Self-supervised learning on images

by Anton Naumov, ISPRAS, fall 2021

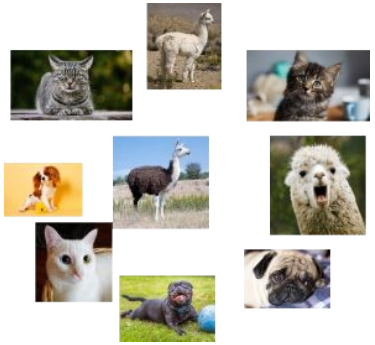
Plan

1. Motivation and concept
2. Evaluation approaches
3. Self-supervised methods
    3.1. “Annotation extraction”
    3.2. Autoencoders
    3.3. Generative models
    3.4. Contrastive methods
4. Demo (optional)
5. Conclusion

Motivation and concept

Motivation

- Annotation is expensive and time consuming
- DL models require a lot of annotated data
- Usually a lot of unannotated data is available



Hint: living things can learn even without supervision

Idea: learn something from unannotated data and use it in other tasks

Disclaimer

Today we are talking about images
(but most methods applicable on other data types).

We don’t cover self-supervised methods which actually require additional data
(learning from video, sensors on robots, image metadata etc).

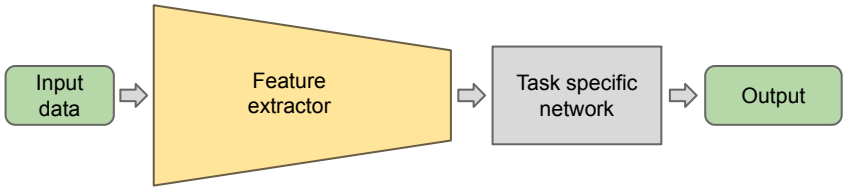
Concept

The dataset with unlabelled samples.

The aim is to train feature extractor.

Evaluation on the other problems like:

- Classification
- Regression
- Captioning
- Detection
- Information retrieval
- ...



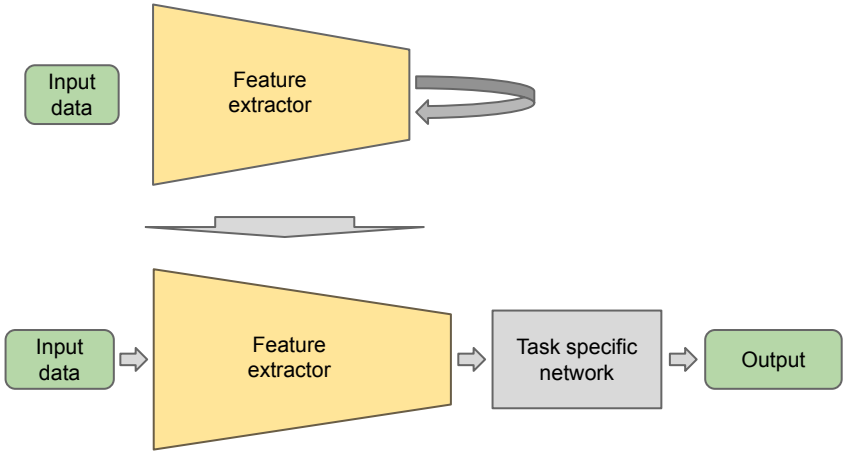
Concept

The dataset with unlabelled samples.

The aim is to train feature extractor.

Evaluation on the other problems like:

- Classification
- Regression
- Captioning
- Detection
- Information retrieval
- ...



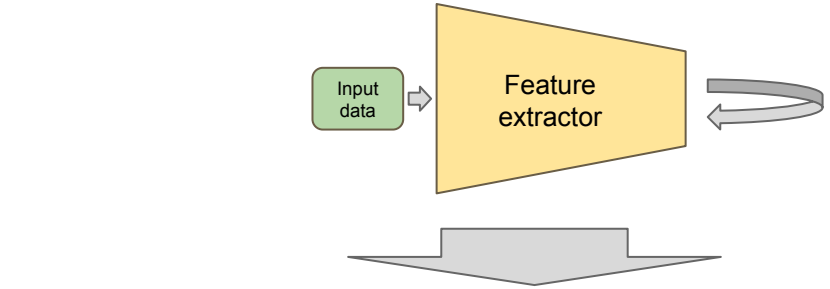
Close concepts

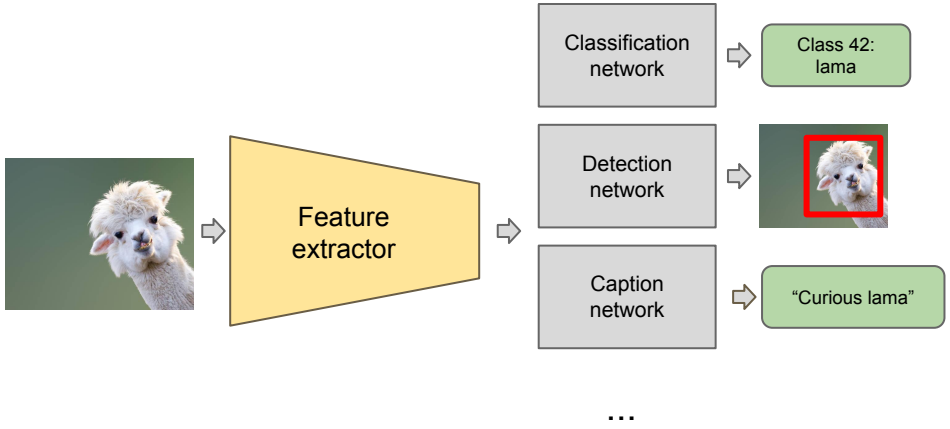
- Semi-supervised learning
  small annotated dataset and lot of unannotated data
- Weakly supervised learning
  Labels are incomplete
  (i.e. class labels provided but the objective is semantic segmentation)
- Few shot learning
  Few samples are available (usually it’s sufficient only for the mapping)
- Clustering
  The objective is to group samples into clusters

Evaluation approaches

Evaluation

- Metric on the target problem
- Samples separation
- Data efficiency
- Convergence speed

