Self-Supervised Learning

Siham Tabik

Dpto. Ciencias de la Computación e I.A.
Universidad de Granada
siham@ugr.es

DaSCI

UNIVERSIDAD DE GRANADA

La semana pasada

- Los bloques fundamentales de una CNN
- Mecanismos de entrenamiento y optimizaciones
- Cómo implementar una CNN con DA y TL usando `pytorch`

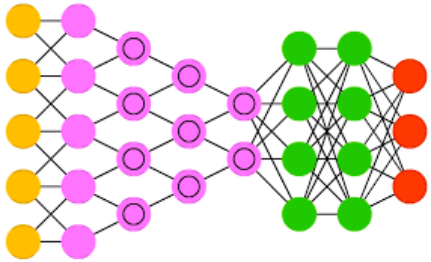
Outline

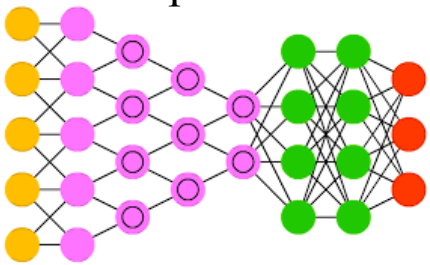
- Transfer learning
- Big transfer learning
- Foundation models
- Self-supervised learning
  - ¿Qué es?
  - Contrastive Learning
  - SimCLR
- Few-shot learning

Transfer Learning

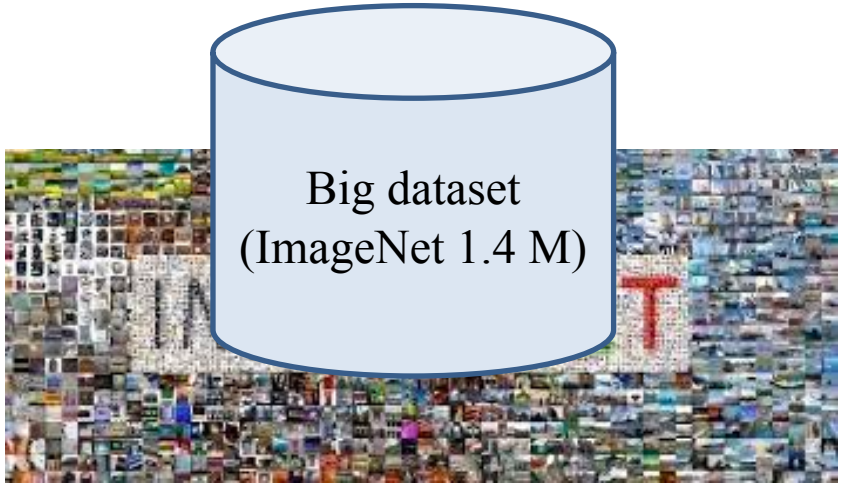
Typically, real world datasets are small

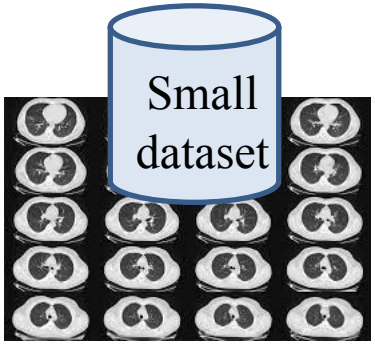
Randomly initialized weights
+ train on ImageNet





finetuning on the new data
(training for few steps)
This step does not discover features





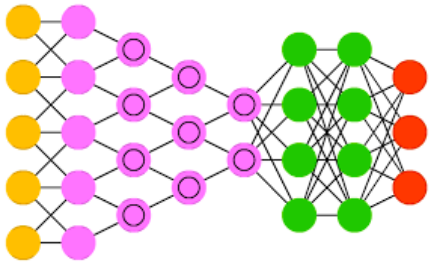
New data,
new classes

Kolesnikov, A et al. (2020). Big transfer (BiT): General visual representation learning. In *European conference on computer vision* (pp. 491-507). Springer,

Big Transfer (BiT) by Google Brain

Idea: Never train form scratch
Find a universal dataset

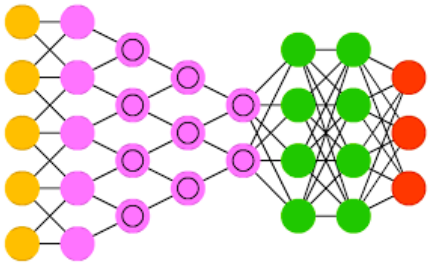
ResNet-50
Randomly initialized weights
+ retrained on JFT-300M

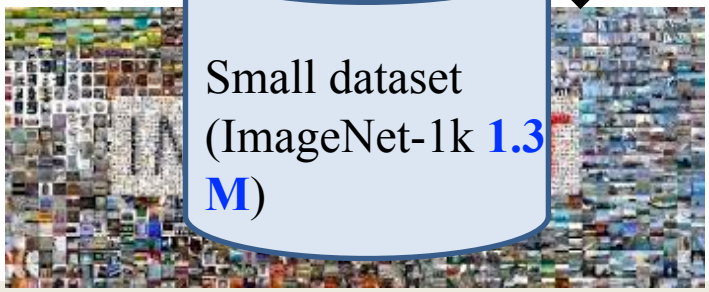


Giant dataset
(JFT-**300M**,
18k classes)
Private

Medium dataset
(ImageNet-21k **14 M**)

finetuned on the new data



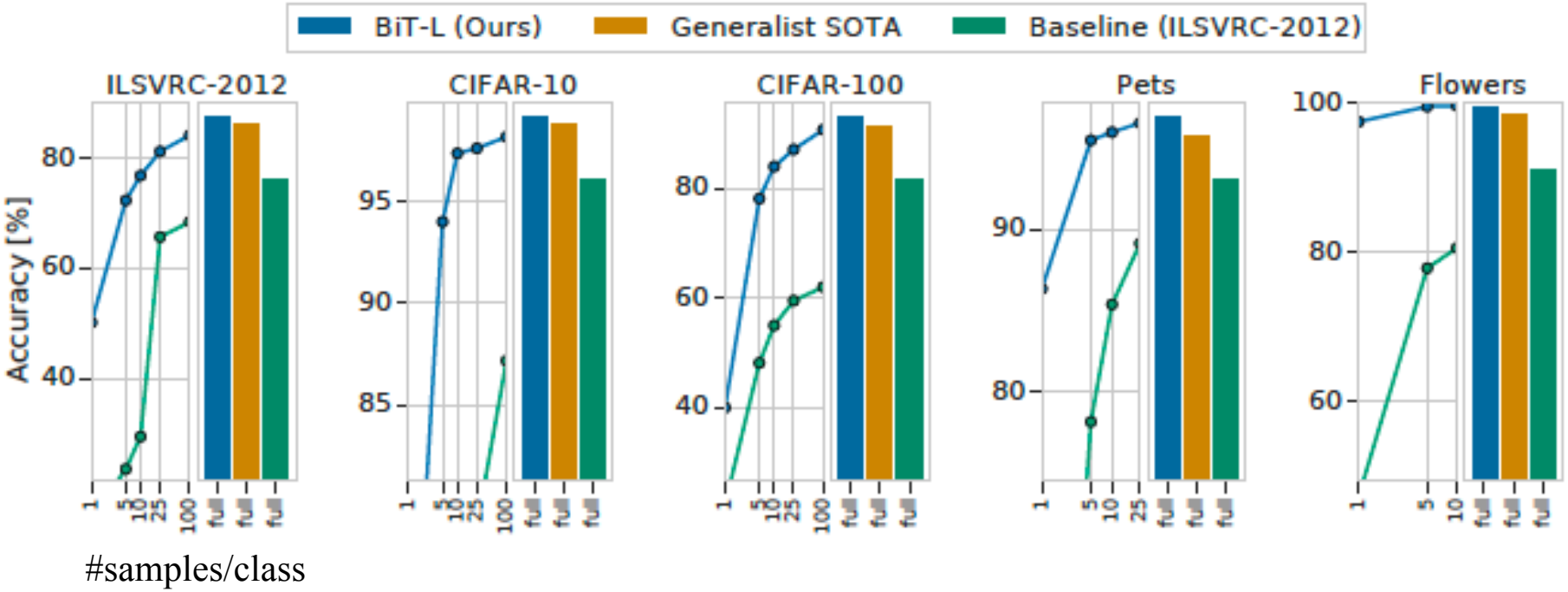


Small dataset
(ImageNet-1k **1.3 M**)

Small dataset

Big Transfer (BiT) by Google Brain

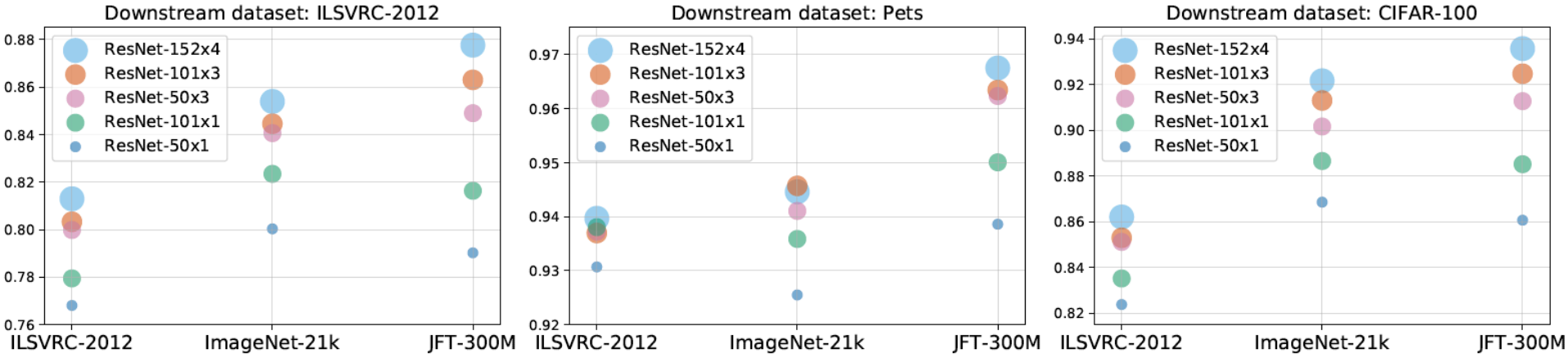
*ResNet152 pretrained on large BiT , finetuned on different tasks, CIFAR-10
*BiT-L not released



Kolesnikov, A et al. (2020). Big transfer (bit): General visual representation learning. In *European conference on computer vision* (pp. 491-507). Springer,

