Machine Learning (CE 40717)

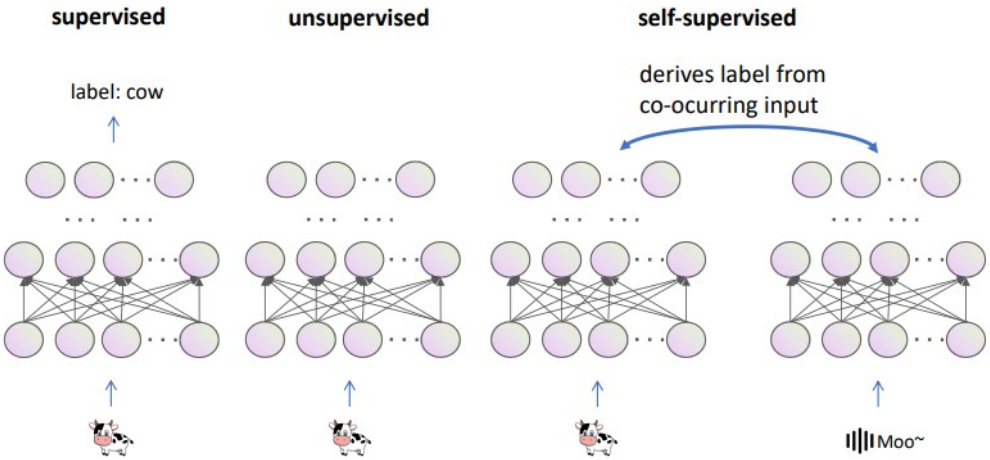
Fall 2024

Ali Sharifi-Zarchi

CE Department
Sharif University of Technology

December 31, 2024

Setting


supervised
unsupervised
self-supervised
label: cow
derives label from
co-ocurring input
Moo~

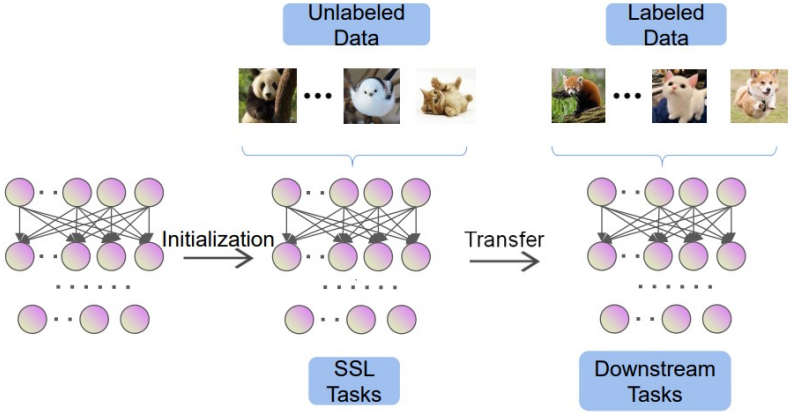
What is SSL

- Self-supervised learning defines a "pretext" task based on unlabeled inputs to produce descriptive and intelligible representations [Hastie et al., 2009, Goodfellow et al., 2016]
  - Labels of these pretext tasks are generated *automatically*
  - Can be used in other downstream tasks.

Motivation

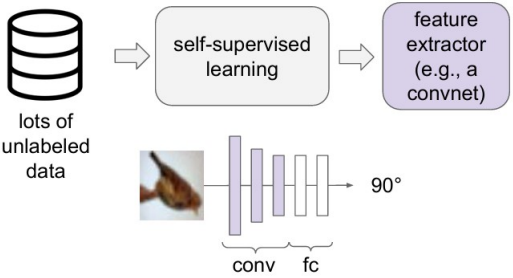
[reference]

General Pipeline



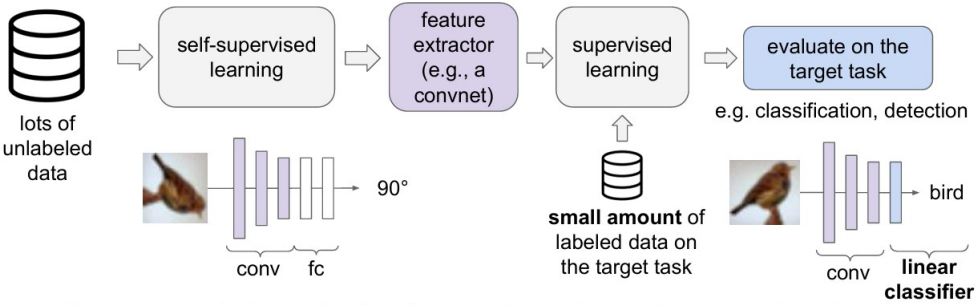
[reference]

Evaluation Cont.



