0.1 Vision-Language Contrast

Introduced as a method for self-supervised learning of image models ((TODO: cite contrastive learning section)), contrastive learning can be extended from unimodal (image) to multimodal applications, such as image and text. As mentioned in the previous section, we aim to maximize the cosine similarity between an image and its corresponding text (i.e., caption), and vice versa. Augmentation is not needed, as we always have pairs: one image and one text. Negative samples for images are captions of other images, and vice versa. In this setting, the model learns to produce similar representations for an image and its caption, describing the same real-world concept, and dissimilar representations for an image and caption that are unrelated. A conceptual example for both vision and vision-language contrastive learning can be seen in Figure 1.

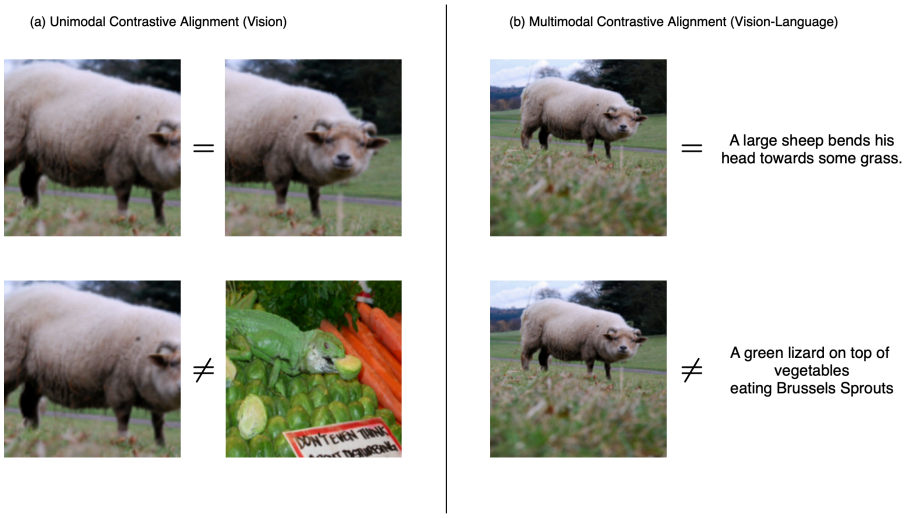

Figure 1: Contrastive learning aims to align the same (or similar) real-world concepts in representation space, while pushing different concepts apart. Multimodal contrastive learning (b) requires existing pairs, e.g. image-text, while for the unimodal case (a) pairs are synthetically created by augmenting the input. Images and text in the figure have been taken from the COCO train set [1].

Contrastive learning requires a (global) representation of the input, which is then used to compare it with other inputs. Since the introduction of the vision Transformer in 2020 by Dosovitskiy et al. [2], most vision-language models are exclusively based on the Transformer architecture, which is why the `[CLS]` token is used as the global representation for both image (`[I_CLS]`) and text (`[T_CLS]`), respectively. There have been other approaches, such as Cross-Modal Late Interaction introduced in FLILP [3], but they usually require significantly more compute [3] and do not outperform global contrastive learning [4], which is what we use here.

The representations are generated by passing the image sequence $\boldsymbol{H}_{v,0}$ and text sequence $\boldsymbol{H}_{w,0}$ through the vision-language model f, and extracting the representations for both tokens ($\boldsymbol{h}_{v,L,\texttt{[I\_CLS]}}$ and $\boldsymbol{h}_{w,L,\texttt{[T\_CLS]}}$) from the output of the final layer $\boldsymbol{H}_{v,L}$ and $\boldsymbol{H}_{w,L}$, which is the output of the Transformer. For the resulting batch of image and text representations $\left\{\boldsymbol{h}_{(v,L,\texttt{[I\_CLS]}),k}, \boldsymbol{h}_{(w,L,\texttt{[T\_CLS]}),k}\right\}_{k=1}^{B}$, where B is the batch size, the cosine similarity between all possible image-text pairs is computed. The cosine similarity is given by:

$$\cos(\boldsymbol{a}, \boldsymbol{b}) = \frac{\boldsymbol{a}\boldsymbol{b}^T}{\|\boldsymbol{a}\|_2 * \|\boldsymbol{b}\|_2} = \frac{\boldsymbol{a}}{\|\boldsymbol{a}\|_2}\frac{\boldsymbol{b}^T}{\|\boldsymbol{b}\|_2} \tag{1}$$

$\boldsymbol{a}\boldsymbol{b}^T$ denotes the simple dot product between both representations. $\|\boldsymbol{a}\|_2$ and $\|\boldsymbol{b}\|_2$ denote the L2-norm of the representations.

