type 연구

AI summary Try on this page

(이과대학 수학) 양민석 Nov 26

Self-Supervised Learning

RESEARCH

OVERVIEW

SimCLR: contrastive learning을 사용한 단순한 framework

특별한 architercture이나 memory bacnk 없이 단순화시켰다.

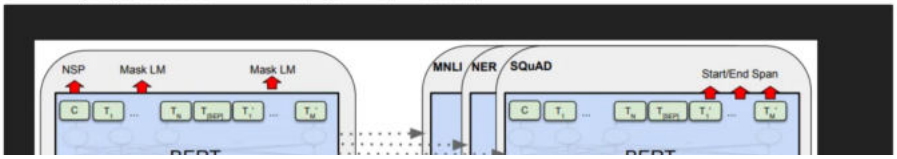
데이터의 증가가 예측 일을 효과적으로 하는데 주요한 역할을 하고, nonlinear transformation을 representation과 contrastive loss 사이에 두어 quality를 향상시킨다. 또한 contrastive learning은 supervised learning에 비교하여 더 큰 batch size와 training step에 이점을 두고 있다.

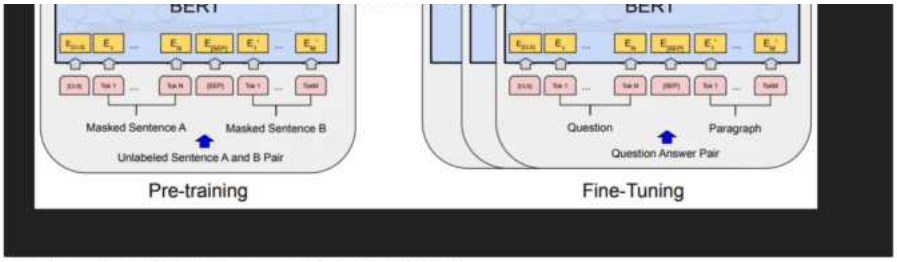
이러한 결론들을 결합하여 self-supervised or semi-supervised learning에서 outperform하는 모델을 만들 수 있다.

CONTENT

0. Self-Supervised Learning

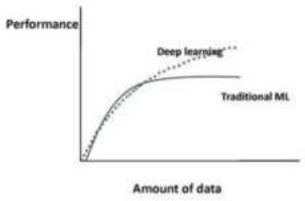
- Label이 없는 Untagged data를 기반으로 한 학습이며, 자기 스스로 학습 데이터에 대한 분류(Supervision)을 수행하기 때문에 Self가 붙었다.
- Unsupervised learning은 Tagged data가 없기 때문에 Data의 특징에 따라 다른 범주로 묶는 Clustering을 수행하였다.
- 반대로 Supervised Learning은 Tag가 함께 있기 때문에 Classification이나 Regression 목적으로 활용되었다.





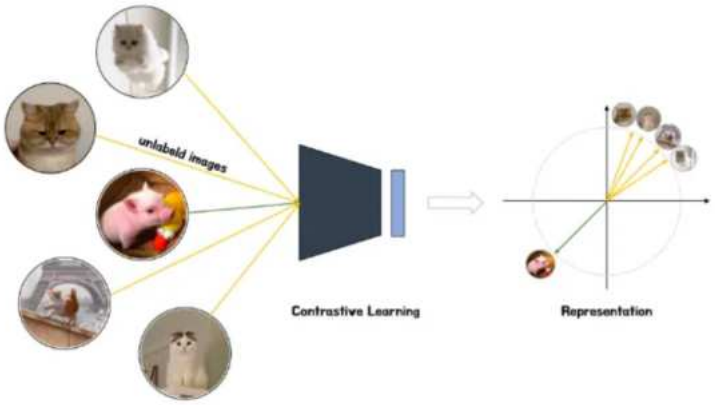
- Self-supervised learning은 **Pre-Trained 모델 생성과 Downstream task**라는 두 단계로 구성되어 있다.

