Knowledge Distillation

Training large vision-language model is computationally expensive, and therefore finanically infeasible for researchers outside of large corporations. Models often need more than 100 million parameters to achieve state-of-the-art (SOTA) performance, and training those models requires a lot of computational resources, e.g. GPUs, time and data. For example, VLMo, one of the best models across vision-langauge tasks as of this writing, has 562 Million parameters, was trained on 14 million images [1] and cost over 28 thousand USD to train[1].

One strategy to avoid high computational costs is transfer learning. Here, a, potentially large, pretrained model is used as a starting point, and finetuned on a specific task, for a potential different use case. The disadvantage of this approach is that the model size does not change, so finetuning is still computationally expensive, especially for large models. A viable strategy would be to use few layers from the pretrained model, but since the environment in which those layers were trained is different from the one in which they are used during finetuning, this approach requires longer training times.

Another option is Knowledge Distillation (KD). Here, a smaller model, the student model, is trained to replicate, or rather predict the outputs of a larger model, the teacher model, for a given sample. KD can be applied for both supervised and self-supervised settings, meaning the teacher model can be trained in a supervised or self-supervised manner, respectively. The former is referred to as response-based KD, the latter as feature-based KD [2]. Both will be used in this work.

Knowledge-Distillation has the advantage that the student model can be much smaller, and have a different architecture, compared to the teacher model. Since the teacher is running in inference mode, no backpropagation is needed, and thus no gradients have to be computed. This makes KD faster and requires less memory compared to finetuning. Most importantly, it has been empirically shown that student models much smaller than their teachers can achieve similar performance. For example, the distilled model of BERT, DistilBERT, reduced the model size by 40%, while maintaining 97% of the performance of the original model [3].

Response-based Knowledge Distillation

In response-based KD, the teacher was trained in a supervised manner, and provides a probability distribution, or just logits, for a given sample, which is the prediction of the teacher. The student model tries to replicate this probability distribution. This is also called soft targets, because the probability distribution is, unless the teacher is 100% sure, not one-hot encoded, but rather a smooth distribution over the classes. This smooth distribution is further smoothed by dividing the logits by a temperature parameter before applying the softmax function. This increases the relative importance of logits with lower values, e.g. the classes with the second and third highest logits, and Hinton et al. [4] argue that this makes the model learn hidden encoded information the teacher model has learned, which are not represented when focusing on just the class with the highest logit/probability. This helps the student model to generalize better, especially on less data, compared to a model trained from scratch [4], [2]. The temperature parameter is usually a tuneable hyperparameter, but research has shown that it can also be learned parameter, especially in other settings such as contrastive learning [1].

The loss function used in response-based KD is the Kullback-Leibler divergence (KL), which measures the difference between two probability distributions. The mathematical formulation is as follows: Let f be the teacher model, g the student model, and x the input sample, for example an

[1]Calculation done based on the price per GPU hour of the NVIDIA V100 16GB GPUs on AWS for instance p3.2xlarge, as of August 2024. VLMo was trained using NVIDIA V100 32GB GPUs [1], so the actual cost is likely higher.

image. We define $\boldsymbol{u} = g(\boldsymbol{x})$ and $\boldsymbol{z} = f(\boldsymbol{x})$ as the output of the teacher and student model, respectively. Those are the logits, and for a classification task of e.g. 1000 classes, vectores of length 1000. Logits are, optionally, divided by a temperature T, and normalized using softmax.

$$p_i = \frac{\exp\left(\frac{u_i}{T}\right)}{\sum_j \exp\left(\frac{u_j}{T}\right)} \tag{1}$$

$$q_i = \frac{\exp\left(\frac{z_i}{T}\right)}{\sum_j \exp\left(\frac{z_j}{T}\right)} \tag{2}$$

i and j denote indices of the classes, and p_i and q_i the probabilities of class i according to the teacher g and student model f, respectively. The goal is to minimize the difference between the probabilities over all classes, computed by the KL divergence:

$$\mathcal{L}_{\text{KD}} = D_{\text{KL}}(\boldsymbol{p} \parallel \boldsymbol{q}) = \sum_j p_j \log\left(\frac{p_j}{q_j}\right) \tag{3}$$

As in Equation 1 and Equation 2, 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 [5].

Feature-based Knowledge Distillation

In feature-based KD, the teacher model could either be trained in a supervised or self-supervised manner. The student model tries to replicate the (intermediate) activations of the teacher model, so not necessarily only the output of the teacher model, although this is also possible. If the teacher model has been trained in a self-supervised manner, feature-based KD is needed, as the teacher model does not provide a probability distribution to regress. This fact is important for the experiments in this work, as we will build work on an end-to-end self-supervised approach for distilling vision-language models.

For regressing the activations of the teacher model, the Mean Squared Error (MSE) is used as loss function, as well as the Mean Absolute Error (MAE) can be used as a criterion, although the latter is less common [2], [6], [7]. We define the MSE as follows:

$$\mathcal{L}_{KD} = \text{MSE}(\boldsymbol{p}, \boldsymbol{q}) = \|\boldsymbol{p} - \boldsymbol{q}\|_2 = \frac{1}{k}\sum_{j=0}^{k}\left(p_j - q_j\right)^2 \tag{4}$$

It is important to note that for feature-based KD in Transformer models this approach requires the student model to have the same hidden size as the teacher model. Otherwise, additional postprocessing steps are required, e.g. a linear projection layer, to align the student hidden size with the teacher hidden size. An illustration of response-based vs. feature-based KD is shown in Figure 1.

