Learning from Unlabeled Data

Guest lecture by Aruni RoyChowdhury

Today’s Class

- Recap
  - Supervised vs Unsupervised Learning
  - Why not always label data?
- Semi-supervised Learning
  - Concepts
  - Example: pseudo-labels / self-training
  - Example: Distillation, Student/Teacher
- Self-supervised Learning
  - Concepts
  - Pretext tasks
  - Contrastive Learning

Today's Class

- Recap
  - Supervised vs Unsupervised Learning
  - Why not always label data?
- Semi-supervised Learning
  - Concepts
  - Example: pseudo-labels / self-training
  - Example: Distillation, Student/Teacher
- Self-supervised Learning
  - Concepts
  - Pretext tasks
  - Contrastive Learning

Recap: Supervised vs Unsupervised Learning

Supervised Learning

Data: (X, y)

X = input/feature/image/…

y = label/target

→ Cat

→ Dog

Unsupervised Learning

Data: X

Just X, no labels

Learn about the *structure* of the data, i.e. P(X)

…..

So let's always use Supervised Learning?

Supervised Learning

Data: (X, y)

X = input/feature/image/…

y = label/target

→ Cat

→ Dog

Cookie cutter Supervised Learning:

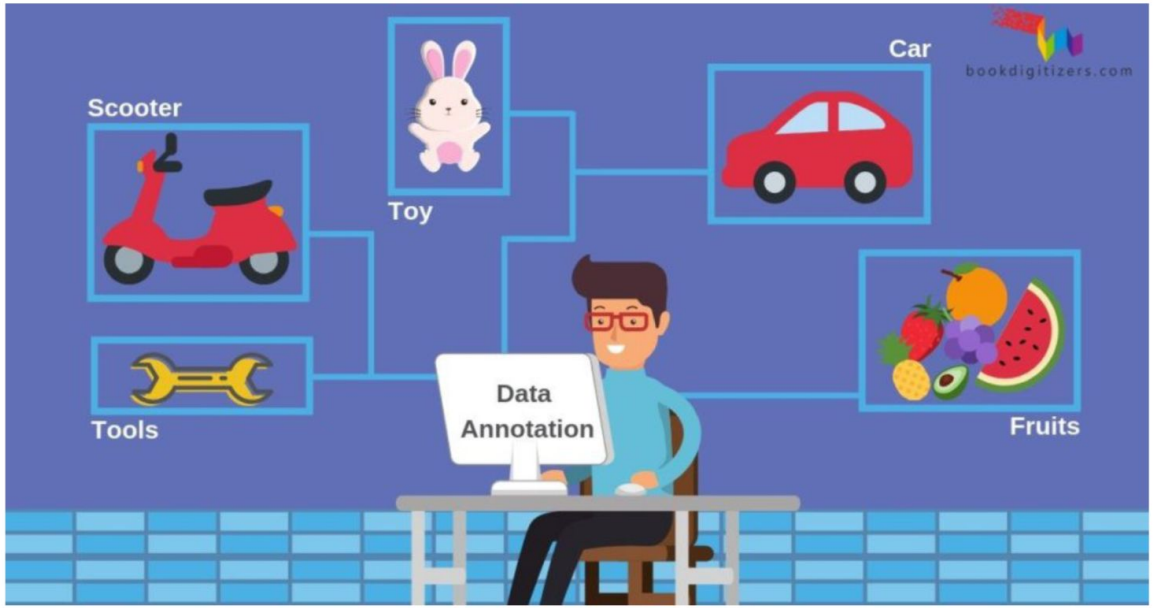
1. Collect a large set of data (images..) as the "training set"
2. Label each one as cat / dog / monkey / …
3. Train a model mapping image to label

$$f : \mathbf{X} \rightarrow y$$

4. Go forth and classify the world with f !

Data Annotation

Supervised Learning first requires labeling a very large amount of data



Slides from Andreas Geiger, MPI Tubingen

Labeling image categories - “easy” until ….

- Over **120 dog categories** in ImageNet dataset for image classification
- Non-expert human labelers may not be aware of these **fine-grained** differences, leading to ***labeling errors***

Slides from Andreas Geiger, MPI Tubingen

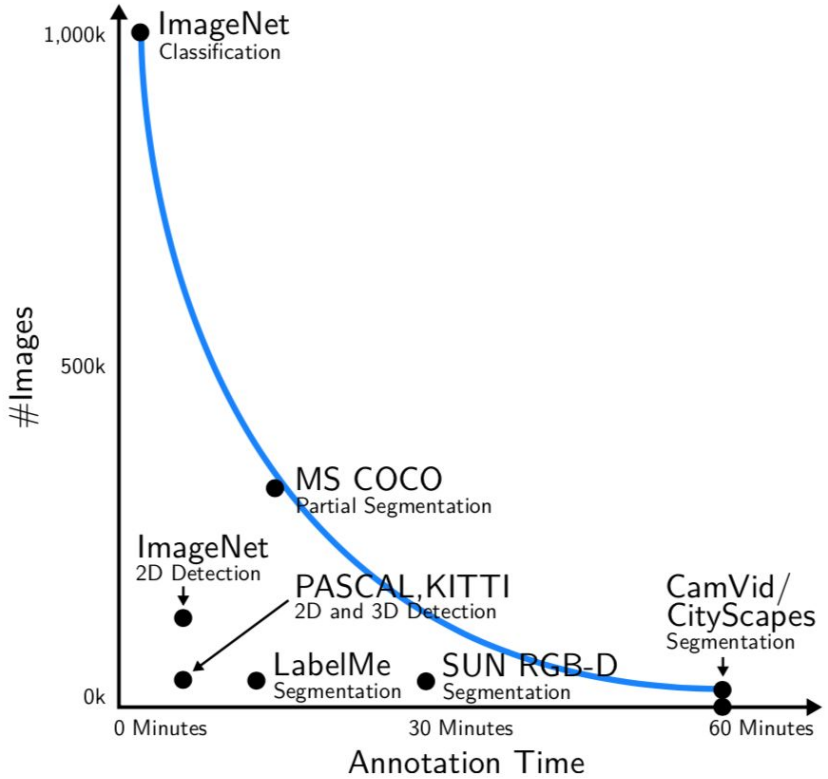
Dense Semantic and Instance Labels

“Cityscape” dataset: Labeling every pixel as person/road/sidewalk …

Annotation time **60-90 minutes per image**

Slides from Andreas Geiger, MPI Tubingen

Annotate everything - expensive, doesn't scale!



Slides from Andreas Geiger, MPI Tubingen

Motivation - Humans learn with sparse signal

Provided with very few “labeled” examples (someone pointing something out to us explicitly), we can generalize quite well.

Slides from Andreas Geiger, MPI Tubingen

Today's Class

- Recap
  - Supervised vs Unsupervised Learning
  - Why not always label data?
- Semi-supervised Learning
  - Concepts
  - Example: pseudo-labels / self-training
  - Example: Distillation, Student/Teacher
- Self-supervised Learning
  - Concepts
  - Pretext tasks
  - Contrastive Learning

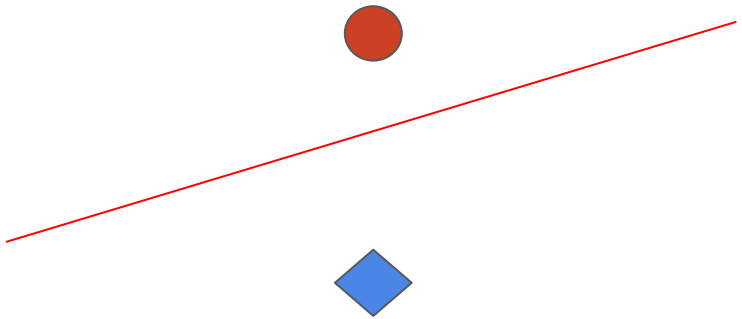
Semi-supervised Learning

- Given a small amount of ***labeled*** data $\mathcal{X}_L$
- Given (usually) large amount of ***unlabeled*** data $\mathcal{X}_U$
- Can $\mathcal{X}_U$ help us in getting a better model?

What is a good decision boundary for these points?

Semi-supervised Learning

- Given a small amount of ***labeled*** data $\mathcal{X}_L$
- Given (usually) large amount of ***unlabeled*** data $\mathcal{X}_U$
- Can $\mathcal{X}_U$ help us in getting a better model?

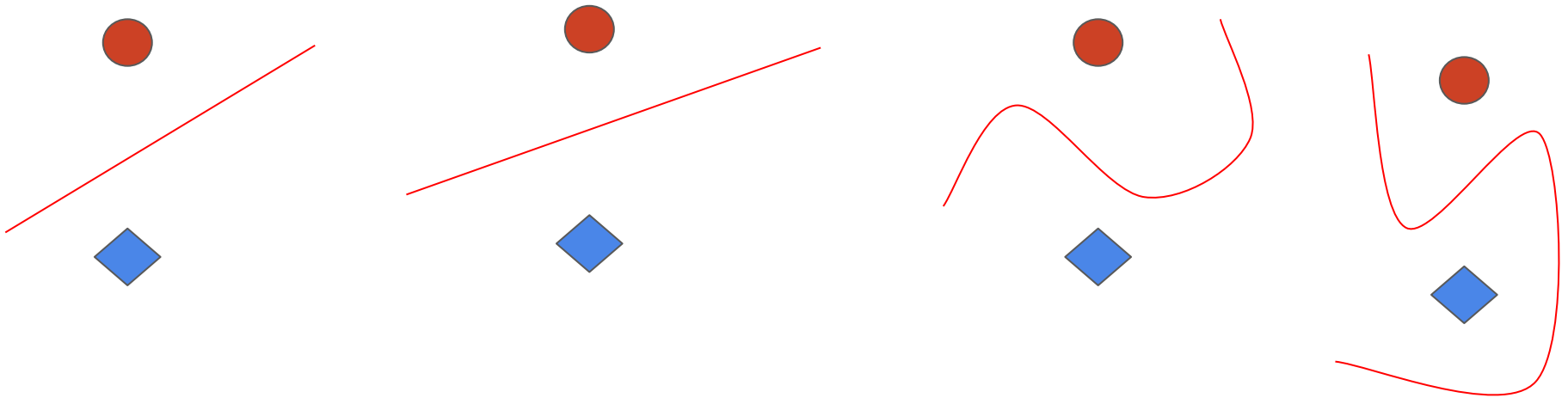


What is a good decision boundary for these points?

