Representation Constrastive Learning

Learning by comparison

Pablo Miralles González

October 23, 2023

2023-10-23

Representation Constrastive Learning

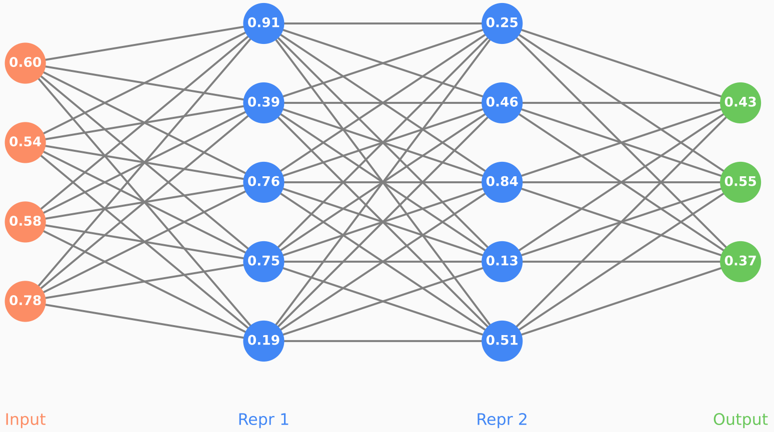
Representation Constrastive Learning
Learning by comparison
Pablo Miralles González
October 23, 2023

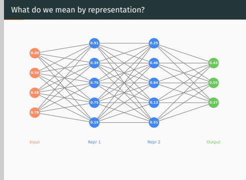
Representation learning
Contrastive learning
Loss functions for contrastive learning
Generating data
Discussion on negative examples
Applications
Conclusions

Representation learning

Representation learning

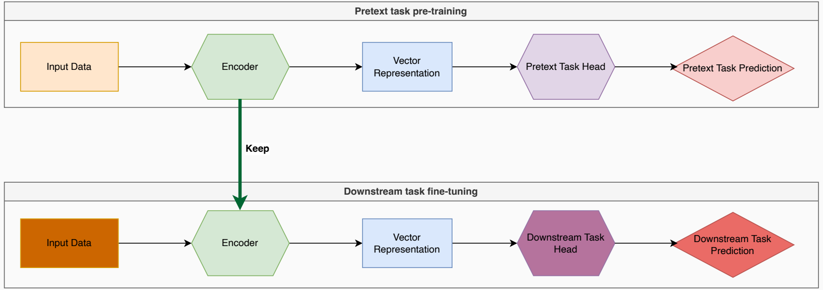
What do we mean by representation?

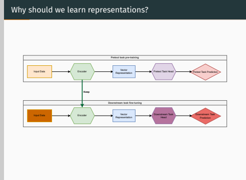




- Internal layers outputs can be viewed as different views or representations of the input data.
- They contain meaningful features for the task.

Why should we learn representations?





- The possibility of transfer learning.
- Train for a complex pretext task to obtain a very general representation of the input data.
- Keep encoder, change head and fine-tune and use for downstream tasks.

Why should we learn representations?

- Overcome data bottlenecks

Why should we learn representations?

- Overcome data bottlenecks

1. We might not have enough data for the downstream task. We select pretext tasks for which we can generate data easily.

Why should we learn representations?

- Overcome data bottlenecks
- Outsource compute resources for training

Why should we learn representations?
- Overcome data bottlenecks
- Outsource compute resources for training

1. People with greater resources can pre-train and upload weights. We can download and fine-tune for any task we want. If the encoder weights are frozen, it is much cheaper: we only update a smaller head. This allows people with fewer resources to use bigger models than they could otherwise.

Why should we learn representations?

- Overcome data bottlenecks
- Outsource compute resources for training
- Better generalization

Why should we learn representations?

- Overcome data bottlenecks
- Outsource compute resources for training
- Better generalization

1. If the data for the downstream task is not representative, we might learn spurious correlations. By pre-training for a complex task with rich data, we make sure the model understand the latent distribution correctly. Still, fine-tuning might lead to representational collapse.

Why should we learn representations?

- Overcome data bottlenecks
- Outsource compute resources for training
- Better generalization
- Zero-shot capabilities

Why should we learn representations?

- Overcome data bottlenecks
- Outsource compute resources for training
- Better generalization
- Zero-shot capabilities

1. For example, ChatGPT (see e.g. text classification, text transformations, code generation, code analysis...) or CLIP (zero-shot image classification with arbitrary classes).

Why should we not learn representations?

- Sometimes overkill

Why should we not learn representations?
- Sometimes overkill

1. If downstream task is very simple and the data is decent, we just don't need to.

Why should we not learn representations?

- Sometimes overkill
- Massive compute requirements

Why should we not learn representations?

- Sometimes overkill
- Massive compute requirements

1. It is very expensive and not everyone can pre-train massive deep learning models.

Why should we not learn representations?

- Sometimes overkill
- Massive compute requirements
- Generally poor zero-shot performance

Why should we not learn representations?

