Self-Supervised Learning with SimCLR and RotNet

Dai, Anna
Duke University
anna.dai@duke.edu

Harutyunyan, Tigran
Duke University
tigran.harutyunyan@duke.edu

Saikia, Deekshita
Duke University
deekshita.saikia@duke.edu

Abstract

A major bottleneck in various use cases of deep convolutional neural networks (ConvNets) in recent days is the lack of fully labeled data. The paper aims to compare performance and learned representations of two self-supervised frameworks: SimCLR and RotNet on partially labeled data. SimCLR is a framework for contrastive Learning of image representations proposed by Chen, et al. [1]. RotNet is an unsupervised semantic feature learning framework proposed by Gidaris, et al. [2] to learn representations from rotated data. We implement both frameworks with a ResNet-20 ConvNet Encoder to evaluate performance on the CIFAR-10 dataset. We find that SimCLR outperforms RotNet using linear evaluations, but RotNet in its semi-supervised state outperforms SimCLR linear evaluations. We further observe in the feature maps of the encoders from both frameworks that the RotNet encoder learns the edges of images in the first two blocks of convolutional layers, whereas the SimCLR encoders learns more comprehensive information about the images.

1 Introduction

Convolutional neural networks (ConvNets) have unparalleled capacity to learn high level semantic image features, but usually require large amounts to labeled data to train. Labeling data comes at a high cost and is not easily scalable. The field of self-supervised learning explores how supervisory signals can be learned from the data itself, often leveraging its underlying structure, thereby reducing the need for labels for predictive tasks like classification. In recent years, approaches like contrastive learning approach with SimCLR [1] and unsupervised learning with RotNet to predict image rotations [2] have been gaining traction. In this project, we seek to replicate these approaches with a lighter network, and benchmark their performance against an established "supervised baseline".

2 Methodology

2.1 Overview

We implement both SimCLR and RotNet frameworks using a ResNet-20 backbone encoder applied on the CIFAR-10 dataset to increase comparability between the two frameworks. Our ResNet-20 encoder has three residual blocks of three convolutional layers each. We also use the vanilla ResNet-20 model as a supervised baseline model to benchmark performance. However, due to the different nature of the frameworks, we proceed to lightly tune hyper-parameters of our model, including `batch_size`, `lr_decay`, `optimizer`, `num_epochs`, separately in order to achieve the best performance we can given our computational limitations.

2.2 SimCLR

2.2.1 The Contrastive Learning framework

Contrastive learning has been shown to be a powerful approach for learning visual representations from large amounts of unlabeled data. The core idea behind contrastive learning is to use pairs of similar and dissimilar images to learn a representation that captures the differences between the images in the pair.

This framework has the following major components:

1. A stochastic data augmentation module that transforms any input image into a pair of correlated views, $\mathbf{x_i}$ and $\mathbf{x_j}$ of the same input, which we call a positive pair. We sequentially apply a random resized crop, followed by a random color distortion.
2. The ResNet-20 base encoder network $F(\cdot)$, which extracts features from the input images and encodes them into a lower-dimensional latent space.
3. A 2-layer MLP projection head $G(\cdot)$, which maps the latent representations to a higher-dimensional space.
4. A contrastive loss function, which is defined for a positive pair in the following manner:

$$l_{i,j} = -log \frac{exp(sim(\mathbf{z_i}, \mathbf{z_j})/\tau}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} exp(sim(\mathbf{z_i}, \mathbf{z_j})/\tau)}$$

where $sim(\mathbf{u}, \mathbf{v}) = \mathbf{u}^\top \mathbf{v}/||u||||v||$ and τ denotes a temperature parameter. The indicator function $\mathbb{1}_{[k \neq i]}$ evaluates to 1 iff $k \neq i$.

