Self-supervised learning: Generative or Contrastive

Seri Lee
Computer Science and Engineering
Seoul National University
Seoul, Republic of Korea
sally20921@snu.ac.kr

***Abstract*—Deep supervised learning has achieved great success in the last decade. However, its deficiencies of dependence on manual labels and vulnerability to attacks have driven people to explore a better solution. As an alternative, self-supervised learning (SSL) attracts many researchers for its soaring performance on representation learning in the last several years. Self-supervised representation learning leverages input data itself as supervision and benefits almost all types of downstream tasks. In this survey, we take a look into new self-supervised learning methods for representation in computer vision, natural language processing, and graph learning. We comprehensively review the existing empirical methods and summarize them into three main categories according to their objectives: generative, contrastive, and generative-contrastive (adversarial). We further investigate related theoretical analysis work to provide deeper thoughts on how self-supervision works. Finally, we briefly discuss open problems and future directions for self-supervised learning.**

***Index Terms*—Self-supervised Learning, Generative Model, Contrastive Learning, Deep Learning**

I. Introduction

Deep neural networks [1] have shown outstanding performance on various machine learning tasks, especially on supervised learning in computer vision (image classification, semantic segmentation), natural language processing (pre-trained language models, sentiment analysis, question answering) and graph learning (node classification, graph classification). Generally, the supervised learning is trained over a specific task with a large manually labeled dataset which is randomly divided into training, validation, and test sets.

However, supervised learning is meeting its bottleneck. It not only relies heavily on expensive manual labeling but also suffers from generalization error, spurious correlations, and adversarial attacks. We expect the neural network to learn more with fewer labels, fewer samples, or fewer trials. As a promising candidate, self-supervised learning has drawn massive attention for its fantastic data efficiency and generalization ability, with many state-of-the-art models following this paradigm. In this survey, we will take a comprehensive look at the development of the recent self-supervised learning models and discuss their theoretical soundness, including frameworks such as Pre-trained Language Models (PTM), Generative Adversarial Networks (GAN), Autoencoder and its extensions, Deep Infomax, and Contrastive Coding.

The term “self-supervised learning” was first introduced in robotics, where the training data is automatically labeled by finding and exploring the relations between different input sensor signals. It was then borrowed by the field of machine learning. In a speech on AAAI 2020, Yann LeCun described self-supervised learning as “the machine predicts any parts of its input for any observed part”. We can summarize them into two classical definitions following LeCun's:

- Obtain “labels” from the data itself by using a “semi-automatic” process.
- Predict part of the data from other parts.

Specifically, the “other part” here could be incomplete, transformed, distorted, or corrupted. In other words, the machine learns to ‘recover’ whole, or parts of, or merely some features of its original input.

People are often confused by unsupervised learning and self-supervised learning. Self-supervised learning can be viewed as a branch of unsupervised learning since there is no manual label involved. However, narrowly speaking, unsupervised learning concentrates on detecting specific data patterns, such as clustering, community discovery, or anomaly detection, while self-supervised learning aims at recovering, which is still in the paradigm of supervised settings.

There exist several comprehensive reviews related to Pre-trained Language Models, Generative Adversarial Networks, Autoencoder and contrastive learning for visual representation. However, none of them concentrates on the inspiring idea of self-supervised learning that illustrates researchers in many fields. In this work, we collect studies from natural language processing, computer vision, and graph learning in recent years to present an up-to-date and comprehensive retrospective on the frontier of self-supervised learning.

II. Background

A. Representation Learning in NLP

Pretrained word representations are key components in natural language processing tasks. Word embeddings are to represent words as low-dimensional real-valued vectors. There are two kinds of word embeddings: non-contextual and contextual embeddings.

Non-contextual embeddings do not consider the context information of the token; that is, these models only map the token into a distributed embedding space. Thus, for each word x in the vocabulary V, embedding will assign it a specific vector $e_x \in R_d$, where d is the dimension of the embedding. These embeddings can not model complex characteristics of word usage and polysemy.

To model both complex characteristics of word usage and polysemy, contextual embedding is proposed. For a text sequence $x_1, x_2, \ldots, x_N, x_n \in V$, the contextual embedding of x_n depends on the whole sequence.

$$[e_1, e_2, \ldots, e_N] = f(x_1, x_2, \ldots, x_N) \tag{1}$$

where $f(\cdot)$ is the embedding function. Since for a certain token x_i, the embedding e_i can be different if x_i in difference context, e_i is called contextual embedding. This kind of embedding can distinguish the semantics of words in different contexts.

Distributed word embeddings represent each word as a dense, real-valued, and low-dimensional vector. The first-generation word embedding is introduced as a neural network language model (NNLM). For NNLM, most of the complexity derives from the non-linear hidden layer in the model. Mikolov et al. proposed Word2Vec Model to learn the word representations efficiently. There are two kinds of implementations: Continuous Bag-of-Words Model (CBOW) and Continuous Skip-gram Model (SG). As a kind of context prediction model, Word2Vec is one of the most popular implementations to generate non–contextual word embeddings for NLP.

In the first-generation word embedding, the same word has the same embedding. Since a word can have multiple senses, therefore, the second-generation word embedding methods are proposed. In this case, each word token has its embedding. These embeddings are also called contextualized word embedding since the embeddings of word tokens depend on its contexts. ELMo(Embeddings from Language Model) is an implementation to generate those contextual word embeddings. It is an RNN-based bidirectional language model which learns multiple embeddings for based on downstream tasks. ElMo is a feature-based approach, that is, the model is used as a feature extractor to extract word embedding, and send those embeddings to the downstream task model. The parameters of the extractor are fixed, and only the parameters in the backend model can be trained.

Recently, BERT (Bidirectional Encoder Representations from Transformers) brings large improvements on 11 NLP tasks. Different from feature-based approaches like ELMo, BERT is a fine-tuned approach. The model is first pretrained on a large amount of corpora through self-supervised learning, and then fine-tuned with labeled data. As the name indicates, BERT uses Transformer as its encoder. In the training stage, BERT masks some tokens in the sentence, and is then trained to predict the masked words. When using BERT, we first initialize the BERT model with pretrained weights, and then fine-tune the pretrained model on downstream tasks.

References

[1] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Zhaoyu Wang, Li Mian, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *arXiv preprint arXiv:2006.08218*, 1(2), 2020.