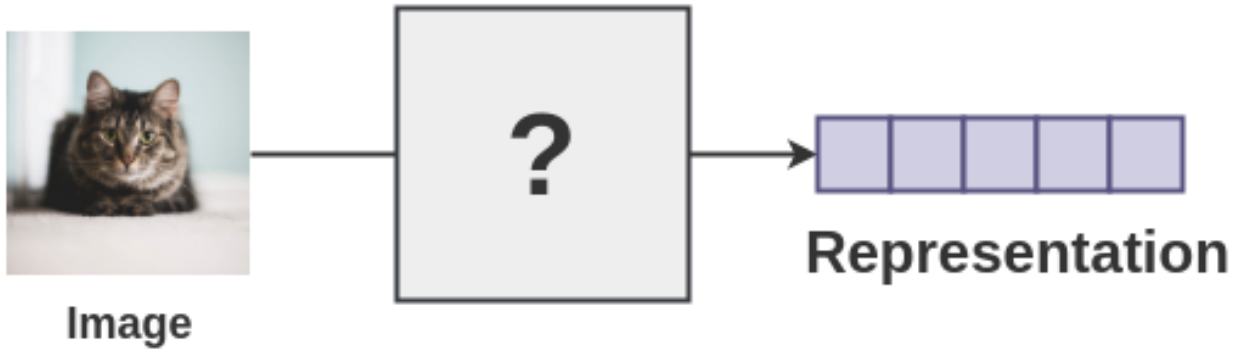
simCLR : A Simple Framework for Contrastive Learning of Visual Representations

Yuval Lavie

Representation Learning

Representation learning replaces the need for manual feature engineering and allows a machine to both learn the features and use them to perform a specific task. It is a field of its own in Machine Learning and often becomes much of the actual effort in deploying machine learning algorithms.

The goal is to find a representation function $f(\cdot)$ such that replacing our input x with $f(x)$ captures meaningful information regarding our task and may reduce the computational and statistical complexity associated with it.



Uses of representations

Transfer learning is the ability of a learning algorithm to exploit commonalities between different learning tasks in order to share statistical strength, and transfer knowledge across tasks. we hypothesize that representation learning algorithms have an advantage for such tasks because they learn representations that capture underlying factors, a subset of which may be relevant for each task, as illustrated in Figure 2. This hypothesis seems confirmed by several empirical results showing the strengths of representation learning algorithms in transfer learning scenarios.

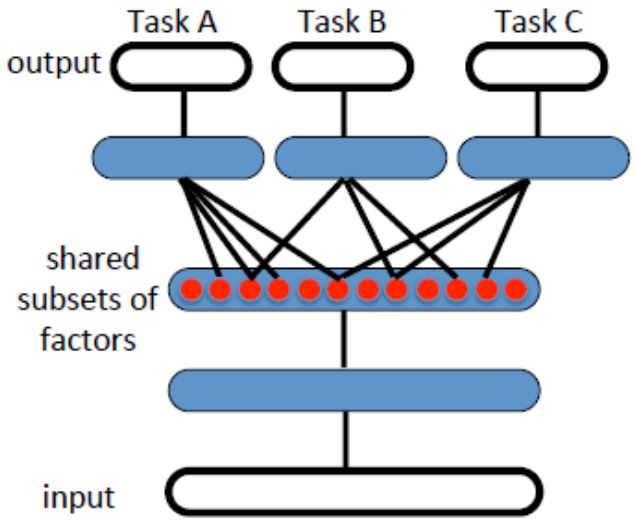


Figure 1. Illustration of representation-learning discovering explanatory factors (middle hidden layer, in red), some explaining the input (semi-supervised settings) And some explaining the target for each task. Because these subsets overlap, sharing of statistical strength helps generalization

Uses of representations

Model Complexity Reduction - For AI-tasks, such as vision and NLP, it seems hopeless to rely only on simple parametric models (such as linear models) because they cannot capture enough of the complexity of interest unless provided with the appropriate feature space. Assuming that pictures of dogs and cats are separable by a hyperplane may be too strict but their learnt representations may be separable.

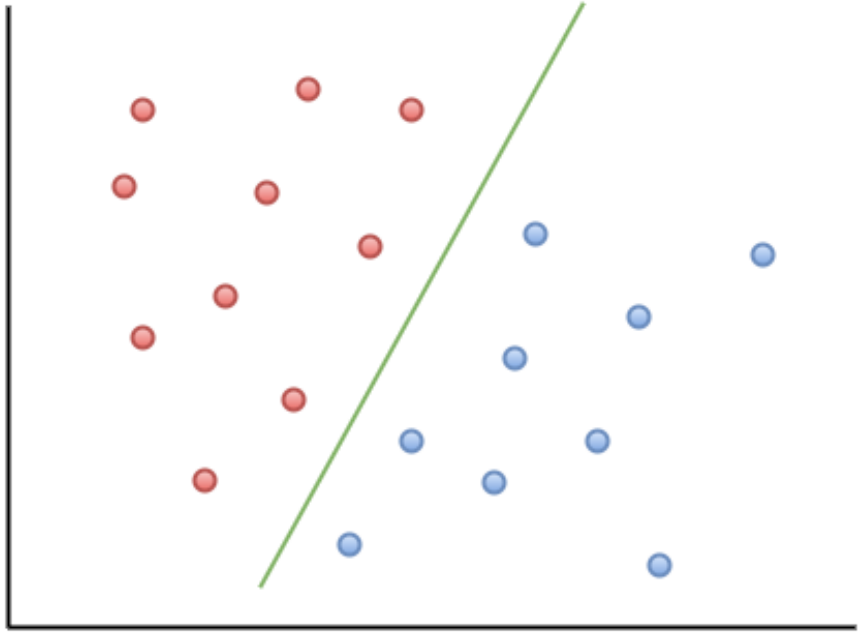


Figure 2. Representations of the data are shown to be linearly separable

Uses of representations

Sample Complexity Reduction - A useful representation which allows us to learn a simpler function may suggest that our task can be learnt with fewer samples. Statistical Learning theory quantifies the requirement as a function of the VC-Dimension. Other analysis methods include Natarajan's Dimension, Rademacher Complexity and PAC Bayesian learning. We would like to make use of this theory to learn representations from a large set of unlabeled data and train our simple models on our small sets of labeled data.

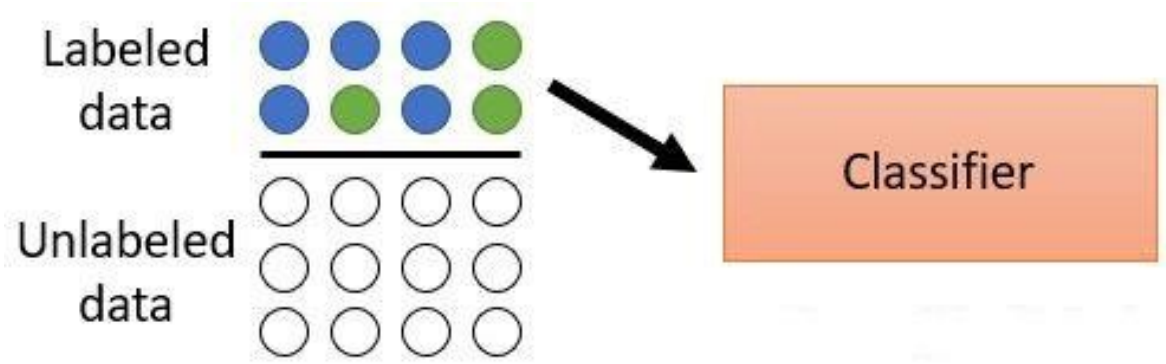


Figure 3. Illustration of our dataset which consists of both labeled And unlabeled data. We learn our representations on the labeled data And then train a classifier on those representations instead of the original labeled data.

Representations in practice

Deep neural networks excel at perceptual tasks when labeled data are abundant, yet their performance degrades substantially when provided with limited supervision.

