📖

[논문 리뷰] Knowledge Distilliation [2015]

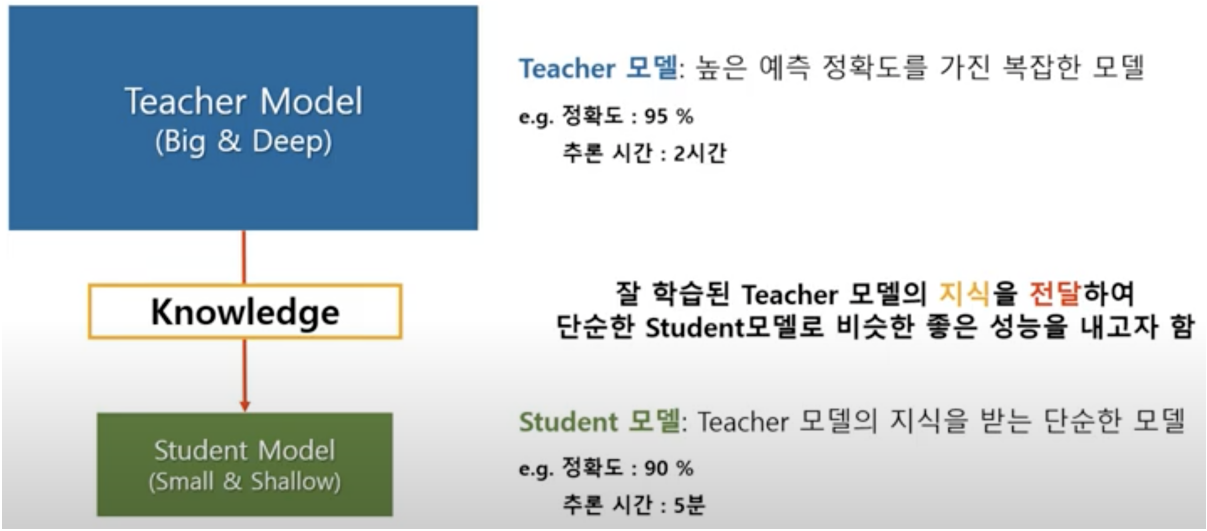
🌟 **본 논문은, Teacher Model이 가지고 있는 지식을 작은 모델인 Student Model에 Transfer하는 Knowledge Distillation에 관한 논문입니다.**

머신러닝에서, **성능을 개선시키는 가장 단순한 방법은, 같은 데이터를 다른 모델들에 학습을 시킨 후, 예측값을 평균화 하는 것**입니다. 하지만, **상당한 Computational이 상당히 많이 소모된다는 단점**이 있습니다.

연구질들은 가볍게 만들어진 단일 모델로 앙상블하여, **Knowledge를 compress를 하는 것이 가능하다는 것**을 증명하였습니다. 그리고 **different compression technique를 개발**하였습니다.

또한, 하나이상의 앙상블 유형과 전체 모델 그리고, 혼동되는 클래스들을 구별하는 것을 학습한 새로운 모델들을 도입하였습니다.

Distillation은 앙상블된 knowledge를 압축하여, Single Model로 distillation하는 방식을 제안합니다.



Introduction

Cumbersome model이 train된 후에, 다양한 방식의 학습이 가능합니다. 상용가능한 작은 모델에 cumbersome model의 knowledge를 transfer하는 것이 가능합니다.
이러한 접근법을 "Knowledge Distillation"이라고, 부릅니다.

> 💡 논문에서는 **large ensemble of Models의하여 얻은 knowledge**를
> **Single Small model에 transfer하는 것을 증명**합니다.