1. Learn good feature extractors from self-supervised pretext tasks, e.g., predicting image rotations

Evaluation Cont.



1. Learn good feature extractors from self-supervised pretext tasks, e.g., predicting image rotations

2. Attach a shallow network on the feature extractor; train the shallow network on the target task with small amount of labeled data

Applications

- **NLP**
  - Models like BERT, GPT, and T5 are based on SSL. They are pre-trained on massive text corpora using tasks like masked language modeling.
  - Improving machine translation quality with SSL-based pre-training on multilingual corpora.
- **Computer Vision**
  - Models like **SimCLR** and BYOL achieve state-of-the-art performance in image classification by learning from unlabeled images.
  - Learning representations for tasks like disease diagnosis from limited annotated medical datasets.

Applications (Cont.)

- **Speech and Audio Processing**
  - Models like Wav2Vec and HuBERT use SSL in speech recognition task to learn representations from raw audio data. It's especially useful for languages with limited data.
  - Identifying individuals using voice features learned through SSL in speaker identification.
- **Robotics**
  - Leveraging SSL in reinforcement learning helps learn useful state representations for control tasks, enabling robots to perform complex actions.

Applications (Cont.)

- **Healthcare**
  - Learning representations of DNA sequences to predict mutations or functional regions.
  - Identifying diseases from clinical notes, imaging data, or time-series data such as ECGs.
- **Autonomous Vehicles**
  - Integrating data from cameras, LiDAR, and radar through SSL.
  - Using SSL for anomaly detection in vehicle systems.

Applications - Rotation (Cont.) [1] [2]

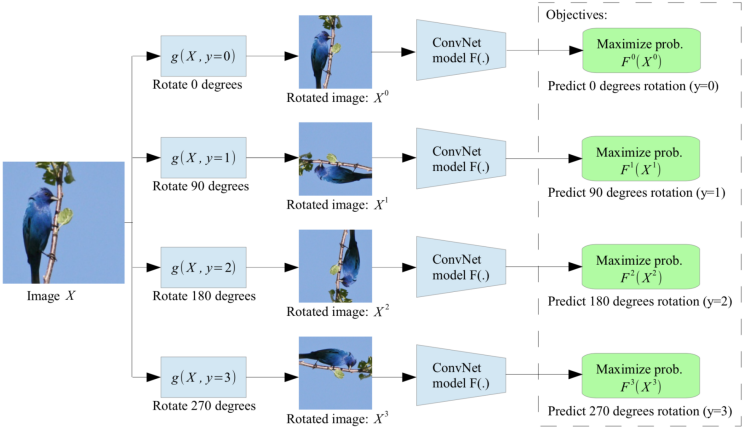


Figure 1: The model learns to predict which rotation is applied.

[1] Some great examples

[2] Unsupervised Representation Learning by Predicting Image Rotations

Applications - Patches (Cont.) [3] [4]

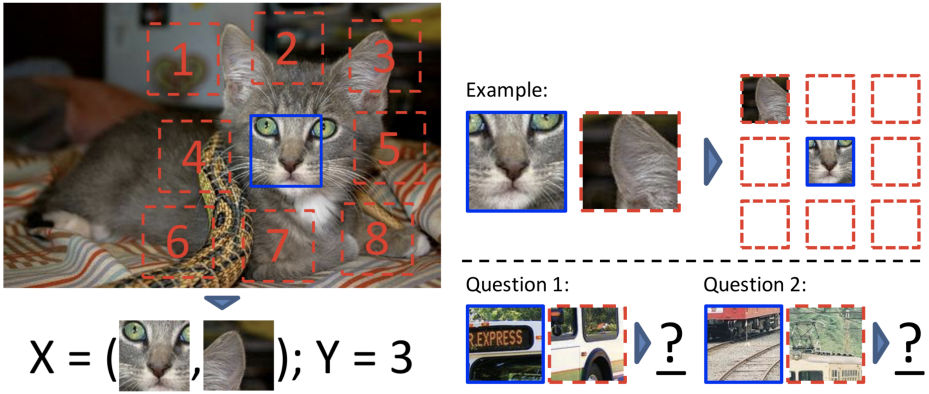


Figure 2: The model learns to predict the relative position of two random patches.

