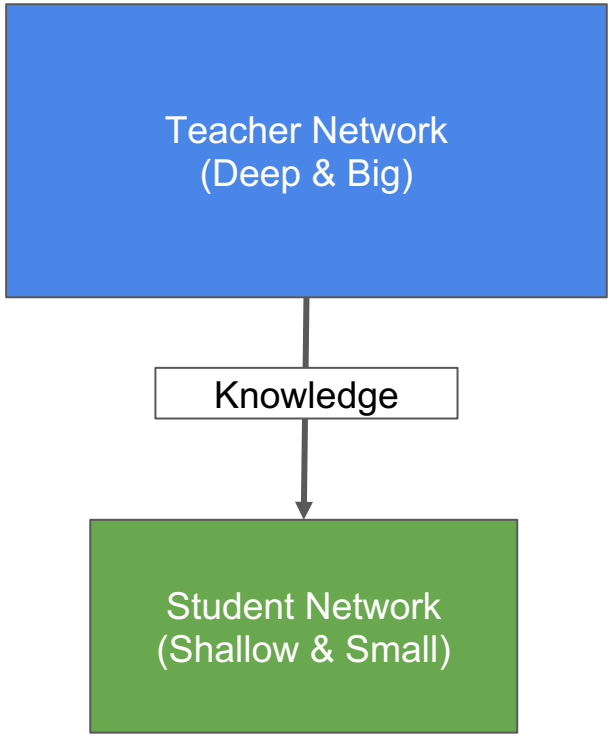
Ba, J., & Caruana, R. (2014). In Advances in neural information processing systems (pp. 2654-2662).
https://papers.nips.cc/paper/2014/hash/ea8fcd92d59581717e06eb187f10666d-Abstract.html
Sau, B. B., & Balasubramanian, V. N. (2016). arXiv preprint arXiv:1610.09650.
https://arxiv.org/abs/1610.09650

- **Distilling Ensemble**
  - Hinton used softmax function to get prob. distribution.
  - 가지고 있는 데이터에서 앙상블 모델을 학습한 후,
  - softmax function을 이용해서, 각 관측치에 class가 해당한 확률을 구함.
  - **softmax function을 통해서 구한 확률값을 가지고, single shallow net 학습하면, 앙상블의 정보가 전이됨.**

Distilling the Knowledge in a Neural Network

hutom

Distilling the Knowledge in a Neural Network (2015)



Teacher Network (T)

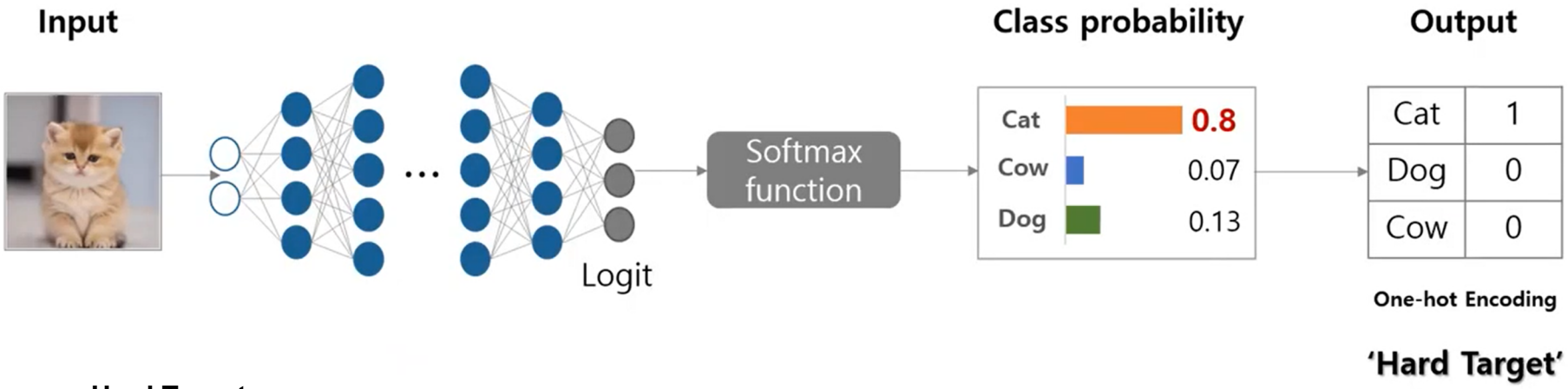
- **cumbersome model**
  e.g. ensemble / a large generalized model
- (pros) excellent performance
- (cons) computationally expansive
- can not be deployed when limited environments
- e.g. 정확도: 95%, 추론 시간: 2시간

잘 학습된 Teacher 모델의 지식을 전달하여
단순한 student 모델로 비슷한 좋은 성능을 내고자 함.

Student Network (S)

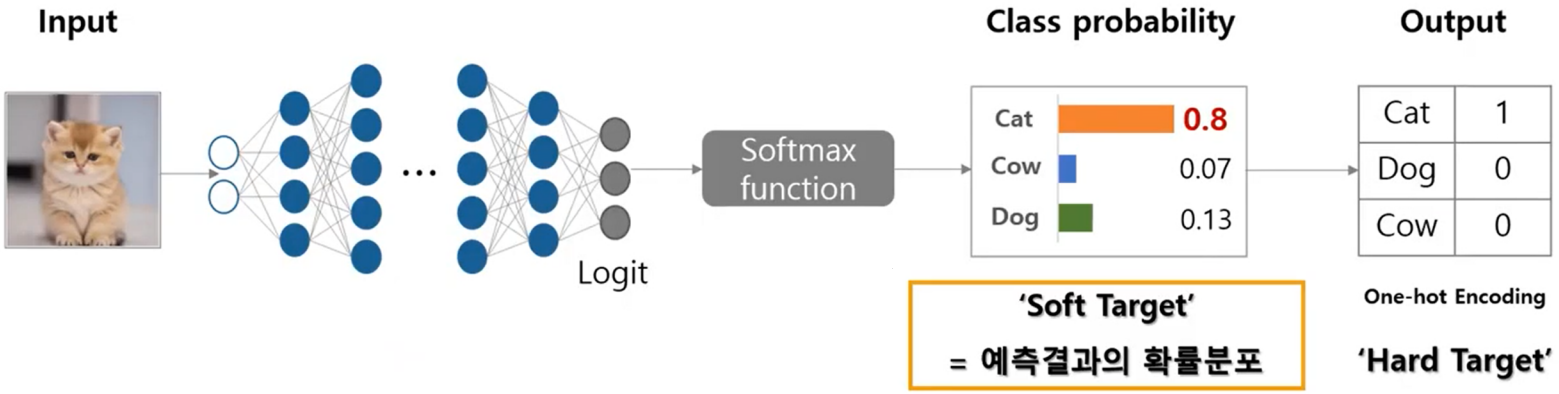
- **small model**
- (pros) fast inference
- (cons) lower performance than T
- suitable for deployment
- e.g. 정확도: 90%, 추론 시간: 5분