1. Pre-Trained 모델
   a. 대량의 Untagged data를 이용해 해당 음율에 대해 전반적인 특징을 학습하는 단계
   b. BERT는 전체 문장에서 하나의 단어를 지운(Masking) 후 해당 단어가 무엇이었을지 추측하는 방법과 다음에 어떠한 문장이 올지 추측하는 방법으로 Pre-Trained 모델 학습을 진행하였다.
2. Downstream task
   a. 소량의 Tagged data를 활용하여 사용 목적에 맞게 Pre-Trained model을 Fine-Tuning한다.

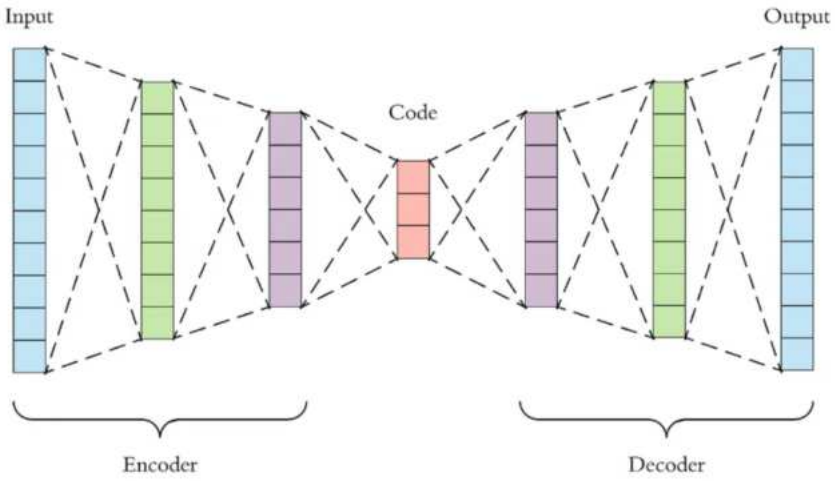


- Self-Supervised learning이 등장하게 된 이유는 Tagged data 수집의 어려움을 해결으로 생겨났다.
- 일반적으로 Deep learning 모델은 모델 parameter 수가 증가함에 따라 정확도가 향상되는데 큰 사이즈의 모델을 적절하게 학습하기 위해서는 대량의 데이터가 필요하다.
- Self-Supervised learning은 Tagged data가 적어도 되서 모델 Size 증가가 쉬워 모델의 정확도가 높아난다는 장점이 있다.

1. Introduction



- Contrastive Learning: 유사한 이미지가 저차원 공간에서 서로 가깝게, 동시에 다른 이미지는 서로 멀리 떨어져있도록 저차원 공간에서 이미지를 인코딩하는 방법을 모델이 학습하는 것이다.
- 그림을 보면 고양이 이미지끼리는 가깝고, 돼지 이미지와는 거리가 먼 것을 확인할 수 있다.
- contrastive learning을 사용하면 주석이나 레이블이 없어도 데이터에 대해 많은 것을 학습하도록 모델을 훈련시킬 수 있다.
- representation learning: input을 잘 표현하여 학습



1. Generative model
   - Autoencoder(Encoder + Decoder)로 Self-Supervised Learning을 하는 것이 Generative Learning이다.
   - 생성되는 입력과 데이터가 특정한 통계적 분포를 따른다고 가정.
   - pixel을 생성하는 것은 비용이 너무 비쌌음.

2. Discriminative(변별) model
   - 입력값 x 가 주어졌을 때 그 결과값(label)이 y일 확률
   - P(y|x), y가 0일지 1일지를 확률적으로 예측
   - 결과적으로 구분선을 찾아내는 것이 중요

1.1 SSL 부연설명

- Pretext task: Self-Supervised Learning을 학습하기 위해서, 사용자가 만든 문제
- Downstream task: Self-Supervised Learning을 적용해서 풀 문제

1. SSL은 처음에 label이 없는 데이터셋을 가지고 학습을 진행한다. (Pretext task)
   a. 각 image에 대한 label를 사용자가 임의로 만들어서 학습
2. SSL을 통해 학습한 모델을 기존의 모델에 적용하여 평가 (Downstream task에 적용)
   a. Linear evaluation: Pretext task를 통해서 학습됐던 모델의 weights를 freeze 시키고 난 후, 뒤에 FC layer를 붙여서 fine-tuning시킨다.
   b. Semi Supervised Learning: 데이터 셋의 label을 1%~10%사이만 이용해서 학습
   c. Transfer learning: ImageNet으로 학습한 모델을 transfer-learning 시켜서 다른 dataset을 평가

