# Bootstrap Your Own Latent
# A New Approach to Self-Supervised Learning

School of Industrial and Management Engineering, Korea University

Hansam Cho

# Contents

# Research Purpose

❖ Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning (2020, NeurIPS)

- DeepMind 소속의 저자, 2022년 06월 06일 기준으로 1438회 인용

## Bootstrap Your Own Latent A New Approach to Self-Supervised Learning

Jean-Bastien Grill[*,1], Florian Strub[*,1], Florent Altché[*,1], Corentin Tallec[*,1], Pierre H. Richemond[*,1,2]
Elena Buchatskaya[1], Carl Doersch[1], Bernardo Avila Pires[1], Zhaohan Daniel Guo[1]
Mohammad Gheshlaghi Azar[1], Bilal Piot[1], Koray Kavukcuoglu[1], Rémi Munos[1], Michal Valko[1]

[1]DeepMind [2]Imperial College

[jbgrill,fstrub,altche,corentint,richemond]@google.com

### Abstract

We introduce Bootstrap Your Own Latent (BYOL), a new approach to self-supervised image representation learning. BYOL relies on two neural networks, referred to as *online* and *target* networks, that 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 a different augmented view. At the same time, we update the target network with a slow-moving average of the online network. While state-of-the art methods rely on negative pairs, BYOL achieves a new state of the art *without them*. BYOL reaches 74.3% top-1 classification accuracy on ImageNet using a linear evaluation with a ResNet-50 architecture and 79.6% with a larger ResNet. We show that BYOL performs on par or better than the current state of the art on both transfer and semi-supervised benchmarks. Our implementation and pretrained models are given on GitHub.[3]





DMQA hcai

# Research Purpose

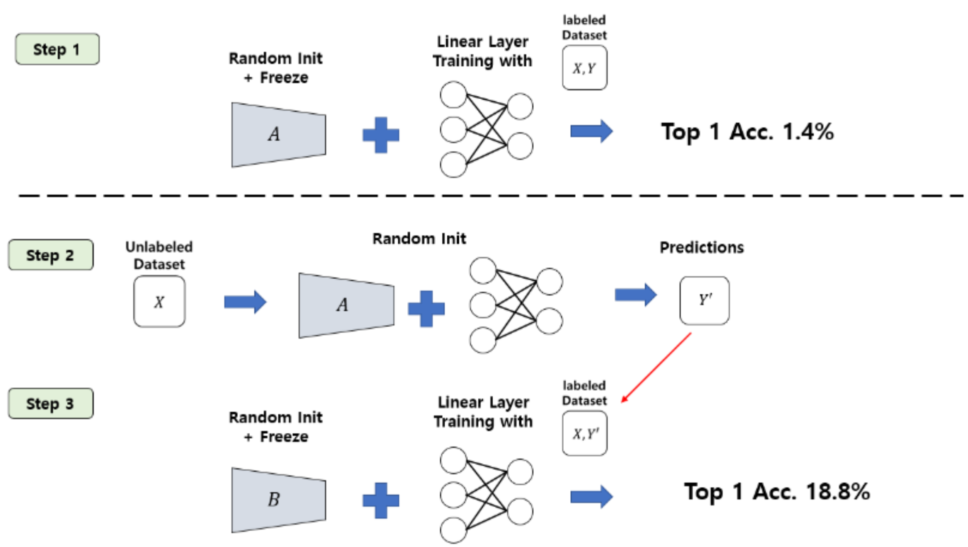
❖ Introduction

- 좋은 representation을 학습하는 것은 중요한 task이며 contrastive learning이 좋은 성능을 보임
- Contrastive learning은 negative pair의 정의와 augmentation 기법에 민감함
- Positive pair만을 활용하는 BYOL이라는 새로운 self-supervised learning 기법을 제안함

# Methods

❖ Motivation

- ImageNet 데이터 활용한 classification 성능 비교
- 실험1: Random representation을 활용, linear classifier 학습 → Top1 Acc. 1.4%
- 실험2: Random representation을 예측하도록 encoder 학습 후 해당 encoder를 고정, linear classifier 학습 → Top1 Acc. 18.8%
- Representation을 예측하는 것 만으로도 좋은 representation 학습 가능

