Exploring Simple Siamese Representation Learning

[CVPR, 2021]

2023. 05. 19.

김성수

Data **M**ining and **Q**uality **A**nalytics

Data Mining
Quality Analytics

고려대학교
KOREA UNIVERSITY

논문소개

- 논문개요

❖ **Exploring Simple Siamese Representation Learning**

- 2021년 CVPR에 개제, FACEBOOK에서 연구 (2,111회 인용)
- 성능은 그렇게 높지는 않지만, Simple함을 강조 (Non-Contrastive Learning의 SimCLR)
- 방법론명: SimSiam

Exploring Simple Siamese Representation Learning

Xinlei Chen Kaiming He

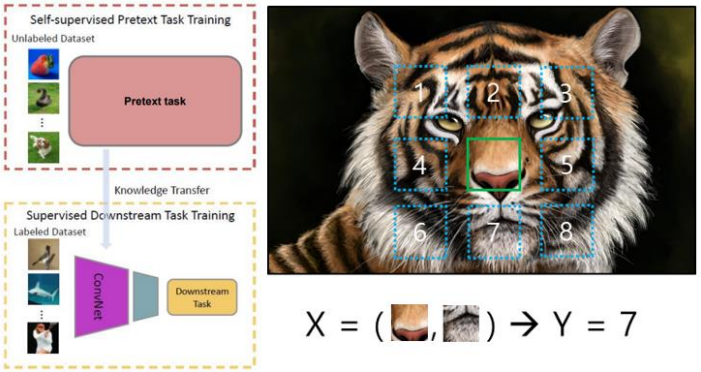
Facebook AI Research (FAIR)

연구배경

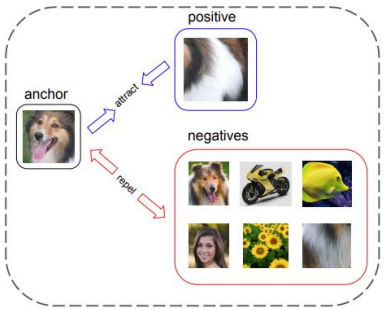
- Task명

❖ Task: Self-supervised Learning

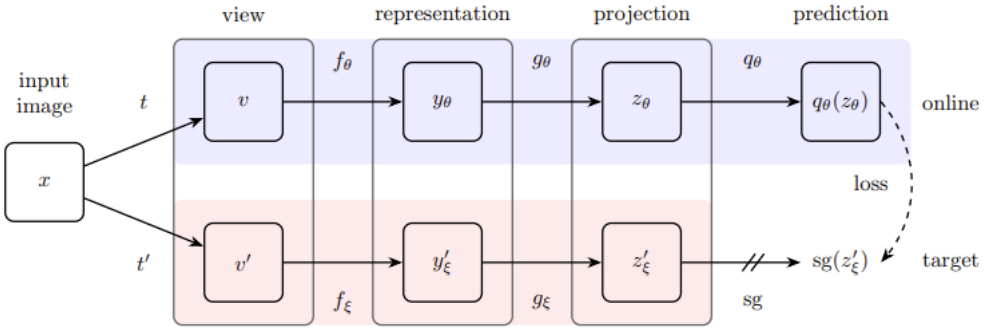
- Unlabeled 데이터만을 활용하여 풍부한 Representation을 학습
- Unlabeled 데이터는 Label을 갖고있지 않기에, 스스로 Supervised를 받을 수 있도록 학습



[Pretext Task]



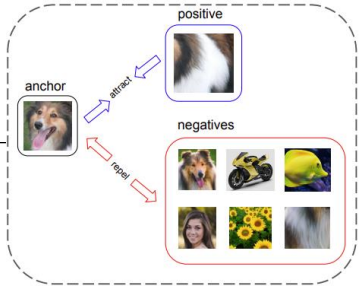
[Contrastive Learning]



[Non-Contrastive Learning]

