Self Supervised Learning - Contrastive Loss

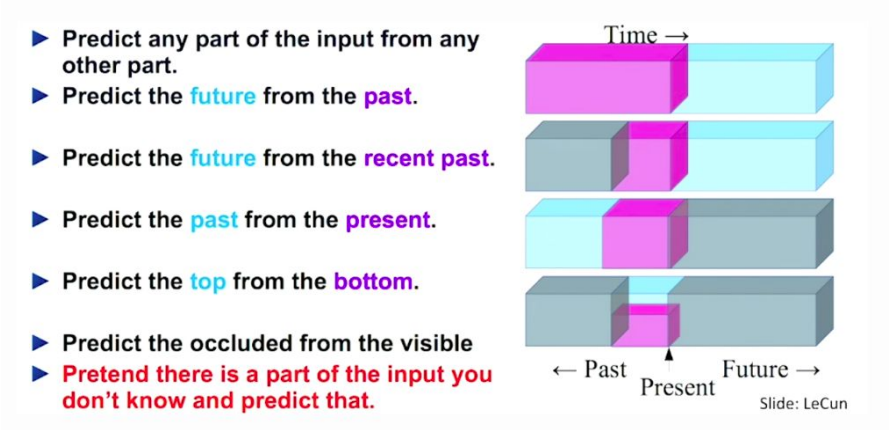
Machine Learning

- Three types of learning
  - **Supervised**: You have the data and you have the annotations/labels, you want to perform classification/regression
  - **Unsupervised**: You have the data but no label, and you want to figure out some sort of latent representation from the data e.g. clustering, dimensionality reduction
  - **Self supervised**: You have the data but no annotations/labels, and you want to perform classification/regression
- Self supervised learning (SSL) tries to take the best from both the worlds of supervised and unsupervised
  - There are often so much data but without labels
  - Labelling requires manual labor, is time consuming and is prone to errors
    - e.g. ImageNet labelling with 14M images took roughly 22 human years [1]

[1] “Self supervised learning - Pretext tasks” - https://atcold.github.io/pytorch-Deep-Learning/en/week10/10-1/

Self Supervised Learning (SSL)

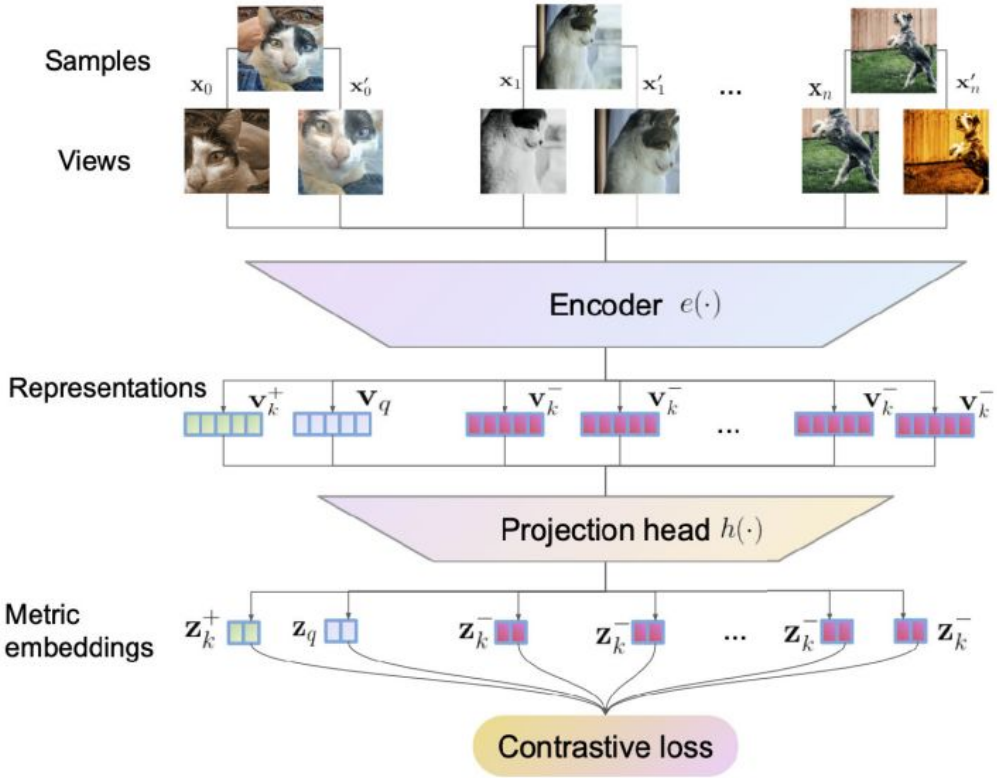
- Self supervised learning is done in two steps:
  - Pretext task: Learn representation from unlabelled data
  - Downstream task: The actual classification/regression task
- The way the learning happens in pretext task can be divided into two broad categories
  - **Contrastive**: Learn by comparing positive and negative samples of the data
  - **Non-contrastive**: Learn only from positive samples of the data
- But how to know positive and negative samples of the data when there are no labels?
  - Seems counterintuitive
  - There is actually a recent work that combines contrastive learning with labels [2]



