Title: Contrastive Language-Image Pre-training

URL: https://en.wikipedia.org/wiki/Contrastive_Language-Image_Pre-training

PageID: 67219182

Categories: Category:Artificial neural networks, Category:Computer vision, Category:Machine learning, Category:Natural language processing

---

Contrastive Language-Image Pre-training (CLIP) is a technique for training a pair of neural network models, one for image understanding and one for text understanding, using a contrastive objective. This method has enabled broad applications across multiple domains, including cross-modal retrieval, text-to-image generation, and aesthetic ranking.

Algorithm

The CLIP method trains a pair of models contrastively. One model takes in a piece of text as input and outputs a single vector representing its semantic content. The other model takes in an image and similarly outputs a single vector representing its visual content. The models are trained so that the vectors corresponding to semantically similar text-image pairs are close together in the shared vector space, while those corresponding to dissimilar pairs are far apart.

To train a pair of CLIP models, one would start by preparing a large dataset of image-caption pairs. During training, the models are presented with batches of N {\displaystyle N} image-caption pairs. Let the outputs from the text and image models be respectively v 1 , . . . , v N , w 1 , . . . , w N {\displaystyle v_{1},...,v_{N},w_{1},...,w_{N}} . Two vectors are considered "similar" if their dot product is large.

The loss incurred on this batch is the multi-class N-pair loss, which is a symmetric cross-entropy loss over similarity scores: – 1 N ∑ i ln ■ e v i · w i / T ∑ j e v i · w j / T – 1 N ∑ j ln ■ e v j · w j / T ∑ i e v i · w j / T {\displaystyle -{\frac {1}{N}}\sum _{i}\ln {\frac {e^{v_{i}\cdot w_{i}/T}}{\sum _{j}e^{v_{i}\cdot w_{j}/T}}}-{\frac {1}{N}}\sum _{j}\ln {\frac {e^{v_{j}\cdot w_{j}/T}}{\sum _{i}e^{v_{i}\cdot w_{j}/T}}}} In essence, this loss function encourages the dot product between matching image and text vectors (v i · w i {\displaystyle v_{i}\cdot w_{i}}) to be high, while discouraging high dot products between non-matching pairs. The parameter T > 0 {\displaystyle T>0} is the temperature , which is parameterized in the original CLIP model as T = e – τ {\displaystyle T=e^{-\tau }} where τ ∈ R {\displaystyle \tau \in \mathbb {R} } is a learned parameter.

Other loss functions are possible. For example, Sigmoid CLIP (SigLIP) proposes the following loss function: L = 1 N ∑ i , j ∈ 1 : N f ((2 δ i , j – 1) (e τ w i · v j + b)) {\displaystyle L={\frac {1}{N}}\sum _{i,j\in 1:N}f((2\delta _{i,j}-1)(e^{\tau }w_{i}\cdot v_{j}+b))} where f (x) = ln ■ (1 + e – x) {\displaystyle f(x)=\ln(1+e^{-x})} is the negative log sigmoid loss, and the Dirac delta symbol δ i , j {\displaystyle \delta _{i,j}} is 1 if i = j {\displaystyle i=j} else 0.

CLIP models

While the original model was developed by OpenAI , subsequent models have been trained by other organizations as well.

Image model

The image encoding models used in CLIP are typically vision transformers (ViT) . The naming convention for these models often reflects the specific ViT architecture used. For instance, "ViT-L/14" means a "vision transformer large" (compared to other models in the same series) with a patch size of 14, meaning that the image is divided into 14-by-14 pixel patches before being processed by the transformer. The size indicator ranges from B, L, H, G (base, large, huge, giant), in that order.

Other than ViT, the image model is typically a convolutional neural network , such as ResNet (in the original series by OpenAI), or ConvNeXt (in the OpenCLIP model series by LAION).

Since the output vectors of the image model and the text model must have exactly the same length, both the image model and the text model have fixed-length vector outputs, which in the original report is called "embedding dimension".

For example, in the original OpenAI model, the ResNet models have embedding dimensions ranging from 512 to 1024, and for the ViTs, from 512 to 768.

Its implementation of ViT was the same as the original one, with one modification: after position embeddings are added to the initial patch embeddings, there is a LayerNorm .

Its implementation of ResNet was the same as the original one, with 3 modifications:

In the start of the CNN (the "stem"), they used three stacked 3x3 convolutions instead of a single 7x7 convolution, as suggested by.

There is an average pooling of stride 2 at the start of each downsampling convolutional layer (they called it rect-2 blur pooling according to the terminology of). This has the effect of blurring images before downsampling, for antialiasing.

The final convolutional layer is followed by a multiheaded attention pooling .

ALIGN a model with similar capabilities, trained by researchers from Google used EfficientNet , a kind of convolutional neural network .

Text model

The text encoding models used in CLIP are typically Transformers .

In the original OpenAI report, they reported using a Transformer (63M-parameter, 12-layer, 512-wide, 8 attention heads) with lower-cased byte pair encoding (BPE) with 49152 vocabulary size. Context length was capped at 76 for efficiency. Like GPT , it was decoder-only, with only causally-masked self-attention. Its architecture is the same as GPT-2 .