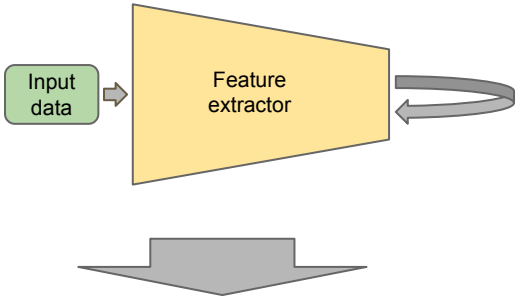


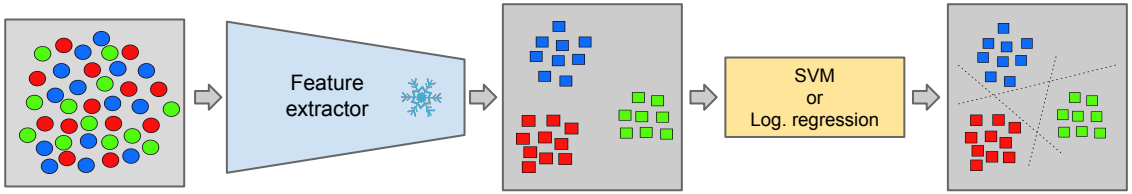


Note: for large annotated datasets there can be no metric improvement

Evaluation

- Metric on the target problem
- Samples separation
- Data efficiency
- Convergence speed

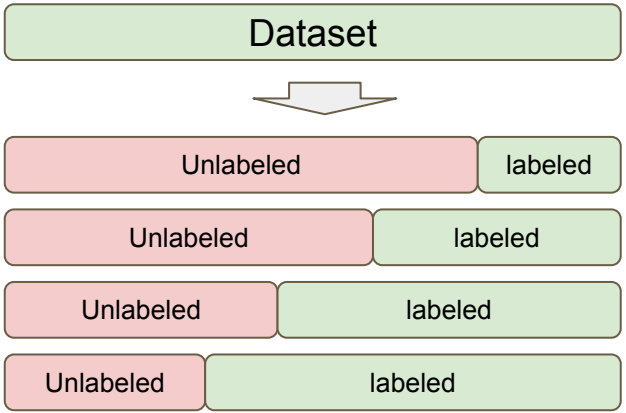




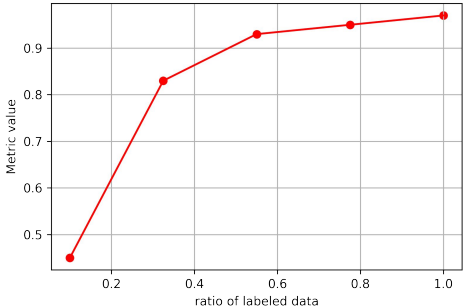
Idea: well-trained extractor will produce features which are easy to separate. Train simple classifier on the top of the features to evaluate the performance.

Evaluation

- Metric on the target problem
- Samples separation
- Data efficiency
- Convergence speed

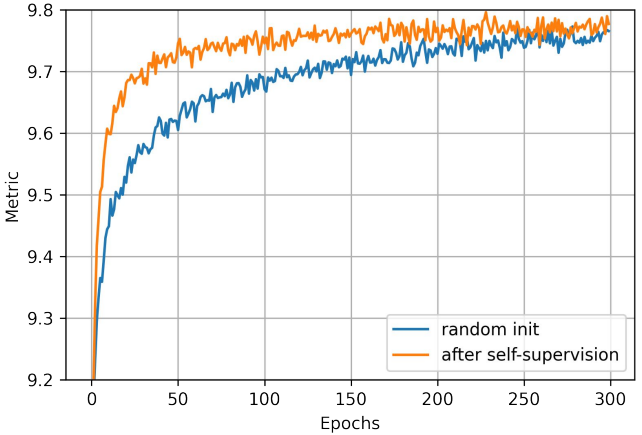


How much labeled data do we actually need?



Evaluation

- Metric on target problem
- Samples separation
- Data efficiency
- Convergence speed



Pretrained network should converge faster than random and maybe even than learned on other domain

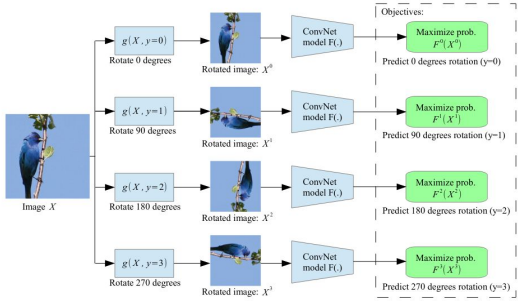
Self-supervised methods

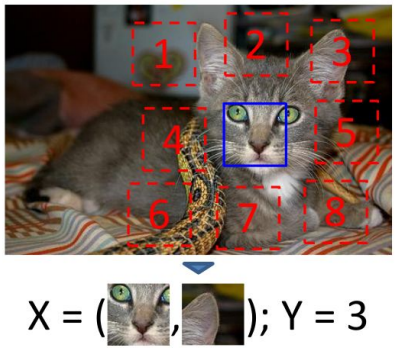
- “Annotation extraction”
- Autoencoders
- Generative models
- Contrastive methods

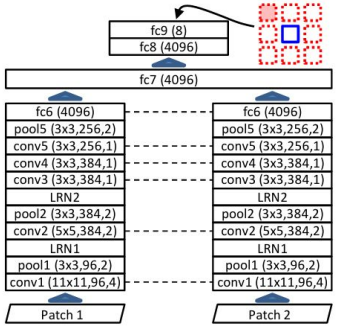
“Annotation extraction”

- Predict rotation

https://arxiv.org/pdf/1803.07728.pdf



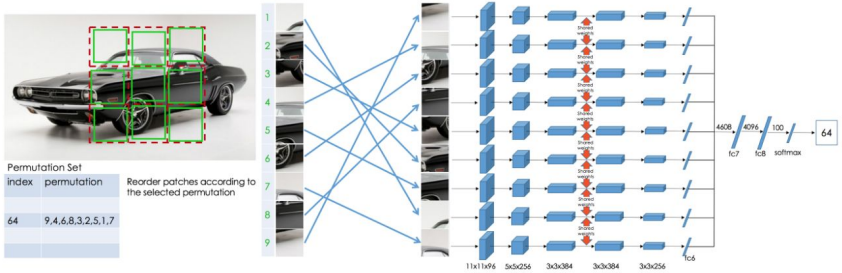




- Predict patch relative position

https://arxiv.org/pdf/1505.05192.pdf

“Annotation extraction”

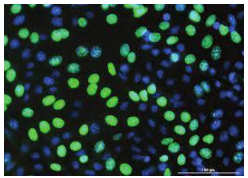


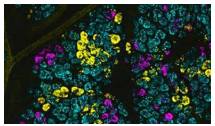
- Solve jigsaw puzzle

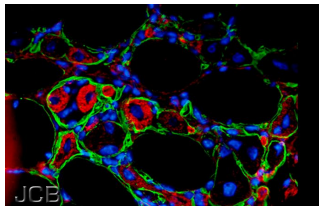
https://arxiv.org/pdf/1603.09246.pdf

These methods may fail for some kind of data.