Big Transfer (BiT) by Google Brain

What's the impact of the size of the dataset on the size of the model?



x-axis(trained on :
Small Dataset : ILSVRC-2012/ Imagenet
Medium dataset: ImageNet-21K
Large dataset: JFT-300M, a weakly labeled dataset

The larger the pretraning-dataset, the larger the model, the larger the computational cost and CO2 emissions

Evolution & trends of DL

ML = evolution(Computational resources + data + models)

The rise of Self-supervised learning

Specialized models: One model for one task

Foundation models

NLP	RNN LSTM					**Transformers** (multi-head Attention)		BERT(Google) DALL-E GPT(OpenAI)				
CV	AlexNet		VGG	ResNet	Densenet				SimCLR			
Years	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023

Objective 1: arq with the highest accuracy

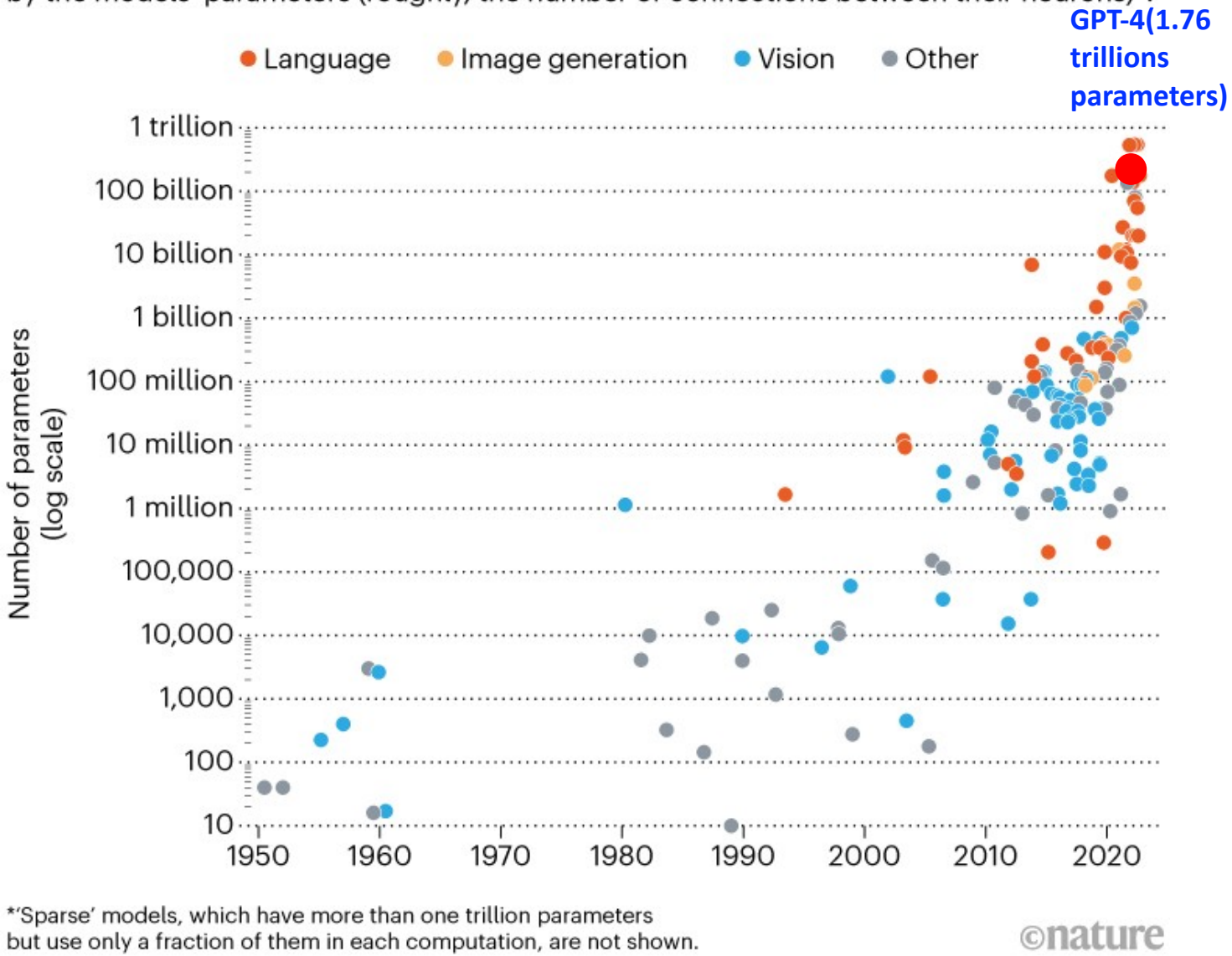
Objective 2: arq lightweight

Objective 3: general model for many apps

van den Oord, A., Li, Y., & Vinyals, O. (2018). "Representation Learning with Contrastive Predictive Coding (CPC)."

THE DRIVE TO BIGGER AI MODELS

The scale of artificial-intelligence neural networks is growing exponentially, as measured by the models' parameters (roughly, the number of connections between their neurons)*.



*'Sparse' models, which have more than one trillion parameters but use only a fraction of them in each computation, are not shown.

©nature

Sevilla, J., Heim, L., Ho, A., Besiroglu, T., Hobbhahn, M., & Villalobos, P. (2022, July). Compute trends across three eras of machine learning. In *2022 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-8). IEEE.

Environmental costs of AI models

WORLD VIEW | 20 February 2024

Generative AI's environmental costs are soaring — and mostly secret

First-of-its-kind US bill would address the environmental costs of the technology, but there's a long way to go.

By Kate Crawford

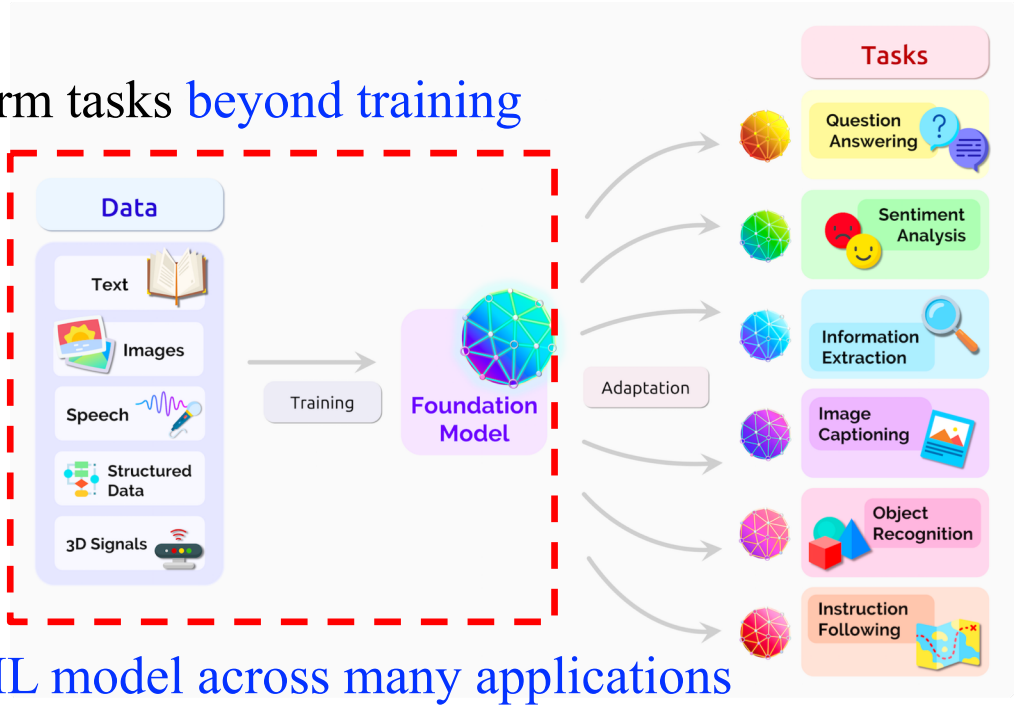
Last month, OpenAI chief executive Sam Altman finally admitted what researchers have been saying for years — that the artificial intelligence (AI) industry is heading for an energy crisis. It's an unusual admission. At the World Economic Forum's annual meeting in Davos, Switzerland, Altman warned that the next wave of generative AI systems will consume vastly more power than expected, and that energy systems will struggle to cope. "There's no way to get there without a breakthrough," he said.

What are Foundation Models?

- A foundation model is a model trained on broad data that can be adapted to a wide range of downstream tasks. Examples: BERT, GPT-3, CLIP
- The used technology is not new:

  Self-supervised learning, neural networks and transfer learning
- What is new?

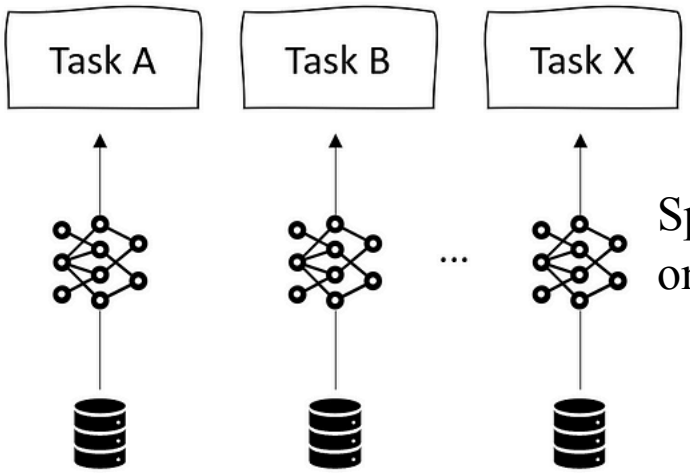
  Scale and ability to perform tasks beyond training



Bommasani R. Et al (2021) On the Opportunities and Risks of Foundation Models. https://arxiv.org/abs/2108.07258

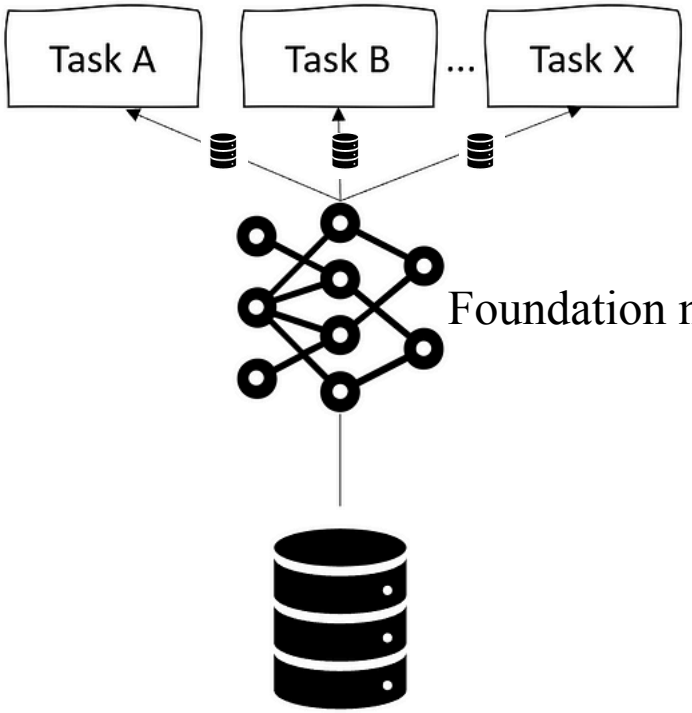
Consolidation of a homogenous ML model across many applications

How Foundation Models are built?



