



Open in app

Get started



Published in Towards Data Science

You have **2** free member-only stories left this month. [Sign up for Medium and get an extra one](#)



Ekin Tiu

Follow

Jan 7, 2021 · 8 min read ★ · Listen



Save



# Understanding Contrastive Learning

Learn how to learn without labels using self-supervised learning.





Open in app

Get started



Photo by [Raquel Martínez](#) on [Unsplash](#)

## What is Contrastive Learning?

**Contrastive learning** is a machine learning technique used to learn the *general features* of a dataset **without labels** by teaching the model which data points are similar or different.

Let's begin with a simplistic example. Imagine that you are a newborn baby that is trying to make sense of the world. At home, let's assume you have two cats and one dog.

Even though *no one tells you* that they are 'cats' and 'dogs', you may still realize that the two cats look similar compared to the dog.





Open in app

Get started



(Left) Photo by [Edgar](#) on [Unsplash](#) | (Right Top) Photo by [Lana Vidnova](#) from [Unsplash](#) | (Right Bottom) Image by [Ruby Schmank](#) from [Unsplash](#)

By merely recognizing the similarities and differences between our furry friends, our brains can learn the *high-level features* of the objects in our world.

For instance, we may subconsciously recognize that the two cats' have pointy ears, whereas the dog has droopy ears. Or we may *contrast (hint-hint)* the protruding nose of the dog to the flat face of the cats.

In essence, **contrastive learning** allows our machine learning model to do the same thing. It looks at which pairs of data points are “*similar*” and “*different*” in order to learn *higher-level features* about the data, *before* even having a task such as classification or segmentation.

*Why is this so powerful?*

It's because we can train the model to *learn a lot* about our data ***without any annotations or labels*** hence the term ***SELF-supervised learning***





Open in app

Get started

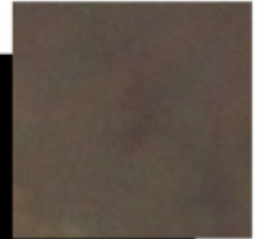
Neovascularization



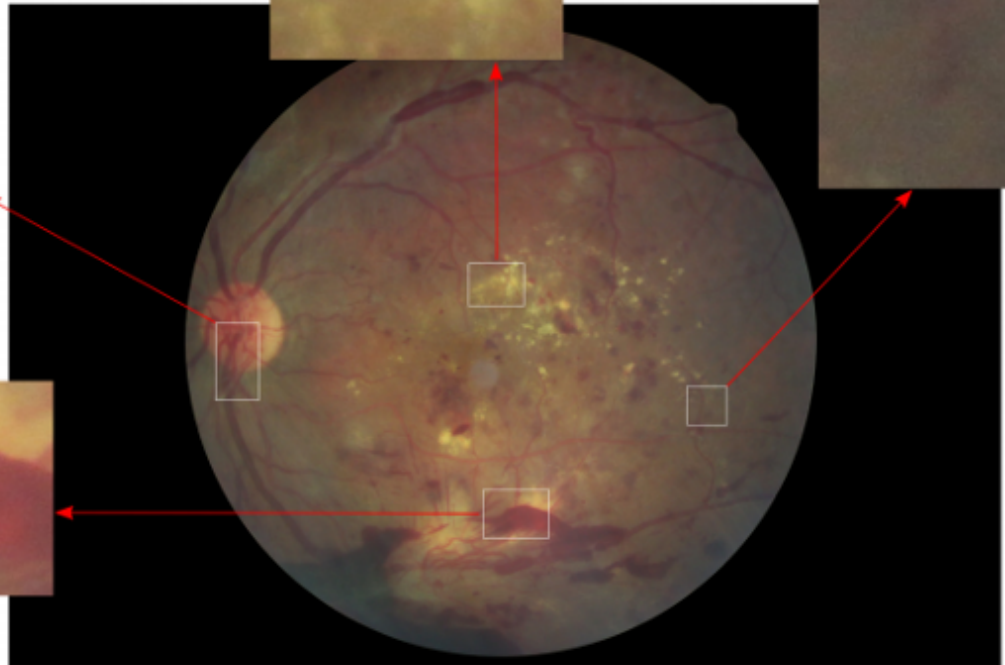
Exudates



Microaneurysms



Haemorrhage



Fundus image of the retina, annotated. Source: [1]

In most real-world scenarios, we *don't* have labels for each image. Take medical imaging, for instance. To create labels, professionals have to spend countless hours looking at images to manually classify, segment, etc.