Examples: images of the sky or cell imaging



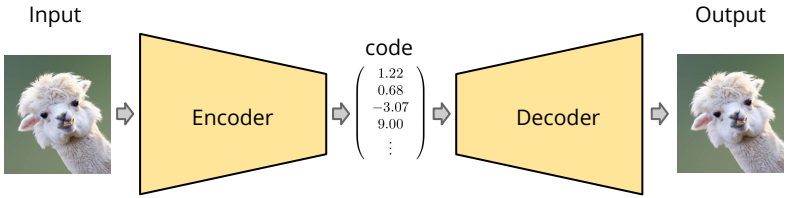




Autoencoders

Vanilla:

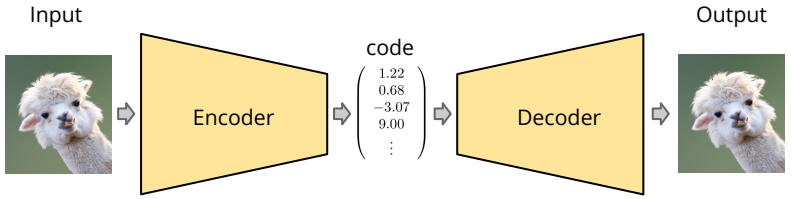
Compress the data with encoder, decompress with decoder.
Loss: pixelvice MSE, CrossEntopy



Autoencoders

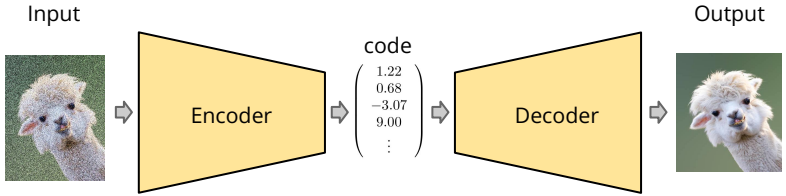
Vanilla:

Compress the data with encoder, decompress with decoder.
Loss: pixelvice MSE, CrossEntopy



Denoising:

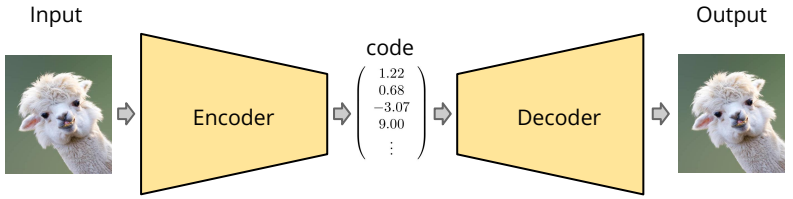
add noise to input, decompress without the noise
https://www.cs.toronto.edu/~larocheh/publications/icml-2008-denoising-autoencoders.pdf



Autoencoders

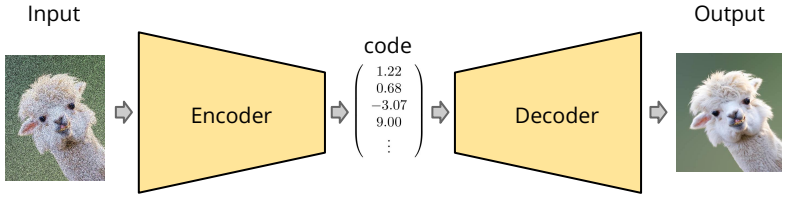
Vanilla:

Compress the data with encoder, decompress with decoder.
Loss: pixelvice MSE, CrossEntopy



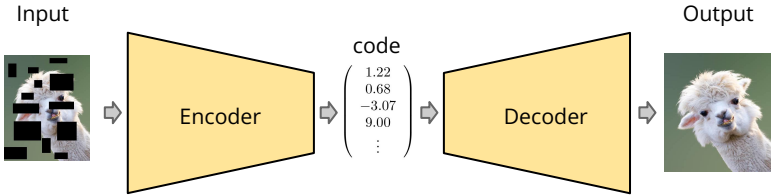
Denoising:

add noise to input, decompress without the noise
https://www.cs.toronto.edu/~larocheh/publications/icml-2008-denoising-autoencoders.pdf



Patch-based:

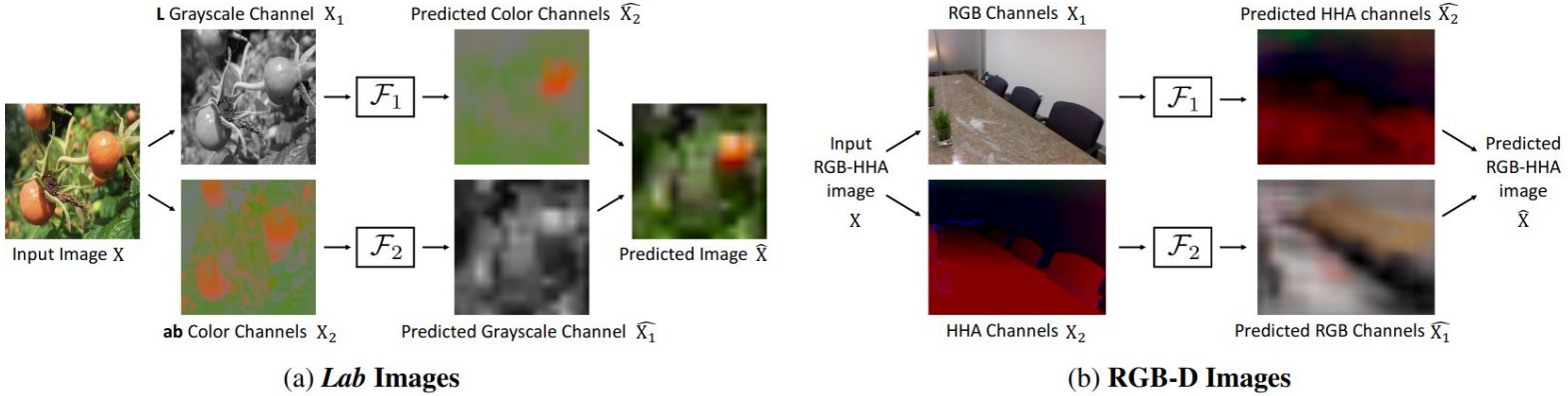
Delete patches from the input, reconstruct the original
https://arxiv.org/abs/1604.07379