Semi-supervised Learning

- Given a small amount of ***labeled*** data $\mathcal{X}_L$
- Given (usually) large amount of ***unlabeled*** data $\mathcal{X}_U$
- Can $\mathcal{X}_U$ help us in getting a better model?

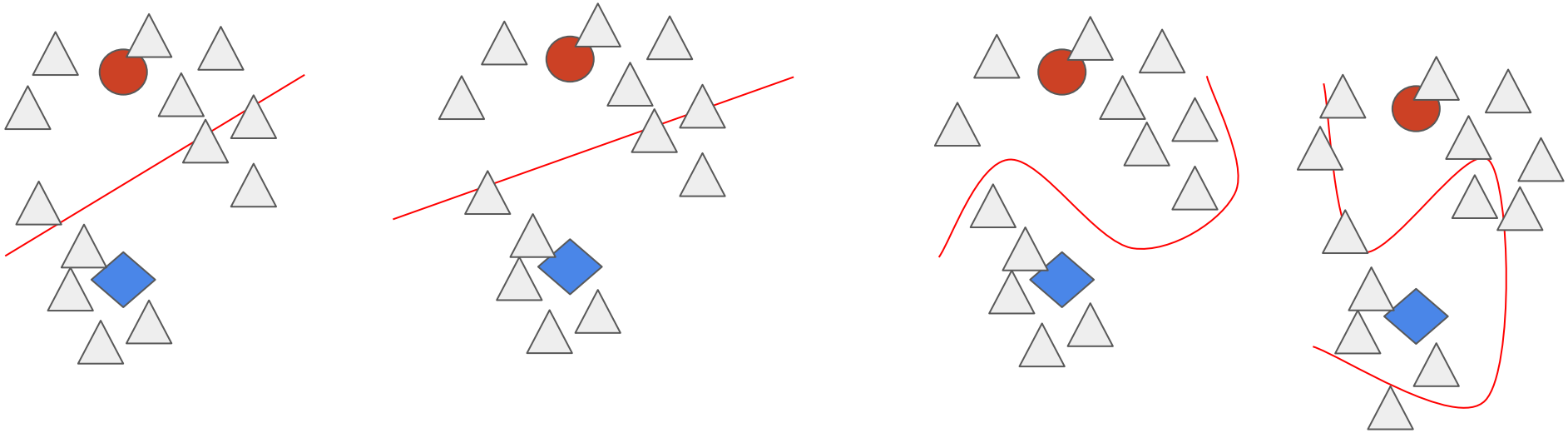
Which one is your favourite?



Semi-supervised Learning

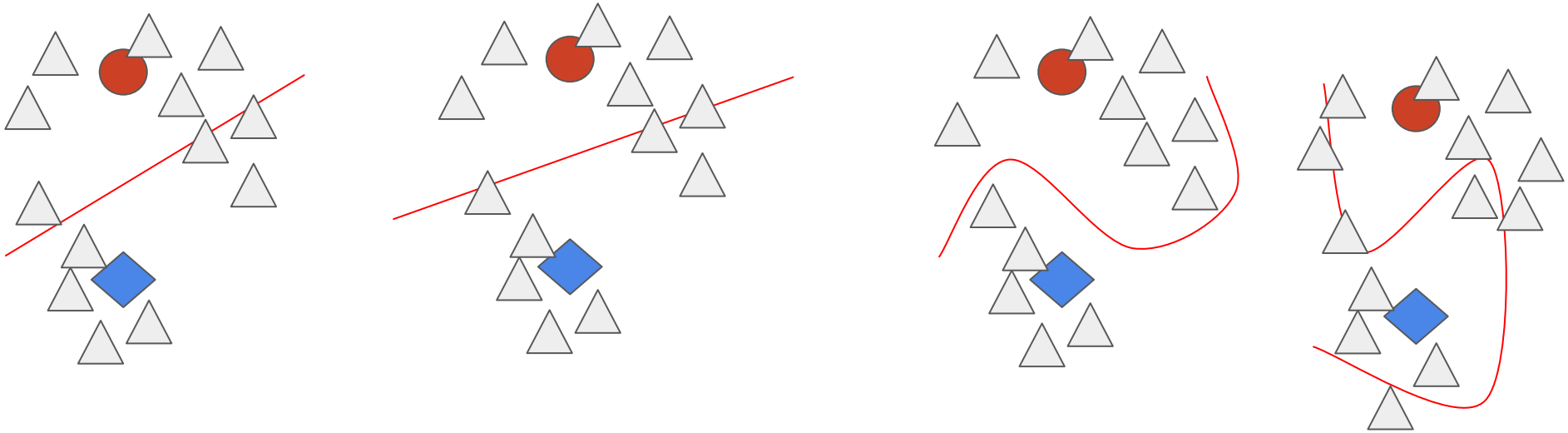
- Given a small amount of ***labeled*** data $\mathcal{X}_L$
- Given (usually) large amount of ***unlabeled*** data $\mathcal{X}_U$
- Can $\mathcal{X}_U$ help us in getting a better model?

Now we see some unlabeled data points ….



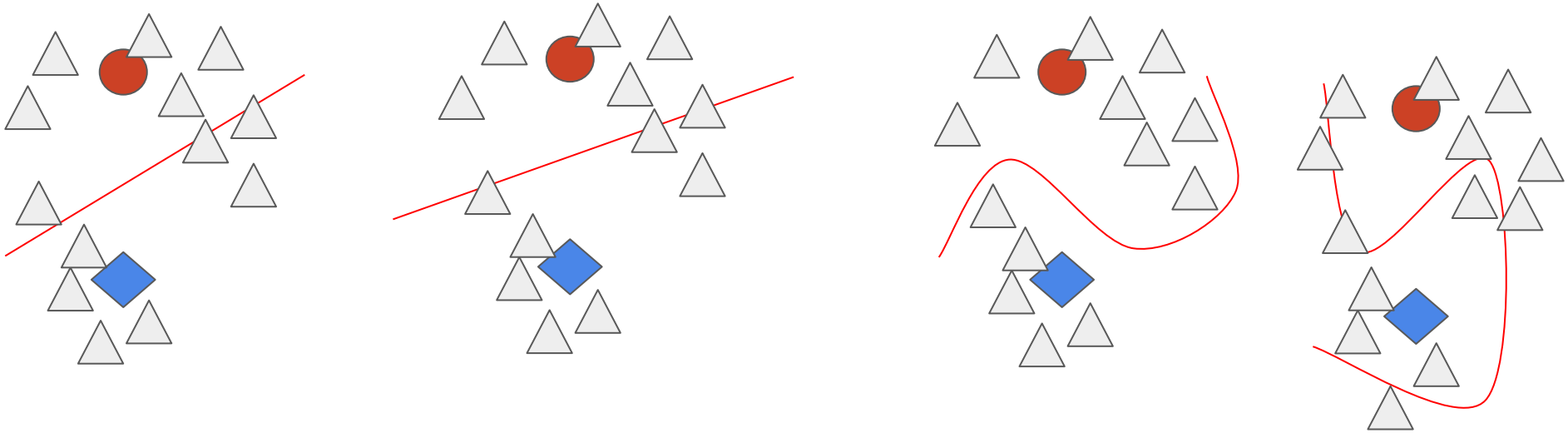
Semi-supervised Learning - intuitions

- Unlabeled samples tell us about P(**X**), which is useful in the predictive posterior P(y | **X**)



Semi-supervised Learning - definitions

- **Smoothness assumption:** if x1, x2 are close, labels y1, y2 are also “close”
- **Low-density separation:** x1, x2 are separated by *low-density*, labels are not close
- **Cluster assumption:** points in same cluster likely to have same label



Semi-supervised Learning Approaches

- We will look at *two* specific approaches to semi-supervised learning
- **Self-training** or **pseudo-labeling**
  - Age-old method
  - Surprisingly good with modern deep learning methods
- **Distillation** and **Student-Teacher**
  - Take the predictions of a “teacher” model
  - Use this to train a “student” model
  - Useful in supervised and semi-supervised applications

Self-training

- Assume: one's own high confidence predictions are correct!

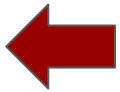
- Train model f on $\mathcal{X}_L := \{x_L, y_L\}$
- Use f to predict "pseudo-labels" on $\mathcal{X}_U := \{x_u\}$
- Add $\{x_u, f(x_u)\}$ to labeled data
- Repeat

Based off Joelle Pineau's COMP-551

Self-training - variations

- Assume: one's own high confidence predictions are correct!