Contrastive Learning

- No decoder
- Contrastive loss(InfoNCE loss)
- Positive와 Negative sample을 비교하면서 학습을 진행
- Augmentation 활용

- Composition of multiple data augmentation: good
- representation → non-linear projection → contrastive loss
- loss를 구할 때 L2 norm 적용 + 적절한 temperature parameter 사용

cf) temperature는 분포를 조금 더 뾰족하게 혹은 평평하게 만들 수 있는 파라미터이다.

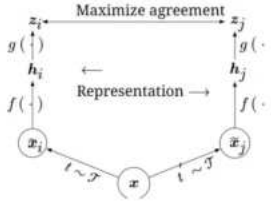
- batch size와 training epoch을 크게하면 좋다. Supervised Learning처럼 network가 깊을수록 더 좋다.

2. Method

2.1 The Contrastive Learning Framwork

단어설명

- Representation: 이미지를 표현 ex) feature vector들로 representation
- latent space(해공간, 잠재공간): space of feature vector
- Encoder: input으로부터 유용한 feature를 뽑아내는 역할을 하는 구조 → 뒤에 FC같은 classifier연결하여 classification 문제 해결 or Decoder 구조를 연결해서 segmentation 문제 해결.



1. X라는 데이터 셋을 각각 2번 Augmentation 시켜서 xi, xj를 얻는다.

$$neural\ network\ base\ encoder\ (f(\cdot) = ResNet(\cdot))$$

2. Encoder f(x)를 이전에 얻어진 augmented data에 적용해서 representation vector를 추출한다. f(x)는 보통 ResNet50을 사용한다.
3. g(x)는 이전에 설명한 non-linear projection이다. Linear → ReLU → Linear 순서이다.

$$z_i = g(h_i) = W^{(2)}\sigma(W^{(1)}h_i),\ \sigma : ReLU$$

4. Projection head에서 나온 embedding 값을 InfoNCE(Contrastive loss)를 이용해서 서로의 유사도를 계산해서 loss function을 계산한다.
   - cross-entropy loss의 x에 simiarity를 대입한 것과 같다.

- N개의 minibatch를 랜덤으로 뽑아서 contrastive prediction task에 사용할 것이다. 그러면 2N개의 data points가 나올 것이다.

$$sim(u, v) = \frac{u^T v}{|u||v|}\ \ using\ l_2\ normalization$$

$$l_{i,j} = -log\frac{exp(sim(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbf{1}_{k \neq i} exp(sim(z_i, z_k)/\tau)}$$

- 분자는 positive sample에 대한 서로의 유사도.
- 분모는 전체 데이터 셋: positive sample + negative sample에 대한 유사도 총합

Algorithm 1 SimCLR's main learning algorithm.

input: batch size N, constant τ, structure of $f, g, \mathcal{T}$.
for sampled minibatch $\{\boldsymbol{x}_k\}_{k=1}^{N}$ **do**
  for all $k \in \{1, \ldots, N\}$ **do**
    draw two augmentation functions $t \sim \mathcal{T}, t' \sim \mathcal{T}$
    # the first augmentation
    $\tilde{\boldsymbol{x}}_{2k-1} = t(\boldsymbol{x}_k)$
    $\boldsymbol{h}_{2k-1} = f(\tilde{\boldsymbol{x}}_{2k-1})$ # representation
    $\boldsymbol{z}_{2k-1} = g(\boldsymbol{h}_{2k-1})$ # projection
    # the second augmentation
    $\tilde{\boldsymbol{x}}_{2k} = t'(\boldsymbol{x}_k)$
    $\boldsymbol{h}_{2k} = f(\tilde{\boldsymbol{x}}_{2k})$ # representation
    $\boldsymbol{z}_{2k} = g(\boldsymbol{h}_{2k})$ # projection
  end for
  for all $i \in \{1, \ldots, 2N\}$ and $j \in \{1, \ldots, 2N\}$ **do**
    $s_{i,j} = \boldsymbol{z}_i^\top \boldsymbol{z}_j / (\|\boldsymbol{z}_i\| \|\boldsymbol{z}_j\|)$ # pairwise similarity
  end for
  define $\ell(i, j)$ **as** $\ell(i, j) = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(s_{i,k}/\tau)}$
  $\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} [\ell(2k-1, 2k) + \ell(2k, 2k-1)]$
  update networks f and g to minimize $\mathcal{L}$
end for
return encoder network $f(\cdot)$, and throw away $g(\cdot)$

- f, g를 update 시킨다.

