Bootstrap Your Own Latent (BYOL)
A New Approach to Self-Supervised Learning

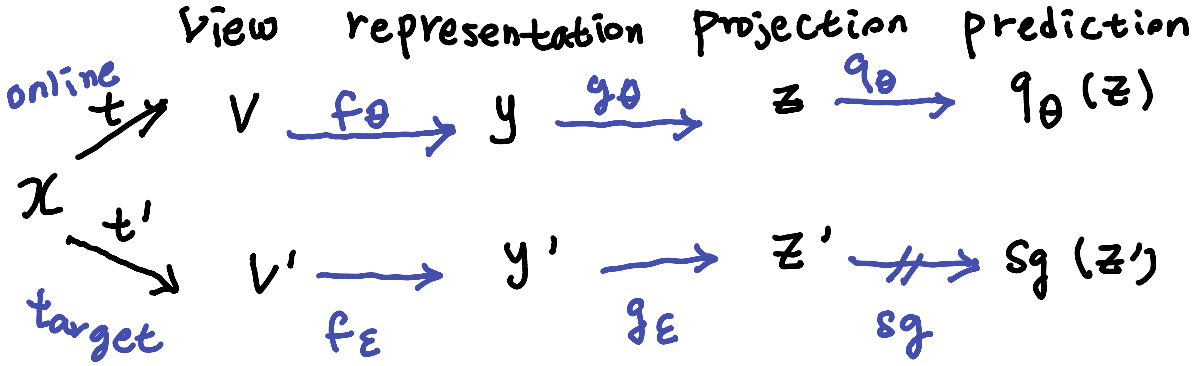
0. Abstract

- A new approach to self-supervised image representation learning
- online and target networks interact and learn from each other
- from an augmented view of an image, we train the online network to predict the target network representation of the same image under different augmented view
- update the target network with a slow-moving average of the online network
- achieves a new state of the art without negative pairs.

1. Introduction

- how to learn good image representations?
- state-of-the-art contrastive methods : by reducing distance between positive pairs, and increasing the distance between negative pairs.
- BYOL achieves higher performance without using negative pairs.
- iteratively bootstraps the output of a network to serve as targets for an enhanced representation
- more robust tot he choice of image augmentations than contrastive methods
- start from an augmented view of an image
- train its online network to predict the target network's representation of another augmented view of the same image
- use slow-moving average of the online network as the target network

Figure 1. BYOL's architecture



minimize similarity loss $q_\theta(z)$ and $sg(z')$

θ trained weights

ξ exponential moving average of θ

sg stop-gradient

everything but f_θ is discarded at the end of training

y is used as image representation

3. Method

- Is negative examples indispensable to prevent collapsing?
- from a given representation (target), we can train a new, potentially enhanced representation (online), by predicting the target representation
- use subsequent online networks as new target networks for further training
- use a slowly moving exponential average of the online networks as the target network instead of fixed checkpoints.
- In deep RL, target networks stabilize the bootstrapping updates provided by the Bellman equation

3.1 Description of BYOL

- goal: learn a representation y
- which can then be used for downstream tasks
- online network: defined by a set of weights θ
- encoder f_θ
  projector g_θ
  predictor q_θ
- target network: defined by a set of weights ξ
- provides the regression targets to train the online network

ξ : exponential moving average of θ

τ : target decay rate $\in [0,1]$

after each training step, we perform the update

$$\xi \leftarrow \tau\xi + (1-\tau)\theta$$

procedure of BYOL

① T, T' : Two distributions of image augmentations

$v = t(x) \rightarrow y = f_\theta(v) \rightarrow z_\theta = g_\theta(y) \rightarrow q_\theta(z_\theta)$

$v' = t'(x) \rightarrow y' = f_\xi(v') \rightarrow z'_\xi = g_\xi(y')$

② l_2-normalize

$\overline{q_\theta}(z_\theta) = q_\theta(z_\theta) / \|q_\theta(z_\theta)\|_2$

$\overline{z'_\xi} = z'_\xi / \|z'_\xi\|_2$

③ define mean squared error between $\overline{q_\theta}(z_\theta)$ and $\overline{z'_\xi}$

$$L_\theta^{BYOL} = \|\overline{q_\theta}(z_\theta) - \overline{z'_\xi}\|_2^2 = 2 - 2\frac{\langle q_\theta(z_\theta), z'_\xi \rangle}{\|q_\theta(z_\theta)\|_2 \cdot \|z'_\xi\|_2}$$

