NTU 2020 Spring Machine Learning

Self-Supervised Learning

Chi-Liang Liu, Hung-Yi Lee
NTU

Refer to UCB CS294-158-SP20

Previously on this Course

- Supervised Learning
  - Given: a dataset $\mathcal{D} = \{(\boldsymbol{x}, \boldsymbol{y})_i\}_{i=1}^N$ and
    a loss function $\ell : \hat{\mathcal{Y}} \times \mathcal{Y} \to \mathbb{R}, (\hat{\boldsymbol{y}}, \boldsymbol{y}) \to \ell(\hat{\boldsymbol{y}}, \boldsymbol{y})$

    Goal: $\min_\theta \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}} \left[\ell(f_\theta(\boldsymbol{x}), \boldsymbol{y})\right]$

  - Works well when labeled data is abundant.
  - Learn useful representation with the supervision.

- Problem:
  Can we learn useful representation without the supervision?

Why Self-Supervised Learning?

Labeled Data

Unlabeled Data

Slide: Thang Luong

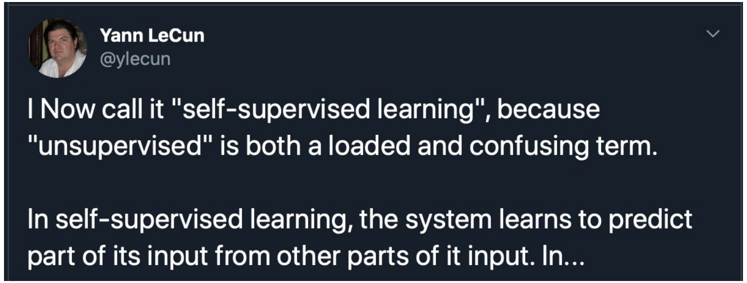
Why Self-Supervised Learning?

- **“Pure” Reinforcement Learning (cherry)**
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**
- **Supervised Learning (icing)**
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits per sample**
- **Self-Supervised Learning (cake génoise)**
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**

Yann LeCun’s cake

What is Self-Supervised Learning?

- A version of unsupervised learning where data provides the supervision.



Yann LeCun
@ylecun

I Now call it "self-supervised learning", because "unsupervised" is both a loaded and confusing term.

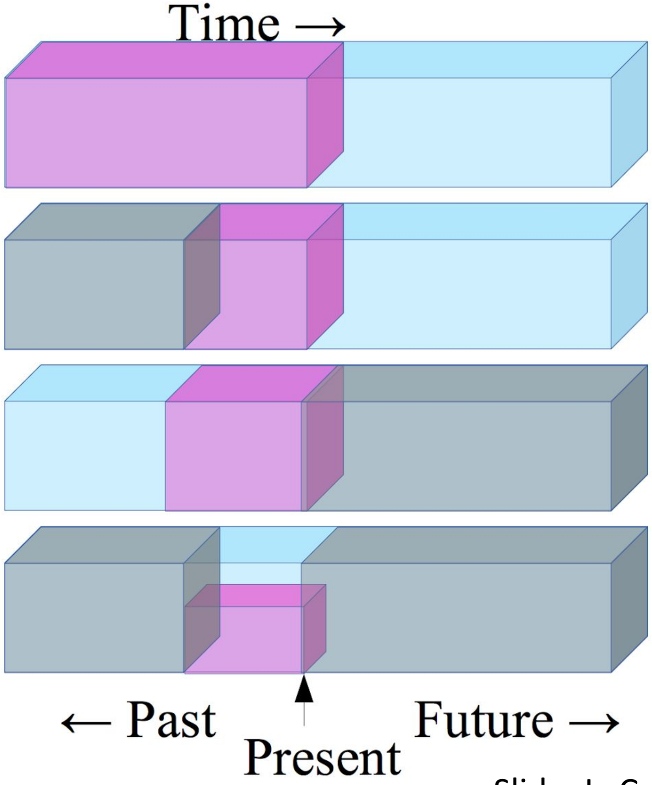
In self-supervised learning, the system learns to predict part of its input from other parts of it input. In...

- In general, withhold some part of the data and the task a neural network to predict it from the remaining parts.

- Goal: Learning to represent the world before learning tasks.

Self-Supervised Learning= Filling the Blanks

- 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.

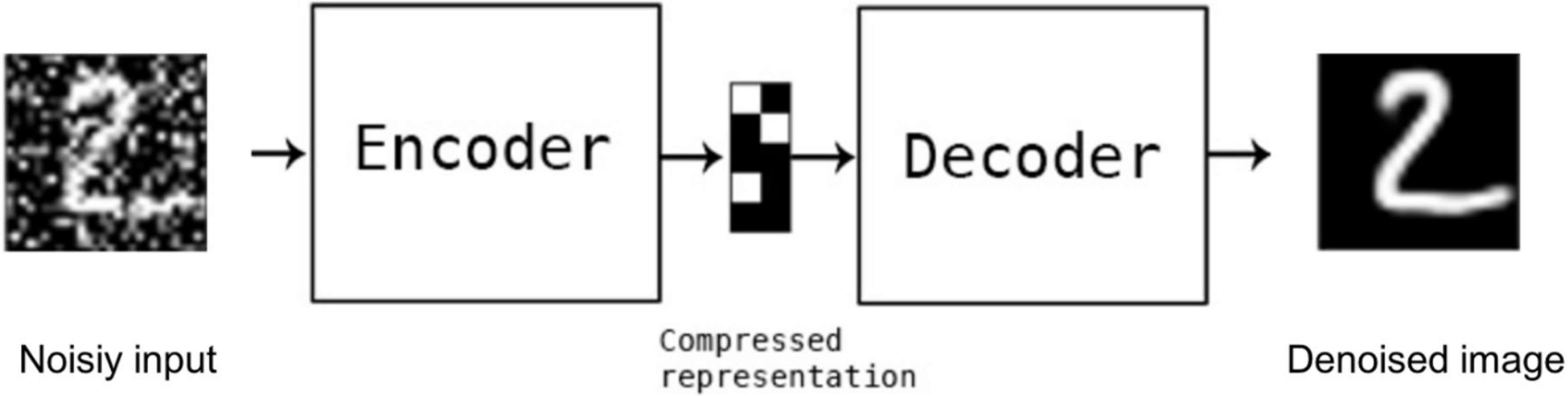


Slide: LeCun

Methods of Self-Supervised Learning

- Reconstruct from a corrupted (or partial) data
  - Denoising Autoencoder
  - Bert-Family (Text)
  - In-painting (Imagae)
- Visual common sense tasks
  - Jigsaw puzzles
  - Rotation
- Contrastive Learning
  - word2vec
  - Contrastive Predictive Coding (CPC)
  - SimCLR

