Self-Supervised Learning

Yang You

Presidential Young Professor at NUS Computer Science

ai.comp.nus.edu.sg

Outline

- **Introduction to Self-Supervised Learning (SSL)**
- SimCLR: state-of-the-art SSL framework
- Step-by-step example of SimCLR
- Results of SimCLR

Self-Supervised Learning maybe the Future of AI

COMMUNICATIONS OF THE ACM

Search

HOME | CURRENT ISSUE | NEWS | BLOGS | OPINION | RESEARCH | PRACTICE | CAREERS | ARCHIVE | VIDEOS

Home / News / Yann LeCun, Yoshua Bengio: Self-Supervised Learning... / Full Text

ACM TECHNEWS

Yann LeCun, Yoshua Bengio: Self-Supervised Learning is Key to Human-Level Intelligence

By VentureBeat
May 6, 2020
Comments

VIEW AS: SHARE:

Yann LeCun and Yoshua Bengio, recipients of the ACM A.M. Turing Award, say self-supervised learning could lead to the creation of artificial intelligence programs that are more humanlike in their reasoning.

Credit: Medium.com

From *VentureBeat*
View Full Article

ACM A.M. Turing Award recipients Yann LeCun and Yoshua Bengio say that self-supervised learning could lead to the creation of artificial intelligence (AI) programs that are more humanlike in their reasoning.

Speaking at the International Conference on Learning Representation (ICLR) 2020, which took place online, LeCun, Facebook's chief AI scientist, said supervised learning systems will play a diminishing role as self-supervised learning algorithms—those that generate labels from data by exposing relationships between the data's parts, believed to be critical to achieving human-level intelligence—comes into wider use.

Meanwhile Bengio, director at the Montreal Institute for Learning Algorithms, predicts new studies will reveal the way high-level semantic variables connect with how the brain processes information, including visual information. Humans communicate these kinds of variables using language, and they could lead to a new generation of deep learning models.

SIGN IN for Full Access

User Name

Password

» Forgot Password?
» Create an ACM Web Account

SIGN IN

MORE NEWS & OPINIONS

Google Hit by Antitrust Lawsuit from nearly 40 States over Alleged Search Monopoly
CNet

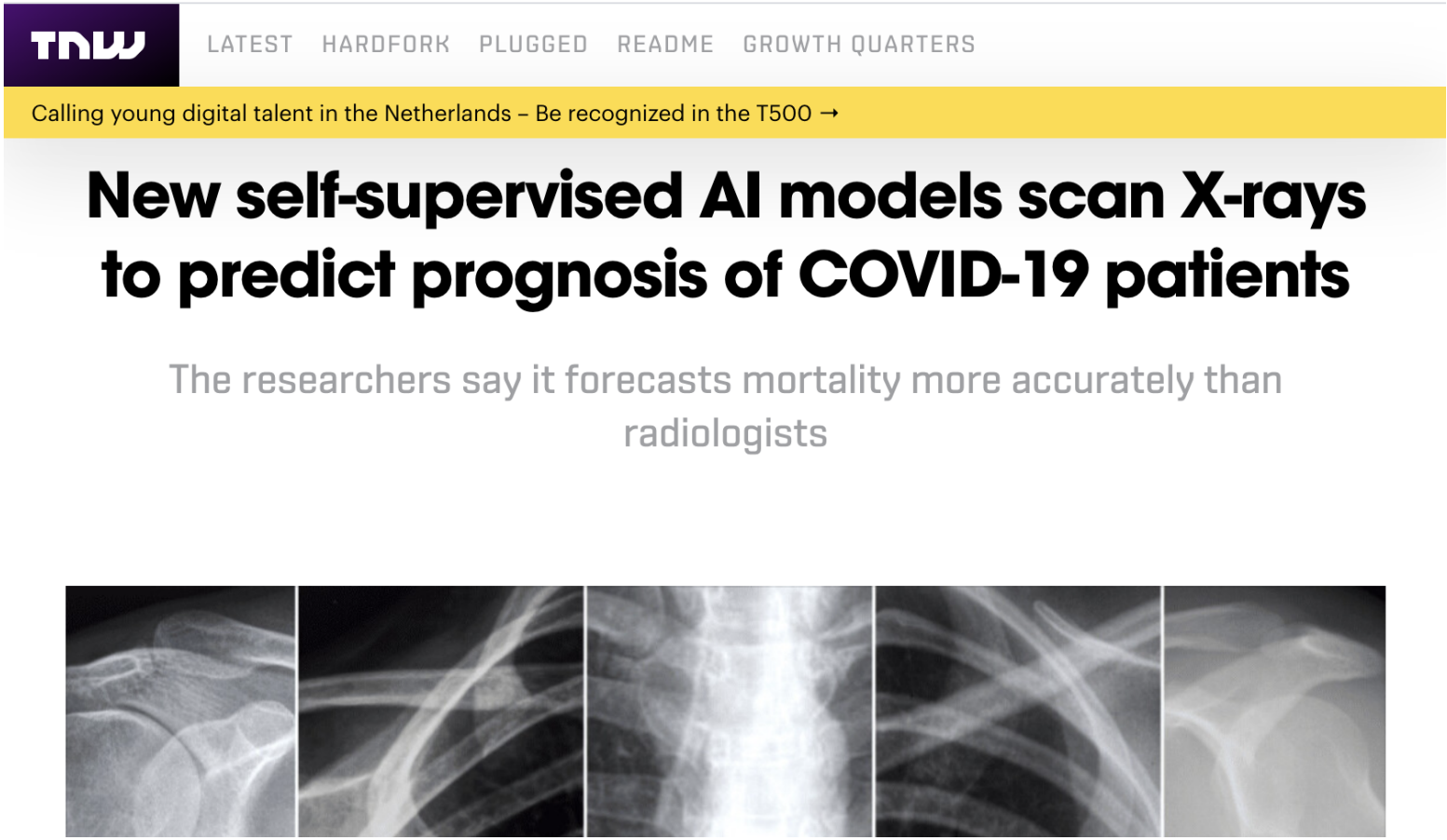
When Algorithms Give Real Students Imaginary Grades
The New York Times

Auditing AI and Autonomous Systems; Building an Infrastructure of Trust
Ryan Carrier

ACM RESOURCES

Security Awareness (Second

Self-Supervised Learning maybe the Future of AI



Self-Supervised Learning maybe the Future of AI

THE WALL STREET JOURNAL.

In self-supervised learning, AI can train itself without the need for external labels attached to the data. It doesn't need to be told "this is a cat" to identify other images of cats, or to distinguish between images of "cats" and "chairs."

Dr. LeCun is now focused on applying self-supervised learning to a more complex problem, computer vision, in which computers interpret images such as a person's face.

The next phase, possible over the next decade or two, is to try to create a machine that can "learn how the world works by watching video, listening to audio and reading text," he says.

Dr. LeCun, who shared the 2018 A.M. Turing Award for his work on deep learning, joined Facebook in 2013.

SHARE YOUR THOUGHTS

In what areas do you think self-learning AI might be transformative? Join the conversation below.

"Yann's a visionary," says Kyunghyun Cho, a professor of computer science and data science at New York University's Courant Institute of Mathematical Sciences, where Dr. LeCun also is affiliated.

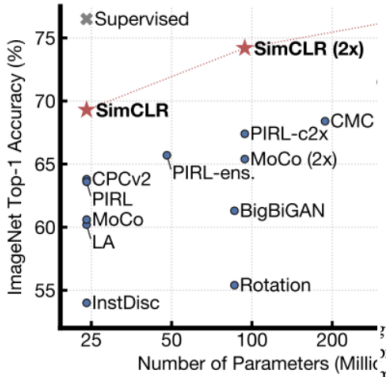
The push for self-supervised learning is a high priority at Facebook, which is under pressure from lawmakers, outside groups and its own users to crack down harder on misinformation and hate speech.

SimCLR has a big impact

Geoffrey Hinton
@geoffreyhinton

Unsupervised learning of representations is beginning to work quite well without requiring reconstruction.

> **Ting Chen** @tingchenai · Feb 14, 2020
>
> Introducing SimCLR: a Simple framework for Contrastive Learning of Representations. SimCLR advances previous SOTA in self-supervised and semi-supervised learning on ImageNet by 7-10% (see next).
>
> arxiv.org/abs/2002.05709
>
> Joint work with @skornblith @mo_norouzi @geoffreyhinton.
>
> Show this thread



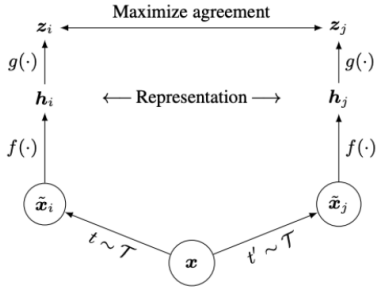


Figure 1. ImageNet top-1 accuracy of linear on representations learned with different self-s (pretrained on ImageNet). Gray cross indicates 50. Our method, SimCLR, is shown in bold.

ure 2. A simple framework for contrastive learning of visu presentations. Two separate data augmentation operators a mpled from the same family of augmentations ($t \sim \mathcal{T}$ a $\sim \mathcal{T}$) and applied to each data example to obtain two correlat ws. A base encoder network $f(\cdot)$ and a projection head g trained to maximize agreement using a contrastive loss. Af ining is completed, we through away the projection head g d use encoder $f(\cdot)$ and representation $\boldsymbol{h}$ for downstream tas

6:28 AM · Feb 15, 2020 · Twitter Web App

546 Retweets **17** Quote Tweets **2,122** Likes

Technical Terms

- Supervised Learning: using data with fine-grained human-annotated labels to train AI models.
- Semi-supervised Learning: using a small amount of labeled data and a large amount of unlabeled data.
- Weakly-supervised Learning: using data with coarse-grained labels or inaccurate labels.
  - The cost of making weak supervision labels is generally much cheaper than fine-grained labels for supervised methods.
  - e.g. label tiger, cat, dog as 'animal'
- Unsupervised Learning: learning methods without using any human-annotated labels.
  - Self-supervised Learning: models are explicitly trained with automatically generated labels.
    - The learned features can be transferred to multiple different tasks.
    - Self-supervised learning is a subset of unsupervised learning.

Self-Supervised Learning maybe the Future of AI

- Supervised learning
  - Supervised learning needs to train a model on a labeled data set.
  - Yann LeCun thinks it will play a diminishing role as self-supervised learning comes into wider use.
- Why Self-Supervised Learning? The motivation is straightforward.
  - Producing a dataset with clean labels is expensive but unlabeled data is being generated all the time.
  - Some areas are supervision-starved, e.g. medical data, where it is hard to obtain annotation.
  - Untapped/availability of vast numbers of unlabeled images/videos
    - Facebook: one billion images uploaded per day
    - 300 hours of videos are uploaded to YouTube every minute
  - To make full use of this much larger amount of unlabeled data, one way is to set the learning objectives properly so as to get supervision from the data itself.
  - Instead of relying on annotations, self-supervised learning generates labels from data by discovering relationships between the data
    - A step believed to be critical to achieving human-level intelligence.

Learning visual representations without human supervision

- Generative approaches
  - It learns to generate or otherwise model pixels in the input space.
  - Pixel-level generation is computationally expensive and may not be necessary for representation learning.
- Discriminative approaches
  - It learns representations using objective functions similar to those used for supervised learning.
  - It trains networks to perform pretext tasks where both the inputs and labels are derived from an unlabeled dataset.
  - Many such approaches used heuristics to design pretext tasks, which could limit the generality of the learned representations.
    - Discriminative approaches based on contrastive learning in the latent space have recently shown great promise, achieving great results.

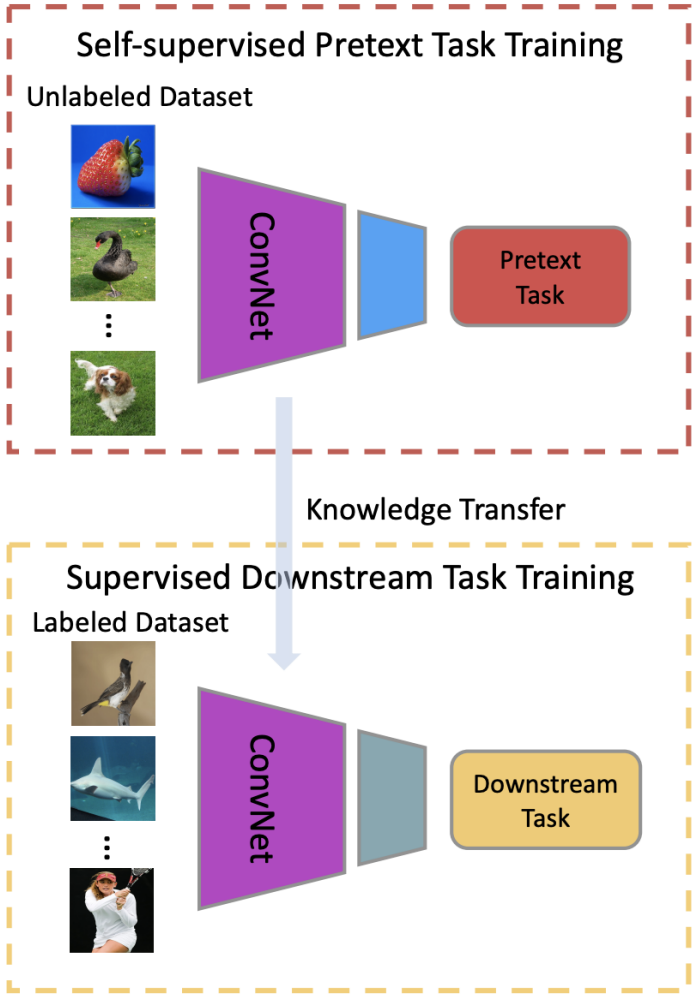
What is our goal? Define a Pretext Task or Proxy Task

- In self-supervised learning, we leverage the information already present in the data rather than any external labels.
  - Self-supervised learning enables us to exploit a variety of labels that come with the data for free.
  - Sometimes we say the model learns on its own.
- In reality, what we do is training a model for some other tasks that indirectly helps us to achieve our target goal.
  - These tasks are called "proxy tasks" or "pretext task".
  - The model supervised itself.