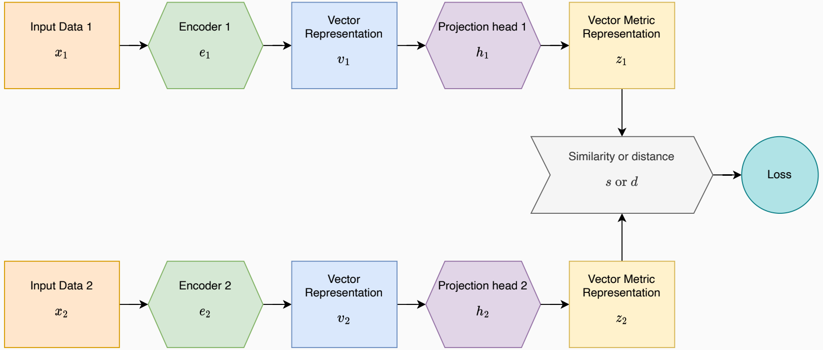
- Sometimes overkill
- Massive compute requirements
- Generally poor zero-shot performance

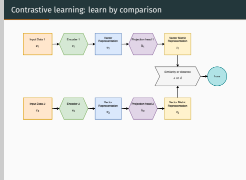
1. Without fine-tuning, models that can perform zero-shot predictions are unlikely to perform very well.

Contrastive learning

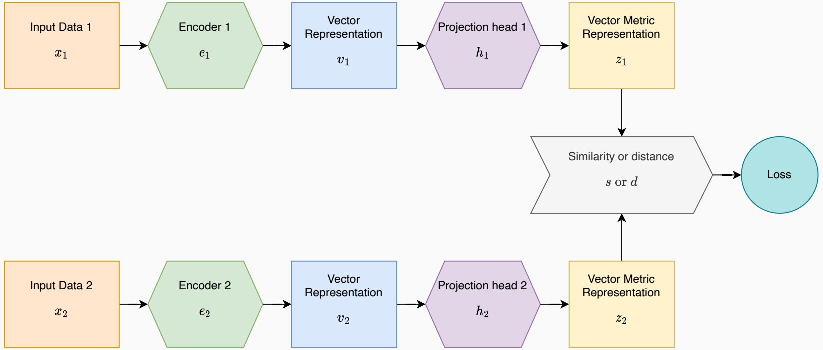
Contrastive learning

Contrastive learning: learn by comparison



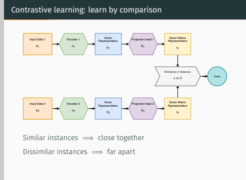


Contrastive learning: learn by comparison



Similar instances $\implies$ close together

Dissimilar instances $\implies$ far apart



- Go over full diagram.
- Input data might be of different modalities.
- For data of the same modality, we can use the same encoder and head with tied weights.

Examples of distance and similarity

- Euclidean distance $d(z_1, z_2) = \|z_1 - z_2\|$
- Cosine similarity $s(z_1, z_2) = \frac{\langle z_1, z_2 \rangle}{\|z_1\| \cdot \|z_2\|}$

Examples of distance and similarity
- Euclidean distance $d(z_1, z_2) = \|z_1 - z_2\|$
- Cosine similarity $s(z_1, z_2) = \frac{\langle z_1, z_2 \rangle}{\|z_1\| \cdot \|z_2\|}$

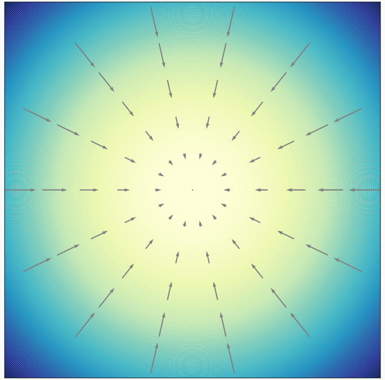
Loss functions for contrastive learning

Loss functions for contrastive learning

Pair loss[1]

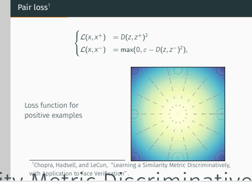
$$\begin{cases} \mathcal{L}(x, x^+) &= D(z, z^+)^2 \\ \mathcal{L}(x, x^-) &= \max(0, \varepsilon - D(z, z^-)^2), \end{cases}$$

Loss function for positive examples



[1] Chopra, Hadsell, and LeCun, "Learning a Similarity Metric Discriminatively, with Application to Face Verification".

[a] Chopra, Hadsell, and LeCun, "Learning a Similarity Metric Discriminative

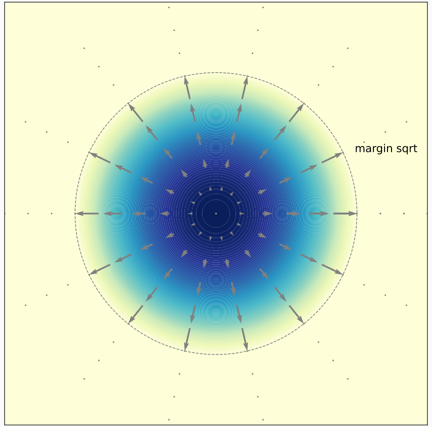


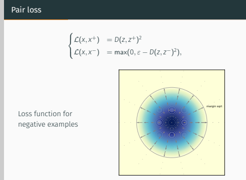
- Different for positive and negative examples.
- Explain plot.