The cosine similarity between all possible image-text pairs can be computed efficiently by organizing all image and text representations in a matrix, which is already given in a batch-wise training, and normalizing every representation.

$$\boldsymbol{h}' = \frac{\boldsymbol{h}}{\|\boldsymbol{h}\|_2} \tag{2}$$

$$\boldsymbol{I} = \left[\boldsymbol{h}'_{(v,L,[\texttt{I\_CLS}]),1}, \boldsymbol{h}'_{(v,L,[\texttt{I\_CLS}]),2}, ..., \boldsymbol{h}'_{(v,L,[\texttt{I\_CLS}]),B}\right] \in \mathbb{R}^{B \times D} \tag{3}$$

$$\boldsymbol{T} = \left[\boldsymbol{h}'_{(w,L,[\texttt{T\_CLS}]),1}, \boldsymbol{h}'_{(w,L,[\texttt{T\_CLS}]),2}, ..., \boldsymbol{h}'_{(w,L,[\texttt{T\_CLS}]),B}\right] \in \mathbb{R}^{B \times D} \tag{4}$$

I denotes the batch/matrix of image representations, and T contains the text representations. D is the dimensionality of the representations, often referred to as the hidden size or hidden dimension in Transformers.

A matrix multiplication of both batches of representations then computes the dot product between every image with every text, and vice versa. Since the representations are normalized, the result will be the cosine similarity between all possible image-text pairs in the batch.

$$\boldsymbol{L} = \boldsymbol{I}\boldsymbol{T}^T, \boldsymbol{L} \in \mathbb{R}^{B \times B} \tag{5}$$

$\boldsymbol{L}_{i,j}$ then denotes the similarity between image i and text j in the batch. The diagonal of the matrix contains the similarity between positive pairs, i.e., the correct image-text pairs (i, i), with $\boldsymbol{L}_{i,i}$ describing their similarity. For an image, all other texts in the batch are considered as negative samples, and vice versa for text. The superscript T denotes the transpose of a matrix, and is not to be confused with the batch of text representations $\boldsymbol{T}$.

For a batch size of 256 ($B = 256$), each image has 255 negative samples (i.e., captions of other images) and one positive sample (i.e., its own caption), the same holds vice versa. This can be seen as a classification problem with 256 classes, where the model has to predict the correct class out of 256 classes, and each class representing one caption or image, respectively. For an image, the logit for the correct class is the similarity (cosine) to its own caption, and the logits for the negative classes are the similarities to the captions of other images. The same holds vice versa for text.

To calculate the loss, the cross-entropy loss is used. For a batch, the loss for selecting the correct caption for each image is given by:

$$\mathcal{L}_{\text{CL}}^{\text{i2t}} = \frac{1}{B}\sum_{i=1}^{B} -\log \frac{\exp(\boldsymbol{L}_{i,i})}{\sum_{k=1}^{B}\exp(\boldsymbol{L}_{i,k})} \tag{6}$$

$\frac{\exp(\boldsymbol{L}_{i,i})}{\sum_{k=1}^{B}\exp(\boldsymbol{L}_{i,k})}$ denotes the softmax-normalized similarity between an image and its correct caption, which is the usual way for calculating the cross-entropy. The result of this normalization is a probability distribution for each image, where each caption in the batch has a probability of being the correct caption for the image, and vice versa. The probability that the correct caption belongs to the current image is then used to calculate the negative log-likelihood, which is the loss.

Accordingly, the loss for selecting the correct image for each caption is given by:

$$\mathcal{L}_{\text{CL}}^{\text{t2i}} = \frac{1}{B}\sum_{i=1}^{B} -\log \frac{\exp(\boldsymbol{L}_{i,i})}{\sum_{k=1}^{B}\exp(\boldsymbol{L}_{k,i})} \tag{7}$$

Here, the softmax-normalization is with respect to the similarity of a text with all other images in the batch. The final loss is the mean of the image-to-text and text-to-image loss:

$$\mathcal{L}_{\text{CL}} = \frac{1}{2} * \left(\mathcal{L}_{\text{CL}}^{\text{i2t}} + \mathcal{L}_{\text{CL}}^{\text{t2i}}\right) \tag{8}$$

Returning to the concept of contrastive learning, this process ensures that the similarity between the representation of an image and its caption is maximized, i.e. close to each other, while the similarity between an image and an unrelated caption is minimized, i.e. far apart. Only this would appropriately minimize the loss, and thus the model learns to align the representations of the same concept across modalities. An illustration of multimodal contrastive learning can be found in Figure 2.