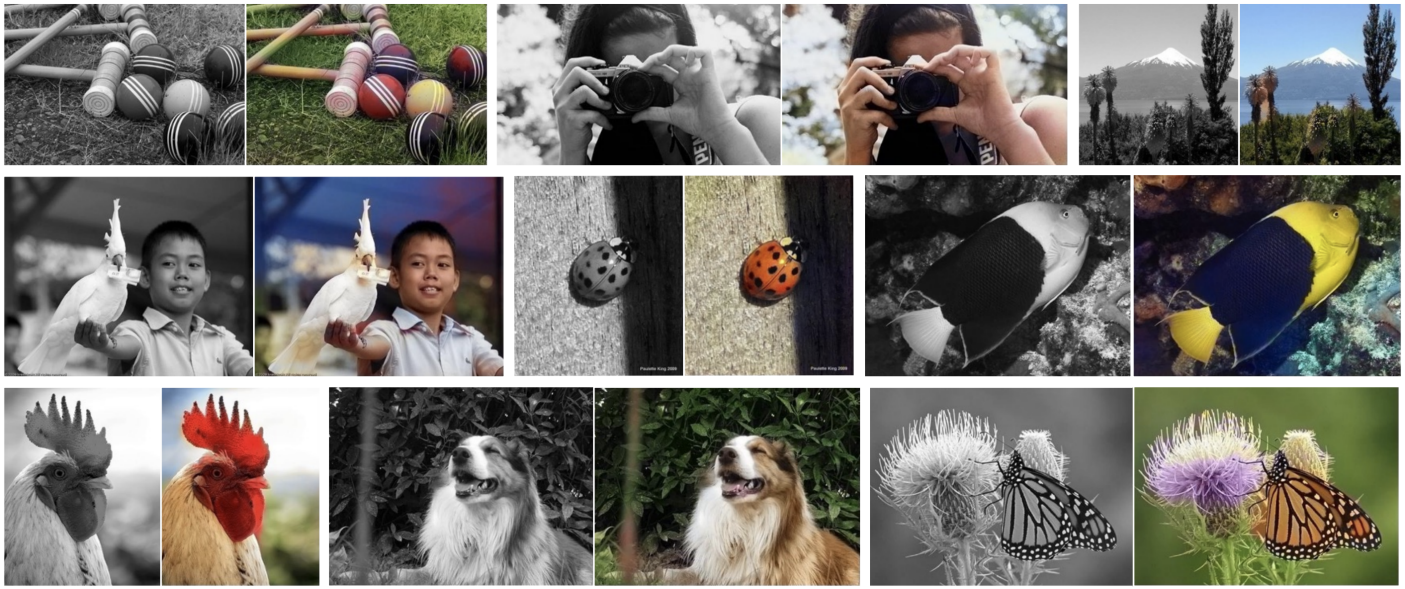
- Figure credit: iStock/RobHainer

The formal definitions

- Pretext Task
  - Pre-designed tasks for networks to solve, and features are learned by learning objective functions of pretext tasks.
- Downstream Task
  - Applications that are used to evaluate the quality of features learned by self-supervised learning.
  - These applications can greatly benefit from the pretrained models when training data are not enough.

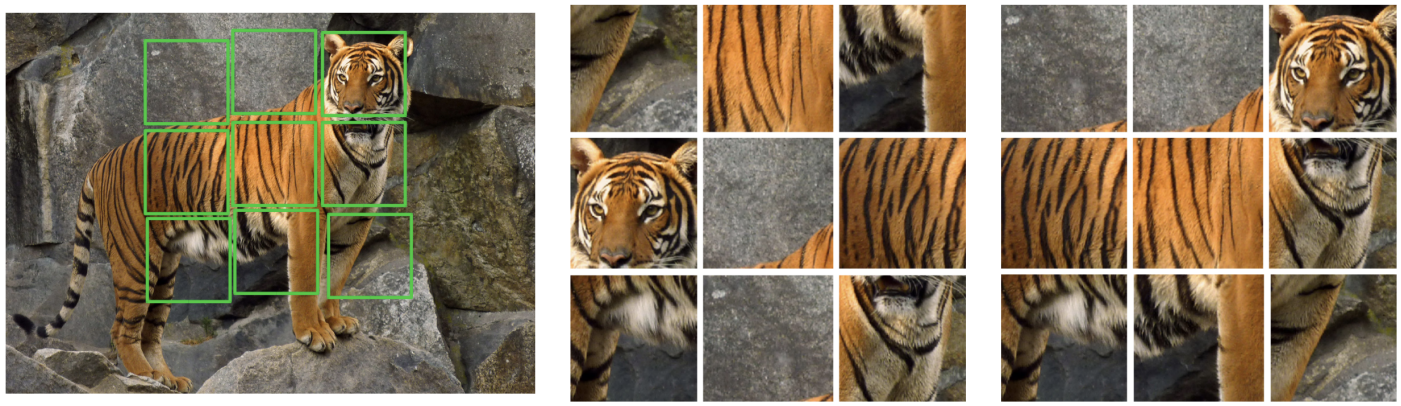


Examples of Pretext Task



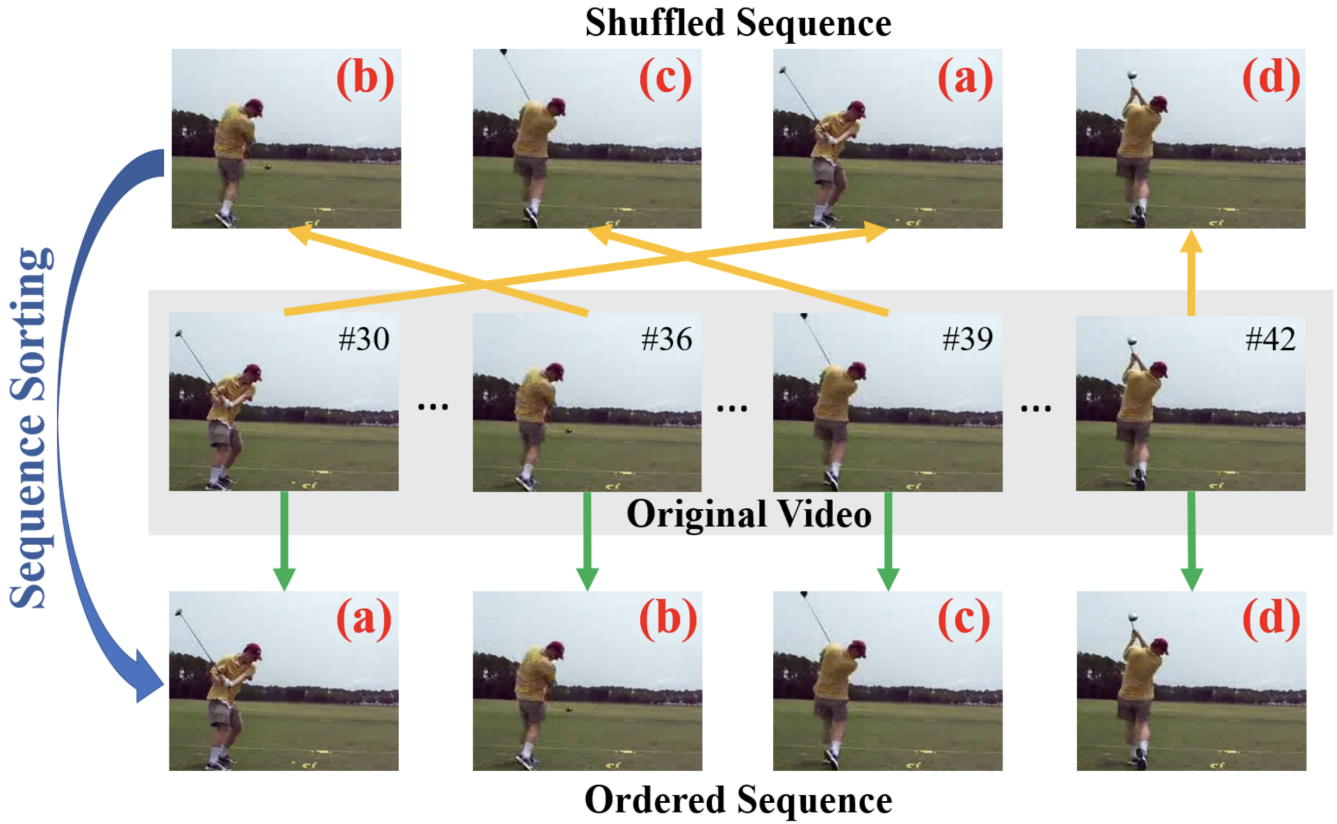
- Colourization
  - CNN model learns to predict colors from a grayscale image.
  - Zhang et al. (https://arxiv.org/abs/1603.08511)

Examples of Pretext Task



- Placing image patches in the right place
  - The patches are extracted from an image and shuffled.
  - The model learns to solve this jigsaw and arrange the patches in the correct sequence.
  - Noroozi and Favaro (https://arxiv.org/abs/1603.09246)

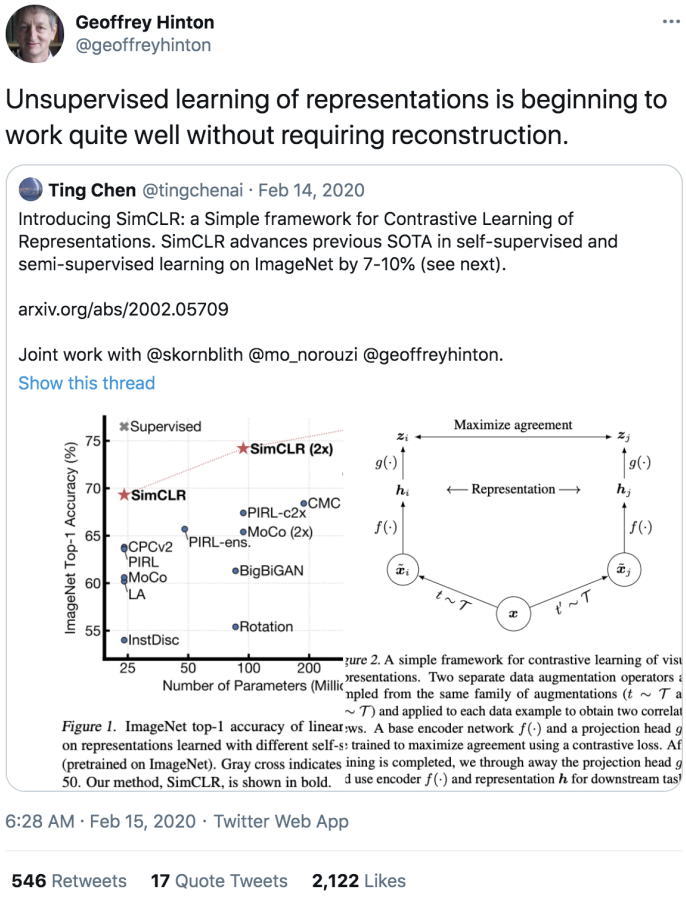
Examples of Pretext Task



- Placing frames in the right order
  - The model learns to sort the shuffled frames in a video sequence.
  - Lee et al. (https://arxiv.org/abs/1708.01246)

How SimCLR builds its Pretext Tasks?

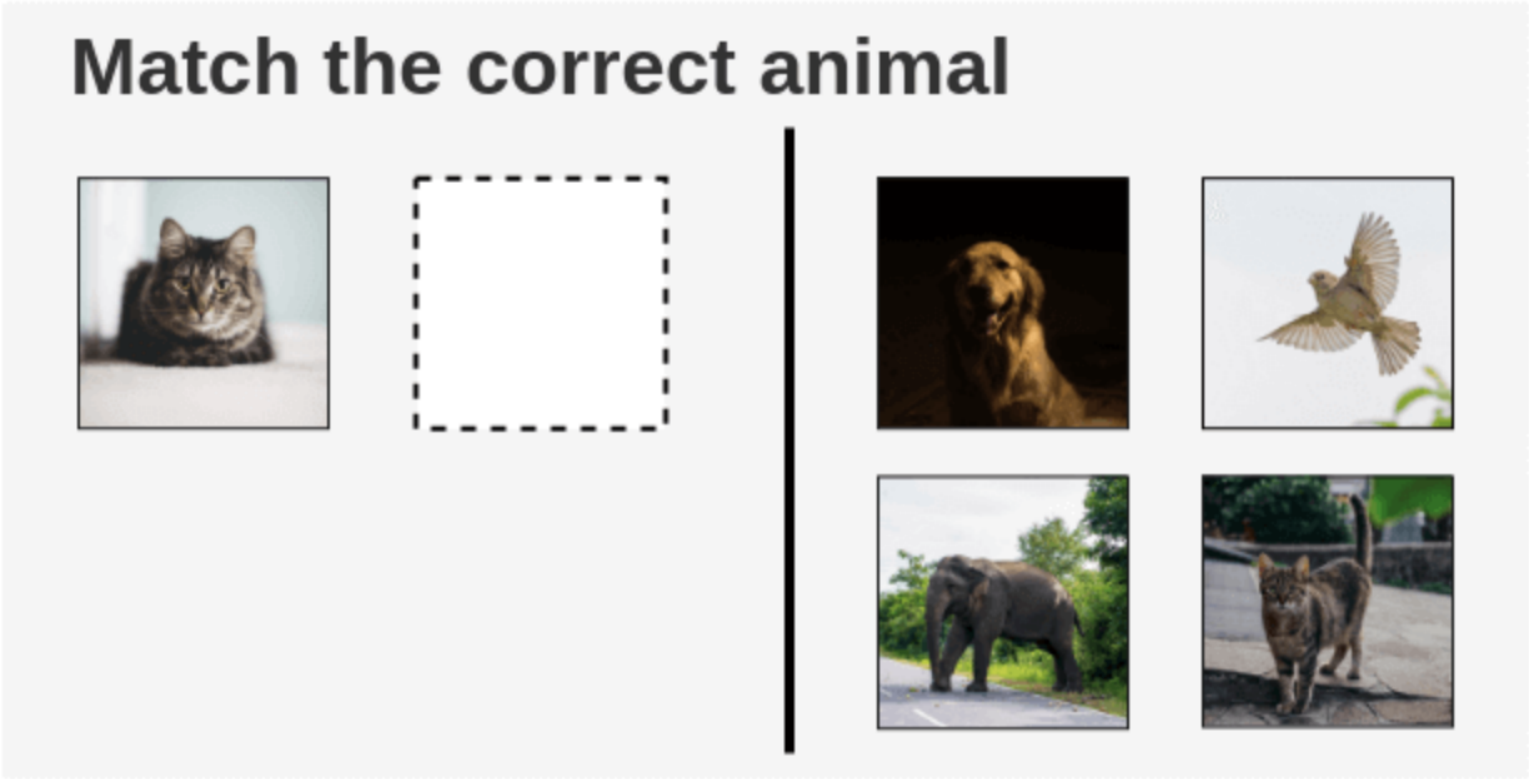
- Contrastive Learning
  - An approach to formulate the task of finding similar and dissimilar things for an ML model.
  - Using this approach, we can train a model to classify between similar and dissimilar images.