[3] Some great examples

[4]

Applications - Patches (Cont.) [5] [6]

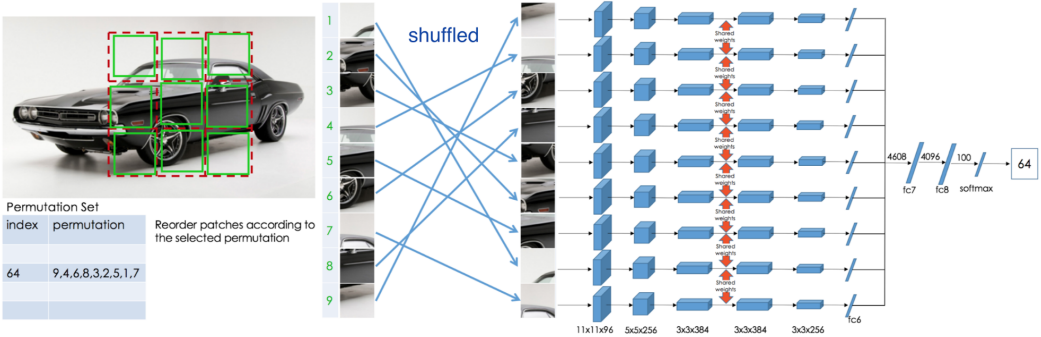


Figure 3: The model learns to solve jigsaw puzzle

[5] Some great examples

[6] Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles.

Applications - Tracking (Cont.) [7] [8]

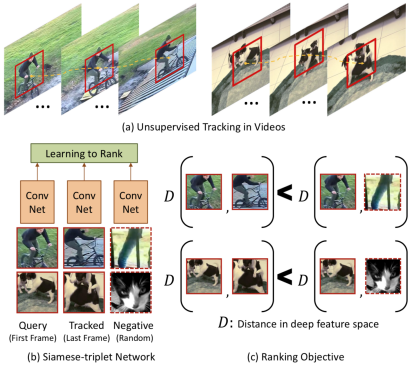


Figure 4: The model learns to track objects in videos.

[7] Some great examples

[8] Unsupervised Learning of Visual Representations using Videos

Idea

- Don't learn from isolated images – take images together with some **context**
- **Video**: Image together with adjacent video frames
- **Sound**: Image with audio track from video
- **3D Image**: Image with depth map or point cloud
- **Language:** Image with natural-language text (e.g., captions or descriptions)

Why Language?

- **Rich Semantics**
  - Just a few words give rich information.
  - Acts as a bridge between sensory data and abstract human understanding.
- **Universality**
  - Language can describe almost any concept
  - Language can act as a **universal medium** for aligning other modalities, even structured data.

Why Language? (Cont.)

- **Large-Scale Data Availability**
  - The internet contains vast amounts of textual data.
  - Text data is relatively easier to collect, clean, and annotate (no need to experts) compared to modalities like video or audio.
  - Available datasets such as COCO (images and captions)
- **Pretrained Language Models (PLMs) as a Strong Foundation**
  - Large pretrained language models with remarkable capabilities.
  - Language models are highly transferable (transfer learning) across tasks, enabling multimodal systems to adapt to various downstream applications efficiently.

Definition

- A machine learning technique for training models to distinguish between similar and dissimilar data points.
- **Key Idea**
  - Bring similar data points closer in the embedding space.
  - Push dissimilar data points farther apart.
- **Purpose:** Learn meaningful representations for downstream tasks like classification, clustering, or retrieval
- **Widely Used In:** Representation learning across domains such as computer vision, NLP, and multi-modal tasks.