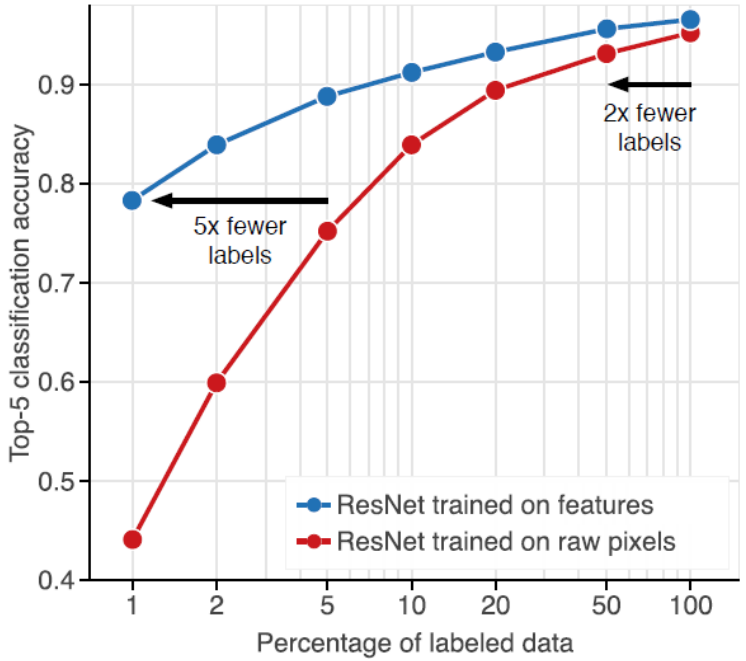
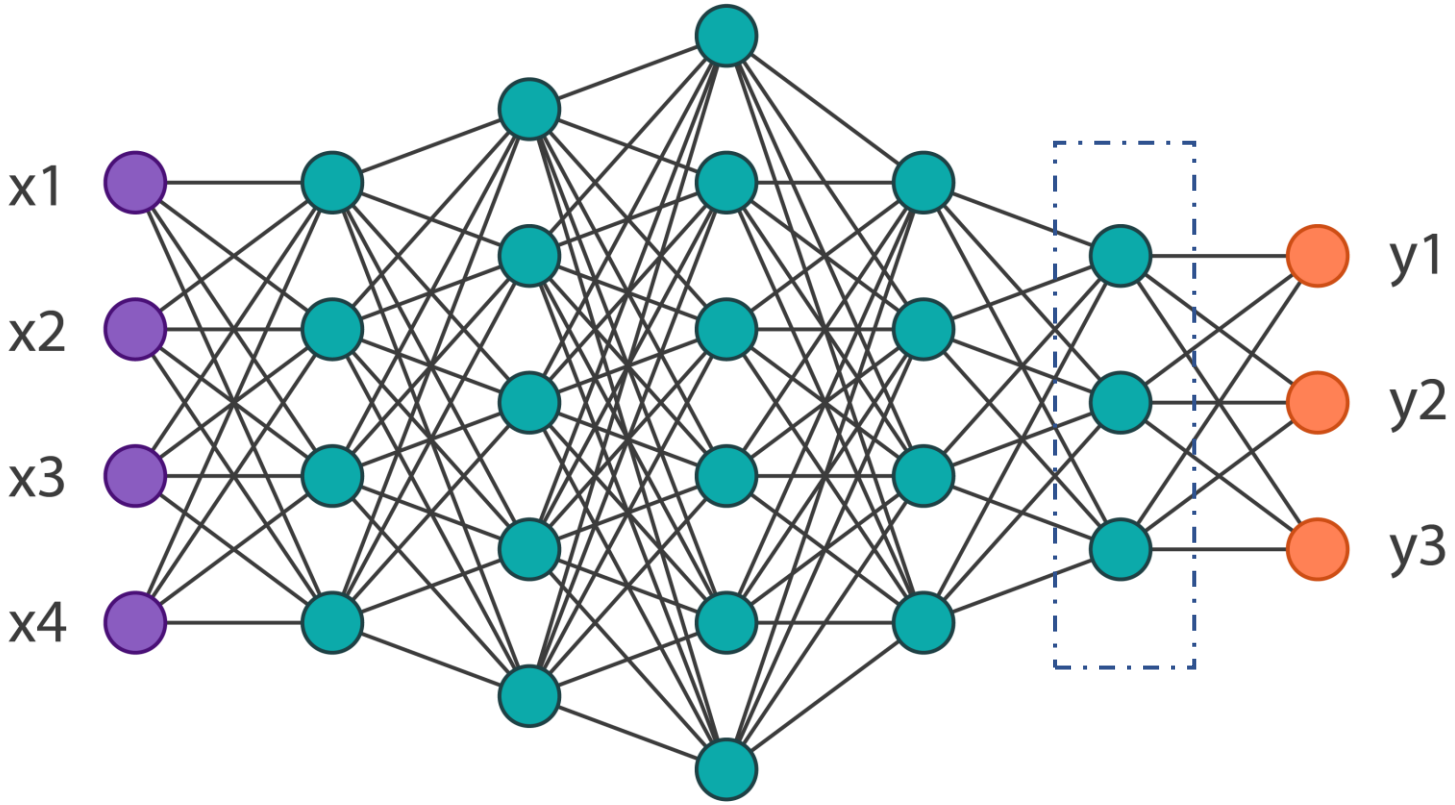


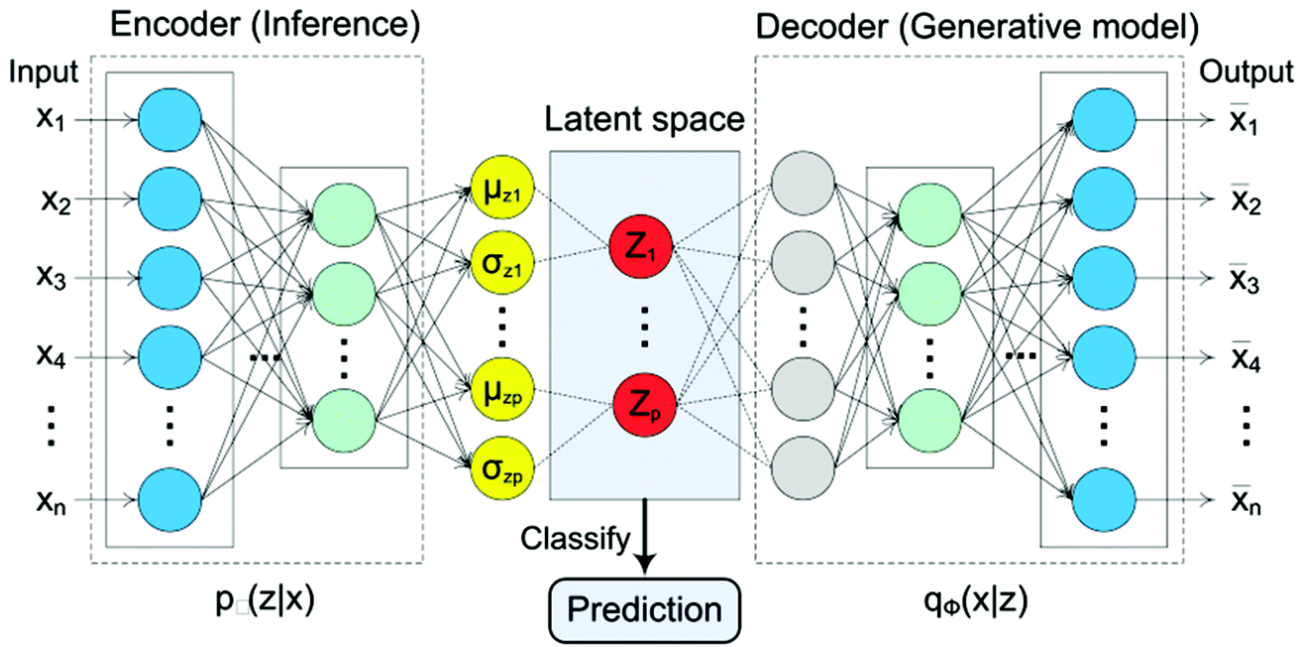
Figure 4. Depicting the change in accuracy of supervised and unsupervised Classifiers when trained on decreasing amounts of labeled data. Unsupervised Classifiers retain high accuracy even when trained on small datasets and can Achieve competitive accuracy with significantly fewer labels (horizontal arrows).

Supervised Representation



For images, a proof of existence for broadly useful representations is the output of the penultimate layer (the one before the softmax) of a network trained on the ImageNet dataset

Unsupervised Representation



An auto-encoder extracts a meaningful low dimensional representation of the data avoiding the curse of dimensionality and reduces the computational complexity of a given task. A good representation may lead us to choosing a simpler decision model thus reducing the amount of labeled data required for a downstream task such as binary classification.

Learning and Evaluating

Let $U = \{x_n\}$ be a large set of unlabeled samples and $S = \{(x_m, y_m)\}$ be a small set of labeled samples then our learning method consists of two optimization problems.