④ symmetrize the loss L_θ^{BYOL} by separately feeding v' to online network and v to the target network to compute $\tilde{L}_\theta^{BYOL}$

⑤ at each training step, perform Stochastic optimization step to minimize $L_\theta^{BYOL} + \tilde{L}_\theta^{BYOL}$ respect to θ

⑥ only keep the encoder f_θ

Appendix A. Algorithm

Input:

D, T, T' set of images and distributions of transformations

$\theta, f_\theta, g_\theta, q_\theta$ initial online parameters, encoder, projector, predictor

ξ, f_ξ, g_ξ initial target parameters, target encoder, target projector

optimizer updates online parameters using the loss gradient

K, N total number of optimization steps and batch size

$\{\tau_k\}_{k=1}^{K}$ $\{\eta_k\}_{k=1}^{K}$ target network update schedule, learning rate schedule

for $k=1$ to K do

$B \leftarrow \{x_i \sim D\}_{i=1}^{N}$ sample a batch of N images

for $x_i \in B$ do

$t \sim T, t' \sim T'$ sample image transformations

$z_1 \leftarrow g_\theta(f_\theta(t(x_i))), z_2 \leftarrow g_\theta(f_\theta(t'(x_i)))$ compute projections

$z_1' \leftarrow g_\xi(f_\xi(t'(x_i))), z_2' \leftarrow g_\xi(f_\xi(t(x_i)))$

$l_i \leftarrow -2 \cdot \left(\frac{\langle q_\theta(z_1), z_1' \rangle}{\|q_\theta(z_1)\|_2 \cdot \|z_1'\|_2} + \frac{\langle q_\theta(z_2), z_2' \rangle}{\|q_\theta(z_2)\|_2 \cdot \|z_2'\|_2} \right)$

end

$\delta\theta \leftarrow \frac{1}{N} \sum_{i=1}^{N} \partial_\theta l_i$ compute the total loss gradient

$\theta \leftarrow \text{optimizer}(\theta, \delta\theta, \eta_k)$ update online parameters

$\xi \leftarrow \tau_k \xi + (1 - \tau_k)\theta$ update target parameters

output: encoder f_θ

3.2 Implementation details

1) Image augmentations
- use the same set of image augmentations as SimCLR
- resized to 224 x 224

2) Architecture
- base parametric encoder: convolutional residual network with 50 layers and post-activation (ResNet-50 (1x) v1)
- representation corresponds to the output of the final average pooling layer, which has feature dimension of 2048
- projector, predictor: MLP (linear layer with output size 4096 followed by batch normalization, ReLU, and a final linear layer with output dimension 256)

3) Optimization
- LARS optimizer with a cosine learning rate schedule over 1000 epochs
- LearningRate = 0.2 x BatchSize / 256
- a global weight decay parameter of 1.5 x 10 ** -6
- the exponential moving average parameter starts from 0.996 and is increased to one during training
- batch size 4096

4. Experimental evaluation
- BYOL's representation after self-supervised pretraining on the training set of the ImageNet ILSVRC-2012 dataset

1) Linear evaluation on ImageNet
- evaluate BYOL's representation by training a linear classifier on top of the frozen representation
- 74.3% top 1 accuracy, 91.6% top 5 accuracy

2) Semi-supervised training on ImageNet
- fine-tune BYOL's representation on a classification task with a small subset of ImageNet's train set, using label information
- consistently outperforms previous approaches

3) Transfer to other classification tasks

- evaluate representation on other classification dataset to assess whether the features learned on ImageNet are generic
- perform linear evaluation and fine-tuning on the set of classification task with dataset CIFAR, SUN397, VOC2007 etc.
- outperforms SimCLR on all benchmarks

4) Transfer to other vision tasks

- evaluate representation on different tasks (semantic segmentation, object detection, depth estimation)
- assess whether BYOL's representation generalizes beyond classification tasks
- VOC2012 semantic segmentation task: classify each pixel in the image
- object detection using a Raster R-CNN architecture: fine tuen on trainval2007, report results on test2007
- depth estimation on NYU v2 dataset: depth map of a scene is estimated given a single RGB image
- Depth prediction measures how well a network represents geometry, and how well that information call be localized to pixel accuracy