- Train model f on $\mathcal{X}_L := \{x_L, y_L\}$
- Use f to predict "pseudo-labels" on $\mathcal{X}_U := \{x_u\}$
- Add $\{x_u, f(x_u)\}$ to labeled data
- Repeat



1) Add only a few most confident predictions on Xu
2) Add all predictions on Xu
3) Add all predictions, weighted by the confidence of the prediction

….

Based off Joelle Pineau's COMP-551

Self-training advantages

- The simplest semi-supervised method!
- It's a "wrapper" - the classifiers or models can be arbitrarily complex, we do not need to delve into those details to apply self-training
- Often quite good in practice, e.g. in natural language tasks

Based off Joelle Pineau's COMP-551

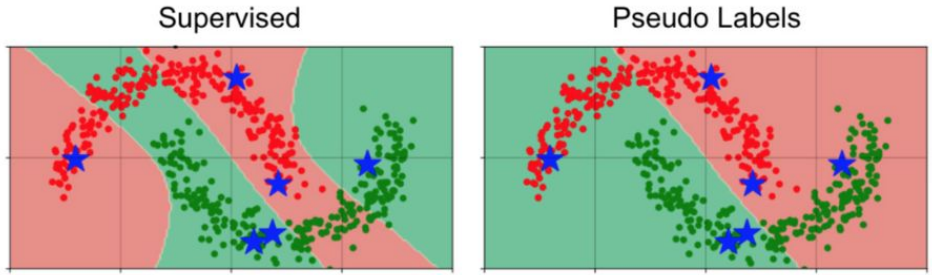
Disadvantages of self-training?

Any guesses?

Disadvantages of self-training?

- Early mistakes can reinforce themselves
  - We have heuristic solutions, like discarding samples if the confidence of prediction falls below some threshold

- Convergence
  - Hard to say if these steps of self-train and repeat will converge



Picture from Pham et al. Meta Pseudo Labels

Distillation - the basic idea

- *Transfer knowledge* from a trained Teacher model (large/complex model or ensemble of models) to a smaller Student model by training it to mimic the teacher's output.

Slides originally from Pavlos Protopapas, Advanced Practical Data Science

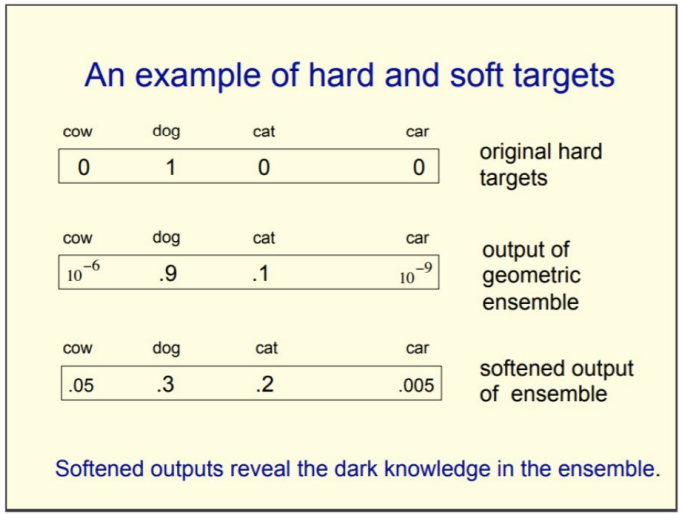
Distillation - the loss function

- We are already familiar with the cross-entropy loss for classification

Modified softmax function with Temperature:

$$qi = \frac{\exp\left(\frac{z_i}{T}\right)}{\sum_j \exp\left(\frac{z_j}{T}\right)}$$

q_i : resulting probability
z_i : logit of a class
z_j : other logits
T: temperature (T=1, "hard output")

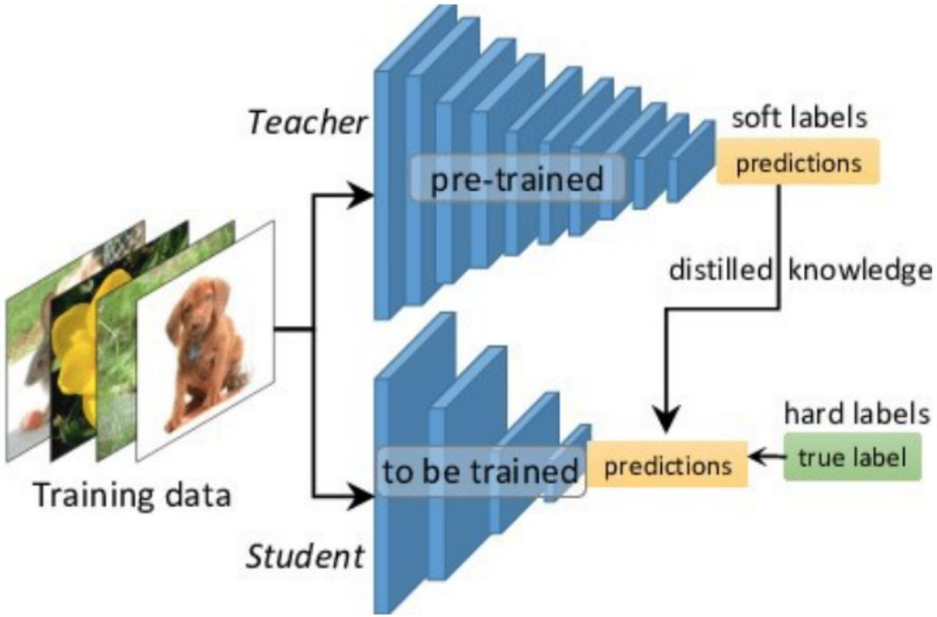


Slides originally from [Pavlos Protopapas, Advanced Practical Data Science]

Distillation - Student-Teacher training

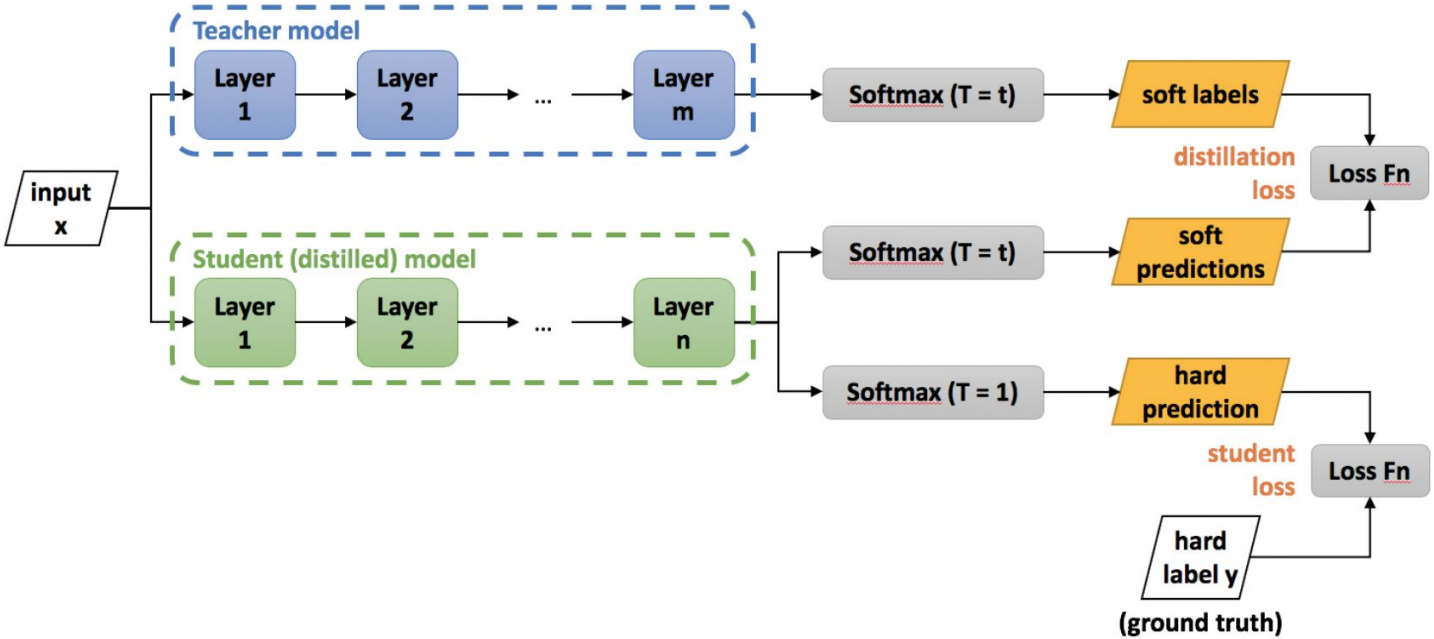
Trained to minimize the sum of two different cross entropy functions:

- one involving the original hard labels obtained using a softmax with T=1
- one involving the softened targets, T>1