- We learn our unsupervised representation with an encoder denoted $f(\cdot)$ by solving

$$\theta^* = argmin_\theta \frac{1}{N} \sum_{n=1}^{N} \ell_{un}\left(f_\theta(x_n)\right)$$

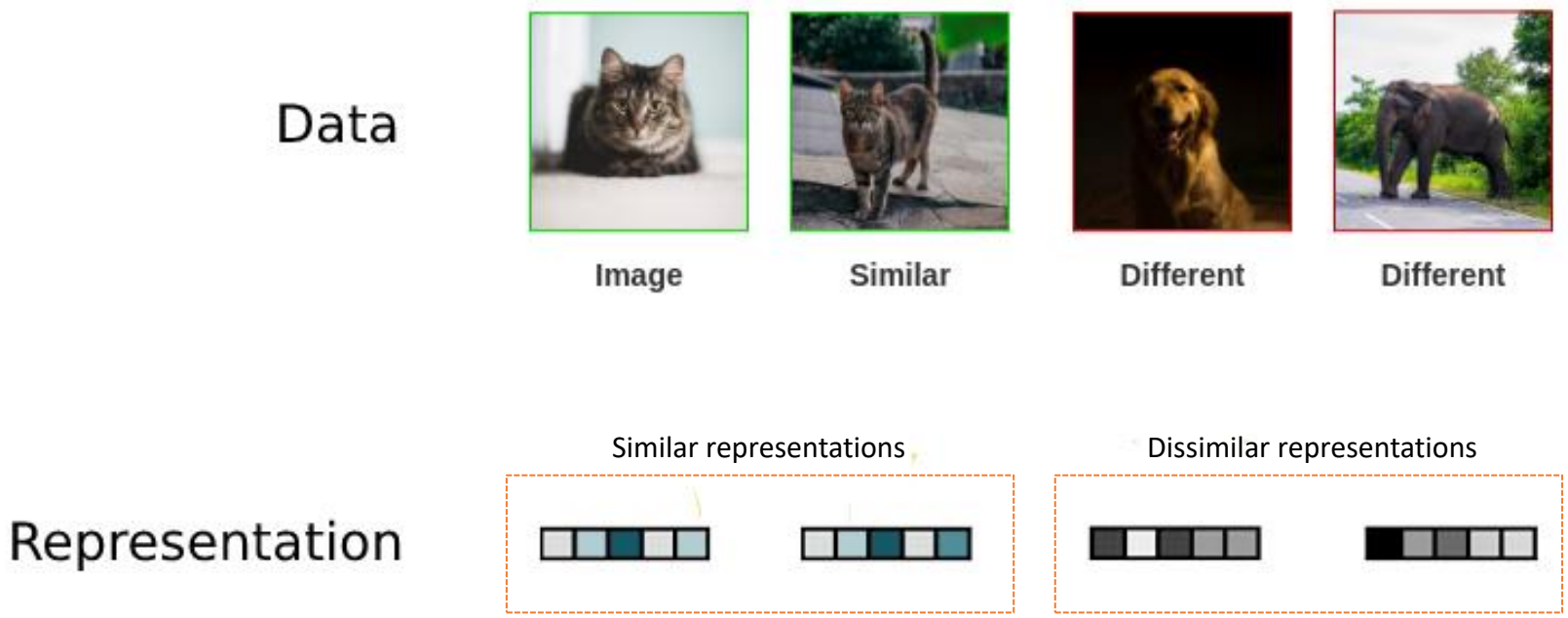
- We learn our classifier on the representations by solving

$$\psi^* = argmin_\psi \frac{1}{N} \sum_{n=1}^{N} \ell_{Sup}\left(h_\psi\left(f_{\theta^*}(x_m)\right), y_m\right)$$

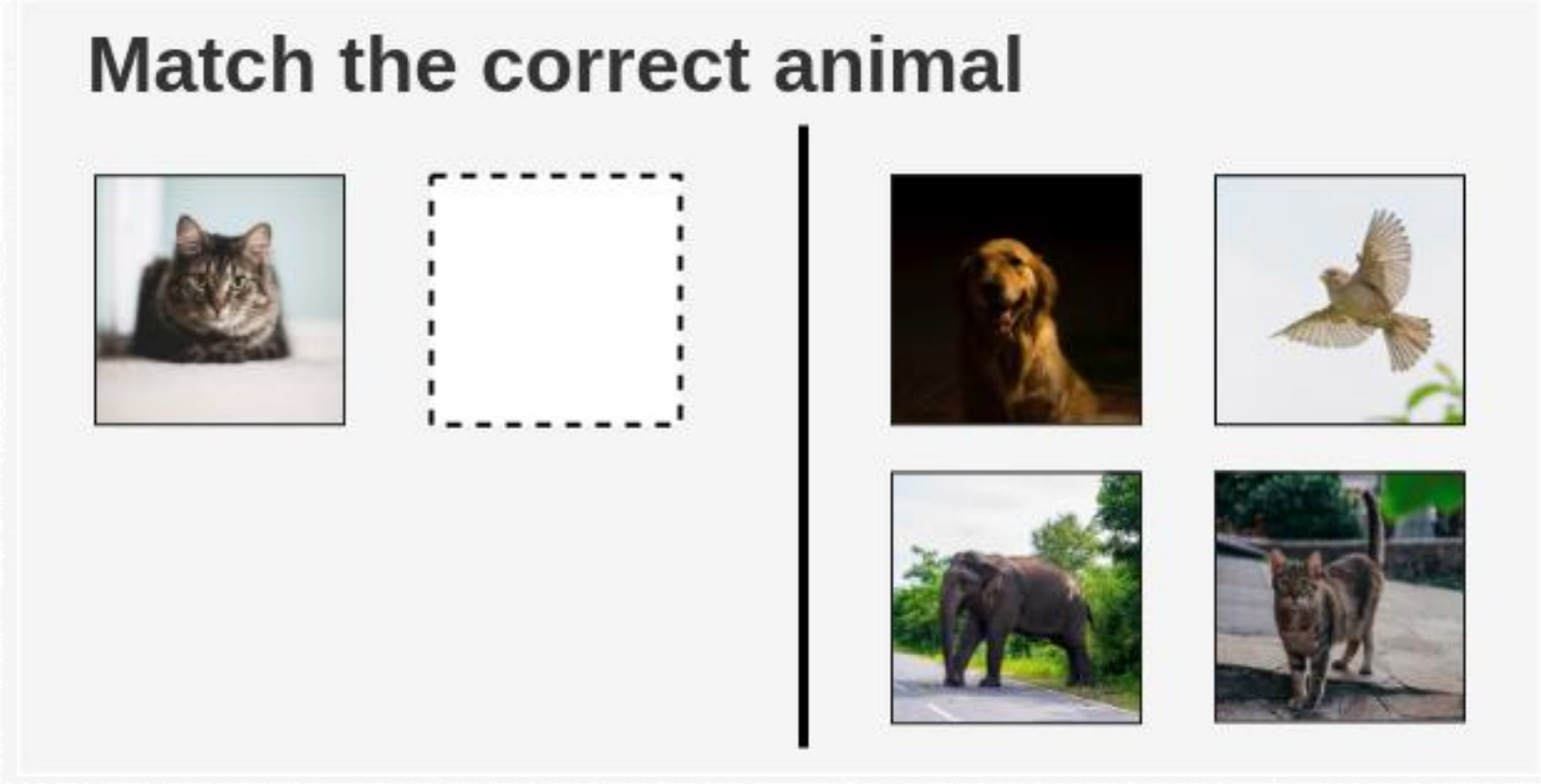
Where H is the class of simple functions used to evaluate the goodness of the representations.

CURL

Contrastive unsupervised representation Learning is a method of learning representations which respect similarity: leveraging availability of “similar” and “dis-similar” data points. The learner then forces the similarity score of representations of similar samples with each other to be higher on average than with dissimilar samples by minimizing a contrastive loss.



Intuition of Contrastive Learning



The way a child would solve it is by looking at the cat at the left side and search for a cat on the right side. Thus learning to contrast the cat from other animals and develop features for recognizing cats by similarity.

Framework for Contrastive Learning

Contrastive learning assumes access to a very specific structure of data, a similarity function and a loss function which reflects our requirement of respecting the similarity function.

1. Data Structure : $\mathrm{X} = (x, x^+, \{x_i^-\}_{i=1}^k)$ where x^+ is similar to x and every x_i^- is dissimilar to x
2. Similarity function : $sim(\cdot,\cdot)$
3. Contrastive loss which is a function of the similarity : $\ell\big(X, sim(\cdot,\cdot)\big)$

Why should it work ?