With contrastive learning, one can significantly improve model performance even when only a fraction of the dataset is labeled.

Now that we understand what contrastive learning is, and why it's useful, let's see *how* contrastive learning works.

## How does Contrastive Learning Work?

In this article, I focus on **SimCLRv2**, one of the recent state-of-the-art contrastive learning approaches proposed by the Google Brain Team. For other contrastive learning methods





Open in app

Get started

## Designing A New Approach

We formulate a framework for characterizing contrastive self-supervised learning approaches and look at AMDIM, CPC...

[towardsdatascience.com](https://towardsdatascience.com)

Fortunately, **SimCLRv2** is very intuitive.

The entire process can be described concisely in three basic steps:

- For each image in our dataset, we can perform two *augmentation combinations* (i.e. crop + resize + recolor, resize + recolor, crop + recolor, etc.). We want the model to learn that these two images are “similar” since they are essentially different versions of the same image.



Figure by Author. Photo by [Edgar](#) on [Unsplash](#)

- To do so, we can feed these two images into our deep learning model (Big-CNN such as ResNet) to create *vector representations* for each image. The goal is to train the

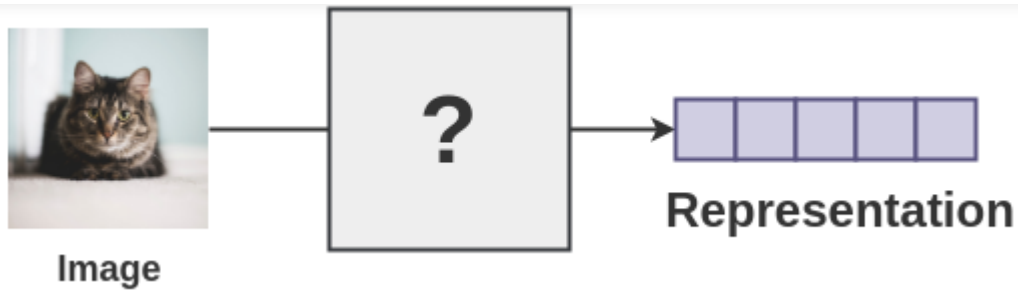






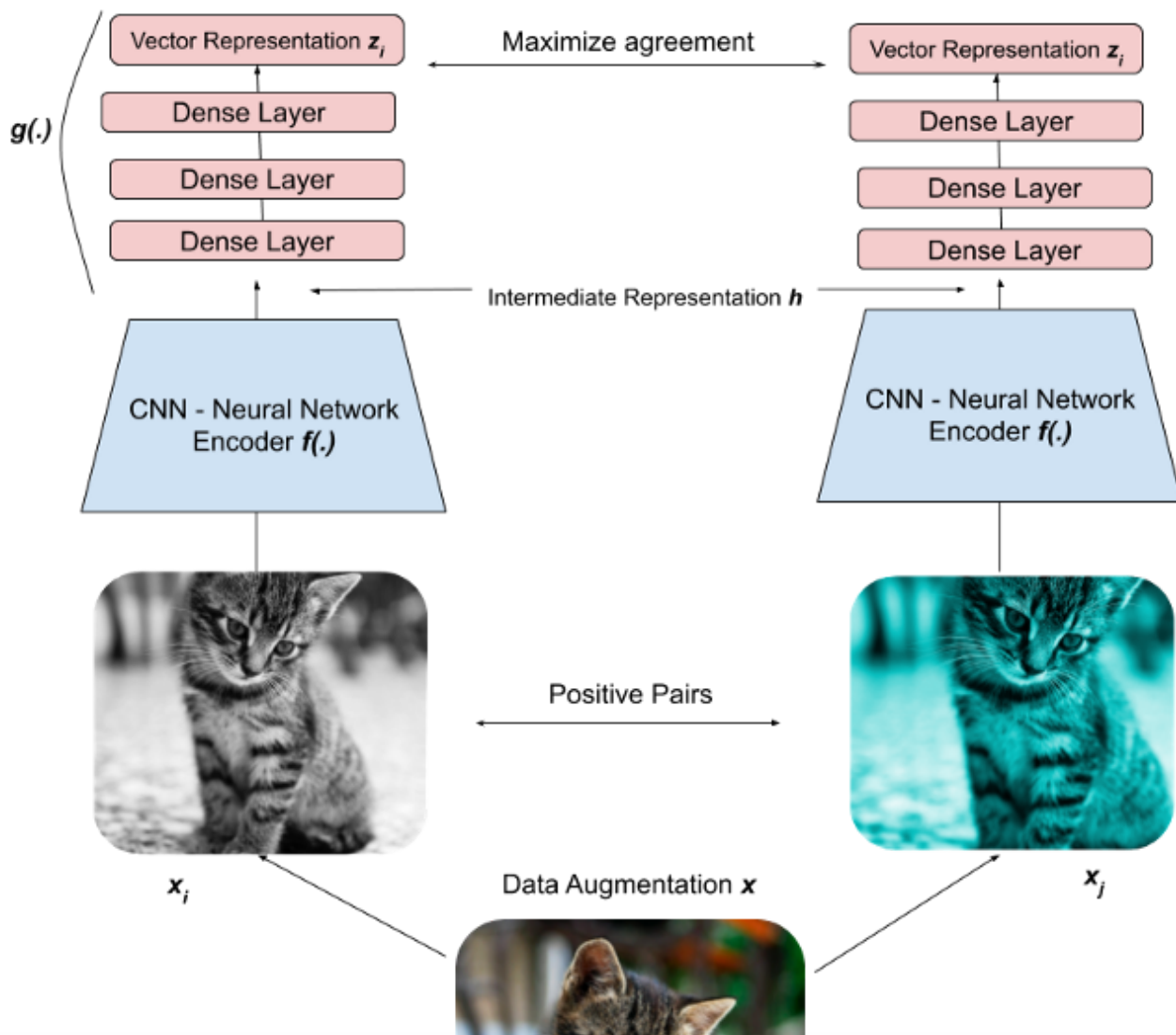
Open in app

Get started