Outline

- Introduction to Self-Supervised Learning (SSL)
- **SimCLR: state-of-the-art SSL framework**
- Step-by-step example of SimCLR
- Results of SimCLR

The Intuition behind Contrastive Learning

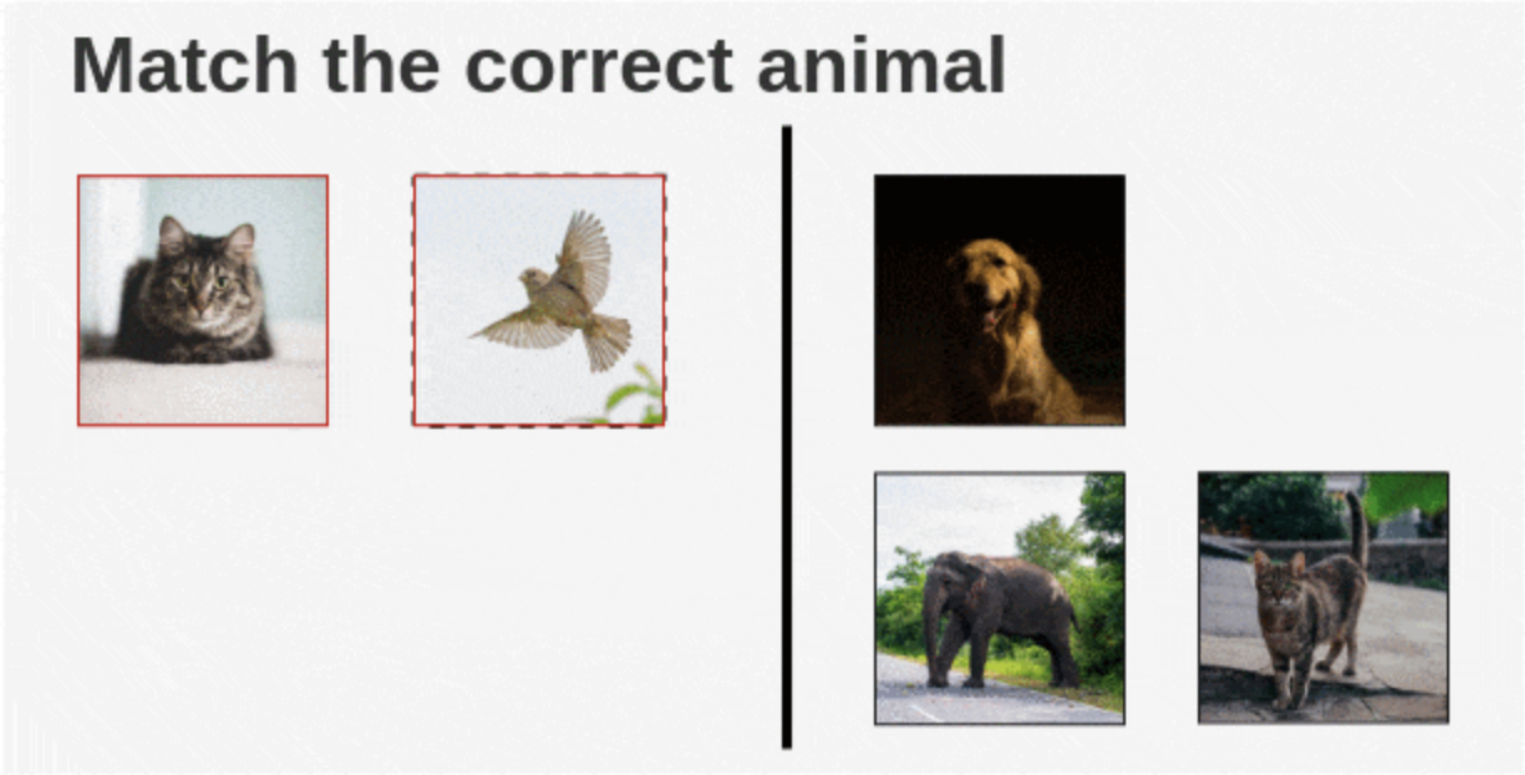


- Children can solve such puzzles easily
  - Looking at the picture on the left side, knowing it's a cat, then search for a cat on the right side
  - Children actually don't need to know the name of each animal

The Intuition behind Contrastive Learning

- Children can solve such puzzles easily
  - Looking at the picture on the left side, knowing it's a cat, then search for a cat on the right side
  - Children actually don't need to know the name of each animal

The Intuition behind Contrastive Learning

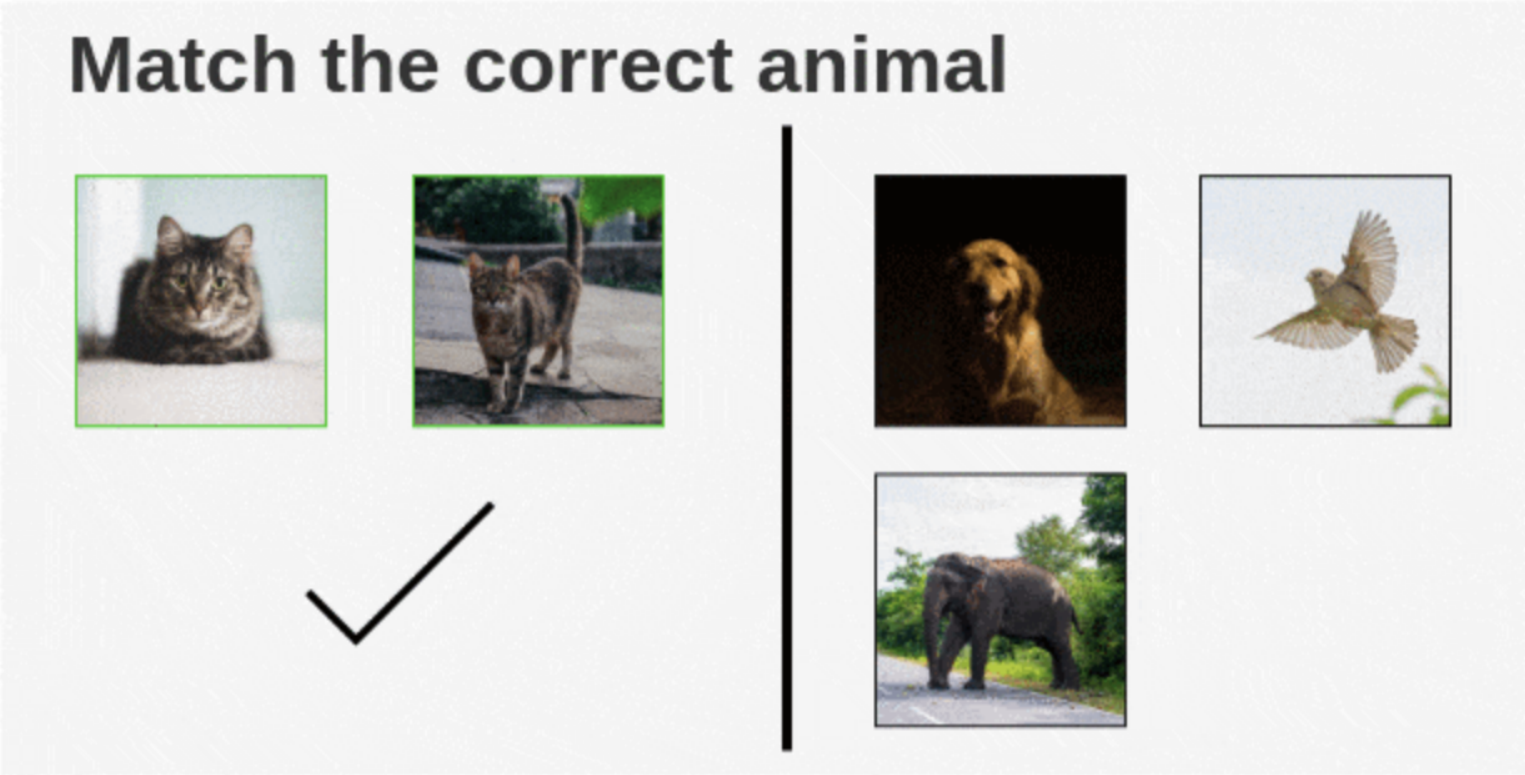


- Children can solve such puzzles easily
  - Looking at the picture on the left side, knowing it's a cat, then search for a cat on the right side
  - Children actually don't need to know the name of each animal

The Intuition behind Contrastive Learning

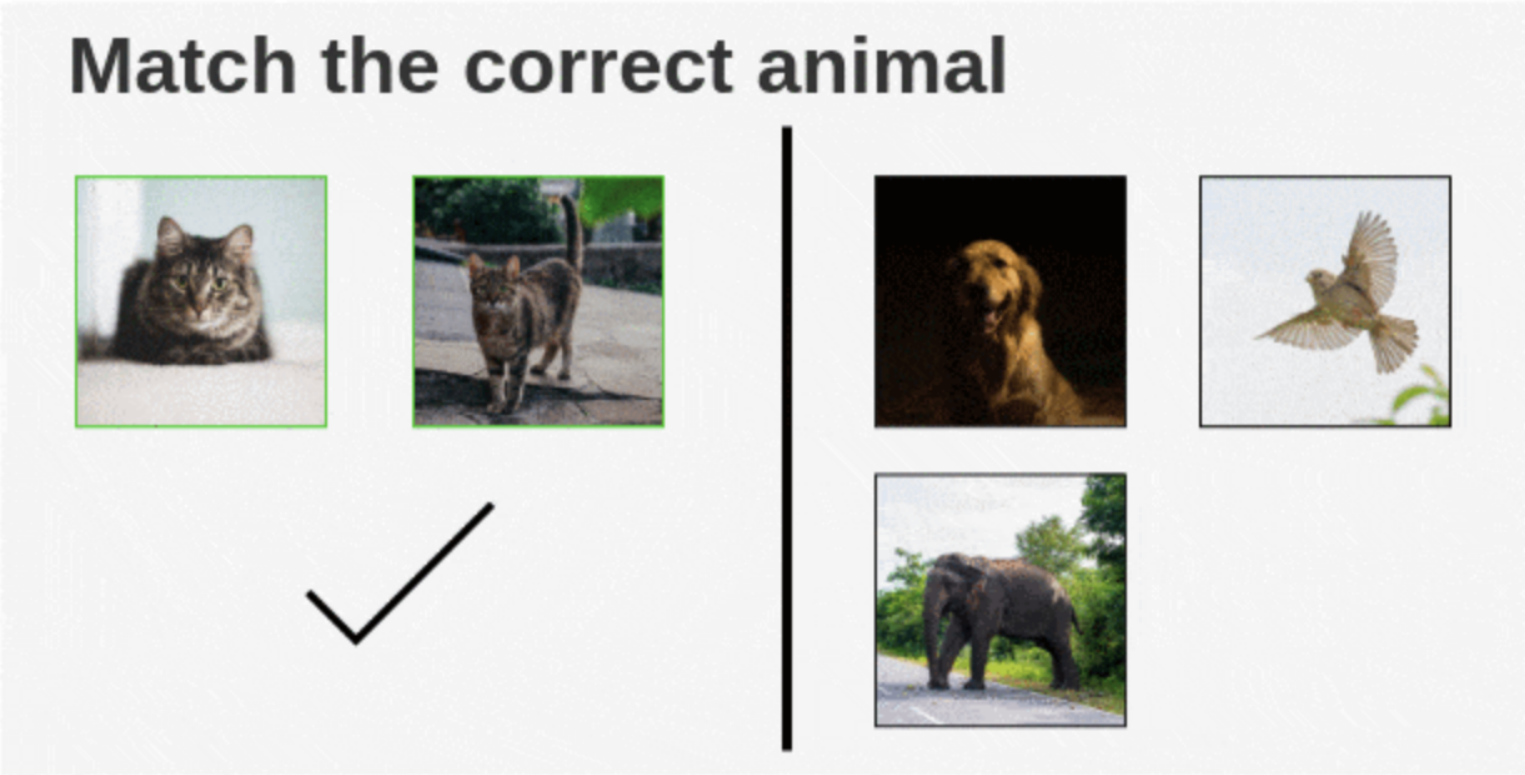
- Children can solve such puzzles easily
  - Looking at the picture on the left side, knowing it's a cat, then search for a cat on the right side
  - Children actually don't need to know the name of each animal

The Intuition behind Contrastive Learning



- Children can solve such puzzles easily
  - Looking at the picture on the left side, knowing it's a cat, then search for a cat on the right side
  - Children actually don't need to know the name of each animal

The Intuition behind Contrastive Learning

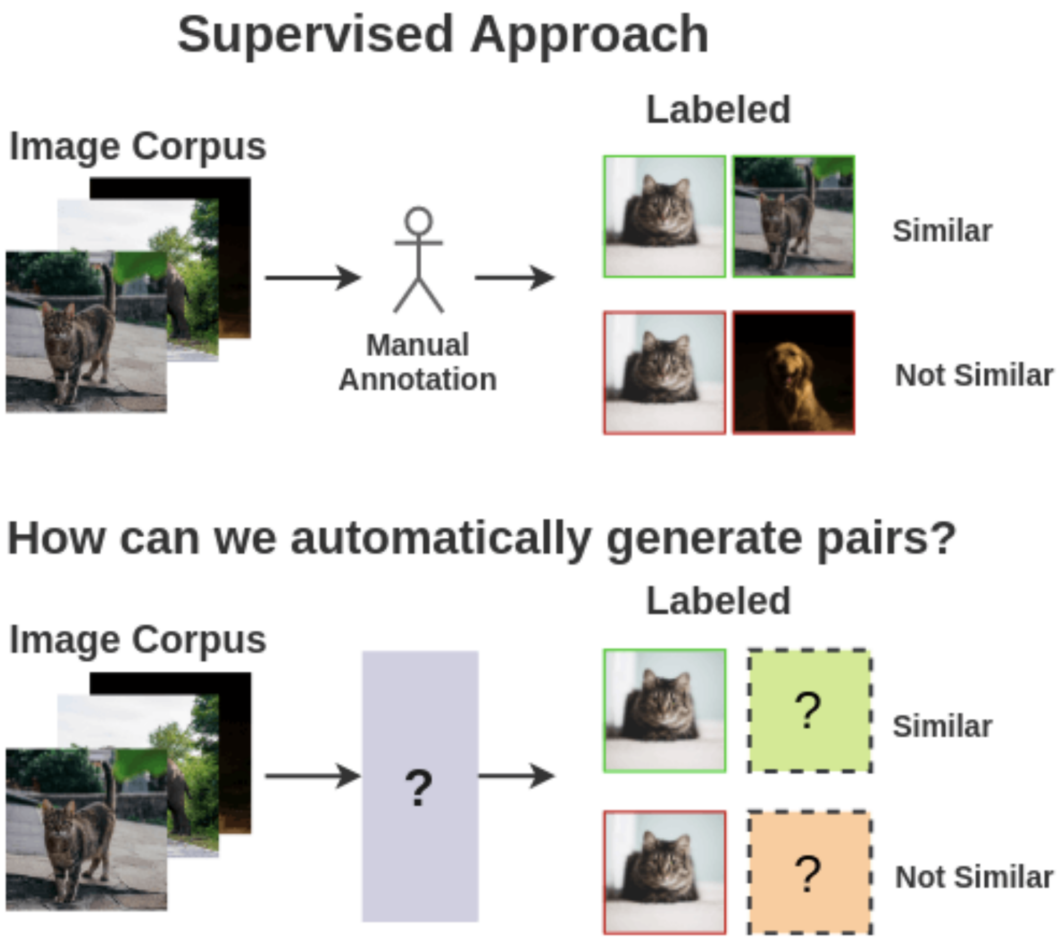


- Such exercises were prepared for children to be able to recognize an object and contrast that to other objects.
  - Can we use a similar way to teach machines?