While it seems **intuitive** that minimizing such loss functions should lead to representations that capture ‘similarity,’ formally it is unclear why the learned representations should do well on downstream linear classification tasks – their somewhat mysterious success is often treated as an obvious consequence. To analyze this success, a framework must connect ‘similarity’ in unlabeled data with the information that is implicitly present in downstream tasks such as classification.

Recent literature has attempted to relate the success of their methods to maximization of mutual information between latent representations. However, it is not clear if the success of contrastive approaches is determined by mutual information or by the specific form of the contrastive loss.

Simple Analysis

We restrict ourselves to a simplified scenario where the supervised task is binary and the negative set contains one element. Thus, we choose the label set to be $Y = \{-1, 1\}$, and the unsupervised set U contains triplets

$$z_i = (x_i, x_i^+, x_i^-)$$

Let C be a set of latent classes and ρ be a probability distribution over C. Each class c comes with a distribution D_c and (x, x^+) are assumed to be drawn from the same class distribution. Note that an input x possibly belongs to multiple classes: take the example of x being an image and C a set of latent classes including "the image depicts a dog" and "the image depicts a cat" (both classes are not mutually exclusive).

The formation of our unlabeled dataset (U) is as following:

1. Draw two latent classes $(c^+, c^-) \sim \rho^2$
2. Draw two similar samples from a single class $(x, x^+) \sim D_{c^+}^2$
3. Draw negative samples from the other class $x^- \sim D_{c^-}$

Analysis continued

The labeled sample S is obtained by fixing two classes $c^{\pm} = \{c^{-}, c^{+}\} \in C^2$ Each class is then mapped on a label of Y . We fix $y_{c^-} = -1$ and $y_{c^+} = 1$ for binary classification. The label is obtained from the latent class distribution restricted to two values $\rho_{c^{\pm}}$:

$$\rho_{c^{\pm}}(c^{-}) = \frac{\rho(c^{-})}{\rho(c^{-})+\rho(c^{+})}, \rho_{c^{\pm}}(c^{+}) = \frac{\rho(c^{+})}{\rho(c^{-})+\rho(c^{+})},$$

The formation of our labeled dataset (S) is as following:

1. Draw a class $c \sim \rho_{c^{\pm}}$ and set the label $y = y_c$
2. Draw a sample $x \in D_c$
3. Set the example (x, y_c)

For more information about the analysis of contrastive learning:

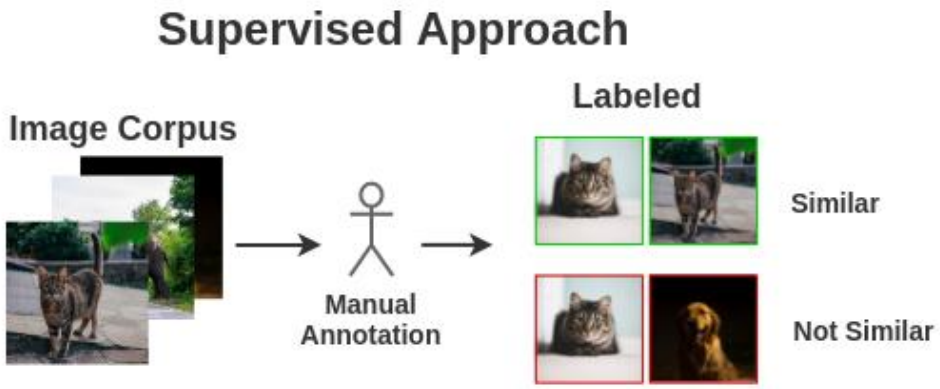
Analysis of Contrastive Unsupervised Representation Learning : https://arxiv.org/pdf/1902.09229.pdf

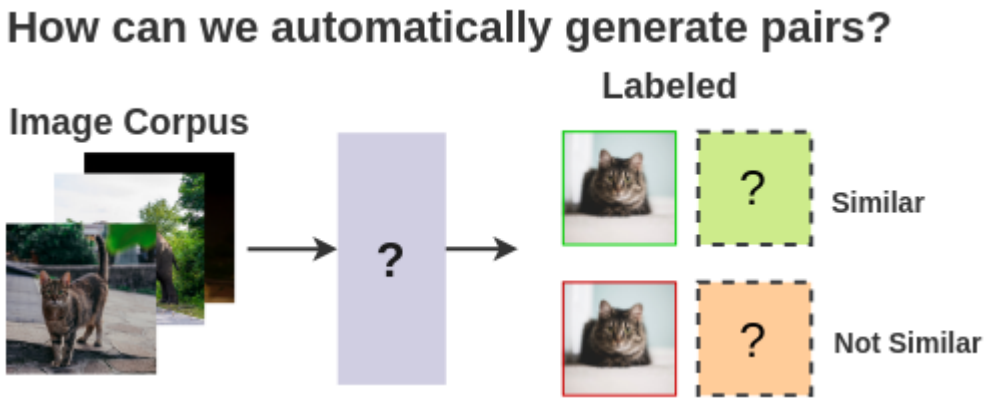
PAC-Bayesian Contrastive Unsupervised Representation Learning : https://arxiv.org/pdf/1910.04464.pdf

Assumptions and Intuitions aside..

Gathering unlabeled data is certainly easier than gathering labeled data, but how do we get access to data with the specific structure needed for contrastive learning ? If we ask humans to manually find similarities between samples we may as well just ask them to label the data and use supervised methods for classification.
Instead we would like to be able to automatically find the structure or to generate this structure ourselves.

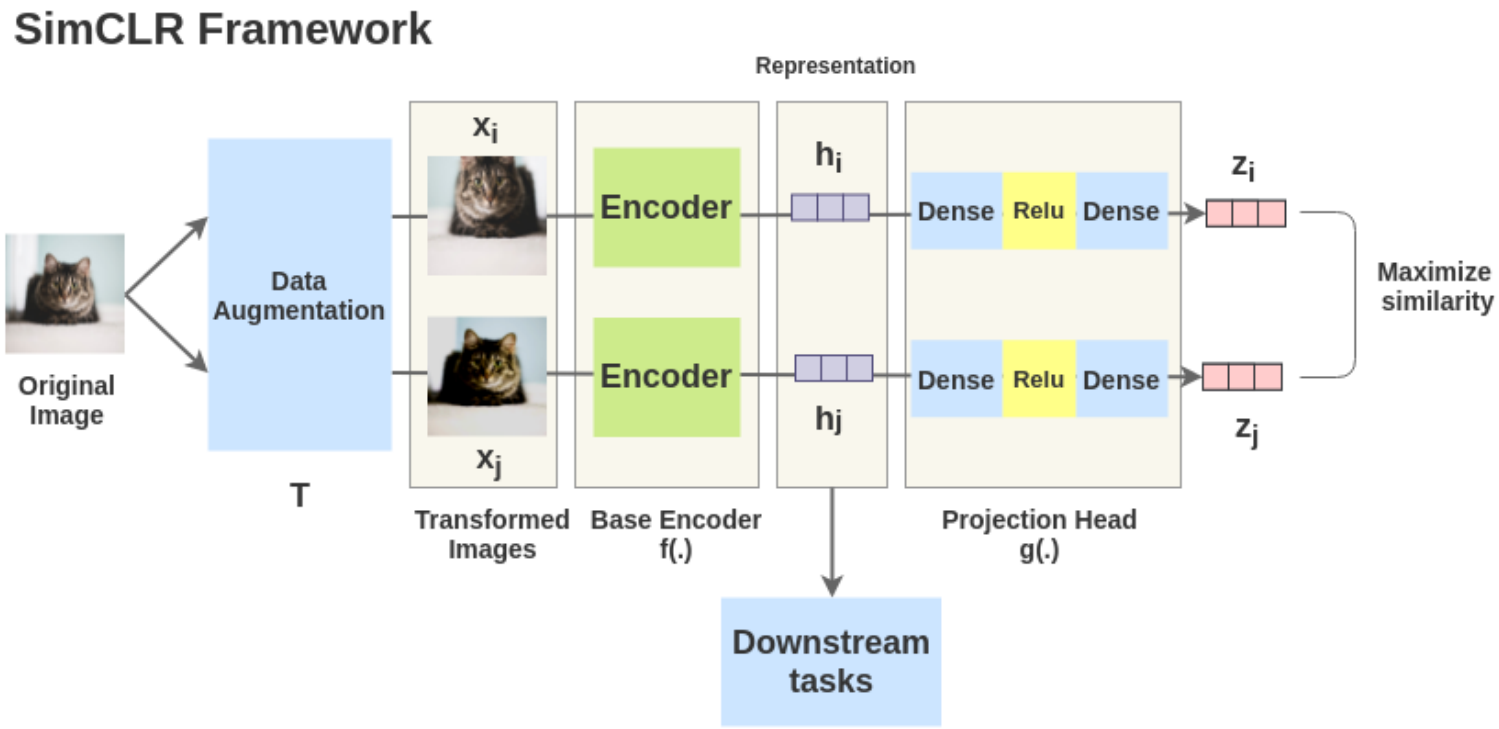
If we decide to automatically find the structure, then we need to find a representation and similarity function which the computer can understand.





