Towards Understanding Self-supervised Learning / Pretraining / Foundation Models

Tengyu Ma (Stanford)

Jeff Z. HaoChen  Colin Wei  Kendrick Shen  Robbie Jones  Ananya Kumar  Sang Michael Xie  Adrien Gaidon, and Percy Liang
Pretrained Representations

- Pretraining: learn representations from **unlabeled data**
- Adaptation: train linear classifiers using the representations as features on **labeled** downstream tasks

Images from shutterstock.com
Pretrained representations may “universally” help a wide range of downstream tasks.

**Computer vision:**
- Pretraining data: millions to billions of unlabeled images from the web
- Downstream tasks: classification, object detection, medical diagnoses, ..
- Downstream domains: street views, videos, cartoons, bioimaging data, ...

**Language:**
- Pretraining data: > 100 GB texts (mixture of books, crawled webpages, Wikipedia ...)
- Downstream tasks: parsing, translation, sentiment analysis, topic classification, question answering, reasoning, ...
- Downstream domains: legal, scientific, media, dialogue, multilingual ...
AI is undergoing a paradigm shift with the rise of models (e.g., BERT, DALL-E, GPT-3) that are trained on broad data at scale and are adaptable to a wide range of downstream tasks. We call these models foundation models to underscore their critically central yet incomplete character.
Why/When/How Does Foundation Model Work?

- Why does pretraining on unlabeled data with an unsupervised (self-supervised) loss help a wide range of downstream tasks?
  - pretraining helps label efficiency of downstream tasks
  - pretraining can give “universal” representations
  - pretraining provides robustness to distribution shift

- Meta question:
  A theoretical framework for us to gradually understanding these questions?

- Classical learning theory requires the same training and test task
This Talk

• Analysis for Contrastive learning and A Framework
  [Provable Guarantees for Self-Supervised Deep Learning with Spectral Contrastive Loss, Haochen-Wei-Gaidon-M.’21]
  Section 4.10 of the foundation model report [Bommasani et a.’21]

• Some follow-up works using the framework
  [Connect, Not Collapse: Explaining Contrastive Learning for Unsupervised Domain Adaptation, Shen-Jones-Kumar-Xie-HaoChen-M.-Liang.’22]
  [Beyond Separability: Analyzing the Linear Transferability of Contrastive Representations to Related Subpopulations, HaoChen-Wei-Kumar-M.’22]
Pretraining Visual Representations with Contrastive Learning

- Pull representations of augmentations of the same image closer
- Push representations of augmentations of different images further

Various implementations: MoCo [He et al.'19], BYOL [Grill et al.'20], SimSiam [Chen et al.'20], SwAV [Caron et al.'20]
Contrastive Learning $\approx$ Spectral Clustering on an Infinite Graph

Roadmap

1. Population (infinite) pretraining data case (without parameterizations)
2. Finite data with neural network approximations
Population Positive-Pair Graph

- Vertex set: all images patches
- Edges: connect two patches if they can share an original image (i.e. they are positive pairs)
- Simplified case: augmentation = small random perturbation
  - Positive-pair graph = proximity graph with $\ell_2$ metric
- Positive-pair graph is very sparse
Clustering Structures: Sub-clusters with Good Intra-connectivity

- Very few edges between different underlying classes
- Connectivity/expansions within the same classes or sub-classes
- Two bulldogs can be connected via a sequence of bulldogs
Formal Definitions

- Draw a natural image $\tilde{x} \in \tilde{X}$ from $P$; then draw two aug. of $\tilde{x}$
- Define graph $G$ with weight $w_{zz'} = $ the probability/density that $(z, z')$ is a positive pair
- E.g., if augmentation is small Gaussian perturbation, then $w_{zz'}$ is big when $(z, z')$ are very close in $\ell_2$ distance
A Simplified Setting

- Data are supported on a mixture of manifolds
  - e.g., mixture of Lipchitz nonlinear transformation of Gaussians

- Augmentation= Gaussian perturbation
  - \[ \implies \text{Positive-pair graph = proximity / geometric graph (aka connecting nearby points)} \]

- We will assume
  - Connectivity within the manifolds: Cheeger’s constant / isoperimetric number of each manifold is larger than \(1/\text{poly}(d)\)
  - Separation between the manifolds
Empirical Justification: Each ImageNet Class Forms Connected Manifolds

- Random walk in the latent space of BigGAN for each class

- Each manifold is quite connected!
- Random walk across classes may have more abrupt changes

**NB:**

- stronger augmentation improves the connectivity
- we also allow sub-clusters

Videos adapted from https://www.youtube.com/watch?v=YY6LrQSxIbc and https://www.youtube.com/watch?v=f47GdDs6LxY&t=898s
Main Results:
Contrastive Learning $\approx$ Spectral Clustering on Positive-Pair Graph

**Theorem** (informal):

With infinite data, representations $f(\cdot)$ learnt by contrastive learning are embeddings of the positive–pair graph by spectral clustering.

That is, the embedding matrix $F$ is a low-rank approximation of the adjacency matrix of the graph.

$$F = \begin{bmatrix} -f(x)^\top & \vdots \\ \vdots \\ -f(x')^\top & \vdots \end{bmatrix}_{x \in \mathcal{X}} \in \mathbb{R}^{|\mathcal{X}| \times k}$$

- We analyze spectral contrastive loss that also works empirically

$$\min_f L(f) = -2 \mathbb{E}_{x,x^+} f(x)^\top f(x^+) + \mathbb{E}_{x,x'} (f(x)^\top f(x'))^2$$

  positive pair (aug. of same image)

  random pair (aug. of random pairs of images)
Our “spectral contrastive loss” achieves performance comparable to state-of-the-art methods.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>CIFAR-10</th>
<th>CIFAR-100</th>
<th>Tiny-ImageNet</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Epochs</strong></td>
<td>200</td>
<td>400</td>
<td>800</td>
</tr>
<tr>
<td>SimCLR (repro.)</td>
<td>83.73</td>
<td>87.72</td>
<td>90.60</td>
</tr>
<tr>
<td>SimSiam (repro.)</td>
<td>87.54</td>
<td><strong>90.31</strong></td>
<td>91.40</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td><strong>88.66</strong></td>
<td>90.17</td>
<td><strong>92.07</strong></td>
</tr>
</tbody>
</table>

Table 1: Top-1 accuracy under linear evaluation protocol.

