16/1/2026

BSC AI Factory

Generative and Self-Supervised ML

Self-Supervised Learning

Javier Béjar - UPC

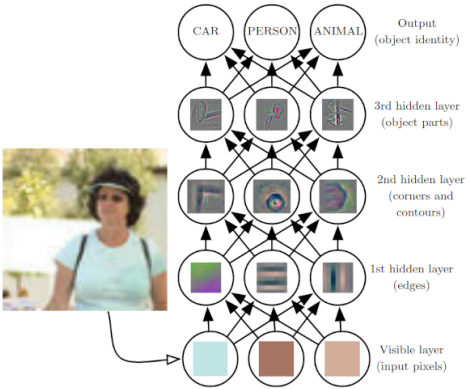
BSC AI Factory Partners

Affiliated entities

Introduction

Features Matter

- It is easier to learn a task from examples if they are represented by adequate features
- Deep Learning is about learning good representations from the data
  - Representations are obtained at different levels of granularity
  - The hierarchy of features is refined each level to obtain more semantically relevant features



Why Self Supervised Learning?

- It is very expensive to label datasets for a specific task (especially for deep learning)
  - Define categories, decide label criteria, create applications for labelling, hire people to label...
- Quality labeled data can be very expensive
  - Anyone can label a cat, but not a tumour
- Very large unlabelled datasets can be collected (on the wild or from data repositories)
- Self Supervised pretrained networks can improve supervised tasks
  - Data efficiency (less labelled data)
  - Performance (as good as features from labelled data)

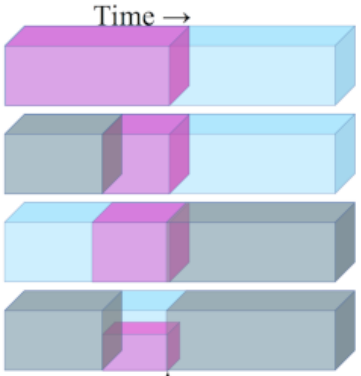
What is Self Supervised Learning

- Self Supervised Learning is a field of Unsupervised Learning where data provides the supervision
- Usually the task involves:
  - hiding a part of the example, so the network has to predict from the remaining information the rest
  - solving common sense tasks, as predicting rotations, relative positions, jigsaw puzzles...
  - predicting the characteristics from a complex transformation of an example, so the network associates similar representation no matter how transformed they are
- The difference among the methods is what loss or pretext tasks are used for learning

Reconstruction

Predicting from the context

- Predict any part of the input from any other part.
- Predict the future from the past.
- Predict the future from the recent past.
- Predict the past from the present.
- Predict the top from the bottom.
- Predict the occluded from the visible
- Pretend there is a part of the input you don't know and predict that.



Slide: LeCun

- Word embeddings are vectorial word representations that allow representing a vocabulary as points in an N-dimensional space
- Each word has a code that can be used as the input of a neural network (or other learning model)
- Models based on estimating the probability of words (uni-grams/bi-grams)

