Towards Domain-Agnostic Contrastive Learning

DACL

Fengjiao Gong 2021.08.13

Contents

- Background
- Problem Definition
- Mixup Noise
- Algorithm
- Theoretical Analysis
- Experiments
- Future work

Background

Self-supervised learning

- deep learning one core objective

to discover useful representations from the raw input signals without explicit labels provided by human annotators

- self-supervised learning

accomplish the objective by reformulating the unsupervised representation learning problem into a supervised learning problem(define a pretext task)

- categorized by the pretext tasks (domain-specific, generally differ from domain to domain)

natural language understanding — predict the neighbouring words (word2vec)\ the next word\next sentence\the masked word \the replaced word in the sentence

computer vision — rotation prediction/relative position prediction of image patches /image colorization / reconstructing the original image from the partial image and predict an odd video subsequence in a video sequence

graph-structured data — predict the context (neighbourhood of a given node) / the masked attributes of the node

contrastive learning — a form of self-supervised learning

pretext is to bring positive samples closer than the negative samples in the representation space

Background

Contrastive learning

- categorized by how the positive and negative samples are constructed

domain-specific augmentations — state-of-the art for computer vision

not domains where semantic-preserving data augmentation does not exist, such as graph-data or tabular data

define the local and global context

not domains where such global and local context does not exist, such as tabular data

use the ordering in the sequential data

not if the data sample cannot be expressed as an ordered sequence, such as graphs and tabular data

simplest solution

add a sufficiently small random noise to a given sample to construct examples that are similar to it

mixup based methods — widely remarkable success in various problems

recently explored in contrastive learning 2020

this paper differs from existing methods in theoretically demonstration, several forms of mixup-noise, applicable to different domains and no additional gradient computation

Problem Definition

contrastive learning objective

- Suppose we have an encoding function h: x->**h**, the anchor sample **x**, its corresponding positive and negative samples x^+ and x^-
- the objective of contrastive learning is to bring the anchor and the positive sample closer in the embedding space the the anchor and the negative sample, which is to satisfy the following condition:

$$sim(h, h^+) > sim(h, h^-)$$

- Suppose that $\{x_k\}_{k=1}^N$ is a set of N samples such that it consists of a sample xi which is semantically similar to xj and dissimilar to all the other samples in the set. Then the InfoNCE loss is defined as followed to maximize the similarity between the positive pair and minimize the similarity between the negative pairs:

$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{h}_i, \boldsymbol{h}_j))}{\sum_{k=1}^{N} 1_{[k \neq i]} \exp(\mathrm{sim}(\boldsymbol{h}_i, \boldsymbol{h}_k))}$$

Mixup Noise

Gaussian-noise

- Consider an image x and adding Gaussian-noise to it for constructing the positive sample:

$$x^+ = x + \delta, \text{ where } \delta \sim N(0, \sigma^2 I)$$

- In this case, to maximize the similarity between x and x^+, the network can learn just to take an average over the neighboring pixels to remove the noise, thus bypassing learning the semantic concepts in the image.
- The central hypothesis of DACL method is that a network is forced to learn better features if the noise captures the structure of the data manifold rather than being independent of it. For mixup noise, it forces the network to learn better features

Linear-Mixup Noise

domains where natural data augmentation methods are not available, data has a fixed topology

Given a data distribution $D = \{x_k\}_{k=1}^N$

- create positive samples using Mixup in the input space by taking its random interpolation with another randomly chosen sample from the same distribution D:

$$x^+ = \lambda x + (1 - \lambda)\tilde{x}$$

where λ is a coefficient sampled from a random distribution such that x^+ is closer to x than x^-.