Key Concepts

- **Embedding Space**
  - The data points are mapped into a high-dimensional space, called the embedding space.
  - Their relative positions encode similarity or dissimilarity.
- **Positive Pairs:** Data points that are semantically similar.
- **Negative Pairs:** Data points that are semantically different.
- **Objectives**
  - Minimize the distance between the embeddings of positive pairs.
  - Maximize the distance between the embeddings of negative pairs.

Common Components

- Dataset :
  - supervised: $D_m = \{(x_1^1, \cdots, x_M^1, y^1), \cdots, (x_1^n, \cdots, x_M^n, y^n)\}$
  - self-supervised: $D_m = \{(x_1^1, \cdots, x_M^1), \cdots, (x_1^n, \cdots, x_M^n)\}$
- The psudo-label or signal generated for SSL can be denoted as $z = P(x_1, \cdot, x_M)$.
- Modality Encoder(s): $c = e_k(x_k^i; \theta_k)$ for each modality k.
- Fusion Module: f_ψ to integrate the encoded information of different modalities
- Pretext task head (like a predictive head) : g_γ and some SSL loss $\mathscr{L}_{SSL}$

Architectures

- There many veriations and structures

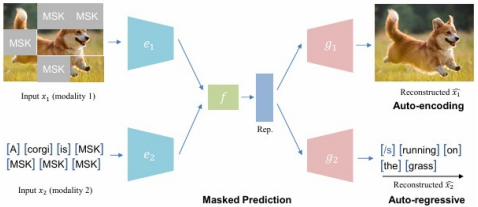


Figure 5: Figure 1 masked prediction frameworks

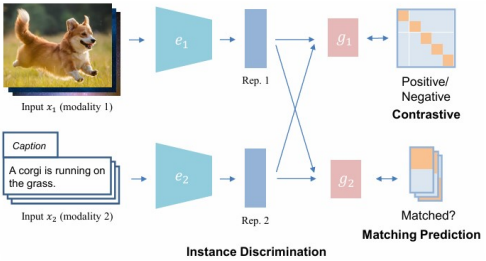


Figure 6: Figure 2 instance discrimination objectives

SimCLR [9] [10]

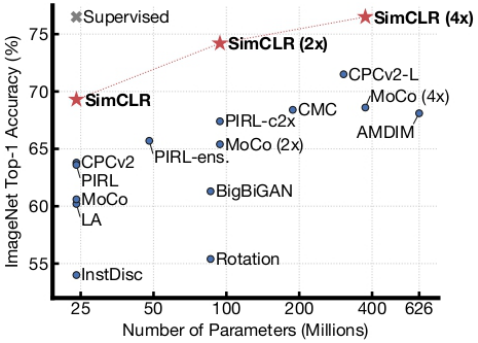


Figure 7: ImageNet Top-1 accuracy of linear classifiers trained on representations learned with different self-supervised methods (pretrained on ImageNet). Gray cross indicates supervised ResNet-50. SimCLR, is shown in bold.

[9] A Simple Framework for Contrastive Learning of Visual Representations

[10] Paper explained: A Simple Framework for Contrastive Learning of Visual Representations

SimCLR (Cont.)

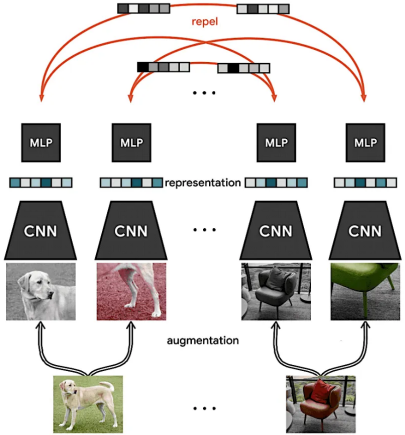


Figure 8: An illustration of the SimCLR training procedure.

SimCLR (Cont.)