Specialized models:
one model/one task

Supervised
Learning



Foundation mode

Self-Supervised
Learning

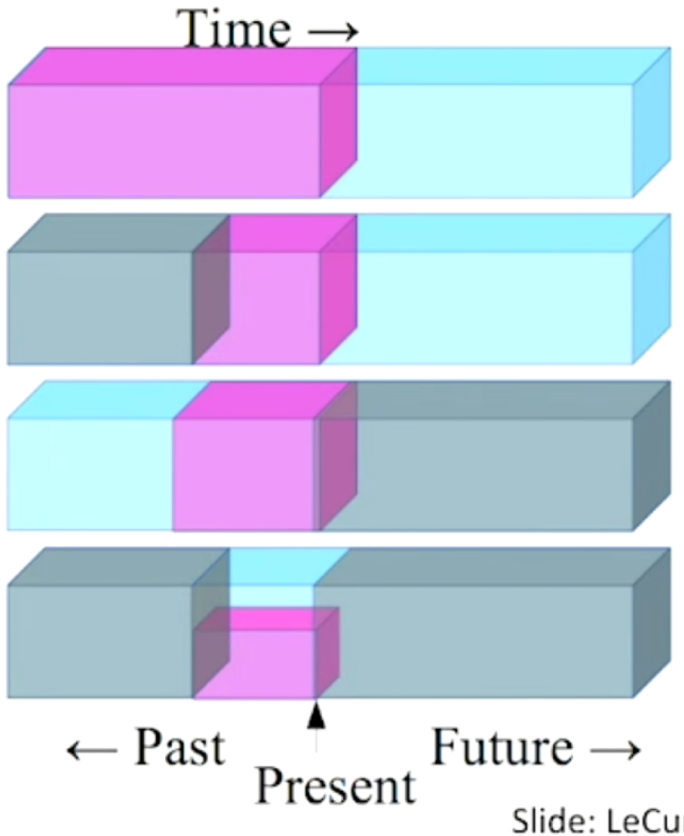
Example of foundation models in CV: Dino_v2(), SAM(Meta), GPT-3 and 4, BERT(Google), CLIP(OpenAI), DALL-E(OpenAI)

Why Self-Supervised Learning?

- Supervised Learning is costly, need a **large amount** of **labeled** data and **computational resources**
- All current AI systems use some form of **self-supervised learning**
- They do not need (Input, Output) like supervised learning
- They do not need (Input, Reward) like in reinforcement learning
- They train the system to model/represent the Input (the data without any human feedback)
- SSL reconstructs the input or missing parts of the input
- SSL is a special type of representation learning that enables learning good representation from unlabeled dataset
- It is motivated by the idea of constructing supervised learning tasks out of unsupervised datasets.

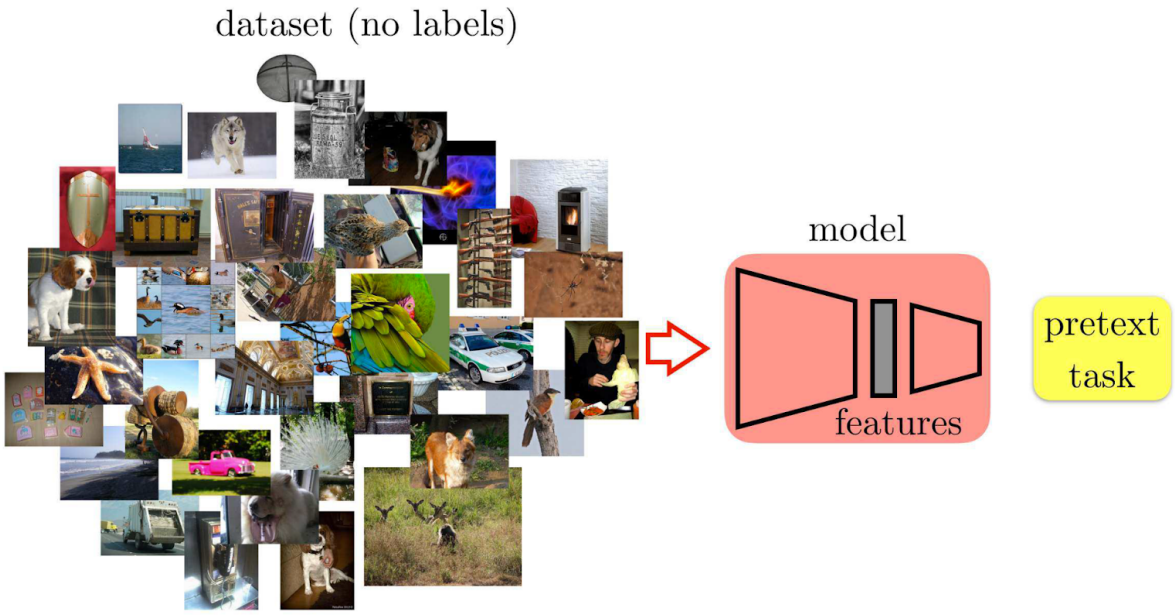
SSL=Training the model to model the input

- Predict any part of the input from any other part.
- Predict the future from the past.
- Predict the future from the recent past.
- Predict the past from the present.
- Predict the top from the bottom.
- Predict the occluded from the visible
- Pretend there is a part of the input you don't know and predict that.