- negative samples — positive samples corresponding to other anchor samples

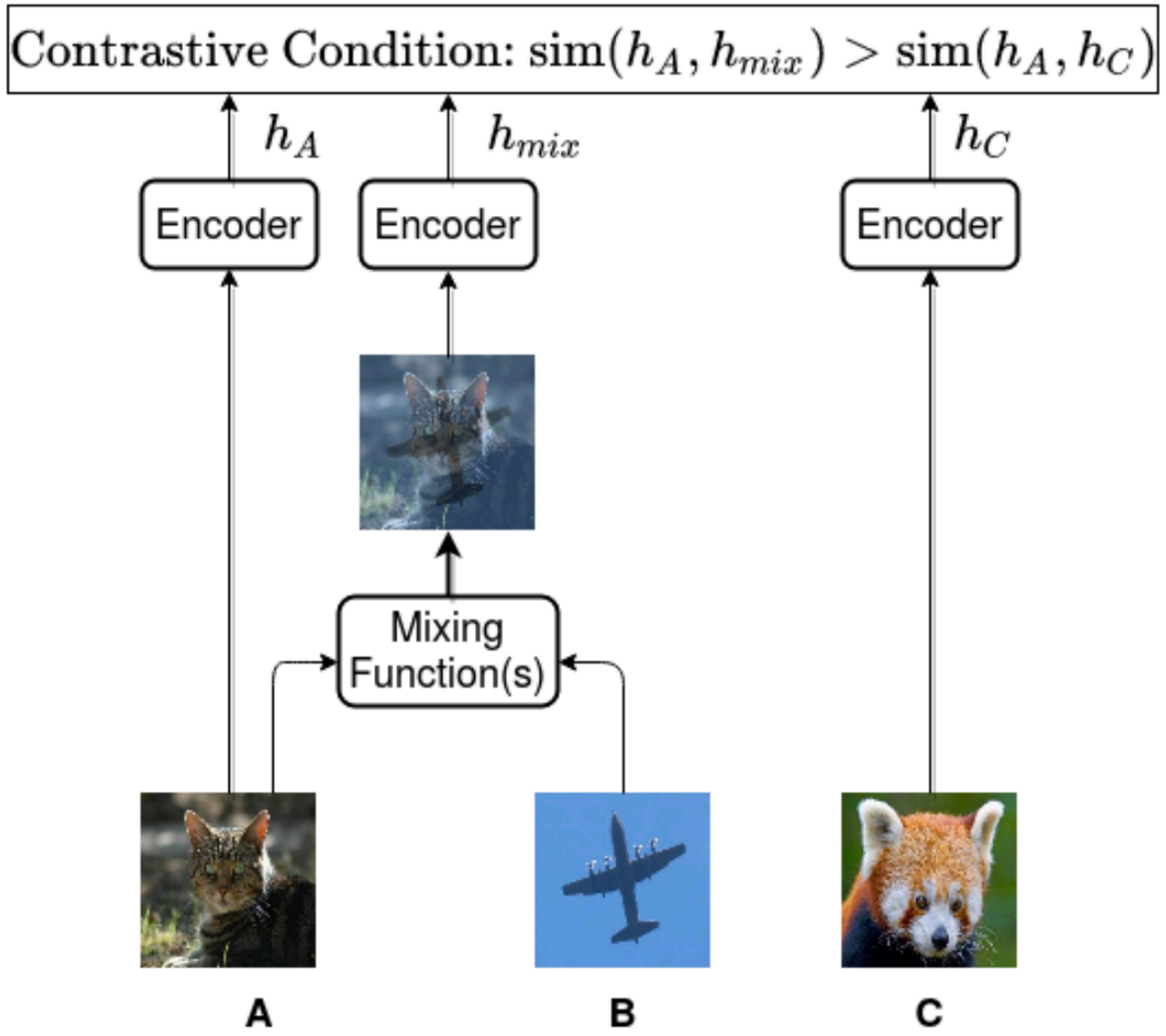

Figure 1. For a given sample A, we create a positive sample by mixing it with another random sample B. The mixing function can be either of the form of Equation 3 (Linear-Mixup), 5 (Geometric-Mixup) or 6 (Binary-Mixup), and the mixing coefficient is chosen in such a way that the mixed sample is closer to A than B. Using another randomly chosen sample C, the contrastive learning formulation tries to satisfy the condition $\mathrm{sim}(\boldsymbol{h_A}, \boldsymbol{h}_{mix}) > \mathrm{sim}(\boldsymbol{h_A}, \boldsymbol{h}_C)$, where sim is a measure of similarity between two vectors.