Expressing an image as a vector representation. Source: The Illustrated SimCLR Framework by Amit Chaudhary, [amitnss](#)

- Lastly, we try to *maximize the similarity* of the two vector representations by minimizing a contrastive loss function.



[Open in app](#)[Get started](#)

An overview of the SimCLRv2 framework. Figure by Author. Photo by [Edgar](#) on [Unsplash](#)

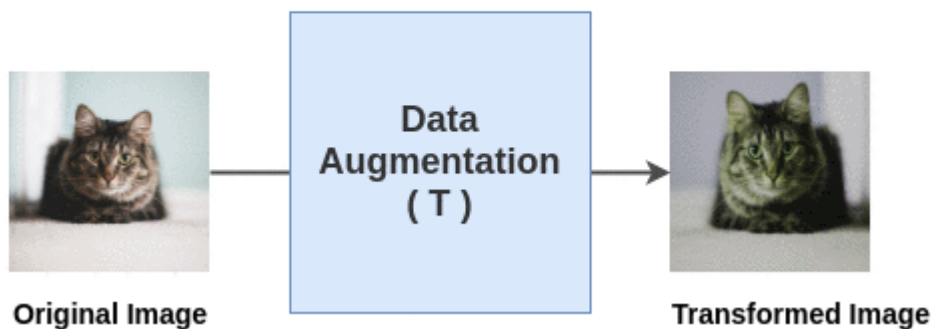
Over time, the model will learn that two images of cats should have similar *representations* and that the representation of a cat should be different than that of a dog.

This implies that the model is able to *distinguish between different types of images* without even knowing what the images are!

We can dissect this contrastive learning approach even further by breaking it down into three primary steps: **data augmentation, encoding, and loss minimization.**

### 1) Data Augmentation

#### Random Transformation



Source: The Illustrated SimCLR Framework by Amit Chaudhary, [amitness](#)

We perform any combination of the following augmentations randomly: crop, resize, color distortion, grayscale. We do this *twice per image in our batch*, to create a **positive pair** of two augmented images.

### 2) Encoding

We then use our Big-CNN neural network, which we can think of as simply a *function*,  $h = f(x)$ , where ' $x$ ' is one of our augmented images, to encode both of our images as vector representations.