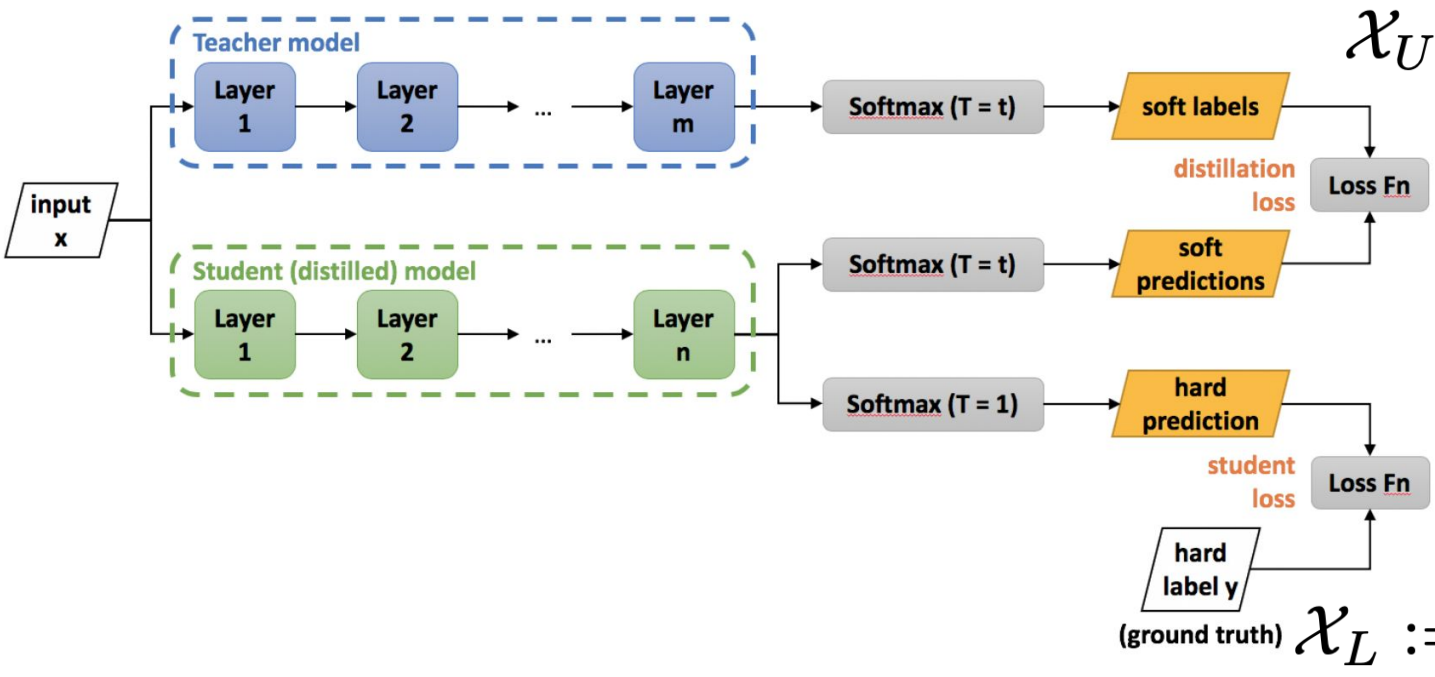
Slides originally from [Pavlos Protopapas, Advanced Practical Data Science]

Distillation - Student-Teacher training



Slides originally from [Pavlos Protopapas, Advanced Practical Data Science]

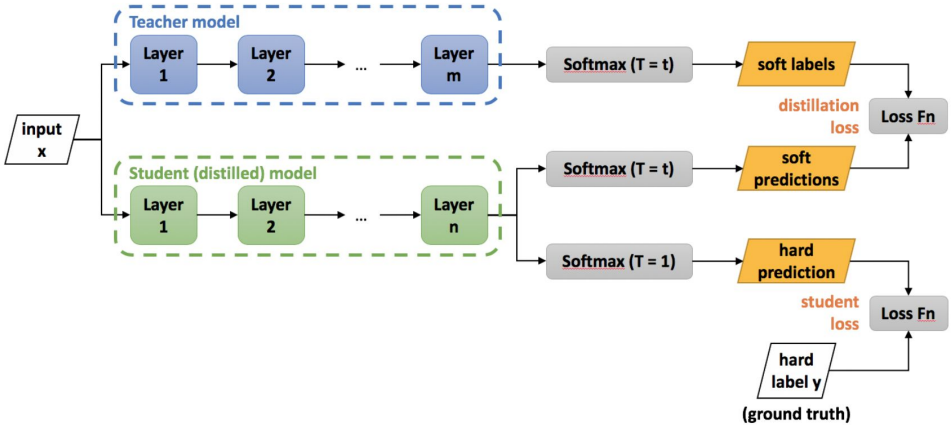
Distillation - may be applied to unlabeled data



Slides originally from [Pavlos Protopapas, Advanced Practical Data Science]

Distillation - design of the Teacher

- **Goal:** Improve the target quality (Teacher's predictions) on unlabeled data
- Simple - model trained on labeled data (.. self-training + distillation)
- Better - keep an average of the saved student model at various iterations



$$\mathcal{X}_U := \{x_u\}$$

$$\mathcal{X}_L := \{x_L, y_L\}$$

Slides originally from [Pavlos Protopapas, Advanced Practical Data Science]

More pointers on SSL

- Semi-supervised Learning (SSL) is a vast area both in terms of ML theory and applications
- Other interesting methods in the deep learning era:
  - **Entropy minimization:** adds a loss that encourages the neural network model to make high confidence predictions (minimize “entropy”) on all unlabeled samples
  - **Variations of Teacher/Student:** Mean Teacher, FixMatch, NoisyStudent …

Questions?

Today's Class

- Recap
  - Supervised vs Unsupervised Learning
  - Why not always label data?
- Semi-supervised Learning
  - Concepts
  - Example: pseudo-labels / self-training
  - Example: Distillation, Student/Teacher
- Self-supervised Learning
  - Concepts
  - Pretext tasks
  - Contrastive Learning

Self-supervised Learning

- **Motivation:** back to how humans learn (and we know already that having humans provide labels for everything is not realistic)

- Provided only very few “labeled” examples, **humans generalize very well**
- Humans learn through **interaction** and **observation**

Slides from Andreas Geiger, MPI Tubingen

Self-supervision - motivations

Humans learn through interaction and observation

Slides from Andreas Geiger, MPI Tubingen

Self-supervision - the basic idea

Observed data

Hidden data

Hidden property of the data

Idea of self-supervision:

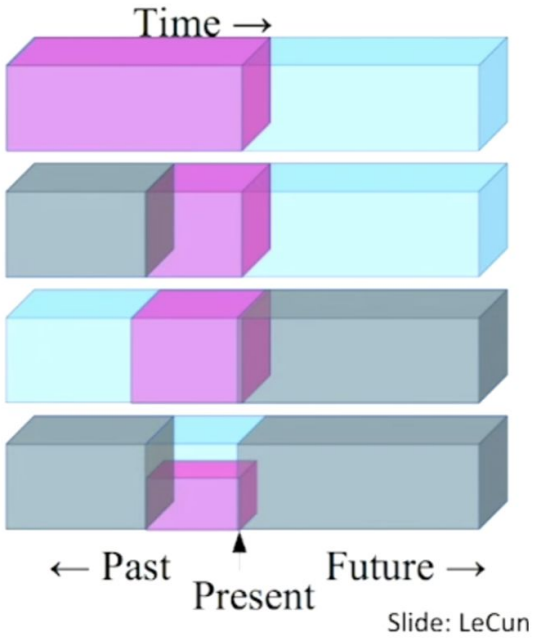
- Obtain labels from raw unlabeled data itself
- Predict parts of the data from other parts

Slide credits: Yann LeCun and Ishan Misra

Slides from Andreas Geiger, MPI Tubingen

Self-supervision - use unseen parts of data in training loss

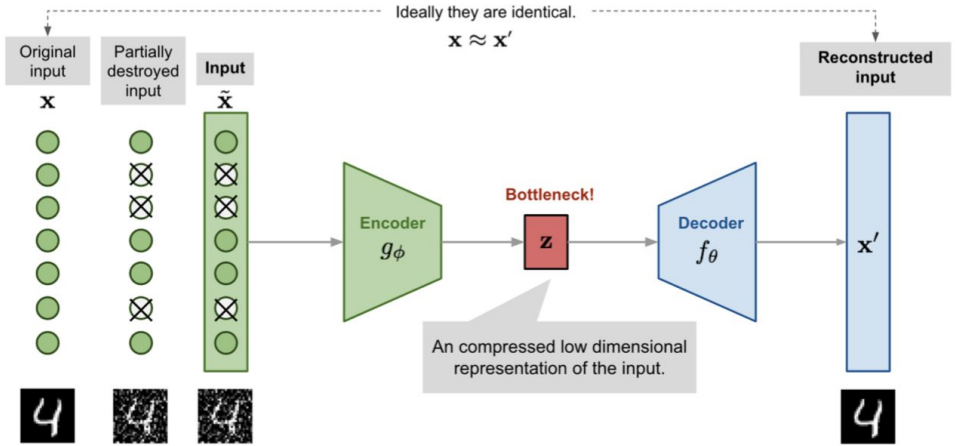
- Predict any part of the input from any other part.
- Predict the future from the past.
- Predict the future from the recent past.
- Predict the past from the present.
- Predict the top from the bottom.
- Predict the occluded from the visible
- Pretend there is a part of the input you don't know and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



Slide credits: Yann LeCun and Ishan Misra

Slides from Andreas Geiger, MPI Tubingen

Example: Denoising Autoencoder



- Example: **Denoising Autoencoder (DAE)** predicts input from corrupted version
- After training, only the encoder is kept and the decoder is thrown away

Vincent, Larochelle, Bengio and Manzagol: Extracting and composing robust features with denoising autoencoders. ICML, 2008.