$$P(w_1, w_2, \ldots, w_n) = \prod_{i=1}^{n} P(w_i)$$

$$P(w_1, w_2, \ldots, w_n) = \prod_{i=2}^{n} P(w_i|w_{i-1})$$

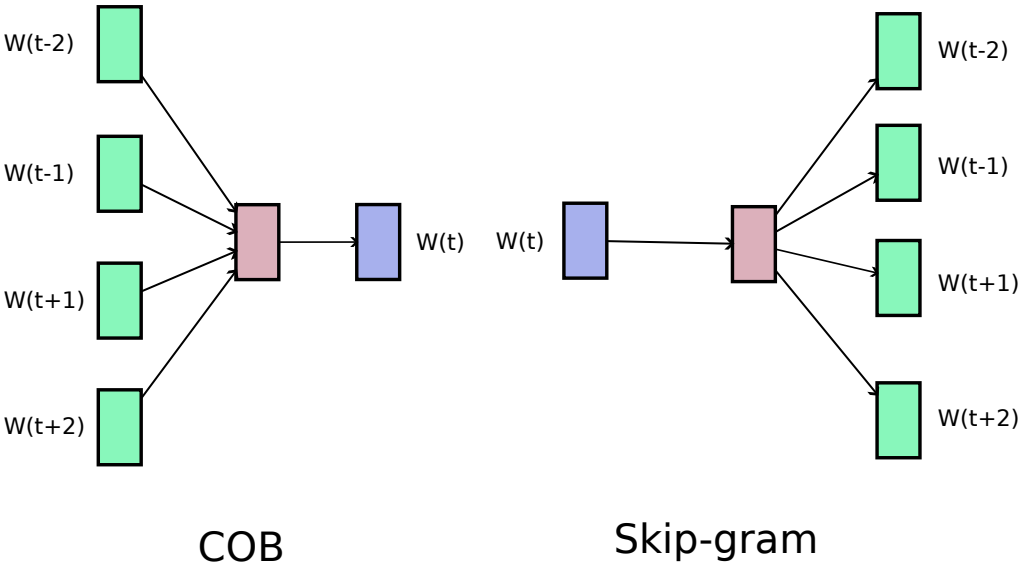
- Continuous bag of words (generalizes the bag of words representation)
  - A word is predicted using the surrounding words

$$P(w_i|w_{i-c},\ldots,w_{i-1},w_{i+1},\ldots,w_{i+c})$$

- Skip gram
  - From a word we predict all the surrounding words

$$P(w_{i-c},\ldots,w_{i-1},w_{i+1},\ldots,w_{i+c}|w_i)$$

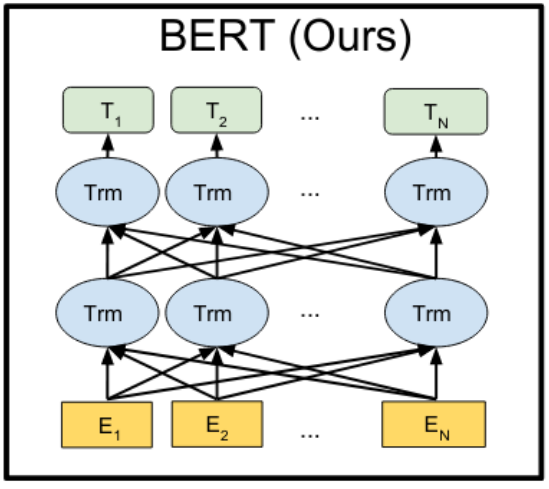
Word2Vec: Architectures

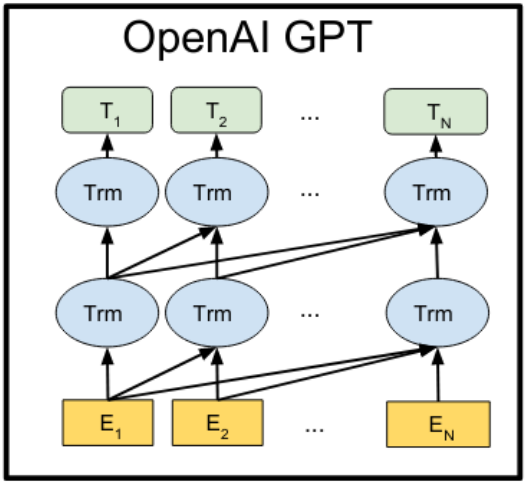


Explosion of Self Supervised models for NLP tasks

- A growing generation of models of word embeddings has emerged from this initial work
- Large corpus of text allow obtaining very good embeddings
  - GPT (Generative Pre-trained Transformer)
    - Transformer architecture, autoregressive model, pre-trained for several tasks, specially for next word prediction
  - BERT (Bidirectional Encoder Representations from Transformers)
    - Transformer architecture, bidirectional language model, fine-tuning for specific NLP tasks

