Less can be more in contrastive learning

Jovana Mitrovic **Brian McWilliams** **Melanie Rey**
DeepMind, UK
{mitrovic, bmcw, melanierey}@google.com

Abstract

Unsupervised representation learning provides an attractive alternative to its supervised counterpart because of the abundance of unlabelled data. Contrastive learning has recently emerged as one of the most successful approaches to unsupervised representation learning. Given a datapoint, contrastive learning involves discriminating between a matching, or positive, datapoint and a number of non-matching, or negative, ones. Usually the other datapoints in the batch serve as the negatives for the given datapoint. It has been shown empirically that large batch sizes are needed to achieve good performance, which led the the belief that a large number of negatives is preferable. In order to understand this phenomenon better, in this work investigate the role of negatives in contrastive learning by decoupling the number of negatives from the batch size. Surprisingly, we discover that for a fixed batch size performance actually degrades as the number of negatives is increased. We also show that using fewer negatives can lead to a better signal-to-noise ratio for the model gradients, which could explain the improved performance.

1 Introduction

Large amounts of supervised data are usually needed to train deep neural networks successfully and acquiring the necessary amount of supervised signal can be costly or impractical. On the other hand, large amounts of unsupervised data are often readily available. We can leverage this data by pretraining representations for the unknown downstream tasks of interest. Self-supervised methods utilize proxy tasks defined on unsupervised data in order to pretrain representations. Recently, a particular kind of self-supervised methods, contrastive learning, has achieved state-of-the-art performance in unsupervised representation learning [10, 6, 7, 2, 3].

Contrastive representation learning methods work by learning to discriminate between datapoints that are similar to the current one and randomly sampled other datapoints, known as the negative examples, which are presumed to be dissimilar. To perform this classification, similarity scores of pairs of similar examples and negative examples are computed. Usually, similar examples are different augmentations of the original datapoint. The most performant contrastive methods such as CPC [10, 6], AMDIM [2] and SimCLR [3] all follow this general approach. The main differences between these approaches are in the data augmentations and encoder architecture used. While CPC and AMDIM use large custom networks, SimCLR uses a standard ResNet50 [5] in combination with strong data augmentations.

Notwithstanding the differences in encoder architectures employed in contrastive learning coupled with some differences in the computation of the similarity scores between representations, [10, 6, 7, 2, 3] all argue for using large sets of negative examples. They argue that this leads to more hard negatives being used in learning, which improves the quality of the resulting representation. Thus, they use very large batches in order to have larger number of negatives available and have reported significant performance gains with increasing batch sizes. Recently, [4] proposed to include a queue of data samples in order to increase the number of negative examples beyond the batch size.

1st I Can't Believe It's Not Better Workshop (ICBINB@NeurIPS 2020), Vancouver, Canada.

In this work we examine the role of negatives in contrastive learning and test the widely held belief that using large numbers of negatives is important for learning good representations. We decouple the batch size from the number of negatives by randomly sampling negatives for each datapoint. We perform experiments on CIFAR100 [9] and downsampled ImageNet [12] for two highly performant contrastive methods, SimCLR [3] and ReLIC [1]. Keeping the batch size fixed, we vary the number of negative examples and discover that using a large number of negatives actually hurts performance. This shows that the widely used heuristic of using as many negatives as feasible can lead to sub-optimal performance. For CIFAR100, we find that the best performance is attained with just 1 and 2 negative examples for ReLIC and SimCLR, respectively. For downsampled ImageNet, we find that the best performance is achieved with 100 negative examples which is much lower than the batch size of 4096.

2 Background

Denote by X the unlabelled observed data and by Y the targets of the unknown downstream task. We want to pretrain a representation $f(X)$ using unsupervised data only such that it is useful for solving the downstream task Y. Contrastive methods learn representations by comparing a data point to a series of similar and dissimilar data points. Specifically, for an observation x from X, the representation $f(x)$ is optimized such that the similarity between $f(x)$ and $f(x^+)$ is maximized, while the similarity between $f(x)$ and $f(x^-)$ is minimized; x^+ and x^- are data points that are semantically similar and dissimilar to x, respectively, and are called positive and negative examples. Given a mini-batch $\{x_i\}_{i=1}^B$, a general form of the contrastive loss is:

$$-\frac{1}{B}\sum_{i=1}^{B}\sum_{x_j^+\in\mathcal{P}_i}\log\frac{\exp(z_i^T z_j^+/\tau)}{\exp(z_i^T z_j^+/\tau)+\sum_{x_n^-\in\mathcal{N}_i}\exp(z_i^T z_n^-/\tau)}, \tag{1}$$

with $z_i = g(f(x_i))$, τ a temperature parameter, $\mathcal{P}_i$ the set of positive examples and $\mathcal{N}_i$ the set of negative examples for data point x_i, and g is a simple function (e.g. identity or a small fully connected network) called the projection head.

