Branching Out for Better BYOL

Azad Singh, Deepak Mishra
Department of Computer Science and Engineering
Indian Institute of Technology Jodhpur
{singh.63, dmishra} @iitj.ac.in

Abstract

BYOL [7] leads migration of self-supervised learning techniques from contrastive to non-constative paradigm. Non-constative techniques do not require negative pairs to learn meaningful representations from the data. Their success mainly depends on the stochastic composition of data augmentation techniques and the siamese configuration of deep neural networks. However, BYOL in its original form is limited to only two augmented views per training cycle. This motivates us to extend BYOL from a single target network branch to multiple branches, which enables simultaneous analysis of multiple augmented views of an input image. Increasing branches of the target networks marginally increase the computational cost as each branch is updated only using exponential moving average of online network's parameters. We demonstrate superior performance of Multi-Target BYOL on several vision datasets by evaluating the representations learned by the online network using linear evaluation protocols.

1 Introduction

Recent advancements in self-supervised learning faclitate effective usage of unlabelled data. Contrastive self-supervised methods like SimCLR [2], MoCo [8], PIRL [15], and InfoMin [19] achieve performances comparable to supervised methods [11; 10; 14]. These methods use contrastive loss, to reduce distance between embeddings of different instances of same image while increasing the distance between embeddings of the different images. However, the contrastive paradigm of self-supervised learning heavily depends on the availability of large negative samples to avoid representational collapses, making them computationally expensive. BYOL provides a non-contrastive approach that mitigates the inherent computational constraint imposed by contrastive methods. Unlike SimCLR and MoCo, it directly minimizes the mean squared error between embeddings generated from two different augmented views of an image and achieves comparable performances to its contrastive counterparts. BYOL relies on stochastic composition of data augmentation techniques and siamese configuration of deep neural networks [17; 18]. Aggressive data augmentation strategy creates numerous samples from a single image which are subsequently processed by siamese network. The two brances of siamese network in BYOL are called as online and target network.

Inspired by this, we extended the BYOL from a single target network to multiple target networks and named it as Multi-Target BYOL (MT-BYOL). The stochastic composition of the data augmentations also increases with each branch of the target network. In particular, MT-BYOL trains to predict the representations generated by the multiple target networks from multiple augmented views of an image. We hypothesize that this helps in better regularization of online network and enables faster convergence. In summary, our contributions are as follows: i) we extend BYOL from a single target network for an online network to multiple target networks for simultaneous processing of multiple augmented views of an image; ii) we show that MT-BYOL achieves considerably better performance as compared to BYOL with only marginally increasing of total computational cost; iii) we empirically show that Multi-Target BYOL is relatively more resilient to changes in batch size: iv) we evaluate