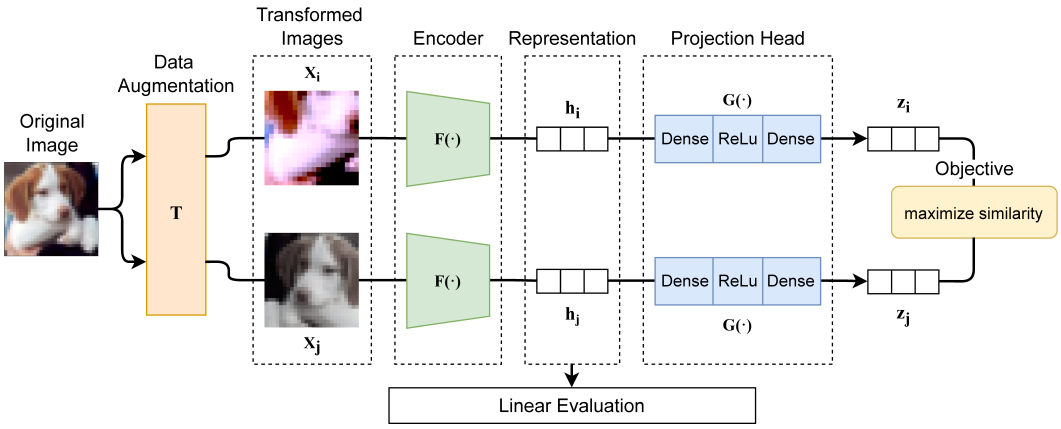


Figure 1: Architectural diagram of SimCLR framework

Hyper-parameters For our implementation, we used the following settings: `EPOCHS=100`, `BATCH_SIZE=1024`.

We optimized a temperature-scaled cross entropy-loss using a LARS optimizer with a learning rate of 0.012 and a weight decay of $1e{-}6$. Further, we use a linear warm-up for the first 20 epochs, and decay the learning rate with the cosine decay schedule.

2.3 RotNet

The RotNet framework from paper [2] involves two primary steps to perform classification tasks: 1) unsupervised training on rotated images to predict its rotation and 2) semi-supervised fine-tuning process on the pre-trained unsupervised model using labeled data to classify images.

2.3.1 Unsupervised model

We rotate each input image (by 0^0,90^0,180^0,270^0), assign labels according to its rotated angel, and train a ConvNet to learn representations of transformed input images. To effectively predict image rotations, the ConvNet must first learn to identify and locate important objects (i.e. dog's ears) in images to understand their context. It must also understand how different types of objects are typically depicted in the input images and use this information to relate the position of the objects to the dominant orientation of the image.

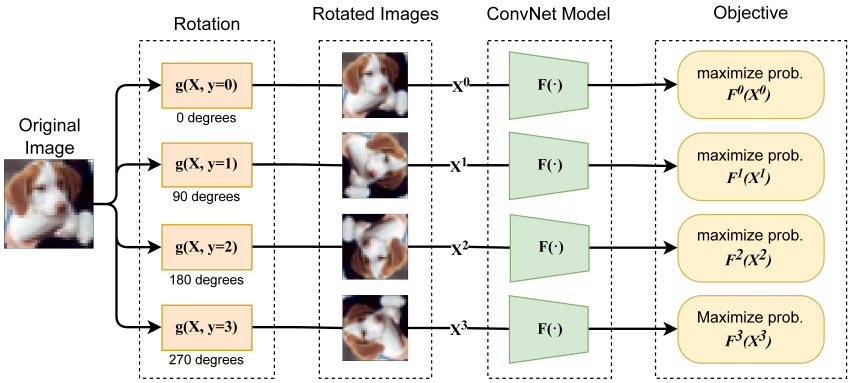


Figure 2: Architectural diagram of RotNet framework

Hyper-parameters For our implemention, we used the following settings: `EPOCHS=100`, `BATCH=128`, `INITIAL_LR=0.1`, `DECAY=0.2`, `DECAY_EPOCHS=[30, 60, 80]`.

2.3.2 Semi-supervised

The architecture of the semi-supervised model is the same as the unsupervised model, except that the fully connected layer has an output size of 10 for image classification from CIFAR10.

2.4 Evaluation Process

To evaluate our model performance we applied linear evaluation protocol on both SimCLR and RotNet encoders. Linear evaluation one of the intended methods to evaluate model performance in the SimCLR paper [1], where we load the representations learned by the encoder, freeze all weights, and train one added linear layer with labeled data (textttout_features = 10) to classify images. It is difficult to find a fair comparison for SimCLR linear evaluation in RotNet due to the different intended use of the RotNet model for downstream tasks. Therefore, we evaluate the RotNet encoder both with the same linear process as SimCLR as well as with the intended semi-supervised setting from the RotNet paper [2]. Semi-supervised setting involves loading the trained encoder model, freezing only the first residual block(s), and retraining the latter block(s) with a fully-connected layer (textttout_features = 10) on labeled data.