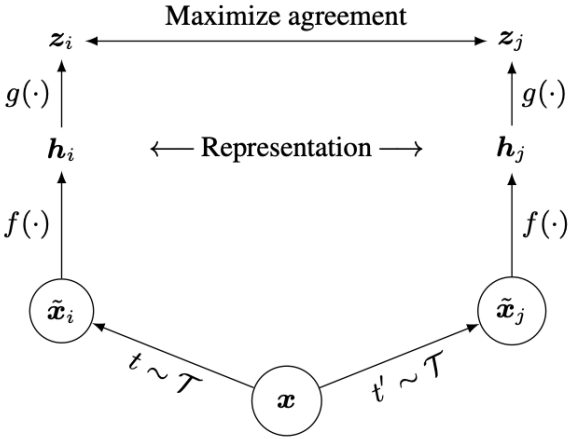


Figure 9: SimCLR overview

SimCLR (Cont.)

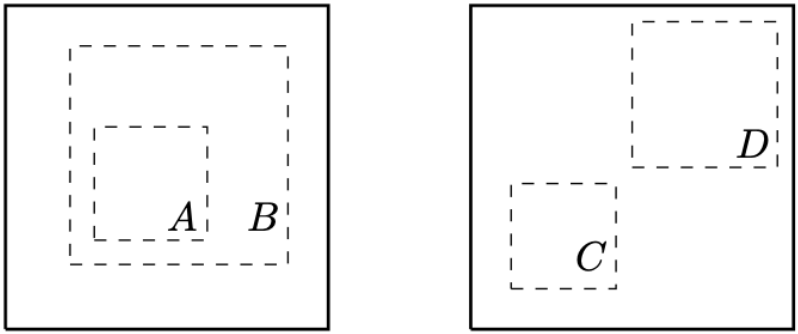


Figure 10: Solid rectangles are images, dashed rectangles are random crops. By randomly cropping images, we sample contrastive prediction tasks that include global to local view ($B \rightarrow A$) or adjacent view ($D \rightarrow C$) prediction.

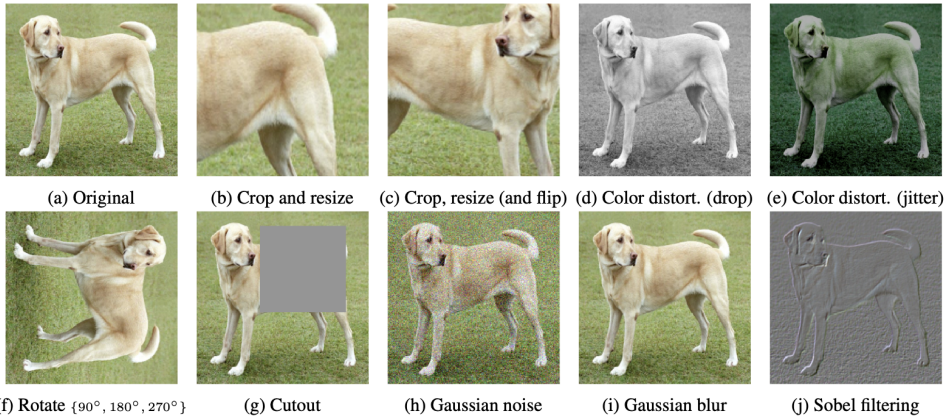
SimCLR (Cont.)

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 $g(\cdot)$

Figure 11: Summarization of SimCLR method.

SimCLR (Cont.)



(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

labelformat=empty

Figure 12: Figure 12

SimCLR (Cont.)