A typical way of constructing an self-supervised learning problem

[2] "Supervised contrastive learning", NEURIPS (2020)

Contrastive SSL



Query	An input data point
Positive key	An augmented/transformed version of the input data point
Negative key	All other data in the dataset except the query and positive key
Encoder	Will learn an embedding space of latent representation
Projection Head	Use some form of projection, contextualization or quantization on the embedding space to learn a better representation
Contrastive loss	Minimizes the “distance” between query and positive key, while maximizing the “distance” between query and positive key pair and negative key(s)

Contrastive SSL (continued)

What does the contrastive loss capture?

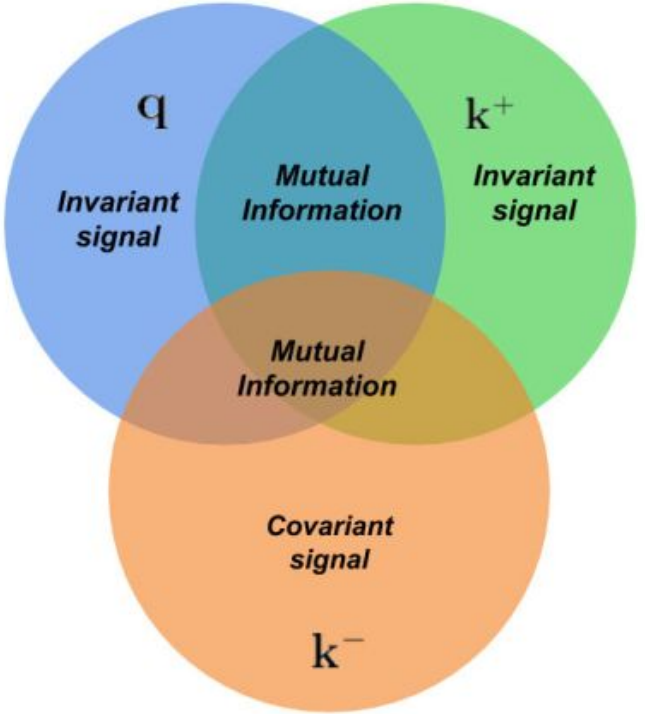
Invariant signal: The signal that is not mutual between query and positive keys.

These representations should be made as close as possible.

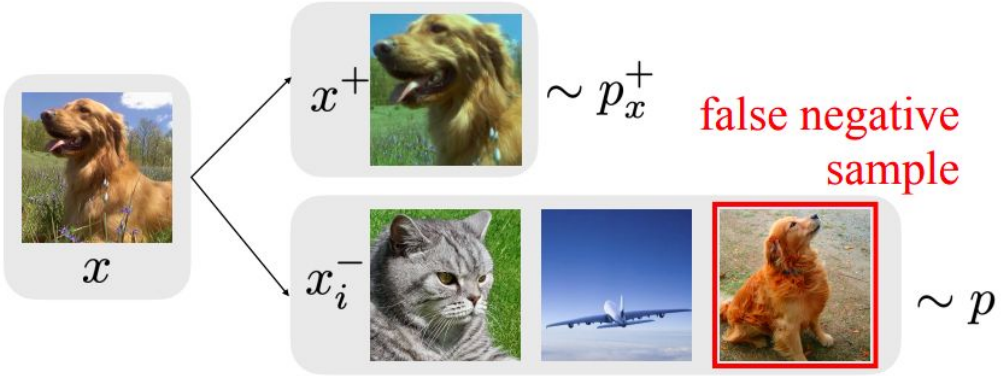
Covariant signal: Signals that are not mutual between:

- Negative keys and query key, or
- Negative keys and positive key

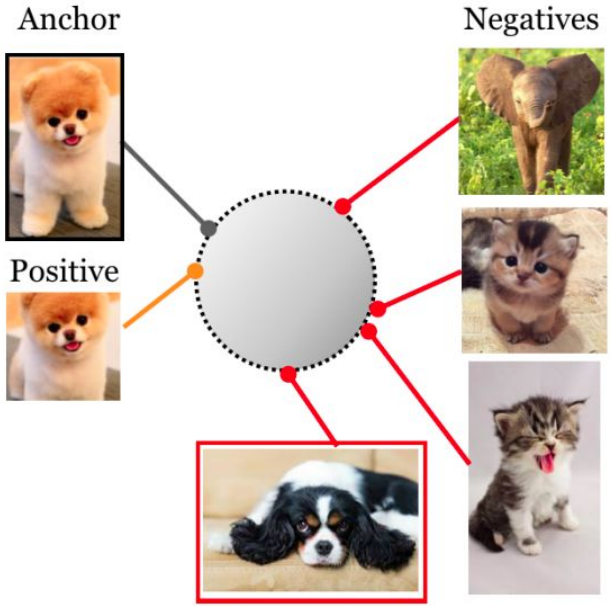
These representations should be made as different as possible so that the positive and/or query keys are easier to distinguish from negative keys