Linear-Mixup Noise

data has a non-fixed topology, such as sequences, trees and graphs

- create positive samples by mixing fixed-length hidden representation of samples:

Formally, let us assume that there exists an encoder function $h : I \rightarrow \mathbf{h}$ that maps a sample I from such domains to a representation $\mathbf{h}$ via an intermediate layer that has a fixed-length hidden representation v, then we create positive sample in the intermediate layer as:

$$v^+ = \lambda v + (1 - \lambda)\tilde{v}$$

- negative samples — positive samples corresponding to other anchor samples
- So, Mixup noise is adding noise to a given sample in the direction of another sample in the data distribution.

Additional Forms of Mixup Noise

Geometric-Mixup, Binary-Mixup

- In GeometricMixup, we create a positive sample corresponding to a sample x by taking its weighted-geometric mean with another randomly chosen sample $\tilde{x}$:

$$\boldsymbol{x}^{+} = \boldsymbol{x}^{\lambda} \odot \tilde{\boldsymbol{x}}^{(1-\lambda)}$$

- In Binary-Mixup (Beckham et al., 2019), the elements of x are swapped with the elements of another randomly chosen sample $\tilde{x}$. This is implemented by sampling a binary mask m $\in \{0, 1\}^k$ (where k denotes the number of input features) and performing the following operation:

$$\boldsymbol{x}^{+} = \boldsymbol{x} \odot \mathbf{m} + \tilde{\boldsymbol{x}} \odot (1 - \mathbf{m})$$

Algorithm 1

Mixup-nose Domain-Agnostic Contrastive Learning

- DACL+

For a given sample x, we randomly select a noise function from LinearMixup, Geometric-Mixup, and Binary-Mixup, and apply this function to create both of the positive samples corresponding to x (line 7 and 13 in Algorithm 1). The rest of the details are the same as Algorithm1. We refer to this procedure as DACL+ in the following experiments.

1: **input:** batch size N, temperature τ, encoder function h, projection-head g, hyperparameter α.
2: **for** sampled minibatch $\{\boldsymbol{x}_k\}_{k=1}^N$ **do**
3: **for all** $k \in \{1, \ldots, N\}$ **do**
4: # Create first positive sample using Mixup Noise
5: $\lambda_1 \sim U(\alpha, 1.0)$ # sample mixing coefficient
6: $\boldsymbol{x} \sim \{\boldsymbol{x}_k\}_{k=1}^N - \{\boldsymbol{x_k}\}$
7: $\tilde{\boldsymbol{x}}_{2k-1} = \lambda_1 \boldsymbol{x_k} + (1-\lambda_1)\boldsymbol{x}$
8: $\boldsymbol{h}_{2k-1} = h(\tilde{\boldsymbol{x}}_{2k-1})$ # apply encoder
9: $\boldsymbol{z}_{2k-1} = g(\boldsymbol{h}_{2k-1})$ # apply projection-head
10: # Create second positive sample using Mixup Noise
11: $\lambda_2 \sim U(\alpha, 1.0)$ # sample mixing coefficient
12: $\boldsymbol{x} \sim \{\boldsymbol{x}_k\}_{k=1}^N - \{\boldsymbol{x_k}\}$
13: $\tilde{\boldsymbol{x}}_{2k-1} = \lambda_2 \boldsymbol{x_k} + (1-\lambda_2)\boldsymbol{x}$
14: $\boldsymbol{h}_{2k} = h(\tilde{\boldsymbol{x}}_{2k})$ # apply encoder
15: $\boldsymbol{z}_{2k} = g(\boldsymbol{h}_{2k})$ # apply projection-head
16: **end for**
17: **for all** $i \in \{1, \ldots, 2N\}$ and $j \in \{1, \ldots, 2N\}$ **do**
18: $s_{i,j} = \boldsymbol{z}_i^\top \boldsymbol{z}_j / (\|\boldsymbol{z}_i\| \|\boldsymbol{z}_j\|)$ # pairwise similarity
19: **end for**
20: **define** $\ell(i,j) = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(s_{i,k}/\tau)}$
21: $\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} [\ell(2k-1, 2k) + \ell(2k, 2k-1)]$
22: update networks h and g to minimize $\mathcal{L}$
23: **end for**
24: **return** encoder function $h(\cdot)$, and projection-head $g(\cdot)$