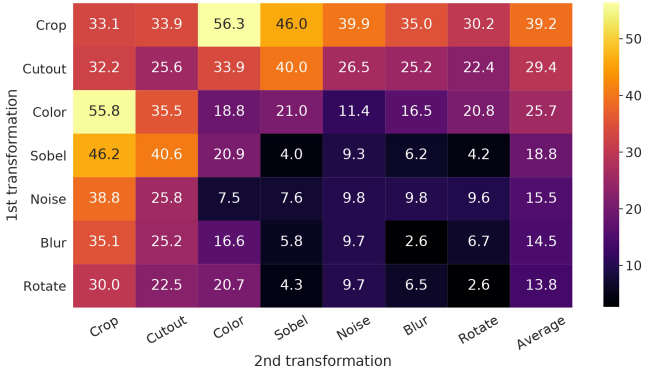


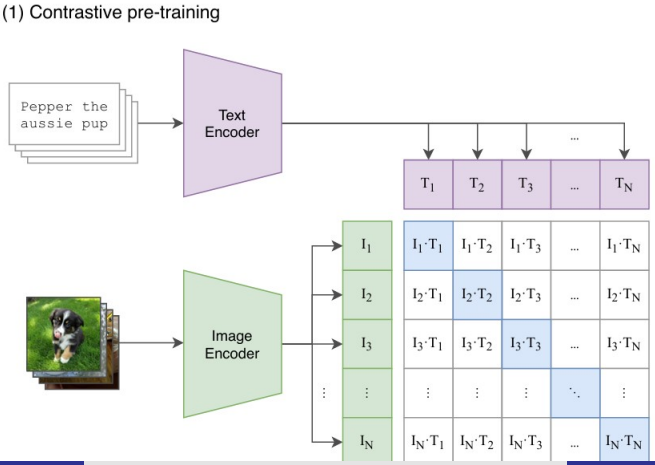
Figure 13: Linear evaluation (ImageNet top-1 accuracy) under individual or composition of data augmentations, applied only to one branch. For all columns but the last, diagonal entries correspond to single transformation, and off-diagonals correspond to composition of two transformations (applied sequentially). The last column reflects the average over the row.

CLIP

- Connecting text and images
- Contrastive Language–Image Pre-training
- CLIP $\Longrightarrow$ a shared representation(embedding) between two modalities (text and images) by training on a large dataset of image-text pairs.

CLIP Cont.

- Image Encoder: a Vision Transformer (ViT) or a ResNet.
- Text Encoder: A Transformer model



CLIP Goals

- CLIP was designed to mitigate a number of major problems:
- Costly datasets: Deep learning needs a lot of data, manually labeled datasets are expensive to construct.
  - CLIP learns from text–image pairs that are already publicly available on the internet
- Narrow: An ImageNet model excels at predicting the 1000 ImageNet categories but requires additional data and fine-tuning for other tasks.
  - CLIP can be adapted to perform a wide variety of visual classification tasks without needing additional training examples.

Loss Functions - Contrastive Loss [11] [12]

- Contrastive loss was first introduced in 2005 by Yann Le Cunn et al.
- Its original application was in Dimensionality Reduction.

[11] Dimensionality Reduction by Learning an Invariant Mapping
[12] Losses explained: Contrastive Loss

Loss Functions - Contrastive Loss (Cont.)

$$D_W\left(x'^{(i)}, x''^{(i)}\right) = \left\| G_W\left(x'^{(i)}\right) - G_W\left(x''^{(i)}\right) \right\|_2$$

- $D_W\left(x'^{(i)}, x''^{(i)}\right)$ is dissimilarity between the two data points $x'^{(i)}$ and $x''^{(i)}$.
- G_W is an embedding function parameterized by W.
- Generally, D_W can be any metric that indicates the dissimilarity between $x'^{(i)}$ and $x''^{(i)}$.

Loss Functions - Contrastive Loss (Cont.)

$$L\left(W, y^i, x'^{(i)}, x''^{(i)}\right) = (1 - y^i) L_S\left(D_W(x'^{(i)}, x''^{(i)})\right) + y^i L_D\left(D_W(x'^{(i)}, x''^{(i)})\right)$$

- $\left(y^i, x'^{(i)}, x''^{(i)}\right)$ is the i-th labeled sample pair.
- $Y = 0$ if $x'^{(i)}$ and $x''^{(i)}$ are deemed similar, and $Y = 1$ if they are deemed dissimilar.
- L_S is the partial loss function for a pair of similar points.
- L_D is the partial loss function for a pair of dissimilar points.
- L_S and L_D must be designed such that minimizing L with respect to W would result in low values of D_W for similar pairs and high values of D_W for dissimilar pairs.

Loss Functions - Contrastive Loss (Cont.)

$$\mathscr{L}(W) = \sum_{i=1}^{P} L\left(W, y^i, x'^{(i)}, x''^{(i)}\right)$$

- P is the number of training pairs (which may be as large as the square of the number of samples).

