0.1 Contrastive Learning

0.1.1 Vision

In settings where masking discrete tokens and predicting them based on a set of possible tokens, as in language models, is not possible, contrastive learning can be used as a self-supervised method. This is especially useful in vision models, as images are continuous, so there is no discrete set of possible tokens to predict.

Contrastive learning, or the contrastive loss, is a method to learn representations of data without the need for labels, and used in computer vision models like MoCo [1], SimCLR [2], and CLIP [3].

In computer vision, contrastive learning exploits the fact that the high-level semantics of an image are invariant to small (or moderate) changes in pixel-level information. This is achieved by augmenting the input image, e.g., by cropping, rotating, or flipping it. Provided the augmentation is not too drastic (e.g., crop size too large), the high-level semantics of the image will remain the same after augmentation, even though pixel-level information do not. The goal of the image model is then to maximize the cosine similarity between the global representations of two augmented versions of the same image. In Transformers, the global representation is usually the [`CLS`] token retuned by the final layer of the model, which is a vector that can be compared with the [`CLS`] token of another image using the cosine similarity. The augmented versions are often referred to as a different *view* of the same image [4], as shown in Figure 1.

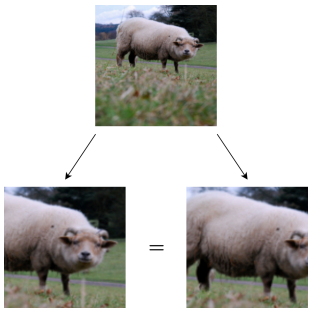

Figure 1: Adding small translations to an image, e.g. a random crop, as illustrated in the figure, will retrain high-level semantic features while changing pixel-level information. The content of the image stays the same, and the same should therefore hold for the representations produced by the model. Image in the figure has been taken from the COCO train set [5].

However, this alone is not sufficient, as the model will collapse to a trivial solution by simply returning the same representation for all inputs, as demonstrated in the papers MoCo [1] and SimSiam [4]. Producing the same representation for all inputs is the simplest way to maximize the cosine similarity between the original image and its augmented versions, because the representation produced for an image would always be the same, therefore maximizing the cosine similarity (a value of 1). To prevent this, negative samples are introduced. Negative samples are other images that do not contain the same content as the original image, and the cosine similarity between the original image and these negative samples should therefore be minimized (a cosine similarity of 0 indicates no similarity between the input vectors). This prevents the model from collapsing to a constant representation, as it would not minimize the cosine similarity and thus not minimize the loss. A simple yet expressive visualization can be found in [6]. This makes self-supervised training of image models possible, and the learned representations represent the high-level semantics of the images, learned without the need for labels.

An implementation and mathematical formulation of the contrastive loss will be introduced in (TODO: cite vision language contrast) on the example of vision-language models.

Bibliography

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

[2] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A Simple Framework for Contrastive Learning of Visual Representations," *arXiv preprint arXiv:2002.05709*, 2020.

[3] 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.

[4] X. Chen and K. He, "Exploring Simple Siamese Representation Learning," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 15745–15753. doi: 10.1109/CVPR46437.2021.01549.

[5] 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.

[6] T. Chen and G. Hinton, "Advancing Self-Supervised and Semi-Supervised Learning with SimCLR." [Online]. Available: https://research.google/blog/advancing-self-supervised-and-semi-supervised-learning-with-simclr/

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

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

[9] 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.

[10] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 67–78, Feb. 2014.

[11] A. Singh *et al.*, "FLAVA: A foundational language and vision alignment model," *CoRR*, 2021, [Online]. Available: https://arxiv.org/abs/2112.04482

[12] 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

0.1.2 Vision-Language

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 2.