SimCLR : Learning visual representations

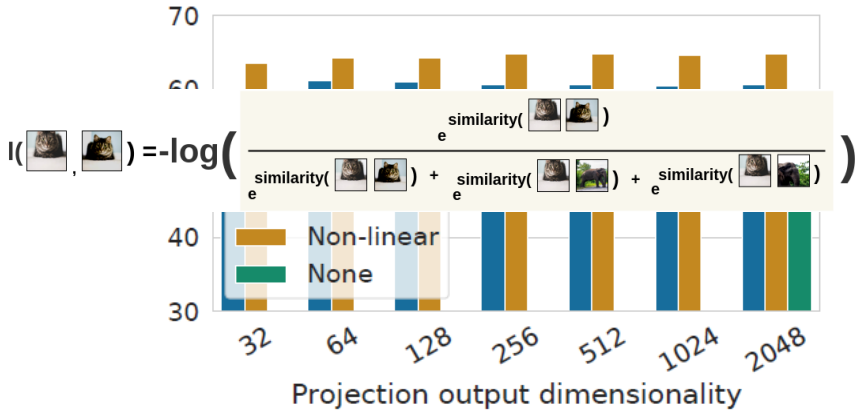
SimCLR is a simple framework for contrastive learning of visual representations which not only outperforms previous work on the subject but is also much simpler and requires no specialized architectures nor a memory bank. SimCLR learns representations by maximizing agreement between differently augmented views of the same image via a contrastive loss in the latent space.



SimCLR : Components

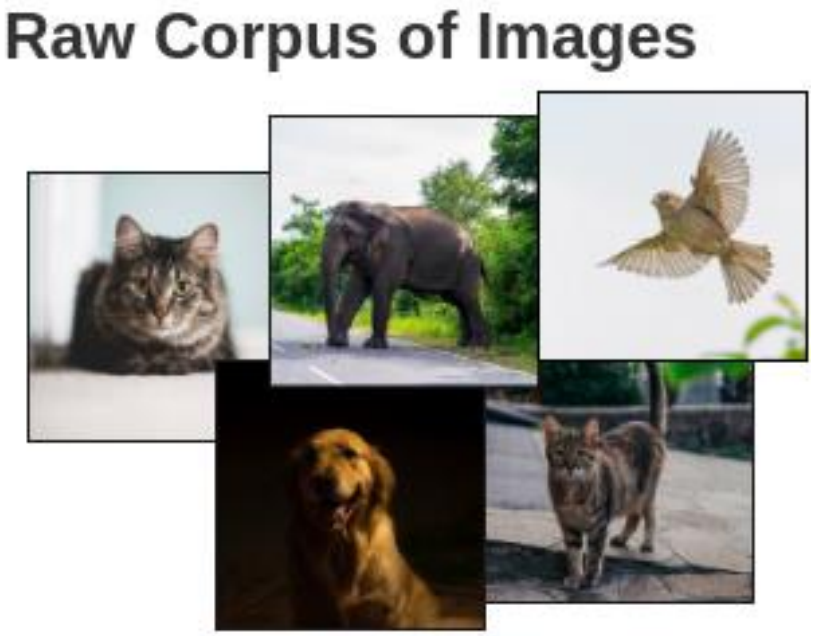
The four major components of the framework are:

1. Data augmentation module
    - Creates the similar objects needed by the framework.
2. An encoding head $f(\cdot)$
    - Creates the learnt representation which will eventually be used.
3. A projector head $g(\cdot)$
    - Projects the representation unto a space where the loss can be calculated.
4. A contrastive loss
    - Which respects and understands the similarity constraint.



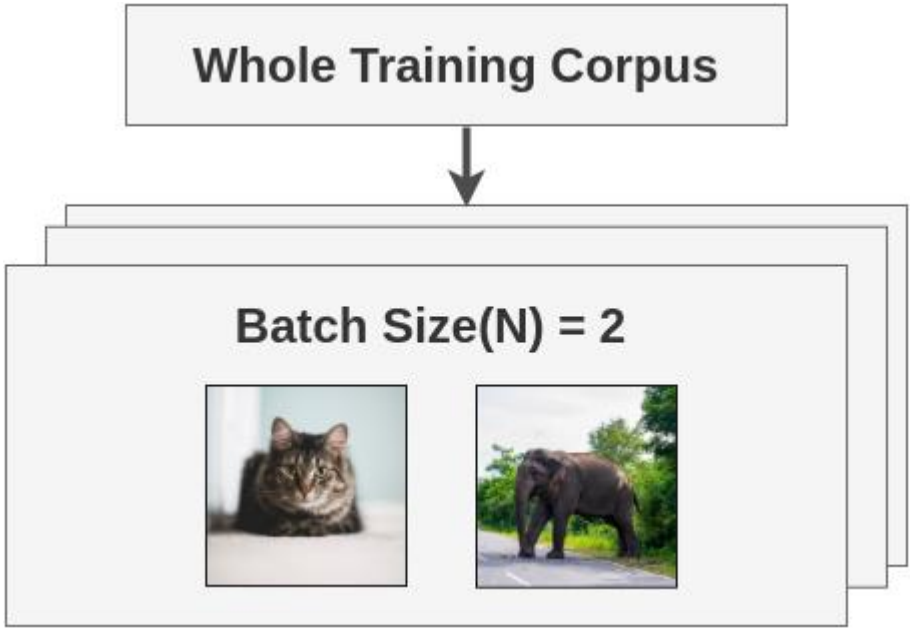
Step by step

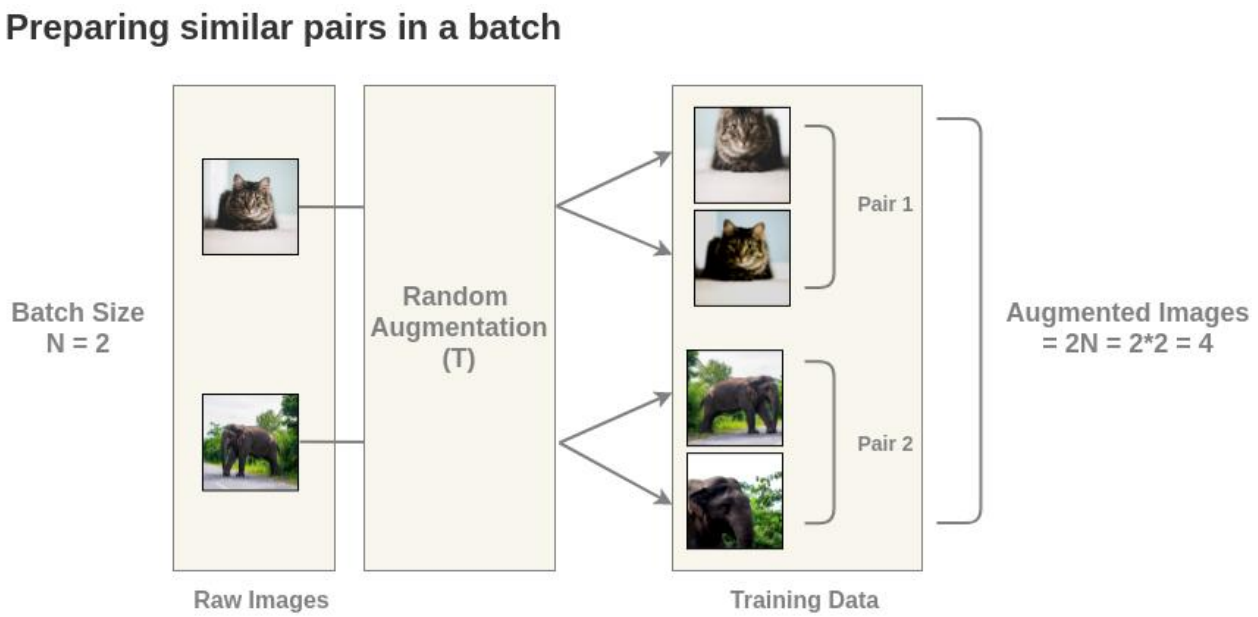
We have a large data set of unlabeled images and would like to learn a contrastive representation