Problem Formulation for Machines

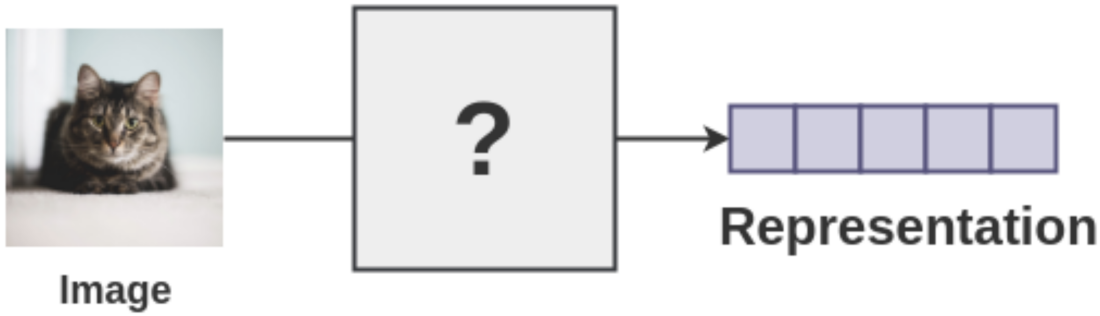
- To model this exercise for a machine rather than a kid, we need at least three things:
  - Examples of similar and dissimilar images
  - Ability to know what an image represents
  - Ability to quantify if two images are similar

Examples of similar and dissimilar images



- We need example pairs of images that are similar and images that are different to train a model.
- The supervised school of thought would require a human to manually annotate such pairs.
  - For unsupervised learning, we need to automate this process.
  - But how can we formulate it?

Ability to know what an image represents



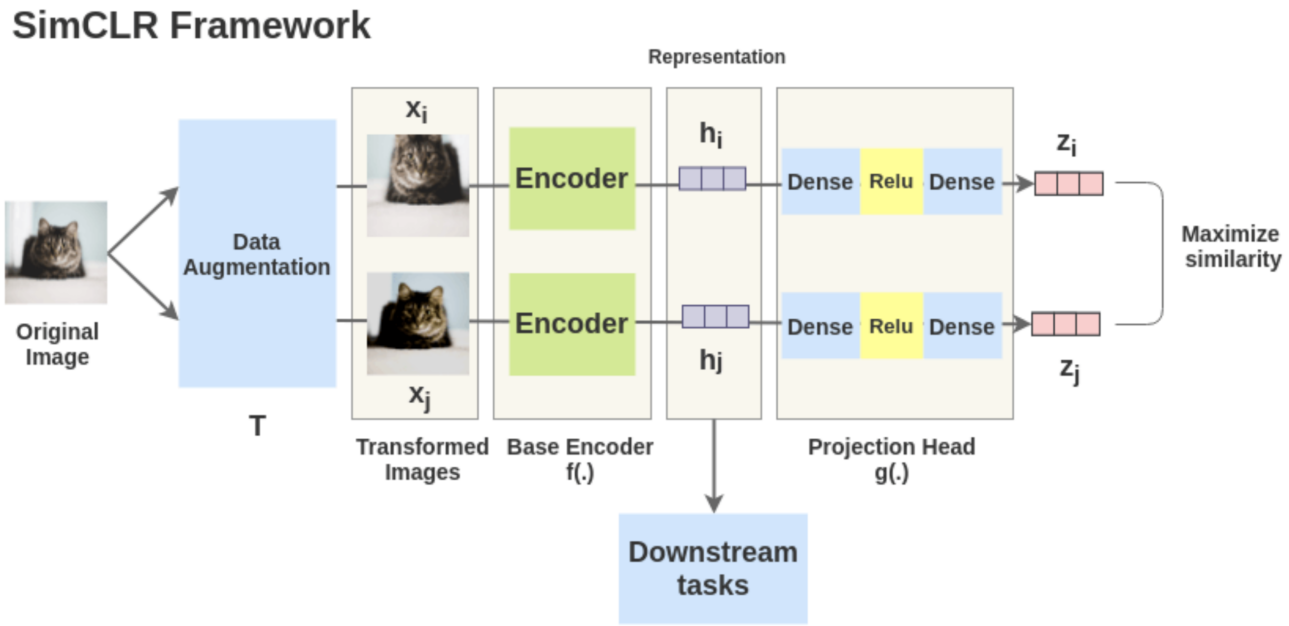
- We need some techniques to get representations that allow the machine to understand an image.
  - To find the pattern

Ability to quantify if two images are similar

similarity( , )

- We need some methods to compute the similarity of two images.

The key idea of SimCLR Framework



- An image is taken and random transformations are applied to it to get a pair of two augmented images x_i and x_j.
- Each image in that pair is passed through an encoder to get representations.
- Then a non-linear fully connected layer is applied to get representations z.
- The task is to maximize the similarity between these two representations z_i and z_j for the same image.

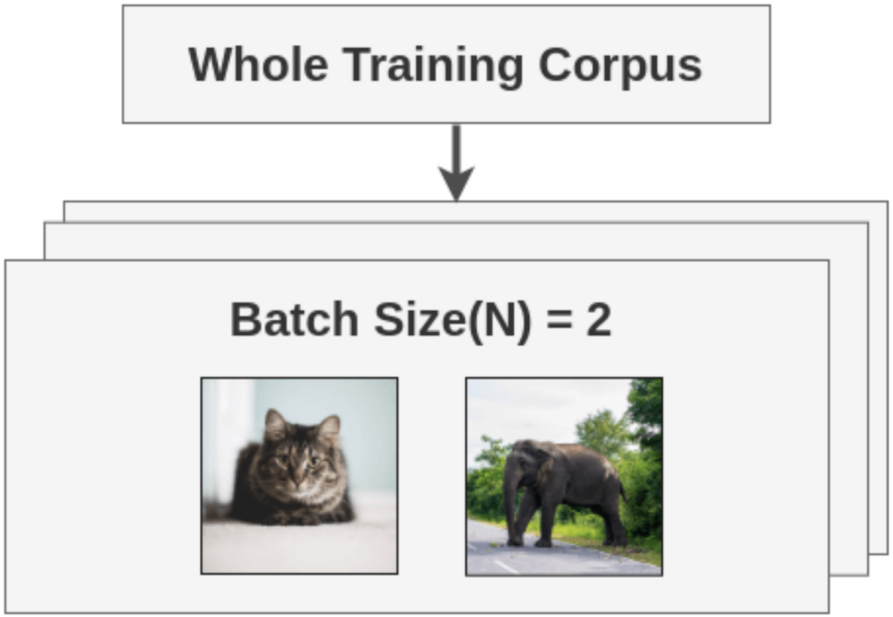
Outline

- Introduction to Self-Supervised Learning (SSL)
- SimCLR: state-of-the-art SSL framework
- **Step-by-step example of SimCLR**
- Results of SimCLR

Step by Step Example for SimCLR

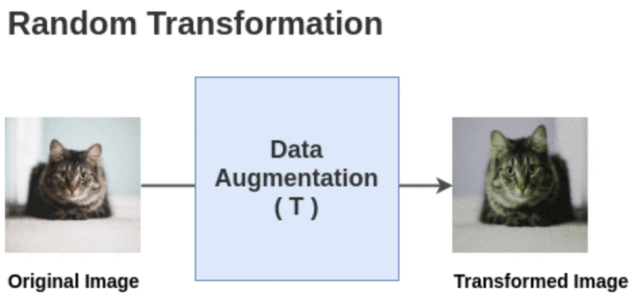
- Suppose we have a training corpus of millions of unlabeled images.
  - Let us explore the components of the SimCLR framework with an example.
- Example figure source: https://cs.stanford.edu/people/karpathy/cnnembed/

1. Self-supervised Formulation [Data Augmentation]

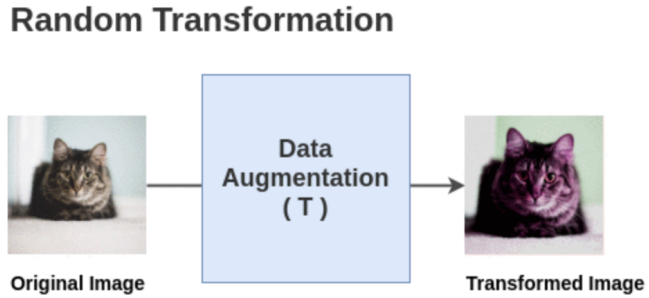


- We generate batches of size N from the raw images.
  - Let's take a batch of size N = 2 for simplicity.
  - The authors of SimCLR use a large batch size of 8192.

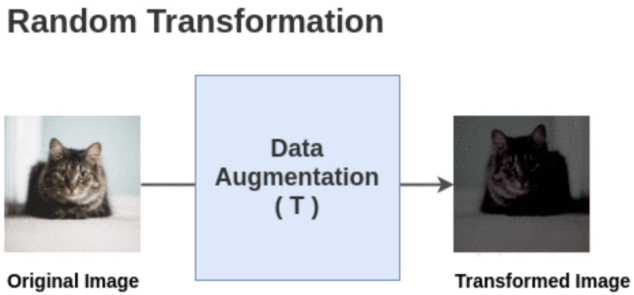
1. Self-supervised Formulation [Data Augmentation]



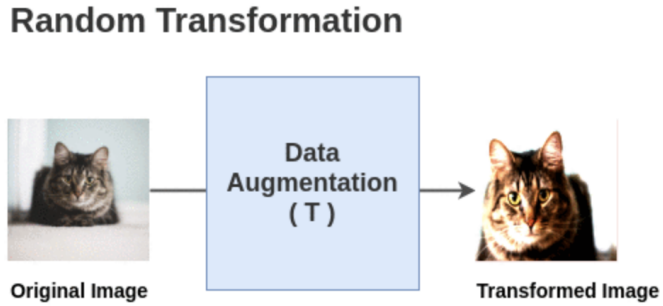
(a) Combination 1



(b) Combination 2



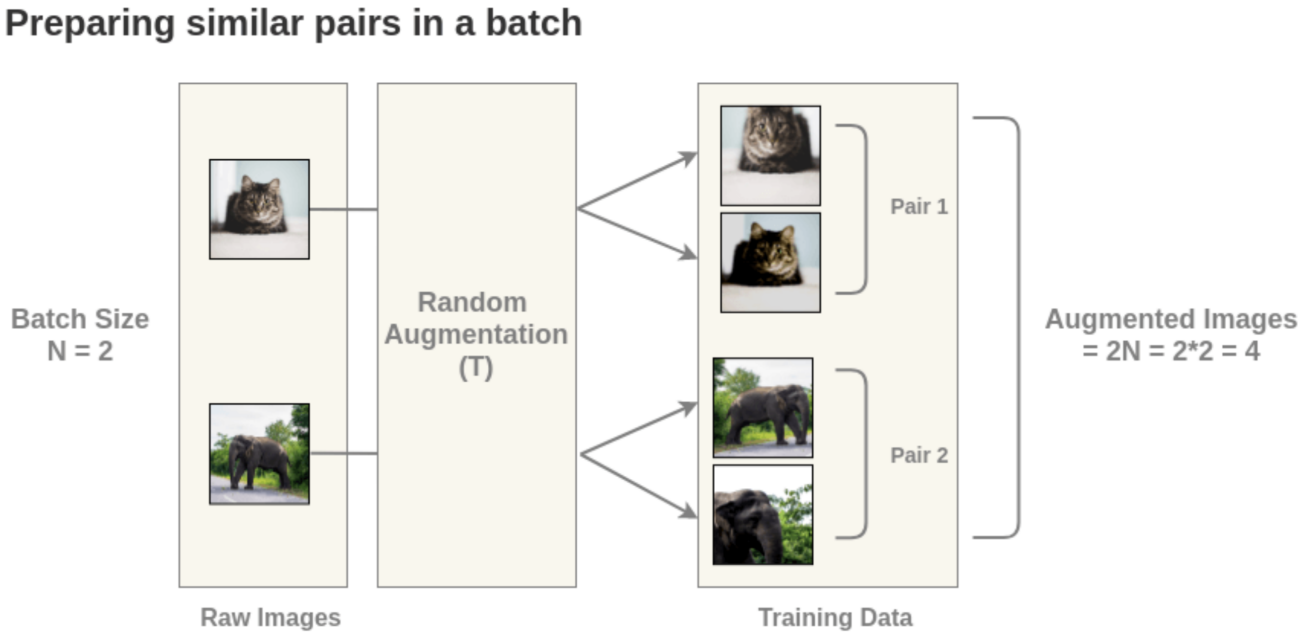
(c) Combination 3



(d) Combination 4

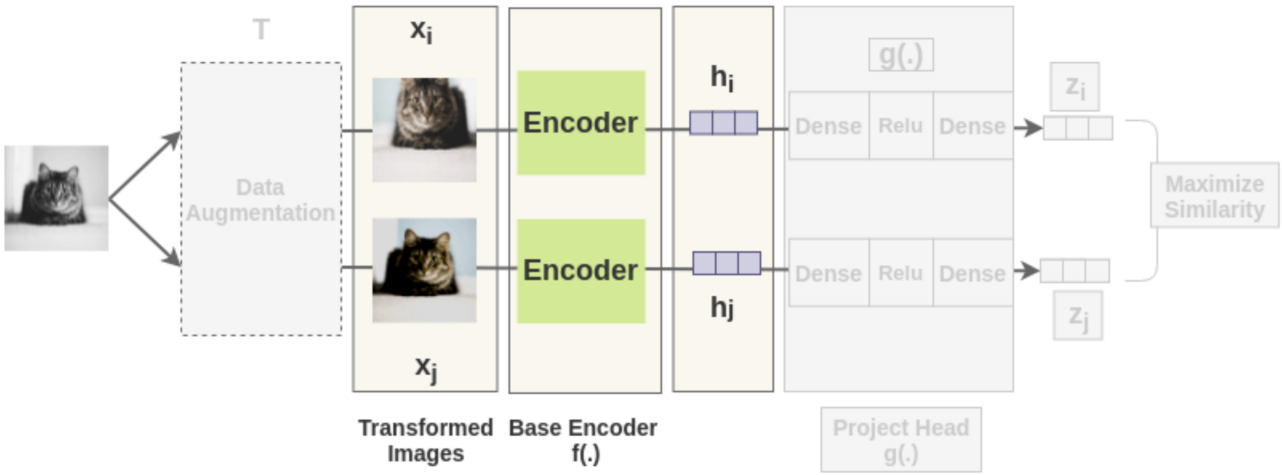
- The authors define a random transformation function T that takes an image and applies a combination of random data augmentation operations (crop + flip + color jitter + grayscale).

