AALBORG UNIVERSITY
DENMARK

Self-Supervised Learning

Sean Bin Yang, Jilin Hu

Center for Data-intensive Systems

{seany, hujilin}@cs.aau.dk

Outline

- Self-supervised learning
  - Motivation
  - Self-supervised learning versus other machine learning techniques
- Self-supervised learning from images
  - Geometric transformation recognition (image rotation)
  - Patches (relative patch position)
  - Generative modeling (context encoders)
  - Automated label generation (deep clustering)
  - Contrastive learning (CPC, other contrastive approaches)
- Self-supervised learning on graphs
  - Node representation learning (DGI, GCC)
  - Graph representation learning (InfoGraph)
  - Other SSL approaches
- Self-supervised learning for NLP
  - BERT
  - GPT3

Supervised vs Unsupervised Learning

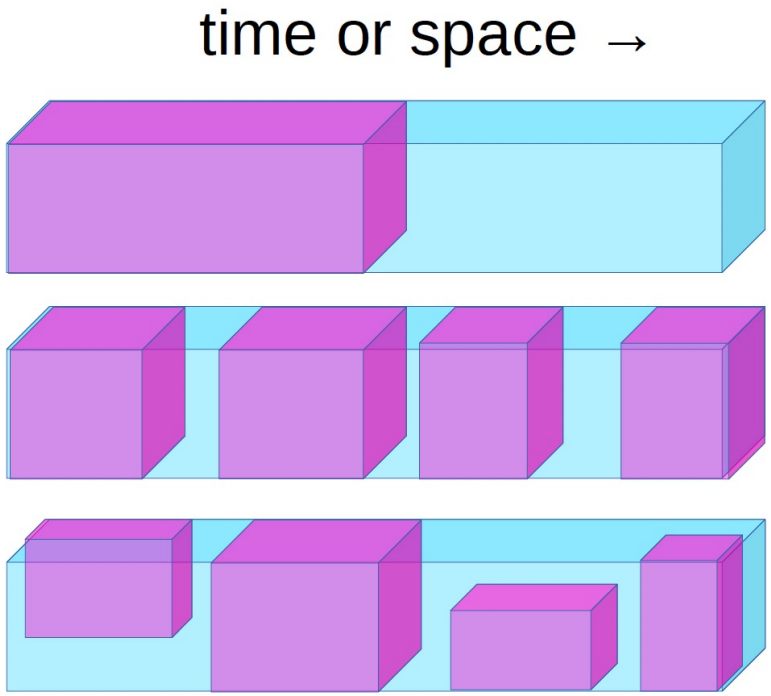
- ***Supervised learning*** – learning with labeled data
  - Approach: collect a large dataset, manually label the data, train a model, deploy
  - It is the dominant form of ML at present
  - Learned feature representations on large datasets are often transferred via pre-trained models to smaller domain-specific datasets
- ***Unsupervised learning*** – learning with unlabeled data
  - Approach: discover patterns in data either via clustering similar instances, or density estimation, or dimensionality reduction …

What is Self-Supervision?

A form of unsupervised learning where data provides the supervision.

Self-Supervised Leanrning =Filling in the Blanks

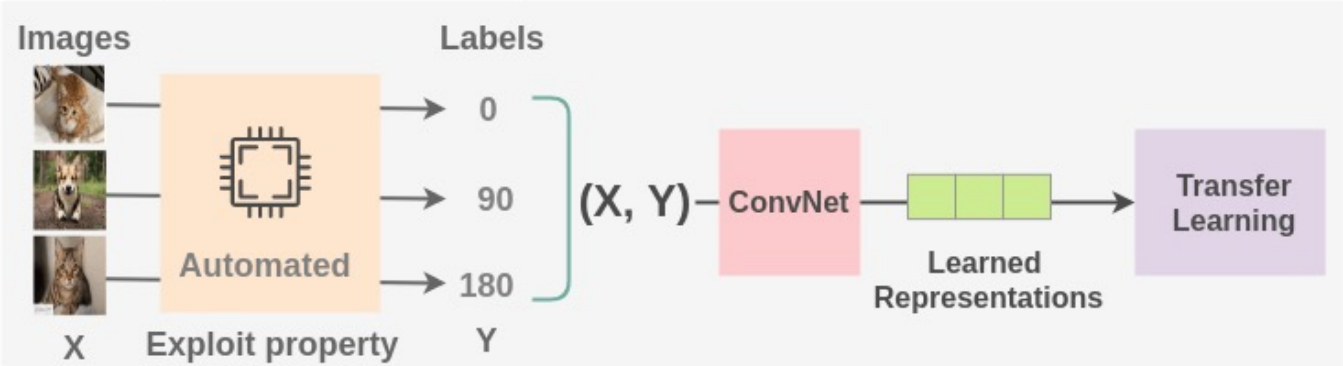
- Predict any part of the input from any other part.
- Predict the future from the past.
- Predict the masked from the visible.
- Predict the any occluded part from all available parts.
- Pretend there is a part of the input you don't know and predict that.
- Reconstruction = SSL when any part could be known or unknow.



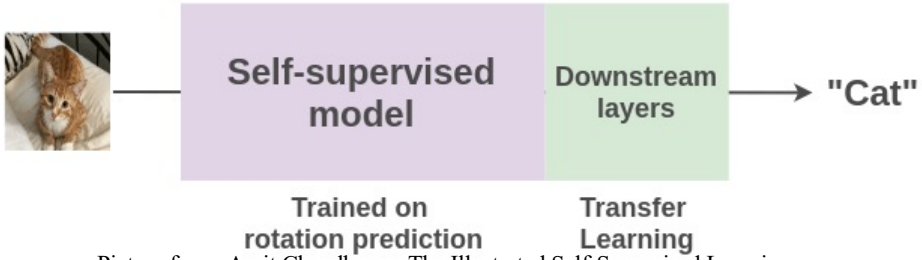
* Slides from Self-Supervised Learing by Yann LeCun

Self-Supervised Learning

- Self-supervised learning example
  - ***Pretext task***: train a model to predict the rotation degree of rotated images with cats and dogs (we can collect million of images from internet, labeling is not required)



  - ***Downstream task***: use transfer learning and fine-tune the learned model from the pretext task for classification of cats vs dogs with very few labeled examples



Picture from: Amit Chaudhary – The Illustrated Self-Supervised Learning

Self-Supervised Learning

- Why self-supervised learning?
  - Creating labeled datasets for each task is an expensive, time-consuming, tedious task
    - Requires hiring human labelers, preparing labeling manuals, creating GUIs, creating storage pipelines, etc.
    - High quality annotations in certain domains can be particularly expensive (e.g., medicine)
  - Self-supervised learning takes advantage of the vast amount of unlabeled data on the internet (images, graph, text)
    - Rich discriminative features can be obtained by training models without actual labels
  - Self-supervised learning can potentially generalize better because we learn more about the world
- ***Challenges*** for self-supervised learning
  - How to select a suitable pretext task for an application
  - There is no gold standard for comparison of learned feature representations
  - Selecting a suitable loss functions, since there is no single objective as the test set accuracy in supervised learning

