0.1.1 Deep Aligned Representations

The motivation for the knowledge distillation approach in this work is provided by the paper "See, Hear, and Read: Deep Aligned Representations" by Aytar et al. (2017) [1]. For simplicity, this paper will be referred to as "SHRe" (for **S**ee, **H**ear, **Re**ad) in this work.

In SHRe, the authors propose a method to align representations of image, text, and audio through Knowledge Distillation from a supervised image model. The student model is a multimodal model with separate modality-specific encoders for image, text, and audio, and a shared encoder on top. The approach utilizes 1D convolutions for audio and text, and 2D convolutions for images. The output feature maps of these encoders are flattened and then passed separately through a shared encoder, which is a 2-layer MLP [1]. However, the approach is independent of the architecture of the components (encoders), meaning that any architecture can be used.

The teacher model was trained in a supervised manner, with the authors utilizing a model pretrained on ImageNet-1K, though it is not specified what exact model used. The approach works by minimizing the KL-Divergence between the teacher and student models. Specifically, the method involves using image-text $\{\boldsymbol{x}^v, \boldsymbol{x}^w\}$ and image-audio $\{\boldsymbol{x}^v, \boldsymbol{x}^a\}$ pairs. Recall from the introduction of knowledge distillation in (TODO: cite knowledge distillation section) that since the teacher model was trained in a supervised manner, it provides a probability distribution over a set of classes for a given sample, which can be predicted by the student model. This is response-based knowledge distillation.

For each pair, the image $\boldsymbol{x}^v$ is passed through the teacher model g, producing a probability distribution over the ImageNet-1K classes (1000 classes), denoted as $g(\boldsymbol{x}^v)$. The same image $\boldsymbol{x}^v$ is also passed through the image encoder of the student model, followed by the shared encoder, also resulting in a probability distribution over the ImageNet-1K classes, defined as $f_v(\boldsymbol{x}^v)$, where f represents the student model and f_v that the image encoder is being used.

The other part of the pair, for example, the text $\boldsymbol{x}^w$ in an image-text pair, is passed through the text encoder of the student model and then through the shared encoder. The output is, again, a probability distribution over the ImageNet-1K classes, represented as $f_w(\boldsymbol{x}^w)$, where f_w indicates that the text encoder is used.

The probability distribution generated by the teacher model for the image can be compared with the probability distribution produced by the student model for the same image using KL-Divergence. Most importantly, however, the probability distribution of the teacher model for the image can be compared with the probability distribution of the student model for the text. For a single image-text pair, the loss is defined as:

$$\mathcal{L}_{\text{KD}} = \frac{1}{2} * (D_{\text{KL}}(g(\boldsymbol{x}^v) \parallel f_v(\boldsymbol{x}^v)) + D_{\text{KL}}(g(\boldsymbol{x}^v) \parallel f_w(\boldsymbol{x}^w))) \tag{1}$$

With D_{KL} being the KL-divergence:

$$D_{\text{KL}}(P \parallel Q) = \sum_i P_i * \log\left(\frac{P_i}{Q_i}\right) \tag{2}$$

Here, j denotes the index of the classes of ImageNet-1K. The loss changes accordingly for image-audio pairs, where the probability distribution over audio is defined as $f_a(\boldsymbol{x}^a)$.

The goal of this approach is to make the probability distributions between teacher and student as similar as possible. Since an image and its corresponding text in an image-text pair describe the same real-world concept, the model can learn to output the same probabilities over the ImageNet-1K classes for both the image and the text. This enables the alignment of modalities at the level of real-

world objects. The same process can be applied to image-audio pairs, allowing the model to align representations across multiple modalities. A visualization of this will be shown when we reproduce this approach in (TODO: cite reproduce section).

Even though all modalities share the same shared encoder, the output of the intermediate layer in the shared encoder will still differ for each modality. This is because KL-Divergence only ensures alignment at the level of classes, so of the output layer, not at the level of internal representations for different modalities in the shared encoder. These internal representations can still vary, as long as the resulting probability distributions are the same.

