Review: Feature Learning of Self-Supervised Contrastive Learning by Data Augmentation

20200130 Yujun Kim

IE539 Convex Optimization

Nov 11 2023

Abstract

Contrastive learning as an instance of self-supervised learning has shown success in encoder-decoder models. Its framework is based on algorithms like [Che+20]. This report mainly summarizes and discusses results from [WL21] which theoretically characterize when contrastive learning learns good features. In particular, they point out theoretically that data augmentation is needed for the network to learn sparse(good) features instead of dense(bad) features that match the empirical results.

1 Backgrounds

1.1 Supervised and Unsupervised Learning

Learning in a broad view can be divided into supervised and unsupervised learning. Supervised learning is the process of learning predictor functions from labeled data points. In terms of empirical risk minimization, when data $S = \{(x_i, y_i)\}_{i=1}^n$ is given, we try to find the function f that best predicts the in the sense of $y_i \approx f(x_i)$. For example in $L-2$ measure $l(y, y') = ||y - y'||^2$, we can minimize

$$\frac{1}{n}\sum_{i=1}^{n} l(y_i, f(x_i))$$

Here, depending on the architecture of the construction of f is parameterized by some parameter and we try to minimize the above formula with respect to this parameter.

On the other hand, unsupervised learning is in some sense learning a distribution of unlabeled data. For example, in clustering of image data, we may be able to cluster images of similar features even if they are unlabeled. Contrastive learning serves as the intermediate bridge between the two regimes.

1.2 Contrastive Learning

Given unlabeled data, can we automatically create labels of the data without humans manually checking each data? The answer to the question suggest the direction of self-supervised learning. Self-supervised learning creates labels from unlabeled data through the process of generating quizzes. There are many different ways of generating quizzes depending on the problem we are solving.

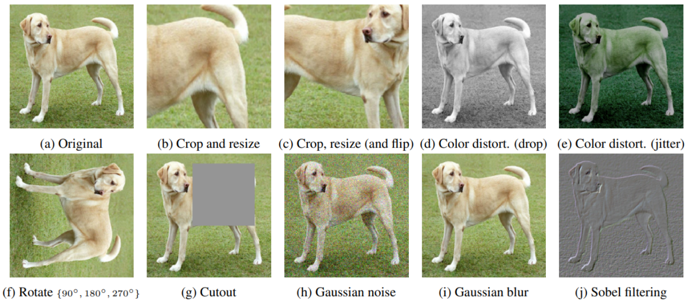
(a) Original
(b) Crop and resize
(c) Crop, resize (and flip)
(d) Color distort. (drop)
(e) Color distort. (jitter)
(f) Rotate {90°, 180°, 270°}
(g) Cutout
(h) Gaussian noise
(i) Gaussian blur
(j) Sobel filtering

Figure 1: Augmentations [Che+20]

- For example, if we want to create a learner that learns spatial information of the image, we can divide each image into smaller sub-images, with label (i, j) for each sub-image indicating where it is positioning in the original image.
- In **contrastive learning**, we can give different augmentations in each image, augmented images from the same image get the same label.

Data Augmentation. There are many different ways to augment data, as given in figure 4. Indeed, we want to find a good embedding f of those data so that augmented images from the same original image are embedded in a similar position compared to the augmented images from two different original images. A pair of augmented images from the same original image is positively labeled and a pair of augmented images from different images is negatively labeled. This analysis is motivated by the success of the encoder-decoder models in state-of-the-art architectures.

1.3 SimCLR

One of the most popular frameworks is SimCLR[Che+20] standing for 'A simple framework for contrastive learning'. Suppose our embedding is given as $f : \mathbb{R}^{d_1} \longrightarrow \mathbb{R}^m$. Then, the similarity between the embedding of x, x' is given as the inner product of the function values:

$$sim_f(x, x) = \langle f(x), f(x') \rangle$$

We want large $sim_f(x, x')$ for positively labeled pair (x, x') and the value to be small for negatively paired data. The exact formulation for the loss function is given in section 2.