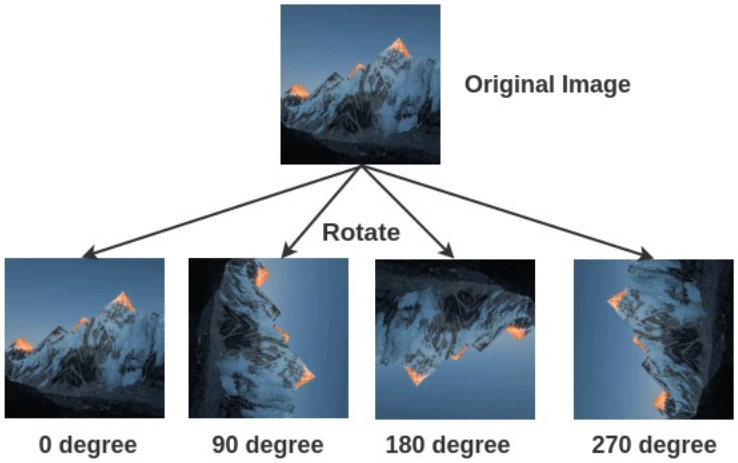
Self-Supervised Learning

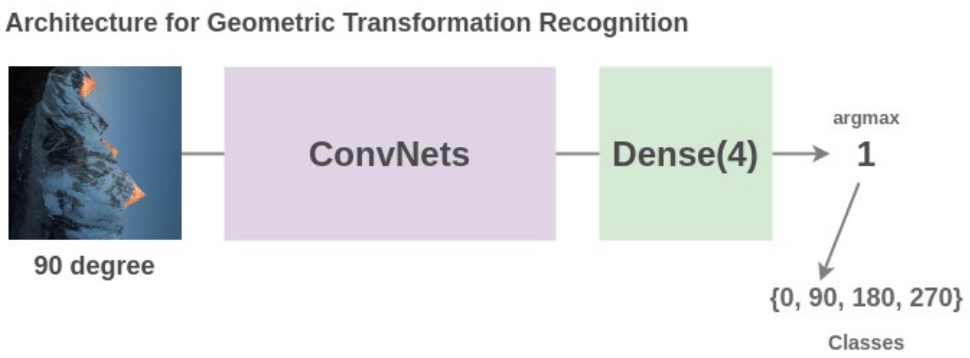
- Self-supervised learning versus unsupervised learning
  - Self-supervised learning (SSL)
    - Aims to extract useful feature representations from raw unlabeled data through ***pretext tasks***
    - Apply the feature representation to improve the performance of ***downstream tasks***
  - Unsupervised learning
    - Discover patterns in unlabeled data, e.g., for clustering or dimensionality reduction
  - Note also that the term "self-supervised learning" is sometimes used interchangeably with "unsupervised learning"
- Self-supervised learning versus transfer learning
  - Transfer learning is often implemented in a supervised manner
    - E.g., learn features from a labeled ImageNet, and transfer the features to a smaller dataset
  - SSL is a type of transfer learning approach implemented in an unsupervised manner
- Self-supervised learning versus data augmentation
  - Data augmentation is often used as a regularization method in supervised learning
  - In SSL, image rotation of shifting are used for feature learning in raw unlabeled data

Part I
Self-Supervised Learning from Images

Image Rotation

- ***Geometric transformation recognition***
  - Gidaris (2018) - Unsupervised Representation Learning by Predicting Image Rotations
- Training data: images rotated by a multiple of 90° at random
  - This corresponds to four rotated images at 0°, 90°, 180°, and 270°
- Pretext task: train a model to predict the rotation degree that was applied
  - Therefore, it is a 4-class classification problem

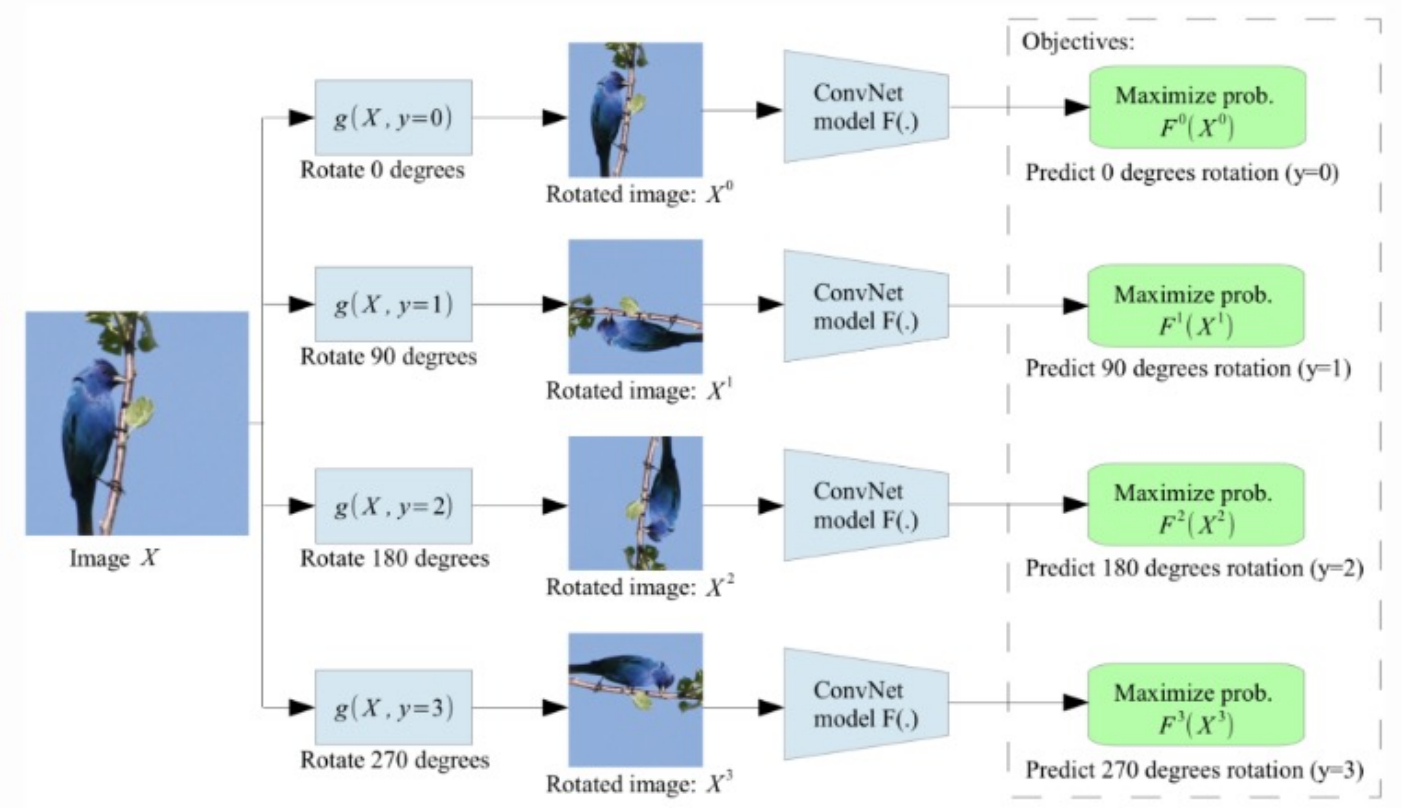




Picture from: Amit Chaudhary – The Illustrated Self-Supervised Learning

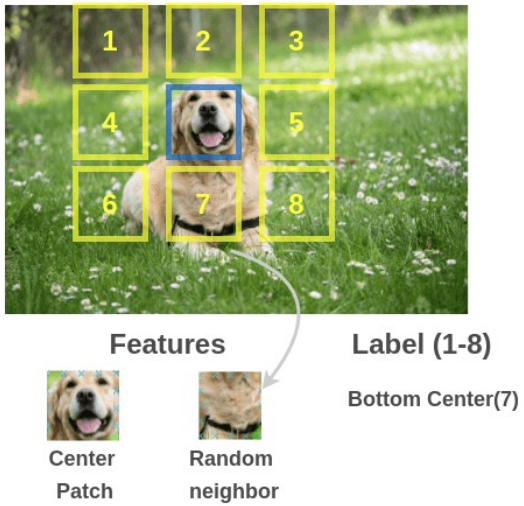
Image Rotation

- A single ConvNet model is used to predict one of the four rotations
  - The model needs to understand the location and type of the objects in images to determine the rotation degree



