Project Report: Using CLIP for CIFAR-10 Image Classification

1. Introduction

In this project, we use the CLIP (Contrastive Language–Image Pretraining) model introduced by Radford et al. (2021). CLIP is a vision-language model that can understand the content of images using textual descriptions.

2. Tools and Resources Used

- **Execution Environment and Resource:** Google Colab (T4 GPU)
- **Model:** OpenCLIP - ViT-B/32
- **Dataset:**CIFAR-10 (PyTorch built-in dataset)

3. Data and Model

3.1 Loading the OpenCLIP Model

```
import open_clip
model,_, preprocess = open_clip.create_model_and_transforms('ViT-B-32', pretrained='openai')
tokenizer = open_clip.get_tokenizer('ViT-B-32')
```

3.2 Preparing Data from Pytorch

```
trainset = torchvision.datasets.CIFAR10(root='./data', train=True, download=True, transform=transform)
testset = torchvision.datasets.CIFAR10(root='./data', train=False, download=True, transform=transform)
```

4. Zero-Shot and Linear Probe Classification using OpenCLIP

- **Zero-shot classifier:** Create text descriptions (prompts) for each class.
- **Linear probe:** Extract embeddings and train a logistic regression model.

4.1. Implementing Zero-Shot

- Create text prompts using the pattern "a photo of a {class}" for each CIFAR-10 class
- The text prompts are tokenized and passed through the CLIP text encoder to get text features
- The input images are passed through the CLIP image encoder to get image features.

4.2. Implementing Linear Probe

- Extract features from the CLIP model for the images.
- Train a linear model (e.g., Logistic Regression) on these embeddings.

4.3. Performance Comparison

Method	Accuracy on CIFAR-10
Zero-Shot (prompt: "a photo of a X")	86.17%
Linear Probe	77.33%

Classification Accuracy of Zero-Shot and Linear Probe Methods on CIFAR-10

CLIP's zero-shot accuracy surpasses a supervised linear probe. This highlights the power of multimodal pretraining, especially when the model has learned strong semantic priors via natural language.

5. Impact of Prompts on OpenCLIP Performance

Using different prompts such as:

- "{class}"
- "a photo of a {class}"
  ⋮
- Completely irrelevant prompt for each class (like "a programming tutorial using the dog language" for class "dog")

5.1. Experimental Results of Prompt Engineering

Prompt	Accuracy on CIFAR-10
"{class}"	0.859
"A photo of {class}"	0.865
"This is {class}"	0.872
Long prompt containing class	0.869
"The president of the {class}"	0.870
Completely irrelevant prompts containing class	0.711

Accuracy of CLIP under various prompt formulations

- The results show that prompt design significantly impacts CLIP's zero-shot classification accuracy. More natural prompts (e.g., "a photo of a {class}") result in higher accuracy.
- The plain {class} prompt achieves near 86%, while using minimal prompt. This likely yielded solid results because CLIP's text encoder is good at mapping raw labels directly.
- In long prompt case, despite being long, this is a very descriptive sentence. CLIP was trained on diverse web data, so phrases like "DSLR camera" might have appeared frequently.
- "The president of the {class}" may seem nonsensical for CIFAR-10, yet achieved solid result! CLIP likely focused on the key token and the rest ("the president of") might have been ignored or given low attention. This prompt is not helpful, but not harmful either to derail the embedding.
- Irrelevant Prompts Significantly Degrade Performance. The drop to 0.7113 for "completely irrelevant prompts" shows what happens when the context is semantically misleading and the class word is used in absurd or uncommon ways.

6. Challenges

- Creating suitable prompts out of each CIFAR-10 class to provide natural language descriptions for zero-shot classification.
- Time-consuming embedding extraction for larger datasets.
- Memory optimization required in Google Colab for larger datasets.