Denoising AutoEncoder



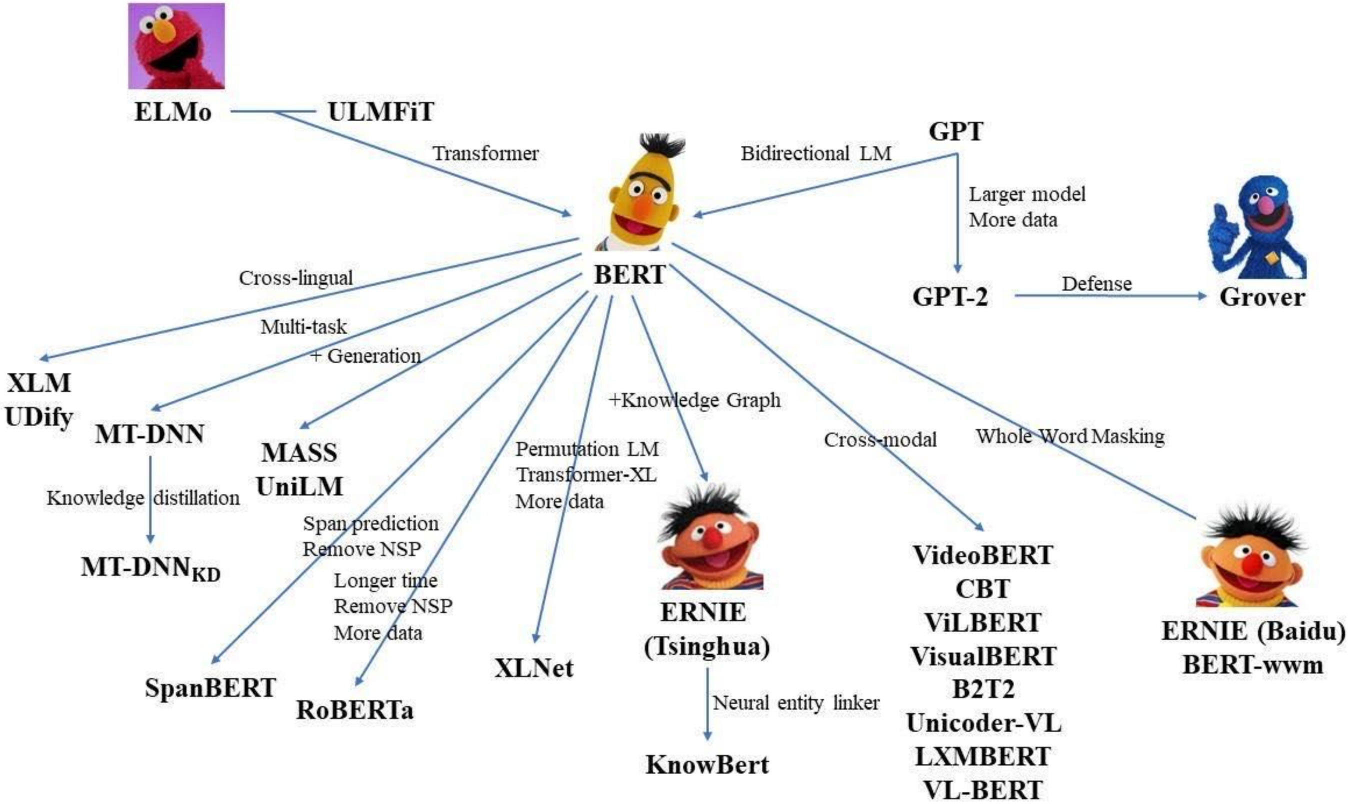
Noisiy input

Denoised image

The feature we want to extract from the image

Slide: CS294-158

BERT-Family


ELMo
ULMFiT
Transformer
BERT
Bidirectional LM
GPT
Larger model
More data
GPT-2
Defense
Grover
Cross-lingual
XLM
UDify
Multi-task
MT-DNN
Knowledge distillation
MT-DNN$_{KD}$
+ Generation
MASS
UniLM
Span prediction
Remove NSP
SpanBERT
Longer time
Remove NSP
More data
RoBERTa
Permutation LM
Transformer-XL
More data
XLNet
+Knowledge Graph
ERNIE
(Tsinghua)
Neural entity linker
KnowBert
Cross-modal
VideoBERT
CBT
ViLBERT
VisualBERT
B2T2
Unicoder-VL
LXMBERT
VL-BERT
Whole Word Masking
ERNIE (Baidu)
BERT-wwm

Language Model

- A statistical **language model** is a probability distribution over sequences of words. Given such a sequence, say of length *m*, it assigns a probability $P(w_1, \ldots, w_m)$ to the whole sequence.
  ex. P("Is it raining now?") > P("Is it raining yesterday?")