Step by step

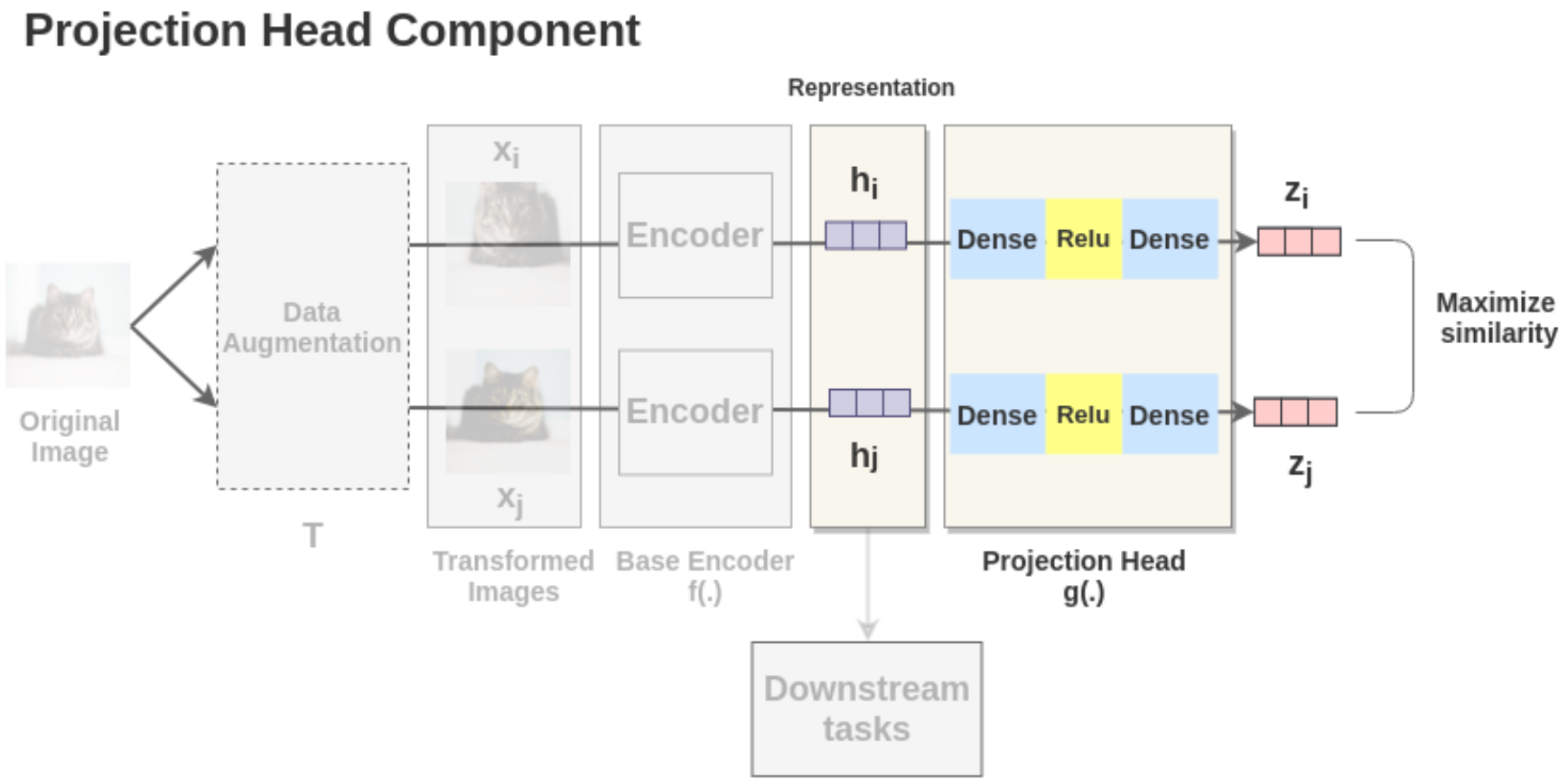
We generate a batch of N images and apply our transformations on each image to receive 2N augmented images





Step by step

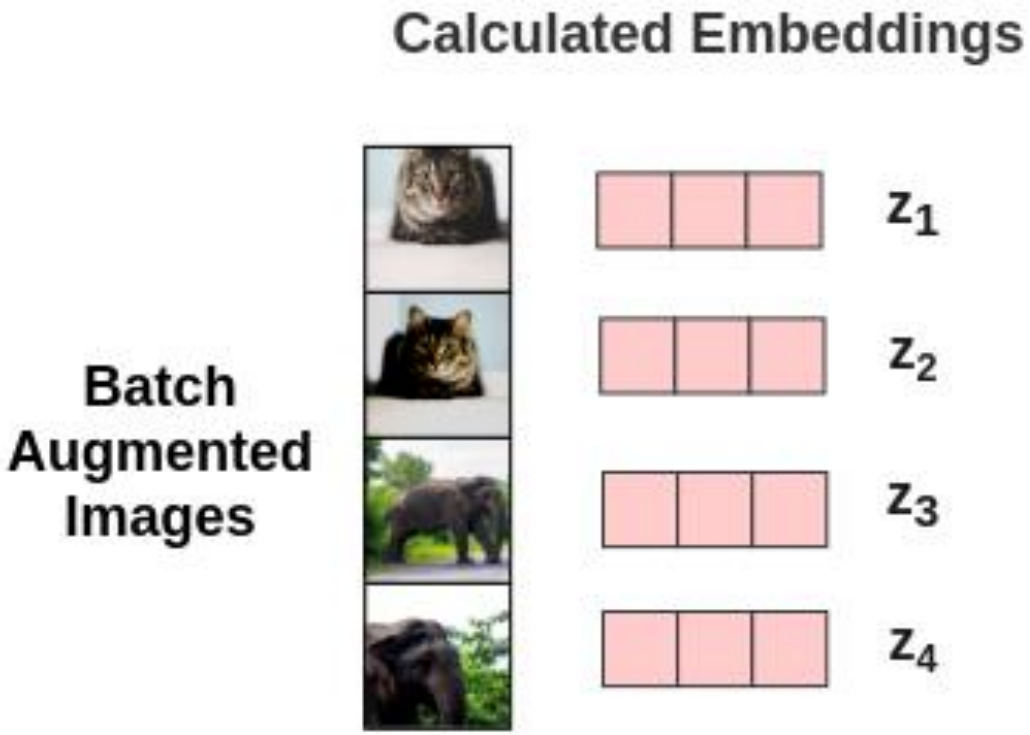
We pass the augmented images through the encoder $f(\cdot)$ and the projection head $g(f(\cdot))$



Step by step

Referring to our structured framework we now have the following data structure

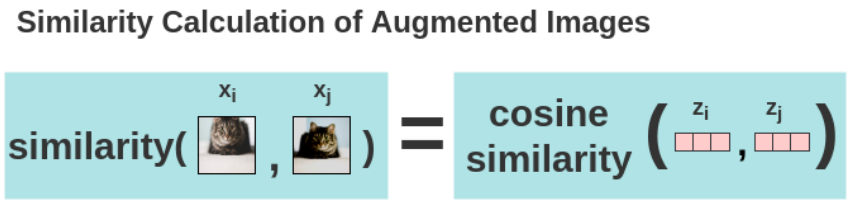
$Z = (z, z^+, \{z_i^-\}_{i=1}^k)$



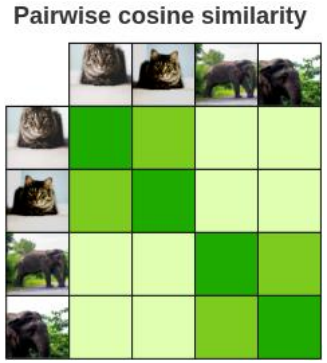
Step by step

We define our similarity as the Cosine Similarity

$$sim(z_i, z_j) = \frac{\langle z_i, z_j \rangle}{\|z_i\| \cdot \|z_j\|}$$



And hope that in the ideal case the similarities will follow our logic



Simulation

NT-Xent: Normalized Temperature-Scaled Cross Entropy

We choose our instantaneous loss functions to be the NT-Xent loss because of its easy interpretability where minimizing the loss is essentially maximizing the similarity between $x = z_i, x^+ = z_j$ and minimizing the similarity between $x = z_i, x^- = \{z_k\}_{k \neq i,j}$

$$\ell(z_i, z_j) = -log \frac{\exp\left(\frac{sim(z_i, z_j)}{\tau}\right)}{\sum_{k=1}^{2N} I(k \neq i) \cdot \exp\left(\frac{sim(z_i, z_k)}{\tau}\right)}$$