Contrastive SSL (continued)



(a)



(b)

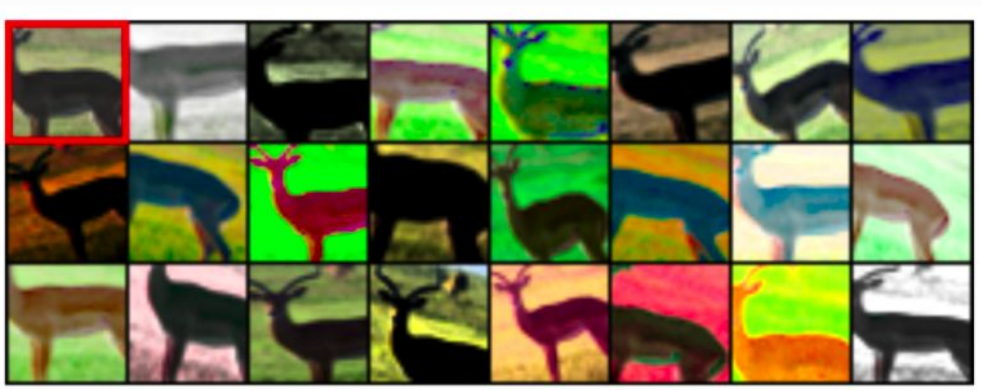
Without access to labels, dissimilar (negative) points are typically taken to be randomly sampled datapoints, implicitly accepting that these points may, in reality, actually have the same label. So, we need to be able to account for the sampling of same-label data points, even without knowledge of the true labels.

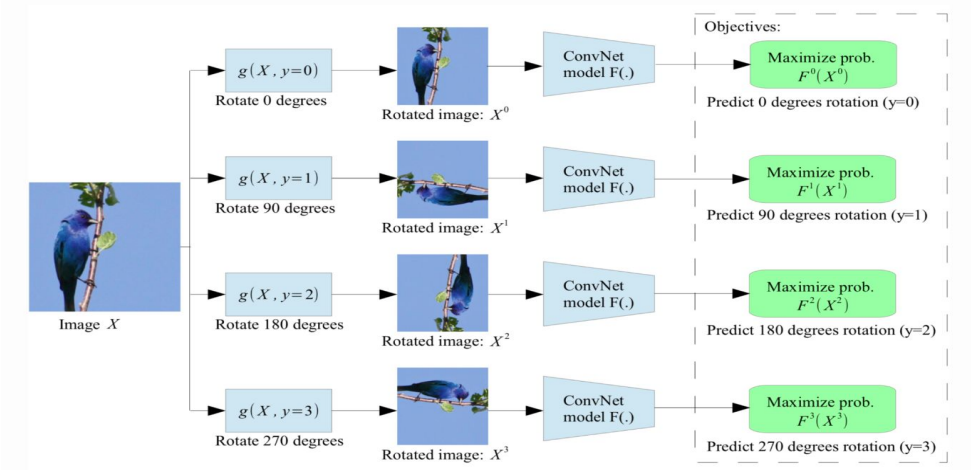
Contrastive SSL (continued)

Views: How the same data can be viewed differently (using transformations/augmentations)

Transformations:

- **Images:** Images can have a number of transformations:
  - Random crop
  - Chromatic Aberration
  - Random rotation
  - Brightness/contrast adjustments
  - Blurring





Examples for image transformations

Contrastive SSL (continued)

Views: How the same data can be viewed differently (using transformations/augmentations)

Transformations:

- **Audio data:** warping, frequency and temporal masking in the Mel spectrogram format
- **Text:** back-translation method [3]