- Self-supervision to train a **feature encoder** (a simple linear classifier on top can work well)

Slides from Andreas Geiger, MPI Tubingen

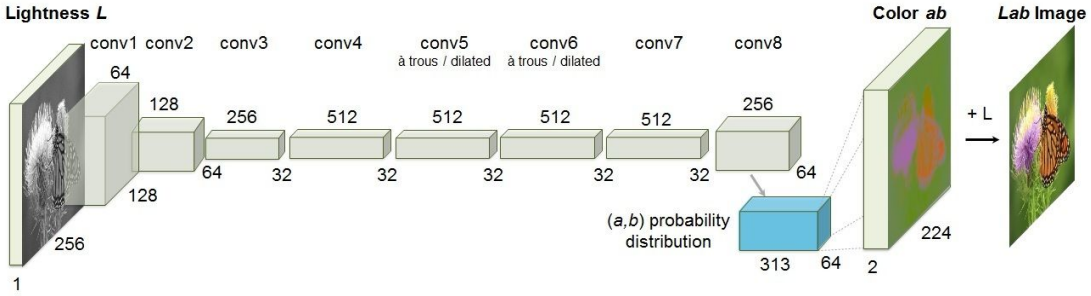
Self-supervision via Proxy Tasks

- The first attempts at self-supervision used ***proxy*** or ***pretext tasks*** like colorizing grayscale images, predicting the depth from a single image, solving jigsaw puzzles, etc.

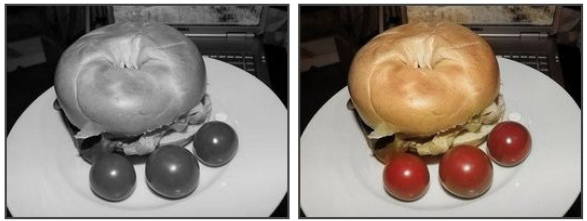
- You can train a neural network on these tasks with just a bunch of images
  - No human-provided labels at all!

- Why would any of these tasks help, say, classify an image as an apple or a cat?
  - These tasks help the network learn about the *structure* of the data (P(**x**))
  - The self-trained neural networks can then be used as **feature extractors**, or **fine-tuned**, on downstream tasks like classifying images

Colorization for learning Visual Representations

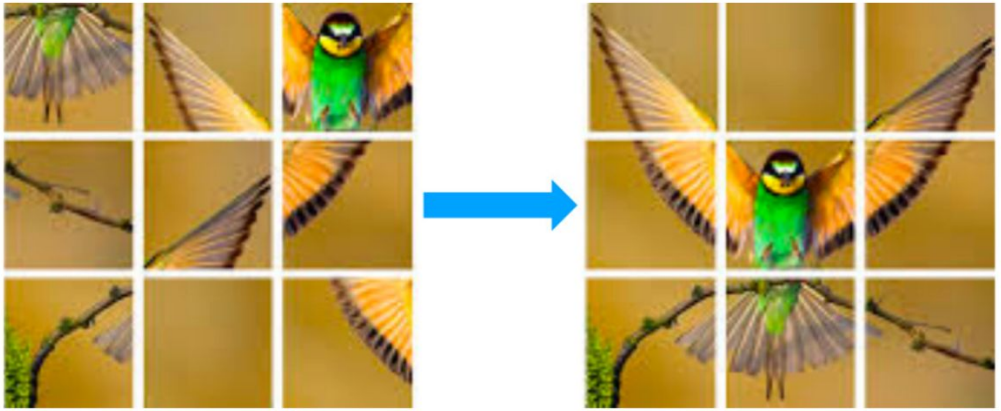


- Given grayscale, predict the color channels
- ***Why does this help in learning about classes/objects?***



Figures and images from Zhang and Efros, and Larsson, Maire and Shakhnarovich

Learn Visual Representations from Jigsaw Puzzles



Input: nine patches
Permute using one of N permutations

Output: N-way classification

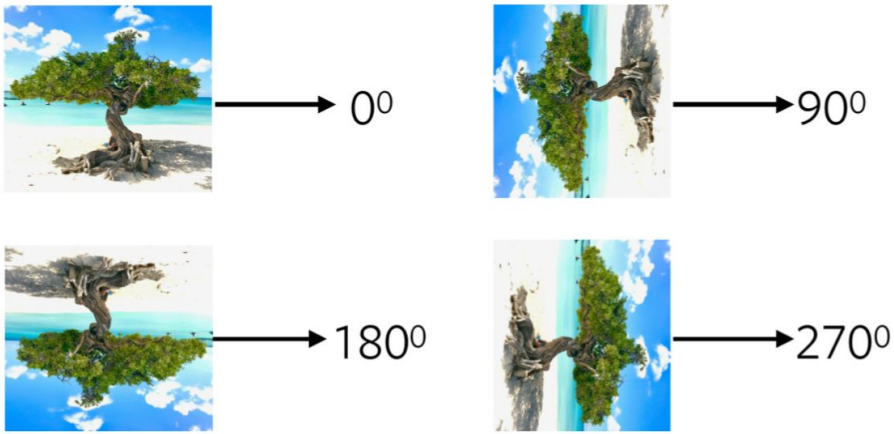
Set N << 9!

- Jigsaw puzzle task: predict one out of 1000 possible random permutations
- Permutations chosen based on Hamming distance to increase difficulty

Noroozi and Favaro: Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. ECCV, 2016.

Slides from Andreas Geiger, MPI Tubingen

Learning to predict image rotations



Input: image rotated by [0, 90, 180, 270]

Output: 4-way classification

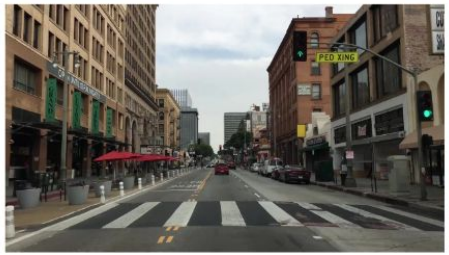
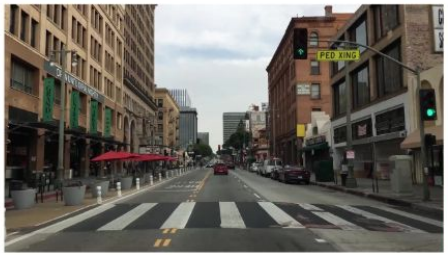
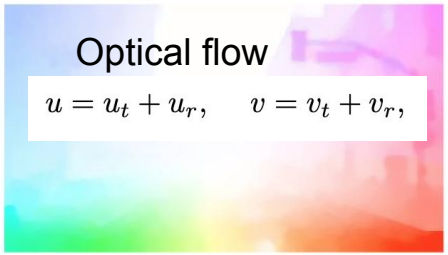
- **Rotation task:** try to recover the true orientation (4-way classification)
- Idea: in order to recover the correct rotation, semantic knowledge is required

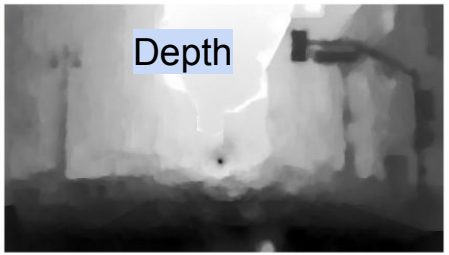
Gidaris, Singh and Komodakis: Unsupervised Representation Learning by Predicting Image Rotations. ICLR, 2018.

Slides from Andreas Geiger, MPI Tubingen

Predicting Depth for Urban Scene Understanding

- Use an existing off-the-shelf method to estimate the “relative depth” from two consecutive video frames (no human labels → “self supervised”)



$$u_t = \frac{-U + xW}{Z}, \quad v_t = \frac{-V + yW}{Z},$$

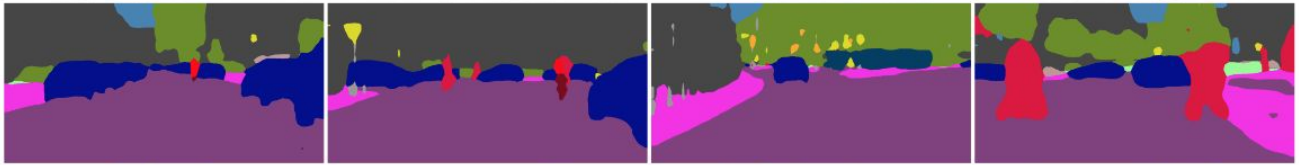
$$Z = \sqrt{\frac{(-U + xW)^2 + (-V + yW)^2}{u_t^2 + v_t^2}}.$$

Huaizu Jiang, et al. “Self-Supervised Relative Depth Learning for Urban Scene Understanding”

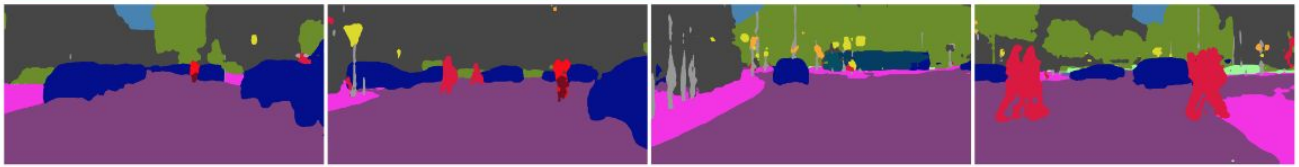
Predicting Depth for Urban Scene Understanding

Images

Baseline



Self-sup



This extra pre-training on massive unlabeled data helps!