Autoencoders

Split-brain:

Compress the data with encoder, decompress with decoder.
Loss: pixelvice MSE, CrossEntopy



https://arxiv.org/abs/1611.09842

Generative

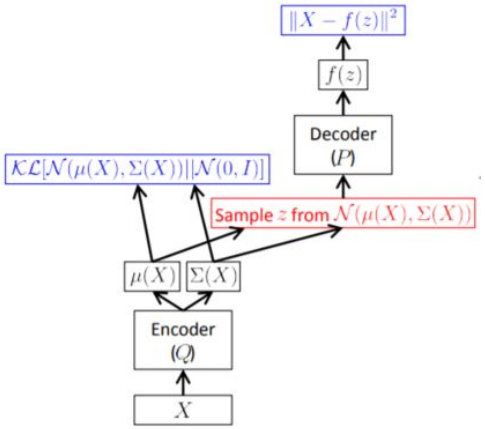
Idea: learn by trying to generate similar samples

- Variations autoencoders (VAE)
- Generative adversarial networks (GAN)
- And many others...

Generative

Idea: learn by trying to generate similar samples

- Variations autoencoders (VAE)
- Generative adversarial networks (GAN)
- And many others...



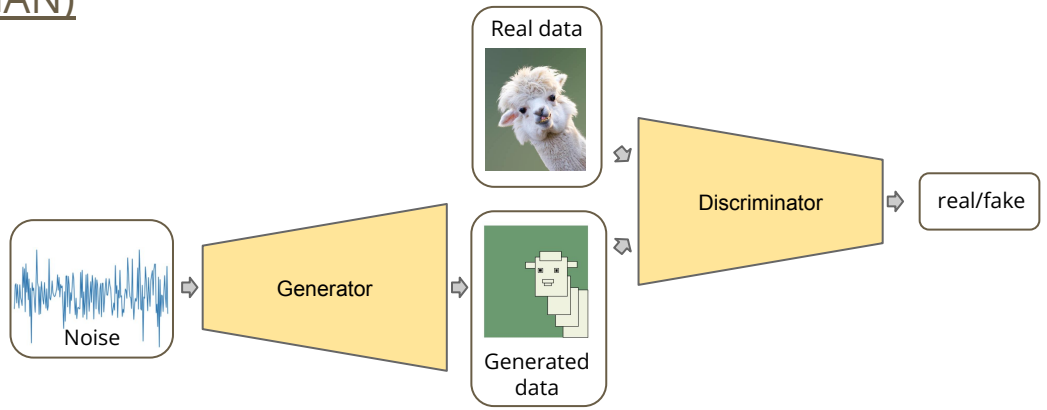
VAE diagram

Original VAE paper: https://arxiv.org/abs/1312.6114

Generative

Idea: learn by trying to generate similar samples

- Variations autoencoders (VAE)
- Generative adversarial networks (GAN)
- And many others...

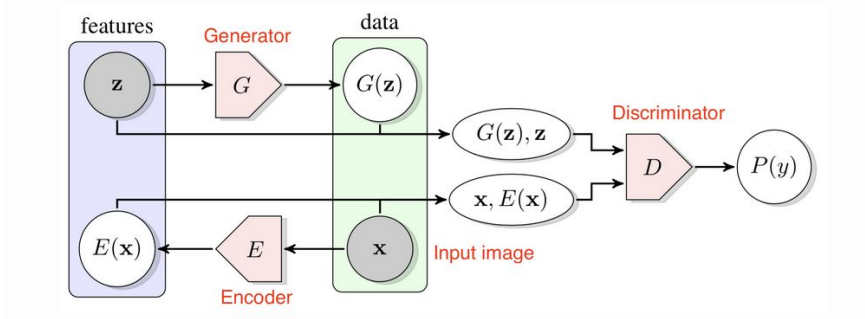


Original gan paper: https://arxiv.org/abs/1406.2661

Generative

Idea: learn by trying to generate similar samples

- Variations autoencoders (VAE)
- Generative adversarial networks (GAN)
- And many others…

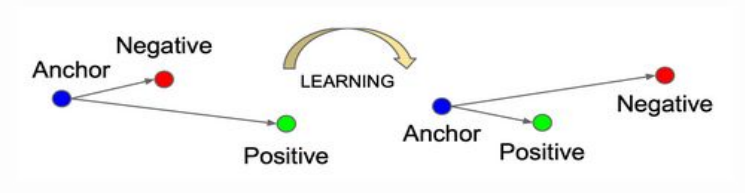


Bidirectional GAN

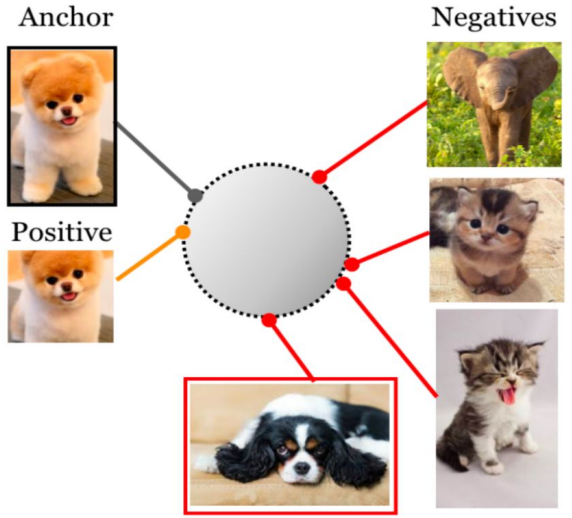
https://arxiv.org/abs/1605.09782

Contrastive methods

Basic idea -- to learn representations which are close for similar samples and distinct for the not similar ones.