일반적인 통념으로는, **trained model과 learned parameter values를 식별해서 보는 경향**이 있습니다.
이러한 **Conceptual block은 Model의 form을 변경하면서, same knowledge를 가질 수 있게 하는지 생각하는 것을 어렵게 합니다.**

일반적인 **모델의 학습은 정답의 대한 확률의 평균을 최대화하는 것을 목표로 합니다. 하지만, 이러한 방식은 모든 incorrect answer에도 확률을 지정**합니다.
뿐만 아니라, **correct answer 와 incorrect answer 사이에서의 probability의 편차**는 상당히 큽니다.

Small Model을 학습시키기 위해 cumbersome model의 일반적인 ability를 small model로 transfer하는 obvious way는 cumbersome model의 Class probabilities를 soft targets으로 활용하는 것이다.

만약, soft target이 high entropy를 가지고 있다면, hard targets보다, 많은 information을 전달합니다. 추가로, training case에서 less variance를 가지게 됩니다. 그리하여, Small Model은 cumbersome model보다 적은 데이터를 활용함에도, 높은 학습률 가지게 됩니다.

실제로 **2020년 efficientNet 기반의 Knowledge Distillation하는 Noisy Student와 Meta Pseudo Label Classification에서 SOTA를 달성**합니다.

Distillation

Distillation의 일반적인 형태는,
transfer set을 model에 distillation함으로서, **knowledge를 transfer**합니다.
그리고, 각각의 **soft target distribution을 사용**합니다. (**T를 조절하여, 여러 case를 만듭니다.**)

Neural networks typically produce class probabilities by using a "softmax" output layer that converts the logit, z_i, computed for each class into a probability, q_i, by comparing z_i with the other logits.

$$q_i = \frac{exp(z_i/T)}{\sum_j exp(z_j/T)} \tag{1}$$

where T is a temperature that is normally set to 1. Using a higher value for T produces a softer probability distribution over classes.

연구에서는 **two different objective function의 weighted를 average하는 것이 better way라는 것을 발견**하였습니다.

1. 첫 번째 **objective function**은 **soft targets에 대한 cross entropy를 활용합니다.**
   (위의 Crossentropy function에 T를 추가한 것을 활용합니다.)
2. **두 번째 objective function은 correct label에 대한 cross entropy를 사용합니다.**

Each case in the transfer set contributes a cross-entropy gradient, dC/dz_i, with respect to each logit, z_i of the distilled model. If the cumbersome model has logits v_i which produce soft target probabilities p_i and the transfer training is done at a temperature of T, this gradient is given by:

$$\frac{\partial C}{\partial z_i} = \frac{1}{T}(q_i - p_i) = \frac{1}{T}\left(\frac{e^{z_i/T}}{\sum_j e^{z_j/T}} - \frac{e^{v_i/T}}{\sum_j e^{v_j/T}}\right) \tag{2}$$

If the temperature is high compared with the magnitude of the logits, we can approximate:

$$\frac{\partial C}{\partial z_i} \approx \frac{1}{T}\left(\frac{1 + z_i/T}{N + \sum_j z_j/T} - \frac{1 + v_i/T}{N + \sum_j v_j/T}\right) \tag{3}$$