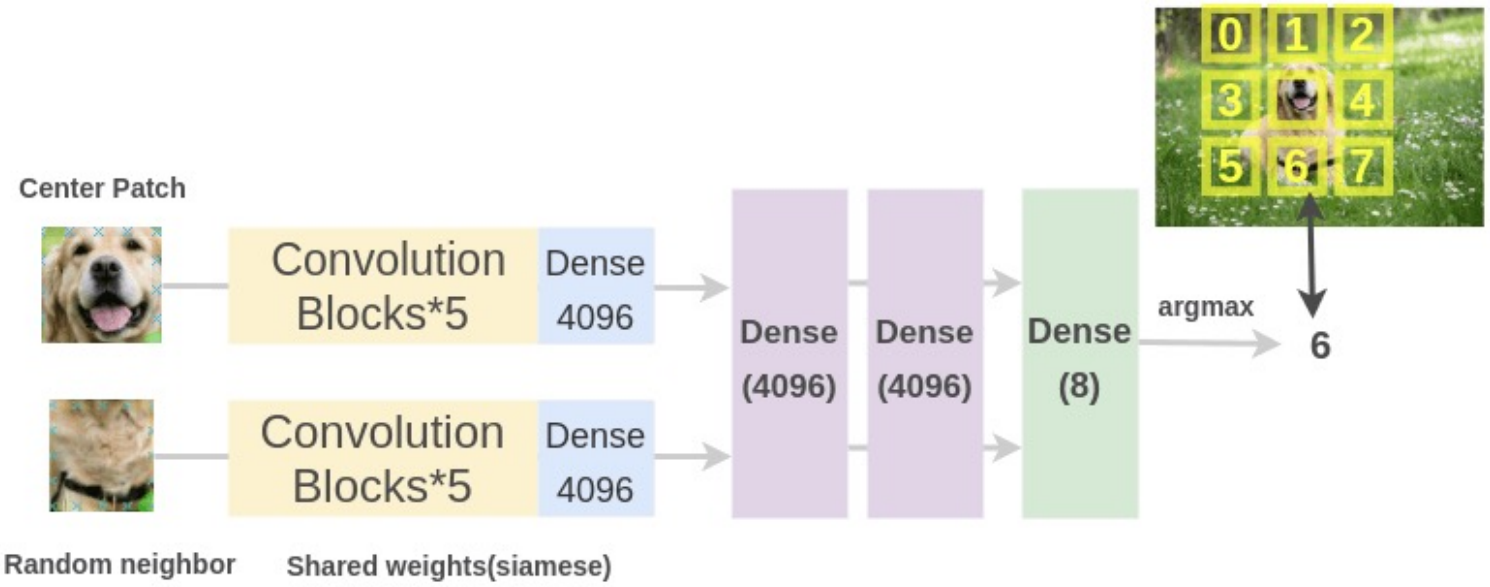
Relative Patch Position

- ***Relative patch position for context prediction***
  - Dorsch (2015) Unsupervised Visual Representation Learning by Context Prediction
- Training data: multiple patches extracted from images
- Pretext task: train a model to predict the relationship between the patches
  - E.g., predict the relative position of the selected patch below (i.e., position # 7)
    - For the center patch, there are 8 possible neighbor patches (8 possible classes)



Relative Patch Position

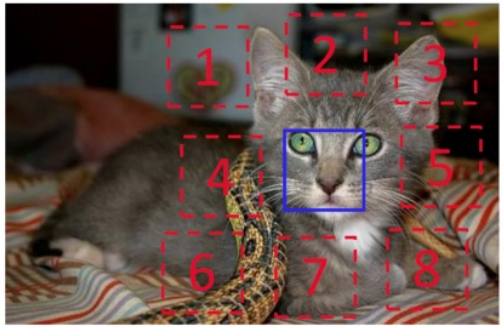
- The patches are inputted into two ConvNets with shared weights
  - The learned features by the ConvNets are concatenated
  - Classification is performed over 8 classes (denoting the 8 possible neighbor positions)
- The model needs to understand the spatial context of images, in order to predict the relative positions between the patches



Picture from: Amit Chaudhary – The Illustrated Self-Supervised Learning

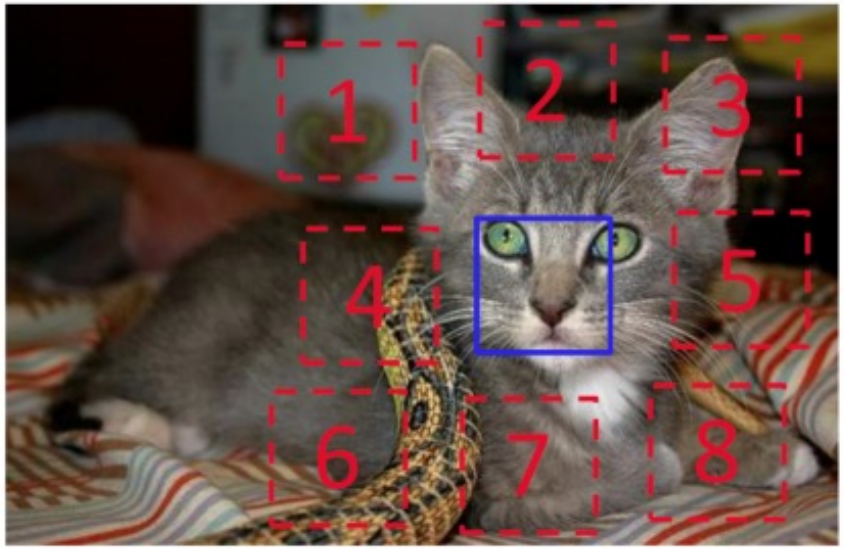
Relative Patch Position

- The training patches are sampled in the following way:
  - Randomly sample the first patch, and consider it the middle of a 3x3 grid
  - Sample from 8 neighboring locations of the first central patch (blue patch)
- To avoid the model only catching low-level trivial information:
  - Add gaps between the patches
  - Add small jitters to the positions of the patches
  - Randomly downsample some patches to reduced resolution, and then upsample them
  - Randomly drop 1 or 2 color channels for some patches



Relative Patch Position

- For instance, predict the position of patch # 3 with respect to the central patch



- Input: two patches $X = ($  ,  $)$

- Prediction: **Y = 3**

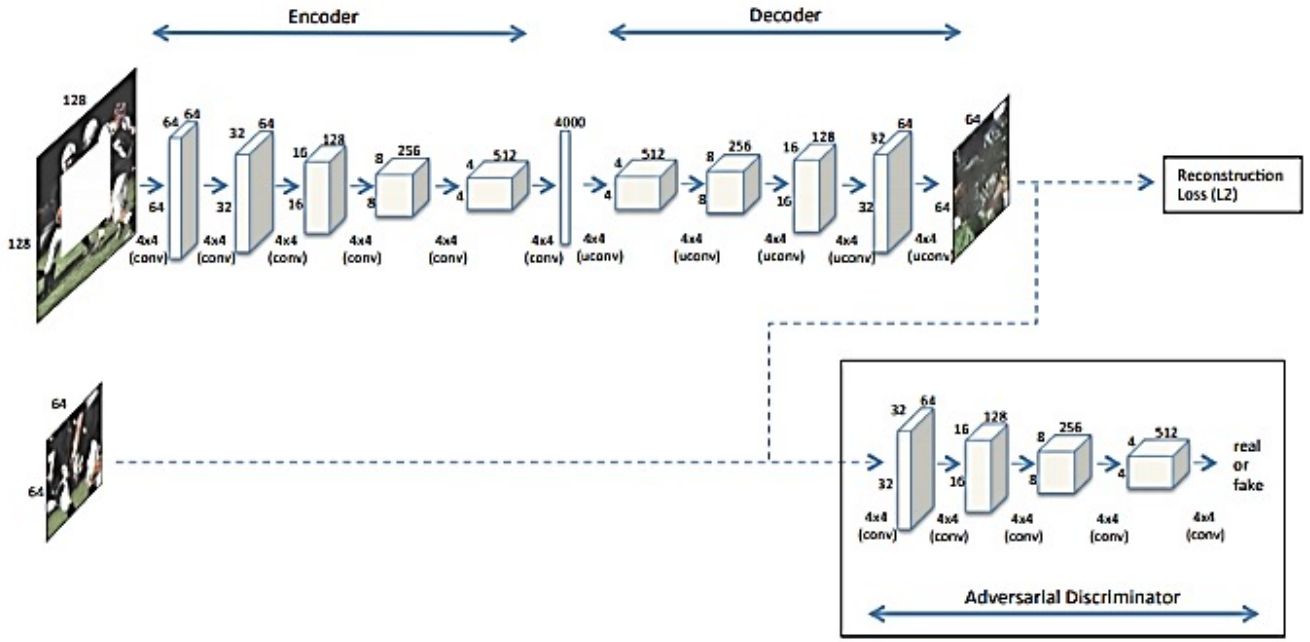
Context Encoders

- ***Predict missing pieces***, also known as ***context encoders***, or ***inpainting***
  - Pathak (2016) Context Encoders: Feature Learning by Inpainting
- Training data: remove a random region in images
- Pretext task: fill in a missing piece in the image
  - The model needs to understand the content of the entire image, and produce a plausible replacement for the missing piece