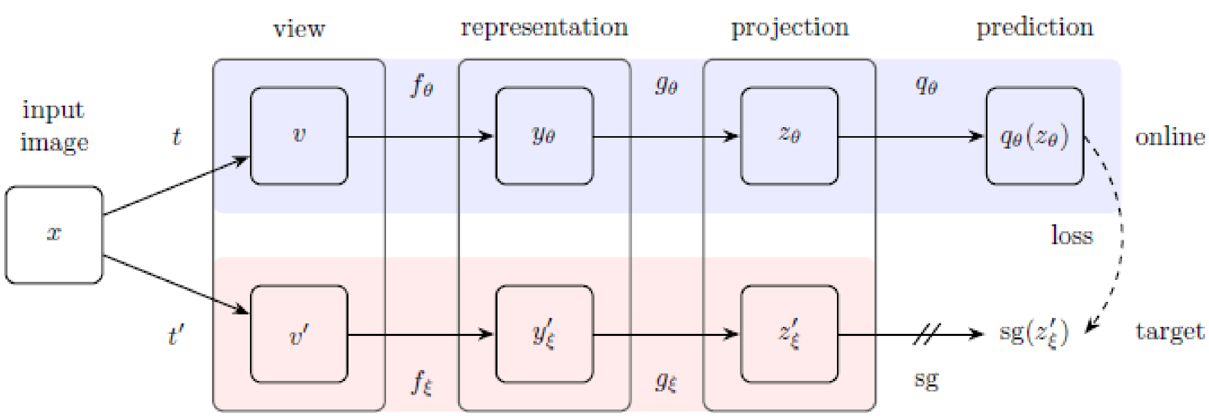



https://hoya012.github.io/blog/byol/

# Methods

❖ BYOL

- Online network: 실제 representation을 학습하는 모델
- Target network: Online network가 학습할 target값을 생성하는 모델
- Online network에만 predictor가 존재 → 모델의 비대칭성을 통해 서로 다른 representation 생성
- 서로 다른 augmentation $(t, t')$을 활용해 두개의 이미지 생성 $(v, v')$ 후 입력
- Online network와 target network의 output에 대해 정규화




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

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



DMQA hcai

# Methods

❖ BYOL

- MSE를 loss로 활용

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

- 입력 이미지를 바꾼 경우에 대해서도 loss 계산, 두 loss를 더해서 최종 loss로 활용

$$\mathcal{L}^{\mathtt{BYOL}}_{\theta,\xi} = \mathcal{L}_{\theta,\xi} + \widetilde{\mathcal{L}}_{\theta,\xi}$$

- Online network만 학습을 진행하며 target network는 online network의 exponential moving average로 파라미터 업데이트

$$\theta \leftarrow \text{optimizer}\left(\theta, \nabla_\theta \mathcal{L}^{\mathtt{BYOL}}_{\theta,\xi}, \eta\right) \quad \text{and} \quad \xi \leftarrow \tau\xi + (1-\tau)\theta,$$

# Methods

❖ Intuitions on BYOL's behavior

- BYOL은 collapse*를 막을 수 있는 장치가 없음 (*collapse: 모든 이미지에 대해서 동일한 representation 생성)
- 하지만 collapse가 일어나는 경우가 불안정하기 때문에 collapse가 발생하지 않음
- Online network predictor가 optimal인 상황을 가정

$$q_\theta = q^\star \text{ with } q^\star \triangleq \arg\min_q \mathbb{E}\left[\left\|q(z_\theta) - z'_\xi\right\|_2^2\right], \quad \text{where} \quad q^\star(z_\theta) = \mathbb{E}\left[z'_\xi | z_\theta\right],$$

- Online network update에 사용되는 gradient는 conditional variance에 영향을 받음

$$\nabla_\theta \mathbb{E}\left[\left\|q^\star(z_\theta) - z'_\xi\right\|_2^2\right] = \nabla_\theta \mathbb{E}\left[\left\|\mathbb{E}\left[z'_\xi | z_\theta\right] - z'_\xi\right\|_2^2\right] = \nabla_\theta \mathbb{E}\left[\sum_i \mathrm{Var}(z'_{\xi,i} | z_\theta)\right]$$