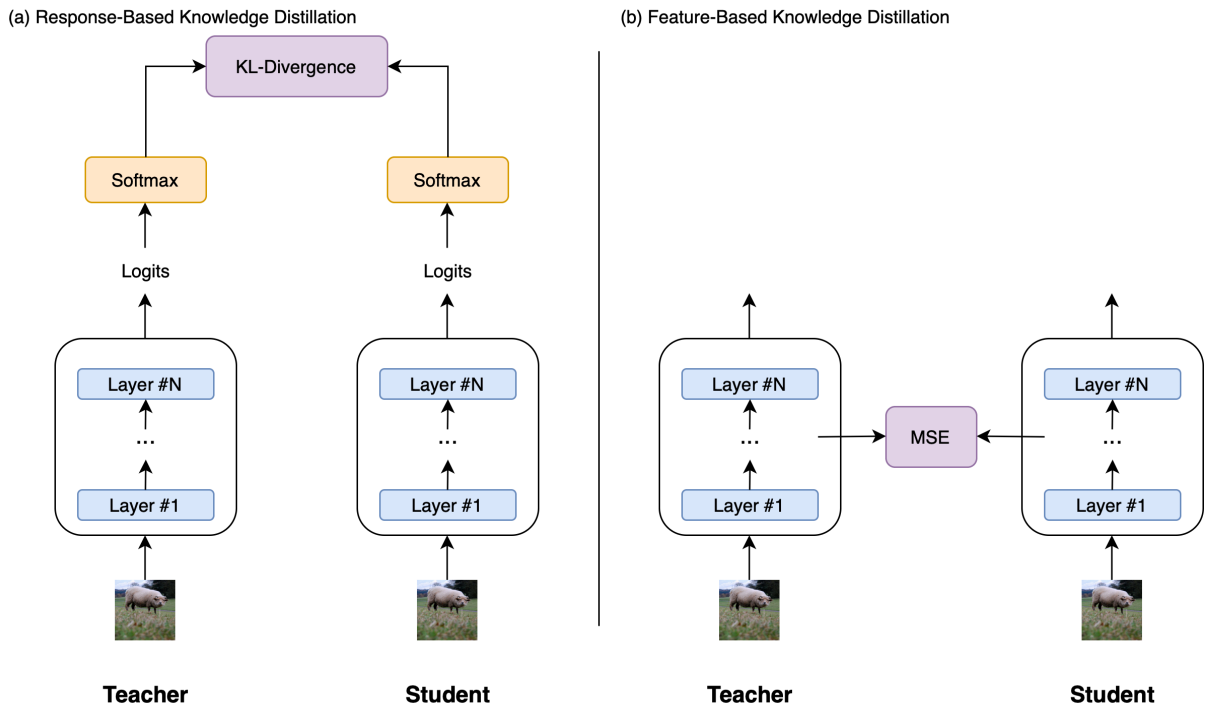

Figure 1: Response-Based Knowledge Distillation (a) requires a supervised teacher to provide logits, from which the probability distribution can be regressed. Feature-Based Knowledge Distillation (b) can be used with when the teacher model has been trained self-supervised. Which teacher activations are regressed by which part of the student model is not fixed, and can be adjusted to the specific use case. An intuitive choice is to regress the activations of the teacher's last layer with the student's last layer. Feature-Based Knowledge Distillation can also be applied on a supervised teacher. In both cases the weights of the teacher are frozen and the teacher is running in evaluation mode. Figure adapted and inspired by [2], image is taken from COCO train set [8].

Bibliography

[1] H. Bao *et al.*, "VLMo: Unified Vision-Language Pre-Training with Mixture-of-Modality-Experts," in *Advances in Neural Information Processing Systems*, 2022. [Online]. Available: https://openreview.net/forum?id=bydKs84JEyw

[2] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge Distillation: A Survey," *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1789–1819, 2021, doi: 10.1007/s11263-021-01453-z.

[3] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," in *NeurIPS EMC^2 Workshop*, 2019.

[4] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network." [Online]. Available: https://arxiv.org/abs/1503.02531

[5] Y. Aytar, C. Vondrick, and A. Torralba, "See, Hear, and Read: Deep Aligned Representations," *arXiv preprint arXiv:1706.00932*, 2017, [Online]. Available: https://arxiv.org/abs/1706.00932

[6] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli, "data2vec: A general framework for self-supervised learning in speech, vision and language," *arXiv abs/2202.03555*, 2022.

[7] A. Baevski, A. Babu, W.-N. Hsu, and M. Auli, "Efficient Self-supervised Learning with Contextualized Target Representations for Vision, Speech and Language." 2022.

[8] T.-Y. Lin *et al.*, “Microsoft COCO: Common Objects in Context,” in *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, D. J. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., in Lecture Notes in Computer Science, vol. 8693. Springer, 2014, pp. 740–755.