NeurIPS 2021 Workshop: Self-Supervised Learning - Theory and Practice

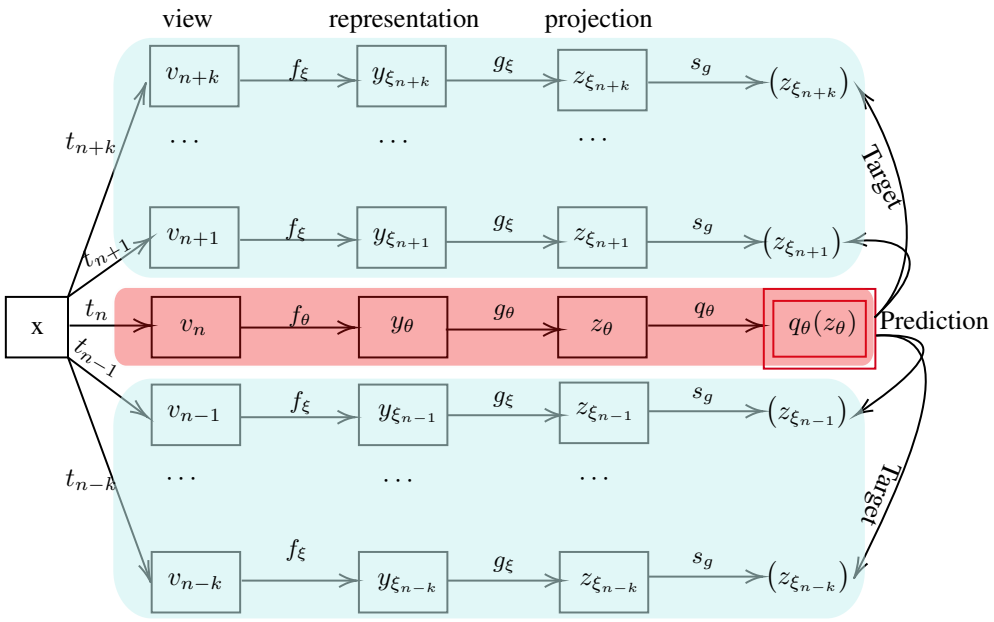


Figure 1: Overview of the proposed MT-BYOL, having multiple target network branches for an online network. $(t_{n+k}, ..., t_{n+1}, t_n, t_{n-1}, ..., t_{n-k})$ are the stochastic compositions of augmentations which create $(v_{n+k}, ..., v_{n+1}, v_n, v_{n-1}, ..., v_{n-k})$ augmented views of image x. θ are the trainable parameters of online network while ξ is updated as EMA of θ. MT-BYOL minimizes the cross-model similarity loss between representations generated by online network and target networks. s_g represents the stop-gradient operator.

the representations learned by MT-BYOL under the linear evaluation protocols on various computer vision datasets and report the corresponding results.

2 Related Work

2.1 Contrastive Self-Supervised learning

Most contrastive learning methods use variants of InfoNCE [16] objective to learn embeddings, as:

$$L_{InfoNCE} = -\log \frac{\exp\left(h\left({z_i}^T z_j/\tau\right)\right)}{\sum_{k=1,k\neq i}^{K} \exp\left(h\left({z_i}^T z_k/\tau\right)\right)} \tag{1}$$

where (z_i, z_j) forms the positive pair and (z_i, z_k) forms the negative pairs of input data points and τ is the temperature hyperparameter. The summation in the denominator is over all the negative pairs, which depends upon the current batch size K. Positive pairs are created from the different augmented views $(v, v^{'})$ of the same image while the negative pairs are formed using the different augmented views of different images. The loss function encourages the network to bring embeddings from from views of the same image closer while that from different image apart. $h(z_i, z_j)$ measure the similarity between z_i and z_j. The contrastive learning methods [7; 2; 8; 1] require a large pool of negative samples, which makes them computationally expensive [3; 4].

2.2 Non-Contrastive Self-Supervised learning

Non-contrastive self-supervised methods do not depend upon the direct sampling of negative pairs to train the networks. Recent methods like BYOL, SimSiam [5] rely only on positive pairs, where the online network predicts the embeddings generated by target network. BYOL minimizes the similarity

loss between projected embeddings from different views of an image, using the following loss term

$$L(\theta) = -\frac{\left\langle q_\theta(z_\theta), z'_\xi \right\rangle}{\|q_\theta(z_\theta)\|_2 \cdot \left\| z'_\xi \right\|_2}. \tag{2}$$

Where θ represents the parameters of online network and updated by L(θ). Target network parameterized by ξ get updates from an exponential moving average (EMA) of the θ i.e. $(\xi \leftarrow (1-\eta)\theta + \eta\xi)$ where η is the decay parameter. $q_\theta(z_\theta)$ is output from prediction head of online network for view v while z'_ξ represents embedding produced by target network for view v' of the input image x. SimSiam is another non-constrastive self-supervised method and is same as BYOL without EMA.