- Collapse 상황의 variance가 일반적인 상황의 variance에 비해 큼

$$\mathrm{Var}(z'_\xi | z_\theta) \leq \mathrm{Var}(z'_\xi | c)$$

- 따라서 학습과정 중 collapse 상황을 쉽게 벗어날 수 있음

# Experiments

❖ Linear evaluation on ImageNet

- BYOL은 online network encoder를 활용해 representation 생성
- Encoder를 고정 후 생성된 representation을 활용해 liner classifier 학습
- BYOL이 다른 방법론에 비해 좋은 성능을 보임

| Method | Top-1 | Top-5 |
|---|---|---|
| Local Agg. | 60.2 | - |
| PIRL [35] | 63.6 | - |
| CPC v2 [32] | 63.8 | 85.3 |
| CMC [11] | 66.2 | 87.0 |
| SimCLR [8] | 69.3 | 89.0 |
| MoCo v2 [37] | 71.1 | - |
| InfoMin Aug. [12] | 73.0 | 91.1 |
| BYOL (ours) | **74.3** | **91.6** |

(a) ResNet-50 encoder.

| Method | Architecture | Param. | Top-1 | Top-5 |
|---|---|---|---|---|
| SimCLR [8] | ResNet-50 (2×) | 94M | 74.2 | 92.0 |
| CMC [11] | ResNet-50 (2×) | 94M | 70.6 | 89.7 |
| BYOL (ours) | ResNet-50 (2×) | 94M | 77.4 | 93.6 |
| CPC v2 [32] | ResNet-161 | 305M | 71.5 | 90.1 |
| MoCo [9] | ResNet-50 (4×) | 375M | 68.6 | - |
| SimCLR [8] | ResNet-50 (4×) | 375M | 76.5 | 93.2 |
| BYOL (ours) | ResNet-50 (4×) | 375M | 78.6 | 94.2 |
| BYOL (ours) | ResNet-200 (2×) | 250M | **79.6** | **94.8** |

(b) Other ResNet encoder architectures.

Table 1: Top-1 and top-5 accuracies (in %) under linear evaluation on ImageNet.

# Experiments

❖ Semi-supervised training on ImageNet

- 일부 데이터의 label 정보만 활용한 semi-supervised 실험, BYOL의 성능이 가장 우수

| Method | Top-1 1% | Top-1 10% | Top-5 1% | Top-5 10% |
|---|---|---|---|---|
| Supervised [77] | 25.4 | 56.4 | 48.4 | 80.4 |
| InstDisc | - | - | 39.2 | 77.4 |
| PIRL [35] | - | - | 57.2 | 83.8 |
| SimCLR [8] | 48.3 | 65.6 | 75.5 | 87.8 |
| BYOL (ours) | **53.2** | **68.8** | **78.4** | **89.0** |

(a) ResNet-50 encoder.

| Method | Architecture | Param. | Top-1 1% | Top-1 10% | Top-5 1% | Top-5 10% |
|---|---|---|---|---|---|---|
| CPC v2 [32] | ResNet-161 | 305M | - | - | 77.9 | 91.2 |
| SimCLR [8] | ResNet-50 (2×) | 94M | 58.5 | 71.7 | 83.0 | 91.2 |
| BYOL (ours) | ResNet-50 (2×) | 94M | 62.2 | 73.5 | 84.1 | 91.7 |
| SimCLR [8] | ResNet-50 (4×) | 375M | 63.0 | 74.4 | 85.8 | 92.6 |
| BYOL (ours) | ResNet-50 (4×) | 375M | 69.1 | 75.7 | 87.9 | 92.5 |
| BYOL (ours) | ResNet-200 (2×) | 250M | **71.2** | **77.7** | **89.5** | **93.7** |

(b) Other ResNet encoder architectures.