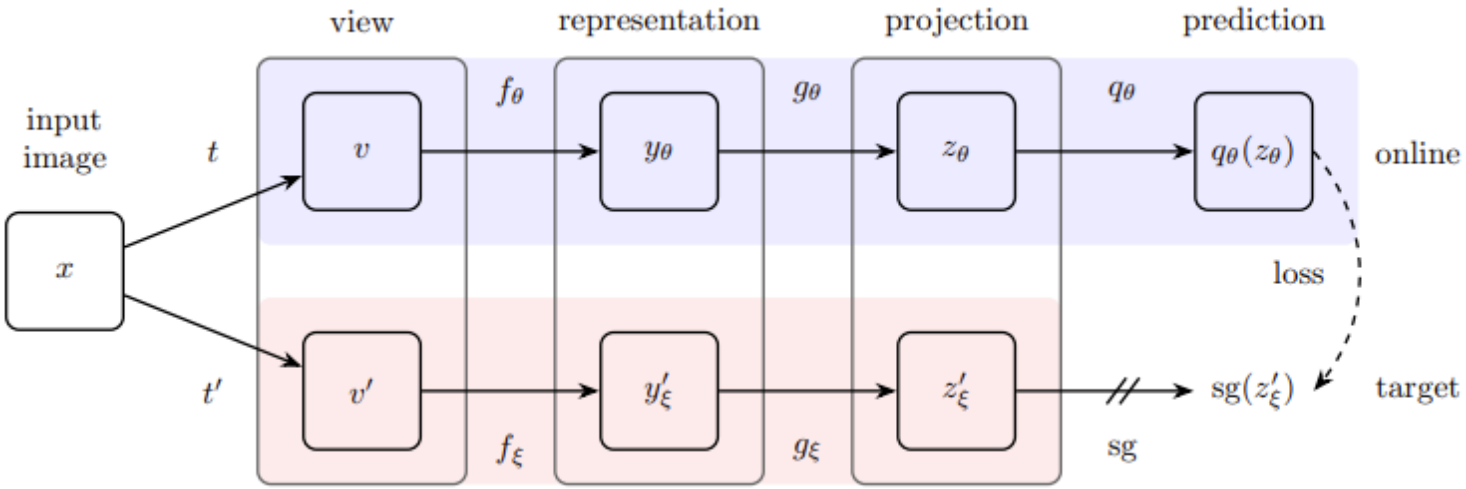
연구배경

- 선행연구의 한계



❖ Limitation of Previous Research

- 기존 Contrastive Learning은 Negative Sample을 필요로 하여 큰 Computational Cost 필요
- BYOL의 경우, 다양한 Component를 가지고 있어 다소 복잡한 구조를 가짐
  - ➢ 사실, 개인적인 관점으로는 BYOL도 충분히 Simple하지만, 더 SimSiam은 더 Simple한 구조를 보여줌

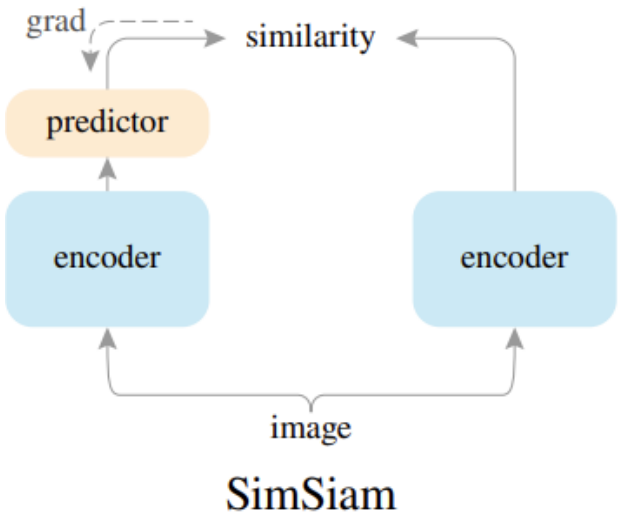


연구배경

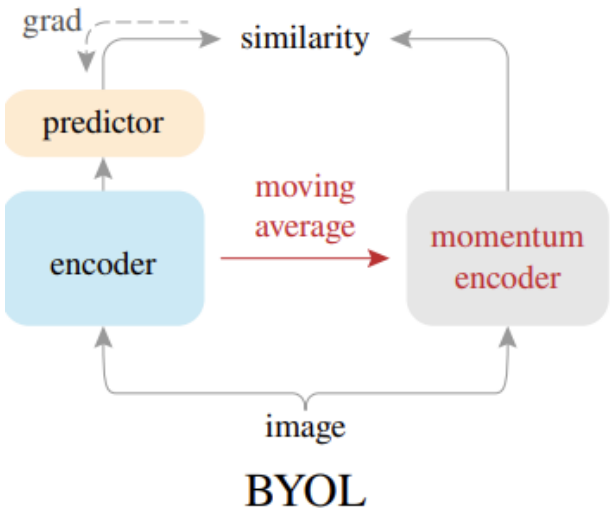
- 선행연구의 한계를 극복과정 (Overview of Research)