3 Method

We extend BYOL to MT-BYOL, which uses multiple target networks for a single online network, which allows simultaneous processing of multiple views of the given image. With each branch of target network, the stochastic composition of data augmentation also increases by multifold. Like BYOL, the online network in MT-BYOL, parametrized by θ, has three stages: a backbone network f_θ, an MLP projection head g_θ, and finally another MLP prediction head q_θ, as shown in Figure 1. All target networks have the same architecture as the online network except the prediction head. The target networks are parametrized by their respective weights ξ_i, where i $\in [n-k, n+k]$. Prediction head of online network predicts the regression targets ($z_{\xi_{n-k}}$,$z_{\xi_{n+k}}$), generated from different augmented views $(v_{n+k}, ..., v_{n+1}, v_n, v_{n-1}, ..., v_{n-k})$ of an input image x by the target networks. MT-BYOL generates these different augmented views by applying augmentations $(t_{n-k}, ..., t_{n+1}, t_n, t_{n-1}, ..., t_{n-k})$ respectively by sampling from a set of standard augmentations. MT-BYOL calculates the similarity score between embeddings $q_\theta(z_\theta)$ and z_{ξ_i} to train the online network. All target networks follow the online network and their respective parameters ξ_i are the EMA of θ. The target networks are different from each other due to η employed in EMA.

Particularly for a given image x sampled uniformly from the set of images D, MT-BYOL generates $v_n \triangleq t_n(x)$ by applying respective augmentations. v_n is then passed to the online network which, outputs $q_\theta(z_\theta)$. From the other augmented views $(v_{n+k}, ..., v_{n+1})$, $(v_{n-1}, ..., v_{n-k})$ different target networks generate embeddings z_{ξ_i} respectively from their projection heads. Similarity score is than calculated between the l_2 normalized $\overline{q}_\theta(z_\theta)$ and $\overline{z}_{\xi_i}$, where $\overline{q}_\theta(z_\theta) \triangleq q_\theta(z_\theta)/\left\|q_\theta(z_\theta)\right\|_2$ and $\overline{z}_{\xi_i} \triangleq z_{\xi_i}/\left\|z_{\xi_i}\right\|_2$ using the loss defined in equation (3).

$$L_\theta^{MT} = \frac{1}{2k} \sum_{i \in n-k, i \neq n}^{n+k} \left\| \overline{q}_\theta({z_\theta}^n) - {\overline{z}_\xi}^i \right\|_2^2 \tag{3}$$

Number of loss pairs depends upon number of target branches employed in the model and defines as $k^2(k+1)$, where k is number or target networks and 1 refers to single online network.

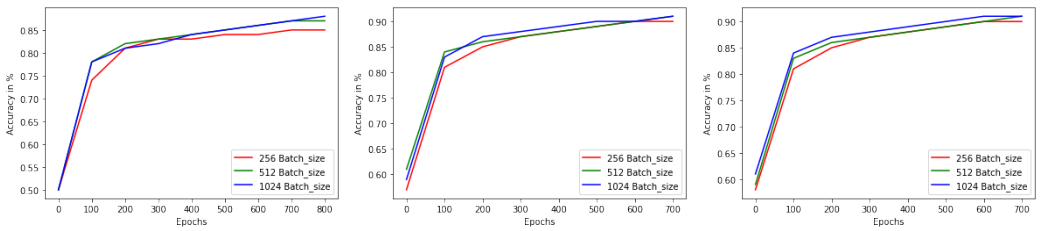
We use symmetric form the loss L_θ^{MT} defined in equation (3) based on the cross-model pairs from different representations corresponding to different augmented views. The parameters θ of the online network get updated by minimizing the L_θ^{MT} using stochastic optimization technique. As stated earlier parameters of the target networks are updated by $\xi_i \leftarrow (1-\eta)\theta + \eta\xi_i$ where $\eta \in [0, 1]$ is the decay rate. Downstream tasks are performed by train a linear classifier which uses embeddings generated by f_θ after freezing θ.

4 Implementation and Experimental details

We trained MT-BYOL upto 3 target network branches for the CIFAR10 [12], CIFAR100 [12], STL-10 [6] and Tiny-Imagenet [13] datasets, with same set of augmentations as in the SimCLR. For CIFAR-10 we use ResNet-18 [9] as backbone network and for other datasets we use Resnet-50 [9]. The hidden layer and output dimensions in both projection and prediction heads are 4096 with 256, respectively. We train MT-BYOL for 800 epcohs using LARS [20] optiization technique with batch sizes of 256, 512 and 1024. We use base learning rate of 0.02 with cosine decay in all our experiments. Effective learning rate depends upon the batch size, which is calculated while training by dividing the base

Table 1: Test set classification accuracy of linear classifier evaluated on embeddings generated by the frozen encoder for different datasets.