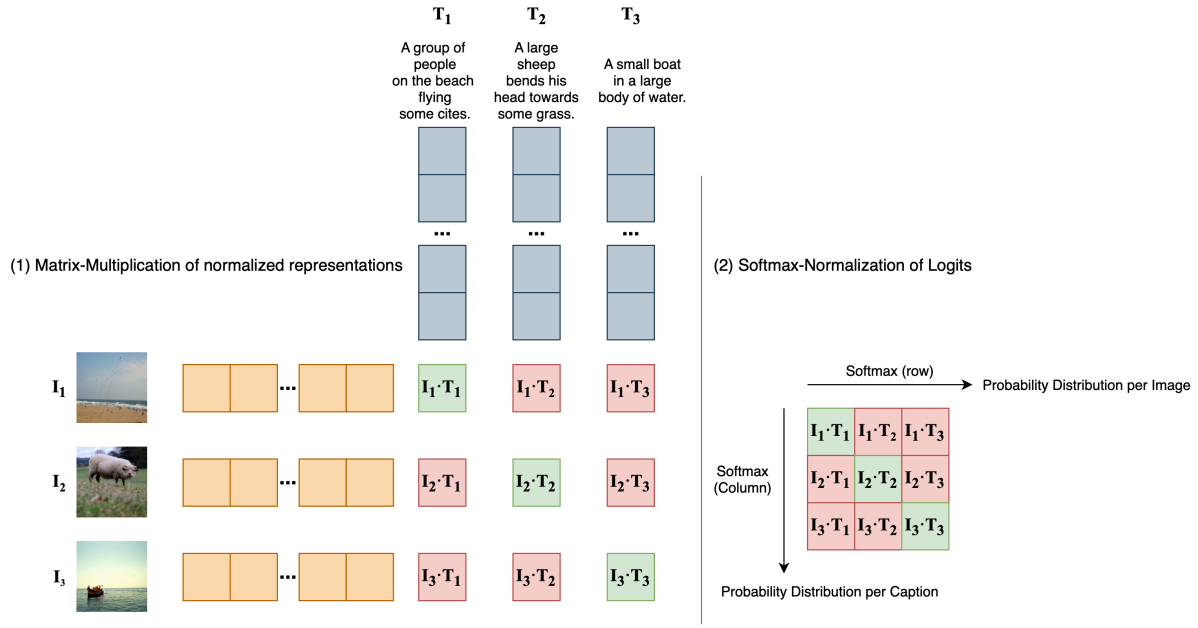

Figure 2: Contrastive Learning is performed using matrix multiplication of normalized representations (1), and the result is matrix $\boldsymbol{L}$ described in Equation 5. The representations are given by the [CLS] token of the respective modality, but are represented as I and T in the figure for simplicity. The diagonal of the resulting matrix contains the cosine similarity between positive samples. The softmax operation along the rows yields a probabilty distribution for each image over all captions, and the softmax operation along the columns vice versa (2). The cross-entropy loss is then used to calculate the loss for the distributions. The final loss is the mean of both losses. Image-Text pairs in the figure have been taken from the COCO train set [1].

The performance of contrastive learning is highly dependent on the number of negative samples available, which directly translates to the batch size. For instance, with a batch size of two, the model only needs to differentiate between one caption that belongs to the image and one that does not (a negative sample), and vice versa. This task is significantly simpler than with 255 negative samples or more, where there might be captions that are semantically similar to the image, but do not belong to it. So with increased negative samples, to probability of encountering hard-negative examples increases, forcing the model to aggregate as much information as possible in [I_CLS] and [T_CLS] to even differentiate between semantically similar concepts.

The results improve with an increased number of negative examples [5], [4], which we will also show later, in the experiments section. More negative samples are usually achieved by using larger batch sizes [5], [6], [4]. However, this typically requires higher VRAM GPUs, or multiple GPUs, which is costly.

Bibliography

[1] 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.

[2] A. Dosovitskiy *et al.*, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” *ICLR*, 2021.

[3] L. Yao *et al.*, “FILIP: Fine-grained Interactive Language-Image Pre-Training,” *CoRR*, 2021.

[4] W. Wang *et al.*, “Image as a Foreign Language: BEIT Pretraining for Vision and Vision-Language Tasks,” in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 19175–19186. doi: 10.1109/CVPR52729.2023.01838.

[5] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum Contrast for Unsupervised Visual Representation Learning,” *arXiv preprint arXiv:1911.05722*, 2019.

[6] A. Radford *et al.*, “Learning transferable visual models from natural language supervision,” in *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, M. Meila and T. Zhang, Eds., in Proceedings of Machine Learning Research, vol. 139. PMLR, 2021, pp. 8748–8763.