One goal of this project is to understand how the SimCLR and RotNet frameworks perform on image classification tasks when we have partially labeled data. Therefore, we train our evaluation models on only 1% and 10% of our labeled data (50 images and 500 images per class respectively).

For linear evaluations, we trained the fully-connected layer over 90 epochs as instructed in the paper [1] on 1% and 10% labels. For the RotNet semi-supervised setting, we froze the first two residual blocks of encoder and retrained the last over 80-100 epochs.

3 Experimental results

3.1 Model accuracy

We observe from table 1 that performance improved significantly as we increase the percentage of labeled data for all models. Of the four models, the supervised baseline model observed the largest

boost in model performance of 30% from when we increased from 1% to 10% labels. Whereas, our SimCLR framework observed the smallest increase in prediction accuracy of around 5%. Further, we observe that SimCLR with linear evaluations outperforms the baseline model by 15% on 1% labeled data, but the baseline model performs better on 10% labeled data. In other words, SimCLR achieves a much better performance on almost no labels to train on but benefits less from more data labels in our implementation.

In comparison to all other models, RotNet performs the worst when linearly evaluated. It achieves a lower test accuracy than our baseline model at 1% and SimCLR model at 10%. This was expected as the RotNet ConvNet representations were not intended, according to the paper [2], to be used in a linear manner downstream. Instead, the framework expects continued training with limited labeled data to be evaluated in its semi-supervised state. In our implementation, our semi-supervised RotNet model achieved the best model accuracy of 70.71% with only 1% data labels and 83.64% with 10%.

Model	1% Labels	10% Labels
Supervised Baseline	50.54%	81.94%
SimCLR Linear Evaluation	64.15%	69.30%
RotNet Linear Evaluation	47.31%	55.59%
RotNet Semi-supervised	70.71%	83.64%

Table 1: Model performance on 1% and 10% labeled data

Overall, the results from our implementations are within reasonable range as compared to the two papers [1] [2]. Our RotNet achieved results on par with the paper while our SimCLR implementation achieved lower accuracy than that from the paper on CIFAR-10. We believe variation in model performance could be due the different encoder model and our limited resources to fine-tune our hyper-parameters and run for longer epochs.

3.2 Discussion

From the linear evaluation experiments performed on both the SimCLR and the RotNet, we observe that SimCLR performs significantly better than the RotNet. A potential implication of this could be that the representations learned by the encoder of the SimCLR are readily applicable to downstream tasks like image classification. Whereas the RotNet would require additional fine-tuning of the last residual block (i.e. its semi-supervised setting) to increase model performance.

