

Deep Learning for Audio

Lecture 8

Pavel Severilov

AI Masters

2023

Outline

1. Self-Supervised Learning: introduction
2. Auto-encoding frameworks
3. Siamese architectures
4. Contrastive Representation Learning

Outline

1. Self-Supervised Learning: introduction
2. Auto-encoding frameworks
3. Siamese architectures
4. Contrastive Representation Learning

Why Self-Supervised Learning?

- ▶ Huge amount of unlabeled data
- ▶ Expensive labeling
- ▶ Solve many task at once
- ▶ Features should be general — not specialized towards a single supervised task

SSL examples from Computer Vision

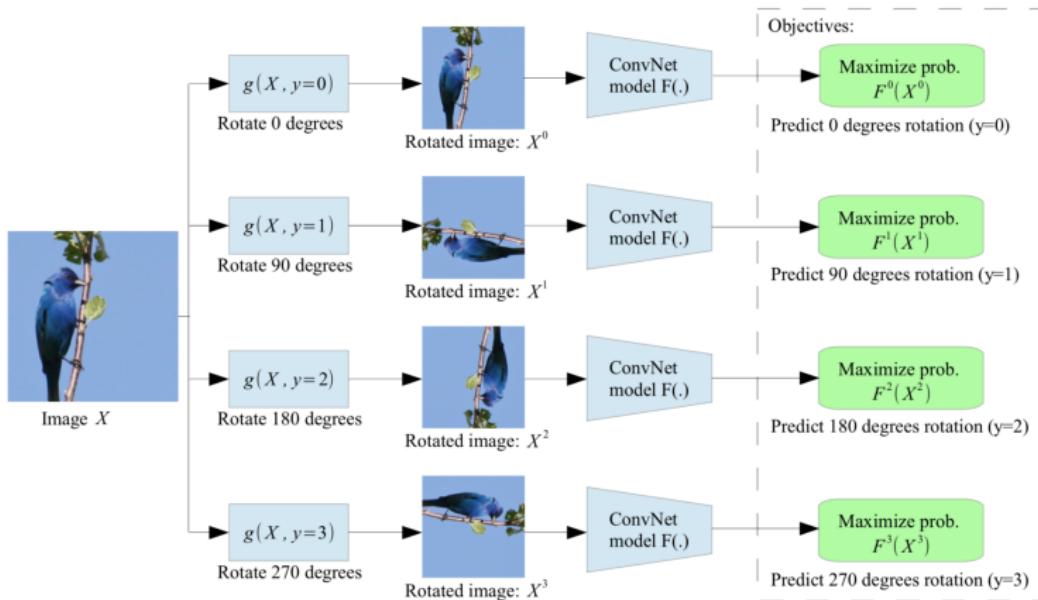


Figure: SSL by rotating the entire input images. The model learns to predict which rotation is applied

SSL examples from Computer Vision

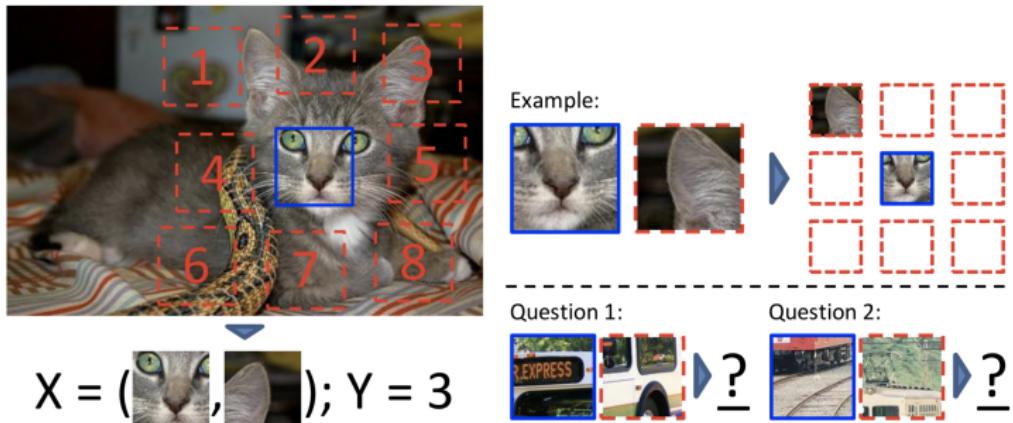
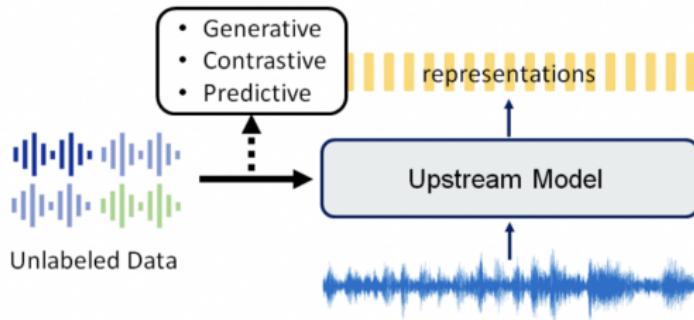