1.4 Feature learning of Contrastive Learning

It is well known that supervised learning learns good features of data. Feature learning is often given as evidence why the neural network models work well. For example, higher layer convolutional filters of Wide-ResNet show rich features as given in the left three figures of 2. The first three figures show the contrastively learned features of higher layer data when trained with augmentation. We can see that even contrastively learned features contain meaningful information. Lastly, empirical analysis suggests that stronger data augmentation helps contrastive learners to learn better features. Figure 3 shows contrastive features when trained with either i)no augmentation, ii)Crop-resize, or iii)Crop-resize with color augmentation. The results show that the stronger augmentation results in better learning of the feature.

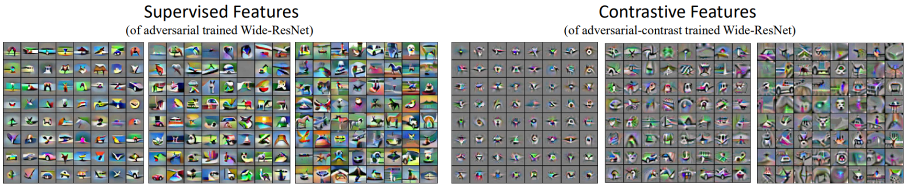
Supervised Features
(of adversarial trained Wide-ResNet)
Contrastive Features
(of adversarial-contrast trained Wide-ResNet)

Figure 2: Supervised versus unsupervised [WL21]

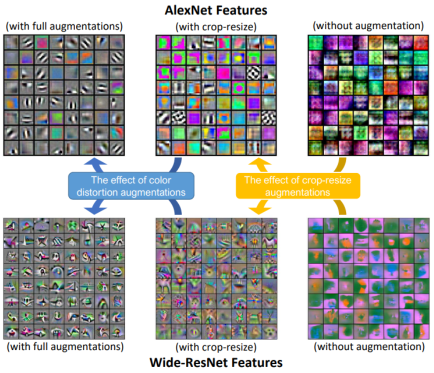
AlexNet Features
(with full augmentations)
(with crop-resize)
(without augmentation)
The effect of color distortion augmentations
The effect of crop-resize augmentations
(with full augmentations)
(with crop-resize)
(without augmentation)
Wide-ResNet Features

Figure 3: Contrastive Features Depending of Augmentation [WL21]

2 Setup to the Analysis

This paper argues that data augmentation is necessary for feature learning of contrastive learning for embedding with two-layer neural network architecture. For the theoretical analysis, we first model our data generation by sparse coding model. We use the random mask for our data augmentation of contrastive learning. By using SGD on the weights of the neural network on contrastive loss function, we obtain an embedding. In the end, we prove that training without augmentation gives a small enough loss function, but learns bad features while training with augmentation gives a small enough loss function and learns good features at the same time. We also check the performance of two downstream tasks - regression and classification - in the embedded space to conclude embedding learned through augmentation is better.

2.1 The Sparse Coding Model

In this paper, we assume the data is generated through a sparse coding model. Our data x(for example an image) is in $\mathbb{R}^{d_1}$ that is generated as sum of two terms

$$x = Mz + \xi \sim \mathcal{D}_x, \qquad z \sim \mathcal{D}_z, \qquad \xi \sim \mathcal{D}_\xi = \mathcal{N}(0, \sigma_\xi^2 I_{d_1})$$

here, the first term Mz is called the sparse signal that is a product of dictionary matrix $M = [M_1, \cdots, M_d] \in \mathbb{R}^{d_1 \times d}$ with the sparse latent variable $z \in \mathbb{R}^d$. The dictionary matrix M is column-wise orthonormal that contains key features of the data generation The latent variable $z = (z_1, \cdots, z_d)^T \in \{-1, 0, 1\}^d$ is sparse in sense that the probability that each element is nonzero is small in scale $\Theta(\frac{loglog(d)}{d})$. M is a fat matrix where $d_1 = poly(d)$. ξ is called the spurious dense noise that has variance of scale $\sigma_\xi^2 = \Theta(\frac{\sqrt{log(d)}}{d})$.

