A Comparison of Representation Learning Methods

Mac Gray, Dany Jabban, Zeyu (Michael) Li

Introduction

One of the biggest factors limiting the development of high performance models is the time intensive and expensive task of data labeling. Further, there is an extremely large amount of unlabeled data publicly available. This imbalance led researchers to the logical question, can unlabeled data be used in an unsupervised manner to help models learn effective visual representations?

In this work we look at RotNet and SimCLR, two of the most recent unsupervised representation learning methods. These models use unlabeled data to extract meaningful representations of input data through augmentations. We then compare the performance of these approaches on various tasks and settings. The two main evaluation protocols used in our experiments are linear evaluation and fine-tuning on limited labels.

One additional discovery made by the SimCLR research is that wider models perform better at contrastive learning. We look into compression techniques that can reduce the size of these models with minimal performance degradation.

Contributions:

Mac Gray: RotNet training, lin eval, fine-tuning, classifier and base arch implementation.

Michael Li: SimCLR training, linear evaluation, fine-tuning, and compression code.

Dany Jabban: All data preprocessing, vanilla resnet code, wide resnet training, plots/figs

Methodology

SimCLR pre-training

- Take two copies of an image from CIFAR-10 dataset and apply 2 sets of random augmentations (as specified by SimCLR paper) to the images
- Pass the two images through a standard ResNet base encoder
- Take the base encoder outputs and pass them through a nonlinear projection head, which is optimal for contrastive loss
- Representations of the augmented pairs from the original image are used to compute the contrastive loss. All other image representations are used as negative examples in the contrastive loss.
- Train models for 1000 epochs on varying batch sizes from 256 to 4096. Use LARS optimizer instead of SGD/momentum to stabilize training for large batch sizes

RotNet pre-training

- Use a batch of images from Cifar10 and rotate each image 0, 90,180, and 270 degrees
- Include all 4 rotations in a batch so each batch contains 4N data points
- Pass each batch through RotNet model to obtain the 4 dimensional output vector
- Each dimension represents predictions of one of the 4 rotation classes
- Use these outputs and the labels of the actual rotations to calculate loss

Linear Evaluation

- For SimCLR we remove the non-linear projection head
- For both models we freeze the weights
- For SimCLR we add a logistic regression head, for RotNet we use a nonlinear head
- Finally use labeled Cifar10 training data to train the head's in a supervised manner

Fine-tuning

- For SimCLR we keep all models' parameters unfrozen and train them on 1%, 10%, 20%, and 100% of the labeled dataset in a supervised manner
- For RotNet we compare do one trial with unfrozen weights and another with frozen weights

Results

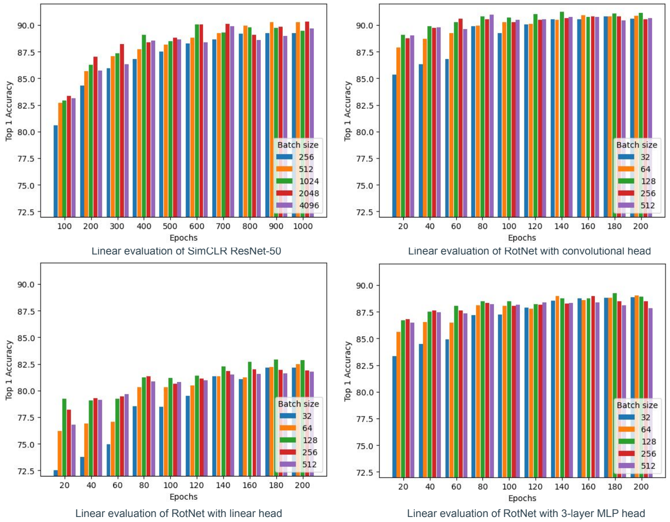

Figure 1. Visualisation of Linear Evaluation Performance.

Table 2: CIFAR-10 accuracy of models fine-tuned with limited labels. Asterisk (*) indicates 6-bit quantization

Method	Architecture	Label Fraction 1%	10%	20%	100%
Supervised Baseline	ResNet-50	28.3	48.6	76.8	95.9
Supervised Baseline	NIN	53.5	81.2	86.1	92.8
RotNet	NIN-Frozen	69.6	83.7	86.5	91.1
RotNet	NIN-Unfrozen	41.8	75.7	81.7	92.1
SimCLR	ResNet-50	89.4	92.0	92.3	94.1
SimCLR	ResNet-50 prune 70% *	88.5	91.3	91.6	93.8
SimCLR	ResNet-50 prune 90% *	73.0	85.2	85.7	91.2
SimCLR	ResNet-50 prune 95% *	35.7	69.6	70.6	87.6
SimCLR	ResNet-50 prune 99% *	13.7	23.0	23.1	64.0
SimCLR	ResNet-50 (2x)	90.0	92.4	92.8	94.7
SimCLR	ResNet-50 (2x) prune 70% *	89.4	91.8	92.4	94.3
SimCLR	ResNet-50 (2x) prune 90% *	83.2	88.8	89.8	93.1
SimCLR	ResNet-50 (2x) prune 95% *	52.1	81.3	81.9	90.8
SimCLR	ResNet-50 (2x) prune 99% *	14.8	25.0	30.3	73.3

Figure 2. Visualisation of representations.

Conclusion

SimCLR consistently outperforms RotNet in representation learning tasks as evidenced by linear evaluation and fine-tuning evaluation protocols. While SimCLR requires large batch sizes to optimally learn feature representations, RotNet excels under relatively smaller batch sizes. We also note that RotNet requires nonlinear heads to be trained during evaluation to achieve impressive metrics. This is because RotNet is designed to use the feature maps from an early layers' convolutional block. Therefore, these features cannot be linearly mapped down to 10 output classes, requiring a nonlinear classifier. It is objectively true that training a nonlinear head for evaluation is advantageous over a linear one, yet SimCLR's features are still on par with RotNet during evaluation.

We see that wider networks are better for representation learning for SimCLR. In addition, the percentage of data required for fine-tuning to achieve high accuracy is lower compared to that required for RotNet.

Compared to the baseline (vanilla ResNet with limited labels), the fine-tuned SimCLR models with limited data (1%, 10%, 20%) perform better. Meanwhile, we see that when there are more than 20% of labels available, RotNet struggles to outperform the supervised baseline. This indicates that SimCLR is the better choice for a wide range of missing label percentages. We also note that due to RotNet's design, it does not benefit from unfreezing its base weights when there is anything shy of 100% of the labels available. This indicates that RotNet has a heavy reliance on the unsupervised representations it has learned, while SimCLR is more flexible to fine-tuning.

Global iterative pruning with 6-bit quantization of the SimCLR network during fine-tuning resulted in lower performance overall. However, the SimCLR model is amenable to compression and quantization since the accuracy does not drop precipitously as more weights are pruned. Finally, the pruned wide ResNet has higher accuracy than the pruned non-wide ResNet when the same fraction of weights are pruned.

References

Johnson, J. Emmanuel, et al, "RotNet: Fast and Scalable Estimation of Stellar Rotation Periods Using Convolutional Neural Networks," 2020.

Linderman, G., et al. "Fast interpolation-based T-Sne for improved visualization of single-cell RNA-seq data," in *Nature Methods*, vol. 16, no. 3, pp. 243–245, 2019.

Chen, T., et al. "A Simple Framework for Contrastive Learning of Visual Representations," 2020.