However, the shared encoder is meant to have the same activations for e.g. an image and text of a positive image-text pair. For tasks such as retrieval ((TODO: cite retrieval)), it further is crucial to have the same output or, more specifically, a similar representation for pairs (e.g., image-text and image-audio). To achieve this, the authors add a ranking loss to the training, which functions similarly to contrastive learning. This ranking loss drives the representations of inputs from the same pair closer together, while pushing the representations of inputs from different pairs further apart. The ranking loss is defined as:

$$\mathcal{L}_{\text{Rank}} = \sum_{i}^{B} \sum_{j \neq i} \max\{0, \Delta - \cos(\boldsymbol{x}_i^v, \boldsymbol{x}_i) + \cos\big(\boldsymbol{x}_i^v, \boldsymbol{x}_j\big)\} \tag{3}$$

Here, B represents the batch size, $\boldsymbol{x}_i^v$ is an image, and $\boldsymbol{x}_i$ is the corresponding text or audio, depending if an image-text or image-audio pair is used. j iterates over negative samples in the batch ($j \neq i$).

Different from contrastive loss, the ranking loss uses cosine embedding loss and only considers a pair of samples at a time. The contrastive loss, on the other hand, uses the cross-entropy over the similarity with respect to all negative pairs in the batch. That means softmax-normalization is employed to maximize the similarity with the positive sample while minimizing the similarity with all negative samples.

The final loss is a combination of the KL-Divergence loss and the ranking loss:

$$\mathcal{L}_{\text{SHRe}} = \mathcal{L}_{\text{KD}} + \mathcal{L}_{\text{Rank}} \tag{4}$$

The authors evaluate SHRe on retrieval tasks, and the results show that SHRe performs significantly better than a random baseline. Interestingly, even though the model is only trained on image-text and image-audio pairs, the alignment also generalizes to text-audio pairs, and the model can retrieve text-audio pairs, albeit not as well as between the modalities it was trained on [1]. This indicates that the image modality acts as an anchor between text and audio, enabling the model to align representations between modalities it was not explicitly trained on. The alignment between modalities becomes transitive.

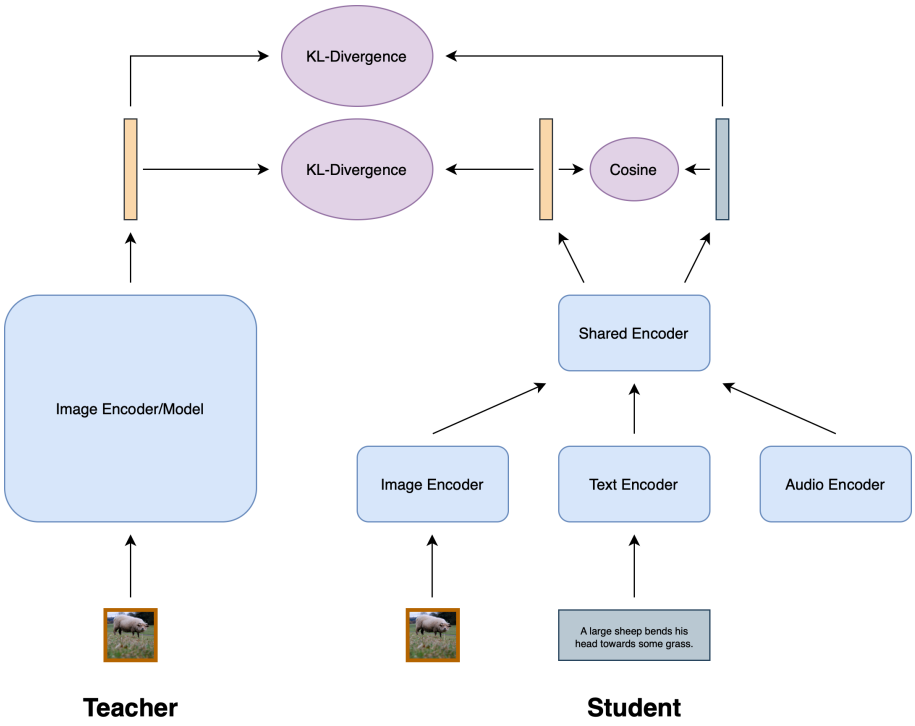

Figure 1: Illustration of the SHRe approach on an image-text pair. The model is trained to output the same probability distribution over ImageNet-1K classes for both the image and the text, while aligning the internal representations using the cosine embedding loss. The output of the models, shown as colored vertical bars, are 1000-dimensional vectors, with each element representing the probability of the input belonging to a specific ImageNet-1K class. For simplicity, only the application of the cosine loss for the (positive) image-text pair is shown. However, cosine loss is also applied to negative pairs in the batch [1]. The figure does not originate from the original paper, but is a custom visualization of the concept. Image-Text pair is taken from the MSCOCO train set [2].

Bibliography

[1] 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

[2] 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.