Loss Functions - InfoNCE Loss[13]

- First, we'll explore this loss from a theoretical perspective which has been discussed in its original paper.
- Next, we'll discuss how it can be applied in practice.
- It's the loss in its original paper:

$$\mathcal{L}_{\mathrm{N}} = -\mathbb{E}_X \left[\log \frac{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})}}{\sum_{x_j \in X} \frac{p(x_t|c_t)}{p(x_t)}} \right]$$

[13] Representation Learning with Contrastive Predictive Coding

Loss Functions - InfoNCE Loss (Cont.)

- Let's start with mutual information.
- We have a set $X = \{x_1, \ldots, x_N\}$ of N random samples containing one positive sample from $p(x_{t+k} \mid c_t)$ and $N-1$ negative samples from the **proposal** distribution $p(x_{t+k})$
- Our purpose is to maximize mutual information:

$$I(x_{t+k}; c_t) = \sum_{x_{t+k}, c_t} p(x_{t+k}, c_t) \log \frac{p(x_{t+k} \mid c_t)}{p(x_{t+k})}$$

- c_t is context latent representation.

Loss Functions - InfoNCE Loss (Cont.)

- We know:

$$I(x_{t+k}; c_t) \leq \log N \rightarrow I(x_{t+k}; c_t) \geq \log N - \mathscr{L}_{\mathrm{N}}$$

- $\mathscr{L}_{\mathrm{N}}$ quantifies the gap between the true mutual information and the approximation.
- Minimizing $\mathscr{L}_{\mathrm{N}}$ effectively maximizes the mutual information.

Loss Functions - InfoNCE Loss (Cont.)

- Categorical cross-entropy of classifying the positive sample correctly, with $\frac{f_k}{\sum_X f_k}$ being the prediction of the model.

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

- We want to optimize it.

Loss Functions - InfoNCE Loss (Cont.)

- Let's write the optimal probability for this loss as $p(d = i \mid X, c_t)$ with $[d = i]$ being the indicator that sample x_i is the **positive** sample.
- The probability that sample x_i was drawn from the conditional distribution $p(x_{t+k} \mid c_t)$ rather than the proposal distribution $p(x_{t+k})$ can be derived as follows:

$$p(d = i \mid X, c_t) = \frac{p(x_i \mid c_t) \prod_{l \neq i} p(x_l)}{\sum_{j=1}^{N} p(x_j \mid c_t) \prod_{l \neq j} p(x_l)} = \frac{\frac{p(x_i|c_t)}{p(x_i)}}{\sum_{j=1}^{N} \frac{p(x_j|c_t)}{p(x_j)}}$$

- As we can see, the optimal value for $f_k(x_{t+k}, c_t)$ in $\mathcal{L}_{\mathrm{N}}$ is proportional to $\frac{p(x_{t+k}|c_t)}{p(x_{t+k})}$ and this is independent of the choice of the number of negative samples $N - 1$.

Loss Functions - InfoNCE Loss (Cont.)

- We can evaluate the mutual information between the variables c_t and x_{t+k} as follows:

$$I(x_{t+k}, c_t) \geq \log(N) - \mathscr{L}_{\mathrm{N}}$$

- It becomes tighter as N becomes larger.
- Minimizing the InfoNCE loss $\mathscr{L}_{\mathrm{N}}$ maximizes a lower bound on mutual information.

Loss Functions - InfoNCE Loss (Cont.)

- In practice, we have:

$$\mathscr{L}_{\mathrm{N}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp\left(\mathrm{sim}\left(x_i, c_i\right)/\tau\right)}{\sum_{j=1}^{N}\exp\left(\mathrm{sim}\left(x_i, c_j\right)/\tau\right)}$$

- Used in models like SimCLR, MoCo, CLIP, and others.
- Next, we want to derive this formula from the theoretical one.

Loss Functions - InfoNCE Loss (Cont.)

- **Step 1:**

$$\frac{p(x \mid c)}{p(x)} = \exp\left(\log\left(\frac{p(x \mid c)}{p(x)}\right)\right)$$

- But in practice, we rarely know the true densities $p(x \mid c)$ and $p(x)$.
- Instead, we learn a function that approximates their log-ratio.
- A common approach is to let a neural network produce embeddings $f(x)$ and $g(c)$.