Pair loss

$$\begin{cases} \mathcal{L}(x, x^+) & = D(z, z^+)^2 \\ \mathcal{L}(x, x^-) & = \max(0, \varepsilon - D(z, z^-)^2), \end{cases}$$

Loss function for negative examples

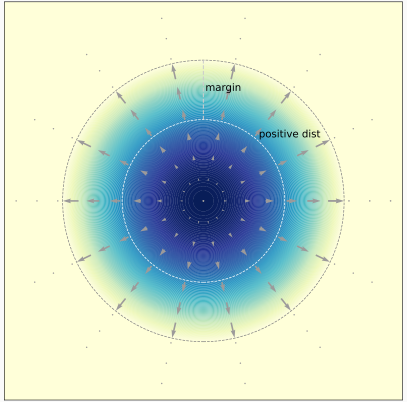




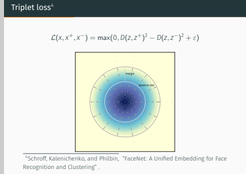
- Dissimilar instances separated by margin ε.

Triplet loss[4]

$$\mathcal{L}(x, x^{+}, x^{-}) = \max(0, D(z, z^{+})^2 - D(z, z^{-})^2 + \varepsilon)$$



[4] Schroff, Kalenichenko, and Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering" .



- Distance between similar and dissimilar instances separated by margin.

Lifted Structured Loss[7]

$$\{x_i\}_{i=1}^{n} = \text{set of examples}$$

$$P = \{\text{pairs of similar instances}\} \quad N = \{\text{pairs of dissimilar instances}\}$$

[7] Song et al., "Deep Metric Learning via Lifted Structured Feature Embedding" .

Lifted Structured Loss[7]

$\{x_i\}_{i=1}^{n}$ = set of examples
P = {pairs of similar instances} N = {pairs of dissimilar instances}

[7] Song et al., "Deep Metric Learning via Lifted Structured Feature Embedding" .

Lifted Structured Loss

$$\{x_i\}_{i=1}^{n} = \text{set of examples}$$

$$P = \{\text{pairs of similar instances}\} \quad N = \{\text{pairs of dissimilar instances}\}$$

$$\mathcal{L}(N, P) = \frac{1}{2|P|} \sum_{(i,j)\in P} L_{i,j}^2$$

$$L_{i,j} = D_{i,j} + \log\left(\sum_{(i,k)\in N} e^{\varepsilon - D_{i,k}} + \sum_{(j,l)\in N} e^{\varepsilon - D_{j,l}} \right)$$

$$D_{i,j} = D(z_i, z_j)$$

Lifted Structured Loss
$\{x_i\}_{i=1}^{n} = \text{set of examples}$
$P = \{\text{pairs of similar instances}\} \quad N = \{\text{pairs of dissimilar instances}\}$
$\mathcal{L}(N, P) = \frac{1}{2|P|} \sum_{(i,j)\in P} L_{i,j}^2$
$L_{i,j} = D_{i,j} + \log\left(\sum_{(i,k)\in N} e^{\varepsilon - D_{i,k}} + \sum_{(j,l)\in N} e^{\varepsilon - D_{j,l}} \right)$
$D_{i,j} = D(z_i, z_j)$

Lifted Structured Loss

$$\{x_i\}_{i=1}^n = \text{set of examples}$$

$$P = \{\text{pairs of similar instances}\} \quad N = \{\text{pairs of dissimilar instances}\}$$

$$\mathcal{L}(N, P) = \frac{1}{2|P|} \sum_{(i,j)\in P} L_{i,j}^2$$

$$\underbrace{L_{i,j}}_{\text{smooth}} \geq \hat{L}_{i,j} = D_{i,j} + \max\left(\max_{(i,k)\in N} \varepsilon - D_{i,k}, \max_{(j,l)\in N} \varepsilon - D_{j,l} \right)$$

$$D_{i,j} = D(z_i, z_j)$$

Lifted Structured Loss

$\{x_i\}_{i=1}^n$ = set of examples

$P = \{\text{pairs of similar instances}\} \quad N = \{\text{pairs of dissimilar instances}\}$

$\mathcal{L}(N, P) = \frac{1}{2|P|} \sum_{(i,j)\in P} L_{i,j}^2$

$\underbrace{L_{i,j}}_{\text{smooth}} \geq \hat{L}_{i,j} = D_{i,j} + \max\left(\max_{(i,k)\in N} \varepsilon - D_{i,k}, \max_{(j,l)\in N} \varepsilon - D_{j,l} \right)$

$D_{i,j} = D(z_i, z_j)$

- We are actually penalizing the small differences between the distance with a positive example and the hardest negative example, up to some margin, similar to the triplet loss.

Binary Noise-Contrastive Estimation[7]

$$X(x_1, x_2) = \begin{cases} 1 & \text{if } x_1 \text{ and } x_2 \text{ are similar} \\ 0 & \text{if } x_1 \text{ and } x_2 \text{ are dissimilar} \end{cases}$$

$$X(x_1, x_2) \sim P(\cdot|x_1, x_2) \quad \rightarrow \quad P(1|x_1, x_2) = \sigma(s(z_1, z_2))$$

[7] Gutmann and Hyvärinen, "Noise-Contrastive Estimation: A New Estimation Principle for Unnormalized Statistical Models".