1. Self-supervised Formulation [Data Augmentation]



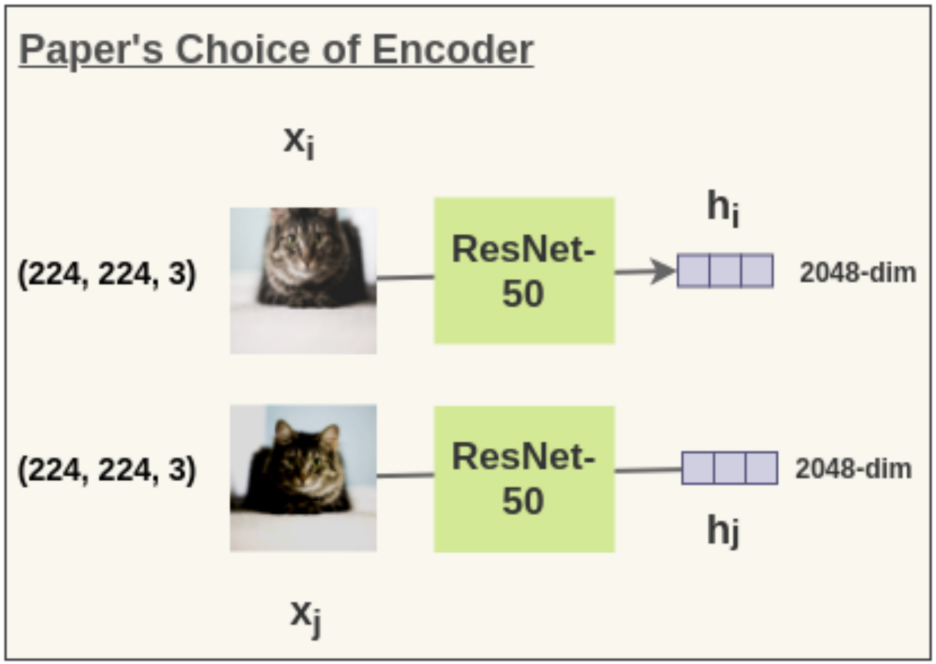
- For each image in this batch, a random transformation function is applied to get a pair of 2 images.
  - Thus, for a batch size of 2, we get $2\times N = 2\times 2 = 4$ total images.

2. Getting Representations [Base Encoder]



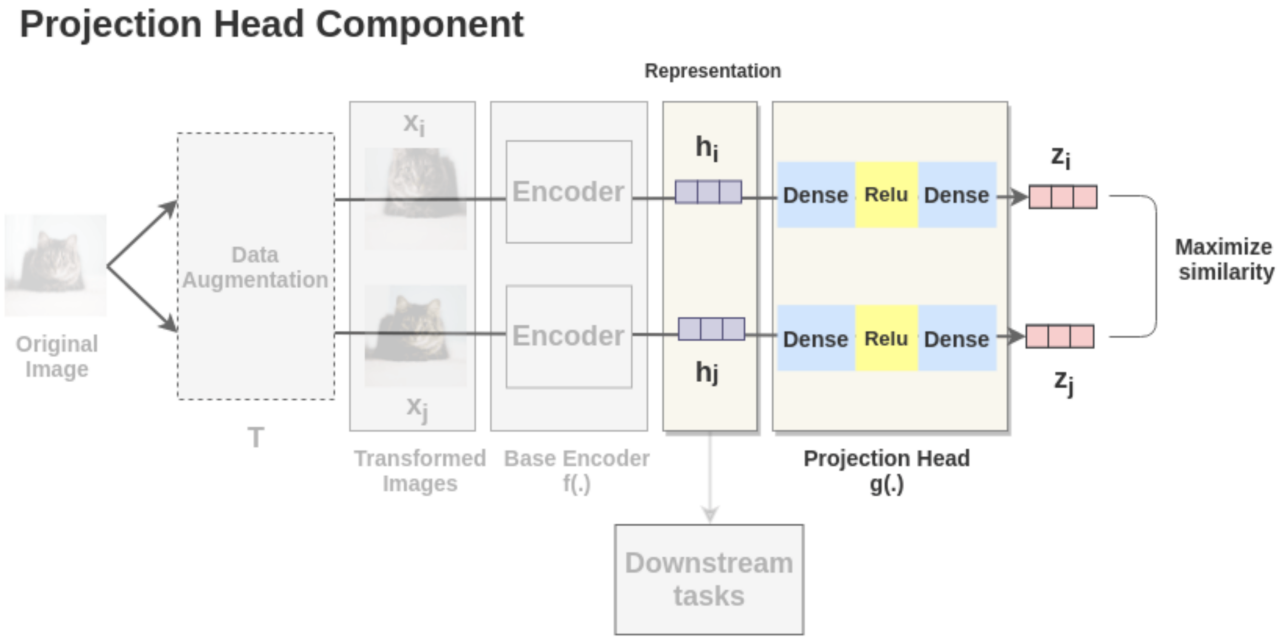
- Each augmented image in a pair is passed through an encoder to get image representations.
  - The encoder is general and replaceable with other architectures.
  - The two encoders have shared weights and we get vectors h_i and h_j.

2. Getting Representations [Base Encoder]



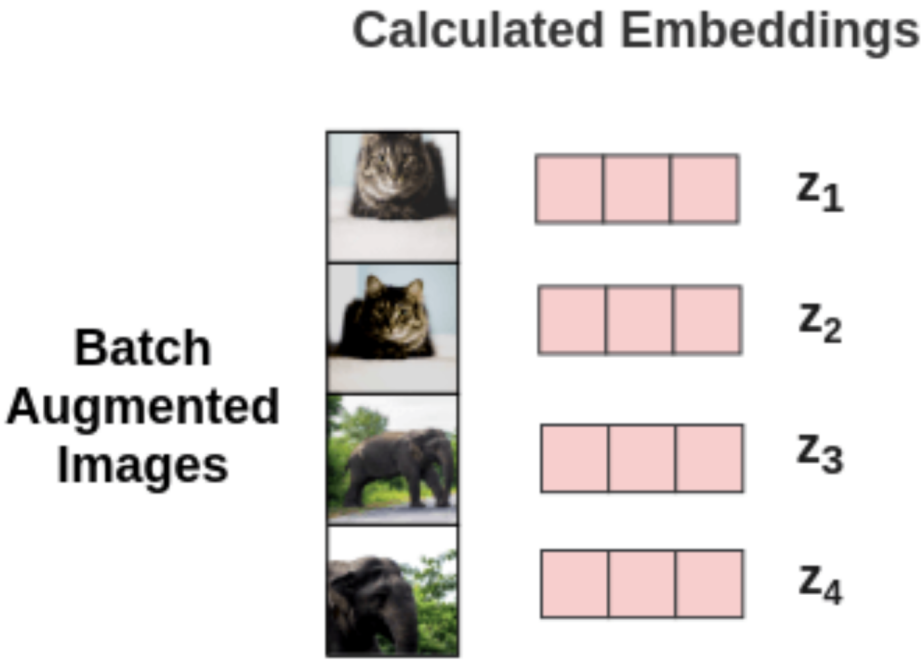
- For example, the authors used ResNet-50 architecture as the ConvNet encoder.
  - The output is a 2048-dimensional vector h.

3. Projection Head



- The representations h_i and h_j of the two augmented images are then passed through several non-linear Dense $\rightarrow$ Relu $\rightarrow$ Dense layers to apply non-linear transformation and project it into a representation z_i and z_j.
  - This is denoted by $\mathbf{g(.)}$ in the paper and called projection head.

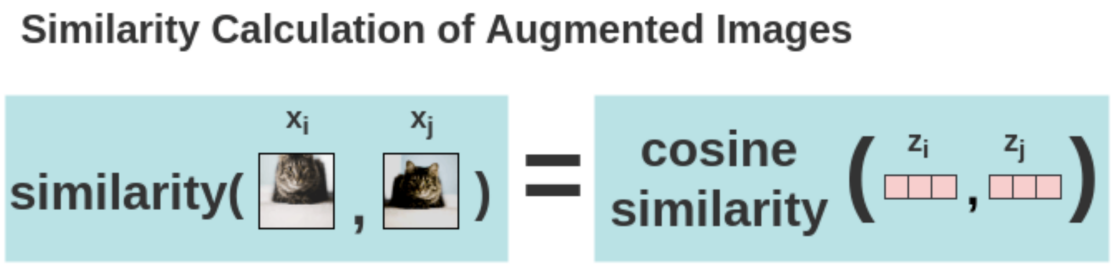
4. Tuning Model: [Bringing similar closer]



- For each augmented image in the batch, we get embedding vectors z for it.
  - From these embedding, how can we calculate the loss?

4-a. Calculation of Cosine Similarity

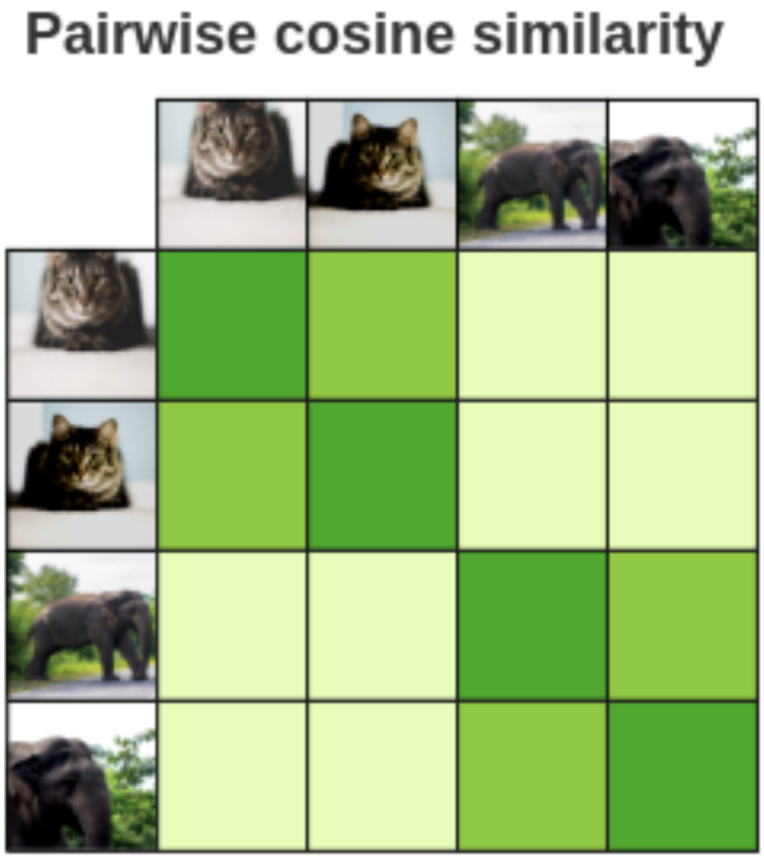
- The similarity between two augmented versions of an image is calculated using cosine similarity.
  - For two augmented images x_i and x_j, the cosine similarity is calculated on its projected representations z_i and z_j.



$$s_{i,j} = \frac{z_i^T z_j}{(\tau ||z_i|| ||z_j||)}$$

- τ is the adjustable temperature hyper-parameter. It can scale the inputs and widen the range [-1, 1] of cosine similarity
- $||z_i||$ is the norm of the vector z_i

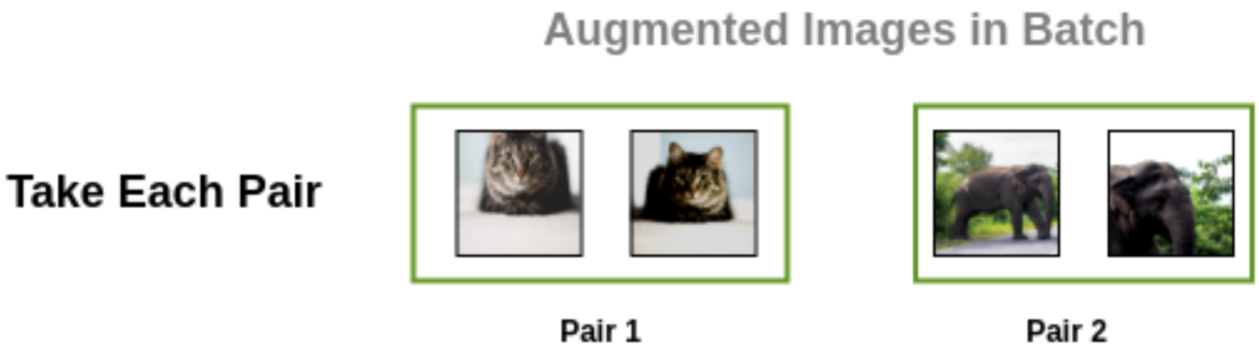
4-a. Calculation of Cosine Similarity



- The pairwise cosine similarity between each augmented image in a batch is calculated using the above formula.
  - In an ideal case, the similarity between augmented images of cats is high while the similarity between cat and elephant images is low.

4-b. Loss Calculation

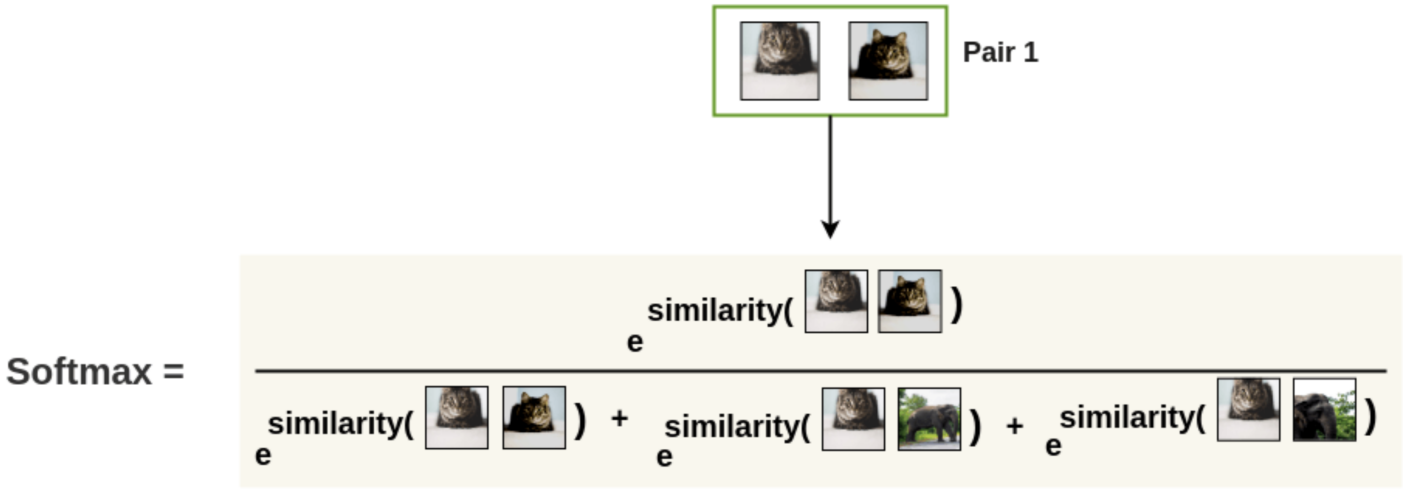
- SimCLR uses a contrastive loss called "NT-Xent loss" (Normalized Temperature-Scaled Cross-Entropy Loss)[1].
- First, the augmented pairs in the batch are taken one by one.