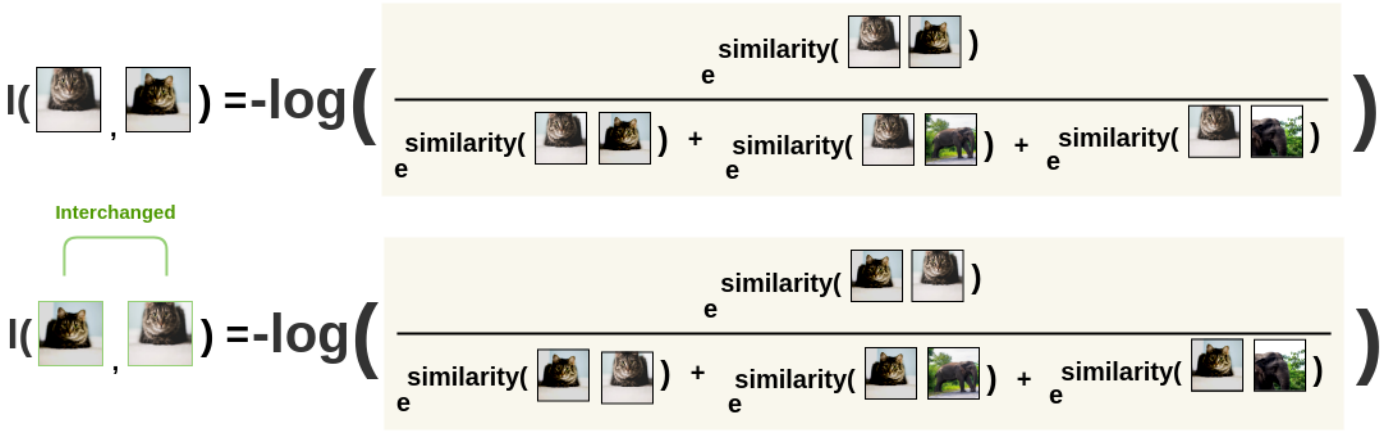
$$\Rightarrow$$

$$\downarrow \ell(z_i, z_j) = \log\left[1 + \frac{\sum_{k=1}^{2N} I(k \neq i, j) \cdot \exp\left(\frac{\downarrow sim(z_i, z_k)}{\tau}\right)}{\exp\left(\frac{\uparrow sim(z_i, z_j)}{\tau}\right)}\right]$$

Training the network

We define our Contrastive Loss function as

$$\mathrm{L} = \frac{1}{2N}\sum_{k=1}^{N}[\ell(2k-1,2k) + \ell(2k,2k-1)]$$



Results - Paper

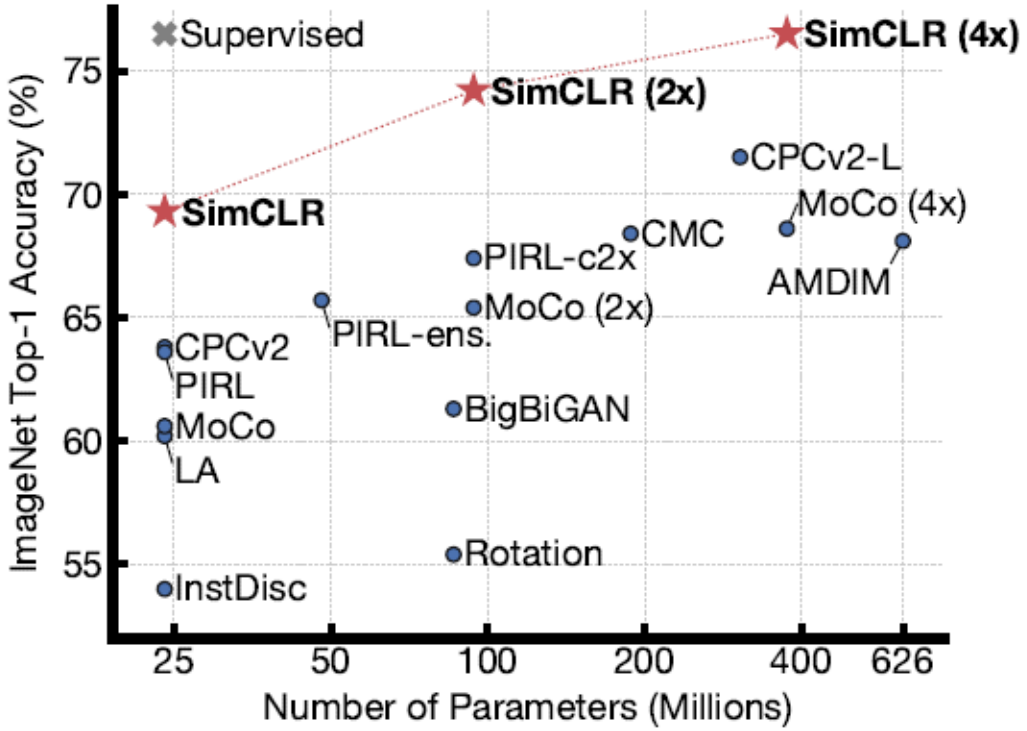


Figure 5. ImageNet Top-1 accuracy of linear classifiers trained on representations learned with different self-supervised methods (pretrained on ImageNet). Gray cross indicates supervised ResNet-50. Our method, SimCLR, is shown in bold.

Results - Paper

A Simple Framework for Contrastive Learning of Visual Representations

	Food	CIFAR10	CIFAR100	Birdsnap	SUN397	Cars	Aircraft	VOC2007	DTD	Pets	Caltech-101	Flowers
Linear evaluation:												
SimCLR (ours)	**76.9**	**95.3**	80.2	48.4	**65.9**	60.0	61.2	**84.2**	**78.9**	89.2	**93.9**	**95.0**
Supervised	75.2	**95.7**	**81.2**	**56.4**	64.9	**68.8**	**63.8**	83.8	**78.7**	**92.3**	**94.1**	94.2
Fine-tuned:												
SimCLR (ours)	**89.4**	**98.6**	**89.0**	**78.2**	**68.1**	**92.1**	**87.0**	**86.6**	**77.8**	92.1	**94.1**	97.6
Supervised	88.7	98.3	**88.7**	**77.8**	67.0	91.4	**88.0**	86.5	**78.8**	**93.2**	**94.2**	**98.0**
Random init	88.3	96.0	81.9	**77.0**	53.7	91.3	84.8	69.4	64.1	82.7	72.5	92.5

Table 8. Comparison of transfer learning performance of our self-supervised approach with supervised baselines across 12 natural image classification datasets, for ResNet-50 $(4\times)$ models pretrained on ImageNet. Results not significantly worse than the best ($p > 0.05$, permutation test) are shown in bold. See Appendix B.8 for experimental details and results with standard ResNet-50.

Real results

A simplified implementation that uses googles pre-trained weights and a fine-tuned multiclass logistic classifier on the STL-10 data set
Shows that simCLR provides competitive results with the supervised counterpart using the ResNet-50 (1x) encoding head.