- How to Compute?
  - n-gram model

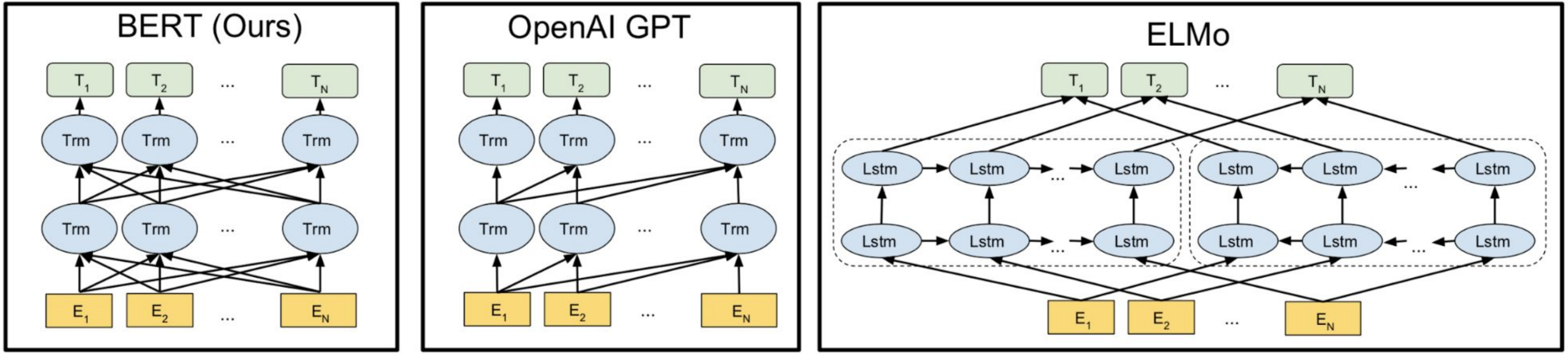
$$P(w_1, \ldots, w_m) = \prod_{i=1}^{m} P(w_i \mid w_1, \ldots, w_{i-1}) \approx \prod_{i=1}^{m} P(w_i \mid w_{i-(n-1)}, \ldots, w_{i-1})$$

