3684 – Advanced Topics in Machine Learning

Home Assignment #3 – Self Supervised Learning (SimCLR)

ID: 201466349

Abstract

In this assignment report, we would review the key concepts examined in assignment #3 which are methods of self-supervision and more specifically, the SimCLR (simple framework for contrastive learning) for Self Supervised Learning. We would cover some of the theoretical background supporting this framework and then demonstrate our implementation and experiments conducted in order to improve a simple image classification model by the generation of better representations of those images.

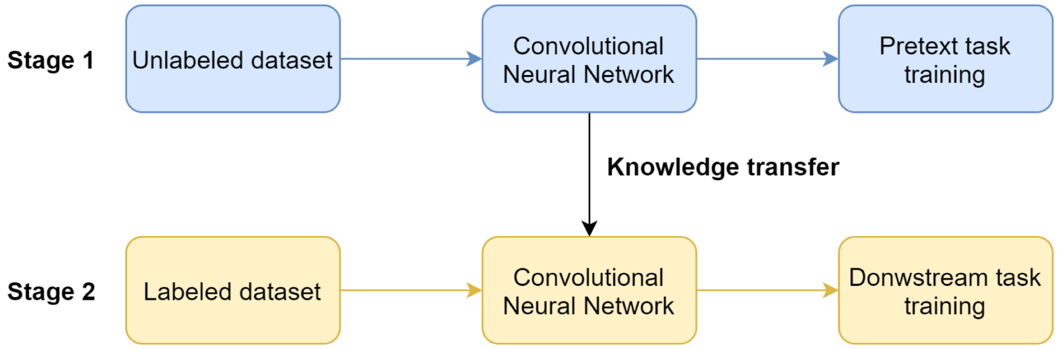
Self-Supervised Learning (SSL)

The SSL practice is a way of generating a useful representation of data, such as images or text, via the process of learning some "pretext" task on self-generated data.

The classical SSL flow is as follows:

1. Define some "pretext" task
2. Given an instance x, generate some augmentations of x: $x^{+}_{1}, \ldots x^{+}_{n}$ to be used as train data, using actual x as their "label"
3. Generate labels automatically, based on x and its augmented instances.
4. Perform model training using the generated training data
5. **Use the model for feature extraction**

By the end of the process, a new representation is created for instance x, that can be later used for any other learning task (i.e. classification, regression, etc.).



Contrastive SSL

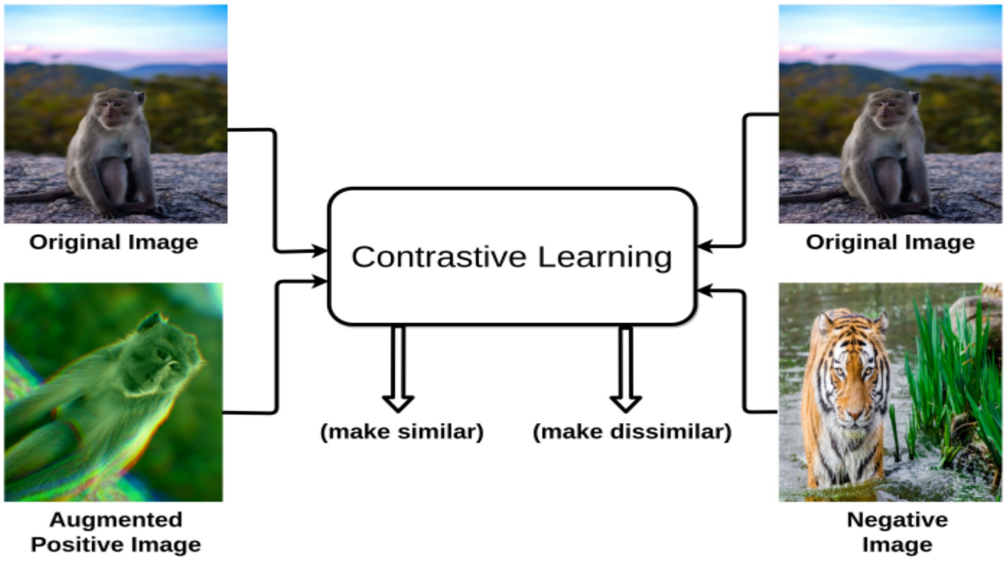
A flavor of classical SSL was created in order to improve its performance.