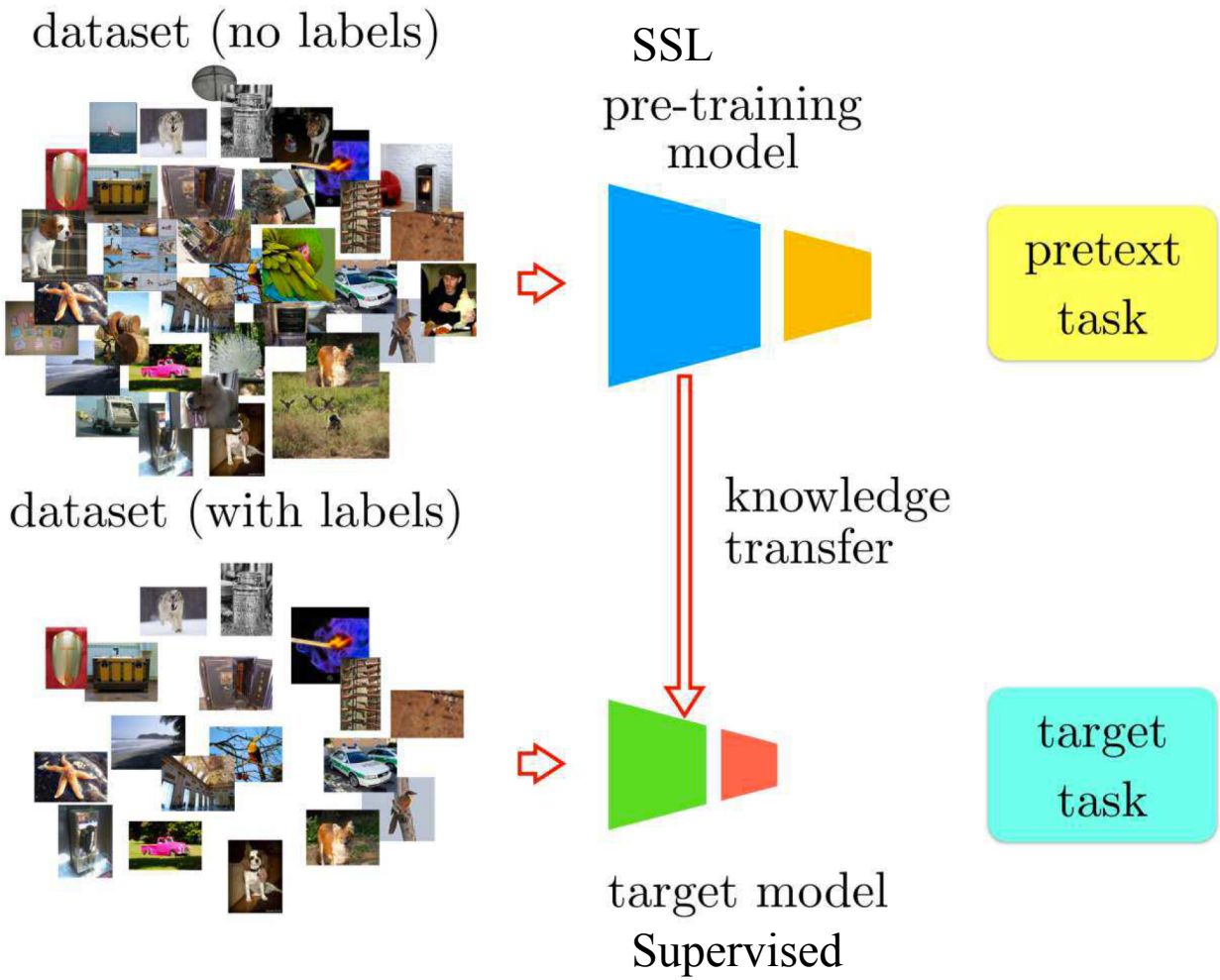


Slide: LeCun

SSL : Pretext model

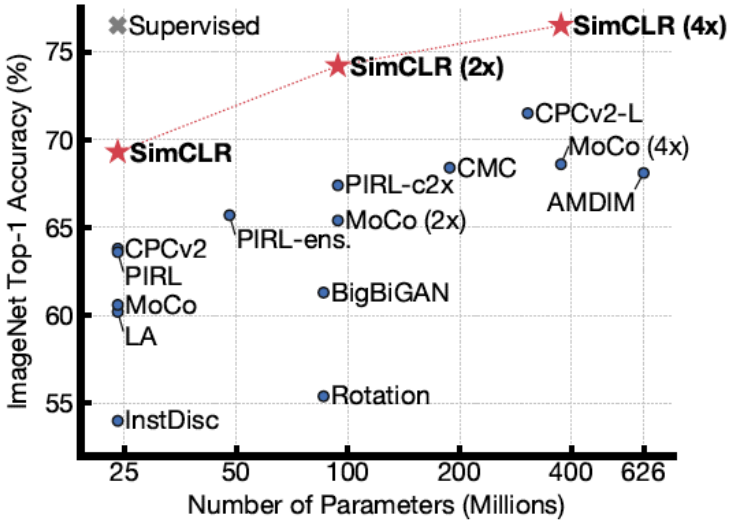


SSL : Pretext & downstream models



Self-supervised Learning

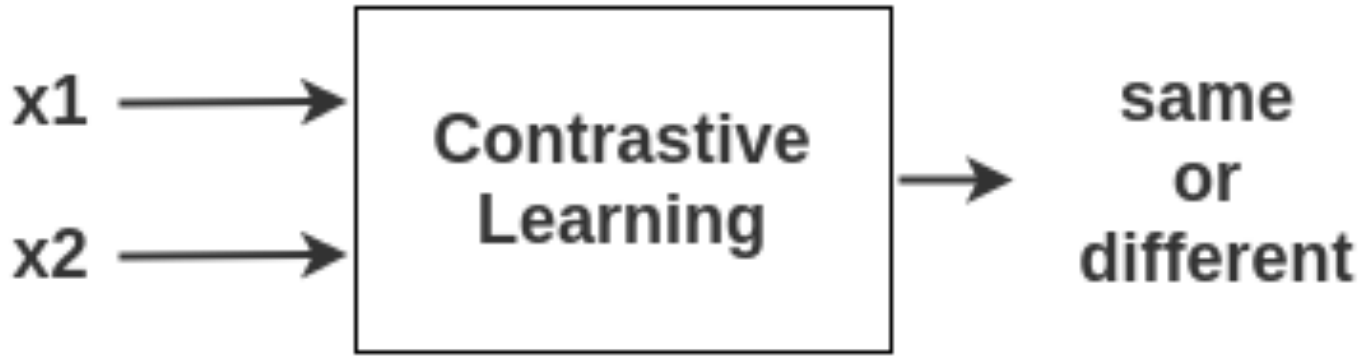
- First work on NNs that learn from raw data “learning meaningful representations from raw data without labeled supervision.” Hinton, G. E., & Zemel, R. S. (1994). "Autoencoders, minimum description length, and Helmholtz free energy.”
- Oord, A. V. D., Li, Y., & Vinyals, O. (2018). Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748.
- In recent years, numerous self-supervised learning methods have been proposed for learning image representations, each getting better than the previous. But, their performance was still below the supervised counterparts.
- Types: **Constrastive**, Non-constrastive, generative
- **Chen et. al (2020)** proposed a new framework in their research paper “SimCLR: A Simple Framework for Contrastive Learning of Visual Representations”. The SimCLR paper not only improves upon the previous state-of-the-art self-supervised learning methods but also beats the supervised learning method on ImageNet classification when scaling up the architecture.



Contrastive Learning

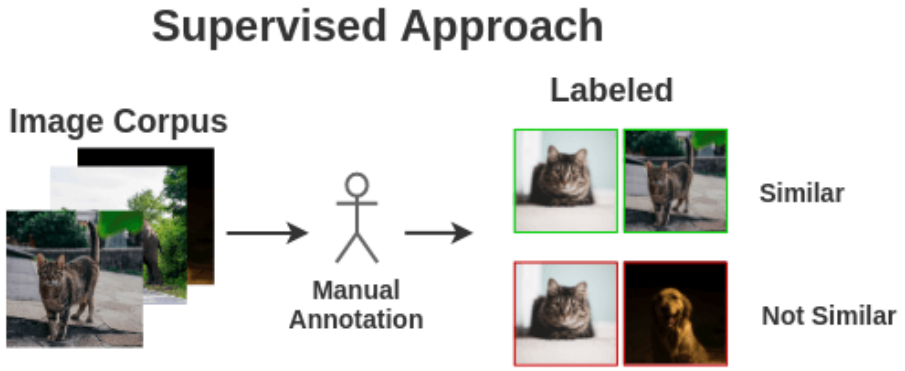
Key Idea:

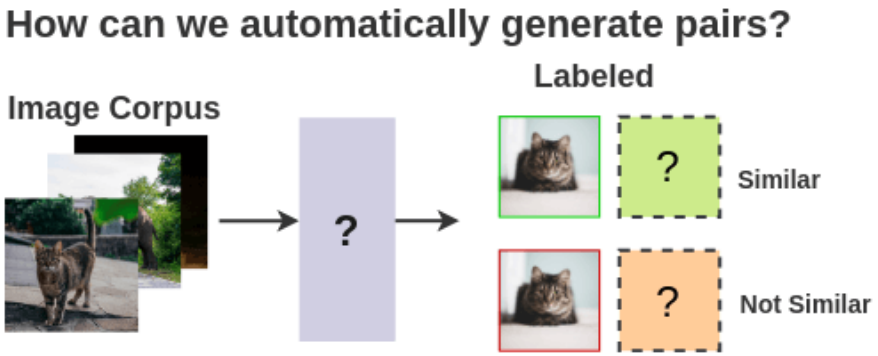
The model learns to distinguish whether x1 and x2 are similar of different



Problem formulation

Need similar and different examples



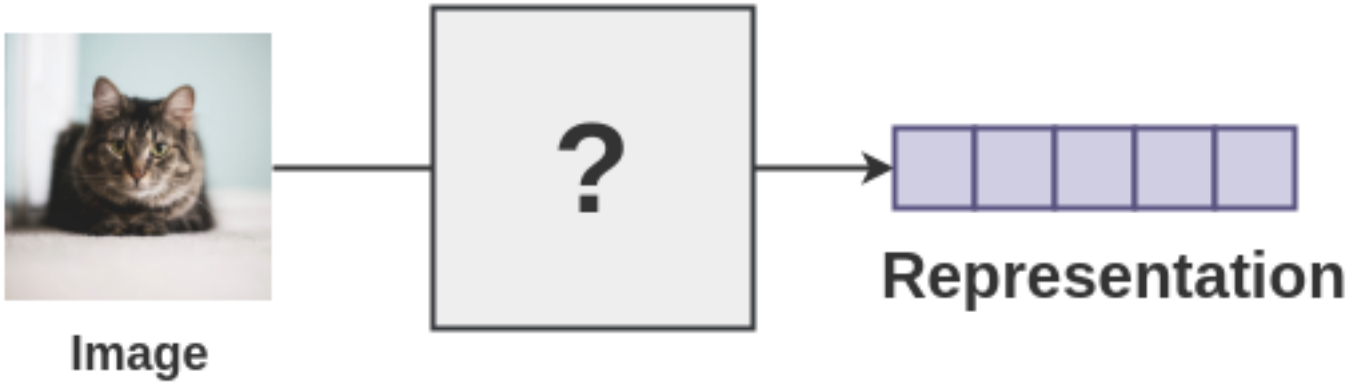


Data-augmentation

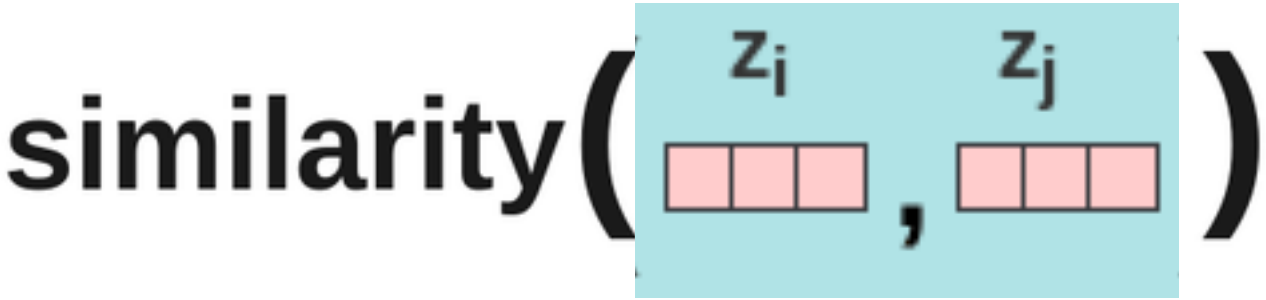
Problem formulation

The model needs:

- Ability to know what an image represents

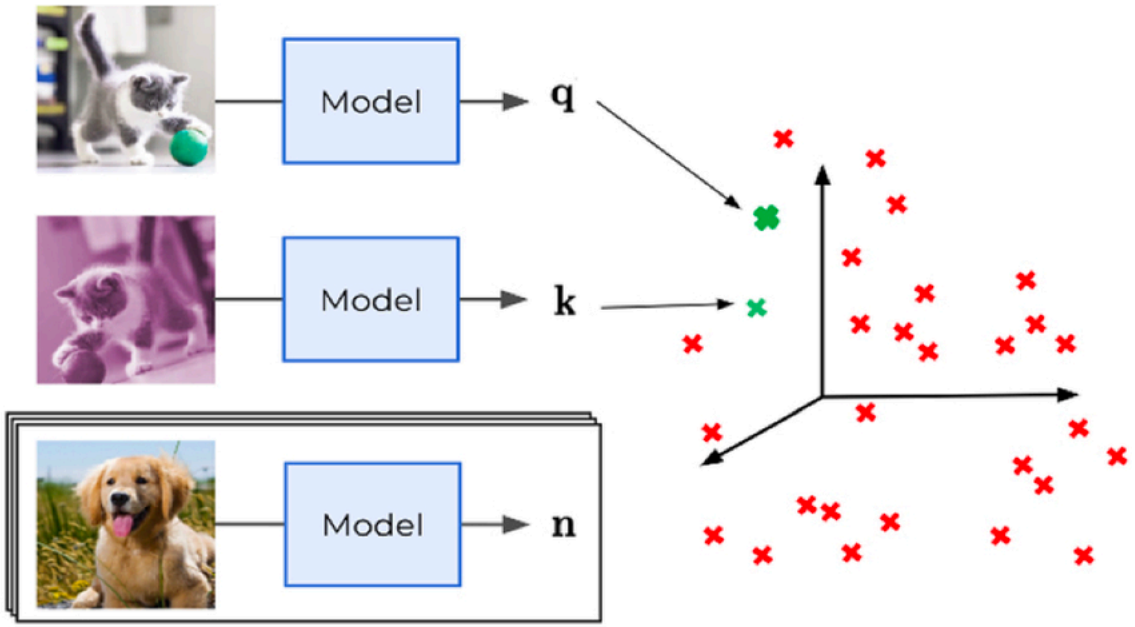


- Ability to quantify if two images are similar



Problem reformulation

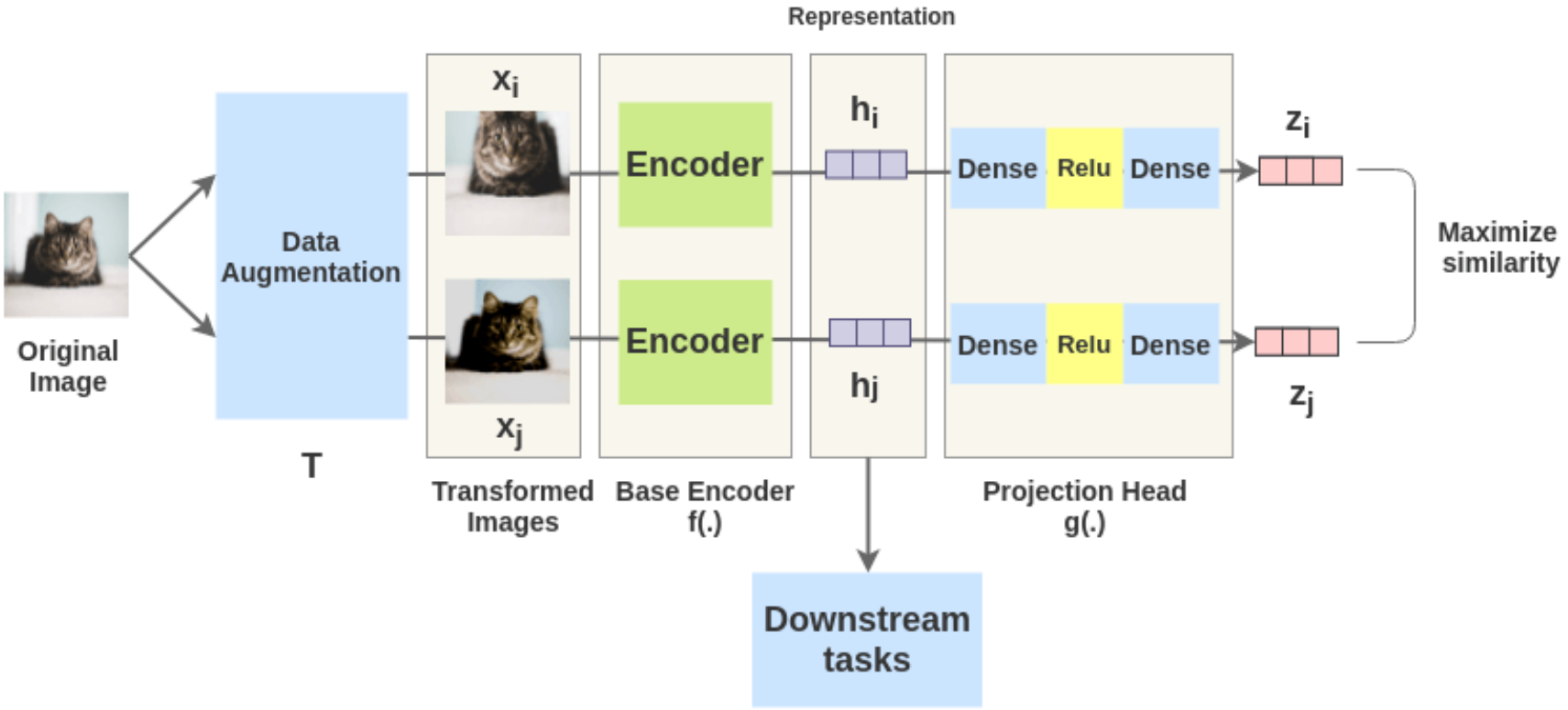
- Encoder/CNN maps images to embeddings in a low-dimensional space.
- Anchor sample, positive and negative



SimCLR framework

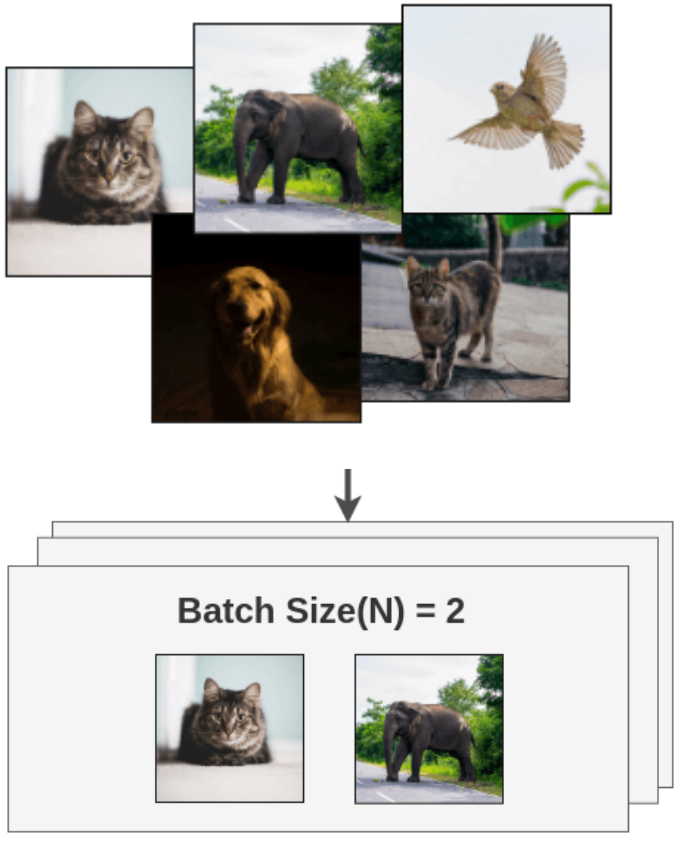
Main components:

Encoder, projection head, contrastive loss



Step by step example

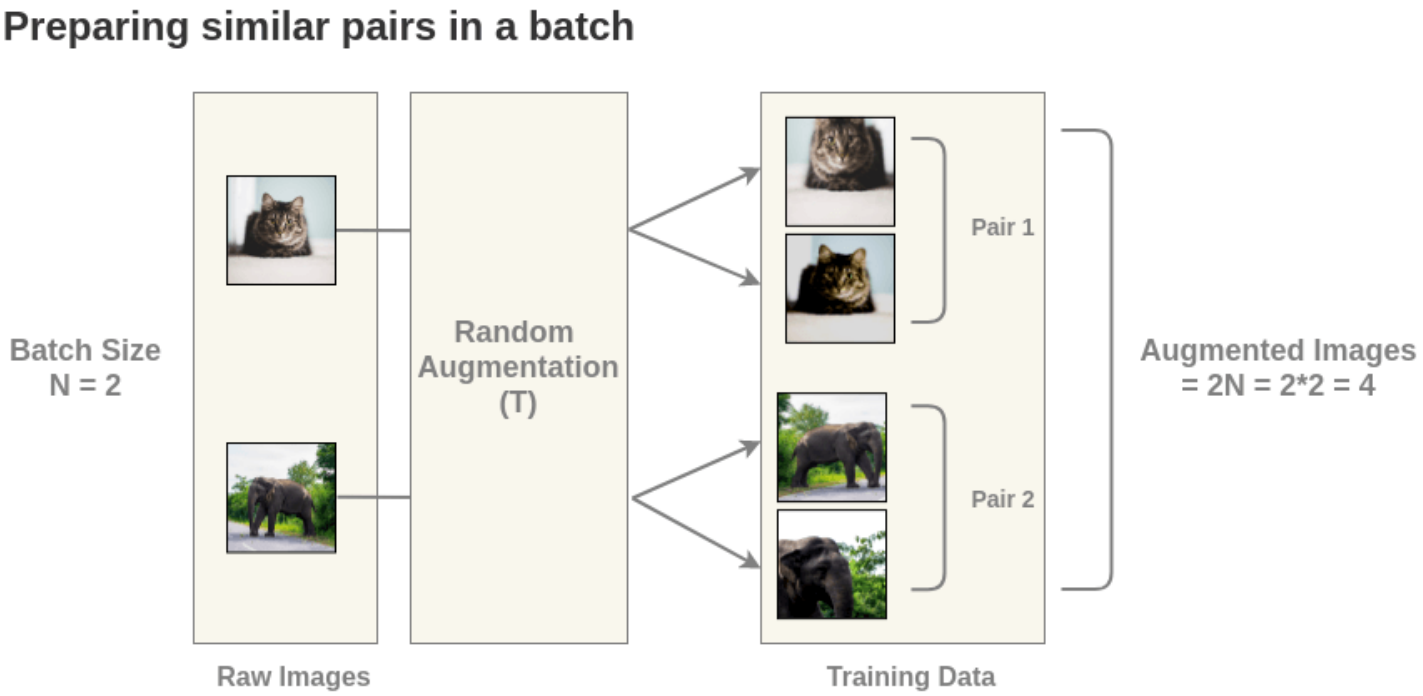
Suppose we have millions of unlabeled training dataset