- Compute relative depth on 1.1M video frames of YouTube “CityDriving” videos
- Train a neural network on this self-supervised task
- Fine-tune on downstream task: **semantic segmentation of city streets**

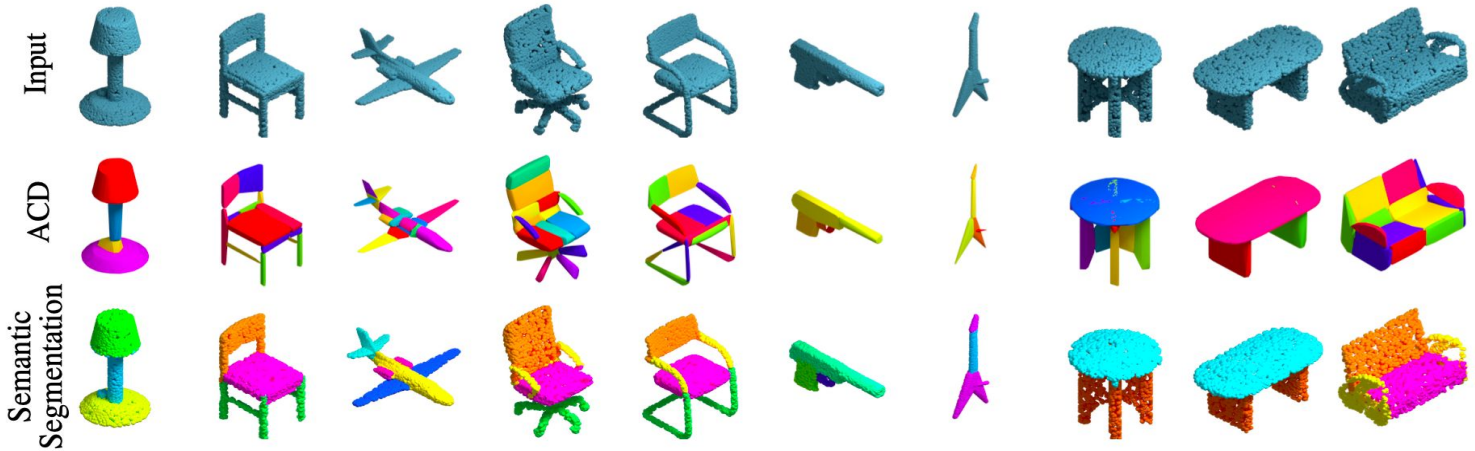
Huaizu Jiang, et al. “Self-Supervised Relative Depth Learning for Urban Scene Understanding”

Another task - 3D shapes and convexity

- **Final Task:** separate 3D ***objects*** (chairs, tables..) into ***parts*** (legs, back, handles…)

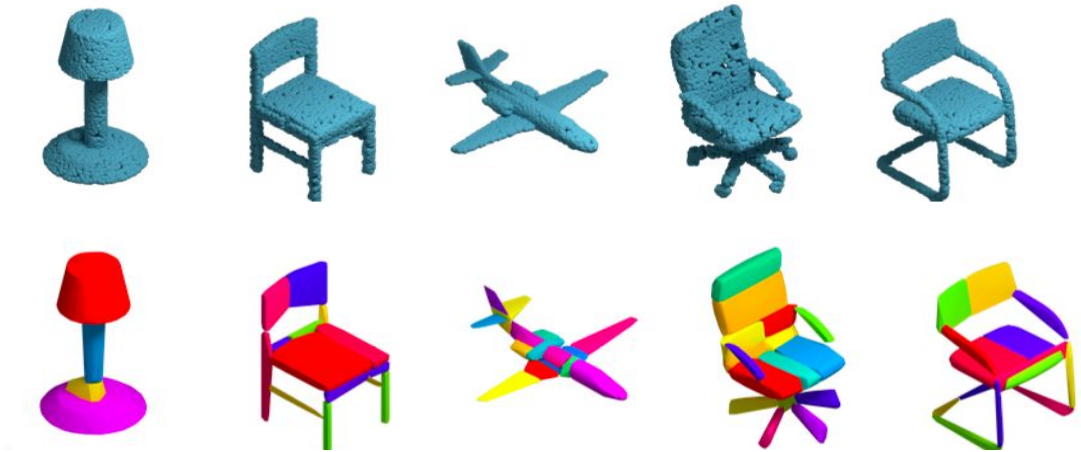
Approx Convex Decompositions and Semantic Parts

- **Final Task:** separate 3D ***objects*** (chairs, tables..) into ***parts*** (legs, back, handles…)
- **Pretext Task:** off-the-shelf package for “approximate convex decomposition”



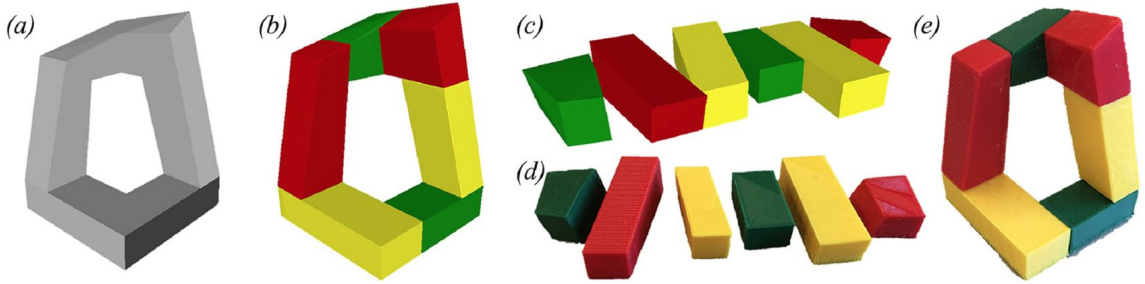
More on the pretext task - approx convexity

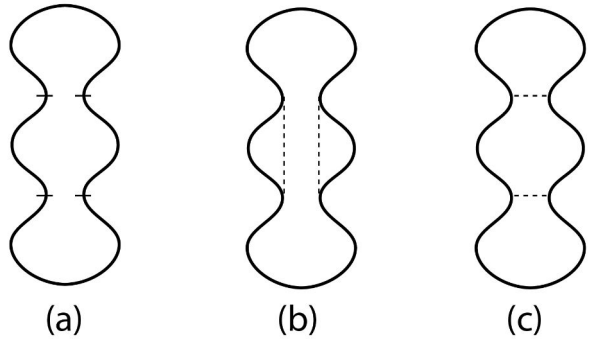
- **Pretext Task:** off-the-shelf package for “approximate convex decomposition”
  - Get a tonne of unlabeled 3D shapes
  - Run off-the-shelf “ACD” software to get decompositions
  - Train your favourite 3D neural network on this, and then apply on final task

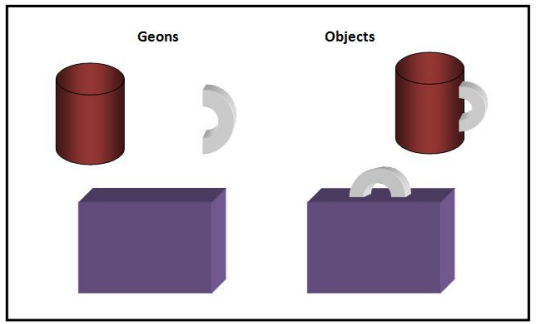


Shape decompositions - the TL;DR version

- Why **convex**?
  - Man-made shapes efficiently assembled from convex or nearly-convex parts
  - Cognitive Science: Part-whole theory [Hoffman], Convex Patches, *Geons* [Biederman]



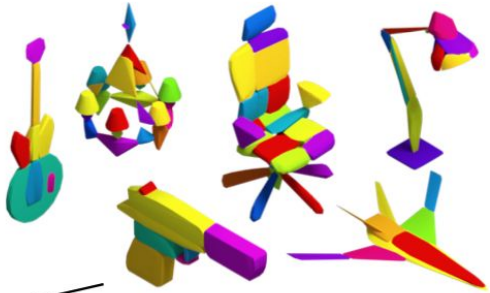




[ECCV 2020]

Approximate Convex Decomposition

ACD

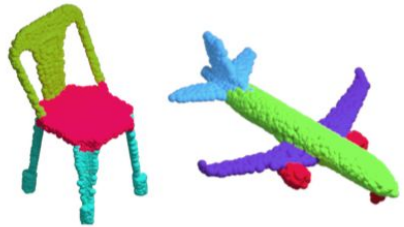


Self-supervision

Neural Network

Human Annotation

Supervision



Gadelha and RoyChowdhury, et al.