Binary Noise-Contrastive Estimation[7]

$$X(x_1, x_2) = \begin{cases} 1 & \text{if } x_1 \text{ and } x_2 \text{ are similar} \\ 0 & \text{if } x_1 \text{ and } x_2 \text{ are dissimilar} \end{cases}$$

$$X(x_1, x_2) \sim P(\cdot|x_1, x_2) \quad \rightarrow \quad P(1|x_1, x_2) = \sigma(s(z_1, z_2))$$

[7] Gutmann and Hyvärinen, "Noise-Contrastive Estimation: A New Estimation Principle for Unnormalized Statistical Models".

Binary Noise-Contrastive Estimation

$$X(x_1, x_2) = \begin{cases} 1 & \text{if } x_1 \text{ and } x_2 \text{ are similar} \\ 0 & \text{if } x_1 \text{ and } x_2 \text{ are dissimilar} \end{cases}$$

$$X(x_1, x_2) \sim P(\cdot | x_1, x_2) \quad \rightarrow \quad P(1 | x_1, x_2) = \sigma(s(z_1, z_2))$$

$$\mathcal{L}_{Bin-NCE} = -\mathbb{E}_{p^+} \log P(1|x_1, x_2) - \mathbb{E}_{p^-} \log(1 - P(1|x_1, x_2)) \approx$$

$$-\frac{1}{|P|} \sum_{(i,j) \in P} \log \sigma(s(z_i, z_j)) - \frac{1}{|N|} \sum_{(i,j) \in N} \log\left(1 - \sigma(s(z_i, z_j))\right)$$

Binary Noise-Contrastive Estimation

$$X(x_1, x_2) = \begin{cases} 1 & \text{if } x_1 \text{ and } x_2 \text{ are similar} \\ 0 & \text{if } x_1 \text{ and } x_2 \text{ are dissimilar} \end{cases}$$

$$X(x_1, x_2) \sim P(\cdot | x_1, x_2) \quad \rightarrow \quad P(1 | x_1, x_2) = \sigma(s(z_1, z_2))$$

$$\mathcal{L}_{Bin-NCE} = -\mathbb{E}_{p^+} \log P(1|x_1, x_2) - \mathbb{E}_{p^-} \log(1 - P(1|x_1, x_2)) \approx$$

$$-\frac{1}{|P|} \sum_{(i,j) \in P} \log \sigma(s(z_i, z_j)) - \frac{1}{|N|} \sum_{(i,j) \in N} \log\left(1 - \sigma(s(z_i, z_j))\right)$$

- Explain probabilistic approach and population sample with batch.

InfoNCE[8]

$$x; \quad S = \{x_0^+, x_1^-, \dots, x_n^-\} \quad \rightarrow \quad \text{rank the positive one!}$$

$$P(i|x,S) = \frac{\exp\big(s(x,x_i)\big)}{\sum_{j=0}^{n} \exp\big(s(x,x_j)\big)}$$

$$\mathcal{L}_{InfoNCE} = -\mathbb{E}\log\frac{\exp\big(s(x,x_0^+)\big)}{\sum_{j=0}^{n} \exp\big(s(x,x_j)\big)}$$

[8] Oord, Li, and Vinyals, *Representation Learning with Contrastive Predictive Coding* .

[a] Oord, Li, and Vinyals, *Representation Learning with Contrastive Predicti*

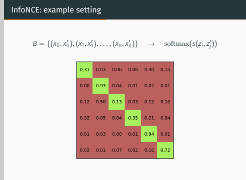
InfoNCE[8]
$x; \quad S = \{x_0^+, x_1^-, \dots, x_n^-\} \quad \rightarrow$ rank the positive one!
$$P(i|x,S) = \frac{\exp(s(x,x_i))}{\sum_{j=0}^{n} \exp(s(x,x_j))}$$
$$\mathcal{L}_{InfoNCE} = -\mathbb{E}\log\frac{\exp(s(x,x_0^+))}{\sum_{j=0}^{n} \exp(s(x,x_j))}$$
[8] Oord, Li, and Vinyals, *Representation Learning with Contrastive Predictive Coding* .

- Instance x, examples S, only x_0^+ positive.