Let’s generate a batch of size N of raw images. For simplicity, let’s take N=2

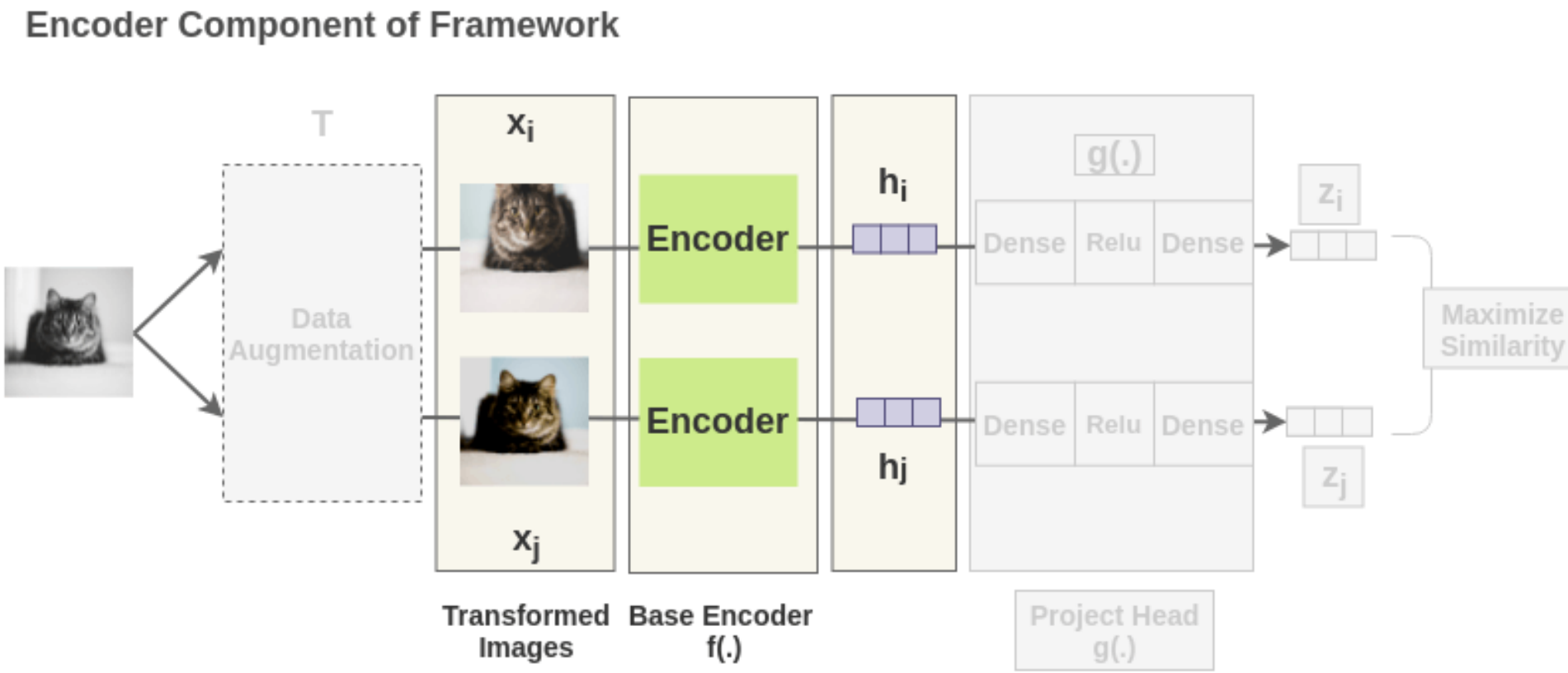
1. Data augumentation

Apply function T = a combination of random (crop + flip + color jitter + grayscale)



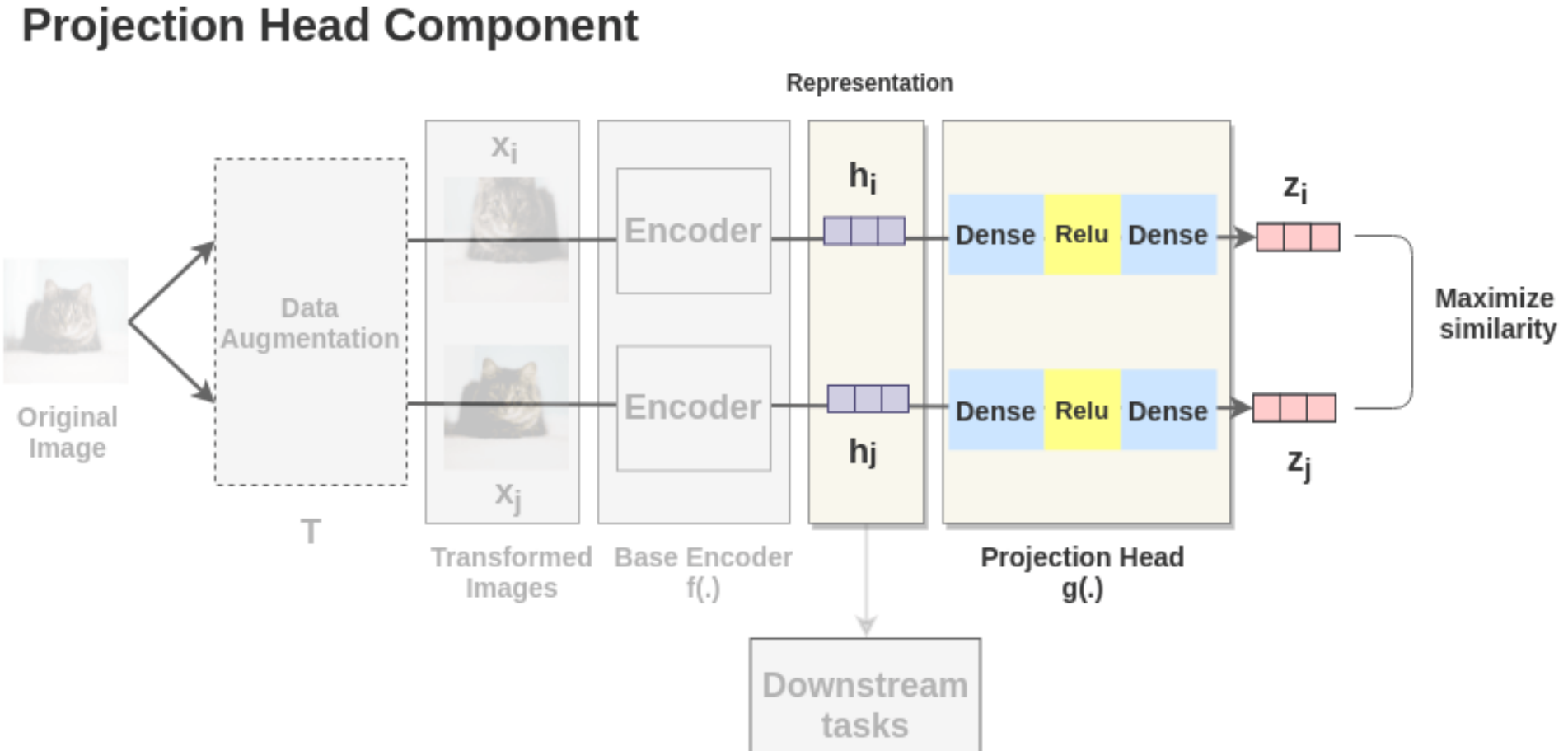
2.Getting representations

Each augmented image in a pair is passed through an encoder to get image representations. The encoder used is generic and replaceable with other architectures. The two encoders shown below have shared weights and we get vectors hi and hj.



3. Proyection head

The representations hi and hj of the two augmented images are then passed through a series of non-linear **Dense -> Relu -> Dense** layers to apply non-linear transformation and project it into a representation zi and zj. This is denoted by g(.)

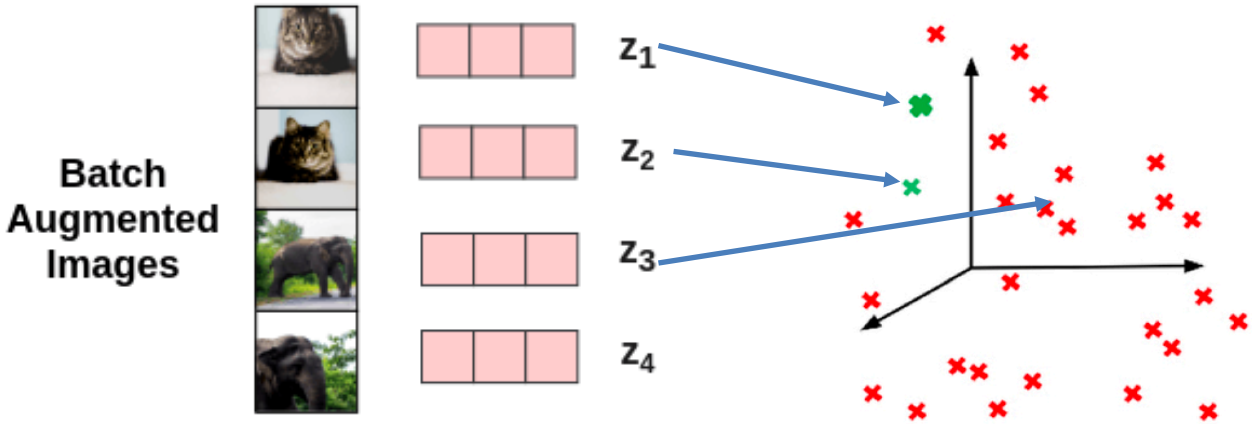


4. Tuning the model

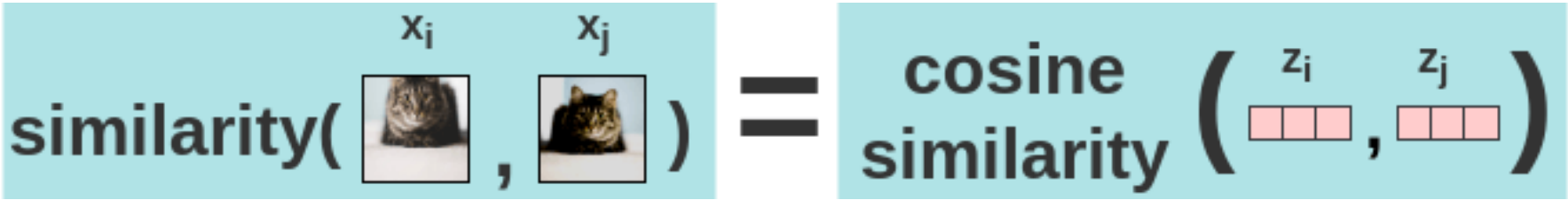
[Bringing similar closer]

a. Calculation of Cosine Similarity

For each augmented image in the batch, we get embedding vectors z for it.



From these embedding, we calculate the loss in following steps:

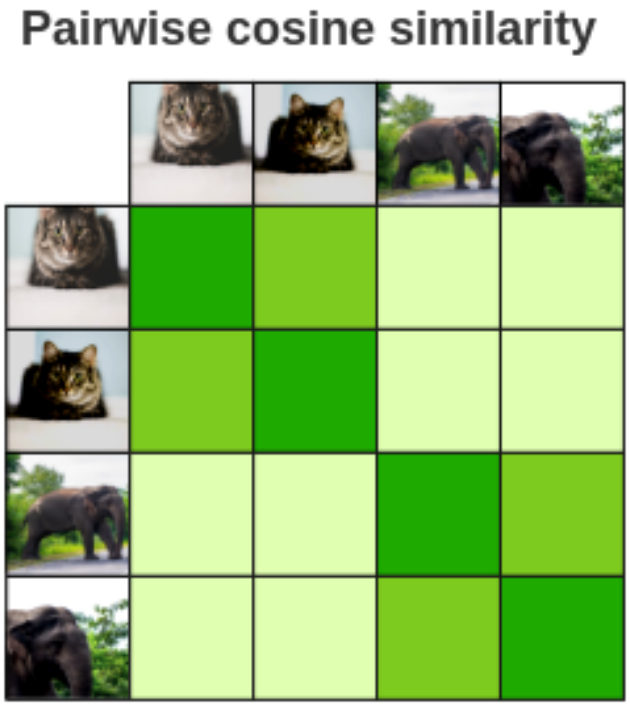


$$s_{i,j} = \boldsymbol{z}_i^\top \boldsymbol{z}_j / (\|\boldsymbol{z}_i\| \|\boldsymbol{z}_j\|) \quad \text{\# pairwise similarity}$$

4. Tuning the model

[Bringing similar closer]

The pairwise cosine similarity between each augmented image in a batch is calculated. As shown in the figure, in an ideal case, the similarities between augmented images of cats will be high while the similarity between cat and elephant images will be lower.