Relies on positive and negative samples for a given anchor.



Can be used in the supervised setting.

Earliest approaches

Contrastive loss:

$$\mathcal{L}_{\text{cont}}(\mathbf{x}_i, \mathbf{x}_j, \theta) = 1[y_i = y_j] \|f_\theta(\mathbf{x}_i) - f_\theta(\mathbf{x}_j)\|_2^2 + 1[y_i \neq y_j] \max(0, \epsilon - \|f_\theta(\mathbf{x}_i) - f_\theta(\mathbf{x}_j)\|_2)^2$$

yann.lecun.com/exdb/publis/pdf/chopra-05.pdf

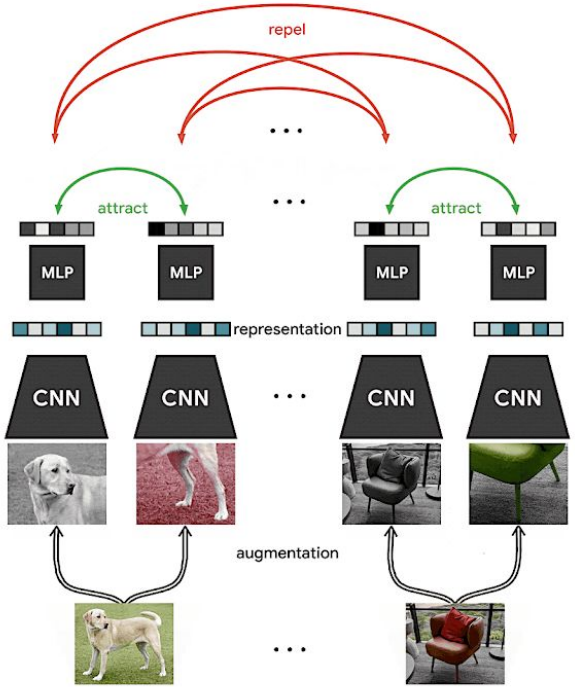
Generalizations: triplet loss, N-pair loss, Lifted Structured Loss

New losses: Mutual information, InfoNCE

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 $k \in \{1, \dots, N\}$ **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 $i \in \{1, \dots, 2N\}$ and $j \in \{1, \dots, 2N\}$ **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 $g(\cdot)$

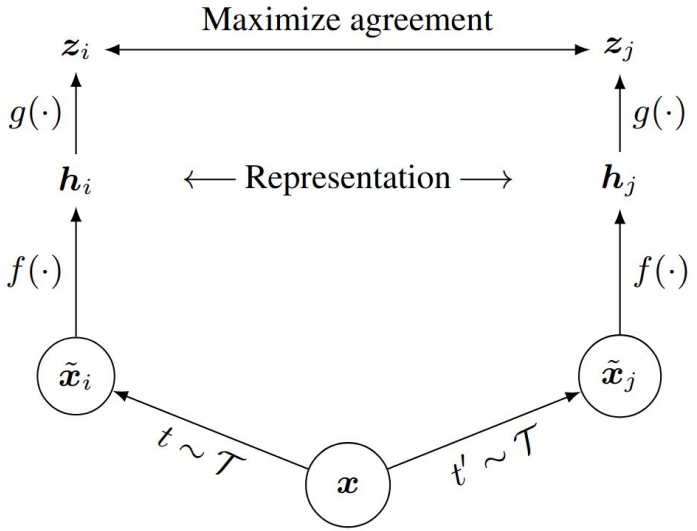


Ting Chen, Simon Kornblith, Mohammad Norouzi, Geoffrey Hinton "A Simple Framework for Contrastive Learning of Visual Representations"
https://arxiv.org/abs/2002.05709

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 $k \in \{1, \dots, N\}$ **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 $i \in \{1, \dots, 2N\}$ and $j \in \{1, \dots, 2N\}$ **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 $g(\cdot)$



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

Ting Chen, Simon Kornblith, Mohammad Norouzi, Geoffrey Hinton "A Simple Framework for Contrastive Learning of Visual Representations"
https://arxiv.org/abs/2002.05709

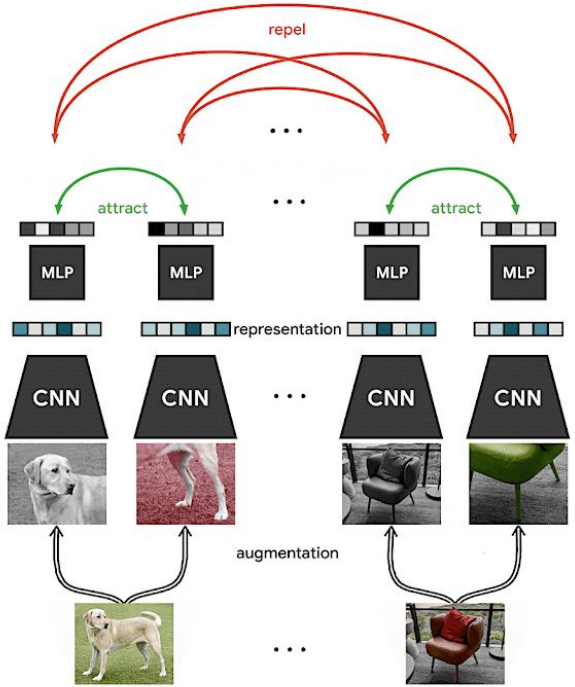
SimCLR

Pros:

- Simple
- Effective

Cons:

- Needs really big batch size

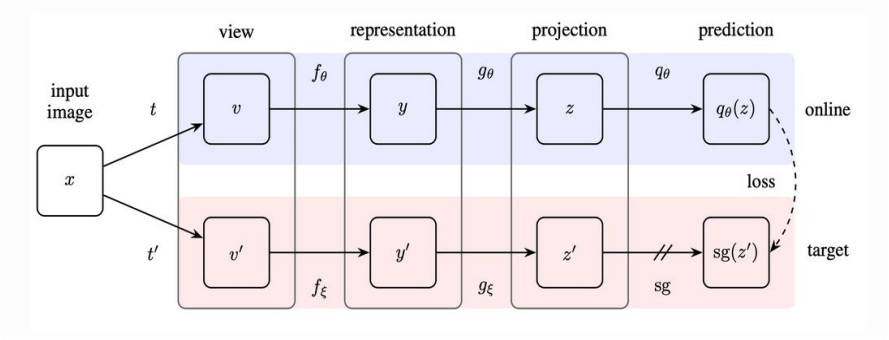


Ting Chen, Simon Kornblith, Mohammad Norouzi, Geoffrey Hinton “A Simple Framework for Contrastive Learning of Visual Representations” https://arxiv.org/abs/2002.05709