일반적인 분류 모델



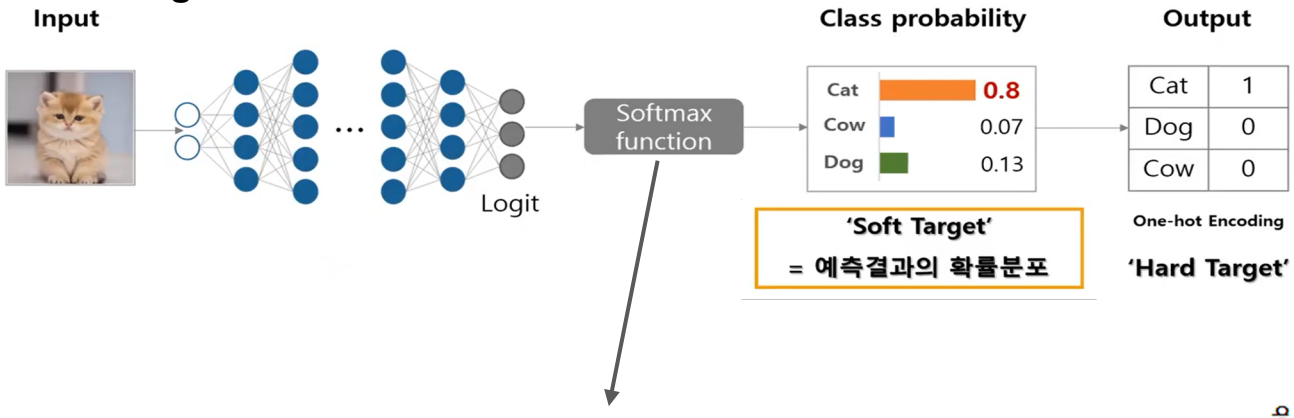
- **Hard Target**
  - One-hot encoding 방식을 사용하여 1, 0으로만 구성된 예측값.
  - *(문제점)* **가장 높은 확률값을 가지는 class를 제외하고는, 다른 class의 확률값은 동등하게 0 값으로 무시됨.**
    e.g. Input 이미지가 Cow (0.07) 보다는 Dog (0.13)에 가까운 형태. 이를 모델이 예측결과의 확률분포로 잘 반영하고 있음.

Knowledge Distillation

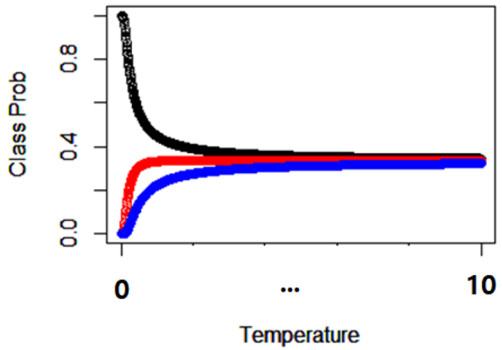


- **Soft Target**
  - **One-hot encoding 방식을 거치지 않은 Soft target이 예측 결과의 확률 분포를 가지고 있음.**
  - 잘 학습된 모델의 지식을 함축하고 있음. => **Teacher 모델의 Knowledge로 soft target 사용.**
  - *(문제점)* 가장 큰 logit 값을 갖는 node의 출력값은 1에 가깝고, 나머지는 0에 가깝게 mapping 됨.
  - 이를 개선하기 위해서, **temperature**라는 하이퍼 파라미터를 softmax function에 추가.