[1] https://paperswithcode.com/method/nt-xent

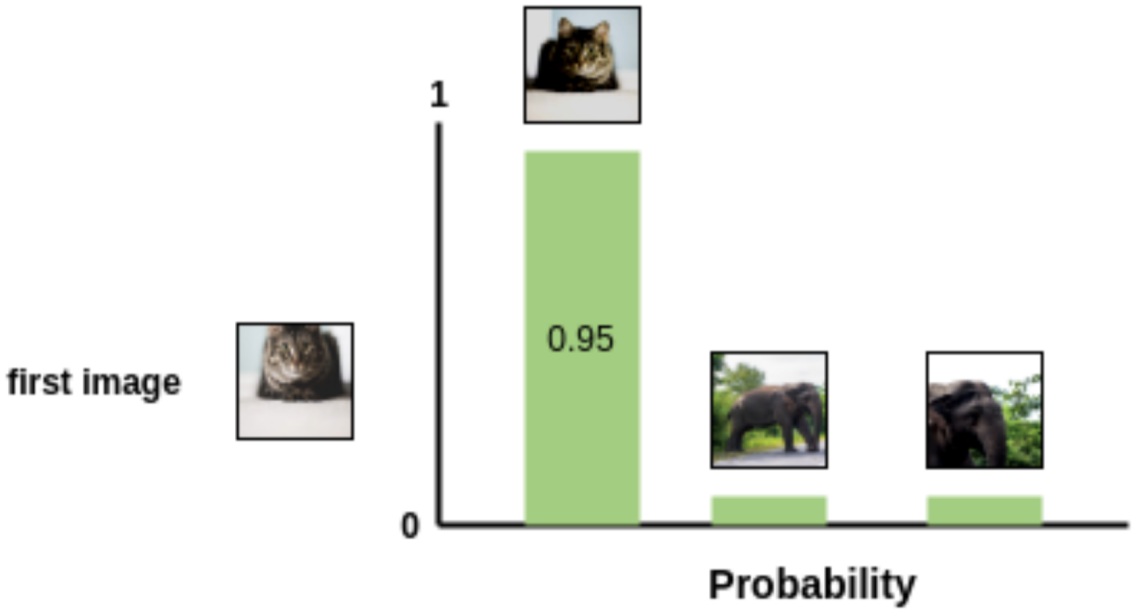
4-b. Loss Calculation

- Next, we apply the softmax function to get the probability of these two images being similar.



4-b. Loss Calculation

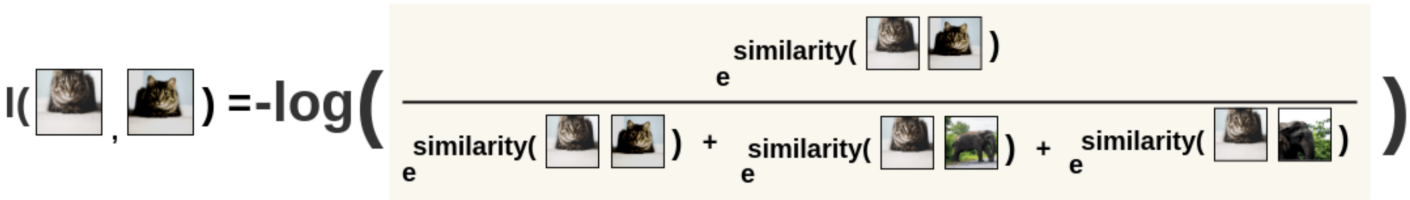
- This softmax calculation is equivalent to getting the probability of the second augmented cat image being the most similar to the first cat image in the pair.
  - Here, all remaining images in the batch are sampled as a dissimilar image (negative pair).
  - Thus, we don't need specialized architecture, memory bank or queue need by previous approaches like InstDisc, MoCo or PIRL.



4-b. Loss Calculation

- Then, the loss is calculated for a pair by taking the negative of the log of the Softmax calculation.
- This formulation is the Noise Contrastive Estimation(NCE) Loss.

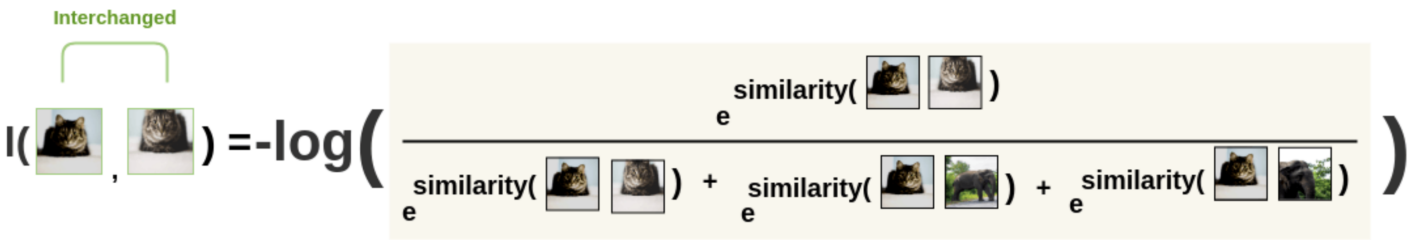
$$l(i,j) = -log \frac{exp(s_{i,j})}{\sum_{k=1}^{2N} l_{[k!=i]} exp(s_{i,k})}$$



$$\frac{a}{a+b+c} = \frac{1}{1+\frac{b+c}{a}}$$

4-b. Loss Calculation

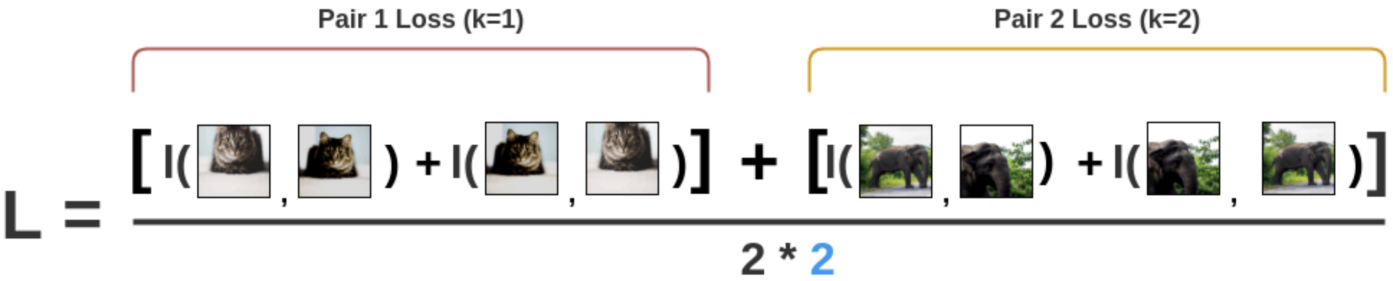
- We interchange the positions of the images and calculate the loss for the same pair again.



4-b. Loss Calculation

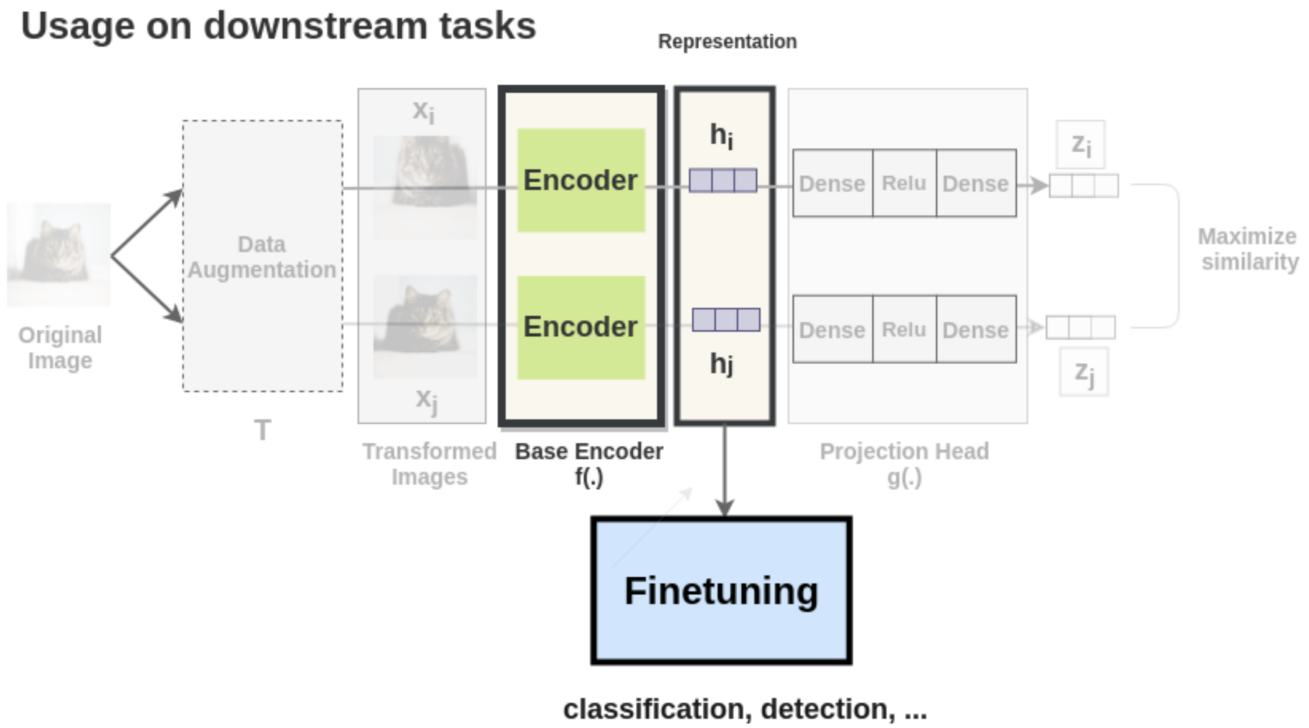
- Finally, we compute loss over all the pairs in the batch of size N and take an average.
  - For this example, $N = 2$

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



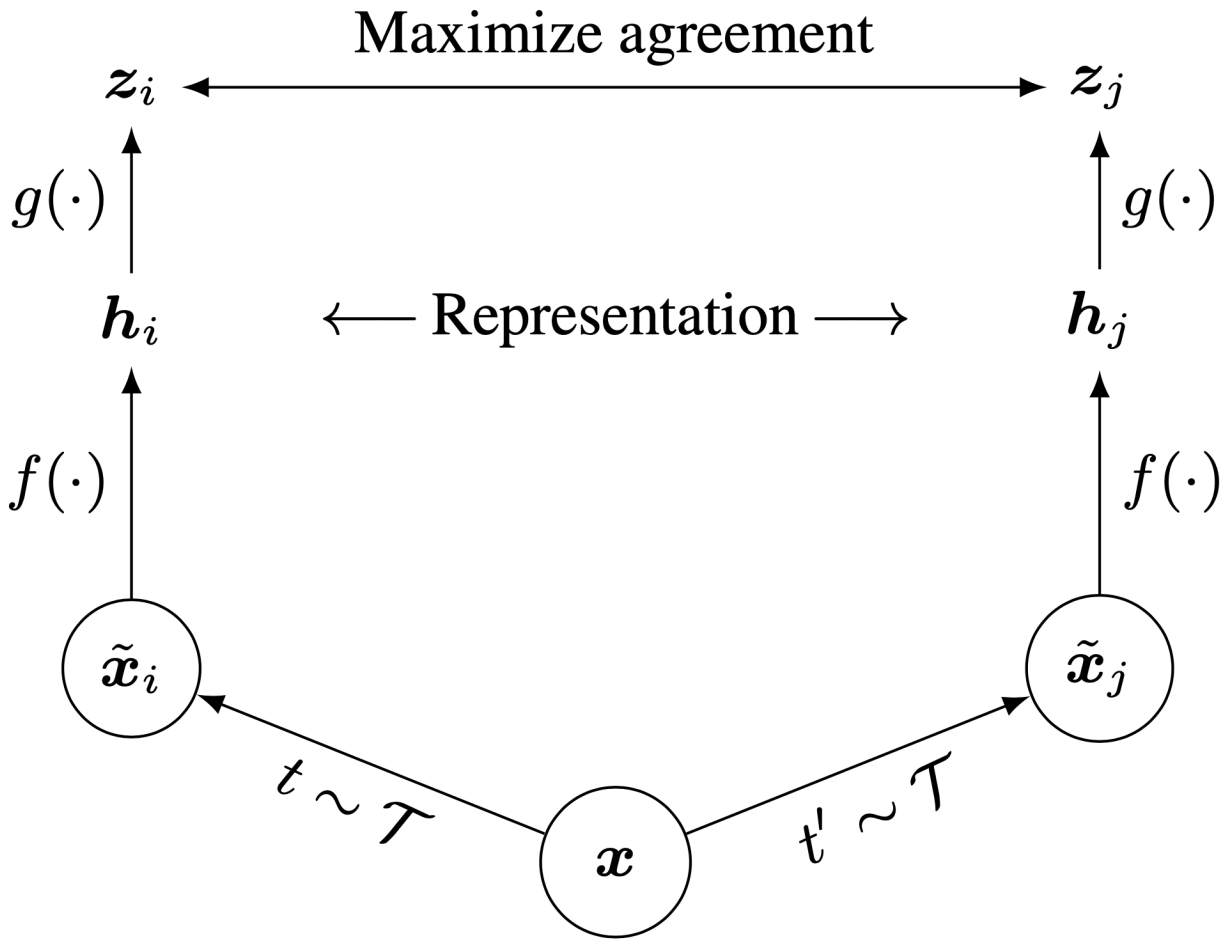
- Based on the loss, the encoder and projection head representations improves over time.
  - The representations obtained place similar images closer.

Downstream Tasks



- Once the SimCLR model is trained on the contrastive learning task, it can be used for transfer learning.
- For this, the representations from the encoder are used instead of representations obtained from the projection head.
- These representations can be used for downstream tasks like ImageNet Classification.

DataFlow of SimCLR



- Two separate data augmentation operators are sampled from the same family of augmentations ($t \sim \tau$ and $t' \sim \tau$) and applied to each data example to obtain two correlated views.
- An encoder $f(\cdot)$ and a projection head $g(\cdot)$ are trained to maximize agreement using a contrastive loss.
- After training is completed, we throw away the projection head $g(\cdot)$ and use encoder $f(\cdot)$ and representation h for downstream tasks.

SimCLR Algorithm

Algorithm 1 SimCLR's main learning algorithm.

input: batch size N, constant τ, structure of $f, g, \mathcal{T}$.
for sampled minibatch $\{\boldsymbol{x}_k\}_{k=1}^{N}$ **do**
  for all $k \in \{1, \ldots, N\}$ **do**
    draw two augmentation functions $t \sim \mathcal{T}$, $t' \sim \mathcal{T}$
    # the first augmentation
    $\tilde{\boldsymbol{x}}_{2k-1} = t(\boldsymbol{x}_k)$
    $\boldsymbol{h}_{2k-1} = f(\tilde{\boldsymbol{x}}_{2k-1})$ # representation
    $\boldsymbol{z}_{2k-1} = g(\boldsymbol{h}_{2k-1})$ # projection
    # the second augmentation
    $\tilde{\boldsymbol{x}}_{2k} = t'(\boldsymbol{x}_k)$
    $\boldsymbol{h}_{2k} = f(\tilde{\boldsymbol{x}}_{2k})$ # representation
    $\boldsymbol{z}_{2k} = g(\boldsymbol{h}_{2k})$ # projection
  end for
  for all $i \in \{1, \ldots, 2N\}$ and $j \in \{1, \ldots, 2N\}$ **do**
    $s_{i,j} = \boldsymbol{z}_i^{\top} \boldsymbol{z}_j / (\|\boldsymbol{z}_i\| \|\boldsymbol{z}_j\|)$ # pairwise similarity
  end for
  define $\ell(i, j)$ **as** $\ell(i, j) = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(s_{i,k}/\tau)}$
  $\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} [\ell(2k-1, 2k) + \ell(2k, 2k-1)]$
  update networks f and g to minimize $\mathcal{L}$
end for
return encoder network $f(\cdot)$, and throw away $g(\cdot)$

4 major components of SimCLR

- A stochastic data augmentation module that transforms any given data example randomly resulting in two correlated views of the same example, denoted $\tilde{x}_i$ and $\tilde{x}_j$ (positive pair).
  - SimCLR sequentially applied 3 simple augmentations:
    - random cropping followed by resize back to the original size
    - random color distortions
    - random Gaussian blur
  - The combination of random crop and color distortion is crucial to a good performance.
- A neural network base encoder $f(\cdot)$ that extracts representation vectors from augmented data examples.
  - SimCLR allows different choices of the network architecture without any constraints.
  - For simplicity, SimCLR used ResNet for $h_i = f(x_i) = \text{ResNet}(x_i)$ where $h_i \in R^d$ is the output after the average pooling layer.