Theoretical Analysis

analyze and compare the properties of Mixup-noise and Gaussian-noise based contrastive learning for a binary classification task

- first prove that for both Mixup-noise and Gaussian-noise, optimizing hidden layers with a contrastive loss is related to minimizing classification loss with the last layer being optimized using labeled data
- then prove that the proposed method with Mixup-noise induces a different regularization effect on the classification loss when compared with that of Gaussian-noise.shows the advantage of Mixup-noise over Gaussian-noise when the data manifold lies in a low dimensional subspace
- contrastive learning with Mixup-noise has implicit data-adaptive regularization effects that promote generalization

Experiments

linear evaluation

- use the linear evaluation protocol to evaluate the learned representations — where a linear classifier is trained on top of a frozen encoder network, and the test accuracy is used as a proxy for representation quality
- discard the projection-head during linear evaluation(similar to SimCLR — as the baseline)
- domains — tabular, images and graphs

Algorithm 1 SimCLR's main learning algorithm.

input: batch size N, temperature 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**
    $s_{i,j} = \boldsymbol{z}_i^\top \boldsymbol{z}_j / (\tau \|\boldsymbol{z}_i\| \|\boldsymbol{z}_j\|)$ # pairwise similarity
  end for
  define $\ell(i,j)$ **as** $\ell(i,j) = -\log \frac{\exp(s_{i,j})}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(s_{i,k})}$
  $\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} [\ell(2k-1, 2k) + \ell(2k, 2k-1)]$
  update networks f and g to minimize $\mathcal{L}$
end for
return encoder network f

Experiments

Tabular Data

- Datasets

Fashion-MNIST + CIFAR-10

- Baselines

No-pretraining

Gaussian-noise based contrastive leaning

Full network supervised training

- Results

DACL performs significantly better than the Gaussian-noise based contrastive learning

DACL+ further improves the performance of DACL

DACL gives better performance than training the full network in a supervised manner

Method	**Fashion-MNIST**	**CIFAR10**
No-Pretraining	66.6	26.8
Gaussian-noise	75.8	27.4
DACL	81.4	37.6
DACL+	**82.4**	**39.7**
Full network supervised training	79.1	35.2

Table 1. Results on tabular data with a 12-layer fully-connected network.

Experiments

Image Data

- Datasets: CIFAR-10 + CIFAR-100
- Baselines:

No-Pretraining,

Gaussian-noise based contrastive learning

SimCLR

- Results:

DACL is better than Gaussian-noise based contrastive learning by a wide margin and DACL+ can improve the test accuracy even further.

DACL falls short of methods that use image augmentations such as SimCLR.

the invariances learned using the image-specific augmentation methods facilitate learning better representations than making the representations invariant to Mixup-noise.

Method	**CIFAR-10**	**CIFAR-100**
No-Pretraining	43.1	18.1
Gaussian-noise	56.1	29.8
DACL	81.3	46.5
DACL+	83.8	52.7
SimCLR	93.4	73.8
SimCLR+DACL	**94.3**	**75.5**

Table 2. Results on CIFAR10/100 with ResNet50($4\times$)

Experiments

Image Data

- Datasets: ImageNet
- Baselines: recent contrastive learning methods
- SimCLR+DACL refers to the combination of the SimCLR and DACL methods, which is implemented using the following steps: (1) for each training batch, compute the SimCLR loss and DACL loss separately and (2) pretrain the network using the sum of SimCLR and DACL losses.
- Results:

combine DACL with SimCLR can improve the performance of SimCLR across all the datasets.

suggest that Mixup-noise is complementary to other image data augmentations for contrastive learning.