Knowledge Distillation

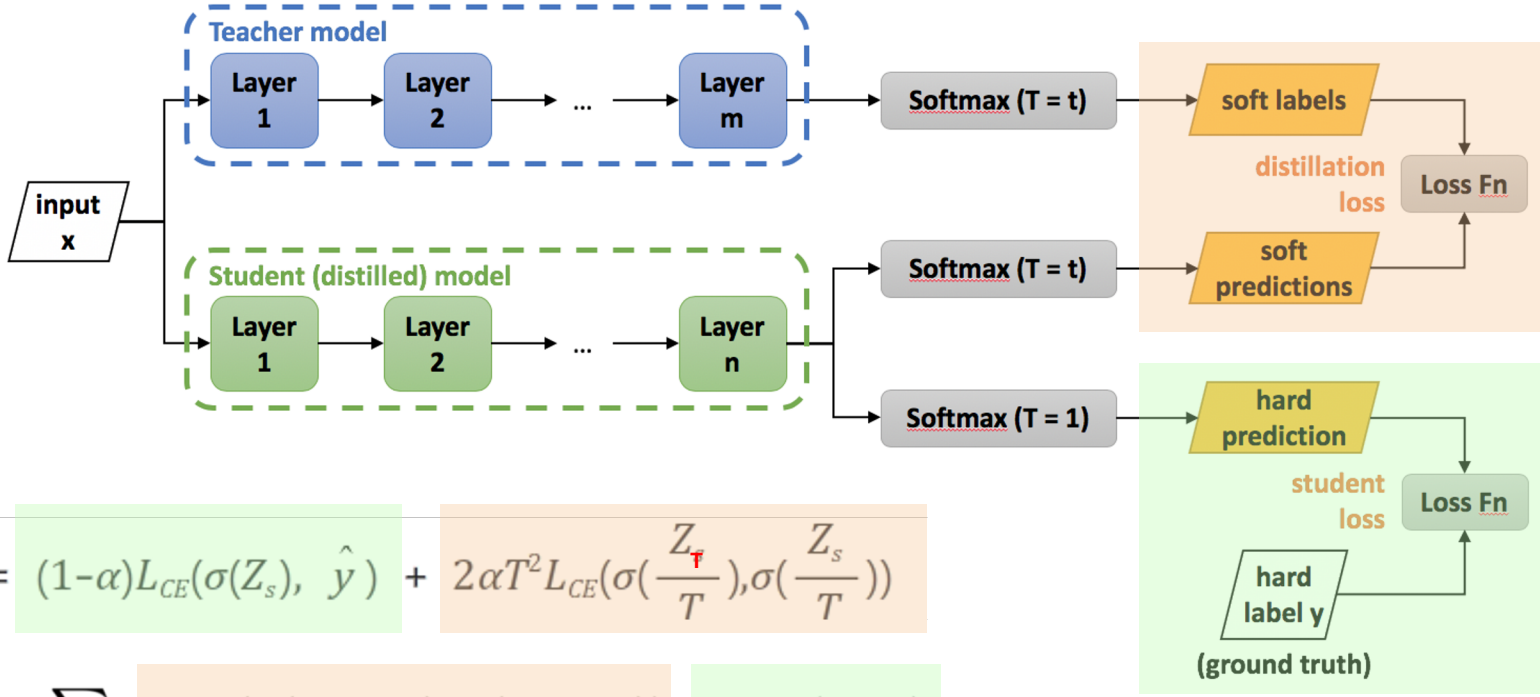


$$Softmax(z_i) = \frac{\exp(z_i)}{\sum_j \exp(z_j)} \quad \Rightarrow \quad Softmax(z_i) = \frac{\exp(z_i/\tau)}{\sum_j \exp(z_j/\tau)}$$



- **Temperature**
  - Scaling 역할.
  - **일부 class에 대한 확률값은 거의 0에 가까워서 학습 시 정보가 거의 전달되지 않을 수 있으므로, temperature를 통해서 확률값을 soft 하게 만들어, 학습에 잘 반영될 수 있도록 함.**
  - Temperature가 1일 때는 기존 softmax function과 동일. **Temperature가 작을수록 soft한 확률 분포.**

Knowledge Distillation Framework

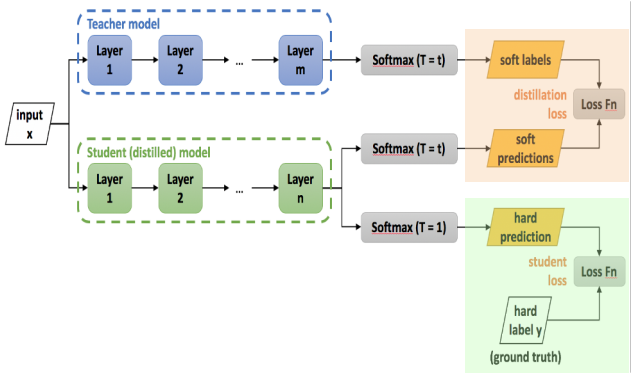


$$Total\ \ Loss = (1-\alpha)L_{CE}(\sigma(Z_s),\ \hat{y}) + 2\alpha T^2 L_{CE}(\sigma(\frac{Z_T}{T}),\sigma(\frac{Z_s}{T}))$$

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

Knowledge Distillation Framework

$$Total\ \ Loss = (1-\alpha)L_{CE}(\sigma(Z_s),\ \hat{y}) + 2\alpha T^2 L_{CE}(\sigma(\frac{Z_t}{T}),\sigma(\frac{Z_s}{T}))$$



$L_{CE}()$: *Cross entropy loss*
$\sigma()$: *Softmax*
Z_s: *Output logits of Student network*
Z_t: *Output logits of Teacher network*
$\hat{y}$: *Ground truth(one-hot)*
α: *Balancing parameter*
T: *Temperature hyperparameter*

- **Student Loss**
  - Ground truth와 Student의 분류 결과와의 차이를 Cross Entropy Loss로 계산.
  - i.e. Student Network hard prediction & Original hard label

- **Distillation Loss**
  - Teacher network와 Student network의 output logit을 softmax로 변환한 후, temperature를 통한 Soft label 된 값의 차이를 Cross Entropy Loss로 계산.
  - Teacher와 Student의 분류 결과가 같다면 작은 값
  - i.e. Student Network soft prediction & Teacher Network soft label

Result 1: MNIST image data

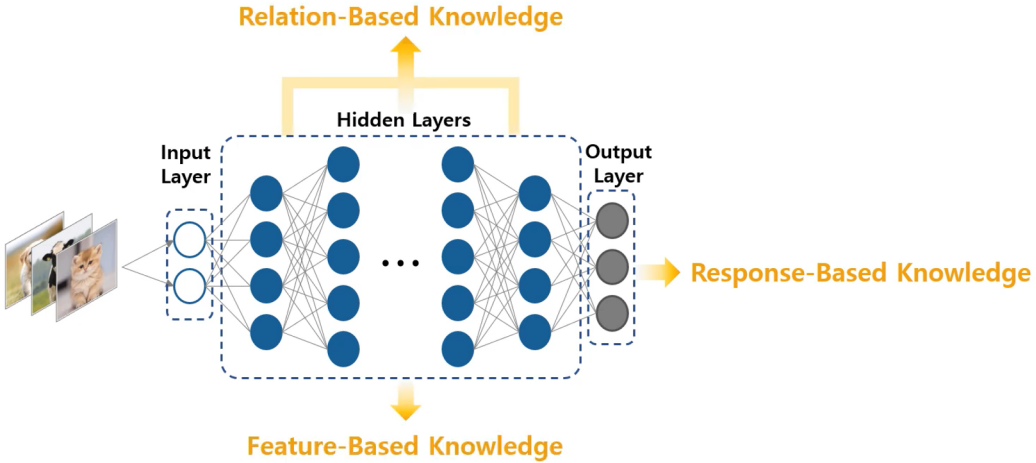
- Two hidden layer + ReLU : 146 test errors **Light model (original)**
- Two hidden layer + ReLU + DropOut : 67 test errors **Heavy model (ensembled)**
- Two hidden layer + ReLU + Soft Target : 74 test errors **Light model (distilled)**

Result 2: speech recognition

System	Test Frame Accuracy	WER
Baseline	58.9%	10.9%
10xEnsemble	61.1%	10.7%
Distilled Single model	60.8%	10.7%

