Distilling the Knowledge in a Neural Network

2022.01.01

주세준

1.Memory Limitation :
모델들의 사이즈가 점점 커지면서 GPU에서 큰 모델을 학습하는 것이 점점 어려워지고 있다.

큰 배치 사이즈가 학습에 효과적이라는 의견이 나오면서 배치 사이즈의 증가는 메모리에 많은 부담이 되고 있다.

2.Training/Inference Speed :
학습에 필요한 gradient는 모델의 크기에 비례하기 때문에 학습 속도를 올리더라도, 모델이 커짐에 따라 학습에 보다 많은 시간이 소요되게 된다.
모델 크기가 증가하면서 추론에 걸리는 시간 역시 늘어나기 때문에 문제가 될 수 있다.

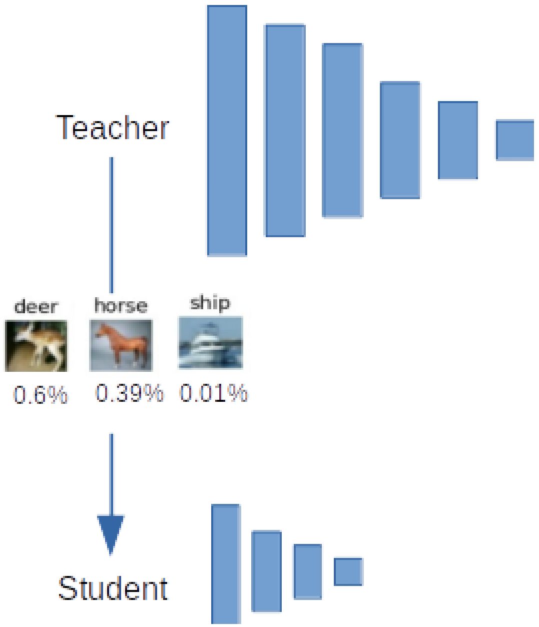
3, Worse Performance :
이런 문제점을 해결하기 위한 한 가지 접근 방법 중 하나는 **분산 학습**입니다.
기존의 많은 연구들은
1)데이터 병렬화와
2)모델 병렬화 방식과 같이 복수의 GPU를 사용한 학습을 통한 해결 시도
하지만 같은 데이터에서 단순히 모델만을 키운다고 성능이 계속해서 증가하지는 않기 때문입니다.
지나치게 큰 모델은 과적합(overfitting)하기 쉽고
이를 막기 위해서는 더 많은 데이터를 사용하거나, 정규화(regularization) 방법을 도입하여 해결할 필요가 있다.

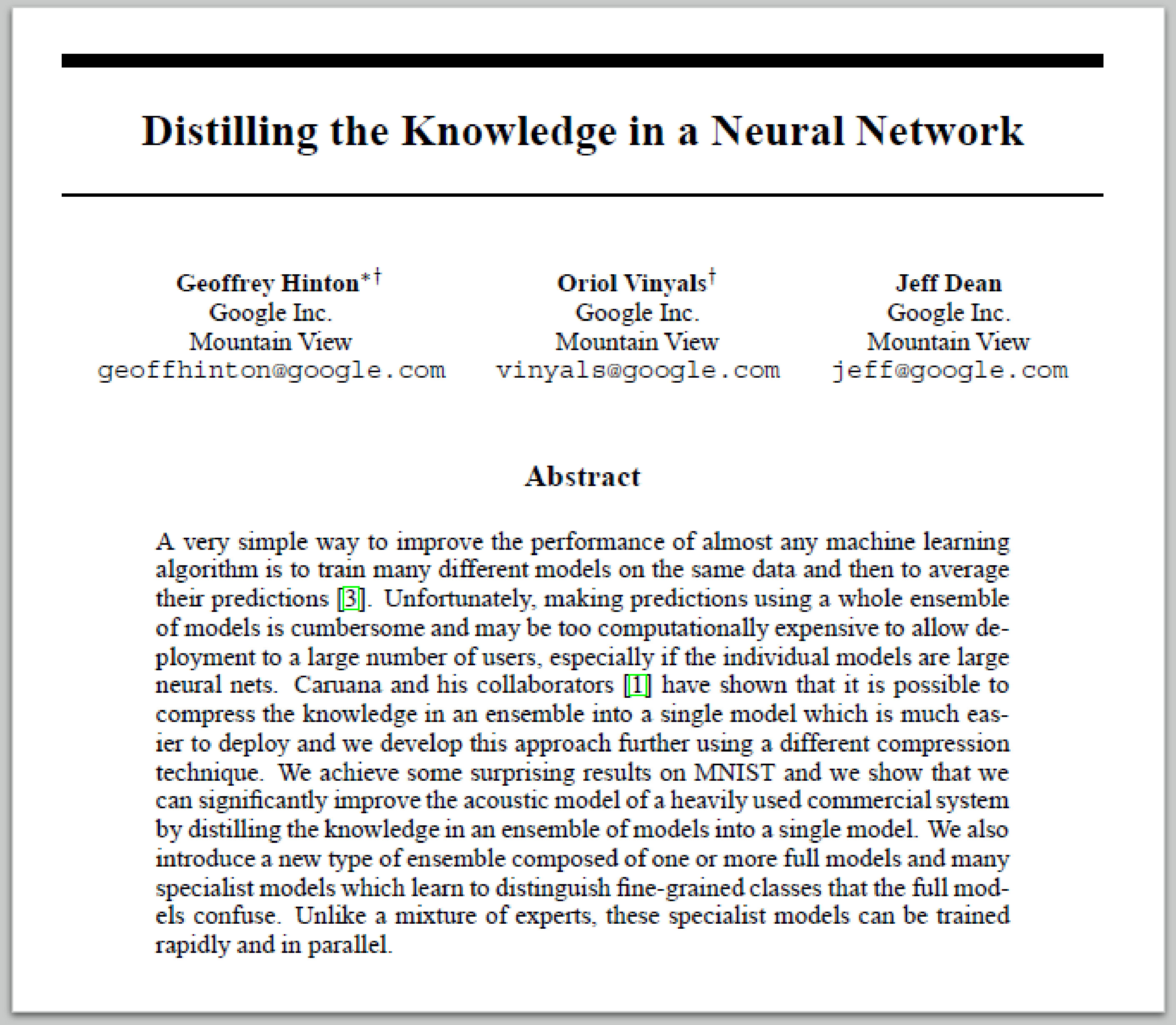
4.Practical problems :
분산 학습을 한다고 하더라도 모델이 커짐에 따라, 많은 GPU를 준비해야 하는 것은 작은 회사/연구소/대학원 등에서는 부담이 될 수 있습니다