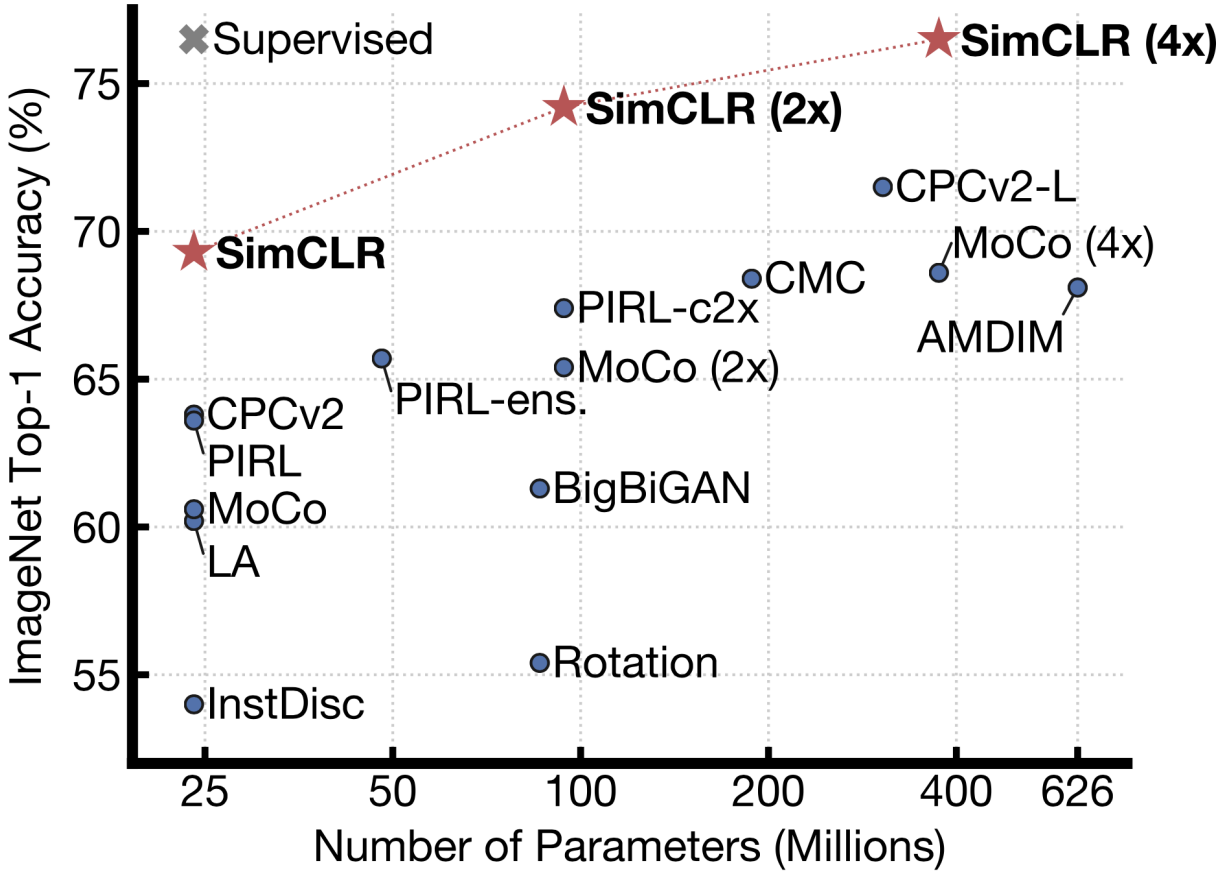
4 major components of SimCLR

- A small neural network projection head $g(\cdot)$ that maps representations to the space where contrastive loss is applied.
  - SimCLR used a MLP with one hidden layer to obtain $z_i = g(h_i) = W^{(2)}ReLU(W^{(1)}h_i)$.
  - It is beneficial to define the contrastive loss on z_i's rather than h_i's.
- A contrastive loss function defined for a contrastive prediction task.
  - Given a data batch $\{\tilde{x}_k\}$ including a positive pair of examples $\tilde{x}_i$ and $\tilde{x}_j$, the contrastive prediction task aims to identify $\tilde{x}_j$ in $\{\tilde{x}_k\}_{k\neq i}$ for a given data sample $\tilde{x}_i$.

Outline

- Introduction to Self-Supervised Learning (SSL)
- SimCLR: state-of-the-art SSL framework
- Step-by-step example of SimCLR
- **Results of SimCLR**

ImageNet Results

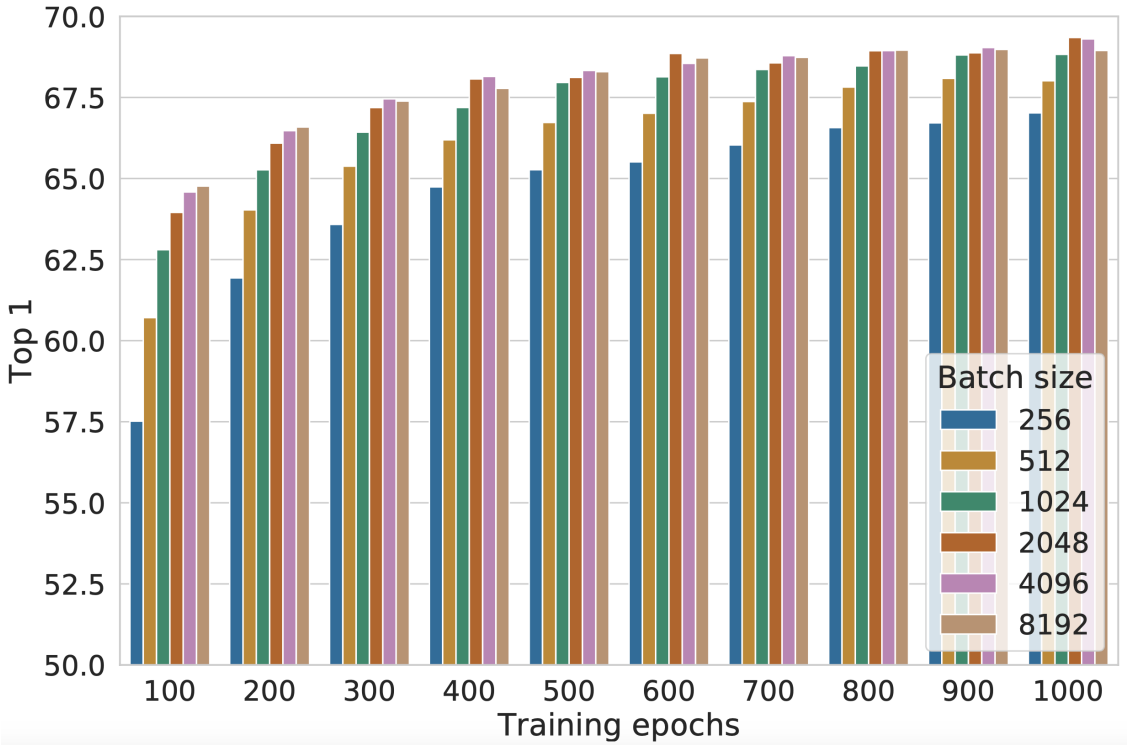


- ImageNet Top-1 accuracy of linear classifiers trained on representations learned with different self-supervised methods (pretrained on ImageNet).
- SimCLR models are trained for 1000 epochs (supervised models only need 90 epochs).
- SimCLR outperformed previous self-supervised methods on ImageNet.
- ResNet-50 in 3 different hidden layer widths (width multipliers of 1×, 2×, and 4×)

Training with Large-Batch Size

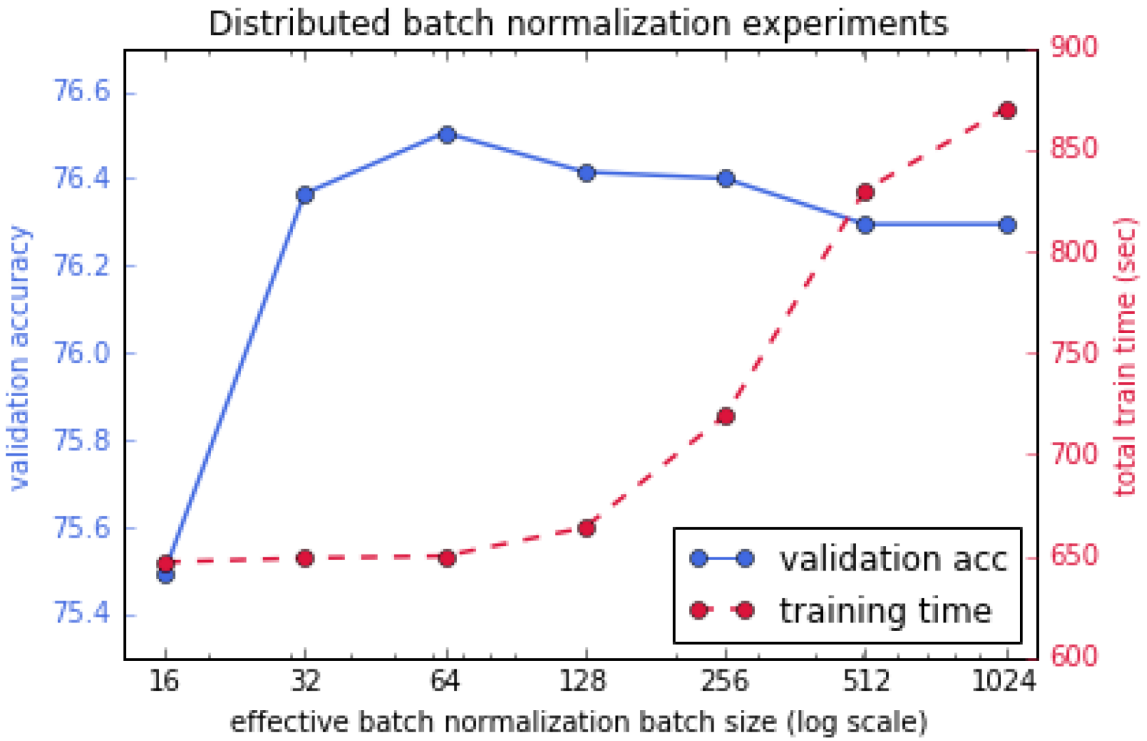
- Negative examples
  - SimCLR randomly sampled a minibatch of N examples and define the contrastive prediction task on pairs of augmented examples derived from the minibatch, resulting in $2N$ data points.
  - SimCLR did not sample negative examples explicitly.
    - Instead, given a positive pair, SimCLR treats the other $2(N-1)$ augmented examples as negative examples.
- SimCLR scales training batch size N from 256 to 8192.
- A batch size of 8192 provides 16382 (i.e. 16K - 2) negative examples per positive pair from both augmentation views.
- Training with large batch size may be unstable when using standard SGD/Momentum with linear learning rate scaling.
- To stabilize the training, SimCLR used LARS optimizer (You et al., 2017) for all batch sizes.

It benefits more from larger batch sizes and longer training



- When the number of training epochs is small (100 epochs), larger batch sizes have a significant advantage over small batch sizes.
  - Why? Larger batch sizes provide more negative examples, facilitating convergence.
- With more training epochs, the gaps between different batch sizes decrease or disappear (provided the batches are randomly re-sampled).
  - Why? Training longer also provides more negative examples, improving the results.

Effective Batch Size of Batch Normalization (BN)



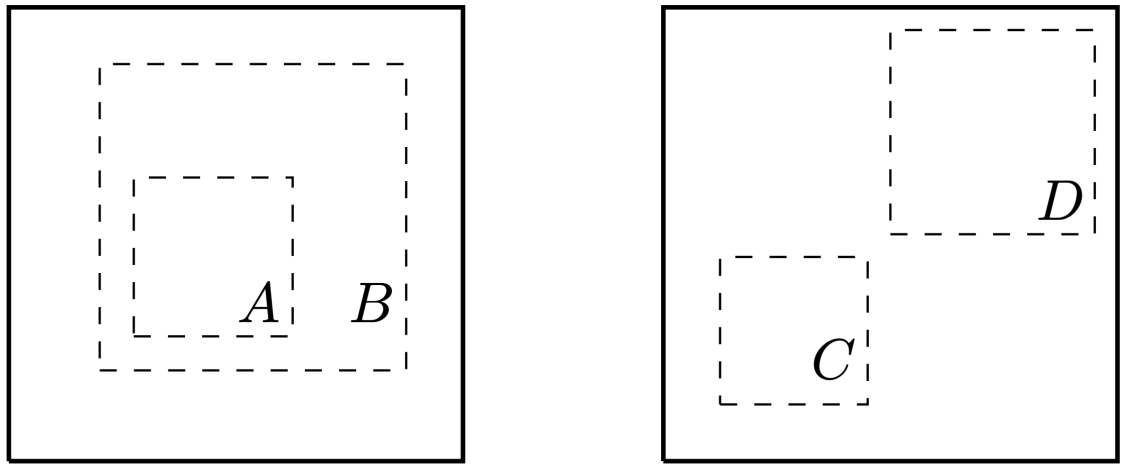
- Results for regular ImageNet/ResNet-50 (supervised learning)
- Ying et al., 2018 (https://arxiv.org/abs/1811.06992)

Effective Batch Size of Batch Normalization (BN)

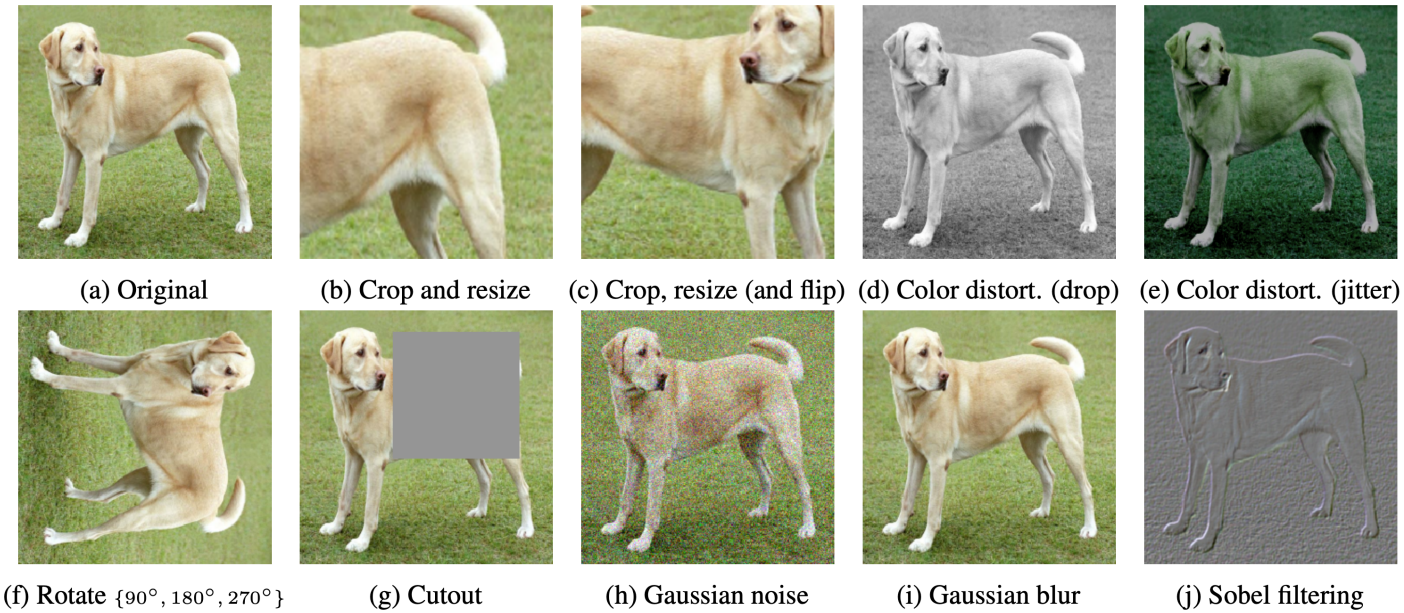
- In distributed training with data parallelism, the BN mean and variance are typically aggregated locally per device.
  - e.g. local batch size (batch size per device) is 64
    - In this situation, even the global batch size is 16384 (256 devices), the effective batch size for BN is 64
- In SimCLR, as positive pairs are computed in the same device, the model can use the local information leakage to improve prediction accuracy without improving representations.
  - They address this issue by aggregating BN mean and variance over all devices during the training.