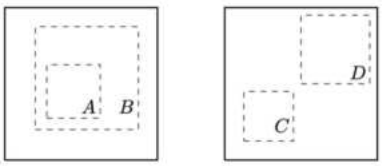
2.2 Training with Large Batch Size

- memory bank를 사용하는 대신에 큰 batch size를 사용할 것이다.
- 큰 batch size는 standard SGD/Momentum을 사용할 때 불안정하다.
- 안정시키기 위해서 **LARS optimizer**를 사용할 것이다.

2.3 Evaluation Protocol

- f: ResNet-50, g: 2-layer MLP(multi-layer Perceptron)
- augmentation: random crop and resize, resize이후 random flip 적용
- color distortion 사용
- gaussian blur 사용

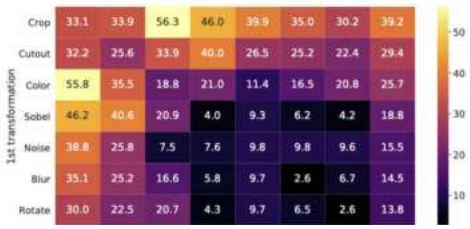
3. Data augmentation for Contrastive representation learning



(a) Global and local views. (b) Adjacent views.

Figure 3. Solid rectangles are images, dashed rectangles are random crops. By randomly cropping images, we sample contrastive prediction tasks that include global to local view ($B \rightarrow A$) or adjacent view ($D \rightarrow C$) prediction.

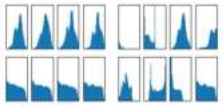
- 기존의 방법에서는 network architercture를 변형시켜서 Contrastive prediction task를 수행했다.
- 예를 들어 global-to-local view prediction을 위해서 network의 receptive field를 조절한다거나, neighboring view prediction을 위해서 context aggregation network를 사용하는 등의 방법을 사용했다.



- 여러가지 augmentation들을 개별적으로 적용하거나 조합하여 적용해서 성능을 뽑아봤을 때, crop + color distortion의 성능이 가장 좋음을 알 수 있다.
- 대체로 단일 적용보다 조합하여 적용하는 경우의 성능이 높았다.

3.1 Composition of data augmentation operations is crucial for learning good representations

- Augmentation을 위한 transformation으로는 크게 cropping/resizing/flipping, rotation, cutout 등의 spatial하거나 geometric한 transformation과 color distortion, Gaussian blur, Sobel filtering 등의 apperance transformation이 있다.
- 위에서 알 수 있듯이 single transformation보다는 pair로 사용하는게 더 좋다는 것을 알았다.



(a) Without color distortion. (b) With color distortion.

Figure 6. Histograms of pixel intensities (over all channels) for different crops of two different images (i.e. two rows). The image for the first row is from Figure 4. All axes have the same range.

- color distortion을 사용해서 contrastive prediction을 단순하지 않게 바꿔 더 일반적인 feature를 학습하도록 한다.
- 즉 deep learning의 shortcut 학습을 방지한다.

3.2 Contrastive learning needs stronger data augmentation than supervised learning

Methods	Color distortion strength					AutoAug
	1/8	1/4	1/2	1	1 (+Blur)	
SimCLR	59.6	61.0	62.6	63.2	64.5	61.1
Supervised	77.0	76.7	76.5	75.7	75.4	77.1

Table 1. Top-1 accuracy of unsupervised ResNet-50 using linear evaluation and supervised ResNet-50', under varied color distortion strength (see Appendix A) and other data transformations. Strength 1 (+Blur) is our default data augmentation policy.

