OpenClip Overview

Ryan Rearden

June 7, 2024

1 Introduction

CLIP stands for Contrastive Language Image Pre-training. It is able to predict relevant text for images, and relevant images for text. OpenCLIP was trained on open source databases including the LAION-5B which is a set 5 billion image-text pairs; around 2.32 billion of those pairs are in English. Predictions estimate a 81.9% zero-shot top-1 accuracy on ImageNet for a ViT-G/14 CLIP model trained on 68B image-text samples [Che+22].

2 Understanding the Code

OpenClip has very few comments and sparse documentation on their specific architecture. The purpose of this section is to analyze the tutorial code more closely in order to understand the background processes.

2.1 Program 1: Matching given text with a given image

```
import torch
from PIL import Image
import open_clip

#create_model_and_transforms:
#takes an image and text model(in this case ViT-B-32) and the pretrained LAION model
#the only thing that is required is the model, by default, pretrained = None
#more details about what other parameters can be added can be found at:
###open_clip/src/open_clip/factory.py Line 360
model, _, preprocess = open_clip.create_model_and_transforms(’ViT-B-32’,
    pretrained=’laion2b_s34b_b79k’)

#Sets model to evaluation mode rather than training mode.
model.eval() # model in train mode by default, impacts some models with BatchNorm or
    stochastic depth active

#get_tokenizer takes in a model, returns the tokenizer
tokenizer = open_clip.get_tokenizer(’ViT-B-32’)


#preprocess converts the image to RGB, makes the image have the correct size, as well as
    many other things
#more information can be found in open_clip/src/open_clip/tansformers.py starting on Line 17
image = preprocess(Image.open("docs/CLIP.png")).unsqueeze(0)

#tokenizer takes a list of strings and converts them into a list of tensors
text = tokenizer(["a diagram", "a dog", "a cat"])

#no_grad() disables gradient calculations
#torch.cuda.amp.autocast() allows for mixed precision (16-bit and 32-bit floating points)
with torch.no_grad(), torch.cuda.amp.autocast():
```

```
#stores a tensor representation of the image
    image_features = model.encode_image(image)
#stores a tensor representation of the text
    text_features = model.encode_text(text)

#printed, the code: image_features.norm(dim=-1, keepdim=True) is ([[1.0000]]
    image_features /= image_features.norm(dim=-1, keepdim=True)
#same with text_features
    text_features /= text_features.norm(dim=-1, keepdim=True)

#matrix multipixation and softmax to determine the most probable text to img
    text_probs = (100.0 * image_features @ text_features.T).softmax(dim=-1)

print("Label probs:", text_probs) # prints: [[1., 0., 0.]] or something to that degree
```

2.2 Program 2: Generating text from a given image

It is also easy to produce a text from an image. The code below was taken from: *docs/Interacting_with_open_coca.ipynb*:

Aside from model.generate, which generates the text from the image, most of the code has the same pieces as above.

```
import open_clip
import torchd

model, _, transform = open_clip.create_model_and_transforms(
  model_name="coca_ViT-B-32",
  pretrained="laion2b_s13b_b90k"
)

from IPython.display import Image
image = Image("data/rocket.png")

from PIL import Image
im = Image.open("data/rocket.png").convert("RGB")
im = transform(im).unsqueeze(0)

with torch.no_grad(), torch.cpu.amp.autocast():
  generated = model.generate(im)

print(open_clip.decode(generated[0]).split("")[0].replace("", ""))
```

3 Things to note

On my machine, I was able to successfully run Program 1. The desktop computer uses an Intel i3 so I was surprised to see how fast the text was able to match to the correct image.

Program 2 ran but never finished. This is due to the size of the models and the lack of GPU power on the desktop computers.

Version 4.39.0 of the package ”transformers” broke open_clips’s way of generating text from a given image. The error message reads: ”RuntimeError: Boolean value of Tensor with more than one value is ambiguous”. To circumvent this error, the user revert back to transformers==4.38.2 [wuj23]

References

[Che+22] Mehdi Cherti et al. *Reproducible scaling laws for contrastive language-image learning.* 2022. arXiv: 2212.07143 [cs.LG].

[wuj23] wujohns. *RuntimeError: Boolean value of Tensor with more than one value is ambiguous.* https://github.com/mlfoundations/open_clip/issues/847. 2023.