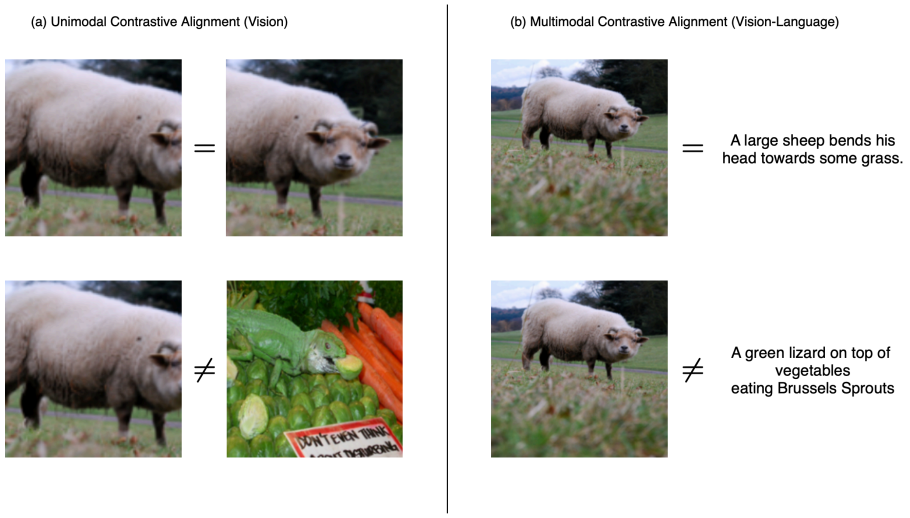

Figure 2: Contrastive learning aims to align the same (or similar) real-world concepts in representation space, while pushing different concepts apart. Multimodal contrastive learning (b) requires existing pairs, e.g. image-text, while for the unimodal case (a) pairs are synthetically created by augmenting the input. Images and text in the figure have been taken from the COCO train set [5].

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. [7], 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 [8], but they usually require significantly more compute [8] and do not outperform global contrastive learning [9], 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 3.

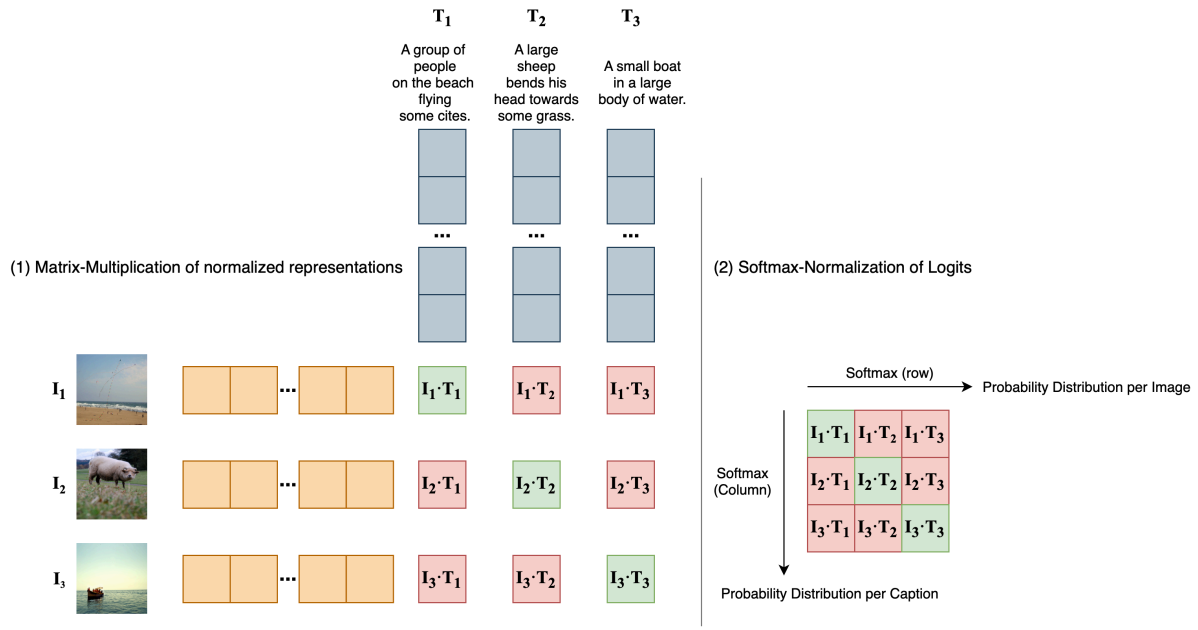

Figure 3: Contrastive Learning is performed using matrix multiplication of normalized representations (1), and the result is matrix $\boldsymbol{L}$ described in Equation 5. The representations are given by the [CLS] token of the respective modality, but are represented as I and T in the figure for simplicity. 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 [5].