The quick brown fox jumps over the lazy dog

→ দ্রুত বাদামী শিয়াল অলস কুকুরের উপর ঝাঁপিয়ে পড়ে

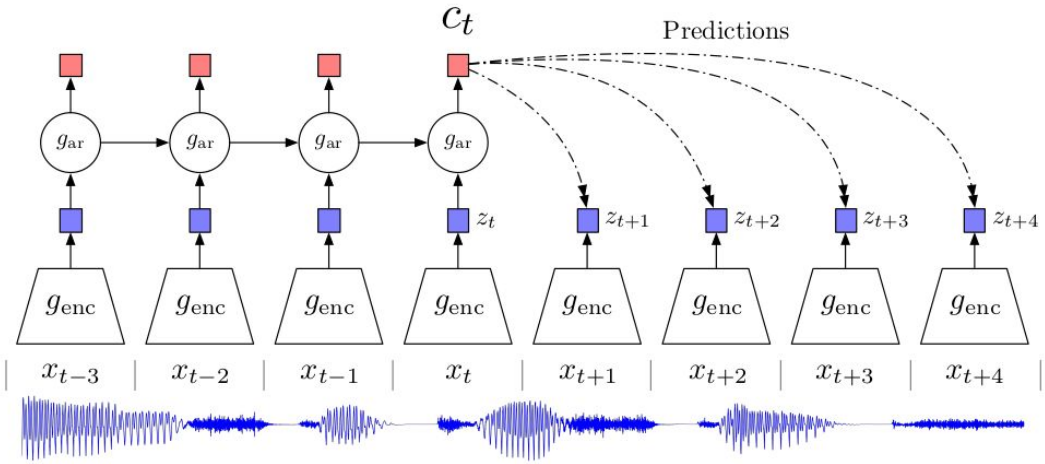
→ Quickly the brown fox jumps on the lazy dog

[3] "CERT: Contrastive self-supervised learning for language understanding", arXiv:2005.12766 (2020)

Contrastive SSL (continued)

Projection Heads: Enable the computation of similarity distance between representations by further reducing dimensions or introducing context/quantization.

- MLP (linear/non-linear)
- GRUs (for contextualization): Aggregate previous representations in timestep to provide context information[4]
- Quantization: Map multiple representations into a single one (code book[5], clustering)



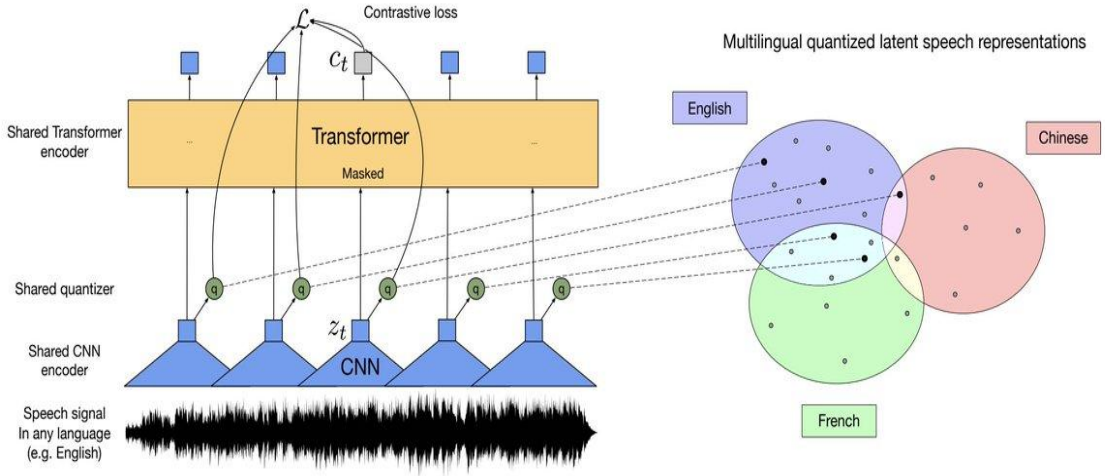
Example for contextualization

[4] "Representation learning with contrastive predictive coding", Google DeepMind (2018)
[5] "Unsupervised Cross-lingual Representation Learning for Speech Recognition", Facebook AI Research (2020)

Contrastive SSL (continued)

Projection Heads: Enable the computation of similarity distance between representations by further reducing dimensions or introducing context/quantization.

- MLP (linear/non-linear)
- GRUs (for contextualization): Aggregate previous representations in timestep to provide context information[4]
- Quantization: Map multiple representations into a single one (code book[5], clustering)



Example for quantization

[4] "Representation learning with contrastive predictive coding", Google DeepMind (2018)
[5] "Unsupervised Cross-lingual Representation Learning for Speech Recognition", Facebook AI Research (2020)

Contrastive SSL (continued)

Contrastive Loss: The key component

$$\mathcal{L}^{self}$$

$$= \sum_{i \in I} \mathcal{L}_i^{self}$$

$$= -\sum_{i \in I} \log \frac{\exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_{j(i)}/\tau\right)}{\sum_{a \in A(i)} \exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_a/\tau\right)}$$