Picture from: Amit Chaudhary – The Illustrated Self-Supervised Learning

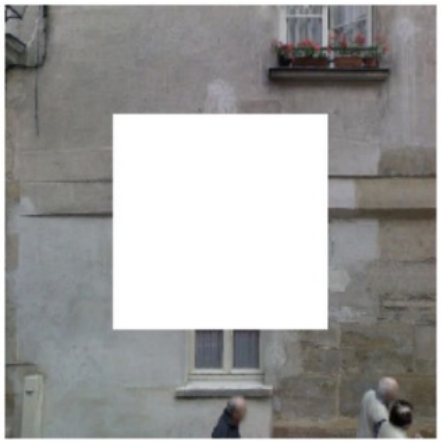
Context Encoders

- The authors found that the reconstruction loss alone couldn't capture fine details in the missing region
- Improvement was achieved by adding a GAN branch, where the generator of the GAN learns to reconstruct the missing piece
  - The overall loss function is a weighted combination of the reconstruction and the GAN losses, i.e., $\mathcal{L} = \lambda_{\text{rec}}\mathcal{L}_{\text{rec}} + \lambda_{\text{gan}}\mathcal{L}_{\text{gan}}$



Context Encoders

- The joint loss function $\mathcal{L} = \lambda_{\text{rec}}\mathcal{L}_{\text{rec}}+\lambda_{\text{gan}}\mathcal{L}_{\text{gan}}$ resulted in improved prediction of the missing piece



Input image

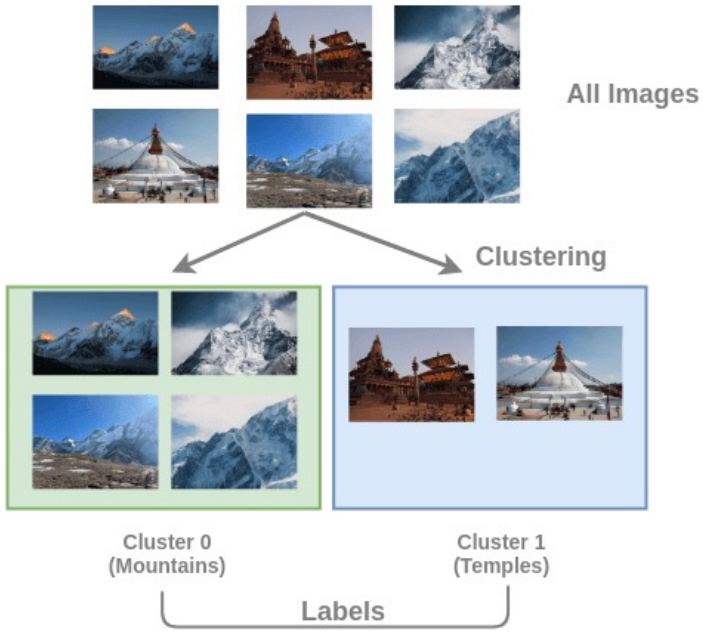
Encoder-decoder with reconstruction loss $\mathcal{L}_{\text{rec}}$

GAN with loss $\mathcal{L}_{\text{gan}}$

Joint loss $\mathcal{L} = \lambda_{\text{rec}}\mathcal{L}_{\text{rec}}+\lambda_{\text{gan}}\mathcal{L}_{\text{gan}}$

Deep Clustering

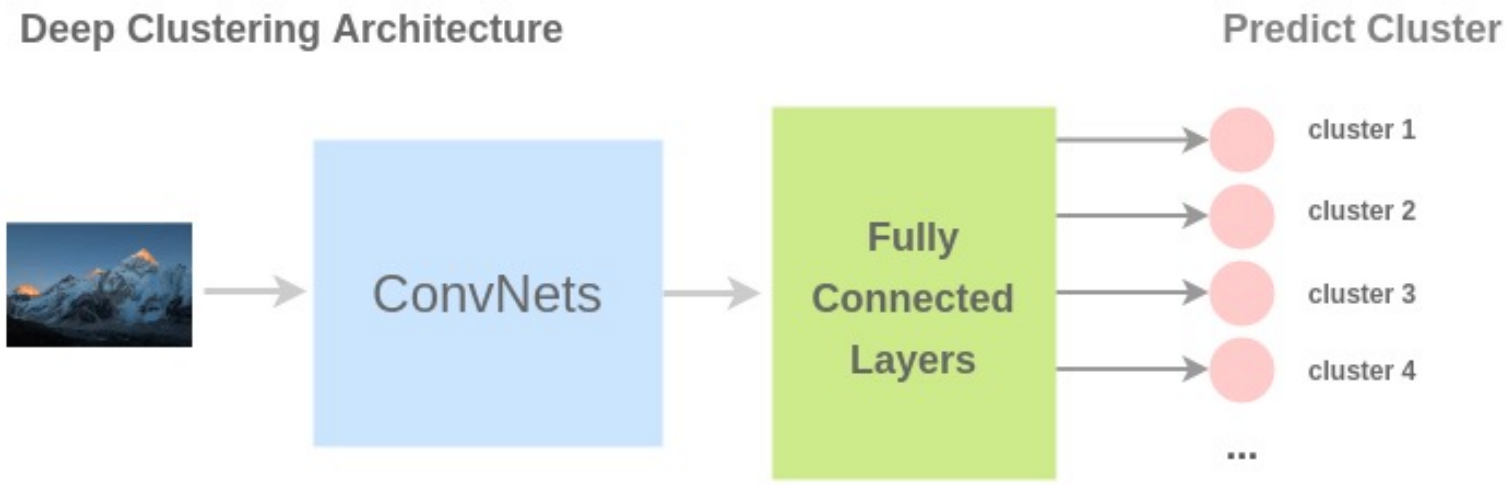
- ***Deep clustering of images***
  - Caron (2019) Deep Clustering for Unsupervised Learning of Visual Features
- Training data: clusters of images based on the content
  - E.g., clusters on mountains, temples, etc.
- Pretext task: predict the cluster to which an image belongs



Picture from: Amit Chaudhary – The Illustrated Self-Supervised Learning

Deep Clustering

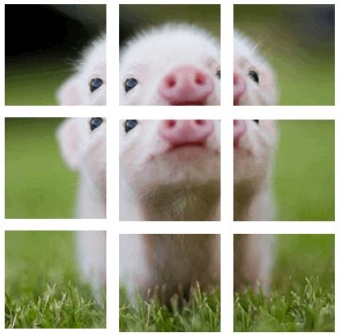
- The architecture for SSL is called deep clustering
  - The model treats each cluster as a separate class
  - The output is the number of the cluster (i.e., cluster label) for an input image
  - The authors used *k*-means for clustering the extracted feature maps
- The model needs to learn the content in the images in order to assign them to the corresponding cluster

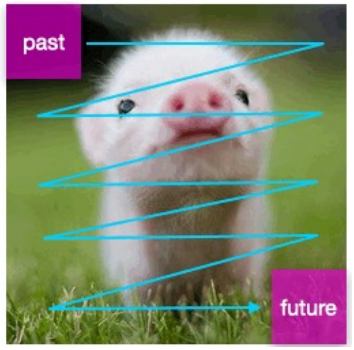


Picture from: Amit Chaudhary – The Illustrated Self-Supervised Learning

Contrastive Predictive Coding

- ***Contrastive Predictive Coding (CPC)***
  - Van der Oord (2018) Representation Learning with Contrastive Predictive Coding
- Training data: extracted patches from input images
- Pretext task: predict the order for a sequence of patches using contrastive learning
  - E.g., how to predict the next (future) patches based on encoded information of previous (past) patches in the image
- The approach was implemented in different domains: speech audio, images, natural language, graph representation learning and reinforcement learning