  - Neural Network

$$P(w_t | \text{context}) \, \forall t \in V$$

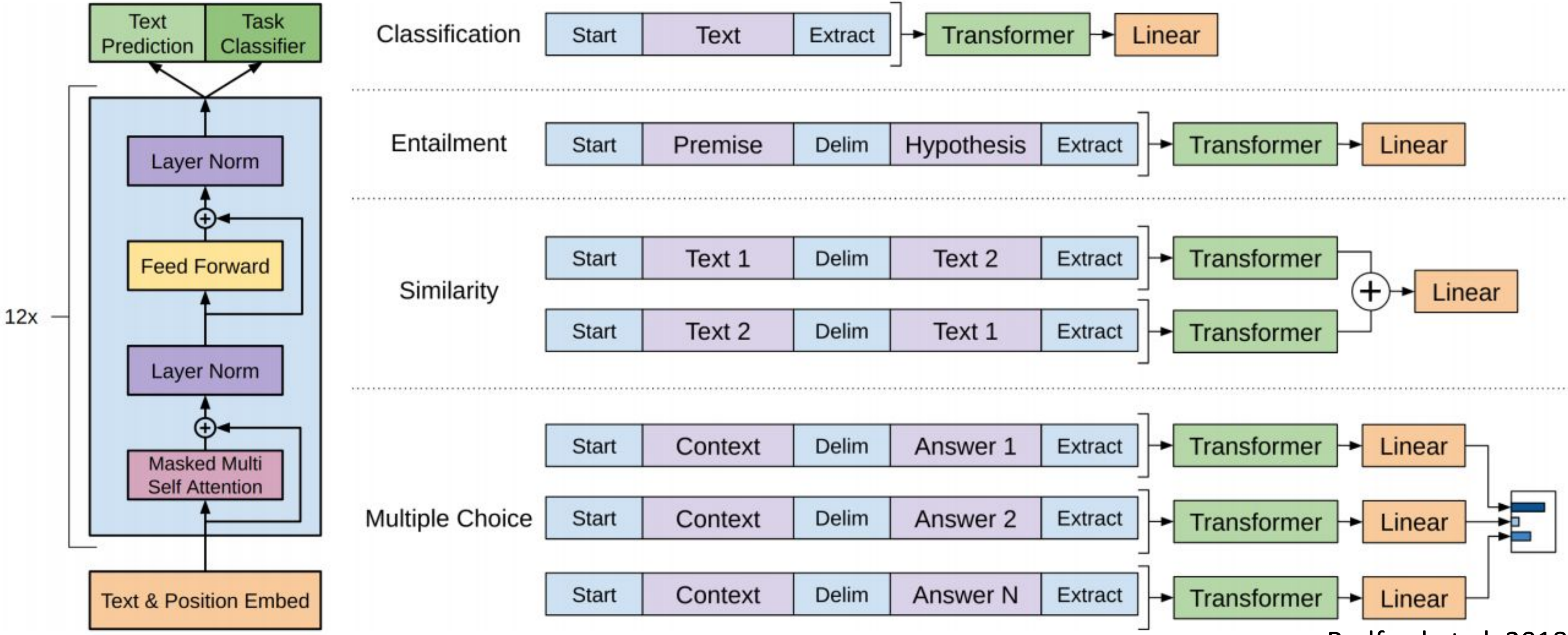
Wiki: Language Model

ELMO & GPT & BERT



Devlin et al. 2018

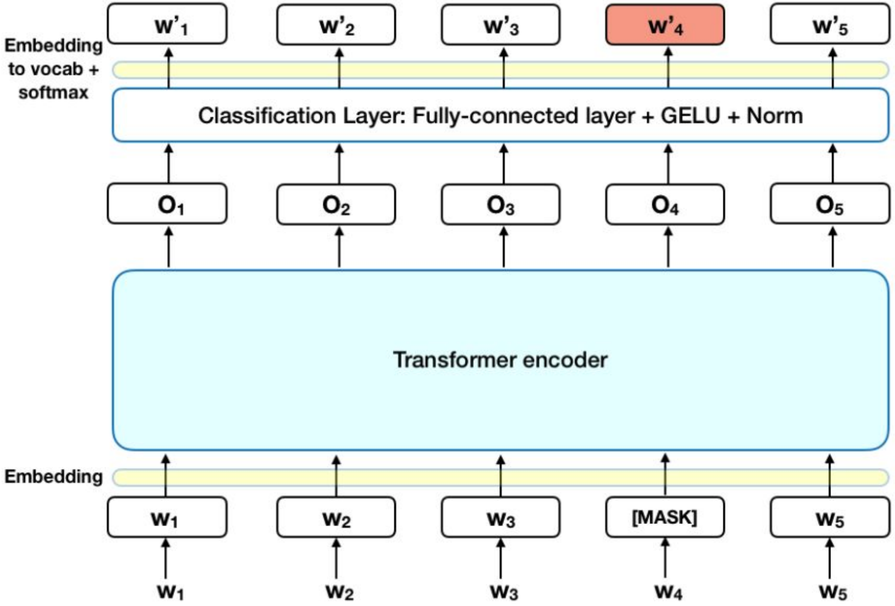
GPT - Language Model



Radford et al. 2018

BERT - Masked LM

Masked LM:

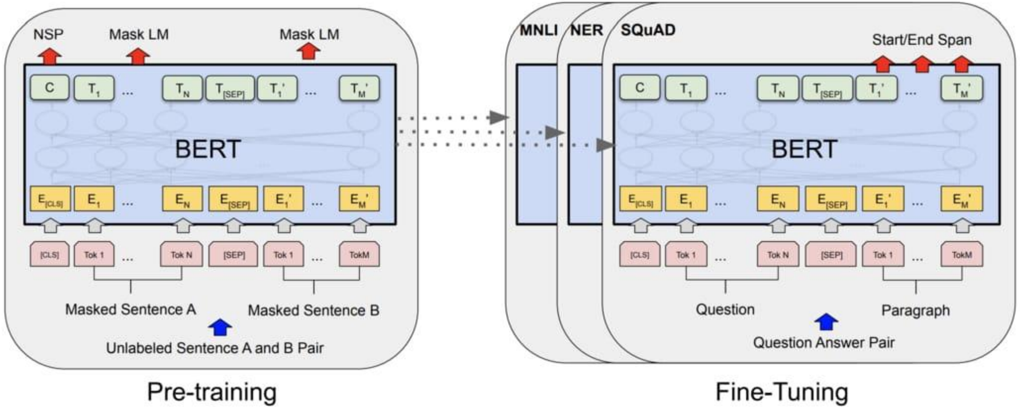

w'₁
w'₂
w'₃
w'₄
w'₅
Embedding to vocab + softmax
Classification Layer: Fully-connected layer + GELU + Norm
O₁
O₂
O₃
O₄
O₅
Transformer encoder
Embedding
w₁
w₂
w₃
[MASK]
w₅
w₁
w₂
w₃
w₄
w₅

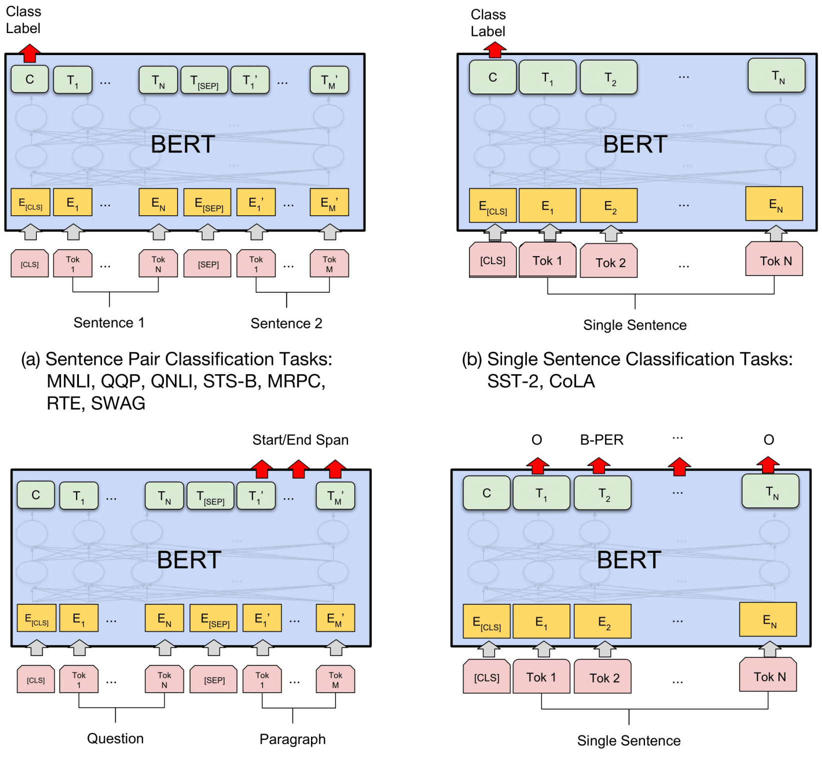
Predict 15% of the tokens in the input.

80% replaced with a masked token
10% replaced with a random word
10% remain the same

ex.
Input: The [mask] sat on the [mask] .
Output: The cat sat on the mat .