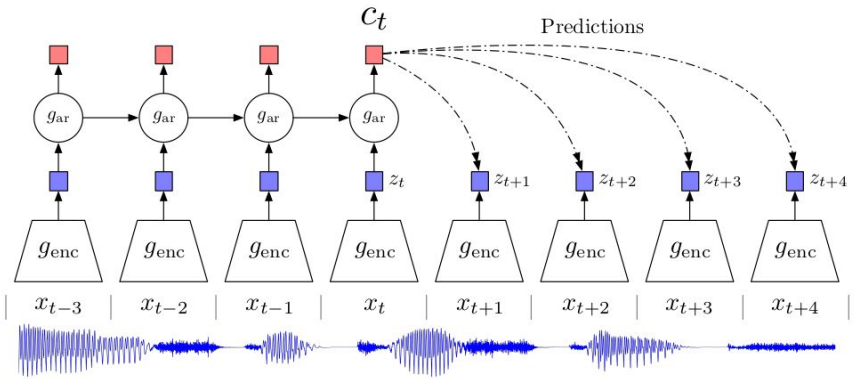
Here, $\boldsymbol{z}_\ell = Proj(Enc(\tilde{\boldsymbol{x}}_\ell)) \in \mathcal{R}^{D_P}$, the $\cdot$ symbol denotes the inner (dot) product, $\tau \in \mathcal{R}^+$ is a scalar temperature parameter, and $A(i) \equiv I \setminus \{i\}$. The index i is called the *anchor*, index $j(i)$ is called the *positive*, and the other $2(N-1)$ indices ($\{k \in A(i) \setminus \{j(i)\}$) are called the *negatives*.

Note that for each anchor $i \in I \equiv \{1...2N\}$, there is 1 positive pair and $2N-2$ negative pairs. The denominator has a total of $2N-1$ terms (the positive and negatives).

Contrastive SSL: Contrastive Predictive Coding [4]

In the case of sequential data, we need to consider the past data (at time step t_i) in order to learn the representation of the future data (at timestep t_{i+1})

- Mutual information is contained between present and future
- But, as we get further into the future from timestep t_i, the mutual information keeps decreasing i.e. there is some global context
- We need to be able to learn a representation that maximizes the mutual information as well as the global context
- Maximizing the mutual information is the key, which can be modeled as a probability distribution



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

[4] "Representation learning with contrastive predictive coding", Google DeepMind (2018)

Contrastive SSL: Contrastive Predictive Coding (continued)

- Do not predict the future directly with a generative model, instead formulate a density ratio between future and present

$$f_k(x_{t+k}, c_t) \propto \frac{p(x_{t+k}|c_t)}{p(x_{t+k})}$$

$$f_k(x_{t+k}, c_t) = \exp\left(z_{t+k}^T W_k c_t\right)$$

- Where, $W_k^T c_t$ is a linear transformation is used for the prediction with a different W□ (learnable) for every step k.
  - But non-linear transformations could have been used too (i.e. non-linear networks or RNNs)
- Either of z_t or c_t can be used for downstream tasks
- Encoder and Autoregressive model agnostic
  - Masked CNNs or Self-attention networks could improve results further

Contrastive SSL: Contrastive Predictive Coding (continued)

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.

Given a set *X* with N random samples containing one positive sample from $p(x_{t+k}|c_t)$ and N-1 negative samples from the proposal distribution $p(x_{t+k})$, the InfoNCE loss can be defined as

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

Essentially, optimizing the negative log probability of classifying the positive sample correctly is the goal

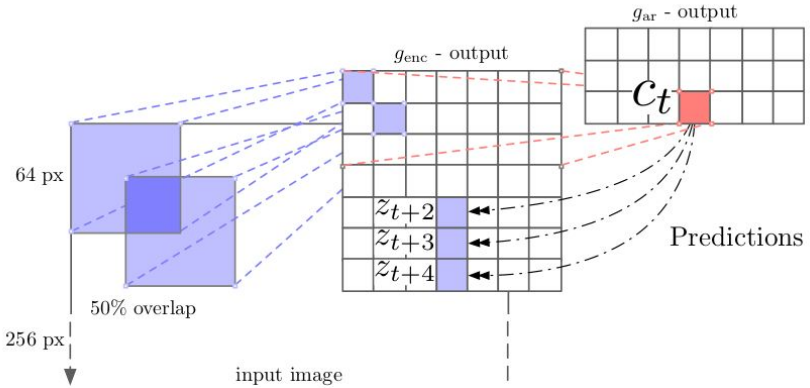
Optimizing this loss will lead to estimating the density ratio $f_k(x_{t+k}, c_t) \propto \frac{p(x_{t+k}|c_t)}{p(x_{t+k})}$

- Minimize the loss ⇒ Maximize the numerator ⇒ Maximize $p(x_{t+k}|c_t)$ ⇒ Maximize the probability that we are able to correctly identify the positive sample from set X ⇒ Mutual information is maximized

Contrastive SSL: Contrastive Predictive Coding (continued)

Training: The encoder and projection head changes with respect to the type of data being dealt with