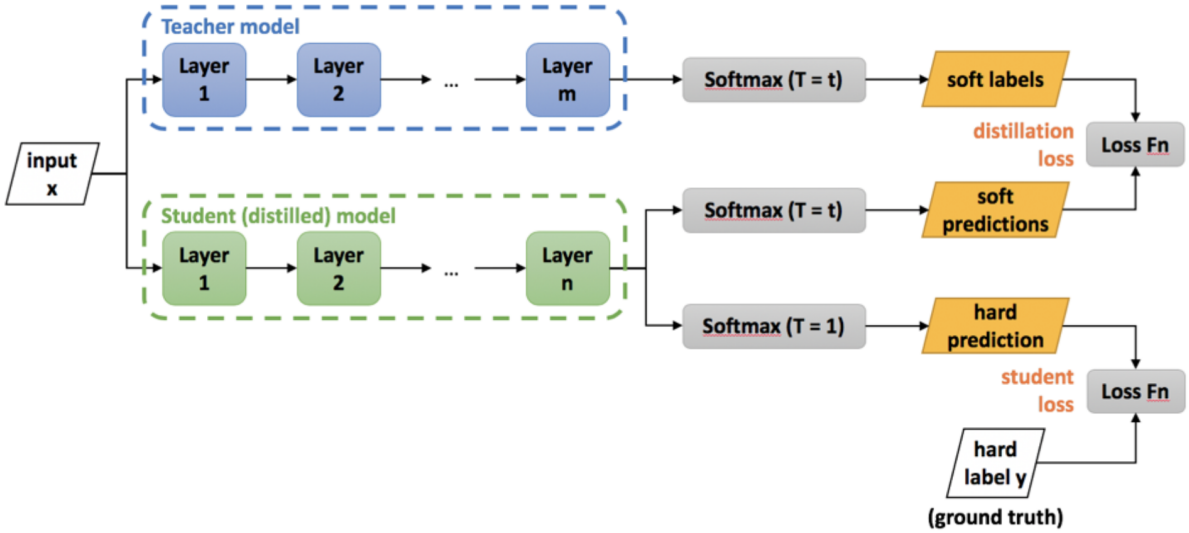
If we now assume that the logits have been zero-meaned separately for each transfer case so that $\sum_j z_j = \sum_j v_j = 0$ Eq. 3 simplifies to:

$$\frac{\partial C}{\partial z_i} \approx \frac{1}{NT^2}(z_i - v_i) \tag{4}$$

Method

1. **Teacher Model을 먼저 training**합니다.
2. **Student & Teacher Model로부터 loss**를 구합니다.
3. **Teacher Model과 Student Model의 loss로부터, distill_loss_batch**를 구합니다.

✨ **일반적으로 alpha : 0.1, Temperature : 10이 성능이 잘 나온다고 합니다.**




```
def distillation(y, labels, teacher_scores, T, alpha):
    return nn.KLDivLoss()(F.log_softmax(y/T), F.softmax(teacher_scores/T)) * (T*T * 2.0 * alpha)
                          + F.cross_entropy(y, labels) * (1. - alpha)
```

Experiments

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

Table 1: Frame classification accuracy and WER showing that the distilled single model performs about as well as the averaged predictions of 10 models that were used to create the soft targets.

Distilling the Knowledge in a Neural Network

A very simple way to improve the performance of almost any machine learning algorithm is to train many different models on the same data and then to average their predictions. Unfortunately, making predictions using a whole ensemble of models is cumbersome and may be too computationally expensive to allow

https://arxiv.org/abs/1503.02531

knowledge-distillation-pytorch/distill_mnist.py at master · haitongli/knowledge-distillation-pytorch

You can't perform that action at this time. You signed in with another tab or window. You signed out in another tab or window. Reload to refresh your session. Reload to refresh your session.

https://github.com/haitongli/knowledge-distillation-pytorch/blob/master/mnist/distill_mnist.py

haitongli/**knowledge-distillation-pytorch**

A PyTorch implementation for exploring deep and shallow knowledge distillation (KD) experiments with flexibility

4 Contributors · 13 Issues · 1k Stars · 302 Forks

[논문 읽기] PyTorch 구현 코드로 살펴보는 Knowledge Distillation(2014), Distilling the Knowledge in Neural Network

안녕하세요, 오늘 읽은 논문은 Distilling the Knoeledge in a Neural Network 입니다. 해당 논문은 Knowledge Distillation을 제안합니다. Knowledge Distillation은 teacher model이 갖고 있는 지식을 더 작은 모델인 student model에 transfer 하는 것을 의미합니다. 사이즈가 큰 teacher model이 갖고 있는 지식을 사이즈가 작은 student model에 지식을 transfer한다면, model compression의 효과가 있습

https://deep-learning-study.tistory.com/699