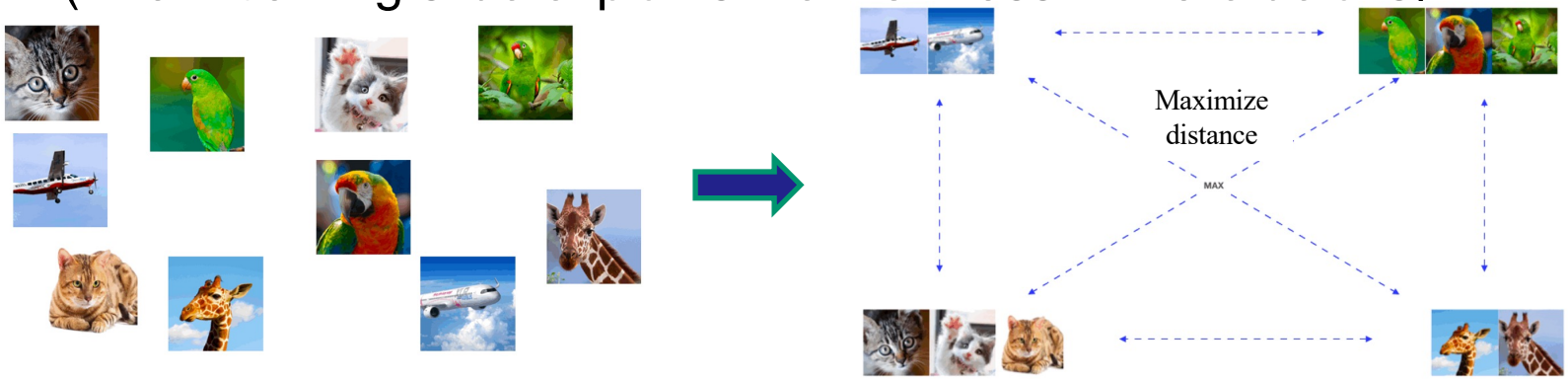




Picture from: William Falcon – A Framework for Contrastive Self-Supervised Learning and Designing a New Approach

Contrastive Predictive Coding

- Contrastive learning is based on grouping similar examples together
  - E.g., cluster the shown images into groups of similar images
- Noise-Contrastive Estimation (NCE) loss is commonly used in contrastive learning
  - The NCE loss minimizes the distance between similar images (positive examples) and maximizes the distance to dissimilar images (negative examples)
  - Other used terms are InfoNCE loss, or contrastive cross-entropy loss
  - (A forthcoming slide explains the NCE loss in more details)



Picture from: William Falcon – A Framework for Contrastive Self-Supervised Learning and Designing a New Approach

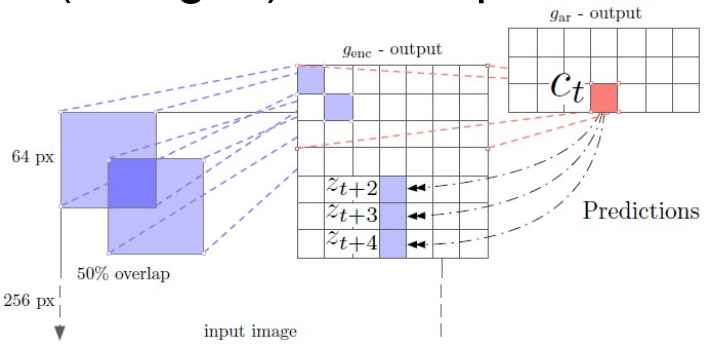
Contrastive Predictive Coding

- For an input image resized to 256×256 pixels, the authors extracted a grid of 7×7 patches of size 64×64 pixels with 50% overlap between the patches
  - Therefore, there are 49 overlapping patches in total for each image

Picture from: Pieter Abbeel, Lecture 7 – Self-Supervised Learning

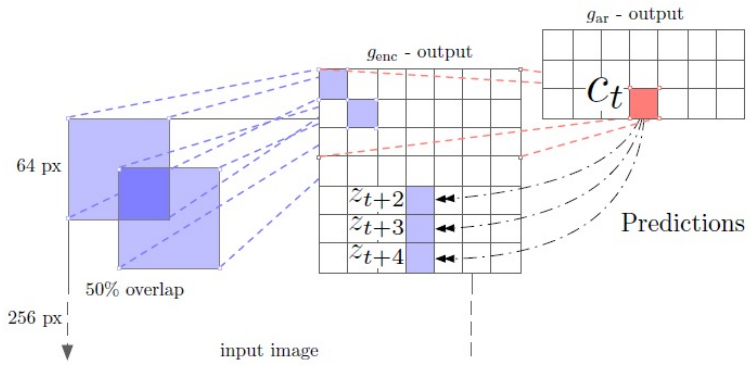
Contrastive Predictive Coding

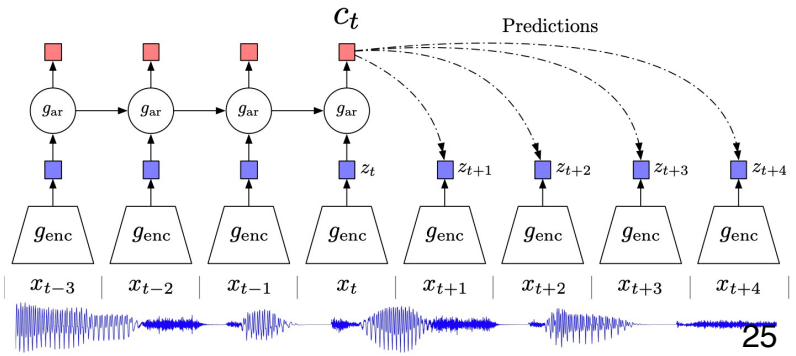
- An encoder g_{enc} is used to project each patch into a low-dimensional latent space
  - The latent representation obtained by encoders is often referred to as code (or context)
- E.g., the leftmost portion of the image depicts extracting patches of 64×64 pixels size with 50% overlap between the patches
  - A ResNet-101 encoder is used for projecting the patch x_t into a code representation z_t
  - The middle image shows the outputs of the encoder g_{enc} for each patch, $z_t = g_{\text{enc}}(x_t)$
  - For the 49 patches (7×7 grid), the outputs are 7×7×1,024 tensors (i.e., $z_1, z_2, \ldots, z_{49}$)



Contrastive Predictive Coding

- CPC considers the patches as an ordered sequence (e.g., like video frames)
  - An autoregressive model g_{ar} is used to predict the future patches in the sequence
    - The output of the autoregressive model for the shown red patch c_t (in row 3 and column 4) is the sum of all vectors $g_{\text{ar}}(z_{\leq t})$ for the previous patches in the sequence (e.g., all patches in the above rows and right columns of the red patch)
  - The code representation of the patch c_t is used to predict the blue patches in the next rows and the same column as the red patch, denoted z_{t+2}, z_{t+3} and z_{t+4}
  - The authors predicted up to five rows for each patch in the grid





Contrastive Predictive Coding

- The NCE loss is used for training the model on each patch in the image
  - The patch at position *t* is considered a positive sample, and all other patches are considered negative samples
- For a set of N random patches $\mathrm{X} = \{x_1, \ldots, x_N\}$, containing one positive sample x_t and $N - 1$ negative samples, the NCE loss is:

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

- c_t is the code representation of the patch at position *t*, i.e., $c_t = g_{\mathrm{ar}}(z_{\leq t})$
- x_{t+k} is a predicted patch in the sequence at position $t + k$
- $f_k(x_{t+k}, c_t)$ is a density function that approximates the probability $p_k(x_{t+k}|c_t)$ of estimating the patch x_{t+k} for a given code representation c_t
  - The authors used a log-bilinear model $f_k(x_{t+k}, c_t) = exp\left(z_{t+k}^T W_k c_t\right)$, where W_k is the matrix of weights for the prediction step k
- The numerator of the NCE loss represents the distance to the positive sample, and the denominator represents the distance to the negative samples