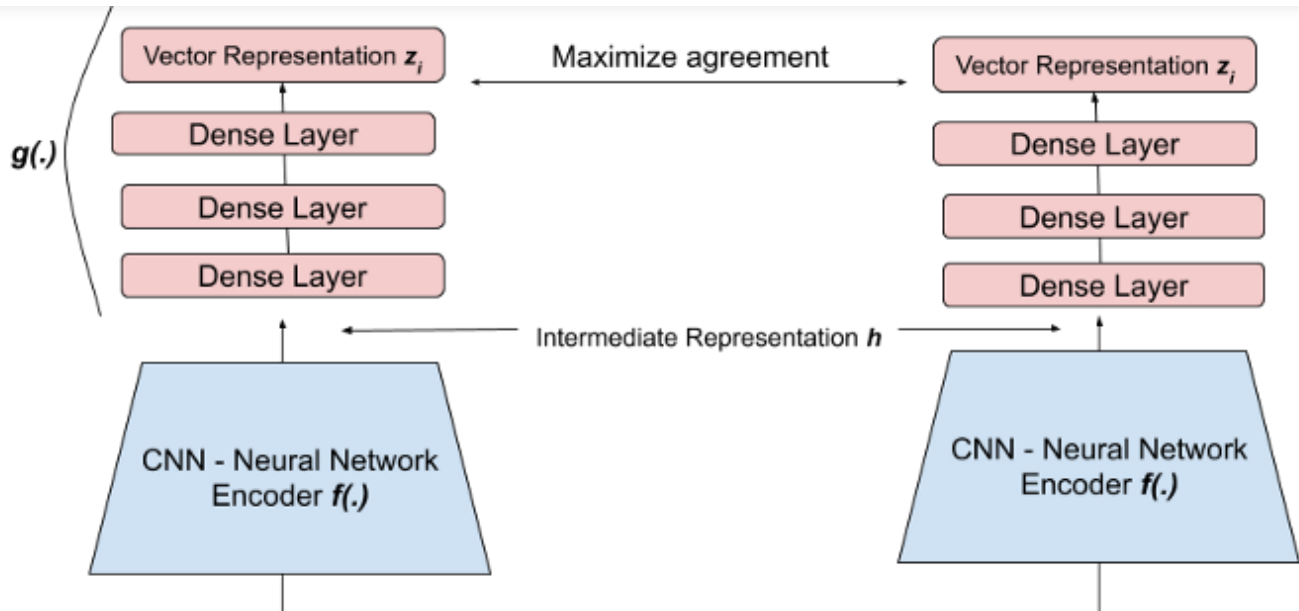
[Open in app](#)[Get started](#)

Image by Author

The output of the CNN is then inputted to a set of Dense Layers called the **projection head**,  $z = g(h)$  to transform the data into another space. This extra step is empirically shown to improve performance [2].

If you are unfamiliar with latent space and vector representations, I highly recommend reading my article that intuitively explains this concept before continuing.

### Understanding Latent Space in Machine Learning

Learn a fundamental, yet often 'hidden,' concept of deep learning

[towardsdatascience.com](https://towardsdatascience.com)

By compressing our images into a latent space representation, the model is able to *learn the high-level features* of the images.

In fact, as we continue to train the model to maximize the vector similarity between similar images, we can imagine that the model is learning *clusters* of similar data points in







Open in app

Get started

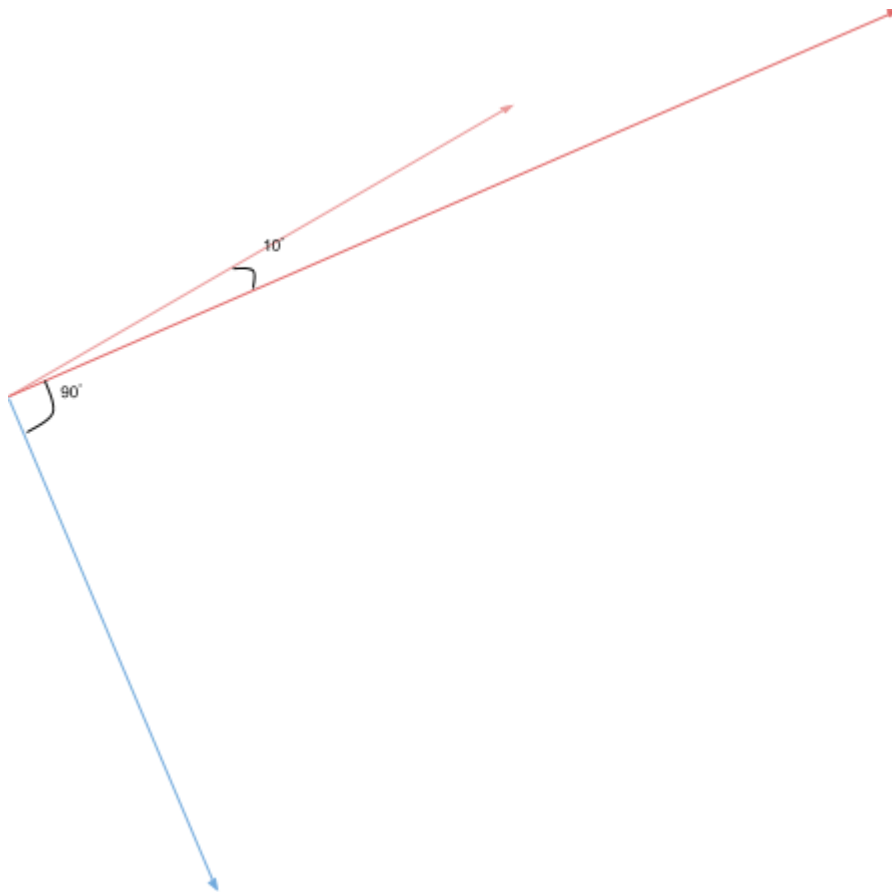
### 3) Loss Minimization of Representations