Loss Functions - InfoNCE Loss (Cont.)

$$\log\left(\frac{p(x \mid c)}{p(x)}\right) \approx \operatorname{sim}\left(f(x), g(c)\right) \xrightarrow{\text{we annotate it as}} \operatorname{sim}(x, c) \rightarrow \frac{p(x \mid c)}{p(x)} \approx \exp\left(\operatorname{sim}(x, c)\right) \tag{1}$$

- $\operatorname{sim}(x, c)$ is similarity function (e.g., cosine similarity or dot product).
- Replacing unknown densities with a similarity function, yielding a **softmax** function (which we'll discuss).
- It's straightforward to implement using standard deep-learning toolkits.

Loss Functions - InfoNCE Loss (Cont.)

- Why $\text{sim}(x, c)$ works?
  - It becomes large (positive) for the true **positive** pair (x, c).
  - It becomes relatively small (negative) for **negative** pairs (x, c').

$$\text{sim}(x, c) \gg \text{sim}(x, c') \longleftrightarrow p(x, c) \gg p(x, c')$$

  - This is the property required to approximate the ratio $p(x \mid c) / p(x)$.

Loss Functions - InfoNCE Loss (Cont.)

- **Step 2:**
- In practice, we don't have the full distribution X or its expectations.
- Instead, we approximate this using batches of size N.
- Each x_{t+k} is treated as the **positive sample**, and the other x_js in the batch are treated as **negative samples**.
- The expectation becomes a summation over batches:

$$\mathcal{L}_{\mathrm{N}} = -\mathbb{E}_X \left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)} \right] \approx -\frac{1}{N} \sum_{i=1}^{N} \log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)} \tag{2}$$

Loss Functions - InfoNCE Loss (Cont.)

- **Step 3:**
- To control the sharpness of the similarity distribution, a temperature parameter τ is introduced:

$$\text{sim}(x, c) \rightarrow \frac{\text{sim}(x, c)}{\tau} \tag{3}$$

- τ helps balance gradients during training:
  - With no τ, large similarity scores might dominate the gradients, leading to unstable updates.
  - A carefully chosen τ scales the scores appropriately, ensuring stable convergence.

Loss Functions - InfoNCE Loss (Cont.)

- τ affects the distribution of similarity scores after applying the softmax function; in other words, it influences the sharpness of the softmax.
- Low τ:
  - High sharpness.
  - The softmax heavily favors the largest score.
  - The distribution becomes more concentrated on the top-scoring pair.
  - Encourages the model to focus strongly on the positive sample while ignoring negatives.
  - The loss becomes more sensitive to small differences in scores.
- High τ:
  - Low sharpness.
  - The softmax smooths the distribution, making it more uniform.
  - This encourages the model to consider a broader set of samples, not just the top-scoring pair.
  - Useful when the data is noisy or when the model needs to generalize better.

Loss Functions - InfoNCE Loss (Cont.)

- **Finally:** From equations (1) to (3), we derive:

$$\mathscr{L}_{\mathrm{N}} = -\mathbb{E}_X \left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k\left(x_j, c_t\right)} \right] \approx -\frac{1}{N} \sum_{i=1}^{N} \log \frac{\exp\left(\mathrm{sim}\left(x_i, c_i\right)/\tau\right)}{\sum_{j=1}^{N} \exp\left(\mathrm{sim}\left(x_i, c_j\right)/\tau\right)}$$

Applications

- CLIP enables searching for images by interpreting natural language descriptions.
- It classifies images into categories defined at inference using textual prompts (e.g., "This is a photo of a [label]").
- Visual Question Answering (VQA) becomes possible with CLIP's text-image alignment.
- Integrated into DALL·E or Stable Diffusion, it improves image generation with scoring and feedback loops.
- CLIP enhances robot-human interaction by understanding instructions about objects or scenes.

Applications

- It matches user queries with product images for seamless search and discovery in e-commerce.
- The model enables zero-shot classification to identify new or rare medical conditions.
- CLIP detects anomalies in manufacturing processes or equipment visuals to ensure quality.
- It personalizes digital experiences based on user preferences described in natural language.

Contributions

These slides are authored by:

- Hooman Zolfaghari
- Amir Mohammad Fakhimi