In contrastive learning, instead of using "positive" examples only, which are augmentations of the same instance x, for each training process, we also introduce **negative** examples to the model, which are new instances $x^{-}_{1}, \ldots x^{-}_{n}$ that have nothing to do with the original instance x.

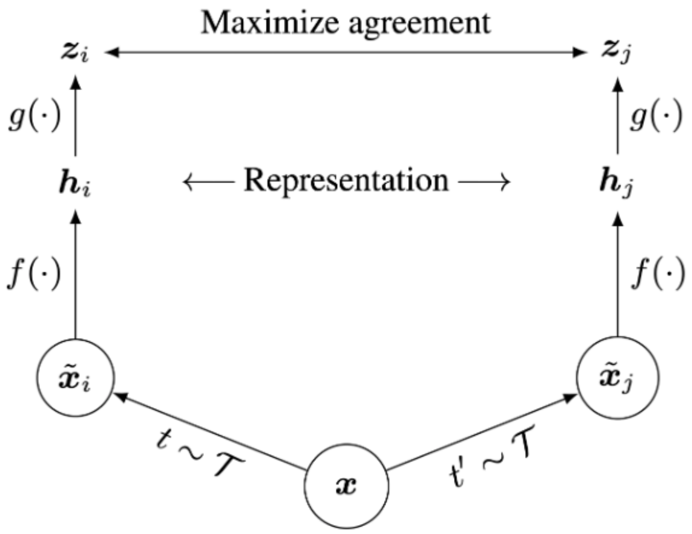
Given some score function we want to learn an encoder f such that

$score(f(x),\ f(x^{+}))\ >>\ score(f(x),\ f(x^{-})$

Meaning, find the encoder that maximizes the score difference between x and its augmentations to x and other unrelated instances.



SimCLR - A framework for contrastive learning
SimCLR introduces a new architecture that uses contrastive learning to learn good visual representations. For each instance in the dataset, SimCLR generates two differently augmented views of that instance, called a positive pair. Then, the model is encouraged to generate similar representation vectors for this pair of images.



Given an instance x, SimCLR uses two different data augmentation schemes t and t' to generate the positive pair of images $\tilde{x}i$ and $\tilde{x}j$.
f is a basic encoder net that extracts representation vectors from the augmented data samples, which yields hi and hj, respectively.
Finally, a small neural network projection head g maps the representation vectors to the space where the contrastive loss is applied.

The goal of the contrastive loss is to maximize agreement between the final vectors $zi=g(hi)$ and $zj=g(hj)$.

After training is completed, we use f and the representation h to perform downstream tasks, such as classification.

Implementation Details

In this assignment, we implemented the SimCLR framework using and images dataset, in order to imporve an image classification task. We used the following definitions:

1. Data augmenataions

   We defined 4 types of possible image augmentations to generate $\tilde{x}$:

   a. Randomly resize and crop to 32x32.

   b. With probability 0.5, Horizontally flip the image

   c. With a probability of 0.8, apply color jitter

   d. With a probability of 0.2, convert the image to grayscale

2. As to the original paper recomendation, we used ResNet as the encoder f to generate h out of the newly augmented images $\tilde{x}$.