Both SimCLR [3] and ReLIC [1] apply augmentations to datapoints before computing the similarities between them and use one positive example per datapoint. The objective in SimCLR is

$$-\frac{1}{B}\sum_{i=1}^{B}\sum_{\substack{v,w\in\{s,t\}\\ v\neq w}}\log\frac{\exp(z_{i,v}^T z_{i,w}/\tau)}{\sum_{k=1}^{B}\sum_{r\in\{s,t\}}\mathbb{1}_{[k\neq i, r\neq v]}\exp(z_{i,v}^T z_{k,r}/\tau)} \tag{2}$$

with $z_{i,a} = g(f(x_{i,a}))$ and $x_{i,a}$ denotes the datapoint x_i under augmentation a. For ReLIC, we use the following version of the objective

$$\frac{1}{B}\sum_{i=1}^{B}\sum_{v,w\in\{s,t\}}\left[-\log\frac{\exp\left(z_{i,v}^T, z_{i,w}/\tau\right)}{\sum_{m=1}^{M}\exp\left(z_{i,v}^T, z_{m,w}/\tau\right)}+\alpha KL(p_i^{vw}, p_i^{wv})\right] \tag{3}$$

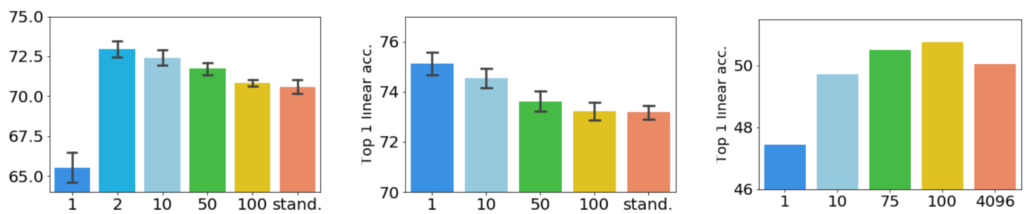
with $p_i^{vw}(m) = \exp\left(z_{i,v}^T z_{m,w}/\tau\right)/Z_i^{vw}$, where Z_i^{vw} is the normalizing constant. KL denotes the Kullback-Leibler divergence, M negative examples are sampled uniformly at random from the batch for each datapoint and α is the invariance penalty weight.

3 Experiments

We study the impact of the number of negative examples on the quality of the learned representation. We measure representation quality under the linear evaluation protocol as proposed in [8] by training a linear classifier on top the frozen representation and reporting the top 1 accuracy on the test set. While in previous research the batch size and number of negatives were tightly coupled, we believe this work is the first one to consider the effect of the number of negative examples separately from the batch size. We achieve this by decoupling the batch size and the number of negatives by restricting the sum in the denominator in (2) and (3) to be over a random subset of all negative examples available in the current batch.

We assess the performance SimCLR and RELIC, both of which use the data-augmentation scheme proposed in [3] but differ in their objectives as described in Section 2. For each method and each number of negatives we sweep over the temperature, learning rate, and, for RELIC the invariance penalty weight. For SimCLR we optimized over the following ranges: learning rate in $[0.2, 1.5]$, temperature in $[0.2, 10]$. For RELIC we considered: learning rate in the range $[0.5, 1.5]$, temperature in the range $[0.2, 50]$ and invariance pentalty weight in the interval $[0.1, 500]$. In all experiments we use Resnet50 [5] as the encoder. For the non-linear projection head we use fully-connected networks with dimensions $[2048, 128]$ for SimCLR and $[4092, 2048, 1024, 512, 128]$ for RELIC. All models are trained for 2000 epochs using the LARS optizimer [13] with the same parameters as in [3] and a batch size of 1120.

In Figure 1a and 1b we report the top 1 accuracy on CIFAR100 obtained for different numbers of negative examples under the linear evaluation protocol [9]. We report the average performance over 5 random seeds, with the errors bars denoting the standard deviation. In Figure 1a we observe that performance decreases with as number of negatives increases apart from when 1 negative is used. In Figure 1b we see that performance decreases with as number of negatives increases. Unlike SimCLR, RELIC has the highest performance with 1 negative. Moreover, the gain when using fewer negatives is more pronounced with RELIC than with SimCLR. These results directly contradict the commonly held belief that a large number of negatives is needed for good performance.