❖ Overcome the Limitation

- BYOL과 같은 학습방식으로 진행
  - ➢ 두 개의 Siamese Network 구조로, 두 증강된 이미지의 유사도가 최대가 되도록 학습
- SimSiam == BYOL without EMA



SimSiam



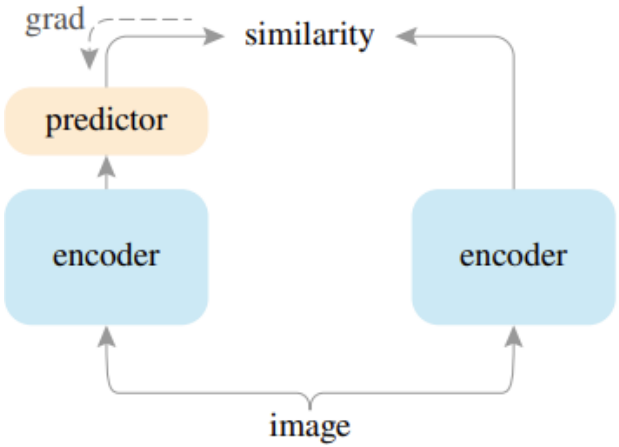
BYOL

연구배경

- Contribution

❖ Contribution

- Simple한 Non-contrastive Learning 방법론을 제안
- 학습 붕괴에 가장 큰 영향을 주는 것은 Stop-gradient라는 것을 실험을 통해 증명
- 아래와 같은 장점을 가짐 → **Simple!**
  - ① 큰 Batch Size를 필요로 하지 않음
  - ② Negative Sample을 필요로 하지 않음
  - ③ Momentum Encoder (EMA)를 필요로 하지 않음



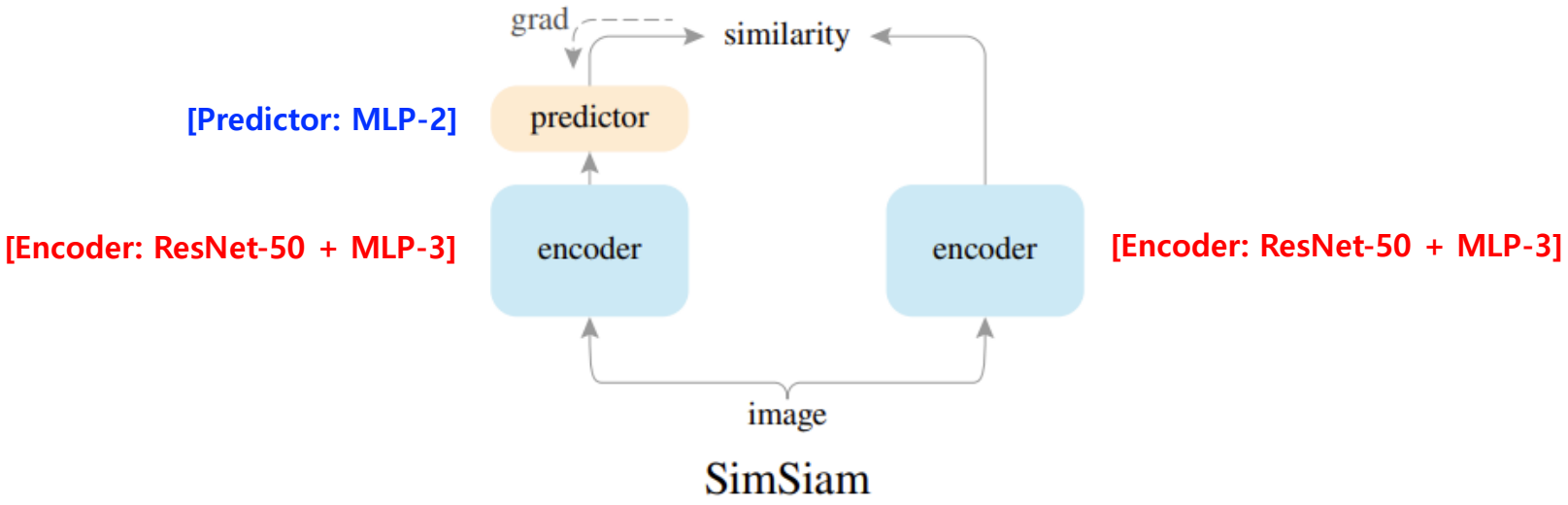
SimSiam

방법론

- Architecture

❖ 방법론 진행과정

- Step1. 주어진 이미지를 두 번 증강 수행
- Step2. 한 이미지는 Encoder + Predictor 통과 / 다른 이미지는 Encoder만 통과
- Step3. 출력된 Output Vector의 Cosine 유사도가 최대가 되도록 학습
  - ➢ BYOL처럼 Symmetric 구조 활용