InfoNCE: example setting

$$B = \{(x_0, x_0'), (x_1, x_1'), \ldots, (x_n, x_n')\} \quad \rightarrow \quad \mathrm{softmax}(s(z_i, z_j'))$$

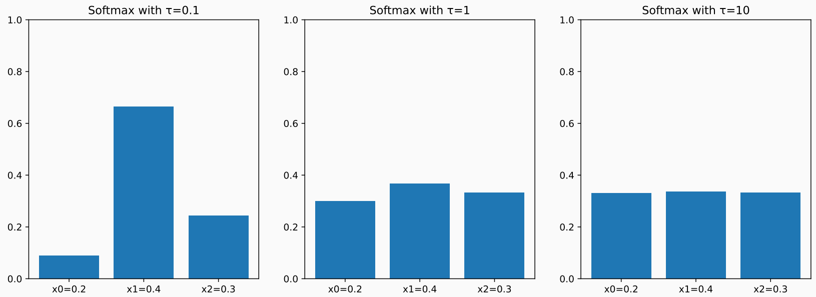
0.31	0.03	0.06	0.06	0.40	0.15
0.00	0.93	0.04	0.01	0.02	0.01
0.12	0.50	0.13	0.03	0.12	0.10
0.32	0.05	0.04	0.35	0.21	0.04
0.01	0.03	0.00	0.01	0.94	0.01
0.02	0.01	0.07	0.02	0.16	0.72



- Batches of pairs of similar instances $\{(x_1, x_1'), \ldots, (x_n, x_n')\}$.
- Instances across pairs are considered to be dissimilar.
- We can compute a similarity matrix $S = (s(z_i, z_j'))_{i,j}$, where the main diagonal values should be high and the rest should be low.
- We can calculate the InfoNCE across rows or columns. It is also possible to average both options, yielding a *symmetric InfoNCE* loss.

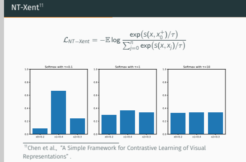
NT-Xent[11]

$$\mathcal{L}_{NT-Xent} = -\mathbb{E}\log \frac{\exp\left(s(x, x_0^+)/\tau\right)}{\sum_{j=0}^{n} \exp\left(s(x, x_j)/\tau\right)}$$



[11] Chen et al., "A Simple Framework for Contrastive Learning of Visual Representations" .

[a] Chen et al., "A Simple Framework for Contrastive Learning of Visual



- A small value of τ makes the softmax sharper, and small differences between the similarity of positive and negative examples already produces a high likelihood.
- A large value of τ forces the difference in similarity to be large.
- This parameter can be viewed as the margin parameter in previous functions.

Generating data

Generating data

Generating data for contrastive learning

Data = pairs of positive and negative examples.

Generating data for contrastive learning

Data = pairs of positive and negative examples.

Generating data for contrastive learning

Data = pairs of positive and negative examples.

- Human supervision.
- Data augmentation.
- Multi-sensor input.
- Local-global relationship.
- Sequential coherence/consistency.

Generating data for contrastive learning

Data = pairs of positive and negative examples.

- Human supervision.
- Data augmentation.
- Multi-sensor input.
- Local-global relationship.
- Sequential coherence/consistency.

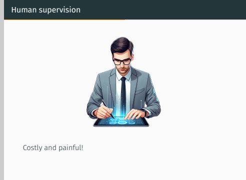
Human supervision

2023-10-23

Human supervision

Human supervision

Costly and painful!



Specially for massive NN models

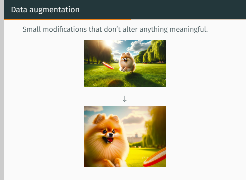
Data augmentation

Small modifications that don't alter anything meaningful.

↓

2023-10-23

Representation Constrastive Learning
└Generating data
└Data augmentation



- Augmented versions of similar instances are similar.
- Augmented versions of dissimilar instances are dissimilar.

Data augmentation

- **Images**. Rotations, translations, cutouts, cropping, resizing...
- **Text**. More complex. E.g. back-translation, masking words, adding noise...

Data augmentation
- Images. Rotations, translations, cutouts, cropping, resizing...
- Text. More complex. E.g. back-translation, masking words, adding noise...

Data augmentation

- **Images**. Rotations, translations, cutouts, cropping, resizing...
- **Text**. More complex. E.g. back-translation, masking words, adding noise...

Remark: performance can be highly sensitive to data augmentation strategies.[14]

[14] Chen et al., "A Simple Framework for Contrastive Learning of Visual Representations" .

Data augmentation

- **Images**. Rotations, translations, cutouts, cropping, resizing...
- **Text**. More complex. E.g. back-translation, masking words, adding noise...

Remark: performance can be highly sensitive to data augmentation strategies.[14]

[14] Chen et al., "A Simple Framework for Contrastive Learning of Visual Representations" .

Multi-sensor input

Video (audio and image), multiple cameras, cameras and other sensors…

2023-10-23

Multi-sensor input

Video (audio and image), multiple cameras, cameras and other sensors…

Local-global relationship

Local and global features should be similarly represented.

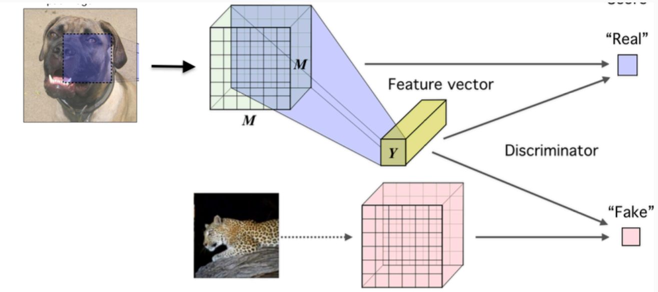
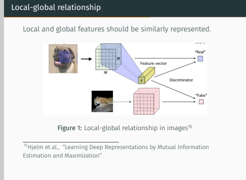


Figure 1: Local-global relationship in images[15]

[15] Hjelm et al., “Learning Deep Representations by Mutual Information Estimation and Maximization” .