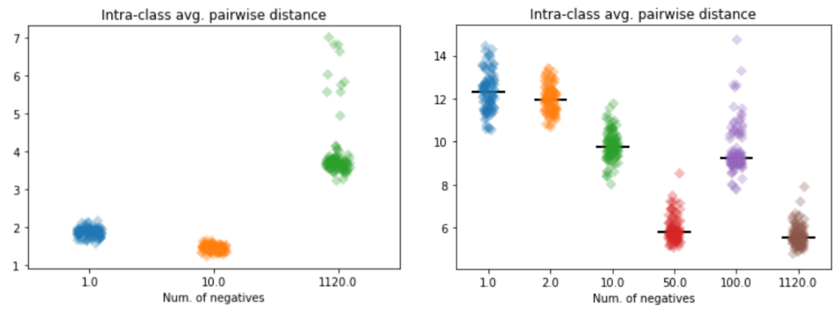
(a) SimCLR on CIFAR100 (b) RELIC on CIFAR100 (c) RELIC on ImageNet

Figure 1: (a) and (b) Top 1 linear test accuracy on CIFAR100 as a function of the number of negative examples. *stand* denotes the standard approach of using all available negative examples in the batch. We report the mean and standard deviation (error bars) over 5 random seeds. (c) Top 1 linear test accuracy on ImageNet for RELIC, for each number of negative examples we report the best performance across across hyperparameters.

We also examined this phenomenon for RELIC and SimCLR on the ImageNet dataset [12] after downsampling the images to 64×64 pixels. We followed the same protocol and experimental setup as for CIFAR100 apart from the batch size which we set to be 4096. In Figure 1c we see that the best performance for RELIC is achieved with 100 negatives which is significantly lower than what the full batch size of 4096; top 1 test accuracy with 100 negatives examples is 0.7% higher than under the standard approach to using negatives that utilizes all the remaining points in the batch as negatives. For SimCLR, we found that we could achieve the same performance as the standard approach utilizing all the remaining points in the batch as negatives with only 75 negatives. While not leading to improved performance, using fewer negatives in SimCLR can help reduce the computational cost significantly.

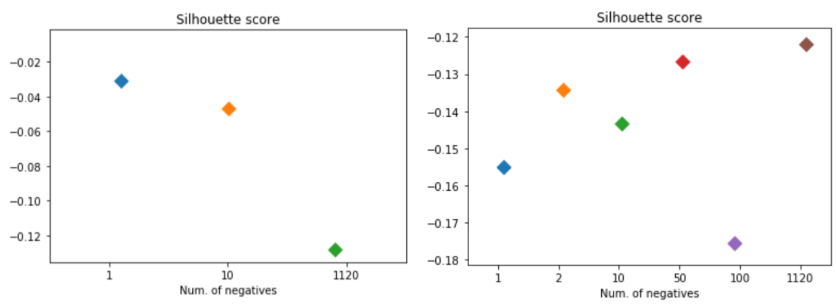
Hypothesis: Better intra-class concentration

We first hypothesized that the improved performance with fewer negatives was due to better intra-class concentration of the learned representations. Better intra-class concentration would make the learned latent space of representations more easily linearly separable and thus directly lead to improved classification performance under the linear evaluation protocol. To evaluate intra-class concentration we used two measures – the intra-class average pairwise distance between representations and the Silhouette score. The Silhouette score is calculated using the mean intra-cluster distance (a) and the mean nearest-cluster distance (that the datapoint is not part of) (b) and computed as $(b-a)/max(a, b)$. Note that higher Silhouette score indicates better intra-class concentration. For RELIC both the intra-class average pairwise distance and the Silhouette score indicate that for lower numbers of negatives there is a better intra-class concentration; see Figures 2a and 3a. For SimCLR, neither the intra-class average pairwise distances nor the Silhouette scores do strongly correlate with downstream performance; see Figure 2b and Figure 3b.



(a) RELIC (b) SimCLR

Figure 2: We compare the intra-class average pairwise distance for different numbers of negatives for RELIC and SimCLR on CIFAR100. The horizontal bars denote the median.



(a) RELIC (b) SimCLR

Figure 3: We compare the Silhouette scores for different numbers of negatives for RELIC and SimCLR on CIFAR100.

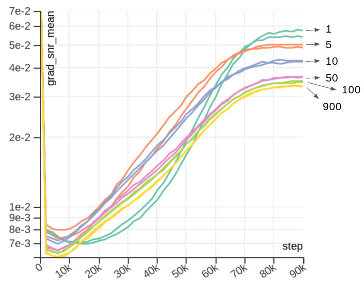
Hypothesis: Better gradient signal-to-noise ratio

Could the surprising performance obtained with fewer negative examples be partially explained by better gradient dynamics? Gradient variance alone is not always a good indicator since gradients with different magnitudes will be affected differently by the same amount of noise. We therefore focus on the gradient signal-to-noise ratio (SNR) which has been established as a useful measure to assess training dynamics [11]. SNR is defined for each model parameter θ_i as the absolute value of the gradient mean divided by its standard deviation:

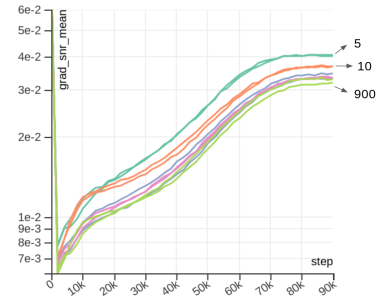
$$\mathrm{SNR}_{\theta_i} = \frac{|\mathbb{E}\left(\nabla_{\theta_i}\mathcal{L}\right)|}{\mathrm{std}\left(\nabla_{\theta_i}\mathcal{L}\right)},$$

where $\nabla_{\theta_i}\mathcal{L}$ denotes the gradient of the loss $\mathcal{L}$ w.r.t. θ_i. As we are dealing with a large number of model parameters and are primarily interested in the evolution of gradient SNR during training, we report both the SNR average over parameters $\frac{1}{\#\theta}\sum_i \mathrm{SNR}_{\theta_i}$ and its standard deviation. We computed SNR by using an exponentially moving average to estimate the first two moments of the gradient during training. Figure 4 shows how these quantities evolve during training on CIFAR100. We observe similar dynamics for SimCLR and RELIC. With a small number of negative examples, the SNR variance is often higher (see bottom panels) but the average SNR is clearly improved (see top panels). This is particularly noticeable for very small numbers of negatives such as 5 and 10. Using only 1 negative example seems to be a special case and the gradient behaviour looks qualitatively different.[1] From these findings, we conclude that the improvement in performance observed when

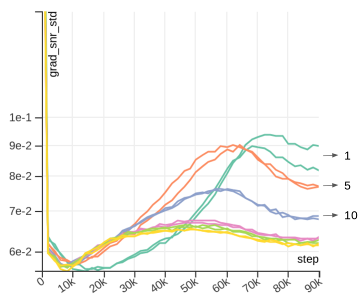
[1]Dynamics for SimCLR with 1 negative example are still to be computed.



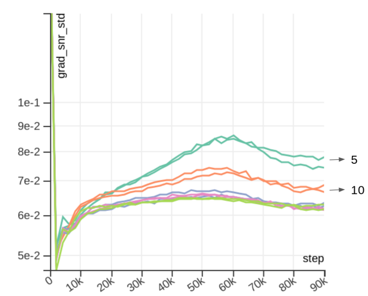
(a) RELIC gradient SNR mean, for different numbers of negatives:1 (green), 5 (orange), 10 (blue), 50 (pink), 100 (light green), 900 (yellow), and two random seeds.



(b) SimCLR gradient SNR mean, for different numbersrs of negatives:5 (dark green), 10 (orange), 50 (blue), 100 (pink), 900 (light green), and two random seeds.



(c) RELIC gradient SNR standard deviation.



(d) SimCLR gradient SNR standard deviation.

Figure 4: Gradient signal-to-noise (SNR) for training on CIFAR100 with the temperature of 1.0 and learning rate of 1.2. The top panels show the evolution of SNR average over all model parameters during training. The bottom panels show the standard deviation of SNR across model parameters. Each experiment was conducted for two random seeds.

using fewer negatives is at least in part due to an improvement in the signal-to-noise ratio in the gradients, i.e. using fewer negatives yields improved gradient dynamics.

4 Conclusion

In this work we examined the role of negative examples in contrastive learning. Specifically, we examined the widely held belief that using as many negatives as possible improves the quality of the learned representation. We decoupled the batch size from the number of negatives in order to perform experiments with fixed batch size and varying number of negatives for two very performant contrastive methods SimCRL and ReLIC. For both methods, we observe that performance increases as the number of negatives decreases, which strongly suggests that the heuristic of using as many negatives as possible does not lead to the best performance. We also notice increasing gradient SNR with decreasing number of negatives and hypothesize that this is one of the causes of improved performance. Finally, together with previous results, our findings point towards large batch sizes and not the number of negatives having a decisive influence on performance.

References

[1] Anonymous. Representation learning via invariant causal mechanisms. In *Submitted to International Conference on Learning Representations*, 2021. under review.

[2] P. Bachman, R. D. Hjelm, and W. Buchwalter. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*, pages 15509–15519, 2019.

[3] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[4] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[6] O. J. Hénaff, A. Razavi, C. Doersch, S. Eslami, and A. v. d. Oord. Data-efficient image recognition with contrastive predictive coding. *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[7] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

[8] A. Kolesnikov, X. Zhai, and L. Beyer. Revisiting self-supervised visual representation learning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1920–1929, 2019.

[9] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. 2009.

[10] A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[11] T. Rainforth, A. R. Kosiorek, T. A. Le, C. J. Maddison, M. Igl, F. Wood, and Y. W. Teh. Tighter variational bounds are not necessarily better, 2019.

[12] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252, 2015.

[13] Y. You, I. Gitman, and B. Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017.