- Images
  - ILSVRC ImageNet competition dataset
  - 300x300 images randomly cropped to 256x256
  - 7x7 grids each of size 64x64 extracted from 256x256 images with a 32px overlap
  - The 64x64 crops are all grayscale and are further randomly cropped to 60x60 and padded back to 64x64
  - Each 64x64 crop encoded by ResNet encoder, resulting in a 7x7x1024 tensor
  - PixelCNN style autoregressive model used to predict latent activations in each row top-to-bottom
  - Predict upto five rows, apply contrastive loss for each patch in the row
  - Use linear classifier on top of CPC features



Contrastive SSL: Contrastive Predictive Coding (continued)

Training: The encoder and projection head changes with respect to the type of data being dealt with

- Text
  - BookCorpus dataset
  - Learn the unsupervised model based on the dataset
  - Linear mapping between word2vec and word embeddings learnt by the model
  - A simple sentence encoder encodes an input sentence to 2400 dimension
  - Followed by a GRU with 2400 hidden units which predicts upto 3 future sentences

Contrastive SSL: Contrastive Predictive Coding (continued)

Training: The encoder and projection head changes with respect to the type of data being dealt with

- Audio
  - 100-hour subset of the publicly available LibriSpeech dataset
  - No labels available other than raw text
  - Obtain force aligned phone sequences using Kaldi toolkit and pre-trained models on Librespeech
  - five convolutional layers with strides [5, 4, 2, 2, 2], filter-sizes [10, 8, 4, 4, 4] and 512 hidden units with ReLU activations
  - Downsampling factor of the network is 160 = a feature vector for every 10 ms of speech
  - GRU with 256 hidden units used as the autoregressive part. The output of the GRU at each timestep is used as the context c.
  - 12 timesteps into the future predicted with contrastive loss

Contrastive SSL (continued)

There are variations to the contrastive loss that try to address the ***sampling bias*** problem

- When sampling negatives, it is possible that we will sample an instance that is of the same class of data as the anchor/query, i.e. false negatives
- We don't have labels
- How to account for the sampling of false negatives?

The following variations of the contrastive loss try to address the aforementioned issue:

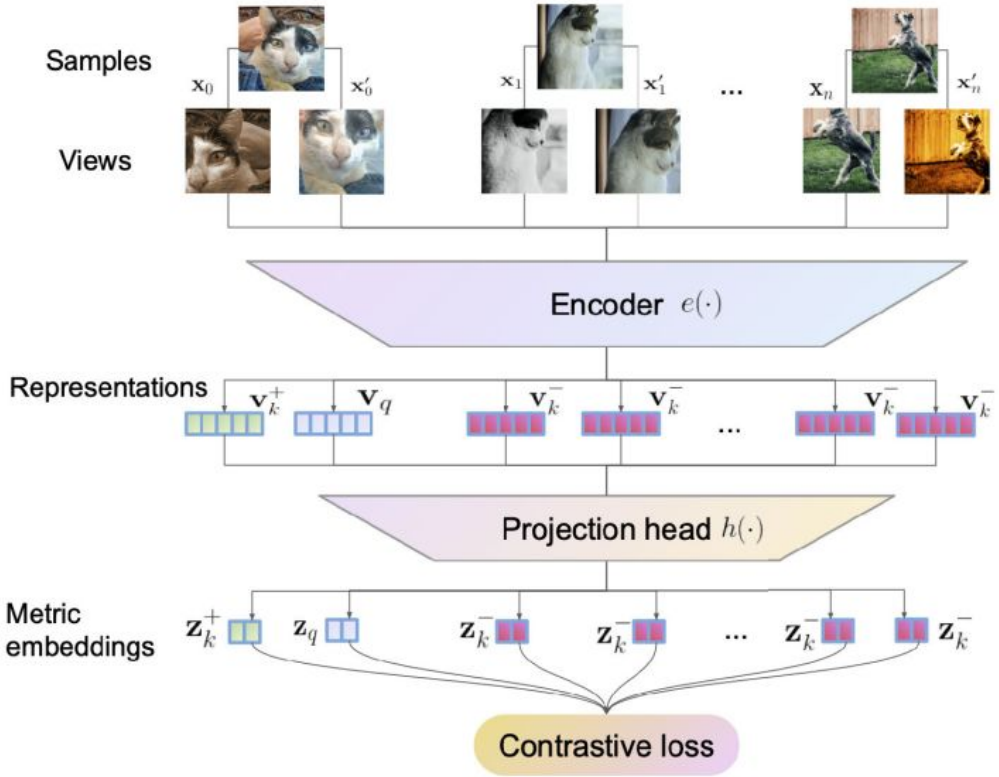
- Supervised Contrastive Learning[2]
  - Contrastive learning, but with the help of labels
- Debiased Contrastive Learning[7]
  - Get an approximation of the number of positive samples in the data
  - Implicitly try to figure out a distribution of the negative data