<table>
<thead>
<tr>
<th>acc. (%)</th>
<th>SimCLR</th>
<th>BYOL</th>
<th>MoCo v2</th>
<th>SimSiam</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>66.5</td>
<td>66.5</td>
<td>67.4</td>
<td>68.1</td>
<td><strong>66.97</strong></td>
</tr>
</tbody>
</table>

Table 2: ImageNet linear evaluation accuracy with 100-epoch pre-training.

- We use batch size of **384**, which is much smaller than SimCLR (*4096*).
- Our algorithm doesn’t require tricks like momentum encoding in BYOL and MoCo v2, or stop-gradient operation in SimSiam.
Recall: eigenvectors relate to graph decompositions

**Theorem** (informal): Suppose the positive-pair graph contains $r$ major clusters and representation dimension $k \geq 2r$. Then, linear classification on representations can solve any downstream task s.t. each cluster has the same label.

- A new but simple proof, using spectral graph theory tools
- Past works on spectral graph theory, stochastic block models, etc., don’t analyze linear separability of the embeddings
Intuitions: a Stylized Case with Completely Disjoint Clusters

- Eigenvectors encodes the cluster-membership information in the columns.
- Rows of eigenvectors (up to scaling) are one-hot embeddings based on the cluster ID.
- The final embeddings learnt will be a rotation of this matrix.
Contrastive Learning $\approx$ Spectral Clustering on an Infinite Graph

Roadmap

1. Population (infinite) pretraining data case (without parameterizations)
2. Finite data with neural network approximations
   - parameterized $f$ with parameter $\theta$
   - empirical loss $(f_\theta) \approx$ population loss $(f_\theta)$ when $n \geq \text{complexity}(\{f_\theta\})$
Theoretical Framework for Analyzing Foundation Models: Suffices to Understand Population Case?

Abstracted away:

- standard generalization: differences between $P$ and $\hat{P}$, $D$ and $\hat{D}$
- success of optimizations

Section 4.10 (theory section) of foundation model paper [Bommasani et al. 21]
Representations Can Also Capture the Relationship Between Clusters
Pretraining features + linear classification gives SOTA performance for unsupervised domain adaptation [Shen-Jones-Kumar-Xie-HaoChen-M.-Liang.’22]

More general analysis in HaoChen-Wei-Kumar-M.’[2022]
Part III: Self-Supervised vs Supervised Representations: Robustness/Diversity

[Self-supervised Learning is More Robust to Dataset Imbalance. Liu et al.’21]
Contrastive (Self-Supervised, SSL) Representations are More Robust to Imbalance Dataset Than Supervised Representations

- Imbalanced data with a long tail class distribution
- Balanced-imbalanced gap (smaller the better)
  - $\Delta^{SSL}$: (balance SSL – imbalanced SSL)/balanced SSL
  - $\Delta^{SL}$: (balance SL – imbalanced SL)/balanced SL

performance measured by finetuning on OOD data or linear probe on ID data
SSL still works for extreme imbalance, when rare classes are almost not seen at all from pretraining.

⇒ SSL learns information from frequent data that can help rare data (more than supervised representations does).

Main intuition (more theory in [Liu et al.’21]):

Ø SL only learns label-relevant features
Ø SSL also learns label-irrelevant-but-transferable features

performance measured by finetuning on OOD data or linear probe on ID data
Summary

- A graph partition perspective
  - contrastive learning $\approx$ spectral clustering on infinite graph
  - contrastive representations also capture semantic relationship between clusters
- SSL learns richer features, and thus is more robust than SL representations

Other works on pretraining/self-supervised learning/foundation models by my group

- analysis for self-training [Wei-Shen-Chen-M., ICLR’21, Chen-Wei-Kumar-M. ICML’21]
- self-training with auxiliary data [Xie et al., ICLR’21]
- robust finetuning for OOD performance [Kumar-Raghunathan-M.-Liang, ICLR’22]
- imbalanced self-supervised learning [Hong-Haochen-Adrien-M. ICLR’22]
- language problems
  - SSL + linear probe/prompt tuning for HMMs and its variants [Wei-Xie-M., NeuRIPS’21]
  - in-context learning [Xie-Raghunathan-Liang-M., ICLR’22]
Fine-tuning distorts pretrained features and underperforms out-of-distribution

<table>
<thead>
<tr>
<th></th>
<th>Pretraining</th>
<th>(a) Fine-tuning</th>
<th>(b) Linear probing</th>
<th>(c) LP-FT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Features</td>
<td>Randomly initialized head</td>
<td>Randomly initialized head</td>
<td>Frozen Features</td>
</tr>
<tr>
<td></td>
<td>Inputs</td>
<td>Backprop</td>
<td>Backprop</td>
<td>Initialize head</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Backprop</td>
</tr>
<tr>
<td>ID test</td>
<td>85.8%</td>
<td>83.5%</td>
<td>86.5%</td>
<td></td>
</tr>
<tr>
<td>OOD test</td>
<td>66.8%</td>
<td>71.6%</td>
<td>74.4%</td>
<td></td>
</tr>
</tbody>
</table>

Average accuracies (6 datasets)