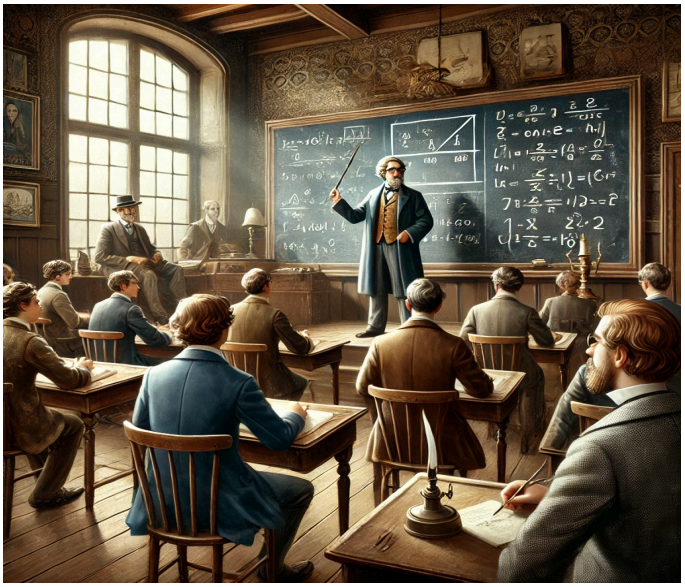
Knowledge Distillation in Neural Networks

Gautam Galada



Abstract

Knowledge distillation is a process where a larger pre-trained model (teacher) transfers its knowledge to a smaller model (student). This technique is used to create more efficient models that help the developers with a significant portion of the accuracy of the larger model while being much more computationally efficient. This article explores the application of knowledge distillation in Natural Language Processing (NLP) and Computer Vision (CV), comparing its impact and effectiveness in both domains.

Introduction to Knowledge Distillation

Knowledge distillation is a powerful technique for creating efficient neural networks by transferring knowledge from a larger, well-trained teacher model to a smaller student model. This method is particularly useful for reducing the model size, decreasing inference time, and making it feasible to deploy models on devices with limited resources, such as mobile phones or embedded systems.

“Teacher explores and exploits, Student exploits further!”

How KL Divergence Helps

Kullback-Leibler (KL) divergence is a measure of how one probability distribution diverges from another expected probability distribution. In the context of knowledge

distillation, KL divergence is used to ensure that the output probabilities of the student model closely match those of the teacher model. The KL divergence between the teacher's soft output probabilities (q) (produced with a **temperature** scaling) and the student's output probabilities (p) is given by :

$$D_{KL}(q||p) = \sum_i q(i) \log \frac{q(i)}{p(i)}$$

This metric helps in training the student model to produce output distributions similar to the teacher model, effectively transferring the "knowledge" of the teacher to the student. By minimizing the KL divergence during training, the student model learns to replicate the teacher model's behavior, thus inheriting its performance characteristics.

The Algorithm

1. Train the Teacher Model : The teacher model is trained on the dataset using standard training procedures.
2. Compute Soft Targets : The teacher model's output logits are converted into probabilities using a softmax function with temperature (T). These soft targets are softer versions of the original probabilities and contain more information about the relative similarities between classes.
3. Train the Student Model : The student model is trained using a combination of the traditional loss function (e.g., cross-entropy loss) and the KL divergence between the soft targets from the teacher and the student's output probabilities.
4. Loss Function: The combined loss function can be represented as:

$$L_{total} = \alpha \cdot L_{student} + (1 - \alpha) \cdot T^2 \cdot D_{KL}(q_T||p_T)$$

where ($L_{student}$) is the cross-entropy loss of the student, (D_{KL}) is the KL divergence, (T) is the temperature, and ***a*** is a weighting factor.

Application in NLP

- NLP models like BERT, GPT, and other transformer-based architectures are typically very large, with hundreds of millions or even billions of parameters. This makes them computationally expensive to deploy in real-time applications.
- Deploying these large models in applications such as chatbots, translation services, and text summarization requires low latency and high scalability. Knowledge distillation allows these models to be compressed significantly,

making them more suitable for deployment without compromising much on performance.
- Many NLP applications need to run on devices with limited memory, such as mobile phones or edge devices. Distilling knowledge from large models to smaller ones makes it possible to run sophisticated NLP models in such environments.

Application in CV

- While CV models like ResNet, EfficientNet, and other convolutional neural networks are large, they are generally smaller compared to state-of-the-art NLP models. But they still benefit significantly from distillation.
- Similar to NLP, many CV applications require deployment on edge devices (e.g., drones, autonomous vehicles, security cameras). Knowledge distillation helps in creating efficient models that can perform tasks like object detection, image classification, and segmentation with reduced latency and resource consumption.
- Applications such as real-time video analytics, augmented reality, and interactive systems benefit from the reduced computational load achieved through knowledge distillation, enabling faster processing and response times.

Comparison: NLP vs. CV

Impact on NLP

The compression ratios achievable in NLP are often more dramatic due to the initial size of models like BERT or GPT. Distilling these models can reduce their size by an order of magnitude while retaining a large percentage of their original performance. In NLP, even small improvements in inference time and model size can have substantial impacts, particularly in applications requiring real-time processing or deployment on millions of devices.

Impact on CV

In CV, the efficiency gains through knowledge distillation are also significant, but the starting point is often less extreme than in NLP. CV models are generally more streamlined in terms of their architecture. CV applications on edge devices benefit greatly from the reduced model size and improved efficiency, making it possible to deploy advanced vision capabilities in a broader range of scenarios.

Why Code from scratch?

Knowledge distillation is beneficial for both NLP and CV, but its impact is often more pronounced in NLP due to the larger initial size of state-of-the-art models and the critical need for efficient deployment on resource-constrained devices. In NLP, the substantial size reduction and performance retention make distillation particularly valuable, enabling sophisticated language models to be deployed in real-time, low-latency applications, whereas in CV, while the improvements in deployment efficiency and real-time processing capabilities are crucial for edge AI applications, the initial models are often smaller than those in NLP.

References

1. https://arxiv.org/abs/1503.02531
2. https://arxiv.org/abs/2105.08919