3. We then implemented functions to calculate to the total loss L, defined as:

$$L = \frac{1}{2N} \sum_{k=1}^{N} [\ l(k,\ k+N) + l(k+N,\ k)\ ]$$

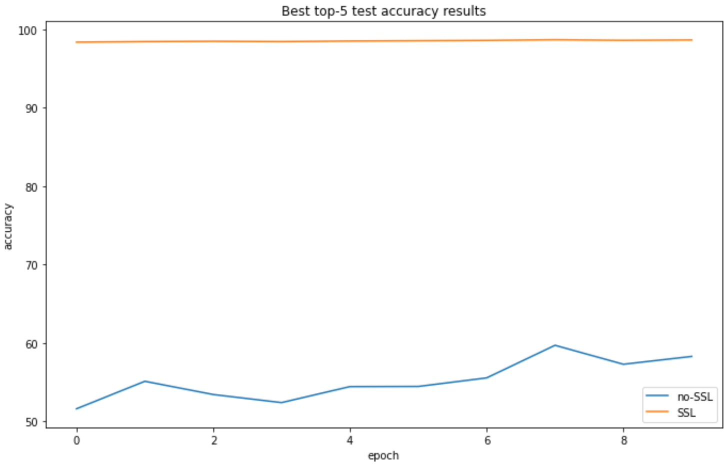
   Where

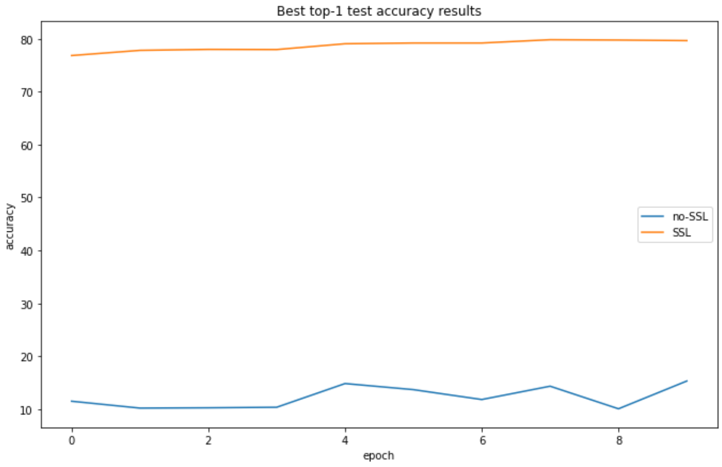
$$l\ (i, j) = -\log \frac{\exp(\ \mathrm{sim}(z_i, z_j)\ /\ \tau)}{\sum_{k=1}^{2N} \mathbb{1}_{k \neq i} \exp(\ \mathrm{sim}(z_i, z_k)\ /\ \tau)} \qquad \mathrm{sim}(z_i, z_j) = \frac{z_i \cdot z_j}{||z_i||||z_j||}$$

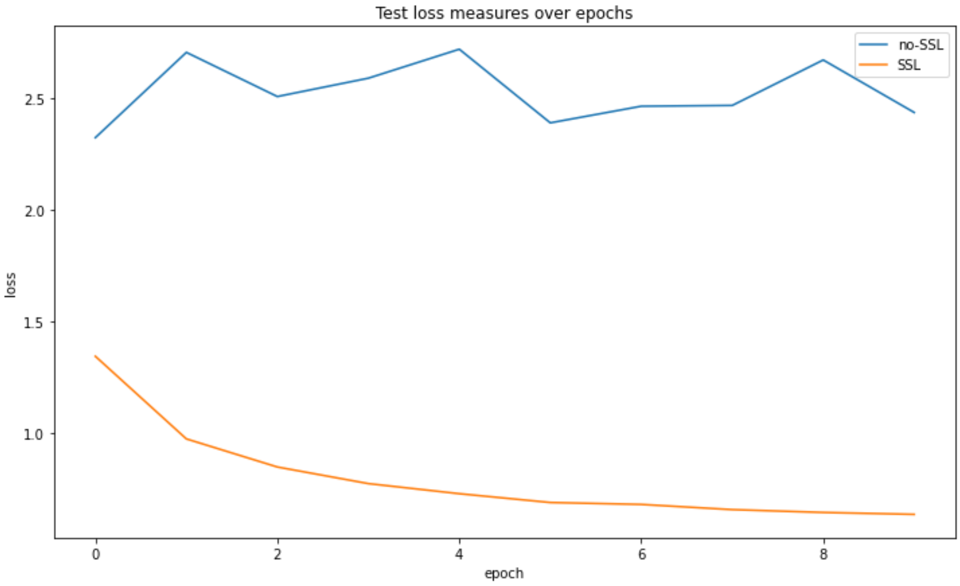
4. We implemented a training process that receives the training data and trains the model to solve the pretext task of maximized agreement.

5. We then used the resulting f pretrained-weights to train an image classification model and compare that to a fully retrained model.

Results:







We can see that using the pre-trained weights generated by the SimCLR run created a much better representation for the images, which resulted in a much better classifier in terms of both accuracy and loss measurements over a 10-epoch training. This demonstrates how beneficial SSL is, and SimCLR specifically, in using a simple pretext task to get a good grasp of the distribution of the data, and thus generating a robust and useful representation of that data to later be used in different tasks.

This also emphasizes the importance of the representation itself and shows that the same architecture can produce very different results, depending on the actual representation of the data being used for training.