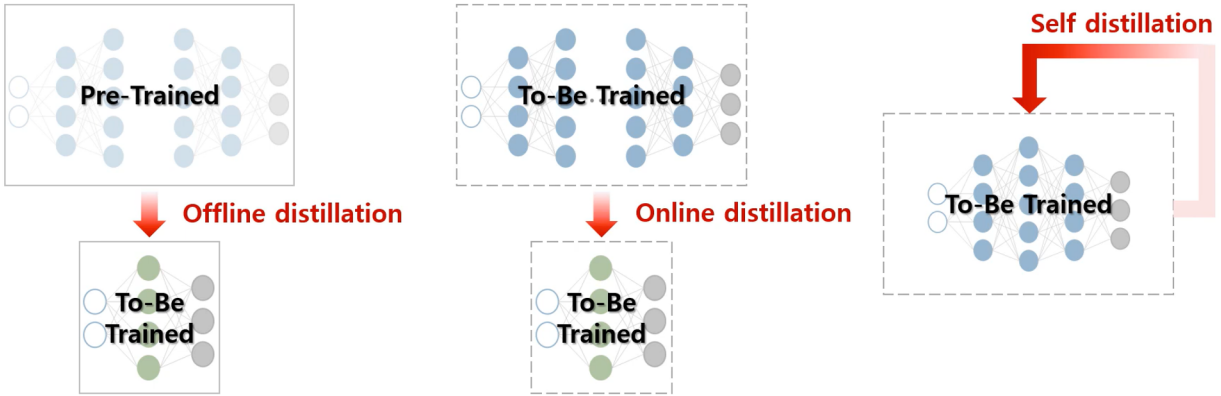
Knowledge Distillation **What**

- Response - Based
- Feature - Based
  - “Knowledge Transfer via Distillation of Activation Boundaries Formed by Hidden Neurons (2018)”
  - https://arxiv.org/abs/1811.03233
- Relation - Based
  - “Relational Knowledge Distillation (2019)”
  - https://arxiv.org/abs/1904.05068



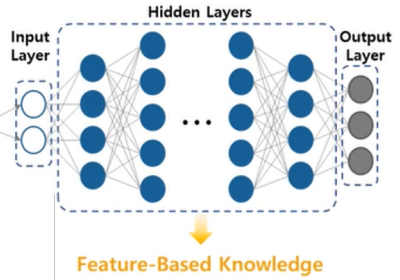
Knowledge **Distillation** How

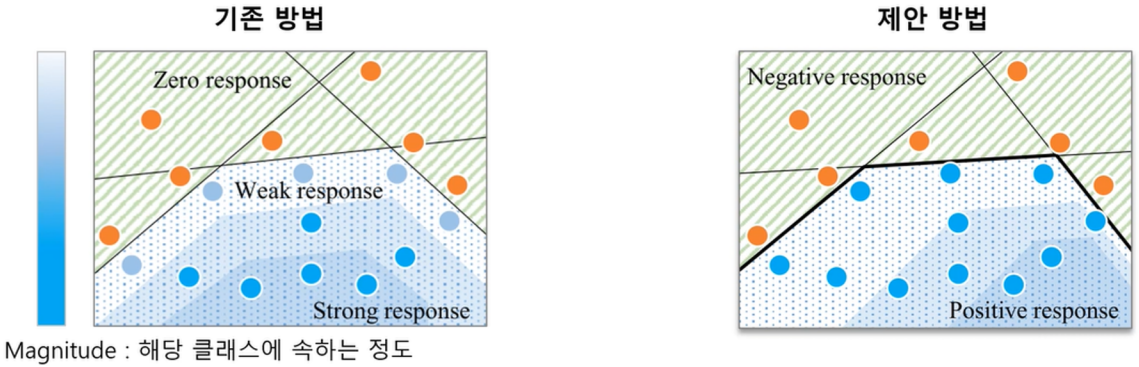
- Offline - Distillation
- Online - Distillation
  - “Large Scale Distributed Neural Network Training Through Online Distillation (2018)”
  - https://arxiv.org/abs/1804.03235
- Self - Distillation
  - “Be your Own Teacher: Improve the Performance of Convolutional Neural Networks via Self Distillation (2019)”
  - https://arxiv.org/abs/1905.08094



Knowledge Distillation

- **Feature - Based**
  - “Knowledge Transfer via Distillation of Activation Boundaries Formed by Hidden Neurons (2018)”
  - 2019 AAAI (Association for the Advancement of Artificial Intelligence) 에 발표된 논문.
  - 목표: **Teacher와 Student의 Activation Boundary만 같아지도록 학습.**
  - “Activation Boundary를 따른다”: 값이 얼마나 커야하는지 작아야 하는지 중요하지 않음.



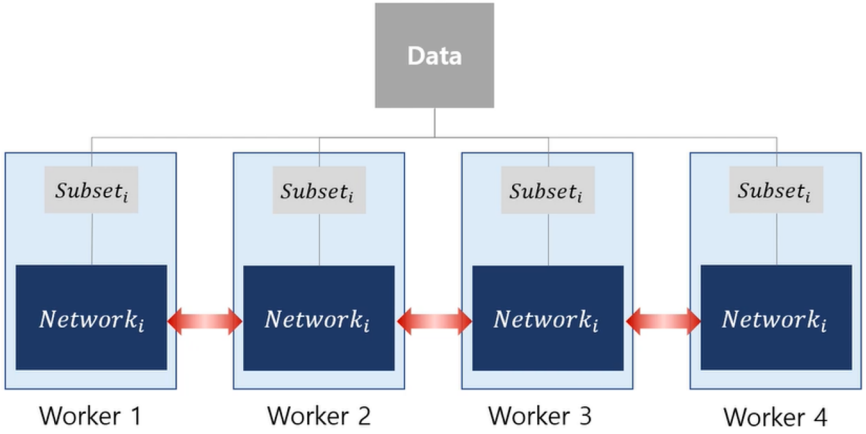


$$L_{activation} = \left\| \rho(T(x_i)) \odot \sigma(\mu\mathbf{1} - S(x_i)) + (\mathbf{1} - \rho(T(x_i))) \odot \sigma(\mu\mathbf{1} + S(x_i)) \right\|_2^2$$

Knowledge **Distillation**

- **Online - Distillation**
  - “Large Scale Distributed Neural Network Training Through Online Distillation (2018)”
  - 2018년 ICLR (International Conference on Learning Representations)에서 발표.

  - **Teacher model와 student model이 동시에 학습하면서 복사된 네트워크끼리 지식을 전달하는 기법.**
  - 멀티 GPU를 통한 데이터 병렬처리와 더불어 복사된 네트워크끼리 서로 지식을 전달.
  - 병렬적으로 파라미터가 업데이트, 다른 모델들의 평균 예측값과 일치하도록 학습.