This sparse model implies that by near orthogonality of high dimensional vectors, the spurious noise dominates the sparse signal as $||\xi||^2 \geq \Omega(poly(d)) \gg ||Mz||_2^2$ with high probability. This makes the problem non-trivial. However, the correlation between the sparse signal and the feature remains higher than the correlation between the spurious data and the feature: If $z_j \neq 0$, $|\langle Mz, M_j\rangle| \geq \Omega(1)$ and $|\langle \xi, M_j\rangle| \leq \tilde{\mathcal{O}}(\frac{1}{\sqrt{d}})$ with high probability.

2.2 The Network Architecture

We model the embedding function $f : \mathbb{R}^{d_1} \longrightarrow \mathbb{R}^m$ parameterized by two layer neural network with ReLU activation as

$$f(x) = (h_1(x), \cdots, h_m(x))^T \in \mathbb{R}^m$$
$$h_i(x) = ReLU(< w_i, x > -b_i) - ReLU(- < w_i, x > -b_i)$$

Here, $w_i \in \mathbb{R}_1^d$ are the learned features. If the embedder successfully learned the original features in the dictionary matrix, each column M_i of the dictionary matrix should have a large correlation to the learned features w_i. Thus, it is natural to look at the feature decomposition as follows.

$$w_i = \sum_{j\in[d]} < w_i, M_j > M_j + \sum_{j\in[d_1]\setminus[d]} < w_i, M_j^\perp > M_j^\perp$$

where $M = [M_j]_{j\in[d]}$ is called the sparse features we want to learn and $M^\perp = [M_j]_{j\in[d_1]\setminus[d]}$ is spurious dense features that we want to avoid to learn.

2.3 Contrastive Loss Function

For the construction of the loss function, we first slightly modify the similarity measure of embedding as

$$Sim_f(x, x') :=< f(x), StopGrad(f(x')) >$$

where StopGrad means not to regard the term as constant when calculating the derivative. This is based on the empirical analysis from [CH20] preventing the learning ends too quickly and collapse. Using this modified similarity measure, we define the contrastive loss function given positive pair x_p, x_p' and a batch of samples $\mathfrak{N}$ that forms a negative pair to x_p as the following formula.

$$\mathcal{L}(f, x_p, x_p', \mathfrak{N}) := -\tau log\left(\frac{e^{Sim_f(x_p,x_p')/\tau}}{\sum_{x\in\mathcal{B}} e^{Sim_f(x_p,x)/\tau}}\right)$$

where $\mathcal{B} = \mathfrak{N} \cup \{x_p'\}$. Reducing this implicates enlarging the similarity between (x_p, x_p') compared to any other pairs (x_p, x) for $x \in \mathfrak{N}$. The hyper-parameter τ is called the temperature.

2.4 Data Augmentation: RamdomMask

For the augmentation of the data, we use a method called a random mask, which is essentially whitening approximately half of the coordinate to 0. Let $x_p \sim \mathcal{D}_x$ be generated from the sparse coding model. We generate $D \sim \mathcal{D}_D$ through $D = diag(D_{(l,l)})_{l\in[d_1]} \sim \mathcal{D}_D$ where $D_{(l,l)} \sim Bernoulli(\frac{1}{2})$ and generate augmented data as

$$x_p^+ = 2Dx_p, \qquad x_p^{++} = 2(I - D)x_p$$

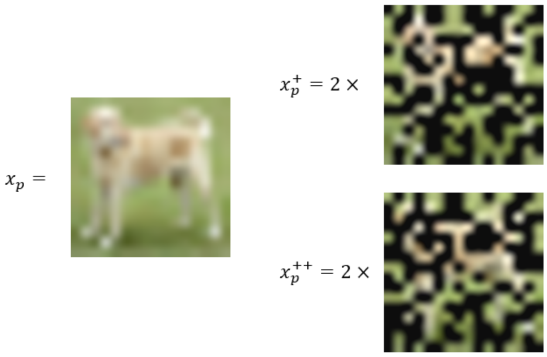

Figure 4: Data Augmentation: Random Mask

For example, pictorial explanation is given as figure 4

2.5 Training Algorithm

We construct two different losses for each case with and without augmentation. For the case **with augmentation**, contrastive learner f_t at iteration t parameterized by $w_i^{(t)}, b_i^{(t)}$ is updated as following.