[ECCV 2020]

Pairwise loss over ACD components:

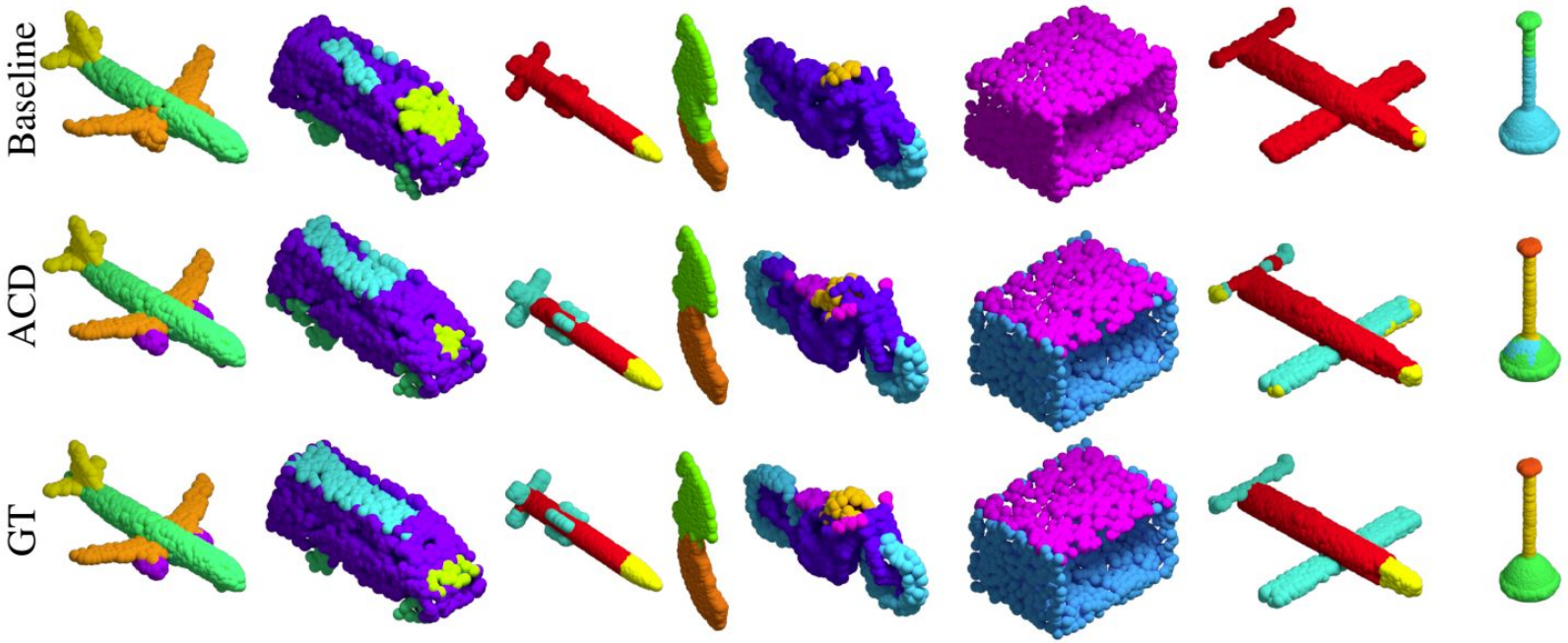
$$\mathcal{L}^{pair}(\mathbf{x}, p_i, p_j, \mathcal{C}) = \begin{cases} 1 - \Phi(\mathbf{x})_i^\top \Phi(\mathbf{x})_j, & \text{if same component} \\ \max(0, \Phi(\mathbf{x})_i^\top \Phi(\mathbf{x})_j - m), & \text{if different component} \end{cases}$$

Embedding of the *i*th point in the point cloud x

Gadelha and RoyChowdhury, et al.

[ECCV 2020]

10-Shot Segmentation Results



Gadelha and RoyChowdhury, et al.

Summary of self-supervision via pretext-tasks

Pretext Tasks:

- Pretext tasks focus on “visual common sense”, e.g., rearrangement, predicting rotations, inpainting, colorization, etc.
- The models are forced learn good features about natural images, e.g., semantic representation of an object category, in order to solve the pretext tasks
- We don’t care about pretext task performance, but rather about the utility of the learned features for downstream tasks (classification, detection, segmentation)

Problems:

- Designing good pretext tasks is tedious and some kind of “art”
- The learned representations may not be general

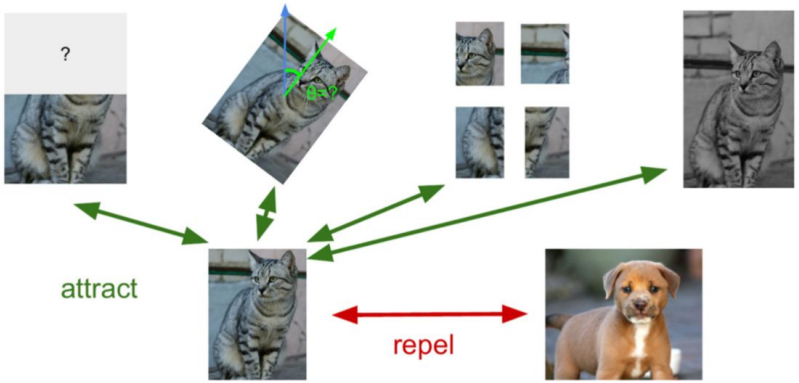
Slides from Andreas Geiger, MPI Tubingen

Contrastive Learning

- Designing the correct pretext-task for a particular downstream task is challenging

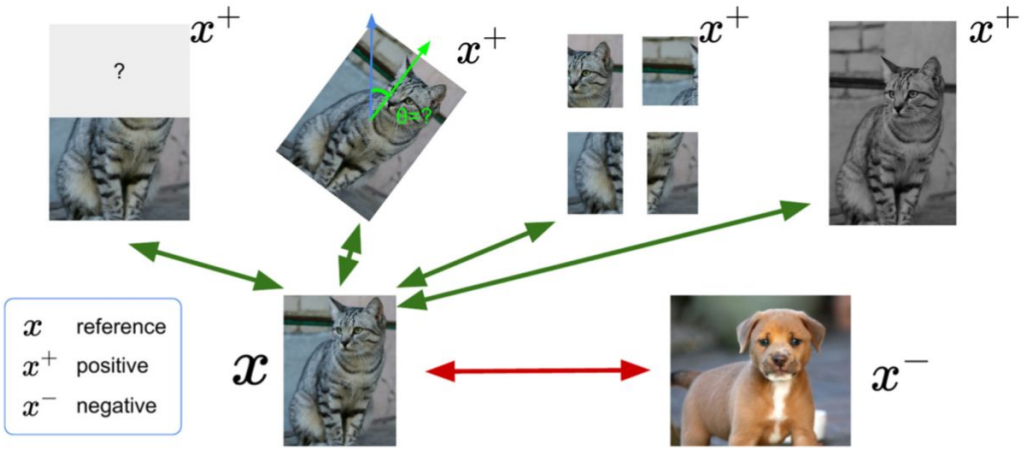
Can we find a more general pretext task?

- Pre-trained features should represent **how images relate** to each other
- They should also be **invariant to nuisance factors** (location, lighting, color)
- Augmentations generated from one reference image are called “views”



Slides from Andreas Geiger, MPI Tubingen

Contrastive Learning



- Given a chosen **score function** $s(\cdot, \cdot)$, we want to learn an encoder f that yields **high score for positive pairs** (x, x^+) and **low score for negative pairs** (x, x^-):

$$s(f(x), f(x^+)) \gg s(f(x), f(x^-))$$

Slides from Andreas Geiger, MPI Tubingen

Contrastive Learning

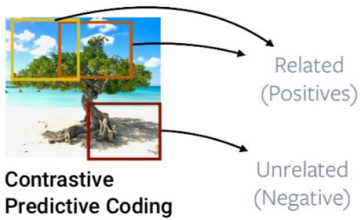
Design Choices

1. Score Function:

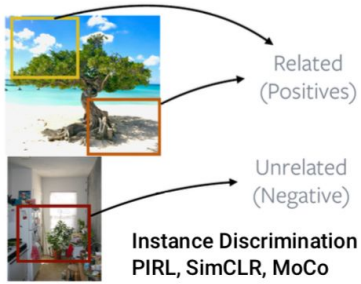
$$s(\mathbf{f}_1, \mathbf{f}_2) = \frac{\mathbf{f}_1^\top \mathbf{f}_2}{\|\mathbf{f}_1\| \, \|\mathbf{f}_2\|}$$

- Cosine similarity
- Commonly used

2. Examples:

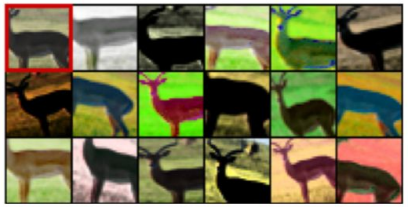


Contrastive Predictive Coding



Instance Discrimination
PIRL, SimCLR, MoCo

3. Augmentations:



- Crop, resize, flip
- Rotation, cutout
- Color drop/jitter
- Gaussian noise/blur
- Sobel filter

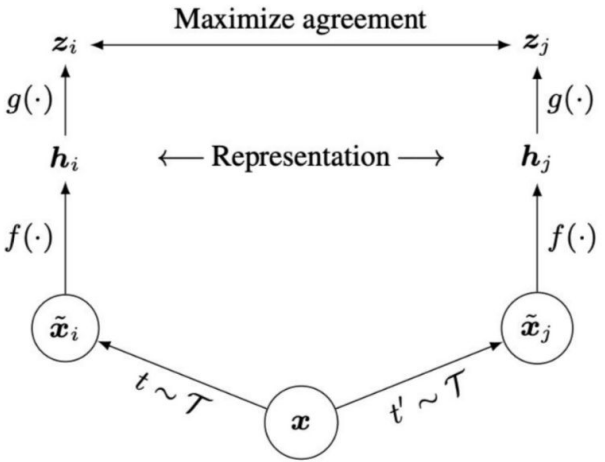
Slides from Andreas Geiger, MPI Tubingen