[2] "Supervised contrastive learning", Google Research (2020)
[7] "Debiased Contrastive Learning", NEURIPS (2020)

Contrastive SSL (continued)

Encoder: Learn an embedding space that can map the input views to a representation vector

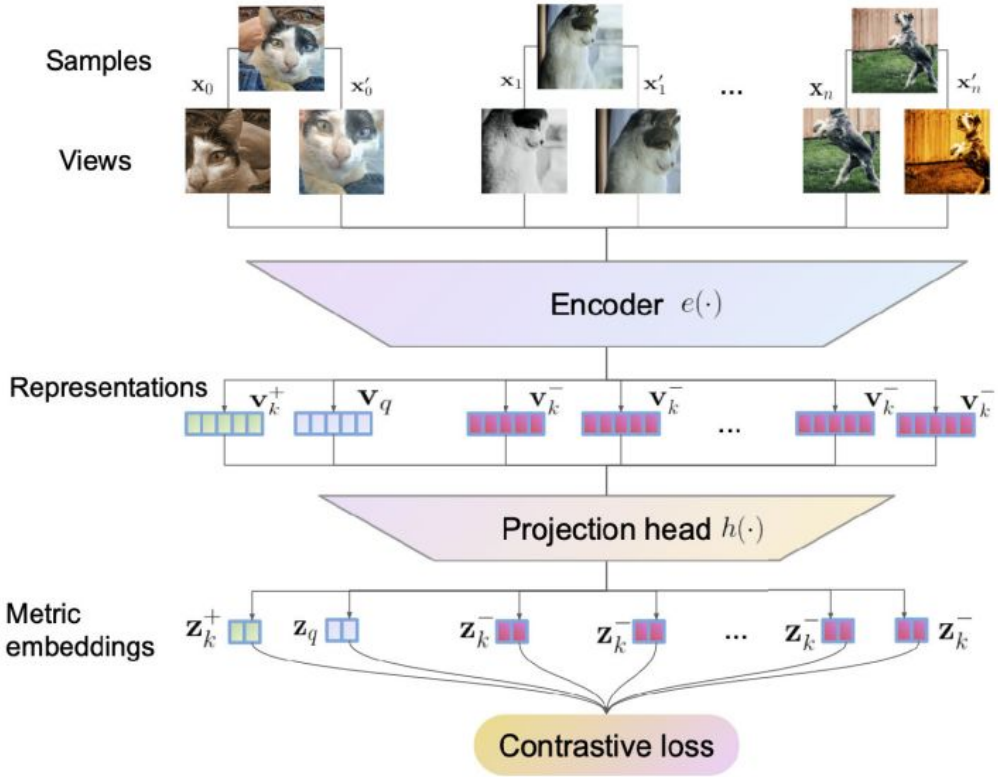
- End-to-end encoders: They are updated directly using gradients back-propagated with respect to the contrastive loss function
  - Expensive on GPU VRAM since the representation and hidden activations need to be stored, limiting the batch size for calculating contrastive loss



Contrastive SSL (continued)

Encoder: Learn an embedding space that can map the input views to a representation vector

- Online-offline encoders
  - Use an offline encoder in addition to an online encoder
  - The online encoder learns the representations of the keys
  - The offline encoder is updated by the online encoder[6], decoupling batch size

