Contrastive Learning

- is used to compare samples (e.g. images) with each other in representation space, typically by some distance metric, of which cosine similarity is the usual choice
- used in self-supervised learning to learn representations without classical labels such as class targets
- goal is to learn (abstract) representation of the input modality, e.g. images
- originally used in computer vision
  - idea is, that a representation of one image should be similar, or very close, to augmented versions of the same image
  - after all, content of the image stays the same after augmentation (provided the augmentation is not too drastic, e.g. crop size too big)
- goal of the (image) model is to maximize the cosine similary between the original image and its augmented versions
- this alone not sufficient, as model will collapse to a trivial solution, by simply return the same representation for all inputs
  - will maximize the cosine similarity between between the original image and its augmented versions, as representation produced for an image will always be the same
- to prevent this, negative samples are introduced
  - negative samples are other images, so not the original image
  - (usually) does not contain the same content as the original image, so cosine similarity between the original image should be minimized
- that way, model can't collapse to a constant representation, as this would not minimize the cosine similarity, and thus not minimize the loss
- this can be extended from unimodal to multimodal applications, in our case: images and text
- here we would like to maximize the cosine similarity between an image and its corresponding text, i.e. caption, and vice versa
- we do not need any augmentation, as we always have pairs: one image and one text
- negative samples for images are captions of other images, and vice versa
- model learns to produce similar representations for an image and its caption, describing the same real-world concept

Implementation:

- staying at our multimodal case, contrastive learning/loss is usually done on the batch-level
- means the multimodal model creates representations for all images and captions in the batch
- then, the cosine similarity between, the representations, of all images and captions in the batch is computed
  - can be done efficiently by normalizing each embedding and then perform matrix multiplication
- for a batch size of e.g. 256, each image has 255 negative samples, i.e. captions of other images, and one positive sample, i.e. its own caption, and vice versa
- can be interpreted as a classification problem with 256 classes, where the model has to predict the correct class, i.e. the positive sample, out of 256 classes/representations
- cross-entropy can then be used as the loss function, metric is accuracy

Problem:

- result highly dependend on the amount of negative samples that are available
  - as an example, if batch size would be two, then the model would have to differentiate between one caption that belongs to the image and one that does not (negative sample), and vice versa
  - a lot simpler than with 255 negative samples, or even more
- result will be better with more negative examples, as task more challenging

- more negative samples can be achieved by using larger batch sizes, but this usually require, depending on the model architecture, higher VRAM GPUs or even multiple GPUs
  - costly

Retrieval