Now that we have two vectors,  $z$ , we need a way to **quantify the similarity between them**.

similarity(  ,  )

Source: The Illustrated SimCLR Framework by Amit Chaudhary, [amitness](#)

Since we are comparing two vectors, a natural choice is **cosine similarity**, which is based on the *angle between the two vectors in space*.



2D Vectors in space. Image by Author

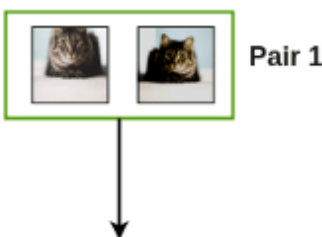


[Open in app](#)[Get started](#)

metric, we will get a high similarity when the angle is close to 0, and a low similarity otherwise, which is exactly what we want.

We also need a **loss function** that we can minimize. One option is NT-Xent (Normalized Temperature-Scaled Cross-Entropy Loss).

We first compute the *probability that the two augmented images are similar*.



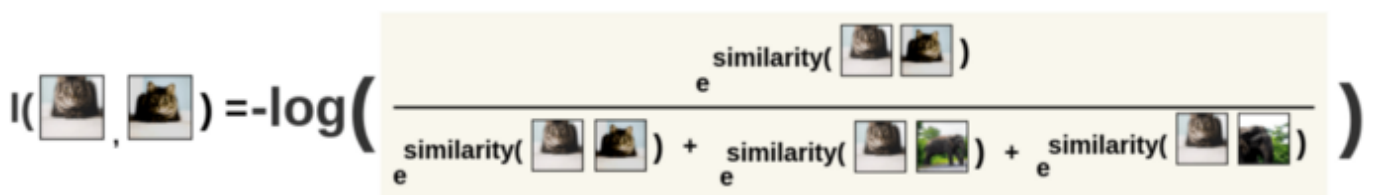
Pair 1

$$\text{Softmax} = \frac{e^{\text{similarity}(\text{Pair 1})}}{e^{\text{similarity}(\text{Pair 1})} + e^{\text{similarity}(\text{Pair 2})} + e^{\text{similarity}(\text{Pair 3})}}$$

Source: The Illustrated SimCLR Framework by Amit Chaudhary, [amitnss](#)

Notice that the denominator is the sum of  $e^{\text{similarity}(\text{all pairs, including negative pairs})}$ . **Negative pairs** are obtained by creating pairs between our augmented images, and all of the other images in our batch.

Lastly, we wrap this value around in a  $-\log()$  so that *minimizing* this loss function corresponds to *maximizing* the probability that the two augmented images are similar.


$$L(\text{Pair 1}) = -\log\left(\frac{e^{\text{similarity}(\text{Pair 1})}}{e^{\text{similarity}(\text{Pair 1})} + e^{\text{similarity}(\text{Pair 2})} + e^{\text{similarity}(\text{Pair 3})}}\right)$$

Source: The Illustrated SimCLR Framework by Amit Chaudhary, [amitnss](#)

For more details about the nuances of SimCLR, I recommend checking out the following



[Open in app](#)[Get started](#)

In recent years, numerous self-supervised learning methods have been proposed for learning image representations, each...  
amitness.com

## SimCLR Version 2

It's useful to note that since the original publication of the SimCLR framework, the authors have made the following major improvements to the pipeline [2]:

- **Larger ResNet Models** for the Big-CNN Encoder — 152-layer Res-Net, 3x wider channels, selective kernels, attention mechanism.
- **More projection heads** — use three Dense Layers when transforming intermediate representations instead of two.