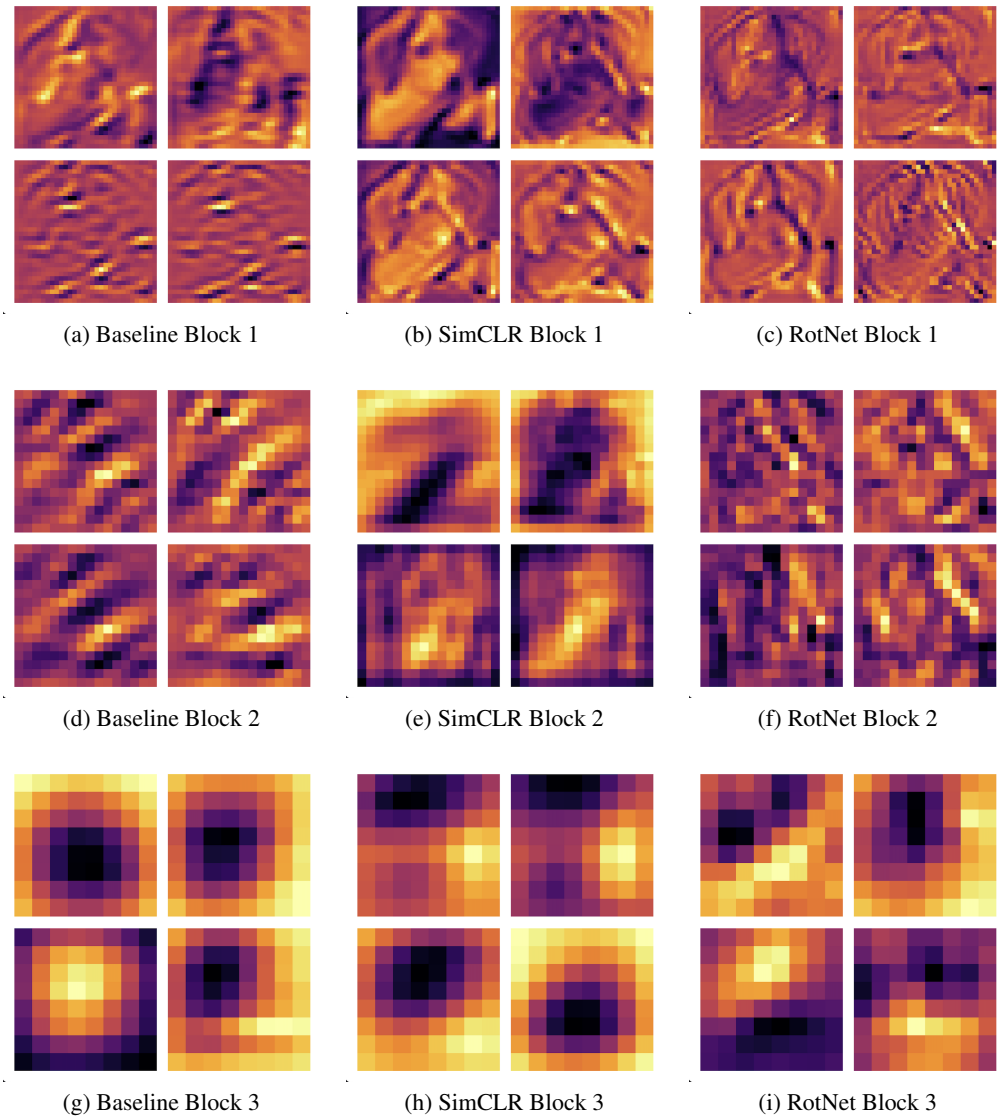
We visualize the representations learned at the end of each block of our ResNet-20 model from our supervised, SimCLR, and RotNet in Figure 3 to gain insight on our results.

If we look at the first blocks in Figure 3(a), (b), and (c), we observe the baseline model appear to learn a variety of different features, whereas SimCLR and RotNet appear to learn more targeted features. Specifically, the RotNet model seem to focus on learning outlines of objects. The difference between the types of features learned by SimCLR and RotNet could explain why SimCLR performs better than RotNet as a linear classifier for this particular dataset.

4 Conclusion

We have implemented a SimCLR and RotNet framework to compare their performance as an image classifier on limited labeled data from the CIFAR-10 dataset. SimCLR performed better than RotNet as a linear classifier, whereas RotNet in its semi-supervised setting outperformed all other models.

Acknowledgments We would like to thank Prof. Yiran Chen, Qijia Huang and all the amazing TAs for their support on this project and their feedback on our poster (which can be found here).



(a) Baseline Block 1 (b) SimCLR Block 1 (c) RotNet Block 1

(d) Baseline Block 2 (e) SimCLR Block 2 (f) RotNet Block 2

(g) Baseline Block 3 (h) SimCLR Block 3 (i) RotNet Block 3

Figure 3: Feature maps from last layer of each ResNet-20 residual block (4 filters shown)

References

[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *CoRR*, abs/2002.05709, 2020.

[2] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *CoRR*, abs/1803.07728, 2018.

A Appendix

Contributions by team-members The team-members made the following contributions to the project:

1. **Anna Dai** - Creating the SimCLR framework, linear evaluation of Resnet-20 encoder for SimCLR
2. **Tigran Harutyunyan** - Creating the RotNet framework, linear evaluation and semi-supervised training for RotNet
3. **Deekshita Saikia** - Creating the SimCLR framework, tuning and pre-training of Resnet-20 encoder for SimCLR

Code The code for this project can be accessed in this GitHub repository.

Peer Review We vote team D5 for best poster, named 'Contrastive Representation Learning using SimCLR and RotNet frameworks'. The poster presents results on the performances of these approaches under different self-supervised settings. The poster is well organized with plots to support their hypothesis. A variety of experiments were run and are well-balanced with their conclusions.