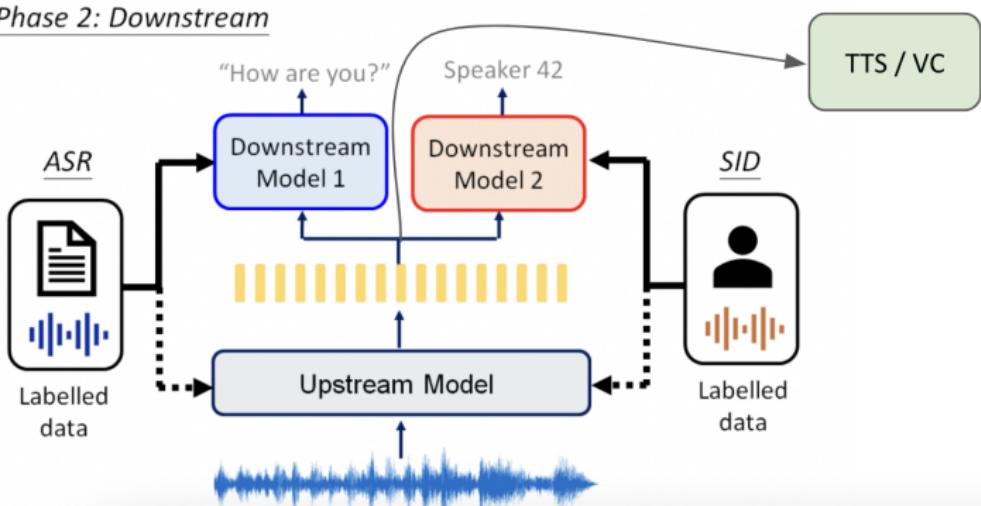
Figure: SSL by predicting the relative position of two random patches.

SSL pipeline

Phase 1: Pre-train



Phase 2: Downstream



SSL frameworks

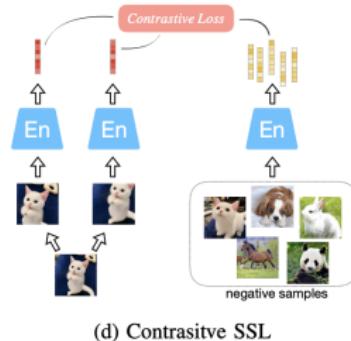
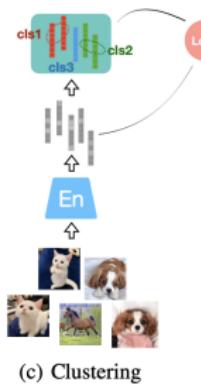
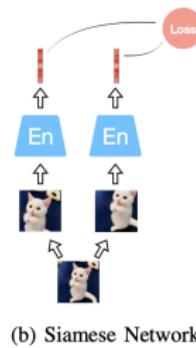
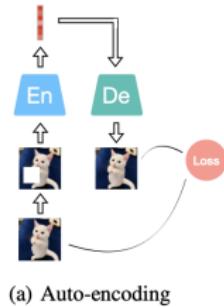
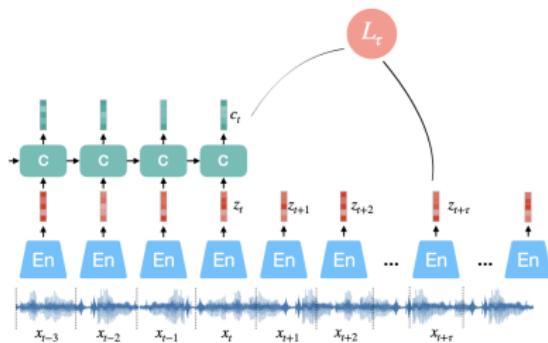


Figure: Predictive SSL frameworks (a-c) and contrastive SSL framework (d)