Method	**Architecture**	**Param(M)**	**Top 1**	**Top 5**
Rotation (Gidaris et al., 2018)	ResNet50 (4×)	86	55.4	-
BigBiGAN (Donahue & Simonyan, 2019)	ResNet50 (4×)	86	61.3	81.9
AMDIM (Bachman et al., 2019)	Custom-ResNet	626	68.1	-
CMC (Tian et al., 2019)	ResNet50 (2×)	188	68.4	88.2
MoCo (He et al., 2020)	ResNet50 (4×)	375	68.6	-
CPC v2 (Hénaff et al., 2019)	ResNet161	305	71.5	90.1
BYOL (300 epochs) (Grill et al., 2020)	ResNet50 (4×)	375	72.5	90.8
No-Pretraining	ResNet50 (4×)	375	4.1	11.5
Gaussian-noise	ResNet50 (4×)	375	10.2	23.6
DACL	ResNet50 (4×)	375	24.6	44.4
SimCLR (Chen et al., 2020b)	ResNet50 (4×)	375	73.4	91.6
SimCLR+DACL	ResNet50 (4×)	375	**74.4**	**92.2**

Table 3. Accuracy of linear classifiers trained on representations learned with different self-supervised methods on the ImageNet dataset.

Experiments

Graph-Structured Data/Graph classification

- datasets: MUTAG, PTC-MR, REDDIT-BINARY, REDDIT-MULTI-5K, IMDB-BINARY, and IMDB-MULTI
- baseline: InfoGraph— based on maximizing the mutual-information between the global and node-level features of a graph by formulating this as a contrastive learning problem.
- it is required to obtain fixed-length representations from an intermediate layer of the encoder and Mixup-noise applied to the output of the encoder
- Results:

DACL closely matches the performance of InfoGraph, with the classification accuracy of these methods being within the standard deviation of each other.

In terms of the classification accuracy mean, DACL outperforms InfoGraph on four out of six datasets. no domain knowledge used for formulating the contrastive loss, yet achieved comparable performance to a state-of-the-art graph contrastive learning method.

Towards Domain-Agnostic Contrastive Learning

Dataset	MUTAG	PTC-MR	REDDIT-BINARY	REDDIT-M5K	IMDB-BINARY	IMDB-MULTI
No. Graphs	188	344	2000	4999	1000	1500
No. classes	2	2	2	5	2	3
Avg. Graph Size	17.93	14.29	429.63	508.52	19.77	13.00
			Method			
No-Pretraining	81.70 ± 2.58	53.07 ± 1.27	55.13 ± 1.86	24.27 ± 0.93	52.67 ± 2.08	33.72 ± 0.80
InfoGraph (Sun et al., 2020)	$\mathbf{86.74 \pm 1.28}$	57.09 ± 1.52	63.52 ± 1.66	$\mathbf{42.89 \pm 0.62}$	63.97 ± 2.05	39.28 ± 1.43
DACL	85.31 ± 1.34	$\mathbf{59.24 \pm 2.57}$	$\mathbf{66.92 \pm 3.38}$	42.86 ± 1.11	$\mathbf{64.71 \pm 2.13}$	$\mathbf{40.16 \pm 1.50}$

Table 4. Classification accuracy using a linear classifier trained on representations obtained using different self-supervised methods on 6 benchmark graph classification datasets.

Future Work

contributions & future work

- contributions of this paper:

propose Mixup-noise as a way of constructing positive and negative samples for contrastive learning and conduct theoretical analysis to show that Mixup-noise has better generalization bounds than Gaussian-noise.

show that using other forms of data-dependent noise (geometric-mixup, binary-mixup) can further improve the performance of DACL.

extend DACL to domains where data has a non- fixed topology (for example, graphs) by applying Mixup-noise in the hidden states.

demonstrate that Mixup-noise based data augmentation is complementary to other image-specific augmentations for contrastive learning, resulting in improvements over SimCLR baseline for CIFAR10, CIFAR100 and ImageNet datasets.

- future work could be:

extend DACL to other domains such as natural language and speech.

extend this analysis to the multi-class setting might shed more light on developing a better Mixup-noise based contrastive learning method

extend the experiments by mixing between more than two samples or learning the optimal mixing policy through an auxiliary network

Q&A

Thanks!