The green box comes from an image of a dog, and the red box from an image of a leopard. The local features of the dog are aggregated into a global representation. The local and global features of the dog are similar, and the global features of the dog and the local features of the leopard are dissimilar.

Sequential coherence/consistency

Exploit continuity in smaller sub-units.

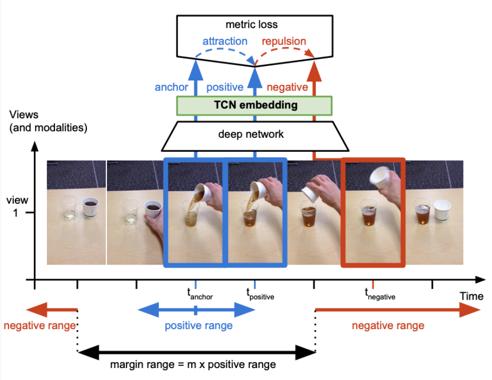
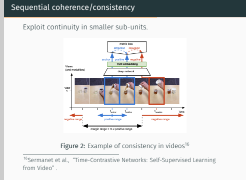


Figure 2: Example of consistency in videos[16]

[16] Sermanet et al., “Time-Contrastive Networks: Self-Supervised Learning from Video” .



For example, in videos, images that are very close in time are likely to be similar, to contain the same concept. Images far apart in time are likely to be different.

Discussion on negative examples

Discussion on negative examples

The importance of negative samples

- Negative examples prevent representational collapse $f \equiv$ constant.
- Empirical evidence of better performance.[17]

Several discussions around the use of negative examples.

[17] Chen et al., "A Simple Framework for Contrastive Learning of Visual Representations" .

The importance of negative samples
- Negative examples prevent representational collapse $f \equiv$ constant.
- Empirical evidence of better performance.[17]

Several discussions around the use of negative examples.

[17] Chen et al., "A Simple Framework for Contrastive Learning of Visual Representations" .

Negative examples?

False negatives in self-supervision
Negative examples?

False negatives in self-supervision

How to **remain self-supervised** and **mitigate bias from false negatives**?

False negatives in self-supervision

How to remain self-supervised and mitigate bias from false negatives?

How to **remain self-supervised** and **mitigate bias from false negatives**?

Work like Debiased Contrastive Learning.[18]

[18] Chuang et al., "Debiased Contrastive Learning" .

False negatives in self-supervision
How to remain self-supervised and mitigate bias from false negatives?
Work like Debiased Contrastive Learning.18
18Chuang et al., "Debiased Contrastive Learning" .

Improved performance by increasing the number of positive examples for a given instances, and adding a lot of complexity.

Hardware bottlenecks

Number of negatives $\approx$ Batch size $\implies$ (# negatives)^2 scaling

Hardware bottlenecks

Number of negatives $\approx$ Batch size $\implies$ (# negatives)^2 scaling

1. It is common to use other examples in the batch as negatives.
2. If we do this, we tie the batch size and the number of negatives.
3. Quadratic scaling of complexity with # of negatives.

Hardware bottlenecks

Number of negatives $\approx$ Batch size $\implies$ (# negatives)^2 scaling

Solution:

- Encoding of all samples stored offline.
- Negatives sampled from offline storage.
- $e_{\text{online}} \neq e_{\text{offline}}$

Hardware bottlenecks
Number of negatives $\approx$ Batch size $\implies$ (# negatives)^2 scaling
Solution:
- Encoding of all samples stored offline.
- Negatives sampled from offline storage.
- $e_{\text{online}} \neq e_{\text{offline}}$

1. It is common to use other examples in the batch as negatives.
2. If we do this, we tie the batch size and the number of negatives.
3. Quadratic scaling of complexity with # of negatives.

When and how to update offline encoder/encodings?

- All the samples after each checkpoint.[19]
- Queue of mini-batches and moving average.[20]

[19] Wu et al., "Unsupervised Feature Learning via Non-parametric Instance Discrimination" .
[20] He et al., "Momentum Contrast for Unsupervised Visual Representation Learning" .

Hardware bottlenecks
When and how to update offline encoder/encodings?
• All the samples after each checkpoint.19
• Queue of mini-batches and moving average.20
19Wu et al., "Unsupervised Feature Learning via Non-parametric Instance Discrimination" .
20He et al., "Momentum Contrast for Unsupervised Visual Representation Learning" .

- @wu2018UnsupervisedFeature sampled negative representations randomly from a memory bank with the full dataset. At the end of each epoch, all the representations in the memory bank are updated with the new checkpoint of the model.
- @he2020MomentumContrast use a queue with a fixed number of mini-batches. After each mini-batch, the new examples are added, and the oldest mini-batch is removed. The queue is used to sample negative examples for the current mini-batch. They separated the online encoder that is being trained from an offline encoder that produces the representations for the queue for empirical reasons. The parameters of the offline encoder are updated through a momentum update rule with the parameters of the online one.