Other Contrastive SSL Approaches

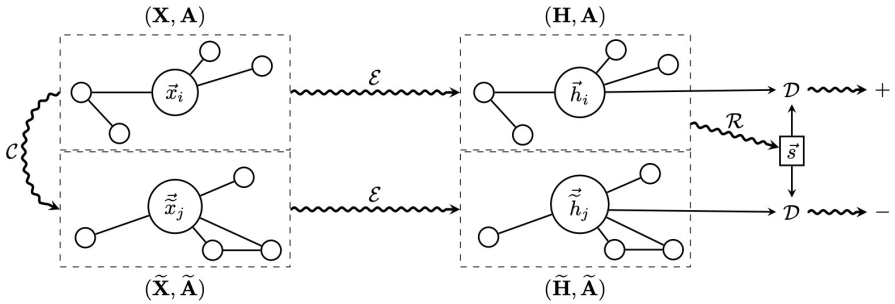
- ***SimCLR**, **a Simple framework for Contrastive Learning of Representations***
  - Chen (2020) A Simple Framework for Contrastive Learning of Visual Representations
- ***Augmented Multiscale Deep InfoMax*** or ***AMDIM***
  - Bachman (2019) Learning Representations by Maximizing Mutual Information Across Views
- ***Momentum Contrast*** or ***MoCo***
  - He (2019) Momentum Contrast for Unsupervised Visual Representation Learning
- ***Bootstrap Your Own Latent*** or ***BYOL***
  - Grill (2020) Bootstrap your own latent: A new approach to self-supervised Learning
- ***Swapping Assignments between multiple Views of the same image*** or ***SwAV***
  - Caron (2020) Unsupervised Learning of Visual Features by Contrasting Cluster Assignments
- ***Yet Another DIM*** or ***YADIM***
  - Falcon (2020) A Framework for Contrastive Self-Supervised Learning and Designing a New Approach

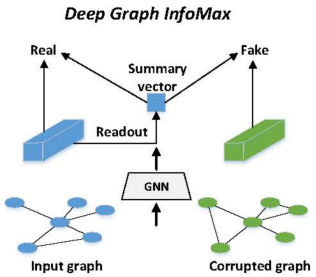
Part II
Self-Supervised Learning on Graphs

Deep Graph InfoMax

- ***Deep Graph InfoMax (DGI)***
  - Petar Velickovic (2019) Deep Graph InfoMax
- Training data: input graph and corrupted graph
- Pretext task: learn useful representation
  - E.g., learning representation for each node for graph or representation for whole graph
- Maximizing the mutual information between the graph representation and hidden representations.
- Noise-contrastive type objective with a standard binary cross-entropy (BCE) loss.

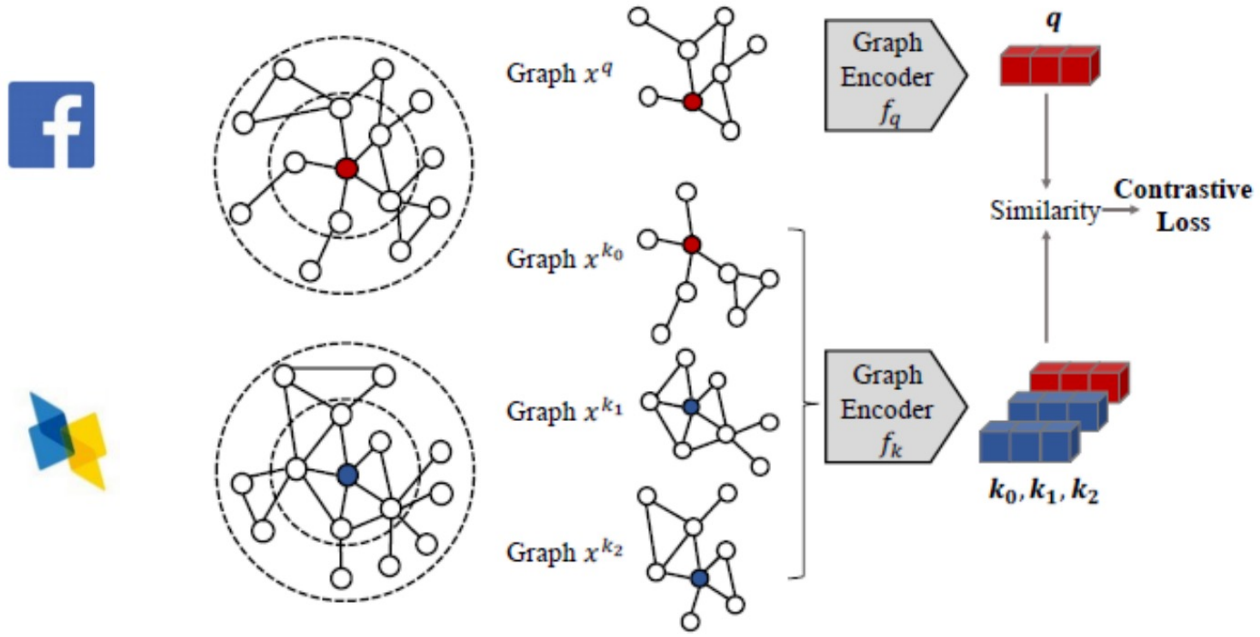
$$\mathcal{L} = \frac{1}{N+M}\left(\sum_{i=1}^{N}\mathbb{E}_{(\mathbf{X},\mathbf{A})}\left[\log\mathcal{D}\left(\vec{h}_i,\vec{s}\right)\right]+\sum_{j=1}^{M}\mathbb{E}_{(\widetilde{\mathbf{X}},\widetilde{\mathbf{A}})}\left[\log\left(1-\mathcal{D}\left(\vec{\tilde{h}}_j,\vec{s}\right)\right)\right]\right)$$





Graph Contrastive Coding

- ***Graph Contrastive Coding (GCC)***
  - Jiezhong Qiu(2020) GCC: Graph Contrastive Coding for Graph Neural Network Pre-Training
- Training data: input graph and other graph
- Pretext task: learn useful representation
  - E.g., learning representation for each node for graph or representation for whole graph

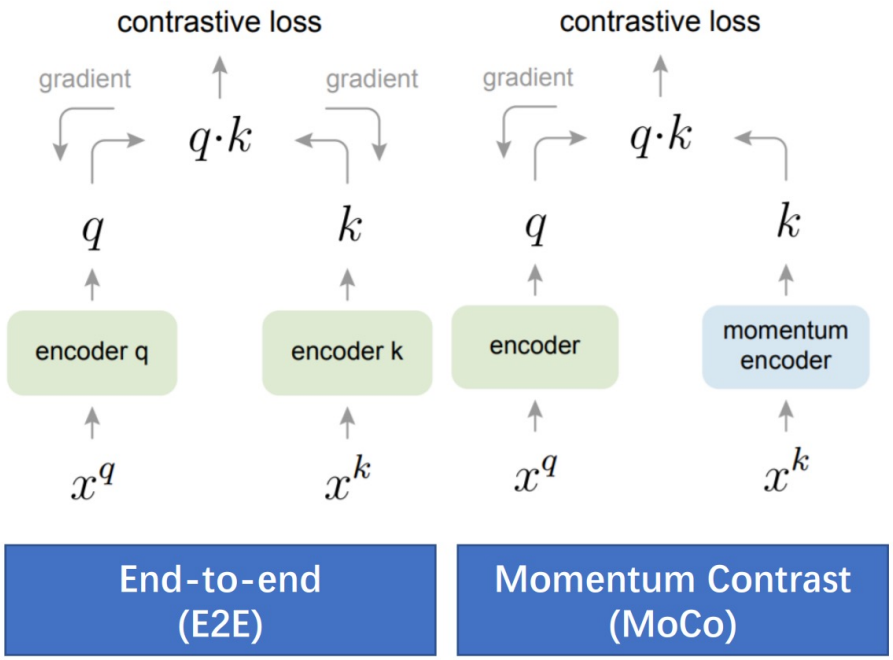


Graph Contrastive Coding

- GCC Pre-traning: Learning Algorithms
  - Encoded query q
  - K+1 encoded keys $\{k_0,\ldots,k_K\}$

$$\mathcal{L} = -\log \frac{\exp\left(\boldsymbol{q}^\top \boldsymbol{k}_+/\tau\right)}{\sum_{i=0}^{K} \exp\left(\boldsymbol{q}^\top \boldsymbol{k}_i/\tau\right)}$$

where τ is the temperature hyper-parameter. Here contrstive learning looks up a single key (denoted by k_+) that q matches in the dictionary.