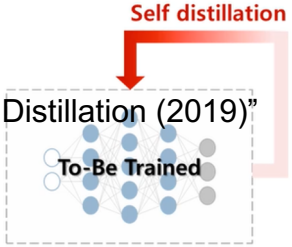
for n steps do
  for θ_i in model − set do
    y_{truth}, x = get_train_example()
    $\theta_i = \theta_i - \eta\nabla_{\theta_i}\{\phi(y_{truth}, F(\theta_i, x))\}$
  end for
end for

while not converged do
  for θ_i in model − set do
    y_{truth}, x = get_train_example()
    $\theta_i = \theta_i - \eta\nabla_{\theta_i}\{\phi(y_{truth}, F(\theta_i, x)) + \psi((\frac{1}{(N-1)}\sum_{j\neq i}F(\theta_j, x), F(\theta_i, x))\}$
  end for
end while

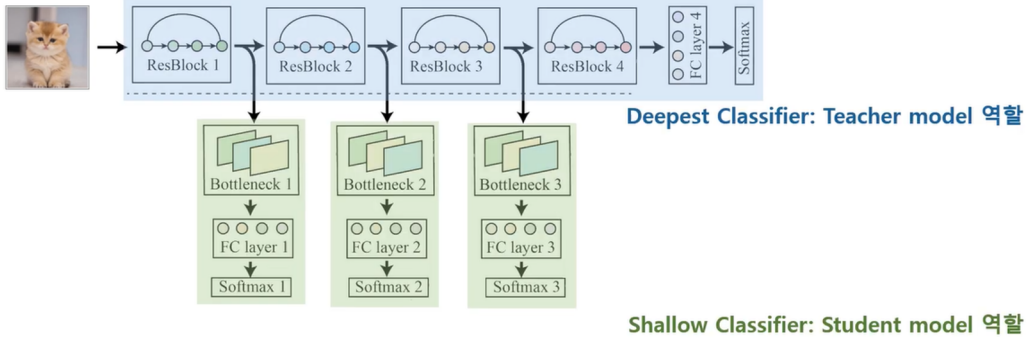
Distillation loss term

나머지 네트워크의 예측값의 평균

Knowledge **Distillation**



- **Self - Distillation**
  - “Be your Own Teacher: Improve the Performance of Convolutional Neural Networks via Self Distillation (2019)”
  - 2019년 ICCV (International Conference on Computer Vision) 에서 발표.
  - 하나의 네트워크 안에서 지식 전달이 이뤄지는 연구.
  - 하나의 모델을 크게 2가지 부분으로 나눌 수 있음.
    - Deepest Classifier: Teacher model 역할, (파란색) : ResNet block으로 구성된 깊은 분류기
    - Shallow Classifier: Student model 역할, (연두색): ResNet block이 끝나는 시점마다 Bottlenect, FC lyaer로 구성되어 얕은 분류기.
  - 학습 과정에서 계속 피드백을 주어서 데이터를 잘 표현.



$$L_{Task} = CrossEntropy\,(softmax(q_i), y_{truth})$$

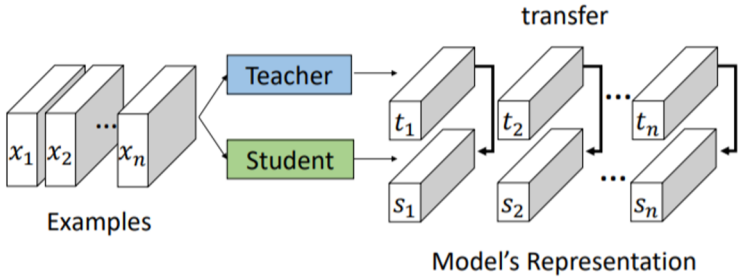
$$L_{soft} = KL(q_i, q_{deep})$$

$$L_{feature} = \left|\left|F_i - F_{deep}\right|\right|_2^2$$

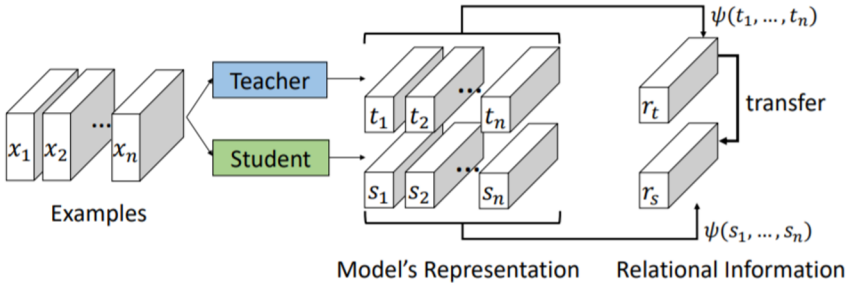
$$L_{total} = (1-\alpha)L_{task} + \alpha \cdot L_{soft} + \lambda \cdot L_{feature}$$

Relational Knowledge Distillation

Knowledge Distillation versus **Relational Knowledge Distillation**



Individual Knowledge Distillation

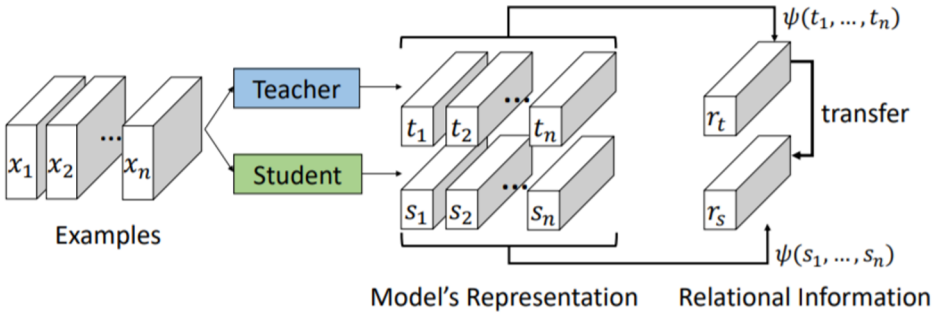


Relational Knowledge Distillation

- **Individual Knowledge Distillation**
  - **Output** of individual examples represented by the teacher
  - teacher의 output과 student의 output이 있었을 때, 각각의 example에 대해서 teacher의 output과 student의 output을 동일하게 하는 것을 목표로 함

- **Relational Knowledge Distillation**
  - **Relations** among examples represented by the teacher
  - teacher에 의해서, 각각의 output 들이 어떻게 관계되는가에 대한 정보를 transfer 하는 것도 하나의 knowledge.
  - 즉, 관계를 transfer.