$$
\begin{aligned}
L(f_t) &= \mathbb{E}_{x_p^+, x_p^{++}, \mathfrak{R}}[\mathcal{L}(f_t, x_p^+, x_p^{++}, \mathfrak{R})] \\
\textbf{Obj}(f_t) &= L(f_t) + \frac{\lambda}{2} \sum_{i \in [m]} ||w_i^{(t)}||_2^2 \\
w_i^{(t+1)} &= w_i^{(t)} - \eta \nabla_{w_i} \textbf{Obj}(f_t)
\end{aligned}
$$

and in contrast for the **without augmentation** case, contrastive learner f_t^{NA} at iteration t parameterized by $w_i^{(t)}, b_i^{(t)}$ is updated as

$$
\begin{aligned}
L_{NA}(f_t^{NA}) &= \mathbb{E}_{x_p, \mathfrak{R}}[\mathcal{L}(f_t^{NA}, x_p, x_p, \mathfrak{R})] \\
\textbf{Obj}_{NA}(f_t^{NA}) &= L_{NA}(f_t^{NA}) + \frac{\lambda}{2} \sum_{i \in [m]} ||w_i^{(t)}||_2^2 \\
w_i^{(t+1)} &= w_i^{(t)} - \eta \nabla_{w_i} \textbf{Obj}_{NA}(f_t^{NA})
\end{aligned}
$$

The main difference between the to is the input of $\mathcal{L}$ to be either augmented x_p^+, x_p^{++} or the original x_p, x_p. The objective is a regularized loss of L, where L is given as the expectation of $\mathcal{L}$. Thus to calculate the gradient over expected value, **stochastic gradient descent** using a sampling of either $x_p^+, x_p^{++}, \mathfrak{R}$ or $x_p, \mathfrak{R}$ is conducted at each iteration. Also, note that b is not updated through the gradient methods, but manually updated using w.

2.6 Downstream Tasks

For the measure of good embedding, we provide the performance on two downstream tasks - regression and binary classification. For both problems, labels are generated by a linear function or linear classifier respectively by the latent variable z. We can regard z as a variable containing the key feature so it is natural to give such a label. In detail, Take $w^\star \in \mathbb{R}^d$

- **Regression** For $x = Mz + \xi \sim \mathcal{D}_x$, give label $y = \langle w^\star, z \rangle$
- **Binary Classification** For $x = Mz + \xi \sim \mathcal{D}_x$, give label $y = sign(\langle w^\star, z \rangle)$

and tries to find the minimizer w by the square loss - for the regression, and logistic loss - for the regression.

3 Results

3.1 Without Augmentation

Without augmentation, the following theorem states that we learn dense features.

Theorem For $x = Mz + \xi \sim \mathcal{D}_x$, with high probability

$$\left\langle \frac{f_t^{NA}(x)}{||f_t^{NA}(x)||_2}, \frac{f_t^{NA}(\xi)}{||f_t^{NA}(\xi)||_2} \right\rangle \geq 1 - \tilde{\mathcal{O}}\left(\frac{1}{poly(d)}\right)$$

which means Mz is overwhelmed by ξ in representation by $f(\cdot)$.

This gives the corollary stating that downstream tasks fail by a constant error when done on an embedded space trained in the non-augmentation case. The result is given as

- Regression
$$\mathbb{E}_{x\sim\mathcal{D}_x}|y - \langle w^*, f_t^{NA}\rangle(x)|^2 \geq \Omega(1)$$
- Classification
$$\mathbf{Pr}_{x\sim\mathcal{D}_x}[y = sign(\langle w^*, f_t^{NA}(x)\rangle)] = o(1)$$

3.2 With Augmentation

For the case with augmentation through random mask, we have the following theorem