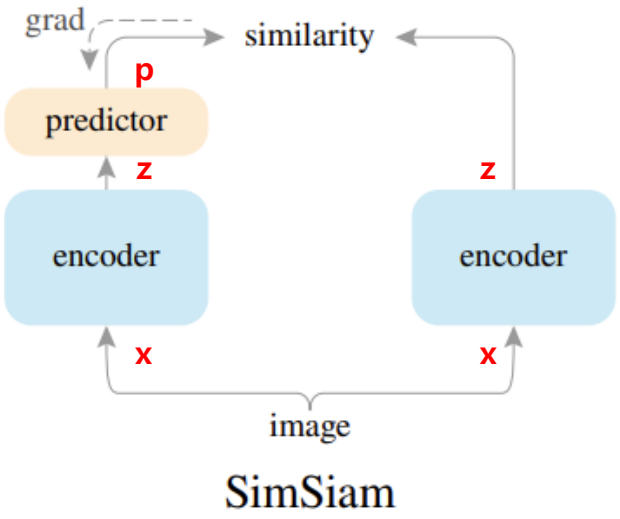


방법론

- Why?

❖ 궁금증 및 해소과정

- Stop Gradient를 적용하면, Target Encoder는 초기 Random Initialized 값?
  - ➢ No. Online Encoder의 Weight를 공유!



SimSiam

Algorithm 1 SimSiam Pseudocode, PyTorch-like

```
# f: backbone + projection mlp
# h: prediction mlp

for x in loader: # load a minibatch x with n samples
   x1, x2 = aug(x), aug(x) # random augmentation
   z1, z2 = f(x1), f(x2) # projections, n-by-d
   p1, p2 = h(z1), h(z2) # predictions, n-by-d

   L = D(p1, z2)/2 + D(p2, z1)/2 # loss

   L.backward() # back-propagate
   update(f, h) # SGD update

def D(p, z): # negative cosine similarity
   z = z.detach() # stop gradient

   p = normalize(p, dim=1) # l2-normalize
   z = normalize(z, dim=1) # l2-normalize
   return -(p*z).sum(dim=1).mean()
```

실험결과

- 비교 모델들과 실험 결과

❖ 다른 방법론들과 실험결과

- Linear Evaluation에서 완전히 SOTA라고 할 수는 없음.
  - ➢ 작은 Batch Size로, 적은 Epoch 시, 좋은 성능을 보이는 것으로 만족
- 다른 데이터 셋에 대한 Transfer Learning에서는 좋은 성능을 보임

method	batch size	negative pairs	momentum encoder	100 ep	200 ep	400 ep	800 ep
SimCLR (repro.+)	4096	✓		66.5	68.3	69.8	70.4
MoCo v2 (repro.+)	**256**	✓	✓	67.4	69.9	71.0	72.2
BYOL (repro.)	4096		✓	66.5	**70.6**	**73.2**	**74.3**
SwAV (repro.+)	4096			66.5	69.1	70.7	71.8
SimSiam	**256**			**68.1**	70.0	70.8	71.3