BYOL: Bootstrap Your Own Latent

Has two networks: *target(T)* and *online(O)*

1. Generate 2 views(augs): $v = t(x)$ and $v' = t'(x)$
2. Calculate projections $z = T(v)$ and $z' = O(v')$
3. Normalize: $\bar{z} = z/\|z\|$ and $\bar{z}' = z'/\|z'\|$
4. Compute $\mathcal{L}_{12} = MSE\,(\bar{z}, \bar{z}')$
5. Switch v and v' and compute $\mathcal{L}_{21}$ in the same manner
6. Compute final loss $\mathcal{L}{=}\mathcal{L}_{12} + \mathcal{L}_{21}$
7. Backprop to O and not to T
8. Update weigts of T as moving average of O: $\omega_t \leftarrow \tau\omega_t + (1-\tau)\omega_o$



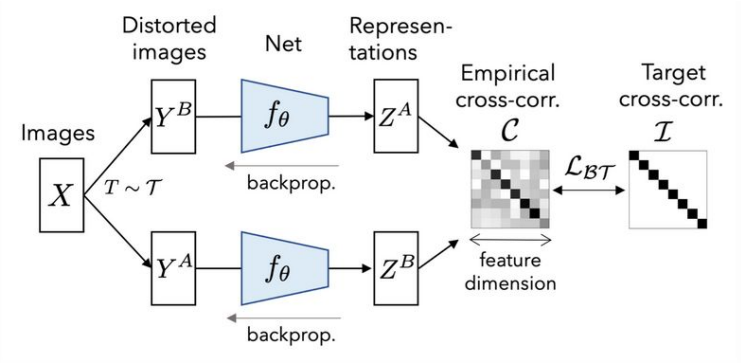
Bonus: doesn't need negative pairs

Grill, Jean-Bastien, et al. "Bootstrap your own latent: A new approach to self-supervised learning."
https://arxiv.org/abs/2006.07733

Barlow Twins

Idea:

1. Train representations for features
2. Optimise cross-correlation matrix for the representations:
   a. Maximize correlation between representations of distort images
   b. Minimize redundancy of the representations



Correlation matrix is optimized to be close to unity.

Bonus: robust to training batch size

$$\mathcal{L}_{\mathrm{BT}} = \underbrace{\sum_i (1 - \mathcal{C}_{ii})^2}_{\text{invariance term}} + \lambda \underbrace{\sum_i \sum_{i \neq j} \mathcal{C}_{ij}^2}_{\text{redundancy reduction term}}$$

$$\text{where } \mathcal{C}_{ij} = \frac{\sum_b \mathbf{z}_{b,i}^A \mathbf{z}_{b,j}^B}{\sqrt{\sum_b (\mathbf{z}_{b,i}^A)^2} \sqrt{\sum_b (\mathbf{z}_{b,j}^B)^2}}$$

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, Stéphane Deny, “Barlow Twins: Self-Supervised Learning via Redundancy Reduction”
https://arxiv.org/abs/2103.03230

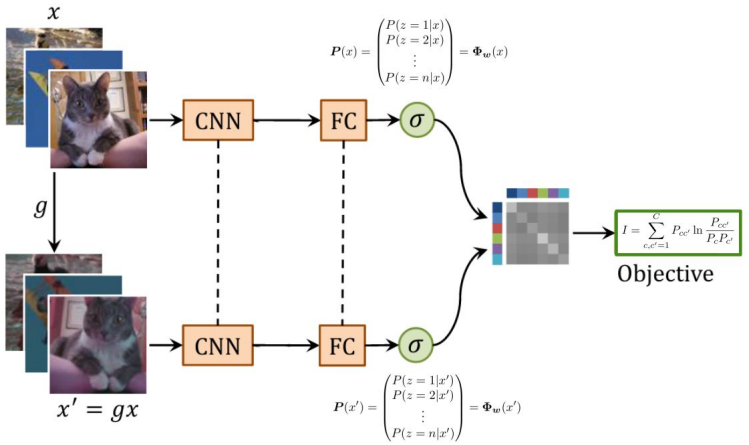
Invariant information clustering

Estimates joint probability distribution and computes mutual information.

Joint probability distribution: $P_{cc'} = \frac{1}{n}\sum_{i=1}^{n} P(z=c|x_i)P(z=c'|x_i)$

Marginals: $P_c = \sum_{c'=1}^{C} P_{cc'}$ $\quad P_{c'} = \sum_{c=1}^{C} P_{cc'}$

Mutual information: $I(x,x') = \sum_{c,c'=1}^{C} P_{cc'} \ln \frac{P_{cc'}}{P_c P_{c'}}$



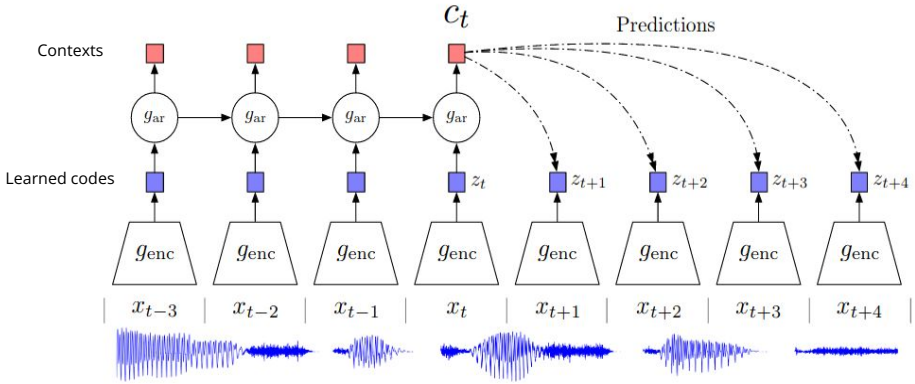
Xu Ji, João F. Henriques, Andrea Vedaldi “Invariant Information Clustering for Unsupervised Image Classification and Segmentation”
https://arxiv.org/abs/1807.06653

My demo: https://github.com/vandedok/IIC_tutorial

Contrastive predictive coding (time series)

1. Calculate codes for the fragments (z_t)
2. Take timestep T
3. Predict contexts (c_t) using on past fragments
4. Maximise mutual information between predicted codes and true future codes
5. Repeat for all T

As the future codes are not predicted directly,
MI cannot be estimated.
But it's possible to estimate its lower bound.