GPT vs BERT

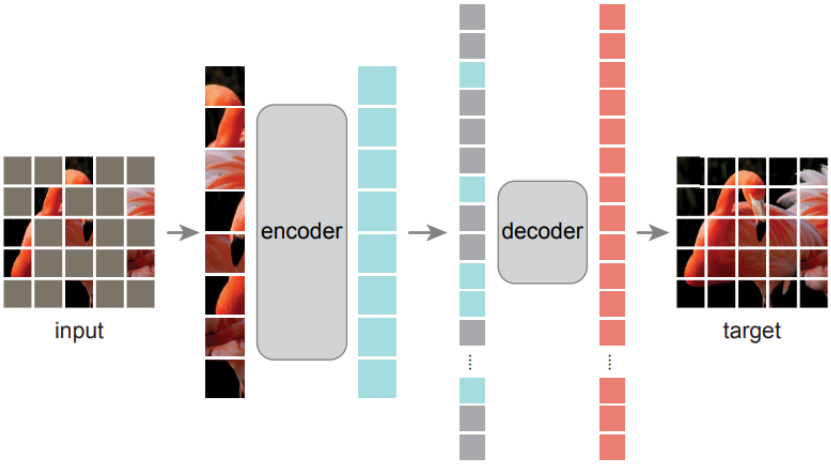

BERT (Ours)
T1
T2
...
TN
Trm
Trm
...
Trm
Trm
Trm
...
Trm
E1
E2
...
EN


OpenAI GPT
T1
T2
...
TN
Trm
Trm
...
Trm
Trm
Trm
...
Trm
E1
E2
...
EN

Masked Autoencoders (MAE) (Facebook)

- A similar idea can be used for images training with masking
- Images are cut in patches, and a percentage is transformed by an encoder to obtain a sequence of tokens
- The sequence is completed by introducing mask tokens at the positions where the discarded patches were, so the masked image is complete
- A decoder predicts all the tokens corresponding to the image from the masked input
- A final layer reconstructs the original image from the tokens, the loss function only takes in account the masked tokens
- Once trained, the encoder can be used for extracting characteristics of images for other tasks

Masked Auto Encoder (MAE)

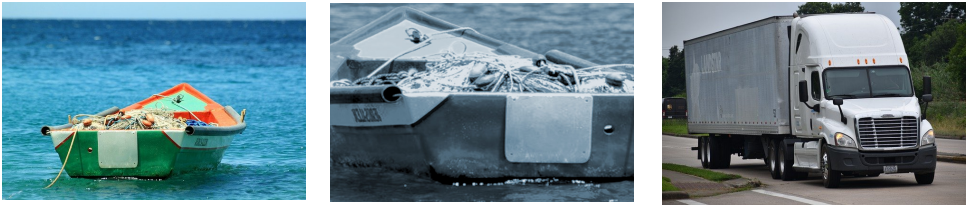


Contrastive Learning

Instance Discrimination

- To use patches or elements of a sequence can be data intensive
- Another approach is to work directly with whole examples
- We can obtain a representation by learning codes that put close examples that are considered the same and far examples that are considered different (contrast)
- This will obtain a space where examples will be easier to discriminate according to their characteristics
- For images we can obtain different versions of an image by using augmentation techniques and contrast them with different images
- We put close the augmented images (they are the same) put farther away the rest (they are different)