❖ Transfer to other classification tasks

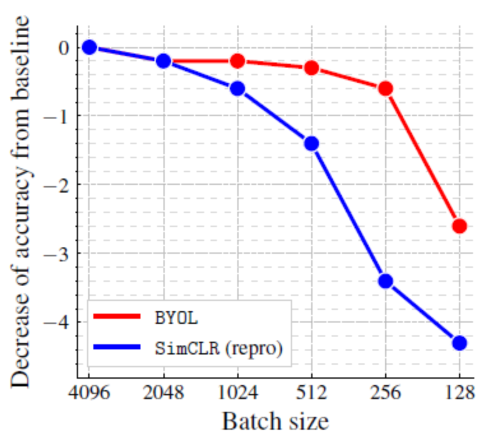
- 다양한 데이터에 적용할 수 있는 일반적인 representation을 생성할 수 있는지에 대한 실험
- Supervised learning을 통해 생성된 representation보다 좋은 성능을 보이기도 함

| Method | Food101 | CIFAR10 | CIFAR100 | Birdsnap | SUN397 | Cars | Aircraft | VOC2007 | DTD | Pets | Caltech-101 | Flowers |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Linear evaluation:* | | | | | | | | | | | | |
| BYOL (ours) | **75.3** | 91.3 | **78.4** | **57.2** | **62.2** | **67.8** | 60.6 | 82.5 | 75.5 | 90.4 | 94.2 | **96.1** |
| SimCLR (repro) | 72.8 | 90.5 | 74.4 | 42.4 | 60.6 | 49.3 | 49.8 | 81.4 | **75.7** | 84.6 | 89.3 | 92.6 |
| SimCLR [8] | 68.4 | 90.6 | 71.6 | 37.4 | 58.8 | 50.3 | 50.3 | 80.5 | 74.5 | 83.6 | 90.3 | 91.2 |
| Supervised-IN [8] | 72.3 | **93.6** | 78.3 | 53.7 | 61.9 | 66.7 | **61.0** | **82.8** | 74.9 | **91.5** | **94.5** | 94.7 |
| *Fine-tuned:* | | | | | | | | | | | | |
| BYOL (ours) | **88.5** | **97.8** | 86.1 | **76.3** | 63.7 | 91.6 | **88.1** | **85.4** | **76.2** | 91.7 | **93.8** | 97.0 |
| SimCLR (repro) | 87.5 | 97.4 | 85.3 | 75.0 | 63.9 | 91.4 | 87.6 | 84.5 | 75.4 | 89.4 | 91.7 | 96.6 |
| SimCLR [8] | 88.2 | 97.7 | 85.9 | 75.9 | 63.5 | 91.3 | 88.1 | 84.1 | 73.2 | 89.2 | 92.1 | 97.0 |
| Supervised-IN [8] | 88.3 | 97.5 | **86.4** | 75.8 | **64.3** | **92.1** | 86.0 | 85.0 | 74.6 | **92.1** | 93.3 | **97.6** |
| Random init [8] | 86.9 | 95.9 | 80.2 | 76.1 | 53.6 | 91.4 | 85.9 | 67.3 | 64.8 | 81.5 | 72.6 | 92.0 |

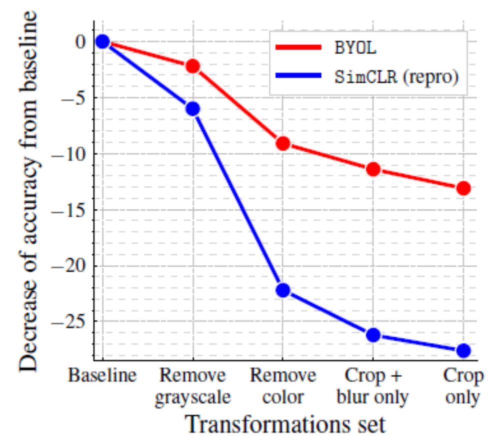
# Experiments

❖ Building intuitions with ablations

- SimCLR에 비해 batch size 크기, augmentation 기법에 대해 강건한 성능을 보임




(a) Impact of batch size




(b) Impact of progressively removing transformations



DMQA hcai

Thank You