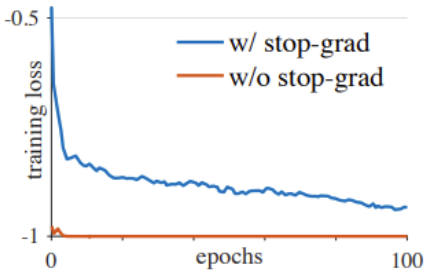
	VOC 07 detection			VOC 07+12 detection			COCO detection			COCO instance seg.		
pre-train	AP_{50}	AP	AP_{75}	AP_{50}	AP	AP_{75}	AP_{50}	AP	AP_{75}	AP^{mask}_{50}	AP^{mask}	AP^{mask}_{75}
scratch	35.9	16.8	13.0	60.2	33.8	33.1	44.0	26.4	27.8	46.9	29.3	30.8
ImageNet supervised	74.4	42.4	42.7	81.3	53.5	58.8	58.2	38.2	41.2	54.7	33.3	35.2
SimCLR (repro.+)	75.9	46.8	50.1	81.8	55.5	61.4	57.7	37.9	40.9	54.6	33.3	35.3
MoCo v2 (repro.+)	**77.1**	**48.5**	**52.5**	**82.3**	**57.0**	**63.3**	**58.8**	**39.2**	**42.5**	**55.5**	**34.3**	**36.6**
BYOL (repro.)	**77.1**	47.0	49.9	81.4	55.3	61.1	57.8	37.9	40.9	54.3	33.2	35.0
SwAV (repro.+)	75.5	46.5	49.6	81.5	55.4	61.4	57.6	37.6	40.3	54.2	33.1	35.1
SimSiam, base	75.5	47.0	50.2	**82.0**	56.4	62.8	57.5	37.9	40.9	54.2	33.2	35.2
SimSiam, optimal	**77.3**	**48.5**	**52.5**	**82.4**	**57.0**	**63.7**	**59.3**	**39.2**	**42.1**	**56.0**	**34.4**	**36.7**

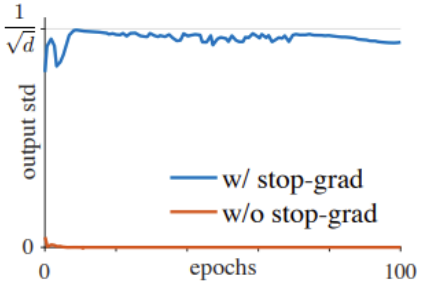
실험결과

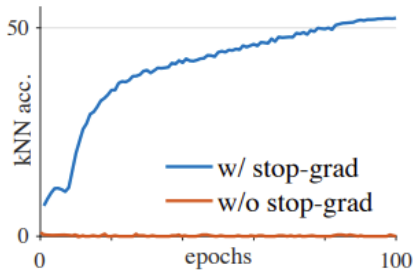
- Ablation Study

❖ Stop Gradient

- Stop Gradient는 성능 증가에 도움을 줄 뿐만 아니라, 학습 Collapse에 지대한 영향을 끼침
  - ➢ Train Loss / Embedding Vector의 분산 / Linear Evaluation
- 학습 Collapse가 일어난 것은 Embedding Vector의 분산이 0에 가깝기에, Constant Output에 의한 것임을 추측 가능







	acc. (%)
w/ stop-grad	67.7±0.1
w/o stop-grad	0.1

실험결과

- Ablation Study

두 요소 모두 Collapse에는 무관하고,
성능 향상에는 기여

❖ Batch Size & Batch Normalization

- 일반적으로 Batch Size가 클수록 좋지만, 엄청 클 경우 좋은 성능을 보이지는 않음
  - ➢ Optimizer가 큰 Batch Size에 적합한 LARS를 활용한 것이 아닌, SGD를 활용했기 때문으로 추측
- 일반적인 Contrastive Learning보다 작은 Batch Size에도 강건
- Projection 및 Predictor에서 BN은 효과적임을 입증

batch size	64	128	256	512	1024	2048	4096
acc. (%)	66.1	67.3	68.1	68.1	68.0	67.9	64.0

Table 2. **Effect of batch sizes** (ImageNet linear evaluation accuracy with 100-epoch pre-training).

	case	proj. MLP's BN hidden	proj. MLP's BN output	pred. MLP's BN hidden	pred. MLP's BN output	acc. (%)
(a)	none	-	-	-	-	34.6
(b)	hidden-only	✓	-	✓	-	67.4
(c)	default	✓	✓	✓	-	68.1
(d)	all	✓	✓	✓	✓	unstable

Table 3. **Effect of batch normalization on MLP heads** (ImageNet linear evaluation accuracy with 100-epoch pre-training).

실험결과

- Ablation Study

❖ Predictor / Similarity Function / Symmetric Loss

- Predictor가 있을 때, 좋은 성능을 보임
- Cosine 유사도가 Cross Entropy보다 좋은 성능
- Symmetric Loss가 효과적임을 입증

세 요소 모두 Collapse에는 무관하고,
성능 향상에는 기여

	pred. MLP h	acc. (%)
baseline	lr with cosine decay	67.7
(a)	no pred. MLP	0.1
(b)	fixed random init.	1.5
(c)	lr not decayed	68.1

	cosine	cross-entropy
acc. (%)	68.1	63.2

	sym.	asym.	asym. 2×
acc. (%)	68.1	64.8	67.3