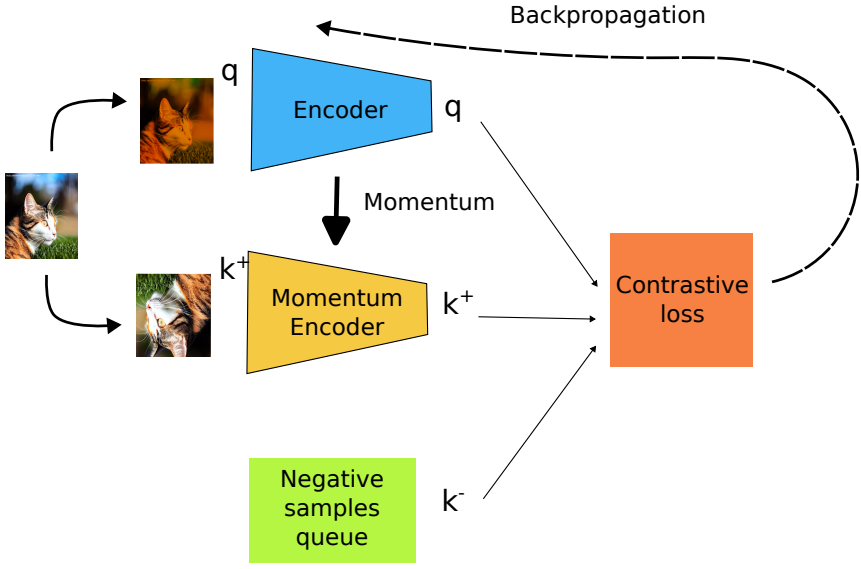
Instance Discrimination



Momentum Contrast (Facebook)

- Contrastive learning can be interpreted as the task of building dynamic dictionaries
- These dictionaries are composed by codes that are obtained using an encoding network
- The learning process trains the encoder to do dictionary lookups
- The obtained code for a new query should be similar to its matching key (a similar element), and different from other keys
- This is obtained by minimizing a contrastive loss
- The goal is to obtain dictionaries that are large and consistent during training

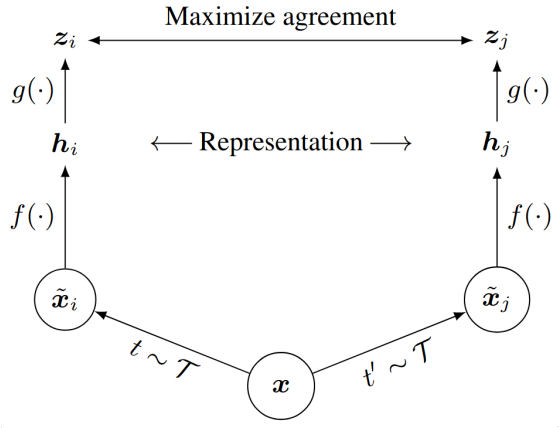
Momentum Contrast - Training



SimCLR (Google)

- SimCLR does not use a queue of negative examples, these are encoded in the same way as the positives (negative samples from the batch)
- An encoding network for the samples, and a non-linear transformation (MLP) between the code and the contrastive loss
- The MLP projects the codes to a lower dimensional space
- That projection is used to compute the constrastive loss (this reduces the curse of dimensionality)
- The projection network is discarded after training

SimCLR: Procedure

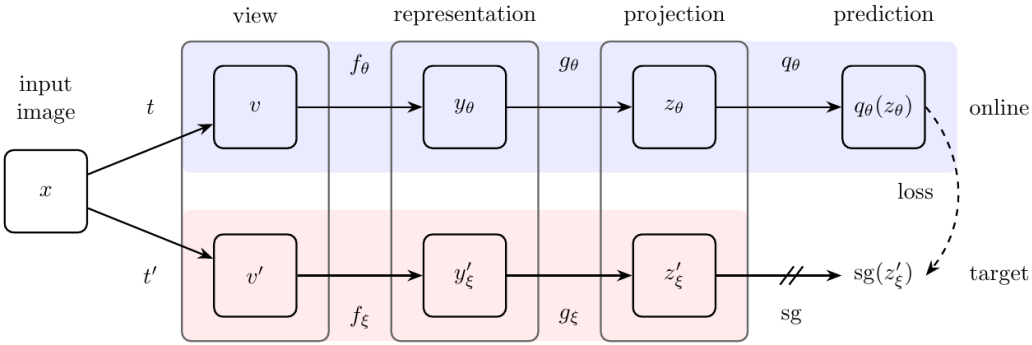


Non contrastive learning

Bootstrap your own latent (BYOL) (Google)

- It turns out that a non constrastive approach also works
- BYOL follows a similar idea for training
  - A teacher and a student network
  - Two augmentations of an example are coded by each network
  - The student learns to predict the representation of the teacher using regression (MSE loss)
- Only the student is updated with the loss and the teacher is updated also using momentum