Like BERT , the text sequence is bracketed by two special tokens [SOS] and [EOS] ("start of sequence" and "end of sequence"). Take the activations of the highest layer of the transformer on the [EOS] , apply LayerNorm , then a final linear map. This is the text encoding of the input sequence. The final linear map has output dimension equal to the embedding dimension of whatever image encoder it is paired with. These models all had context length 77 and vocabulary size 49408.

ALIGN used BERT of various sizes.

Dataset

WebImageText

The CLIP models released by OpenAI were trained on a dataset called "WebImageText" (WIT) containing 400 million pairs of images and their corresponding captions scraped from the internet. The total number of words in this dataset is similar in scale to the WebText dataset used for training GPT-2 , which contains about 40 gigabytes of text data.

The dataset contains 500,000 text-queries, with up to 20,000 (image, text) pairs per query. The text-queries were generated by starting with all words occurring at least 100 times in English Wikipedia , then extended by bigrams with high mutual information , names of all Wikipedia articles above a certain search volume, and WordNet synsets .

The dataset is private and has not been released to the public, and there is no further information on it.

Data preprocessing

For the CLIP image models, the input images are preprocessed by first dividing each of the R, G, B values of an image by the maximum possible value, so that these values fall between 0 and 1, then subtracting by [0.48145466, 0.4578275, 0.40821073] , and dividing by [0.26862954, 0.26130258, 0.27577711] .

The rationale was that these are the mean and standard deviations of the images in the WebImageText dataset, so this preprocessing step roughly whitens the image tensor. These numbers slightly differ from the standard preprocessing for ImageNet, which uses [0.485, 0.456, 0.406] and [0.229, 0.224, 0.225] .

If the input image does not have the same resolution as the native resolution (224×224 for all except ViT-L/14@336px, which has 336×336 resolution), then the input image is scaled down by bicubic interpolation , so that its shorter side is the same as the native resolution, then the central square of the image is cropped out.

Others

ALIGN used over one billion image-text pairs, obtained by extracting images and their alt-tags from online crawling. The method was described as similar to how the Conceptual Captions dataset was constructed, but instead of complex filtering, they only applied a frequency-based filtering.

Later models trained by other organizations had published datasets. For example, LAION trained OpenCLIP with published datasets LAION-400M, LAION-2B, and DataComp-1B.

Training

In the original OpenAI CLIP report, they reported training 5 ResNet and 3 ViT (ViT-B/32, ViT-B/16, ViT-L/14). Each was trained for 32 epochs. The largest ResNet model took 18 days to train on 592 V100 GPUs. The largest ViT model took 12 days on 256 V100 GPUs.

All ViT models were trained on 224×224 image resolution. The ViT-L/14 was then boosted to 336×336 resolution by FixRes, resulting in a model. They found this was the best-performing model.

In the OpenCLIP series, the ViT-L/14 model was trained on 384 A100 GPUs on the LAION-2B dataset, for 160 epochs for a total of 32B samples seen.

Applications

Cross-modal retrieval

CLIP's cross-modal retrieval enables the alignment of visual and textual data in a shared latent space, allowing users to retrieve images based on text descriptions and vice versa, without the need for explicit image annotations. In text-to-image retrieval , users input descriptive text, and CLIP retrieves images with matching embeddings. In image-to-text retrieval , images are used to find related text content.

CLIP’s ability to connect visual and textual data has found applications in multimedia search, content discovery, and recommendation systems.

Image classification

CLIP can perform zero-shot image classification tasks. This is achieved by prompting the text encoder with class names and selecting the class whose embedding is closest to the image embedding. For example, to classify an image, they compared the embedding of the image with the embedding of the text "A photo of a {class}.", and the {class} that results in the highest dot product is outputted.

CLIP for multimodal learning

CLIP has been used as a component in multimodal learning . For example, during the training of Google DeepMind 's Flamingo (2022), the authors trained a CLIP pair, with BERT as the text encoder and NormalizerFree ResNet F6 as the image encoder. The image encoder of the CLIP pair was taken with parameters frozen and the text encoder was discarded. The frozen image encoder was then combined with a frozen Chinchilla language model , by finetuning with some further parameters that connect the two frozen models.

Applications in other domains

CLIP's image encoder is a pre-trained image featurizer . This can then be fed into other AI models.

Models like Stable Diffusion use CLIP's text encoder to transform text prompts into embeddings for image generation. CLIP can also be used as a gradient signal for directly guiding diffusion ("CLIP guidance") or other generative art.

Fine-tuned CLIP models can be used to rank images by aesthetic quality, which may be useful as a step in filtering a large dataset into a smaller one with higher quality.

CLIP can be used to generate image captions by matching text inputs to image embeddings.

Notes

References

External links

OpenAI's CLIP webpage

OpenCLIP: An open source implementation of CLIP

Arora, Aman (2023-03-11). "The Annotated CLIP (Part-2)" . amaarora.github.io . Retrieved 2024-09-11 .