Hard negative mining

$\Uparrow$ # negatives $\Longrightarrow$ harder negatives $\Longrightarrow$ performance?

[21]Kalantidis et al., “Hard Negative Mixing for Contrastive Learning” .

Hard negative mining
$\Uparrow$ # negatives $\Longrightarrow$ harder negatives $\Longrightarrow$ performance?
[21]Kalantidis et al., “Hard Negative Mixing for Contrastive Learning” .

Hard negative mining

$\Uparrow$ # negatives $\implies$ harder negatives $\implies$ performance?

Some work on the topic.[21]

[21] Kalantidis et al., “Hard Negative Mixing for Contrastive Learning” .

Hard negative mining

$\Uparrow$ # negatives $\implies$ harder negatives $\implies$ performance?

Some work on the topic.[21]

[21] Kalantidis et al., “Hard Negative Mixing for Contrastive Learning” .

Hard negative mining

$\Uparrow$ # negatives $\implies$ harder negatives $\implies$ performance?

Some work on the topic.[21]

Increased false negatives?

[21] Kalantidis et al., "Hard Negative Mixing for Contrastive Learning" .

Hard negative mining
$\Uparrow$ # negatives $\implies$ harder negatives $\implies$ performance?
Some work on the topic.[21]
Increased false negatives?
[21] Kalantidis et al., "Hard Negative Mixing for Contrastive Learning" .

- What if increasing negatives only improved performance by having a better chance of drawing hard negatives (negatives that are close to the instance)?
- Some work on mining hard negatives, that is, trying to select the hardest negatives to train.
- This increases the chance of false negatives when the encoder improves, in the self-supervised setting.

Are negative examples really necessary?

Do they **only** avoid collapse?

Are negative examples really necessary?
Do they **only** avoid collapse?

Are negative examples really necessary?

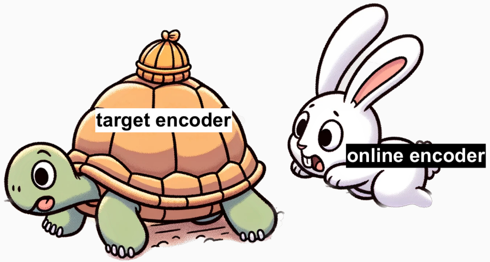
Do they **only** avoid collapse? Can this be done in a different way?

Are negative examples really necessary?
Do they **only** avoid collapse? Can this be done in a different way?

Are negative examples really necessary?

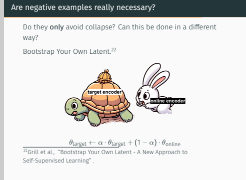
Do they **only** avoid collapse? Can this be done in a different way?

Bootstrap Your Own Latent.[22]



$$\theta_{\text{target}} \leftarrow \alpha \cdot \theta_{\text{target}} + (1 - \alpha) \cdot \theta_{\text{online}}$$

[22] Grill et al., "Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning" .



- 2 neural networks, an online (predictive) network and a target network. They use only positive examples, and for those the online network tries to predict the metric representation from the target network.
- The parameters of the target network are updated after every iteration with an exponential moving average of the online parameters.
- The authors argue that since the update to the target parameters is not exactly according to the gradient of the loss with respect to θ_{target}, there is no a priori reason to believe that the target network would converge to a collapsed representation.
- Informal discussion on whether batch normalization plays a role.

Applications

Applications

Sentence-BERT[23]

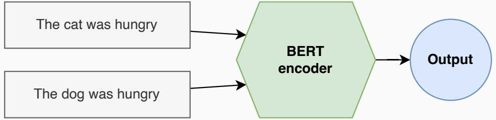
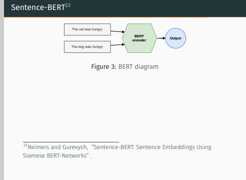


Figure 3: BERT diagram

[23] Reimers and Gurevych, "Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks" .

2023-10-23



Sentence-BERT

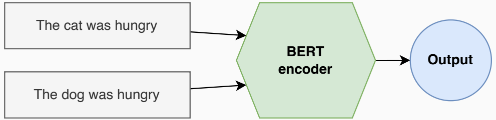


Figure 3: BERT diagram

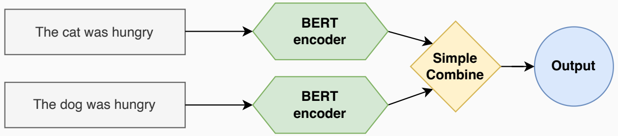
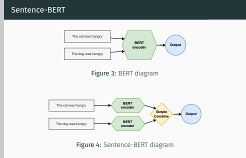


Figure 4: Sentence-BERT diagram



[a] Reimers and Gurevych, "Sentence-BERT: Sentence Embeddings Using

Explain the difference in approach using the diagrams

Sentence-BERT

- 10000 sentences, find most similar pair, **65 hour** $\rightarrow$ **5 seconds**.

Sentence-BERT
- 10000 sentences, find most similar pair, 65 hour → 5 seconds.