- color distortion이 강할수록 SimCLR의 성능이 좋아지는 것을 확인할 수 있다.
- 반대로 Supervised에서는 별차이가 없는 것을 확인할 수 있다.
- 결론적으로
- unsupervised contrastive learning은 더 강한 data augmentation으로부터 성능 이득을 본다.

4. Architectures for Encoder and Head

4.1 Unsupervised contrastive learning benefits more from bigger models

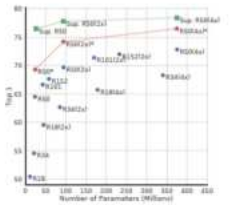


Figure 7. Linear evaluation of models with varied depth and width. Models in blue dots are ours trained for 100 epochs, models in red stars are ours trained for 1000 epochs, and models in green crosses are supervised ResNets trained for 90 epochs" (He et al., 2016).

- 모델이 깊어질수록 linear evaluation 성능이 좋아지는 것을 확인할 수 있다.
- unsupervised가 모델이 깊을수록 성능 향상이 더 좋다.

4.2 A nonlinear projection head improves the representation quality of the layer before it

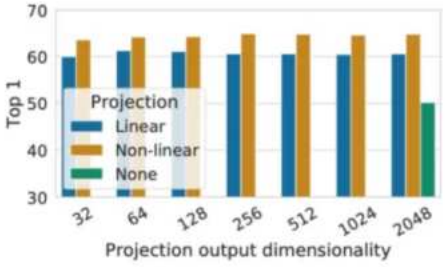


Figure 8. Linear evaluation of representations with different projection heads $g(\cdot)$ and various dimensions of $\boldsymbol{z} = g(\boldsymbol{h})$. The

jection heads $g(\cdot)$ and various dimensions of $\boldsymbol{z} = g(\boldsymbol{h})$. The representation $\boldsymbol{h}$ (before projection) is 2048-dimensional here.

- project g를 Linear/non-linear/none일 때의 성능을 비교하였다.
- non-linear이 성능 면에서 좋은 것을 확인할 수 있다.

What to predict?	Random guess	Representation $\boldsymbol{h}$	Representation $g(\boldsymbol{h})$
Color vs grayscale	80	99.3	97.4
Rotation	25	67.6	25.6
Orig. vs corrupted	50	99.5	59.6
Orig. vs Sobel filtered	50	96.6	56.3

- h에 대한 linear evaluation이 z에 대한 결과보다 더 좋았는데 이는 project 전의 hidden layer가 더 좋은 representation을 포함하고 있다는 것이다.
- contrastive loss로 인해 g는 data transformation에 invariant하게 학습되다보니 데이터의 중요한 정보를 잃어버리는 것으로 추측할 수 있다.

5. Loss Functions and Batch Size

5.2 Contrastive learning benefits (more) from larger batch sizes and longer training

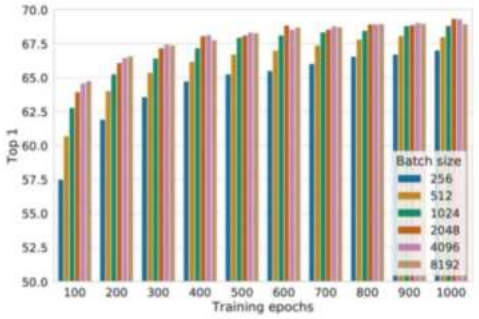


Figure 9. Linear evaluation models (ResNet-50) trained with different batch size and epochs. Each bar is a single run from scratch.[10]

- Contrastive learning은 positive sample에 비해서 negative sample이 훨씬 많아야 학습이 잘 된다.
- unlabelled data이므로 실제로 같은 class임에도 다른 instance로 분류를 하는 경우가 있으므로, 이를 온화하기 위해서 많은 negative sample이 필요하다
- **batch size에 크기에 따른 성능 차이가 심하다.**

단어장

systematically: 체계적으로

augmentation: 증강, 증가(데이터를 증가)

state-of-the-art: 최첨단의 기술

asymmetric: 비대칭적인

SEARCH

REFER

#	Title	DOI	Link
[1]			