Overview

Slide 1
Labeling Large Repositories

Slide 2
Self-Supervised Learning (SSL) in General
Contrastive learning

Slide 3
Practice: Supervised Contrastive Pretext

Slide 4
Practice with SimCLR(started)

Labeling Large Repositories

Solution 1: Manual labeling, for example:

- Manually label a repository of ten million animal images.
- Identify positive, negative, and neutral statements in thousands of text documents.

Manual labeling is often a costly, labor-intensive, and time-consuming task.

Solution 2: Manually label a small portion of the data first. Next, develop a supervised learning model to learn from the labeled data as the training and label the rest.

- Dataset: ten million images of cats and dogs. Only 10,000 (0.1%) of them manually labeled.
  How to label the rest of unlabeled images?
- Tasks to address: the model is highly probable to be inaccurate and/or overfitted as the training rate is only 0.1%.
- We can benefit from transfer learning and fine-tuning techniques to increase the accuracy of such supervised models.

Self-Supervised Learning (SSL) in General

Previous example: ten million images of cats and dogs, only 10,000 (0.1%) manually labeled.

- The solutions mentioned on the previous slide entirely ignore the unlabeled repository.
- The Self-Supervised Learning (SSL) objective is to implement the repository’s labeled and unlabeled data together to develop a base model in a pretext task and transfer and fine-tune that base model using the limited labeled data in a so-called downstream task.

Two primary SSL techniques exist to create a pretext model: **contrastive learning** and **generative learning**.

Contrastive learning

- **In contrastive learning**, the input data, labeled or unlabeled, are augmented automatically using an augmentation technique.
- First, an augmentation is selected. Next, that augmentation is automatically applied to the input data, labeled or unlabeled.
- So there will be pairs of original and corresponding augmented images.

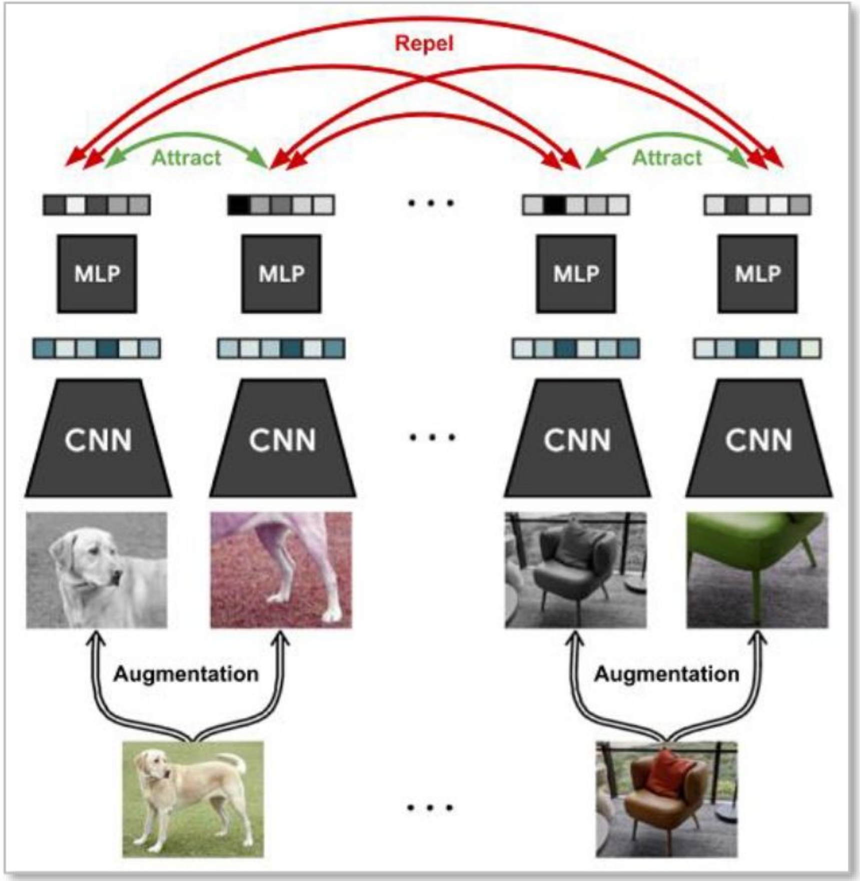
Contrastive Pretext Task

Article: https://atcold.github.io/pytorch-Deep-Learning/en/week10/10-1/

SimCLR

Steps:

- Create 2 versions of an image by applying data augmentation techniques
- Apply a CNN (like ResNet) and obtain as output a 1D feature vector
- Apply a small MLP on that feature vector
- The output features of the two augmented images are then trained to be close to each other, while all other images in that batch should be as different as possible.


Repel
Attract
Attract
MLP
MLP
MLP
MLP
CNN
CNN
CNN
CNN
Augmentation
Augmentation