Simple Framework for Contrastive Learning

- **Cosine similarity** as score function:

$$s(\mathbf{z}_i, \mathbf{z}_j) = \frac{\mathbf{z}_i^\top \mathbf{z}_j}{\|\mathbf{z}_i\| \, \|\mathbf{z}_j\|}$$

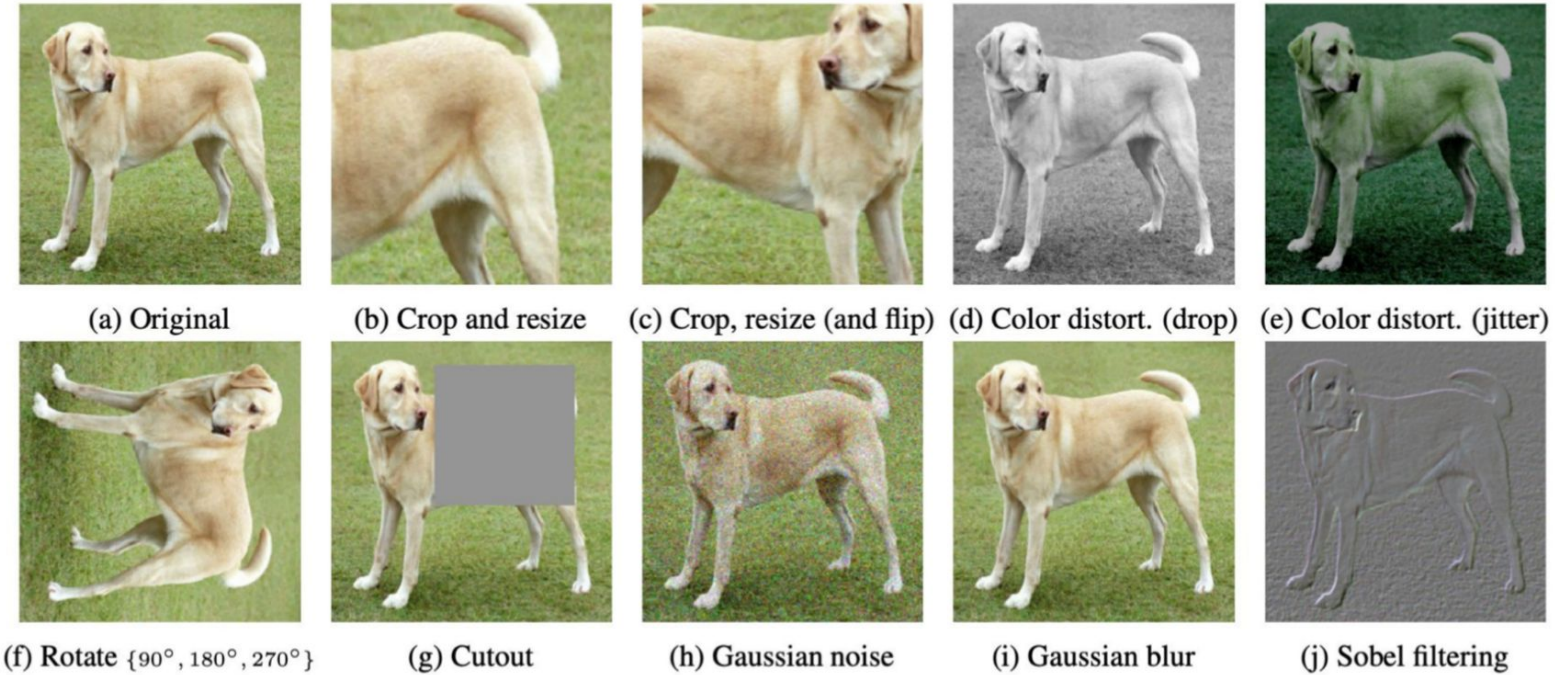
- SimCLR uses a **projection network** $g(\cdot)$ to project features to a space where contrastive learning is applied
- The projection improves learning (more relevant information preserved in $\mathbf{h}$ which is discarded in $\mathbf{z}$)



Chen, Kornblith, Norouzi and Hinton: A Simple Framework for Contrastive Learning of Visual Representations. ICML, 2020.

Slides from Andreas Geiger, MPI Tubingen

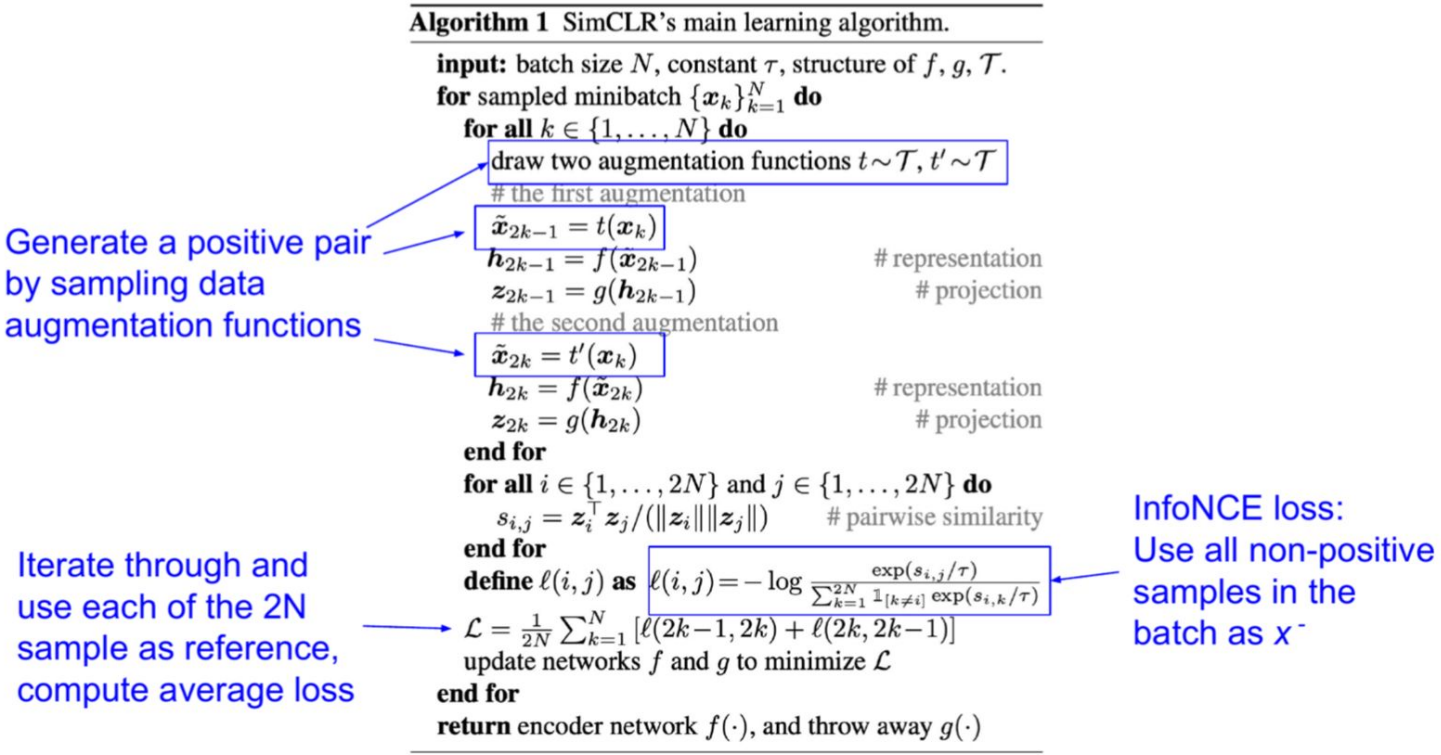
Simple Framework for Contrastive Learning



Chen, Kornblith, Norouzi and Hinton: A Simple Framework for Contrastive Learning of Visual Representations. ICML, 2020.

Slides from Andreas Geiger, MPI Tubingen

Simple Framework for Contrastive Learning



Chen, Kornblith, Norouzi and Hinton: A Simple Framework for Contrastive Learning of Visual Representations. ICML, 2020.

Slides from Andreas Geiger, MPI Tubingen

Contrastive Learning - Results on Image Classification

Method	Architecture	Label fraction 1%	Label fraction 10%
		Top 5	
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**

Table 7. ImageNet accuracy of models trained with few labels.

Train feature encoder on **ImageNet** (entire training set) using SimCLR.

Finetune the encoder with 1% / 10% of labeled data on ImageNet.

Chen, Kornblith, Norouzi and Hinton: A Simple Framework for Contrastive Learning of Visual Representations. ICML, 2020.

Slides from Andreas Geiger, MPI Tubingen

More pointers on Contrastive Learning

- These slides give an overall idea of Contrastive Learning, and show the details of a single method – “SimCLR”

- Other variants include
  - MoCO - MoMentum Contrast
  - Barlow Twins
  - ….

Summary

- Creating labeled training data is time-consuming and expensive
- Semi-supervised approaches utilize both labeled and unlabeled data
  - **Pseudo-labeling** or self-training
  - **Distillation** applications like Mean Teacher …
- Self-supervision methods learn from (unlabeled) data alone
  - Then either fine-tuned or used as feature extractors on downstream tasks with limited labeled data
- Pretext tasks (colorization, rotation, jigsaw) may not always align well with target task
- Contrastive learning gives a more general way to learn these representations

Thank You

Questions?