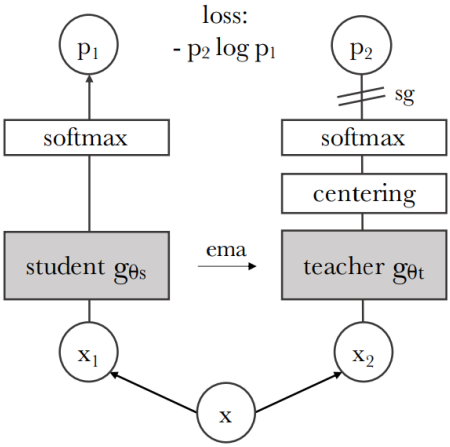
BYOL


view
representation
projection
prediction
input image
t
t'
x
v
v'
f_θ
f_ξ
y_θ
y'_ξ
g_θ
g_ξ
z_θ
z'_ξ
q_θ
$q_\theta(z_\theta)$
online
loss
$\mathrm{sg}(z'_\xi)$
target
sg

DINO (Facebook)

- Based on the Vision Transfomer (ViT)
- Defined as a problem knowledge distillation
- The network solves an artificial classification problem where classes are obtained as the probability for k outputs computing a sofmax
- A student network has to reproduce the probabilities of the teacher
- Each network receives image augmentations, but the teacher receives the complete image and the student a clipped image
- The loss function is cross entropy
- As usual the student is updated with the loss and the teacher uses momentum

DINO


loss:
- p2 log p1
p1
p2
sg
softmax
softmax
centering
student gθs
ema
teacher gθt
x1
x2
x

DINO: Attention Maps

- DINO is able to learn object segmentations without supervision
- The attention maps focus on the principal object in the image
- This is the base of the Segment Anything Model (SAM) from Facebook

Multi Modal Self Supervised Learning

- Multi modal data can not be compared directly
- Embedding each modality to separate spaces does not allow comparing either
- Self supervised techniques can be used to learn a joint embedding that has low distances for matching examples from different modalities
- This can be learned contrastively, making representations of different modalities of the same sample close and different samples from any modality far

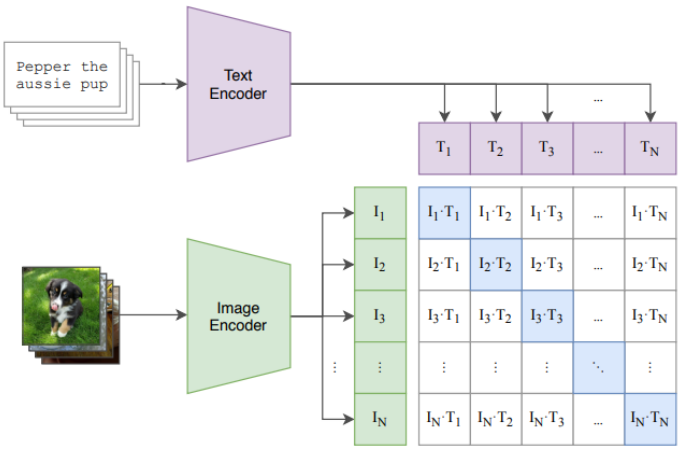
CLIP (OpenAI)

- CLIP (Contrastive Language-Image Pre-training) trains a text encoder and an image encoder with paired images (image-caption)
- Image encoder (ViT) and text encoder (Transformer) have the same output dimension
- Image and Text embeddings are compared using cosine similarity

$$sim(x, y) = \frac{x \cdot y}{||x||||y||}$$

- Trained with massive dataset (400M image-text pairs)

CLIP training



CLIP Applications

- Feature extraction
- Text based image search
- Zero Shot classifier (with open set of labels, “A photo of ...”)
- Evaluation of text to image models (CLIP Score)

This Python Notebook shows examples of self supervised models

- Autoregressive Models Notebook (click here to open the notebook in colab)

If you download the notebook you will be able to use it locally (run jupyter notebook/visual code to open the notebooks)
This notebook needs Torch for GPU (with CUDA) for a fast execution.

BSC AI Factory

AI Factories

Co-funded by the European Union

Plan de Recuperación, Transformación y Resiliencia

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101234399 and Spain, Portugal, Romania and Türkiye. The JU receives support from the European Union's Horizon Europe Programme.

BSC AI Factory Partners

Affiliated entities