Methods of Compressing

- Pruning - Removes unnecessary parts of the network after training. This includes weight magnitude pruning, attention head pruning, layers, and others. Some methods also impose regularization during training to increase prunability (layer dropout).
- Weight Factorization - Approximates parameter matrices by factorizing them into a multiplication of two smaller matrices. This imposes a low-rank constraint on the matrix. Weight factorization can be applied to both token embeddings (which saves a lot of memory on disk) or parameters in feed-forward / self-attention layers (for some speed improvements).
- **Knowledge Distillation** - Aka “Student Teacher.” Trains a much smaller Transformer from scratch on the pre-training / downstream-data. Normally this would fail, but utilizing soft labels from a fully-sized model improves optimization for unknown reasons. Some methods also distill BERT into different architectures (LSTMS, etc.) which have faster inference times. Others dig deeper into the teacher, looking not just at the output but at weight matrices and hidden activations.
- Weight Sharing - Some weights in the model share the same value as other parameters in the model. For example, ALBERT uses the same weight matrices for every single layer of self-attention in BERT.
- Quantization - Truncates floating point numbers to only use a few bits (which causes round-off error). The quantization values can also be learned either during or after training.
- Pre-train vs. Downstream - Some methods only compress BERT w.r.t. certain downstream tasks. Others compress BERT in a way that is task-agnostic.

- NIPS 2014 워크샵에 나온 논문으로서 Distillation의 개념을 제안하였다.
- Transfer knowledge in a computationally expensive Ensemble model to a simple single model