BERT - Pipeline





(a) Sentence Pair Classification Tasks:
MNLI, QQP, QNLI, STS-B, MRPC,
RTE, SWAG

(b) Single Sentence Classification Tasks:
SST-2, CoLA

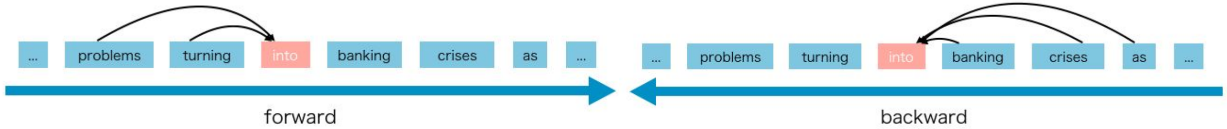
(c) Question Answering Tasks:
SQuAD v1.1

(d) Single Sentence Tagging Tasks:
CoNLL-2003 NER

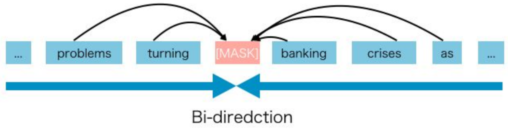
Devlin et al. 2018

ARLM vs AELM

- Autoregressive Language Model (ARLM)
  - Pro: Does not rely on data corruption
  - Con: Can only be forward or backward



- Autoencoding Language Model (AELM a.k.a MaskedLM)
  - Pro: Can use bidirectional information (Context Dependency)
  - Con: Pretrain-Finetune discrepancy (Input Noise), Independence Assumption



Yang et al. 2019

XLNet - Permutation LM

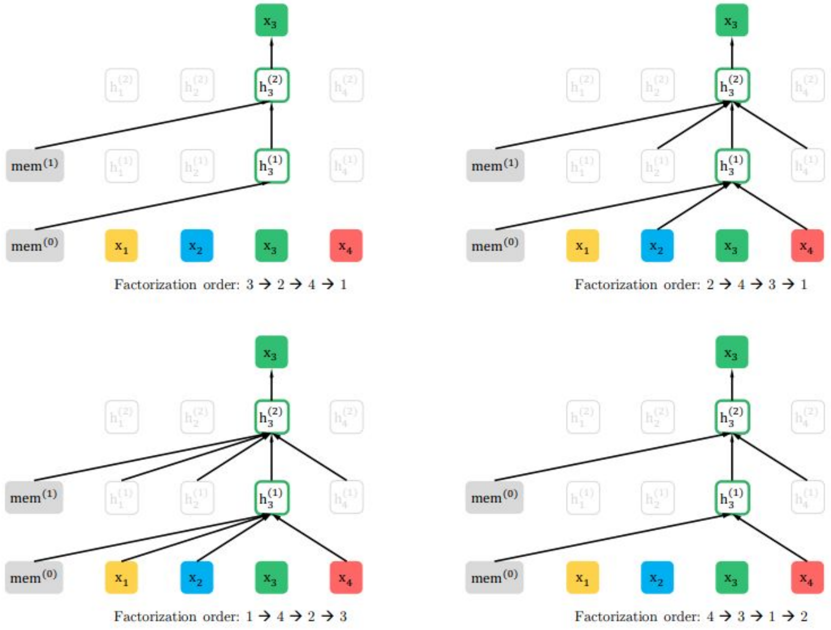
Permutation LM:

Step 1. Permutation
Step 2. Autoregressive

ex.
Given Sequence [x1, x2, x3, x4]

Step 1. [x1, x2, x3, x4] -> [x2, x4, x3, x1]
Step 2. Given [x2] predict [x2, x4]
-> Given [x2, x4] predict [x2, x4, x3]
-> Given [x2, x4, x3] predict [x2, x4, x3, x1]

Yang et al. 2019



Factorization order: 3 → 2 → 4 → 1

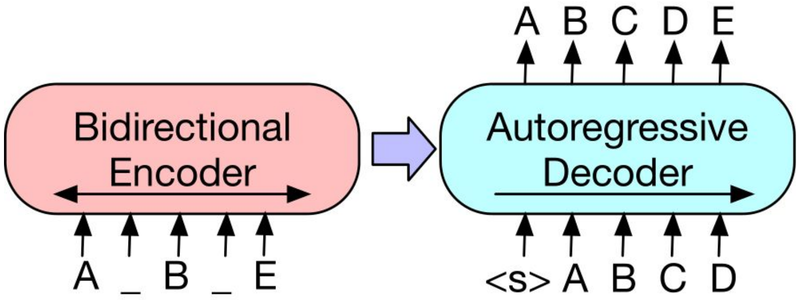
Factorization order: 2 → 4 → 3 → 1

Factorization order: 1 → 4 → 2 → 3

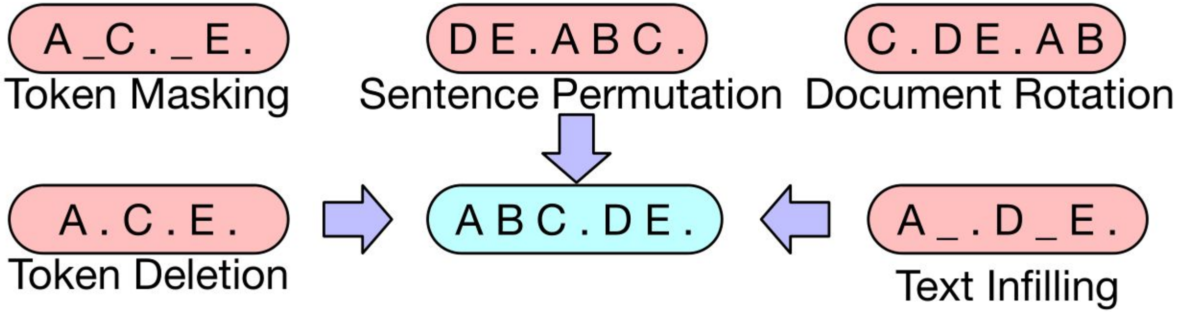
Factorization order: 4 → 3 → 1 → 2

BART - Encoder & Decoder

- Previous: Fixed-length to Fixed-length
- BART:

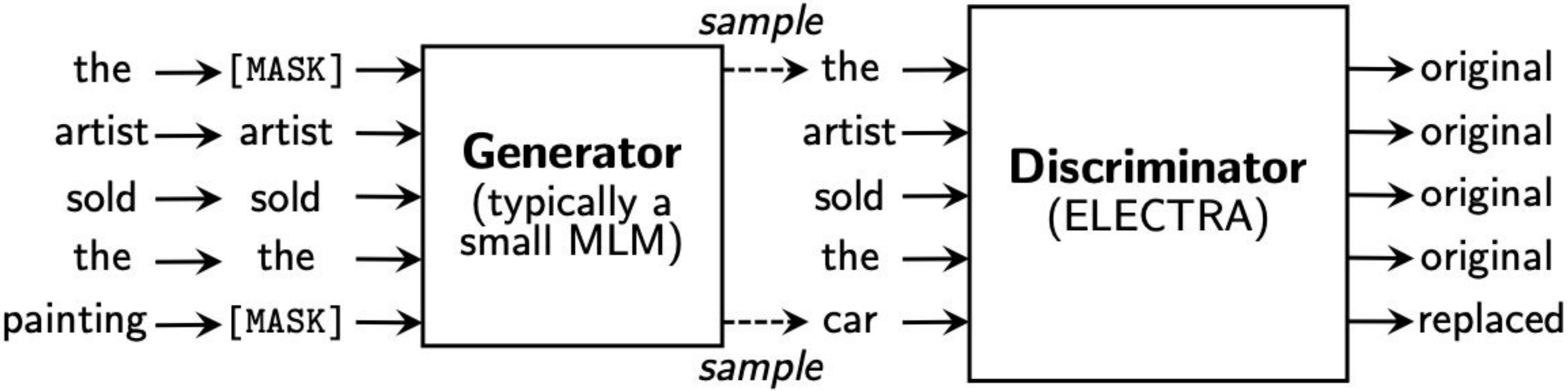


- Tasks:



Lewis et al. 2018

ELECTRA - Discriminator

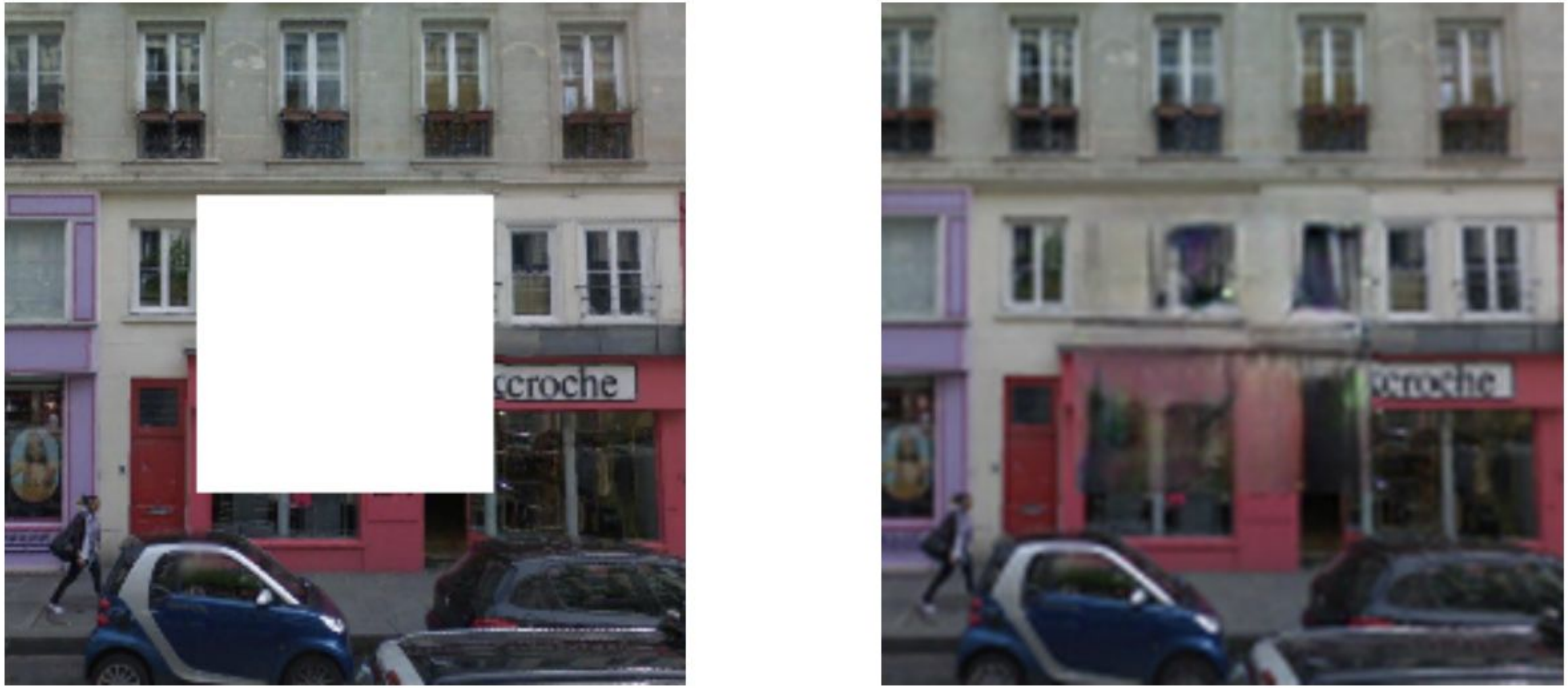


Clark et al. 2018

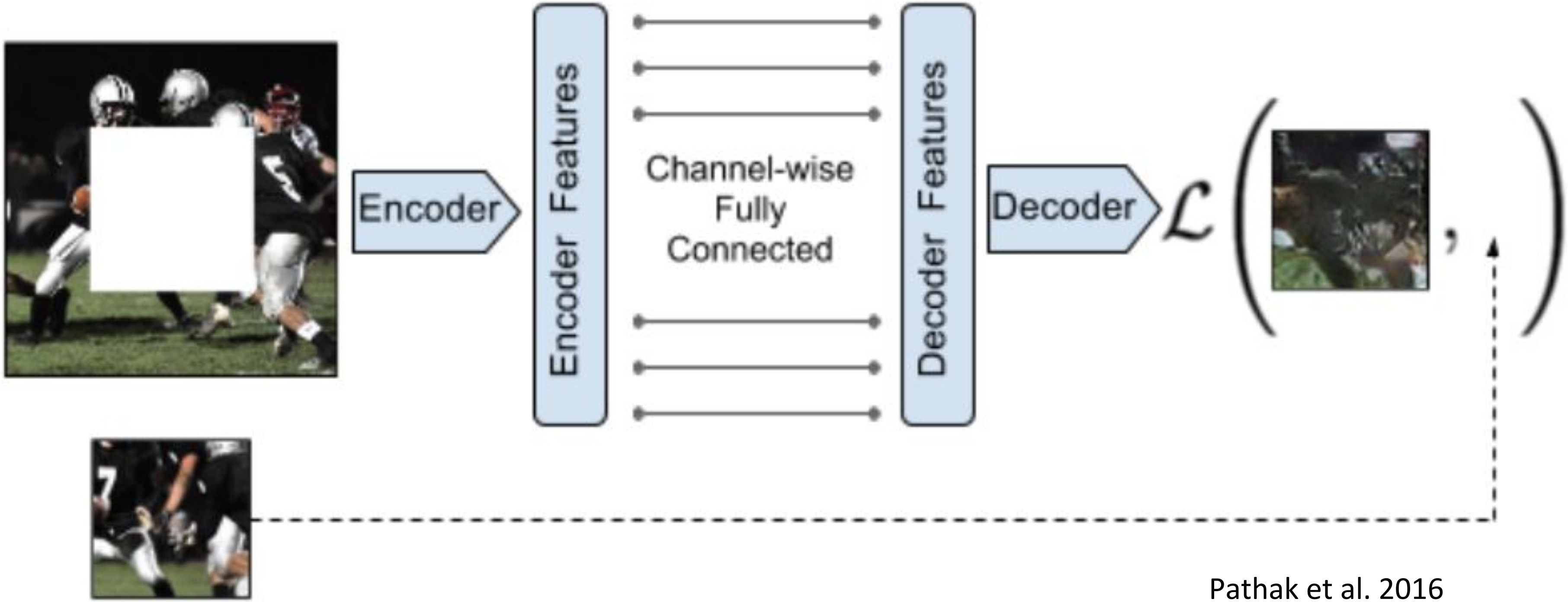
Others

- RoBERTa: A Robustly Optimized BERT Pretraining Approach
- ALBERT: A LITE BERT FOR SELF-SUPERVISED LEARNING OF LANGUAGE REPRESENTATIONS

