Knowledge Distillation

- training large image/text models is computationally expensive
- models often need up to 100 million parameters to achieve state-of-the-art performance
- training those models requires a lot of computational resources, e.g. GPUs, time and data
- for example, Meta's largest Llama 2 model was trained for more than 1.7 million GPU hours
  - used NVIDIA A100 80GB GPUs
  - if one would cost 5 USD per hour, the training would cost more than 8.5 million USD[1]
- infeasible to train such, or similar, models for researchers, students, or companies with limited resources
- transfer learning was first strategy to profit from large pretrained models by finetuning them on a specific task, for a, potential, different use case
- disadvantage is that model size does not change, finetuning is still computationally expensive, especially for large models
- other option is Knowledge Distillation (KD)
- here we do not finetune an existing model, but train a smaller model, the student model, to replicate, or rather predict, the outputs of a larger model, the teacher model, for a given sample
- can be applied for both supervised and self-supervised settings, meaning the teacher model has been trained in a supervised or self-supervised manner, respectively
  - former referred to as response-based KD, latter as feature-based KD
  - both will be used in this work
- has the advantage that the student model can be much smaller than the teacher model, and can, depending which of the former settings is used, have a different architecture
- emirically shown that student models much smaller than teacher models can achieve similar performance

Response-based Knowledge Distillation

- teacher model is/was trained in a supervised manner
- provides logits as predictions for a given sample
- are the target of the student model
- regress the probabilty distribution of the teacher model, so the output of the teacher after softmax has been applied on logits
  - also called soft targets, because logit for each class is divided by a temperature parameter before softmax is applied
  - smoothens the distribution and increases the relative importance of logits with lower values -> Hinton et al. argue that this makes the model learn encoded information the teacher model has learned that is not encoded in the activation for the correct class, which also helps the student model to generalize better, especially on less data, compared to a model trained from scratch
  - usually a tuneable hyperparameter, but can also be learned, as shown in < TODO: @vlmo_section >
- here Kullback-Leibler divergence (KL) is used as loss function
- mathematical formulation is as follows:
- let f be the teacher model, g the student model, and x the input sample, for example an image
- we define $u = g(x)$ and $z = f(x)$ as the output of the teacher and student model, respectively
  - those are the logits, and for a classification task of e.g. 1000 classes, 1000-dimensional vectors
- after that, we apply the softmax function on the logits, with a temperature T, to get the soft targets

[1] Calculation done based on the price per GPU hour of the NVIDIA A100 80GB GPU on AWS for instance p4de.24xlarge, as of July 2024.

$$p_i = \frac{\exp\left(\frac{u_i}{T}\right)}{\sum_j \exp\left(\frac{u_j}{T}\right)}$$

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

- p_i denotes the probability of class i according to the teacher model, and q_i the probability of class i according to the student model, goal is to minimize this difference, computed by the KL divergence:

$$L_{KD} = D_{KL}(p \parallel q) = \sum_j p_j \log\left(\frac{p_j}{q_j}\right)$$

- where j is the index of a class, and p_j and q_j the probabilities of class j according to the teacher and student model, respectively

Feature-based Knowledge Distillation

- teacher model is/was trained in a self-supervised, or supervised, manner
- student model tries to replicate/predict the (intermediate) activations of the teacher model
  - so not necessarily only the output of the teacher model, although this is also possible
- if teacher model has been trained in a self-supervised manner, feature-based is needed, as the teacher model does not provide logits or a probabilty distribution to regress
- if teacher model has been trained in a supervised manner, feature-based can also be used, but response-based is more common
- here usually the Mean Squared Error (MSE) is used as loss function, and defined as follows:

$$L_{KD} = \mathrm{MSE}(p, q) = \|p - q\|_2 = \frac{1}{k} \sum_{j=0}^{k} \left(p_j - q_j\right)^2$$

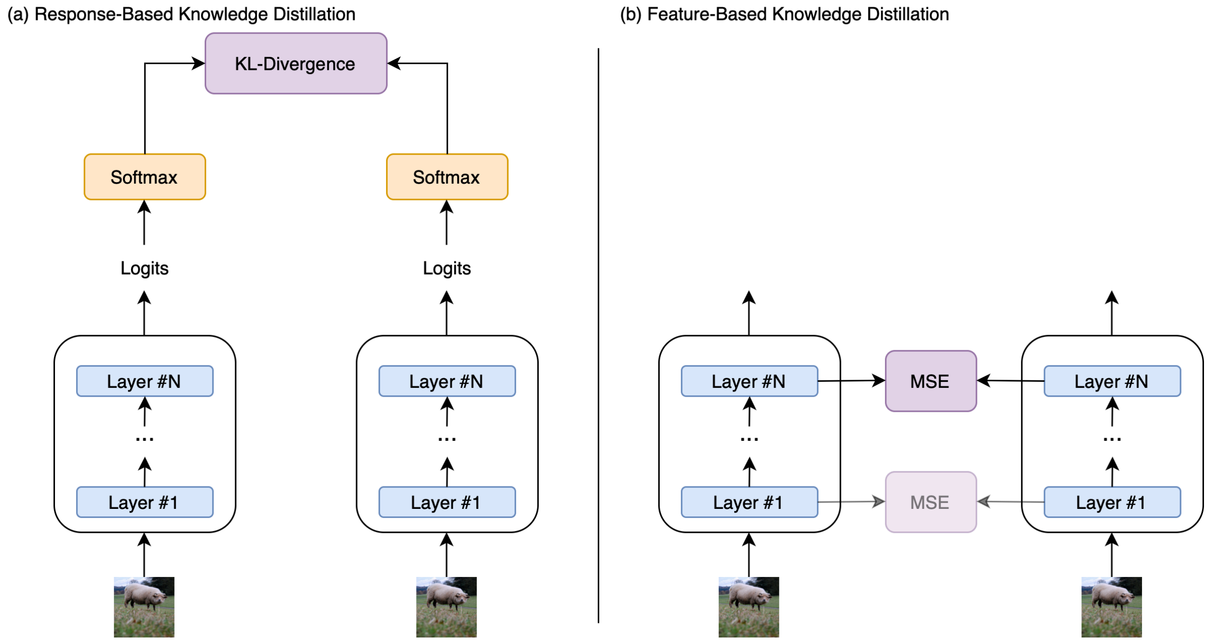

Figure 1: