CLIP

Marco Willi

Import libraries.

```
from PIL import Image
import requests

from transformers import CLIPProcessor, CLIPModel
```

Specify cache dir to which the models are downloaded.

```
cache_dir="/home/jovyan/work/data/hf_cache"
```

```
model = CLIPModel.from_pretrained("openai/clip-vit-base-patch32", cache_dir=cache_dir)
processor = CLIPProcessor.from_pretrained("openai/clip-vit-base-patch32", cache_dir=cache_di
```

Download an image.

```
url = "http://images.cocodataset.org/val2017/000000039769.jpg"
image = Image.open(requests.get(url, stream=True).raw)
```

```
image
```

Create two prompts and process them along with the image.

```
inputs = processor(text=["a photo of a cat", "a photo of a dog"], images=image, return_tenso
```

Now we create embeddings for the prompts and the image.

```
outputs = model(**inputs)
```

We evaluate the similarities between the text and the image embeddings.

```
logits_per_image = outputs.logits_per_image  # this is the image-text similarity score
probs = logits_per_image.softmax(dim=1)  # we can take the softmax to get the label probabili
```

```
probs
```