Data Augmentation for Contrastive Prediction Task

- Many existing approaches define contrastive prediction tasks by changing the architecture rather than using data augmentation.
  - Global-to-local view prediction via constraining the receptive field in the network architecture (Hjelm et al., 2018)
  - Neighboring view prediction via a fixed image splitting procedure and a context aggregation network (Oord et al., 2018)
- This complexity can be avoided by performing simple random cropping (with resizing) of target images.
  - Solid rectangles are images, dashed rectangles are random crops.
  - SimCLR sampled contrastive prediction tasks like global to local view (B → A) or adjacent view (D → C).



Data Augmentation for Contrastive Prediction Task

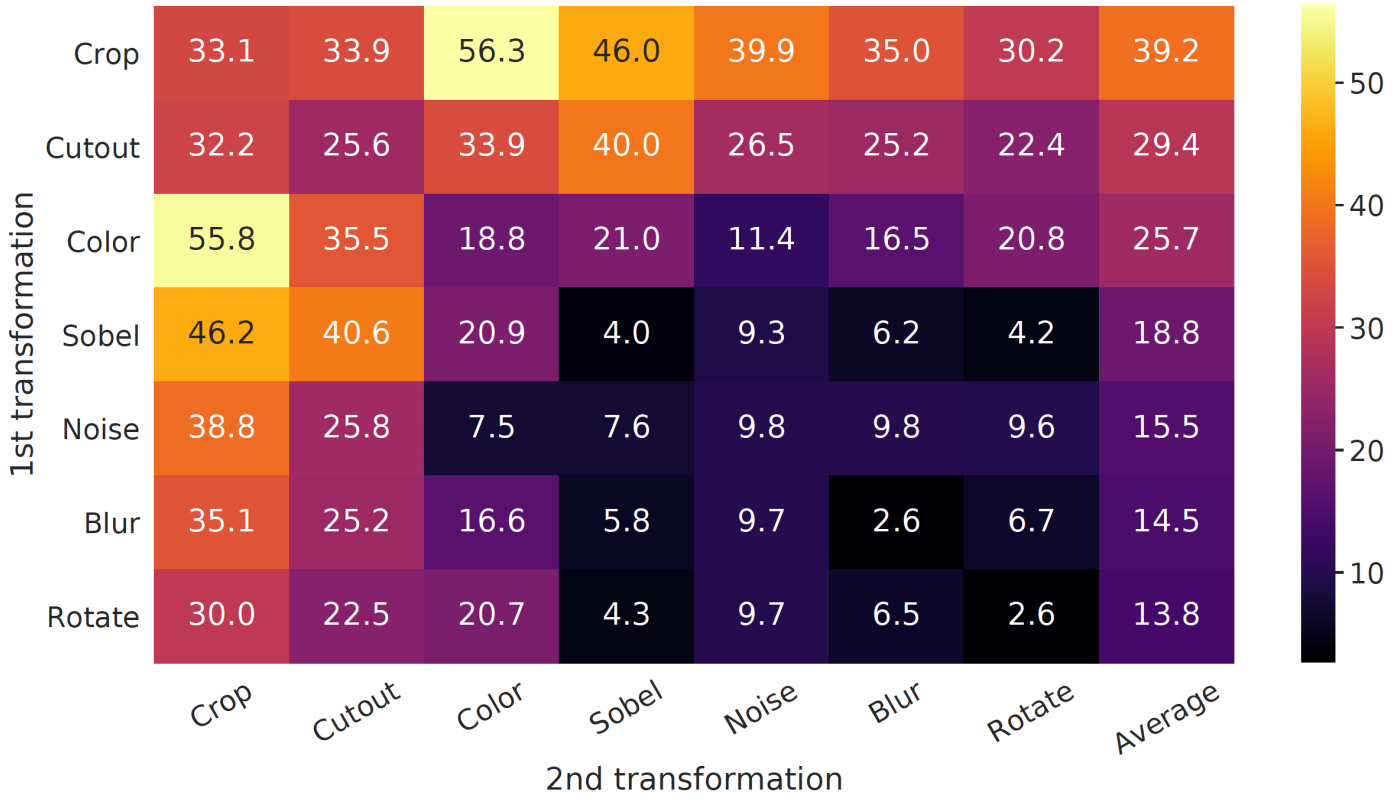


(a) Original (b) Crop and resize (c) Crop, resize (and flip) (d) Color distort. (drop) (e) Color distort. (jitter)

(f) Rotate $\{90°, 180°, 270°\}$ (g) Cutout (h) Gaussian noise (i) Gaussian blur (j) Sobel filtering

- Data augmentation operators considered by SimCLR.
- The augmentation policy used by SimCLR only includes random crop (with flip and resize), color distortion, and Gaussian blur.

Effects of Different Data Augmentation Combinations



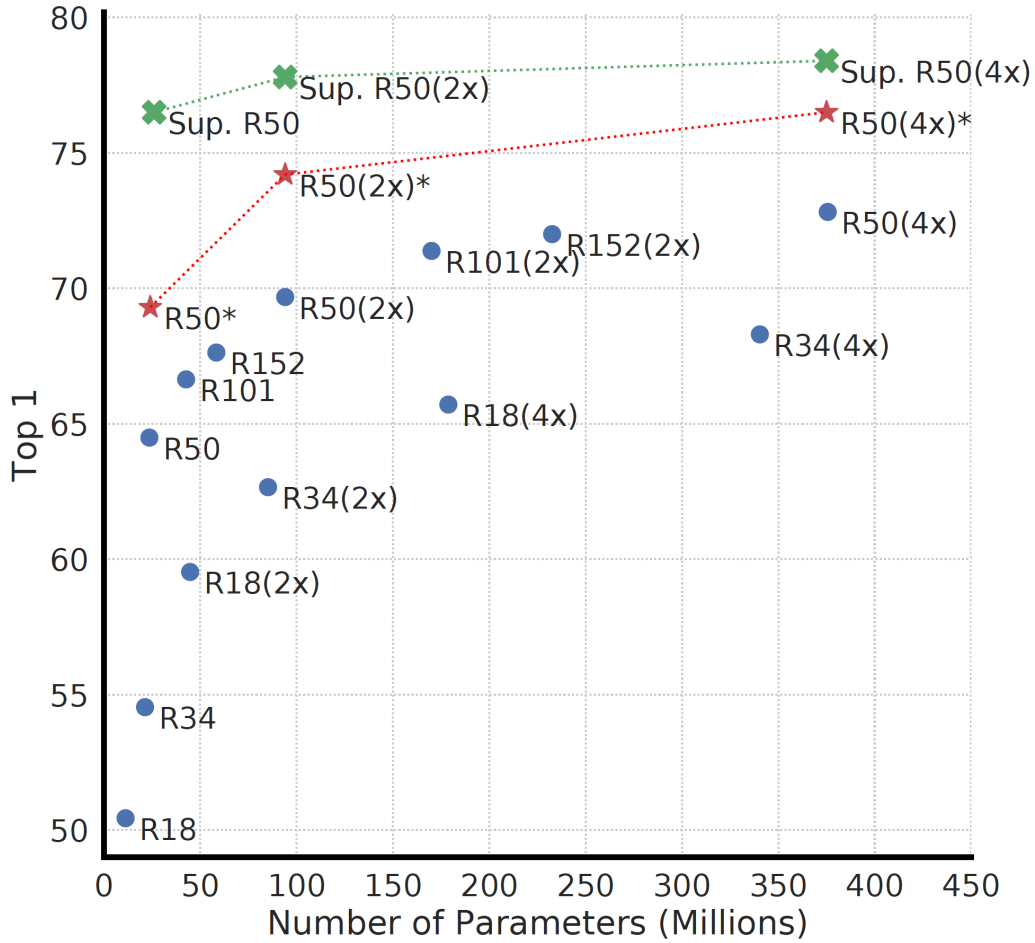
- Linear evaluation (ImageNet accuracy) under combinations of data augmentations, applied only to one branch of SimCLR.
- For all columns but the last, diagonal entries correspond to single transformation, and off-diagonals correspond to combination of two transformations (applied sequentially).
- Best combination: random cropping and random color distortion.

Benefits of Data Augmentation

Methods	Color distortion strength 1/8	1/4	1/2	1	1 (+Blur)	AutoAug
SimCLR	59.6	61.0	62.6	63.2	64.5	61.1
Supervised	77.0	76.7	76.5	75.7	75.4	77.1

- Top-1 accuracy of unsupervised ResNet-50 using linear evaluation and supervised ResNet-50
- Contrastive learning needs stronger data augmentation than supervised learning
  - Stronger color augmentation substantially improves the linear evaluation of the learned unsupervised models.
  - Stronger color augmentation does not improve or even hurts the performance of supervised learning.

Unsupervised learning benefits more from bigger models



- The gap between supervised models and linear classifiers trained on unsupervised models shrinks as the model size increases.

Compared to previous methods

Method	Architecture	Param (M)	Top 1	Top 5
Methods using ResNet-50:				
Local Agg.	ResNet-50	24	60.2	-
MoCo	ResNet-50	24	60.6	-
PIRL	ResNet-50	24	63.6	-
CPC v2	ResNet-50	24	63.8	85.3
SimCLR (ours)	ResNet-50	24	**69.3**	**89.0**
Methods using other architectures:				
Rotation	RevNet-50 (4×)	86	55.4	-
BigBiGAN	RevNet-50 (4×)	86	61.3	81.9
AMDIM	Custom-ResNet	626	68.1	-
CMC	ResNet-50 (2×)	188	68.4	88.2
MoCo	ResNet-50 (4×)	375	68.6	-
CPC v2	ResNet-161 (∗)	305	71.5	90.1
SimCLR (ours)	ResNet-50 (2×)	94	74.2	92.0
SimCLR (ours)	ResNet-50 (4×)	375	**76.5**	**93.2**

- ImageNet accuracies of linear classifiers trained on representations learned with different self-supervised methods.

ImageNet accuracy of models trained with few labels

Method	Architecture	Label fraction (Top 5) 1%	10%
Supervised baseline	ResNet-50	48.4	80.4
Methods using other label-propagation:			
Pseudo-label	ResNet-50	51.6	82.4
VAT+Entropy Min.	ResNet-50	47.0	83.4
UDA (w. RandAug)	ResNet-50	-	88.5
FixMatch (w. RandAug)	ResNet-50	-	89.1
S4L (Rot+VAT+En. M.)	ResNet-50 (4×)	-	91.2
Methods using representation learning only:			
InstDisc	ResNet-50	39.2	77.4
BigBiGAN	RevNet-50 (4×)	55.2	78.8
PIRL	ResNet-50	57.2	83.8
CPC v2	ResNet-161(∗)	77.9	91.2
SimCLR (ours)	ResNet-50	75.5	87.8
SimCLR (ours)	ResNet-50 (2×)	83.0	91.2
SimCLR (ours)	ResNet-50 (4×)	**85.8**	**92.6**

- SimCLR sampled 1% or 10% of the labeled ILSVRC-12 training datasets in a class-balanced way (12.8 and 128 images per class respectively).
- They simply fine-tune the whole base network on the labeled data without regularization.
- The supervised baseline from is strong due to intensive search of hyper-parameters (including augmentation).

SimCLR Implementations

- The official implementation of SimCLR in Tensorflow
  - https://github.com/google-research/simclr
- Official pretrained models for 1x, 2x, and 4x variants of the ResNet50 architectures
  - https://github.com/google-research/simclr#pre-trained-models-for-simclrv1
- Unofficial PyTorch implementations
  - https://github.com/leftthomas/SimCLR
  - https://github.com/Spijkervet/SimCLR

Takeaway of SimCLR

- Combination of multiple data augmentation operations is crucial in defining the contrastive prediction tasks that yield effective representations.
  - In addition, unsupervised contrastive learning benefits from stronger data augmentation than supervised learning.
- Introducing a learnable nonlinear transformation between the representation and the contrastive loss.
  - It substantially improves the quality of the learned representations.
- Representation learning with contrastive cross entropy loss benefits from normalized embeddings and an properly tuned temperature parameter.
- Contrastive learning benefits from larger batch sizes and longer training compared to supervised learning.
- Like supervised learning, contrastive learning benefits from deeper and wider networks.

References for further reading

- Chen, Ting, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. "A simple framework for contrastive learning of visual representations." In International conference on machine learning, pp. 1597-1607. PMLR, 2020.
- Grill, Jean-Bastien, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch et al. "Bootstrap your own latent: A new approach to self-supervised learning." arXiv preprint arXiv:2006.07733 (2020).
- Jing, Longlong, and Yingli Tian. "Self-supervised visual feature learning with deep neural networks: A survey." IEEE Transactions on Pattern Analysis and Machine Intelligence (2020).
- Ying, Chris, Sameer Kumar, Dehao Chen, Tao Wang, and Youlong Cheng. "Image classification at supercomputer scale." arXiv preprint arXiv:1811.06992 (2018).
- Chaudhary, Amit. "The Illustrated SimCLR Framework." https://amitness.com/2020/03/illustrated-simclr, 2020.

Any Questions?

THANK YOU
FOR YOUR
ATTENTION