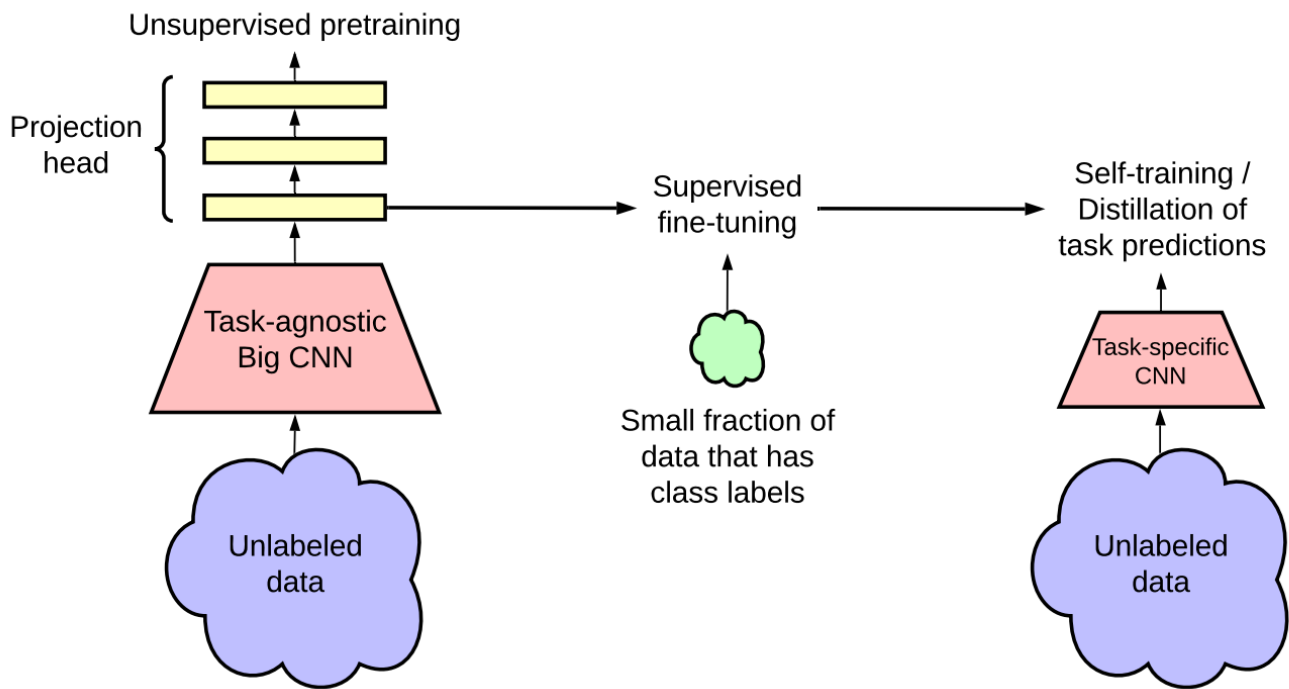
## Applications of Contrastive Learning

### Semi-Supervised Learning

When we *have very few labels*, or if it's hard to obtain labels for a specific task (i.e. clinical annotation), we want to be able to use both the labeled data *and the unlabeled data* to optimize the performance and learning capacity of our model. This is the definition of **semi-supervised learning**.

A methodology that is gaining traction in literature is the *unsupervised pre-train, supervised fine-tune, knowledge distillation* paradigm [2].