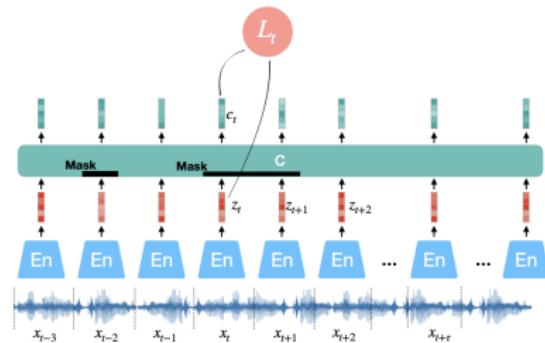
Outline

1. Self-Supervised Learning: introduction
2. Auto-encoding frameworks
3. Siamese architectures
4. Contrastive Representation Learning

APC, MPC



(a) APC



(b) MPC

Figure: Diagrams of auto-regressive predictive coding (APC) and masked predictive coding (MPC).

Outline

1. Self-Supervised Learning: introduction
2. Auto-encoding frameworks
3. Siamese architectures
4. Contrastive Representation Learning

Siamese architectures

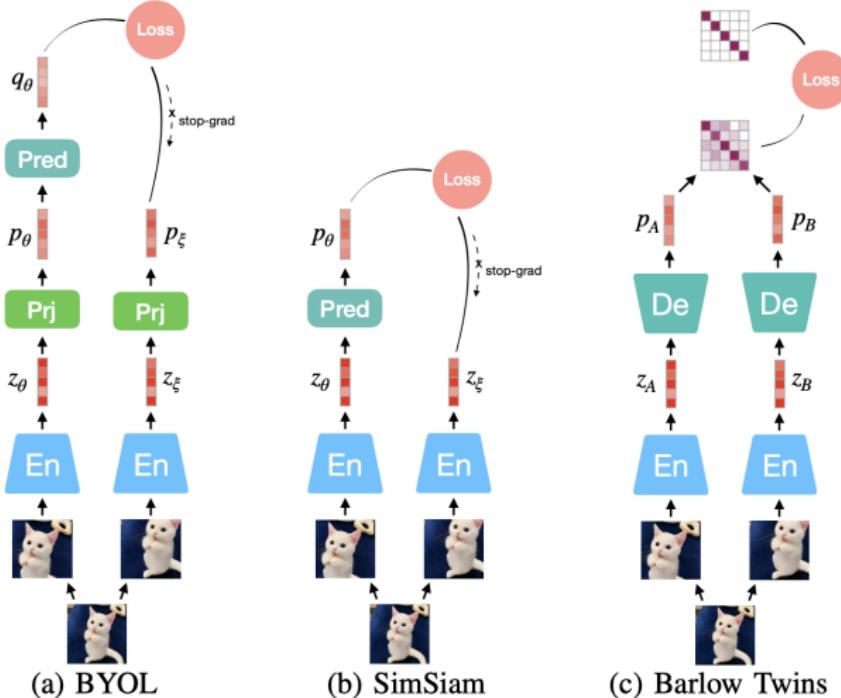


Figure: Diagrams for Predictive Models using Siamese architectures

Outline

1. Self-Supervised Learning: introduction
2. Auto-encoding frameworks
3. Siamese architectures
4. Contrastive Representation Learning

Contrastive Loss

- ▶ We would like to learn a function $f_\theta(\cdot) : \mathcal{X} \rightarrow \mathbb{R}^d$ that encodes samples x_i into an embedding vectors
- ▶ Contrastive loss takes a pair of inputs and minimizes the embedding distance when they are from the same class but maximizes the distance otherwise.
- ▶

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

where ϵ – hyperparameter, the lower bound distance between samples of different classes.

Triplet Loss

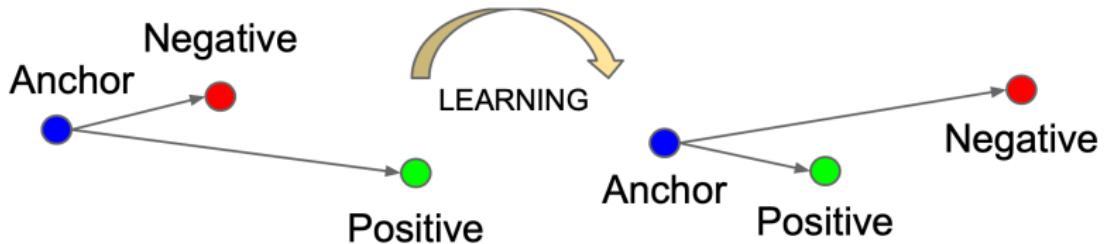


Figure: Illustration of triplet loss given one positive and one negative per anchor.