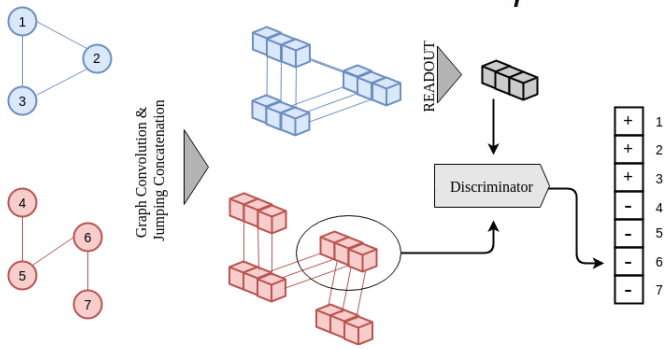


*Slides from GCC: Graph Contrastive Coding for Graph Neural Network Pre-Training

InfoGraph

- ***InfoGraph***
  - Fanyun Sun(2020) InfoGraph: Unsupervised Whole-Graph Representation Learning
- Training data: input graph and other graph
- Pretext task: learn useful representation using contrastive learning
- Maximizing the mutual information between the whole graph representation $H_\phi(G)$ and all the sub-structure representatio $\vec{h}^u_\phi$

$$\hat{\phi}, \hat{\psi} = \arg\max_{\phi,\psi} \sum_{G \in \mathbf{G}} \frac{1}{|G|} \sum_{u \in G} I_{\phi,\psi}(\vec{h}^u_\phi; H_\phi(G)).$$



- The Jensen-Shannon MI estimator:

$$I_{\phi,\psi}(h^i_\phi(G); H_\phi(G)) :=$$

$$\mathbb{E}_{\mathbb{P}}[-\mathrm{sp}(-T_{\phi,\psi}(\vec{h}^i_\phi(x), H_\phi(x)))] - \mathbb{E}_{\mathbb{P}\times\tilde{\mathbb{P}}}[\mathrm{sp}(T_{\phi,\psi}(\vec{h}^i_\phi(x'), G_\phi(x)))]$$

- Where x is an input example, x' is a negative graph sample, $\mathrm{sp}(z)=\log(1+e^z)$, T(,) is a neural network.

Other SSL Approaches on Graph

- ***Survey***
  - Wu(2021) Self-supervised on Graphs: Contrastive, Generative, or Predictive
- ***MERIT***
  - Jin (2021) Multi-Scale Contrastive Siamese Networks for Self-Supervised Graph Representation Learning
- ***Graph-level Representation Learning***
  - Xu (2021) Self-supervised Graph-level representation learning with local and global structure
- ***GMI***
  - Peng(2020) Graph Representation Learning via Graphical Mutual Information Maximization
- ***Contrastive Mutli-View***
  - Hassani (2020) Contrastive Multi-View Representation Learning on Graphs
- ***Combine self-supervision with GCNs***
  - Yuning You (2020) When Does Self-supervision Help Graph Convolutional Networks?
- ***Graph Contrastive***
  - Zhu(2020) Deep Graph Contrastive Representation Learning

Part III
Self-Supervised Learning for NLP

NLP

- Self-supervised learning has driven the recent progress in the ***Natural Language Processing*** (NLP) field
  - Models like ELMO, BERT, RoBERTa, ALBERT, Turing NLG, GPT-3 have demonstrated immense potential for automated NLP
- Employing various pretext tasks for leaning from raw text produced rich feature representations, useful for different downstream tasks
- Pretext tasks in NLP:
  - Predict the center word given a window of surrounding words
    - The word highlighted with green color needs to be predicted

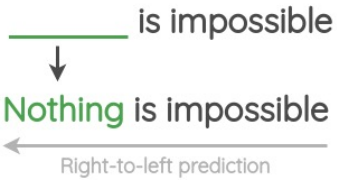
  - Predict the surrounding words given the center word

Picture from: Amit Chaudhary – Self-Supervised Representation Learning in NLP

NLP

- Pretext tasks in NLP:
  - From three consecutive sentences, predict the previous and the next sentence, given the center sentence

  - Predict the previous or the next word, given surrounding words



  - Predict randomly masked words in sentences

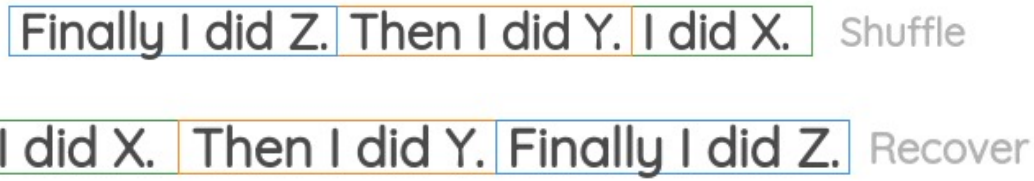
Picture from: Amit Chaudhary – Self-Supervised Representation Learning in NLP

NLP

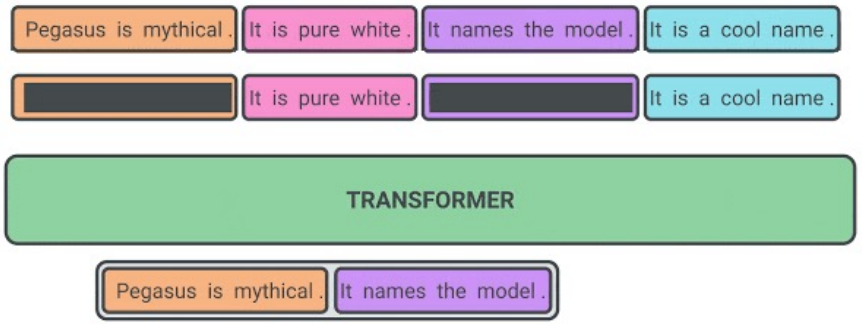
- Pretext tasks in NLP:
  - Predict if the ordering of two sentences is correct

Sentence 1	Sentence 2	Next Sentence
I am going outside	I will be back in the evening	yes
I am going outside	You know nothing John Snow	no

  - Predict the order of words in a randomly shuffled sentence



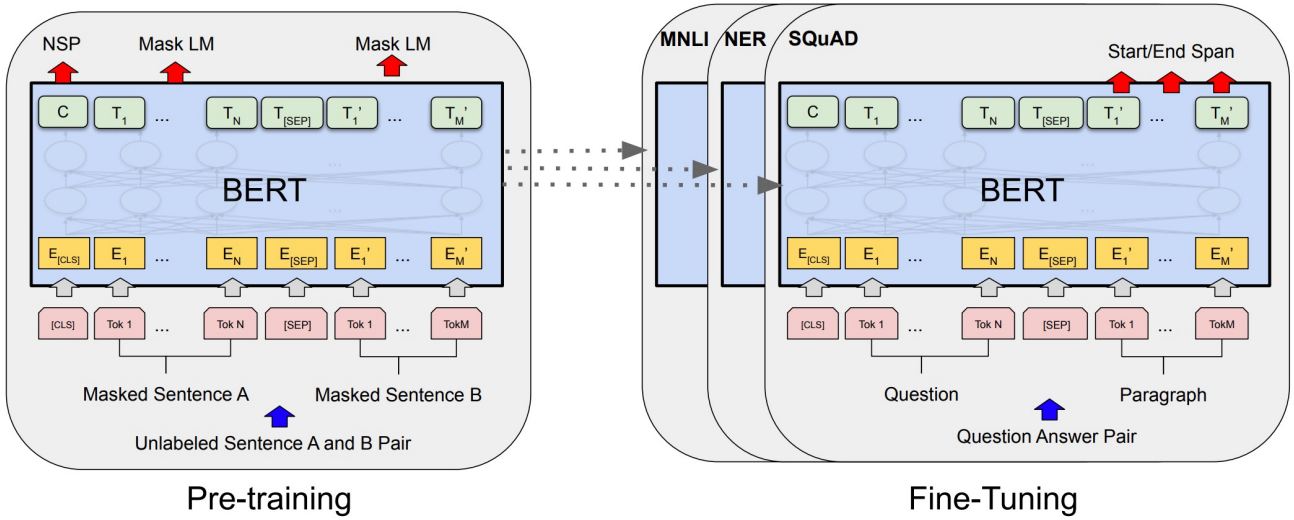
  - Predict masked sentences in a document