Aaron van den Oord, Yazhe Li, Oriol Vinyals "Representation Learning with Contrastive Predictive Coding "
https://arxiv.org/abs/1807.03748

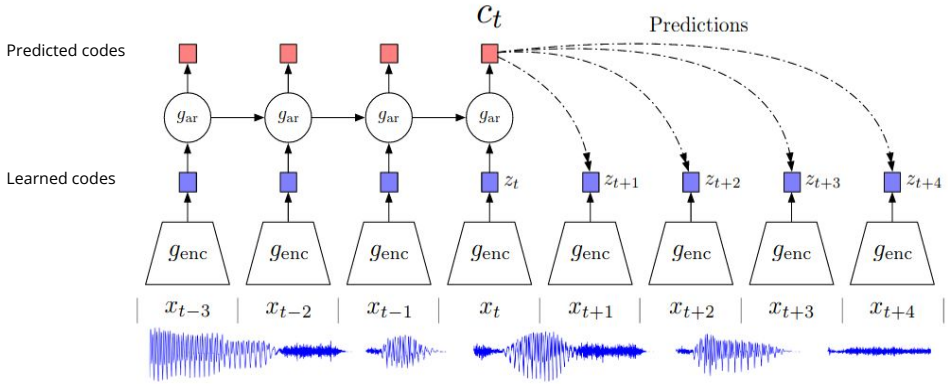
Contrastive predictive coding (time series)

1. Calculate codes for the fragments (z_t)
2. Take timestep T
3. Predict contexts (c_t) using on past fragments
4. Maximise MI lower bound between predicted codes and true future codes
5. Repeat for all T

InfoNCE loss:

Density estimate: $f_k(x_{t+k}, c_t) = \exp(z_{t+k}^\top W_k c_t) \propto \frac{p(x_{t+k}|c_t)}{p(x_{t+k})}$

MI lower bound: $\mathcal{L}_N = -\mathbb{E}_X\left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{i=1}^N f_k(x_i, c_t)}\right]$



Aaron van den Oord, Yazhe Li, Oriol Vinyals “Representation Learning with Contrastive Predictive Coding “ https://arxiv.org/abs/1807.03748

Contrastive predictive coding (time series)

1. Calculate codes for the fragments (z_t)
2. Take timestep T
3. Predict contexts (c_t) using on past fragments
4. Maximise MI lower bound between predicted codes and true future codes
5. Repeat for all T

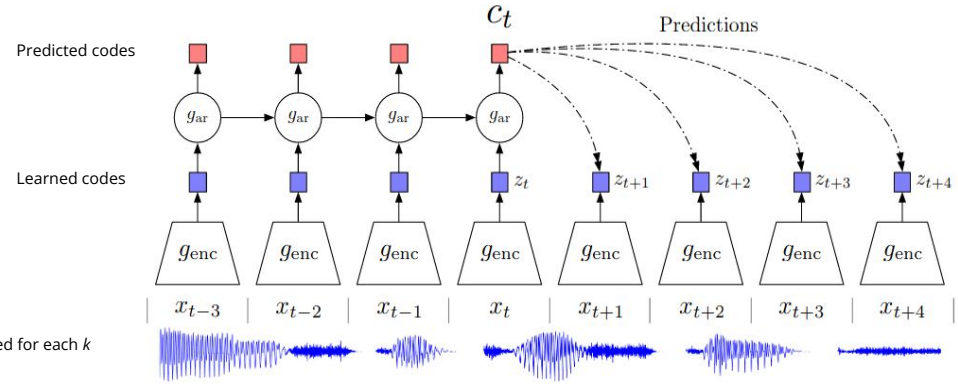
InfoNCE loss:

Density estimate: $f_k(x_{t+k}, c_t) = \exp(z_{t+k}^\top W_k c_t) \propto \frac{p(x_{t+k}|c_t)}{p(x_{t+k})}$

MI lower bound: $\mathcal{L}_N = -\mathbb{E}_X\left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{i=1}^N f_k(x_i, c_t)}\right]$

Also learned for each *k*

Contains one positive and N-1 negative samples.
There are different sampling strategies



Aaron van den Oord, Yazhe Li, Oriol Vinyals “Representation Learning with Contrastive Predictive Coding “
https://arxiv.org/abs/1807.03748

Contrastive predictive coding (images)

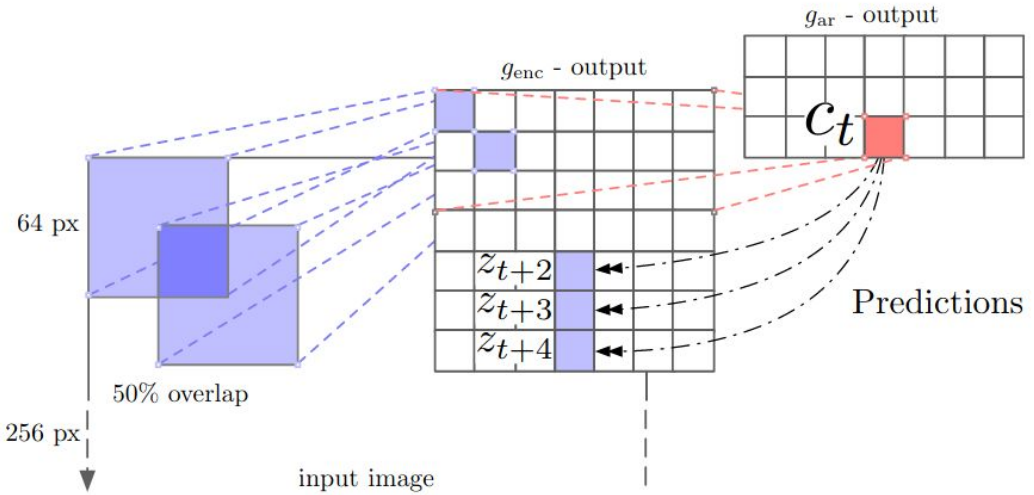
Differences for images:

g_{ar} -- masked ConvNet

(sees only what above the given position)