Relational Knowledge Distillation



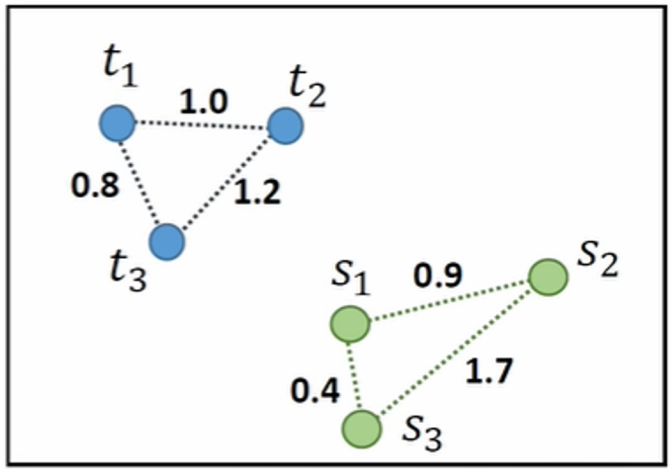
Relational Knowledge Distillation

$$\mathcal{L}_{\text{RKD}} = \sum_{(x_1,..,x_n)\in\mathcal{X}^N} l\big(\psi(t_1,..,t_n), \psi(s_1,..,s_n)\big),$$

- **Relational Knowledge Distillation: Generalization**
  - potential 함수를 어떻게 정의하느냐 따라서, 어떠한 relation을 transfer 하는지가 정해짐.
  - 제안하는 방식에서는 embedding space 상에서의 structure를 transfer 하는
    - **Distance - wise loss function** (pair)
    - **Angle - wise loss function** (triplet)

=> structure의 정보를 담고 있을 수 있다는 distance - wise loss function과 angle - wise loss function을 사용해서 relational KD는 embedding space 상의 structure를 transfer 하는 방식을 제안.

- **Distance - wise loss function (RKD-D)**
  - t1, t2, t3 간의 3개의 변이 가지고 있는 embedding space 내의 상대적인 거리와 student example에서 가지고 있는 s1, s2, s3 간의 상대적 거리를 동일하게 만드는 것.



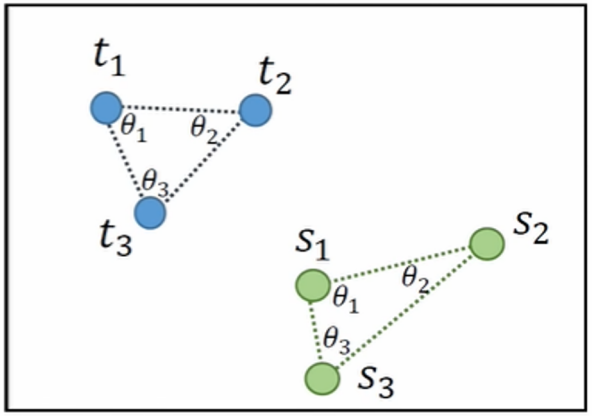
$$\psi_{\mathrm{D}}(t_i, t_j) = \frac{1}{\mu} \|t_i - t_j\|_2, \qquad \mu = \frac{1}{|\mathcal{X}^2|} \sum_{(x_i, x_j) \in \mathcal{X}^2} \|t_i - t_j\|_2.$$

$$\mathcal{L}_{\text{RKD-D}} = \sum_{(x_i, x_j) \in \mathcal{X}^2} l_\delta\big(\psi_{\mathrm{D}}(t_i, t_j), \psi_{\mathrm{D}}(s_i, s_j)\big),$$

$$l_\delta(x, y) = \begin{cases} \frac{1}{2}(x-y)^2 & \text{for } |x-y| \le 1, \\ |x-y| - \frac{1}{2}, & \text{otherwise.} \end{cases}$$

=> structure의 정보를 담고 있을 수 있다는 distance - wise loss function과 angle - wise loss function을 사용해서 relational KD는 embedding space 상의 structure를 transfer 하는 방식을 제안.

- **Angle - wise loss function (RKD-A)**
  - embedding space 상에서의 t1, t2, t3 점 3개에 대해서 정의되는 각도와 s1, s2, s3 각도와 같아지도록 학습.



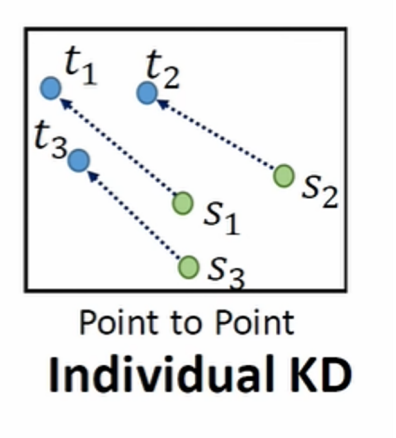
$$\psi_{\mathrm{A}}(t_i, t_j, t_k) = \cos \angle t_i t_j t_k = \langle \mathbf{e}^{ij}, \mathbf{e}^{kj} \rangle$$

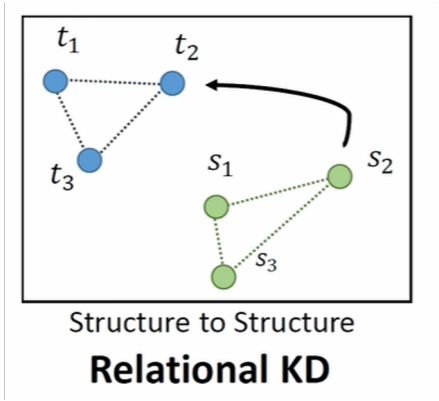
$$\text{where} \quad \mathbf{e}^{ij} = \frac{t_i - t_j}{\|t_i - t_j\|_2}, \mathbf{e}^{kj} = \frac{t_k - t_j}{\|t_k - t_j\|_2}.$$

$$\mathcal{L}_{\text{RKD-A}} = \sum_{(x_i, x_j, x_k) \in \mathcal{X}^3} l_\delta \left(\psi_{\mathrm{A}}(t_i, t_j, t_k), \psi_{\mathrm{A}}(s_i, s_j, s_k) \right),$$

=> structure의 정보를 담고 있을 수 있다는 distance - wise loss function과 angle - wise loss function을 사용해서 relational KD는 embedding space 상의 structure를 transfer 하는 방식을 제안.