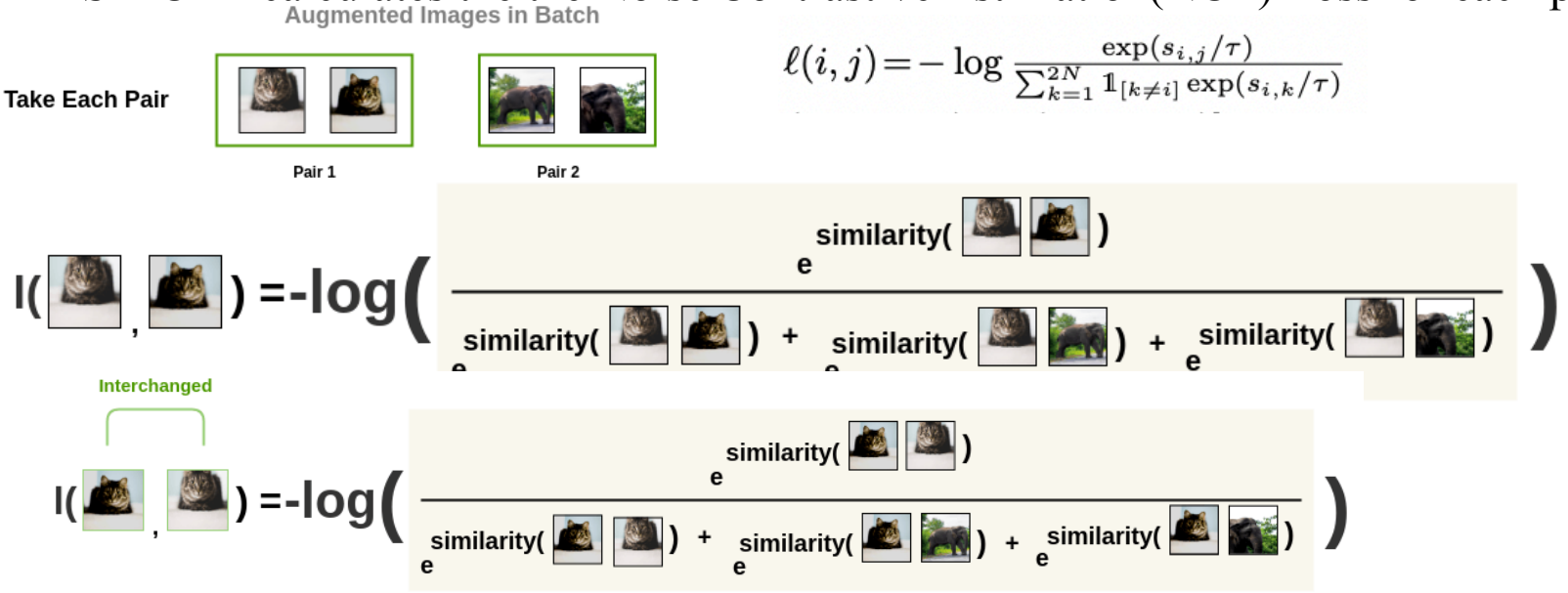


4. Tuning the model

[Bringing similar closer]

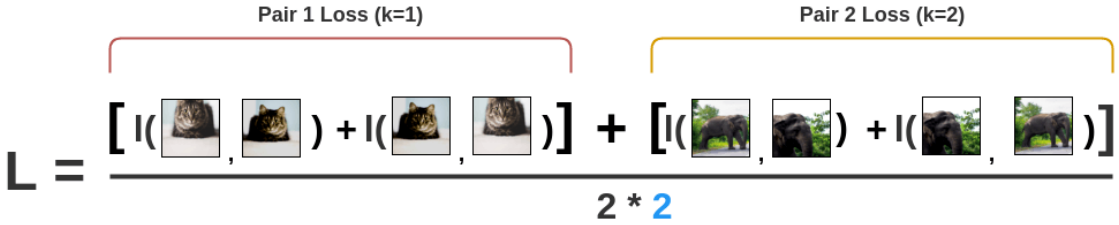
Contrastive Loss calculation

- SimCLR calculates the the Noise Contrastive Estimation(NCE) Loss for each pair



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

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



SimCLR

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 $g(\cdot)$

Cosin.similarity

NT-Xent loss" (Normalized Temp
Scaled Cross-Entropy Loss

The Contrastive Loss Function

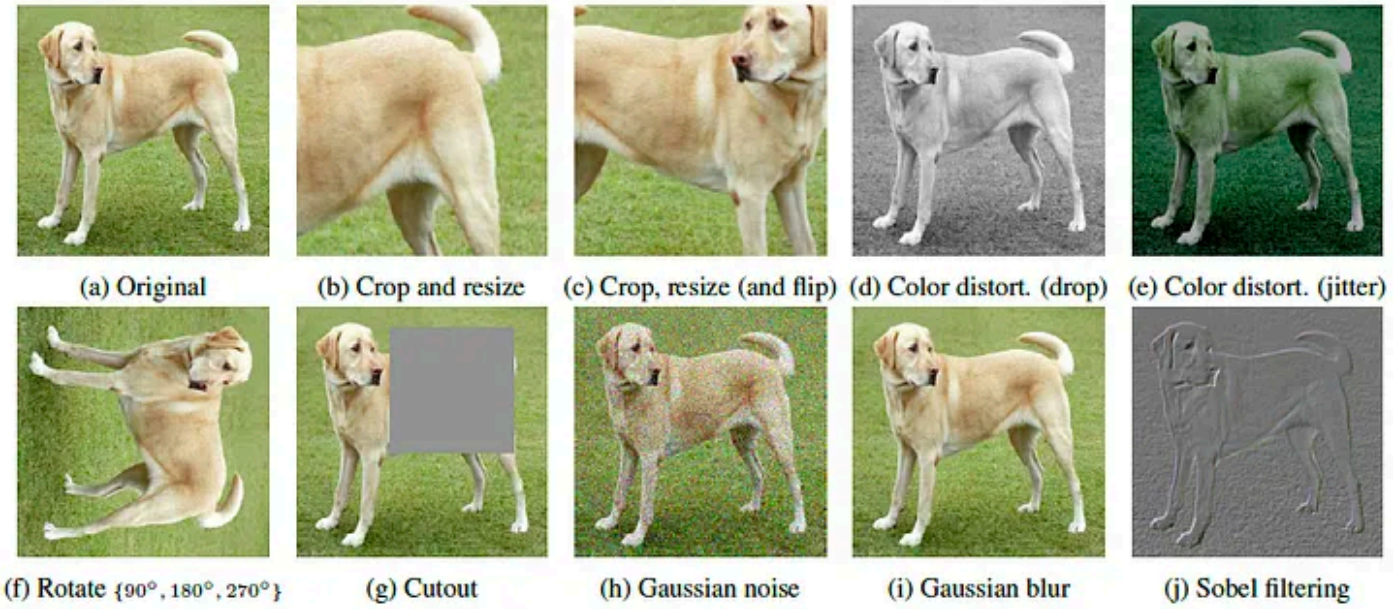
For a positive pair (i,j):

Similarity of positive pairs

$$\ell_{i,j} = -\log \frac{\exp\left(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau\right)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp\left(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau\right)}$$

Similarity of negative pairs

Data augmentation for Contrastive learning

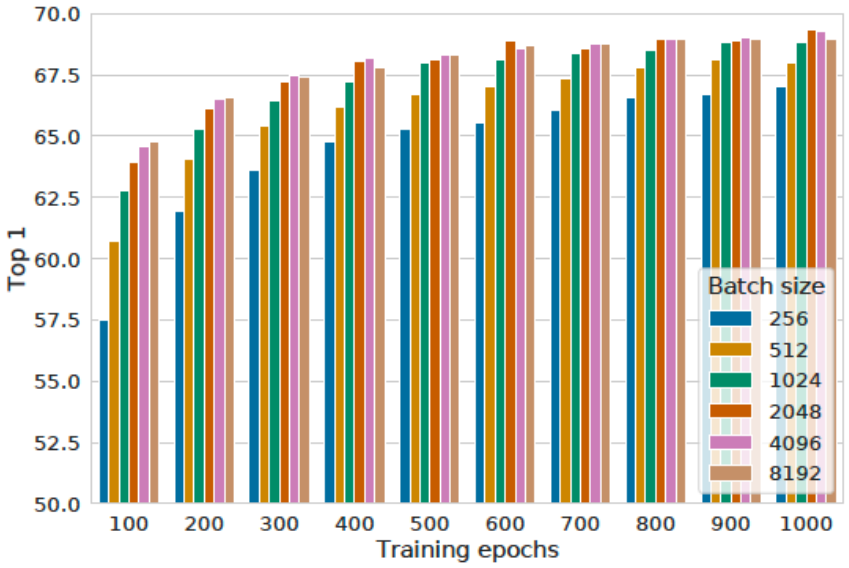


Contrastive learning needs stronger data augmentation than supervised learning

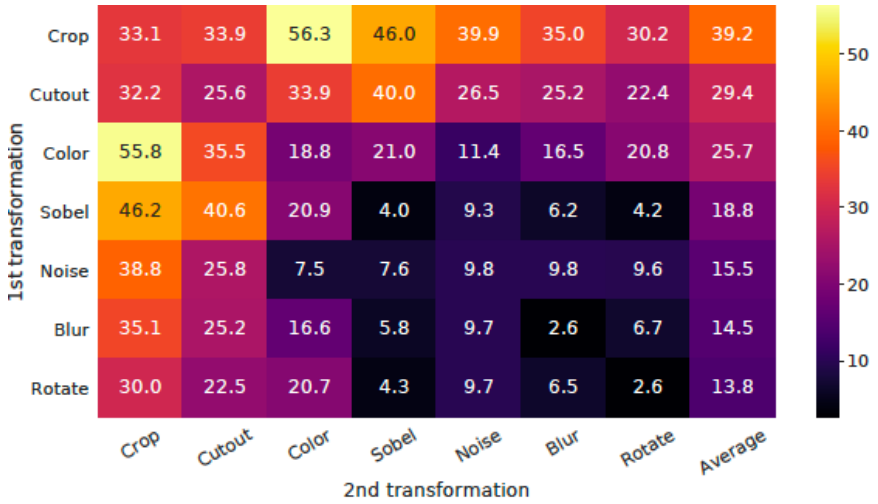
SimCLR

SimCLR is sensitive to

- batchsize and



- to the set of applied data-augumentations



Finetuning from supervised versus Finetuning from SimCLR

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.