Positive samples -- patches below

Negative samples -- other patches from this images and from other images



Aaron van den Oord, Yazhe Li, Oriol Vinyals “Representation Learning with Contrastive Predictive Coding “ https://arxiv.org/abs/1807.03748

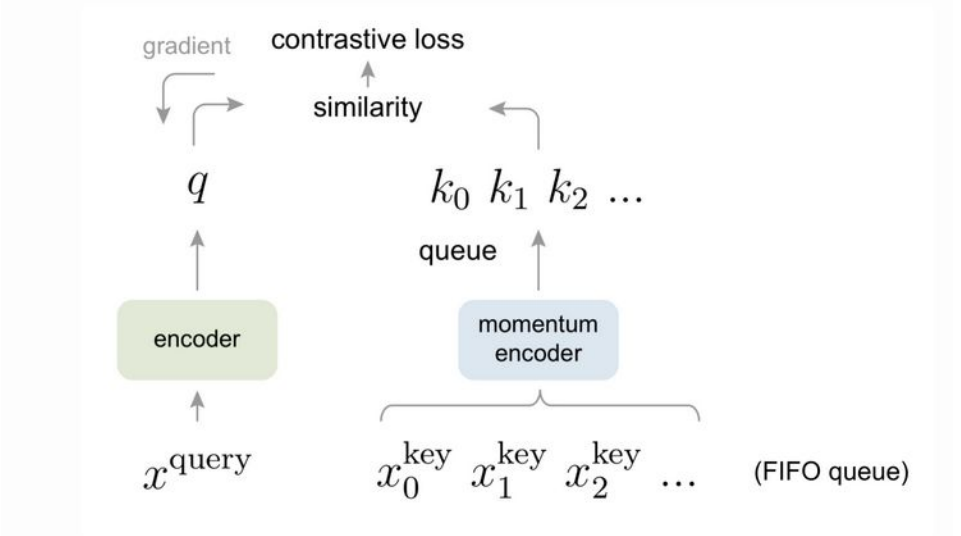
Momentum Contrast

Problem: to get enough negative samples big batch is required

Idea: store representations from previous batches

InfoNCE loss:

$$\mathcal{L}_{\text{MoCo}} = -\log \frac{\exp(\mathbf{q} \cdot \mathbf{k}^{+}/\tau)}{\sum_{i=1}^{N} \exp(\mathbf{q} \cdot \mathbf{k}_i/\tau)}$$

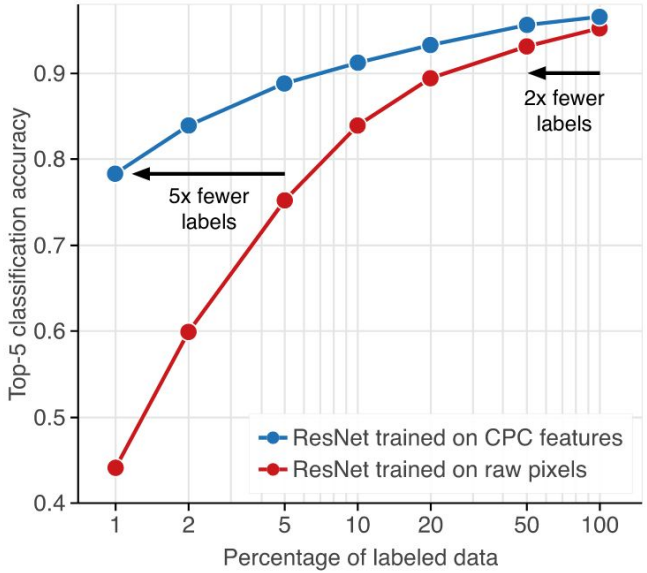


Update rule for momentum encoder:

$$\omega_{m.enc} \leftarrow m\omega_{m.enc} + (1-m)\omega_{enc}$$

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, Ross Girshick "Momentum Contrast for Unsupervised Visual Representation Learning"
https://arxiv.org/abs/1911.05722

Benchmarking



Olivier J. Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, S. M. Ali Eslami, Aaron van den Oord
"Data-Efficient Image Recognition with Contrastive Predictive Coding"
https://arxiv.org/abs/1905.09272

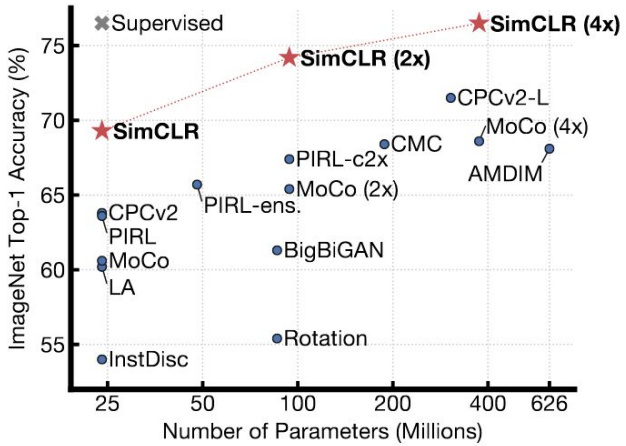


Figure 1. ImageNet Top-1 accuracy of linear classifiers trained on representations learned with different self-supervised methods (pretrained on ImageNet). Gray cross indicates supervised ResNet-50. Our method, SimCLR, is shown in bold.

Ting Chen, Simon Kornblith, Mohammad Norouzi, Geoffrey Hinton
"A Simple Framework for Contrastive Learning of Visual Representations"
https://arxiv.org/abs/2002.05709

Conclusion

- DL models can learn without annotation
- In many ways, actually
- And usually they are better, than transfer learning from other domains
- But they take time and resources
- Engineering tricks are crucial
- Evaluation usually requires some external knowledge

Thanks!

Further read

Dyakonov post:

https://dyakonov.org/2020/06/03/%D1%81%D0%B0%D0%BC%D0%BE%D0%BE%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5-self-supervision/

Self-supervised and contrastive methods reviews from Lilian Weng:

https://lilianweng.github.io/lil-log/2019/11/10/self-supervised-learning.html#contrastive-predictive-coding
https://lilianweng.github.io/lil-log/2021/05/31/contrastive-representation-learning.html