Theorem For $i \in [m], t \in [\frac{d^1.01}{\eta}, \frac{d^1.99}{\eta}] = I$, learned feature has following decomposition

$$w_i^{(t)} = \sum_{j\in\mathcal{N}_i} \alpha_{i,j} M_j + \sum_{j\notin\mathcal{N}_i} \alpha'_{i,j} M_j + \sum_{j\in[d_1]\setminus[d]} \beta_{i,j} M_j^\perp$$

where $\alpha_{i,j} \in [\frac{\tau}{d^c}, \tau]$ for $c < 1/1000$, $\alpha'_{i,j} \leq o(\frac{1}{\sqrt{d}})||w_i^{(t)}||^2$, $|\mathcal{N}_i| = \mathcal{O}(1)$, $|\beta_{i,j}| \leq o(\frac{1}{\sqrt{d}})||w_i^{(t)}||^2$, and $\Omega(1) = |\{i \in [m] | j \in \mathcal{N}_i\}| = o(\frac{m}{d})$. To make a long story short, each column of the dictionary matrix M is reflected in $\Omega(1)$ number of w_i and each w_i reflects $\mathcal{O}(1)$ number of columns in the dictionary matrix. Columns M_j for $j \in \mathcal{N}_i$ has a high reflection to w_i and the other directions have a low reflection.

As a corollary, we obtain a successful guarantee of downstream tasks in the embedded space. The result is given as For $t \in I$

- **Regression.** With samples at most $\tilde{\mathcal{O}}(d^{1.001})$, we can obtain $w^* \in \mathbf{R}^m$ such that
$$\mathbb{E}_{x\sim\mathcal{D}_x}|y - \langle w^*, f_t(x)\rangle|^2 = o(1)$$
- **Classification.** With samples at most $\tilde{\mathcal{O}}(d^{1.001})$, we can obtain $w^* \in \mathbf{R}^m$ such that
$$\mathbf{Pr}_{x\sim\mathcal{D}_x}[y = sign(\langle w^*, f_t(x)\rangle)] = 1 - o(1)$$

4 Discussions and Further Directions

4.1 Intuition on the Proof

First, for the without augmentation case, suppose $x = Mz + \xi, x' = Mz' + \xi'$. Then, $\langle \xi, \xi \rangle$ is large while $\langle \xi, \xi' \rangle$ is small

4.2 Summary

In summary, we conclude under the sparse coding model, the contrastive learning *fails* to learn good representation without augmentation while *success* to learn good representation with augmentation. This could be checked through theorems where embedding was overwhelmed by the spurious dense noise for the case without augmentation and the feature decomposition showed a large relation with the columns of the dictionary matrix for the with augmentation case. Also, the guarantee for the two downstream tasks showed that data augmentation is necessary for contrastive learning to learn good features.

4.3 Related Topics

- It is known that supervised data can be augmented as well to provide a richer data set. The framework behind this intuition gives the algorithm for supervised contrastive learning, which helps resolve the problems in self-supervised contrastive learning such as the collapse of the model. It would be an interesting topic to also research how the learned features change when we use data augmentation in supervised contrastive learning.

- Still there are many assumptions in the current analysis. We are not fully training the network, as we are manually updating the bias term using the weight in the neural network. Also, the dimension assumption of each data given as the polynomial of the dimension of the latent variable may not be reasonable sometimes. Thus, reducing such constraints might be a straightforward work to do.

- Whether this feature learning is due to the specific choice of random mask data augmentation is one question to ask. There are many kinds of data augmentation used in practice, and it is considerable to research the implications of other kinds of data augmentation.

- Finally, we can question what happens to data generated in a way different from the sparse coding model.

5 References

References

[Che+20] Ting Chen et al. *A Simple Framework for Contrastive Learning of Visual Representations.* 2020. arXiv: `2002.05709 [cs.LG]`.

[CH20] Xinlei Chen and Kaiming He. *Exploring Simple Siamese Representation Learning.* 2020. arXiv: `2011.10566 [cs.CV]`.

[WL21] Zixin Wen and Yuanzhi Li. *Toward Understanding the Feature Learning Process of Self-supervised Contrastive Learning.* 2021. arXiv: `2105.15134 [cs.LG]`.