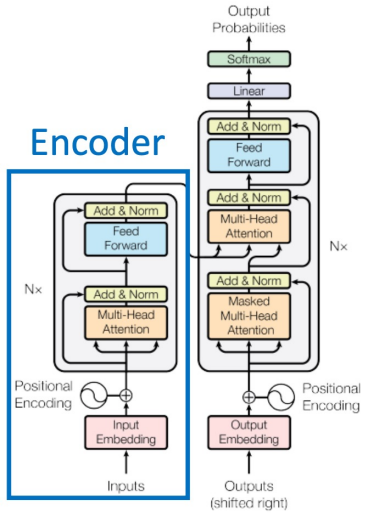


Picture from: Amit Chaudhary – Self-Supervised Representation Learning in NLP

BERT

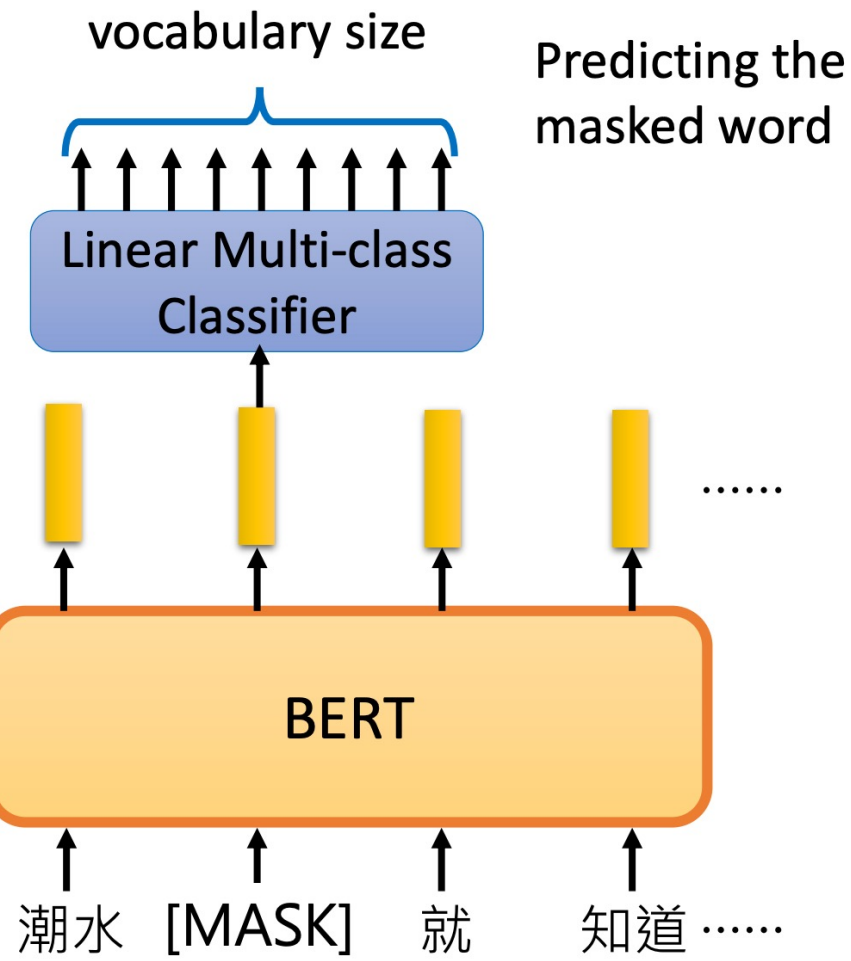
- ***BERT***
  - Devlin(2019) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
- Training data: sentence
- Pretext task: masked LM and next sentence prediction
- BERT = Encoder of Transformer
  - Learning from large amount of text without annotation





BERT

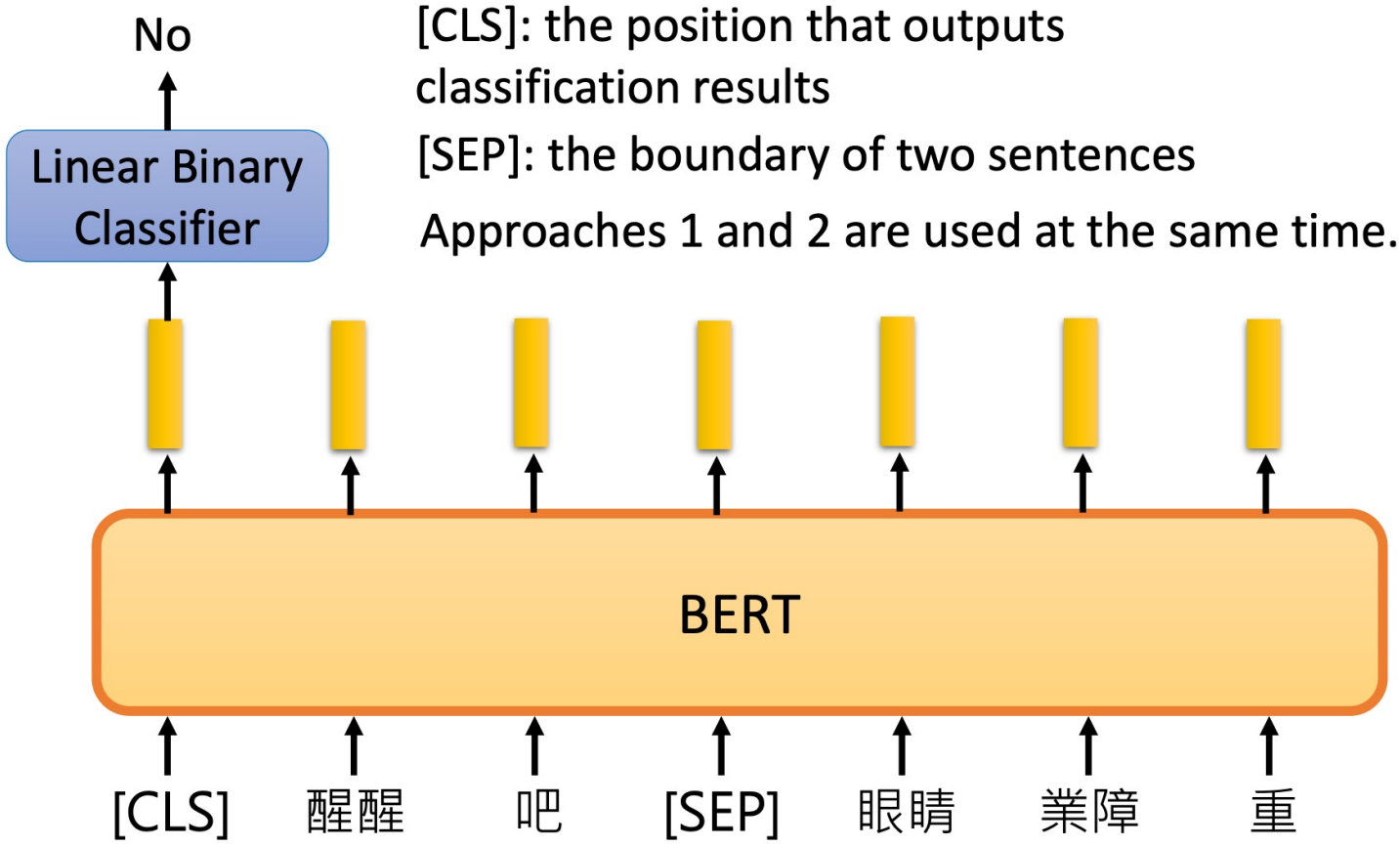
- Pre-Training of BERT
  - Approach 1: Predict the masked word



* Slides from Hungyi Lee's Lecture

BERT

- Pre-Training of BERT
  - Approach 2: Next Sentence Prediction



* Slides from Hungyi Lee's Lecture

GPT-3

- ***GPT-3*** stands for ***Generative Pre-trained Transformer***
  - It was created by OpenAI, and introduced in May 2020
- Transformers are currently the most common model architecture for NLP tasks
  - They employ attention blocks for discovering correlation in text
- GPT-3 generates text based on initial input prompt from the end-user
  - It is trained using next word prediction on huge amount of raw text from the internet
  - The quality of text generated is often undistinguishable from human-written text
  - GPT-3 can also be used for other tasks, such as answering questions, summarizing text, automated code generation, and many others
- It is probably the largest NN model at the present, having 175 billion parameters
  - The cost for training GPT-3 reportedly is $ 12 million
  - For comparison, Microsoft's Turing NLG (Natural Language Generation) model has 17 billion parameters

Thanks!
Q&A