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 [1], [9], which we will also show later, in the experiments section. More negative samples are usually achieved by using larger batch sizes [1], [3], [9]. However, this typically requires higher VRAM GPUs, or multiple GPUs, which is costly.

0.1.3 Vision-Language Retrieval

The goal of image-text retrieval (ITR) is to find the matching caption for a given image in a set of captions, and likewise, finding the matching image for a given caption in a set of images. The process begins with embedding and normalizing a set of samples, which become a set of keys. For some normalized candidate representation, called the query, the most similar key is retrieved among the set of keys is the retrieved sample. This is exactly what is learned through contrastive learning, where we try to maximize the similarity between an image or caption (query) and its paired caption or image among other samples (keys), respectively. For that, we can use the same batch-wise computation introduced in the previous section about the contrastive loss. The similarity is

computed by the cosine similarity, which is, again, computed by matrix multiplication of the normalized embeddings.

Image-Text retrieval can be viewed as a form of semantic search, which has significant practical relevance in areas like recommendation systems, e.g. to find fitting images based on a given text query. This is precisely what is learned through multimodal contrastive learning.

Image-Text retrieval is a simple and efficient way to benchmark the quality of the learned representations of a vision-language model, as it does not require any finetuning, just the embeddings produced by the model. The metric used for benchmarking is Rank@K (R@K), where K determines at which rank the paired/correct sample has to be in the ranking of keys, in order for the retrieval to be considered correct. We use R@1, R@5, and R@10, where R@1 is the normal accuracy, i.e., the paired sample has to be the most similar one. R@5 means that the paired sample has to be in the top 5 most similar samples, and for R@10, it has to be in the top 10 most similar samples.

In this thesis, we use the 5K test set of MSCOCO [5], and the 1K test set of Flickr30k [10] for benchmarking, which are the standard benchmarking dataset for multimodal models like FLAVA [11], CLIP [3], VLMo [12], and BEiT-3 [9]. MSCOCO contains 5K images with 5 captions for each image [5], and Flickr30k contains 1K images with 5 captions each [10]. For both datasets, all images and all texts are embedded and normalized, so that each image and each text is represented by the respective [`CLS`] token returned by the model. Then, matrix multiplication between all images and all captions of a dataset is performed, resulting in a matrix of shape (N, M), where N is the number of images and M is the number of captions in the dataset. So for MSCOCO, the matrix is of shape (5K, 25K), and for Flickr30k, the matrix is of shape (1K, 5K).

For each image, R@1, R@5, and R@10 are computed. The mean of R@1, R@5, and R@10 over all images are then called text-retrieval of the respective metrics (e.g. R@1-text-retrieval). We call this text-retrieval, because we are trying to retrieve the correct caption for a given image. The same is done for each caption, resulting in image-retrieval of the respective metrics (e.g. R@1-image-retrieval). For each dataset, we have 6 metrics in total: R@1, R@5, and R@10 for text-retrieval and image-retrieval, respectively. We will report the results of image-text retrieval in the format seen in Table 1.

Model	**MSCOCO (5K test set)**						**Flickr30K (1K test set)**					
	Image → Text			Text → Image			Image → Text			Text → Image		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
FLAVA [11]	42.74	76.76	-	38.38	67.47	-	67.7	94.0	-	65.22	89.38	-
CLIP [3]	58.4	81.5	88.1	37.8	62.4	72.2	88.0	98.7	99.4	68.7	90.6	95.2
BEiT-3 [9]	84.8	96.5	98.3	67.2	87.7	92.8	98.0	100.0	100.0	90.3	98.7	99.5

Table 1: Benchmarks of different vision-language models on the MSCOCO and Flickr30K datasets for image-text retrieval.