Sentence-BERT

+ 10000 sentences, find most similar pair, **65 hour $\rightarrow$ 5 seconds**.
+ Allows fast k-NN approximations.

Sentence-BERT

+ 10000 sentences, find most similar pair, **65 hour $\rightarrow$ 5 seconds**.
+ Allows fast k-NN approximations.

Sentence-BERT

+ 10000 sentences, find most similar pair, **65 hour** $\rightarrow$ **5 seconds**.
+ Allows fast k-NN approximations.
+ Great performance for STS.

Sentence-BERT

+ 10000 sentences, find most similar pair, 65 hour → 5 seconds.
+ Allows fast k-NN approximations.
+ Great performance for STS.

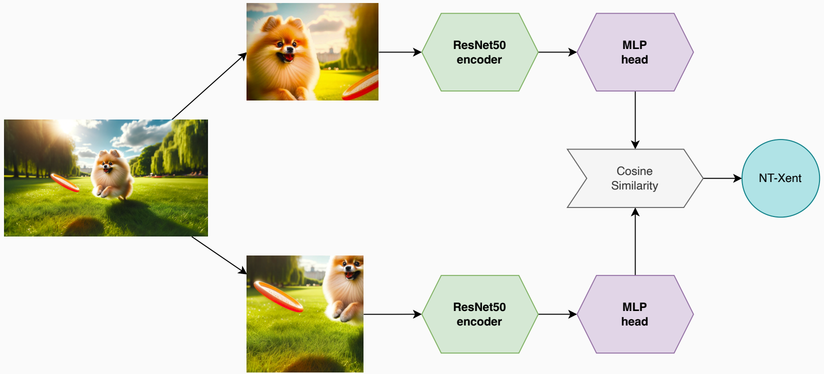
Sentence-BERT

+ 10000 sentences, find most similar pair, **65 hour** $\rightarrow$ **5 seconds**.
+ Allows fast k-NN approximations.
+ Great performance for STS.
- Worse performance on more complex tasks.

Sentence-BERT

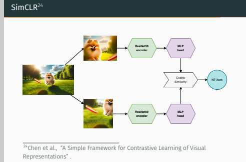
+ 10000 sentences, find most similar pair, **65 hour** $\rightarrow$ **5 seconds**.
+ Allows fast k-NN approximations.
+ Great performance for STS.
- Worse performance on more complex tasks.

SimCLR[24]



[24] Chen et al., "A Simple Framework for Contrastive Learning of Visual Representations" .

[a] Chen et al., "A Simple Framework for Contrastive Learning of Visual



- Go over diagram.
- Other examples in batch assumed negatives.

SimCLR

- State-of-the-art performance in self-supervised, semi-supervised (1% and 10% of labels) and transfer learning for classification tasks.

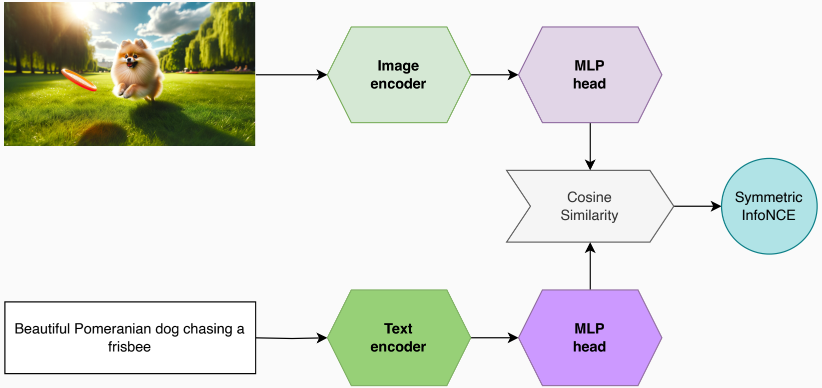
SimCLR
- State-of-the-art performance in self-supervised, semi-supervised (1% and 10% of labels) and transfer learning for classification tasks.

- State-of-the-art performance in self-supervised, semi-supervised (1% and 10% of labels) and transfer learning for classification tasks.
- Competitive performance with fully-supervised models on ImageNet (with 4x the model size, but without labeled data).

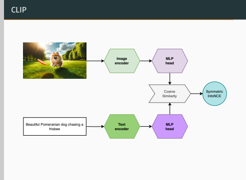
SimCLR
- State-of-the-art performance in self-supervised, semi-supervised (1% and 10% of labels) and transfer learning for classification tasks.
- Competitive performance with fully-supervised models on ImageNet (with 4x the model size, but without labeled data).

Evaluations using a linear classifier on the learned representation

CLIP



2023-10-23



CLIP

+ Zero-shot classification (including geo-localization, OCR, facial emotion recognition, action recognition…)
- Well below SOTA performance.

CLIP

+ Zero-shot classification (including geo-localization, OCR, facial emotion recognition, action recognition…)
- Well below SOTA performance.

Conclusions

Conclusions

Conclusions

- Great potential for representation learning, specially self-supervised.
- Many challenges: resource intensity, zero-shot low performance, false negatives...

└─Conclusions

Conclusions

- Great potential for representation learning, specially self-supervised.
- Many challenges: resource intensity, zero-shot low performance, false negatives...