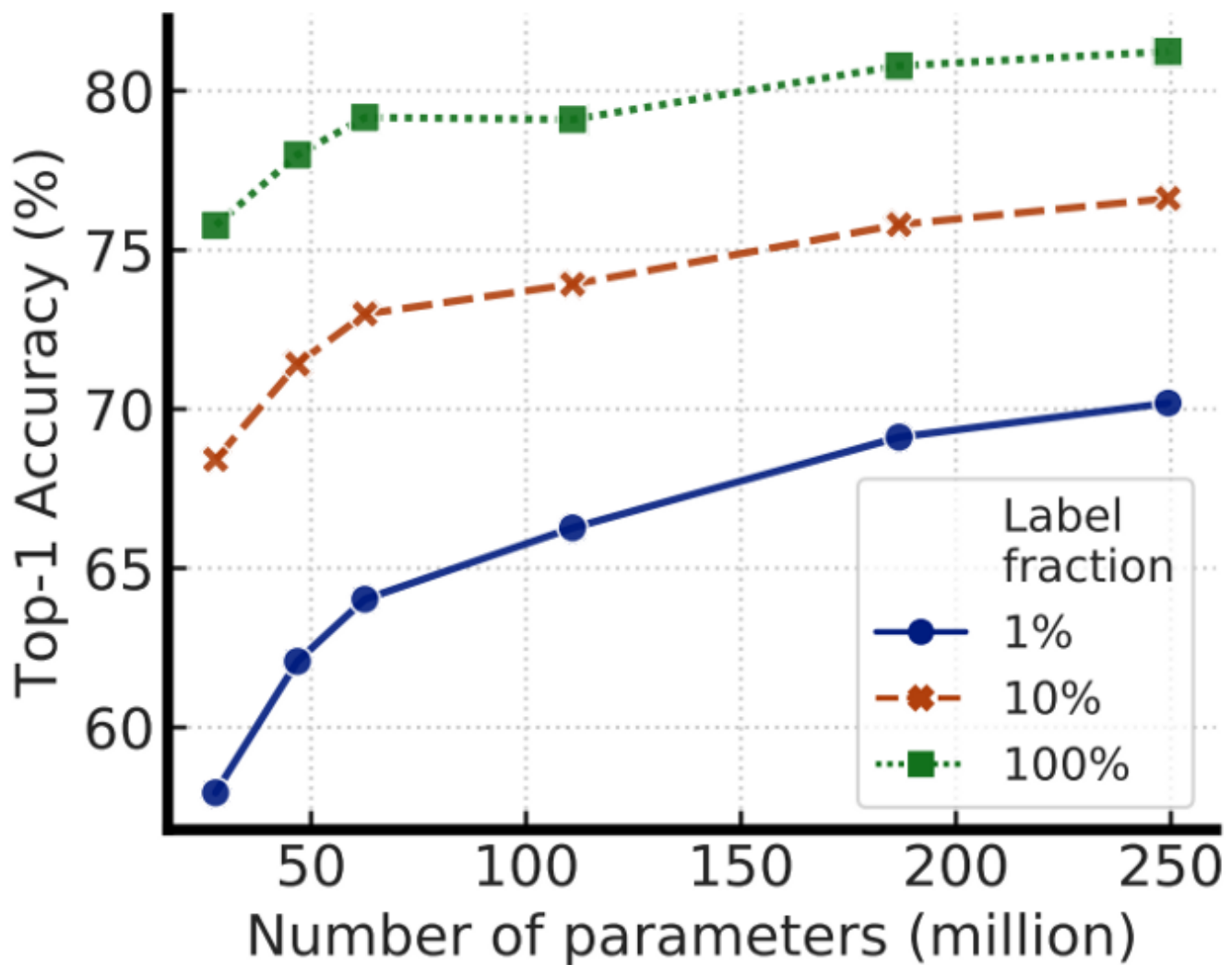
[Open in app](#)[Get started](#)

An overview of the semi-supervised learning paradigm. Source: [2]

In this paradigm, the self-supervised contrastive learning approach is a crucial ‘pre-processing’ step, that allows the Big CNN model (i.e. ResNet-152) to first learn general features from unlabeled data before trying to classify the images using limited labeled data.

The Google Brain team demonstrated that this semi-supervised learning approach is *very label-efficient* and that larger models can lead to greater improvements, especially for low label fractions.



[Open in app](#)[Get started](#)

## (c) Semi-supervised (y-axis zoomed)

Top-1 Accuracy of the model at various label fractions as models get bigger. Source: [2]