- **Distance - wise loss function**
- **Angle - wise loss function**





- IKD: 서로의 output 값을 matching 시키는 방식: point-to-point, 절대적인 student의 output 값과 teacher의 output 값이 같아지도록 학습.
- RKD: 서로의 상대적인 구조, 서로의 상대적인 위치에 대해서만 고려함: structure - to - structure

Training with RKD

- **Where to Apply?**
  - relational 한 knowledge가 가장 중요하다고 생각될 수 있는, embedding space나 hidden activation에 대해서는 어떠한 layer에도 적용될 수 있음.
  - 즉, 절대적인 output 값이 중요하지 않은 embedding layer, embedding network, hidden layer 등에 적용하면 성공적으로 지식 전이에 활용할 수 있음.

  - 그러나, 각각의 individual output 값이 중요한 layer (e.g. softmax layer for classification) 에서는 적용될 수 없음.
  - 왜냐하면, RKD는 output 값들이 가지고 있는 relation을 transfer 하는 방법으로, 각각의 output 의 값을 없애 정보들이 사라지기 때문.

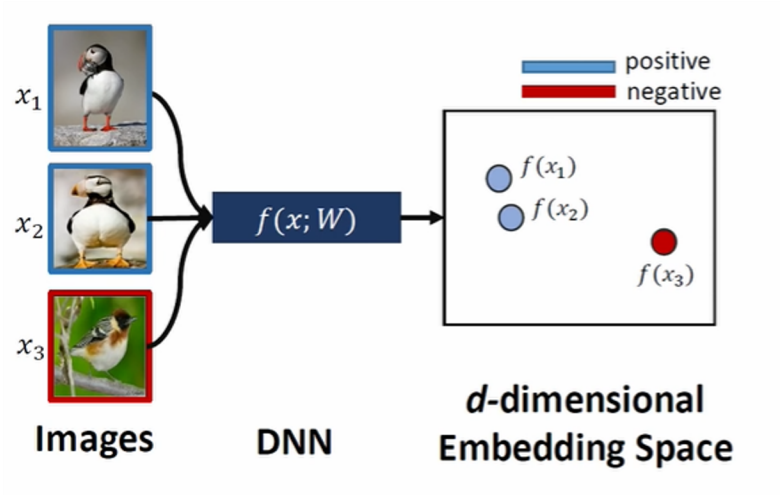
- **How to Apply?**
  - RKD loss can be combined with task-specified loss, $\mathcal{L}_{task} + \lambda \cdot \mathcal{L}_{RKD}$

  - RDK loss can be used solely for training embedding network, $\mathcal{L}_{RKD}$

Experiment

- **Metric Learning (Image retrieval)**
- **Image Classification**
- **Few-shot learning**

- **Metric Learning**
  - It aims to train an embedding model.
  - In embedding space, distances between projected examples correspond to their semantic similarity.



Experiment; Metric Learning

- Evaluation
  - Image retrieval(task), recall@k(성능을 측정하는 지표)
- Dataset
  - Cars 196
  - CUB-200-2011
  - Stanford Online Products
- Architecture
  - Teacher: ResNet50 (backbone) + 512-d fc layer (embedding layer) + L2 normalization
  - Student: ResNet18 + various dimension fc layer + L2 normalization (optional)
- Targeting layer of RKD
  - Final embedding outputs of teacher and student
- Training Objective
  - Teacher: Triplet loss & Distance-weighted sampling
  - Student: Triplet loss, RKD-D, RKD-A, RKD-DA, DarkRank

Experiment; Metric Learning

	Baseline (Triplet)	DarkRank	Ours			
			Triplet+RKD-DA	RKD-D	RKD-A	RKD-DA
L2 normalization	O	O	O	O / X	O / X	O / X
ResNet18-16	37.71	44.77	45.44	46.34 / 48.09	45.59 / **48.60**	45.76 / 48.14
ResNet18-32	44.62	51.96	53.39	52.68 / **55.72**	53.43 / 55.15	53.58 / 54.88
ResNet18-64	51.55	54.93	55.93	56.92 / 58.27	56.77 / 58.44	57.01 / **58.68**
ResNet18-128	53.92	56.52	57.11	58.31 / 60.31	58.41 / **60.92**	59.69 / 60.67
ResNet50-512 (Teacher)	61.24					

(a) Recall@1 on CUB-200-2011

	Baseline (Triplet)	DarkRank	Ours			
			Triplet+RKD-DA	RKD-D	RKD-A	RKD-DA
L2 normalization	O	O	O	O / X	O / X	O / X
ResNet18-16	45.39	52.92	61.46	63.23 / 66.02	61.39 / **66.25**	61.78 / 66.04
ResNet18-32	56.01	65.90	73.13	73.50 / **76.15**	73.23 / 75.89	73.12 / 74.80
ResNet18-64	64.53	70.34	78.08	78.64 / **80.57**	77.92 / 80.32	78.48 / 80.17
ResNet18-128	68.79	73.24	78.78	79.72 / 81.70	80.54 / 82.27	80.00 / **82.50**
ResNet50-512 (Teacher)	77.17					

(b) Recall@1 on Cars 196

Experiment; Metric Learning (Self-Distillation)

- Teacher: ResNet50 + 512-d fc + L2 normalization
  - Trained using triplet loss
- Student: ResNet50 + 512-d fc
  - Trained using RKD-DA

	CUB [40]	Cars [14]	SOP [21]
ResNet50-512-Triplet	61.24	77.17	76.58
ResNet50-512@Gen1	**65.68**	**85.65**	**77.61**
ResNet50-512@Gen2	65.11	85.61	77.36
ResNet50-512@Gen3	64.26	85.23	76.96

Experiment; Metric Learning (Comparison with SOTA methods)

		CUB-200-2011 [40]				Cars 196 [14]				Stanford Online Products [21]			
	K	1	2	4	8	1	2	4	8	1	10	100	1000
GoogLeNet [35]	LiftedStruct [21]-128	47.2	58.9	70.2	80.2	49.0	60.3	72.1	81.5	62.1	79.8	91.3	97.4
	N-pairs [34]-64	51.0	63.3	74.3	83.2	71.1	79.7	86.5	91.6	67.7	83.8	93.0	97.8
	Angular [41]-512	54.7	66.3	76.0	83.9	71.4	81.4	87.5	92.1	70.9	85.0	93.5	98.0
	A-BIER [22]-512	57.5	68.7	78.3	86.2	82.0	89.0	93.2	96.1	74.2	86.9	94.0	97.8
	ABE8 [13]-512	60.6	71.5	79.8	87.4	**85.2**	**90.5**	94.0	96.1	**76.3**	**88.4**	94.8	98.2
	RKD-DA-128	60.8	72.1	81.2	**89.2**	81.7	88.5	93.3	96.3	74.5	88.1	95.2	98.6
	RKD-DA-512	**61.4**	**73.0**	**81.9**	89.0	82.3	89.8	**94.2**	**96.6**	75.1	88.3	**95.2**	**98.7**
ResNet50 [10]	Margin [42]-128	63.6	74.4	83.1	90.0	79.6	86.5	91.9	95.1	72.7	86.2	93.8	98.0
	RKD-DA-128	**64.9**	**76.7**	**85.3**	**91.0**	**84.9**	**91.3**	**94.8**	**97.2**	**77.5**	**90.3**	**96.4**	**99.0**

- CUB dataset에서 SOTA performance (regardless of backbone network)
- Cars 196 & Stanford Online Products dataset에서 second-best performance
    - Note that, ABE8-512 network는 본 논문의 model 보다 더 많은 computing resource가 필요함.