Outline

- Transfer learning
- Big transfer learning
- Foundation models
- Self-supervised learning
  - ¿Qué es?
  - Contrastive Learning
  - SimCLR
- **Few-shot learning**

N-shot learning

- The need for models able to quickly generalize to new categories
- N-shot learning uses transfer learning and meta-learning methods to train models recognize new clases
- Thee types: few-shot, one-shot and zero-shot learning

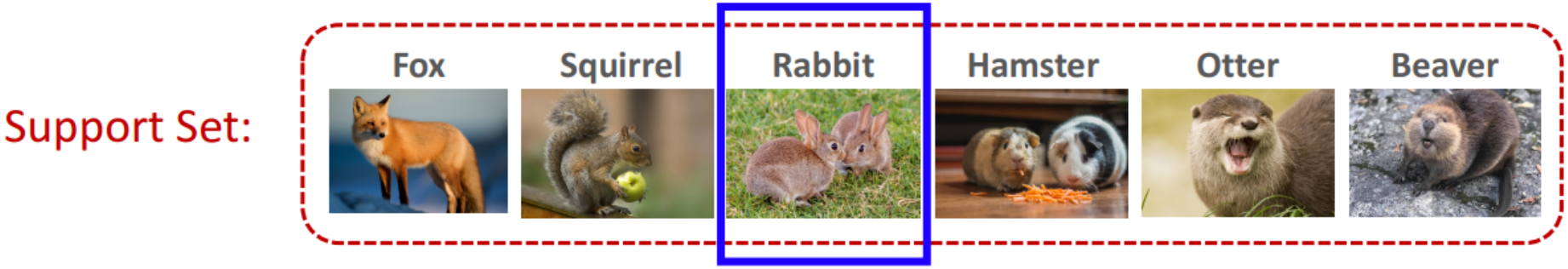
Few-shot learning

- Making classification based on a very small number of examples

Supervised learning verus few-shot learning

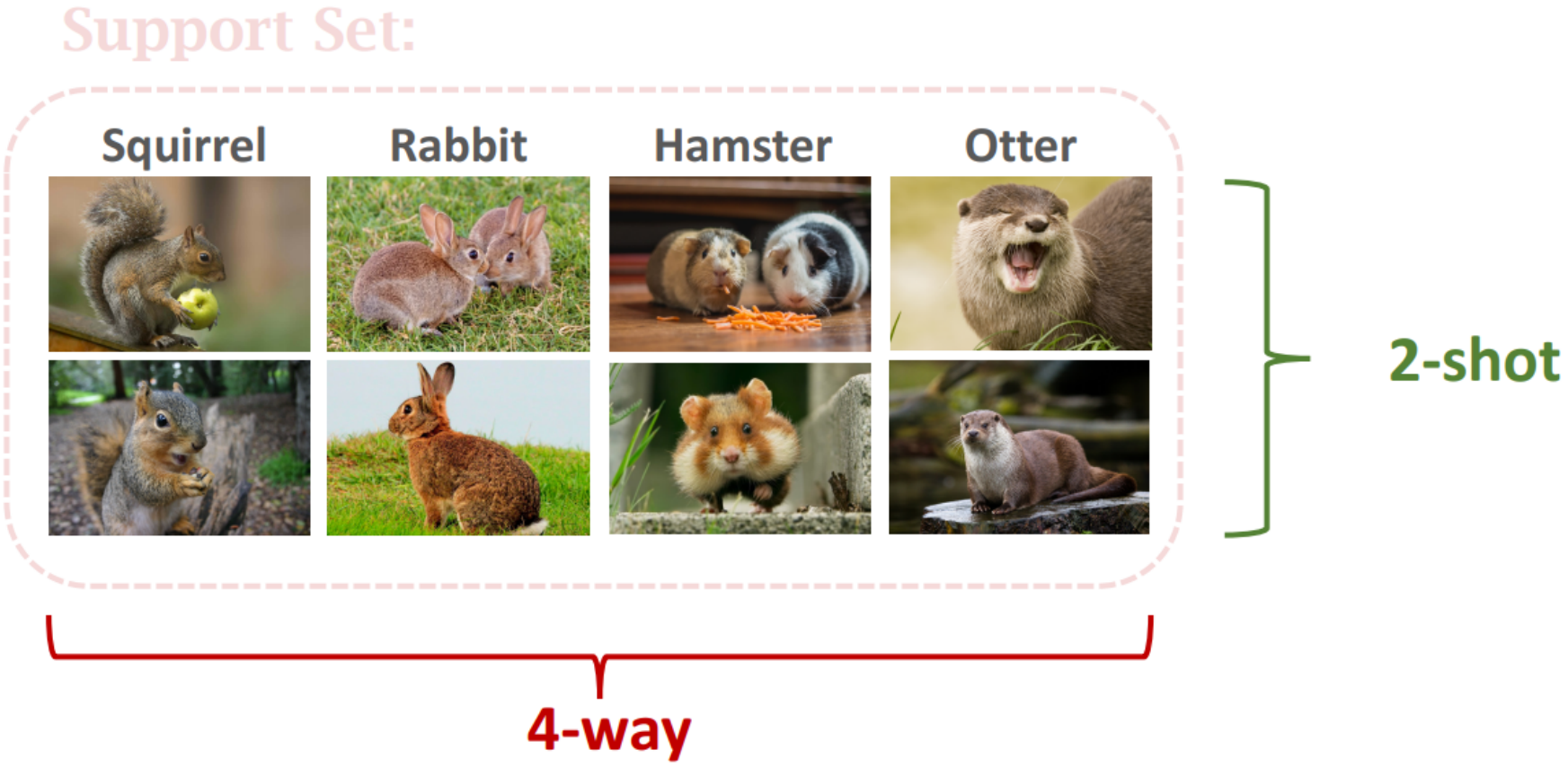
Training set, support set and query

- Query are samples never seen before, the query simple is from unknown class



k-way, *n*-shot, support set

- *k*-way: the support set has *k* clases
- *n*-shot: every class has *n* samples



¿How to find determine the query class?

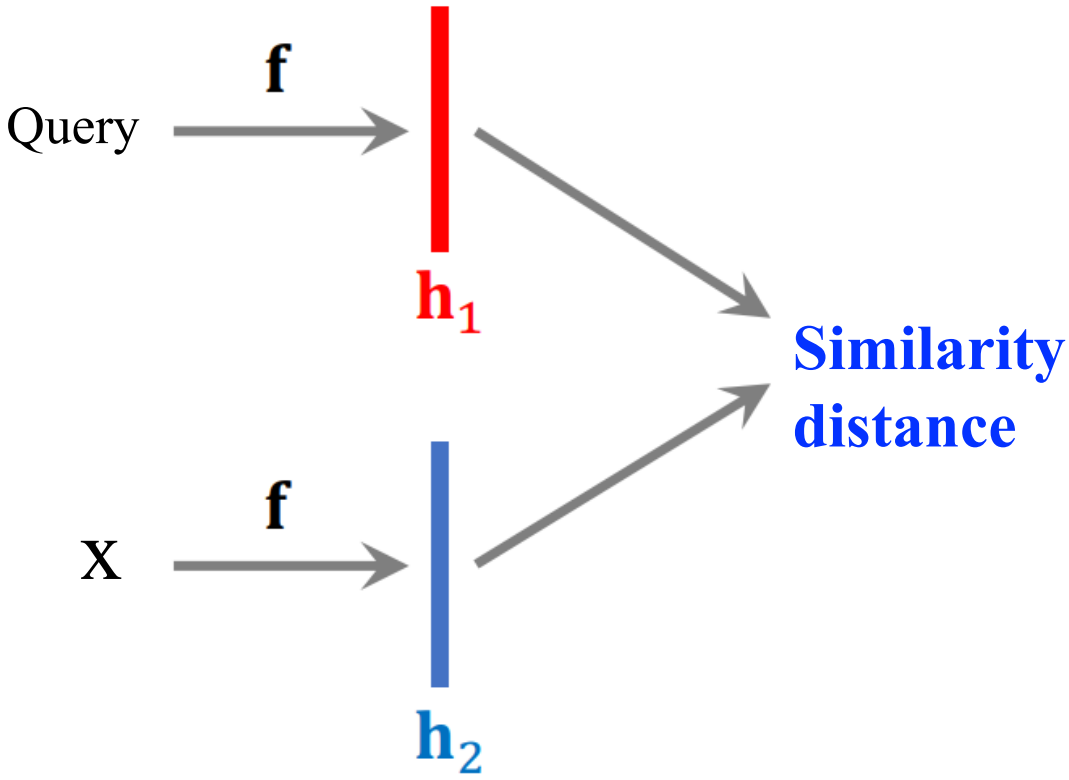
One-Shot Prediction

Query:

sim = 0.2	sim = 0.9	sim = 0.7	sim = 0.5	sim = 0.3	sim = 0.4
Fox	Squirrel	Rabbit	Hamster	Otter	Beaver
					

How to predict similarity?

Train a model to predict similarity function:



Tarea

Laboratorio de clasificación de imágenes usando Deep Learning

Este laboratorio se centra en crear un sistema automático de clasificación de imágenes de hormigas y abejas. Para ello se necesita descargar la base de datos pequeña llamada "hymenoptera_data" a través de este link:

https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.htm

Tenemos unas 120 imágenes por clase para entrenamiento y 75 imágenes por clase para test.

Selecciona una de las redes más influyentes, por ejemplo, ResNet, EfficientNet … etc, para la clasificación de esta BD y resume en una tabla los resultados en términos de recall, precisión y F1_score aplicando:

1. *Full finetuning* de toda la red con y sin *transfer learning* de ImageNet.
2. *Linear probing* congelando el extractor de característica y entrenando sólo las últimas FC.
3. *Linear probing* más 3 transformaciones de *data-augumentation*.
4. Entrenando el modelo durante 5 epocas usando linear probing y 3 epocas usando *Finetuning*

Se entregará un código .ipynb, con los puntos 1, 2, 3 y 4, incluyendo en un celda de texto la tabla mencionada anteriormente. Se recomienda comentar bien cada parte.