Method	**BYOL**			**MT-BYOL(2)**			**MT-BYOL(3)**		
Batchsize	256	512	1024	256	512	1024	256	512	1024
CIFAR10	85.46	87.59	88.34	90.29	90.51	91.31	90.64	91.19	**91.56**
CIFAR100	62.21	63.29	64.78	66.11	66.72	67.21	66.38	67.47	**67.58**
STL10	87.31	88.48	89.72	91.11	92.23	92.37	91.73	92.67	**92.71**
Tiny-ImageNet	54.46	55.79	56.63	56.54	56.78	57.12	56.73	57.03	**57.43**



(a) BYOL (b) MT-BYOL(3) (c) MT-BYOL(3)

Figure 2: Convergence plots of BYOL and MT-BYOL(3) on CIFAR10. (a) Original BYOL (b) MT-BYOL(3) with same η for all three target network branches (c) MT-BYOL(3) with different values of η.

learning rate with batch size. Different branches of target networks follow the online network using EMA of θ with different initial values of η. As the training progress η approaches to 1.

To evaluate the quality of learned representations we follow the same pipeline as in [7; 2; 8; 15; 19]. We train linear layer model on frozen representations generated by online encoder. We report test set accuracy on different benchmark datasets in Table 1. We use BYOL with aforementioned settings as our baseline which is identical to MT-BYOL with a single target network branch. MT-BYOL(2) and MT-BYOL(3) refer to BYOL with 2 and 3 target networks branches, respectively. Experiments show that multiple target network branches considerably improve BYOL's performance across all the datasets. We see marginal improvements in performance with MT-BYOL(3) over MT-BYOL(2), therefore we restrict our experiments with 3 target network branches. With 2 target network branches, 12 cross-model views are generate which provide enough regularization to the online network. To better understand the behaviour of MT-BYOL, we compare the performance of BYOL with MT-BYOL(3) with varying batch sizes and different initial values of η and plot the observations in Figure 2. Figure 2(a) shows the convergence plot of BYOL. Figure 2(b) shows the performance of MT-BYOL(3) with identical initial value of η for all the target network branches while figure 2(c) have the plot for different values $\eta's$ corresponding to different branches of target networks. We observe effect of variation in intital values of η is marginal whic provid ethat empirical evidences, that multiple augmented views are the major factor in the performance of MT-BYOL. Further it is important to note that both Table 1 and Figure 2 shows that MT-BYOL is less sensitive to variations in batch size. A limitation of MT-BYOL, which we believe is inherited from BYOL, is when number of images per class is low, the performance is also limited.

5 Conclusion

In this work we purposed MT-BYOL, which have multiple branches of target network. In MT-BYOL online network is forced to match the outcomes of different target network branches, which use multiple augmented views of an input image. This creates several pairs of augmented views and corresponding losses to realize a strong regularization effect on the online network and enables the backbone encoder to learn effective representations. Multiple branches of target network cost marginal computational overhead as their parameters are estimated by EMA of online network. Further we empirically observe resilience of MT-BYOL towards variations in batch size. In future we will explore the effect of further increasing the branches and other augmentation techniques.

References

[1] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *arXiv preprint arXiv:1906.00910*, 2019.

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[3] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020.

[4] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[5] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

[6] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

[7] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[8] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9[1]:2, 2021.

[11] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[12] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[13] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7[7]:3, 2015.

[14] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[15] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020.

[16] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[17] Attaullah Sahito, Eibe Frank, and Bernhard Pfahringer. Semi-supervised learning using siamese networks. In *Australasian Joint Conference on Artificial Intelligence*, pages 586–597. Springer, 2019.

[18] Chon Hou Sio, Yu-Jen Ma, Hong-Han Shuai, Jun-Cheng Chen, and Wen-Huang Cheng. S2siamfc: Self-supervised fully convolutional siamese network for visual tracking. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1948–1957, 2020.

[19] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *arXiv preprint arXiv:2005.10243*, 2020.

[20] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017.