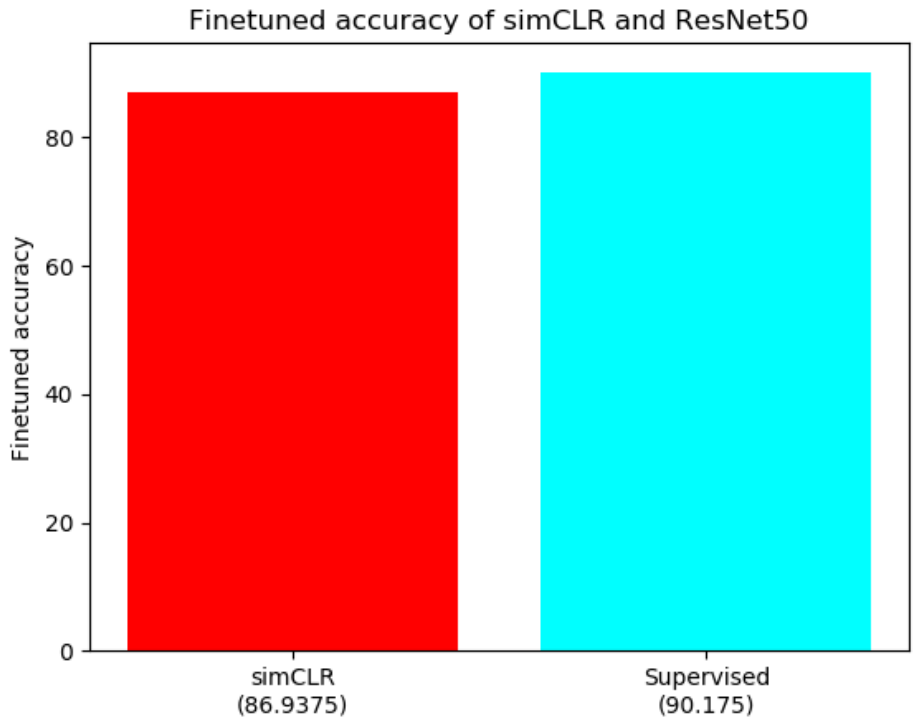


Figure 6. Top-1 accuracy of linear classifiers fine-tuned on top of representations learnt by simCLR and by a supervised method. Code for result recreation can be found at : https://github.com/yuvallavie/simCLR

Real results

We hypothesized that unsupervised methods should suffer less from the lack
Of unlabeled data to fine tune on.. But what happens in reality ?

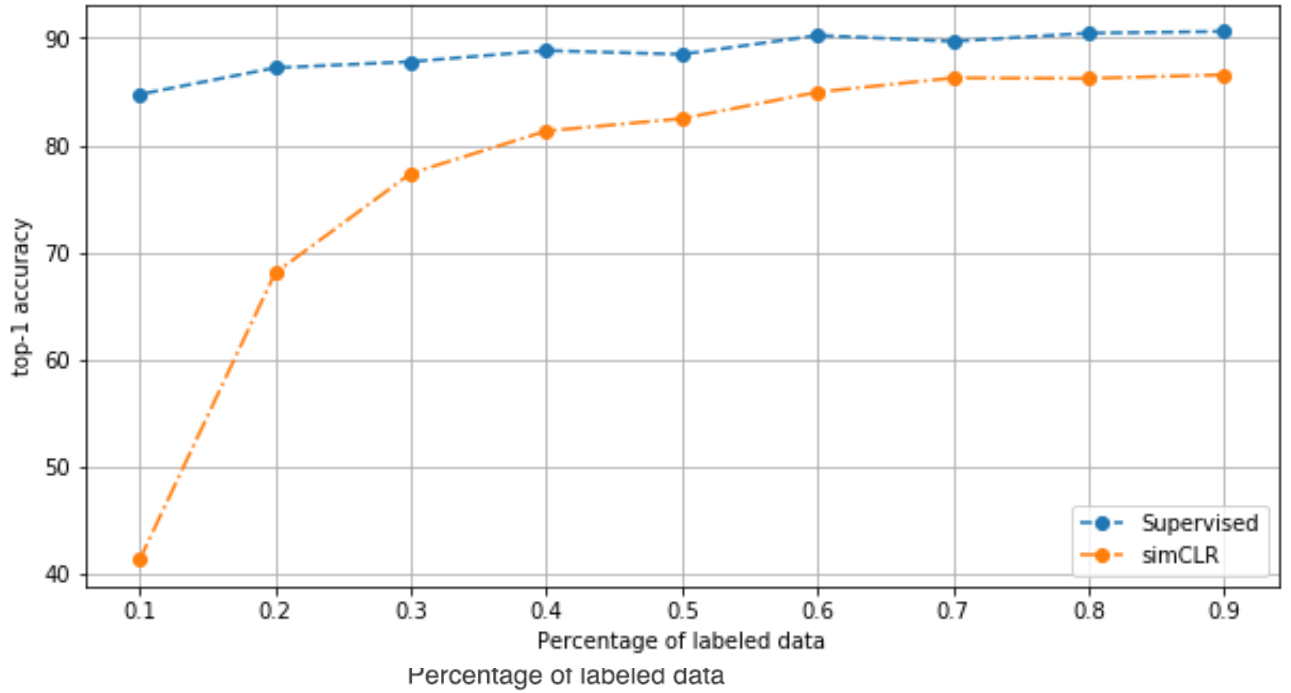


Figure 7. Depicting the change in accuracy of supervised and unsupervised
Classifiers when trained on different percentages of labeled data. Unsupervised
Classifiers like simCLR can leverage unlabeled data to achieve competitive
Accuracy with significantly fewer labels (horizontal arrows).

Conclusions

1. Supervised methods are still much better at creating meaningful representations and I disagree with many papers that try to compete with supervised methods. The main issue is the lack of labeled data.
2. The success of this framework rekindles interest in self-supervised and unsupervised pretraining methods as seen by the many new articles released on this subject recently.
3. simCLR's results show that the complexity of previous methods for self-supervised learning is not necessary and generally opens room for thought about how complex a model should be.
4. Theoretical understanding of contrastive learning is lacking, results are treated to an obvious consequence. New theoretical analysis based on PAC-Bayes has emerged recently and has already allowed the creation of new contrastive algorithms based on its bounds.
5. Even though its simple to implement, the framework requires extensive computing power. It benefits from larger batch sizes and longer training.
6. It opens room for thought about more methods which mimic the human brain.

What’s next ?

SimCLRv2 is an improvement on SimCLR which explores different encoding heads, projection heads and different training techniques. A major change is the way unlabeled data is used twice. Once to generate task-agnostic representations and once again to distill the representations towards a specific task.

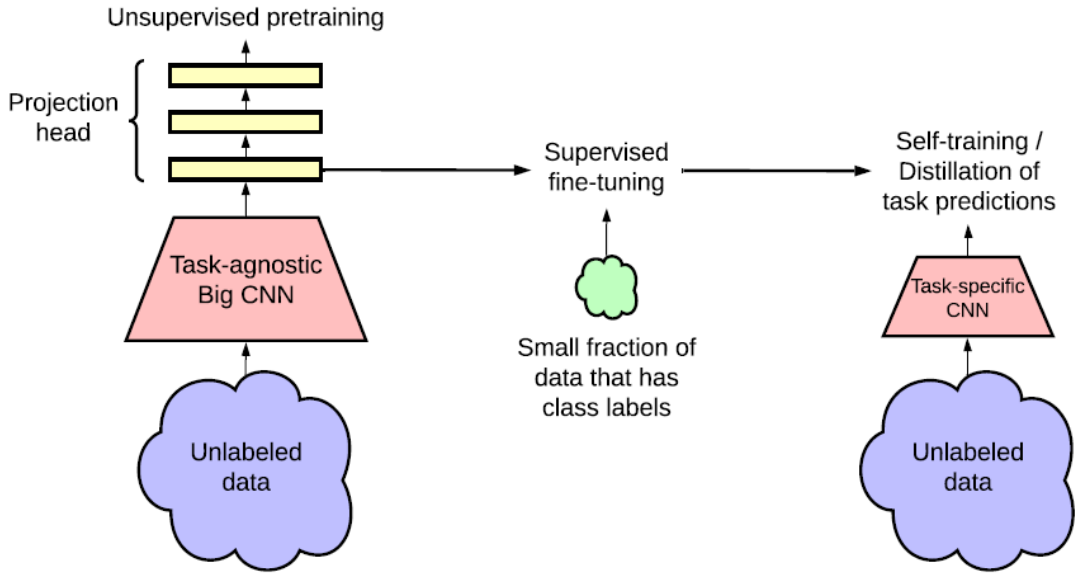


Figure 7. Depicts the process of creating preliminary task-agnostic Representations and then finetuning them towards task-specific Representations with the SimCLRv2 method.

References

- A Simple Framework for Contrastive Learning of Visual Representations (Original paper)

  https://arxiv.org/pdf/2002.05709.pdf
- Big Self-Supervised Models are Strong Semi-Supervised Learners (simCLR Version 2)

  https://arxiv.org/pdf/2002.05709.pdf
- Representation Learning: A Review and New Perspectives

  https://arxiv.org/pdf/1206.5538.pdf
- Google's GitHub for both versions of the framework (TensorFlow)

  https://github.com/google-research/simclr
- A PyTorch implementation by sthalles

  https://github.com/sthalles/SimCLR