Distilling the Knowledge in a Neural Network

Geoffrey Hinton[*†]
Google Inc.
Mountain View
geoffhinton@google.com

Oriol Vinyals[†]
Google Inc.
Mountain View
vinyals@google.com

Jeff Dean
Google Inc.
Mountain View
jeff@google.com

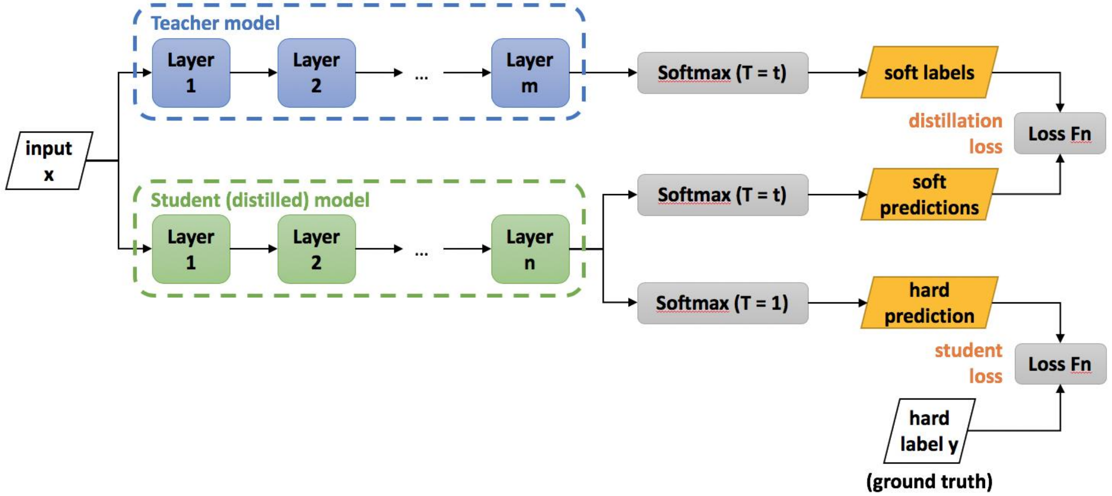
Abstract

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 [3]. Unfortunately, making predictions using a whole ensemble of models is cumbersome and may be too computationally expensive to allow deployment to a large number of users, especially if the individual models are large neural nets. Caruana and his collaborators [1] have shown that it is possible to compress the knowledge in an ensemble into a single model which is much easier to deploy and we develop this approach further using a different compression technique. We achieve some surprising results on MNIST and we show that we can significantly improve the acoustic model of a heavily used commercial system by distilling the knowledge in an ensemble of models into a single model. We also introduce a new type of ensemble composed of one or more full models and many specialist models which learn to distinguish fine-grained classes that the full models confuse. Unlike a mixture of experts, these specialist models can be trained rapidly and in parallel.

Softer softmax

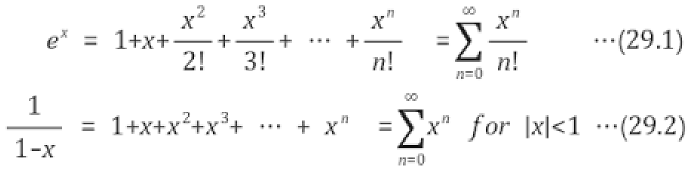
$$q_i = \frac{\exp\left(\frac{z_i}{T}\right)}{\sum_j \exp(\frac{z_j}{T})}$$

- logit z_i에 대해 prob q_i 를 위처럼 정의한다.
- Temperature (T) 라는 개념 도입
- T=1 이면 softmax (hard)
- T가 커질 수록 softer prob 생성



Matching logits is a special case of distillation

- $\frac{\partial C}{\partial z_i} = \frac{1}{T}(q_i - p_i) = \frac{1}{T}\left(\frac{e^{\frac{z_i}{T}}}{\sum_j e^{\frac{z_j}{T}}} - \frac{e^{\frac{v_i}{T}}}{\sum_j e^{\frac{v_j}{T}}}\right)$
- v_i logits of teacher model
- z_i logits of student model



$$e^x = 1+x+\frac{x^2}{2!}+\frac{x^3}{3!}+ \cdots + \frac{x^n}{n!} = \sum_{n=0}^{\infty}\frac{x^n}{n!} \quad \cdots(29.1)$$

$$\frac{1}{1-x} = 1+x+x^2+x^3+ \cdots + x^n = \sum_{n=0}^{\infty}x^n \quad for\ |x|<1 \quad \cdots(29.2)$$

- $\frac{\partial C}{\partial z_i} \approx \frac{1}{T}\left(\frac{1+\frac{z_i}{T}}{N+\sum_j \frac{z_j}{T}} - \frac{1+\frac{v_i}{T}}{N+\sum_j \frac{v_j}{T}}\right)$
- If T is high compared to mag of logits

- $\frac{\partial C}{\partial z_i} \approx \frac{1}{NT^2}(z^i - v^i)$
- Assume that logits are zero-meaned for each transfer

→ $\sum_j z_j = \sum_j v_j = 0$

Matching logits is a special case of distillation

- $\frac{\partial C}{\partial z_i} \approx \frac{1}{NT^2}(z^i - v^i)$
- 결국 loss는 $\frac{\partial C}{\partial z_i}$를 다 적분시킨 것
- 그래서 $(z_i - v_i)^2$를 최소화 시키는것이 목표

- Distilled model이 parent model에 비해서 크기가 작아 knowledge를 모두 담아내지 못한다면 temp 줄여보자 (ignore large negative logits)

- use temperature values ranging from 1 to 20.
- when the student model is very small compared to the teacher model, lower temperatures work better.

→ as we raise the temperature, the resulting soft-labels distribution becomes richer in information, and a very small model might not be able to capture all of this information. However, there's no clear way to predict up front what kind of capacity for information the student model will have.

- **Extreme Language Model Compression with Optimal Subwords and Shared Projections**
- **DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter**
- **Distilling Task-Specific Knowledge from BERT into Simple Neural Networks**
- **Distilling BERT into Simple Neural Networks with Unlabeled Transfer Data**
- **MKD: a Multi-Task Knowledge Distillation Approach for Pretrained Language Models**
- **Patient Knowledge Distillation for BERT Model Compression**
- **TinyBERT: Distilling BERT for Natural Language Understanding**
- **MobileBERT: Task-Agnostic Compression of BERT by Progressive Knowledge Transfer**

감사합니다