Experiment; Image Classification

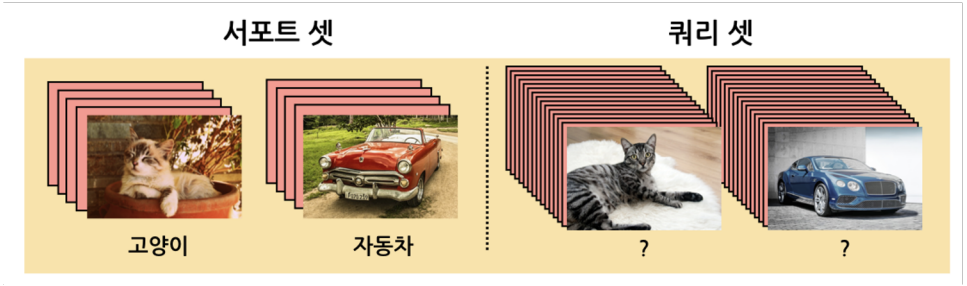
- Dataset
  - CIFAR-10, CIFAR-100
- Architecture
  - Teacher: ResNet50 (backbone)
  - Student: VGG-11 with BatchNorm
- Targeting layer of RKD
  - Teacher: output of *avgpool* layer
  - Student: output of *pool5* layer
- Training Objective
  - Teacher: cross-entropy
  - Student: cross-entropy + (Hinton et al., RKD-D and RKD-DA)

	CIFAR-10	CIFAR-100
Baseline	92.47	71.26
Hinton *et al.*	92.84	74.26
RKD-D	92.64	72.27
RKD-DA	93.02	72.97
RKD-DA + Hinton *et al.*	**93.11**	**74.66**
Teacher	95.01	77.76

(a) Accuracy (%) on CIFAR-10 and CIFAR-100.

Experiment; Few-shot learning

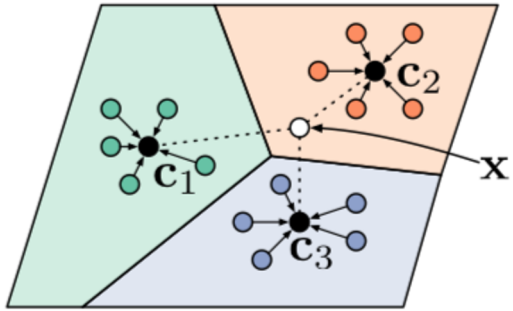
- Few-shot learning **(N-way K-shot)**
  - training dataset의 수가 매우 적은 task에서의 모델 학습.
  - 데이터 수가 매우 적은 few-shot learning task에서는 데이터셋을 2가지로 구분
    - Support data: 데이터셋을 훈련에 사용
    - Query data: 데이터셋을 테스트에 사용
  - N-way K-shot
    - N-way: 범주의 수
    - K-shot: 범주별 support data 수
    - (e.g. 2-way 5-shot)



  - 즉, few-shot learning은 K가 매우 작은 상황에서의 모델 학습.
  - few-shot learning은 N과 반비례하며, K와 비례하는 관계
    - N이 커질수록 모델 성능이 낮아짐 (e.g. 5지 선다형 문제에서 답을 찍는 것과, 100지 선다형 문제에서 답을 찍을 때 기대 성적이 다름)
    - K가 커질수록 모델 성능이 높아짐

Experiment; Few-shot learning

- Few-shot learning (N-way K-shot)
- **Prototypical Network for few-shot learning**
  - training set에서 보지 못했던 새로운 class의 이미지가 몇 개 주어졌을 때, test set에서 주어진 class에 대해서 classification을 수행하는 task.
  - embedding network를 사용해서 embedding space 학습.
  - embedding space를 사용해서 classification 수행.
  - e.g. 5-way 5-shot task가 주어졌을 때,
    - 기존 방식에서는 support data (5*5) 25와 쿼리 데이터 간의 거리를 일일이 계산.
    - Prototypical network for few-shot learning 에서는 범주별 서포트데이터의 평균 위치인 prototype (프로토타입) 개념 사용. 결과론적으로 모델은 5개의 범주를 대표하는 프로토타입 벡터와 쿼리 벡터와의 거리만 계산하면 됨.
  - https://arxiv.org/abs/1703.05175



(a) Few-shot

Experiment; Few-shot learning

- Dataset
  - Omniglot, miniImageNet
- Architecture
  - Teacher: 4 convolutional layers
  - Student: Same with teacher
- Targeting layer of RKD
  - Teacher: Final embedding output
  - Student: Same with teacher
- Training Objective
  - Teacher: Snell et al. (prototypical networks)
  - Student: Snell et al. + (RKD-D or RKD-DA)

	1-Shot 5-Way	5-Shot 5-Way
RKD-D	49.66 ± 0.84	67.07 ± 0.67
RKD-DA	$\mathbf{50.02} \pm 0.83$	$\mathbf{68.16} \pm 0.67$
Teacher	49.1 ± 0.82	66.87 ± 0.66

(a) Accuracy (%) on *mini*ImageNet.

	5-Way Acc.		20-Way Acc.	
	1-Shot	5-Shot	1-Shot	5-Shot
RKD-D	98.58	**99.65**	95.45	**98.72**
RKD-DA	**98.64**	99.64	**95.52**	98.67
Teacher	98.55	99.56	95.11	98.68

(a) Accuracy (%) on Omniglot.

Conclusion

- Relational KD that effectively transfer knowledge using relations among data examples represented by the teacher.
- Experiments conducted on different tasks and benchmarks show that the Relational KD improves the performance of the educated student networks with a significant margin.
- RKD framework opens a door to a promising area of effective knowledge transfer with high-order relations.

End of the Document