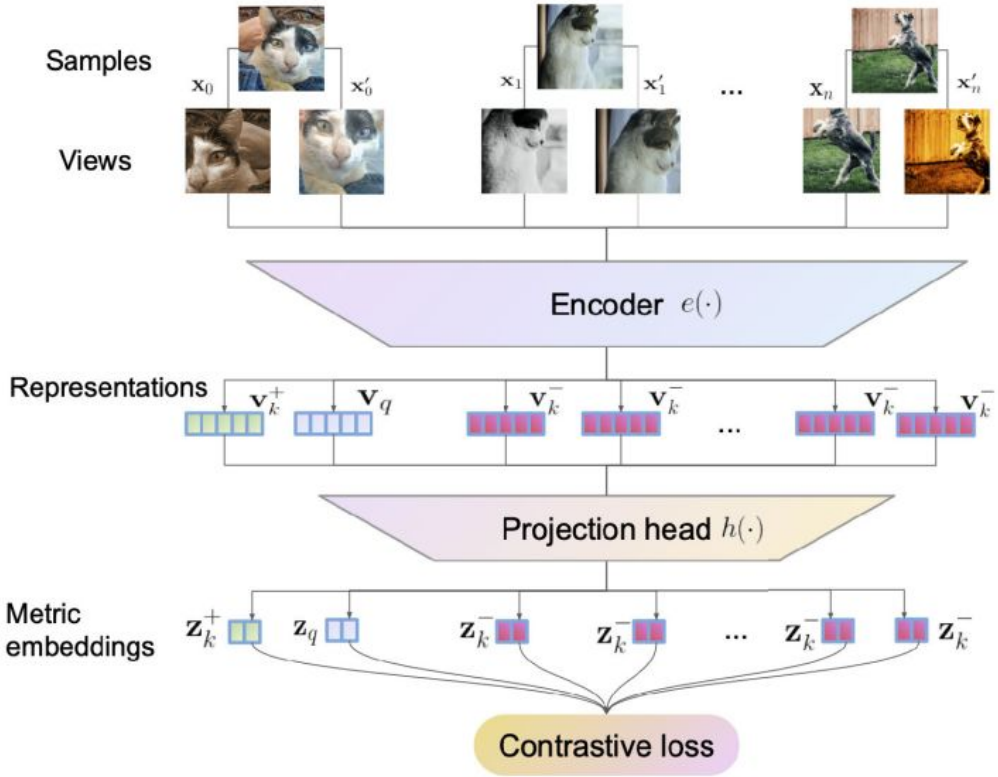


[6] "Momentum Contrast for Unsupervised Visual Representation Learning", CVF (2019)

Contrastive SSL (continued)

Encoder: Learn an embedding space that can map the input views to a representation vector

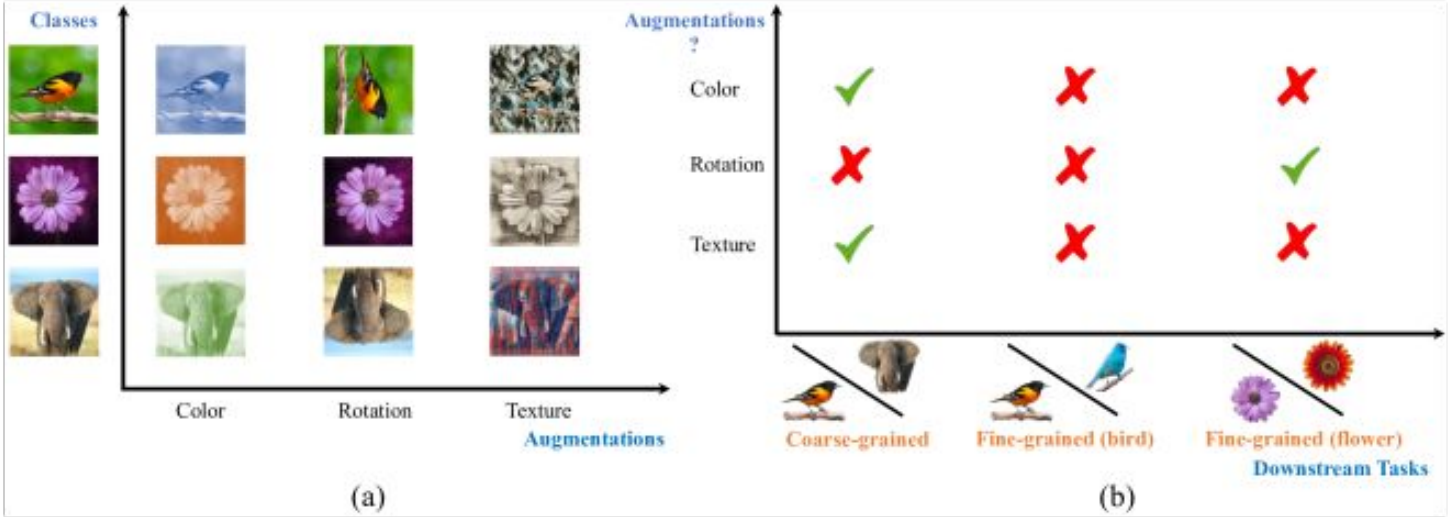
- Pre-trained encoders
  - Useful in a teacher-student network context
  - A smaller student encoder network tries to match the representation of the parent encoder network
  - Parent encoder encodes the keys
  - Student encoder encodes the queries
  - Parent encoder has frozen weights



Contrastive SSL (continued)

Caveats

- The data augmentation part is tricky
  - You need to be aware of the dataset as well as the context of the task



Contrastive SSL (continued)

Caveats

- How much negative data is needed?
  - More negative data usually means that the representation will not collapse into a single cluster
  - Computationally expensive
- Quality of the negative data?
  - More careful selection of negative samples has been shown to improve the convergence rate and performance of the learned embeddings on downstream tasks
  - Consistent with hard negative and positive mining techniques
- Trade-off between quality and quantity
- There are works that do not use negative samples at all (non-contrastive), e.g. Bootstrap Your Own Latent (BYOL), Simple Siamese (SimSiam)

Additional Resources

1. Contrastive Representation Learning: A Framework and Review, IEEE Access (2020)
2. “Self Supervised Representation Learning” blog post by L. Weng (2019) - link
3. “Contrastive Representation Learning” blog post by L. Weng (2021) - link