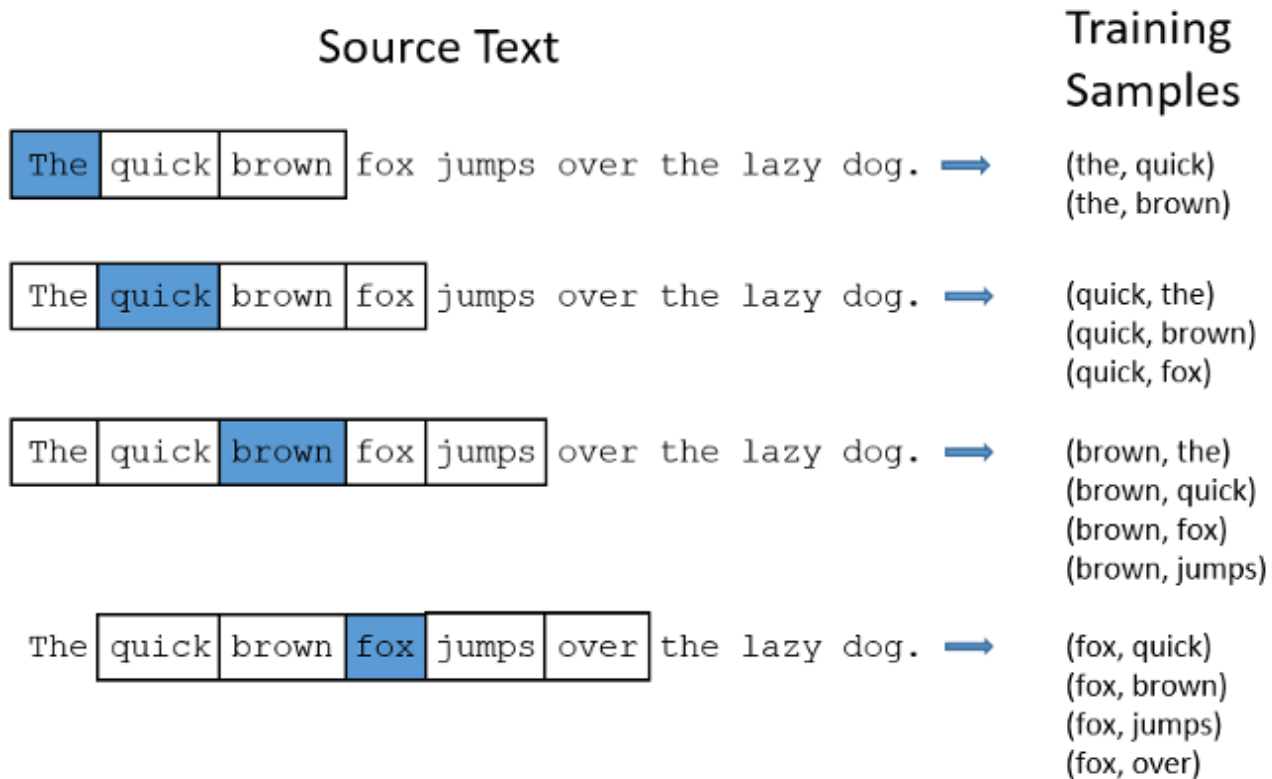
### NLP Analog

It's interesting to note that similar self-supervised methods have already been used extensively in the Natural Language Processing realm.

For instance, **Word2Vec**, an algorithm to convert text to embedded vectors, uses a similar self-supervised approach. In this case, we want the words that are *closer to each other* in a sentence, to have more similar vector representations.





[Open in app](#)[Get started](#)

Source: Figure by Chris McCormick, [Word2Vec Tutorial — The Skip-Gram Model](#)

Thus, we create our ‘positive pairs’ by creating pairs between words within a window. We use a technique called **negative sampling** to create our negative pairs.

This post contains an intuitive and detailed explanation of the Word2Vec algorithm.

### The Illustrated Word2vec

Discussions: Hacker News (347 points, 37 comments), Reddit r/MachineLearning (151 points, 19 comments) Translations...

[jalammar.github.io](http://jalammar.github.io)

Just like SimCLRv2, Word2Vec allows ‘similar’ words to have more similar vector representations in the latent space, and we can use these learned representations for more specific, downstream tasks such as text classification.



[Open in app](#)[Get started](#)

- Contrastive learning is a *self-supervised, task-independent* deep learning technique that allows a model to learn about data, even *without labels*.
- The model *learns general features* about the dataset by learning which types of images are similar, and which ones are different.
- SimCLRv2 is an example of a contrastive learning approach that learns how to represent images such that similar images have similar representations, thereby allowing the model to learn how to distinguish between images.
- The *pre-trained model* with a general understanding of the data can be *fine-tuned* for a specific task such as image classification *when labels are scarce* to significantly improve label efficiency, and potentially surpass supervised methods.

I hope this article provided a clear intuition about what contrastive learning is, how contrastive learning works, and when you can apply contrastive learning for your own projects. Self-supervised learning is truly amazing!

## References

- [1] [Islam et al. Deep Learning-based Early Detection and Grading of Diabetic Retinopathy Using Retinal Fundus Images, 2018](#)
- [2] [Chen et al. Big Self-Supervised Models are Strong Semi-Supervised Learners, 2020](#)

---

**Sign up for The Variable**

By Towards Data Science





Open in app

Get started

Get this newsletter