$$\begin{aligned}\mathcal{L}_{\text{triplet}}(\mathbf{x}, \mathbf{x}^+, \mathbf{x}^-) &= \\ &= \sum_{\mathbf{x} \in \mathcal{X}} \max \left(0, \|f(\mathbf{x}) - f(\mathbf{x}^+)\|_2^2 - \|f(\mathbf{x}) - f(\mathbf{x}^-)\|_2^2 + \epsilon \right)\end{aligned}$$

where ϵ – minimum offset between distances of similar vs dissimilar pairs.

InfoNCE loss

- ▶ Uses categorical cross-entropy loss to identify the positive sample amongst a set of unrelated noise samples
- ▶ Given a context vector c , the positive sample should be drawn from the conditional distribution $p(x|c)$, while $N - 1$ negative samples are drawn from the proposal distribution $p(x)$, independent from the context c
- ▶ InfoNCE loss:

$$\mathcal{L}_N = -\mathbb{E}_X \left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)} \right]$$

InfoNCE loss

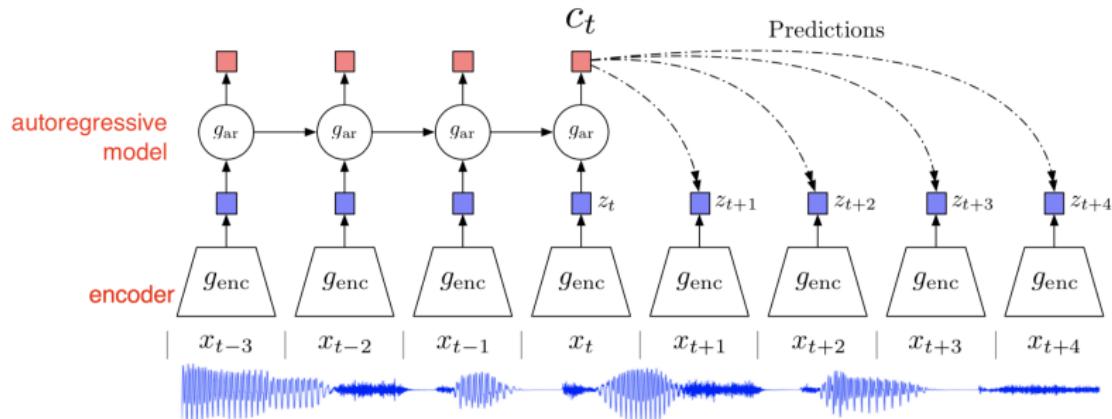
The InfoNCE loss optimizes the negative log probability of classifying the positive sample correctly

The probability of detecting the positive sample correctly is:

$$\begin{aligned} p(C = \text{pos} | X, \mathbf{c}) &= \frac{p(x_{\text{pos}} | \mathbf{c}) \prod_{i=1, \dots, N; i \neq \text{pos}} p(x_i)}{\sum_{j=1}^N \left[p(x_j | \mathbf{c}) \prod_{i=1, \dots, N; i \neq j} p(x_i) \right]} = \\ &= \frac{\frac{p(x_{\text{pos}} | \mathbf{c})}{p(x_{\text{pos}})}}{\sum_{j=1}^N \frac{p(x_j | \mathbf{c})}{p(x_j)}} = \frac{f(x_{\text{pos}}, \mathbf{c})}{\sum_{j=1}^N f(x_j, \mathbf{c})} \end{aligned}$$

where the scoring function is $f(\mathbf{x}, \mathbf{c}) \propto \frac{p(\mathbf{x} | \mathbf{c})}{p(\mathbf{x})}$.

Contrastive Predictive Coding (CPC)



- ▶ Non-linear encoder maps input sequence to a sequence of latent representations
- ▶ Autoregressive model produces context representation from all history
- ▶ Scoring function to maximize mutual information between context and future target
- ▶ The end-to-end training relies on the NCE-inspired contrastive loss.

Contrastive Predictive Coding (CPC)

Rather than modeling the future observations $p_k(x_{t+k} | c_t)$ directly, CPC models a density function to preserve the mutual information between x_{t+k} and c_t :

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

Mutual Information:

$$I(x; c) = \sum_{x,c} p(x, c) \log \frac{p(x | c)}{p(x)}$$

$$I(x_{t+k}, c_t) \geq \log(N) - \mathcal{L}_N$$