Predict missing pieces

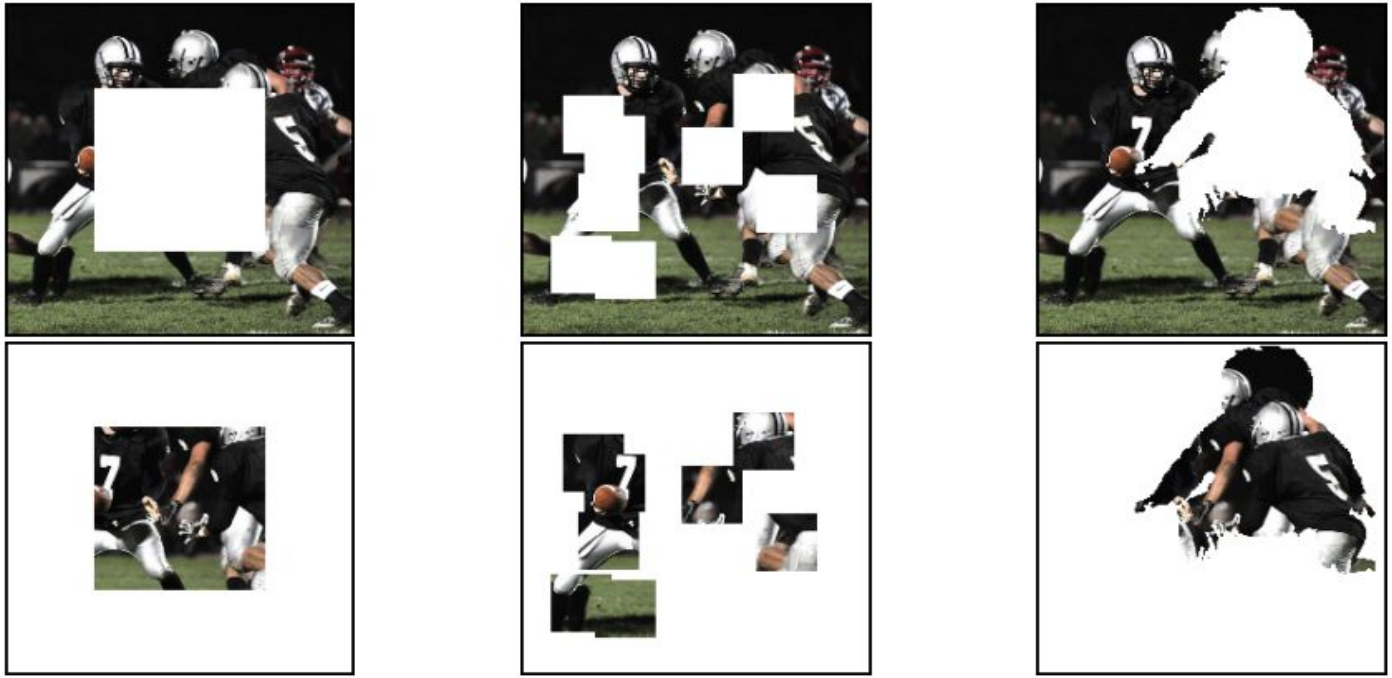


Pathak et al. 2016
Slide: CS294-158



Pathak et al. 2016
Slide: CS294-158

Context Encoder



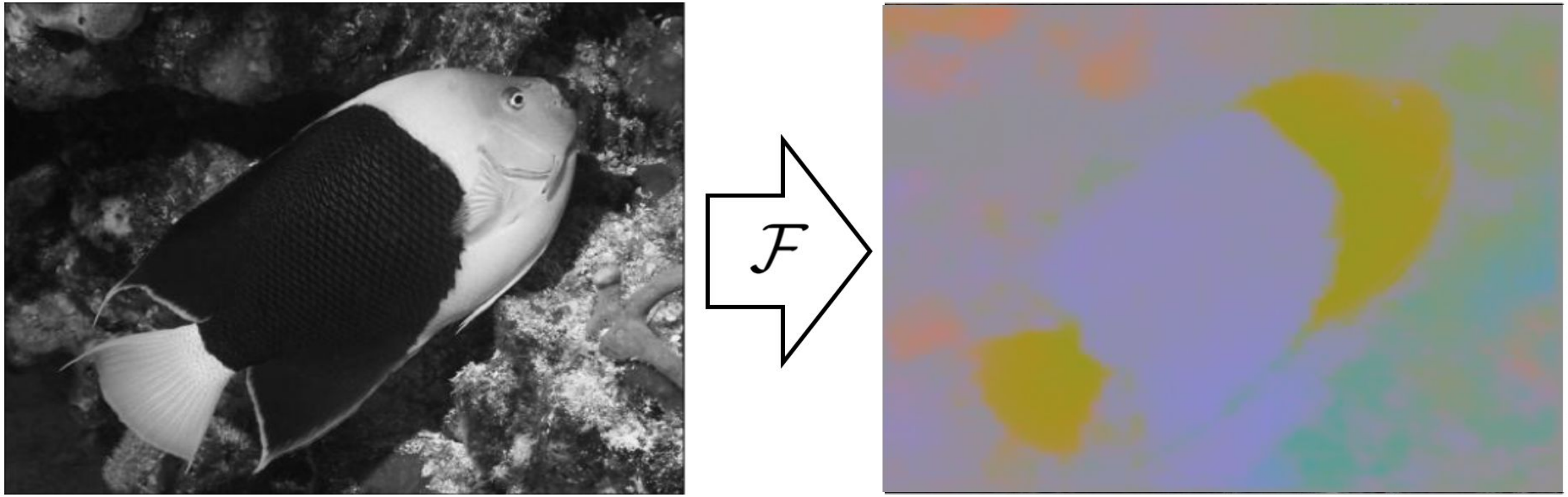
(a) Central region

(b) Random block

(c) Random region

Pathak et al 2016
Slide: CS294-158

Predicting one view from another

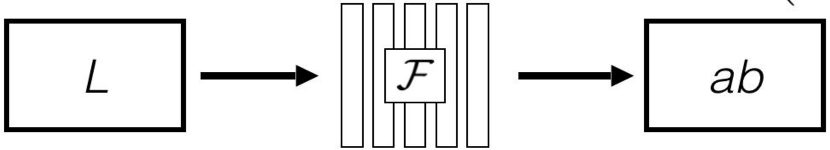


Grayscale image: *L* channel

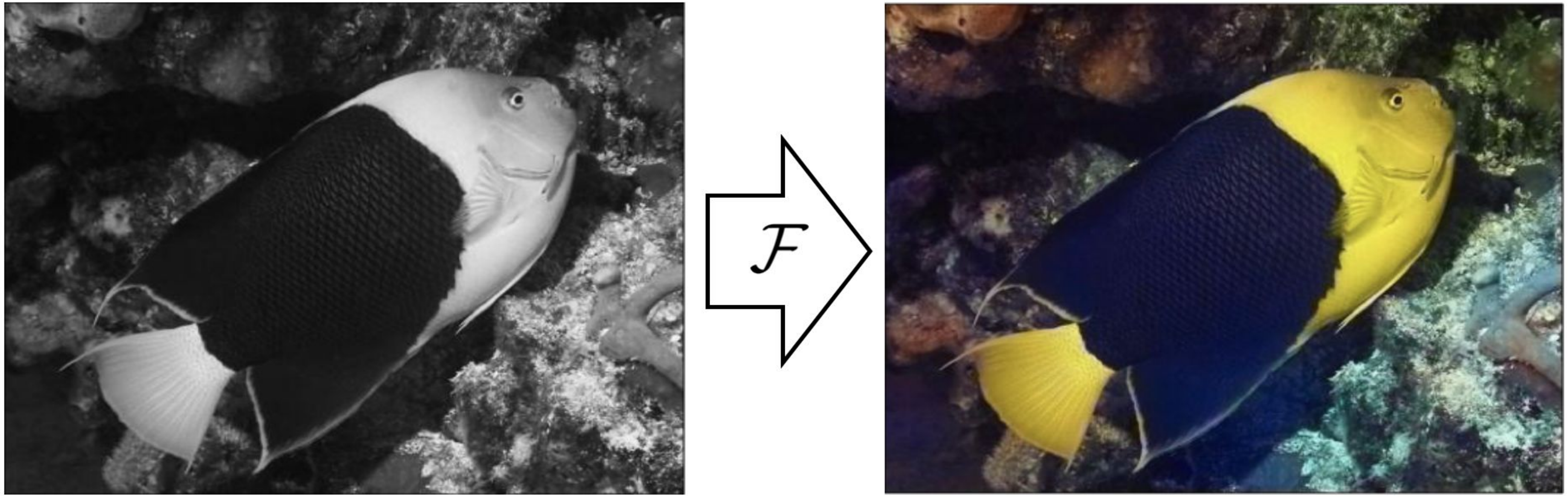
$$\mathbf{X} \in \mathbb{R}^{H\times W\times 1}$$

Concatenate (*L*,*ab*) channels

$$(\mathbf{X}, \widehat{\mathbf{Y}})$$



Slide: Richard Zhang

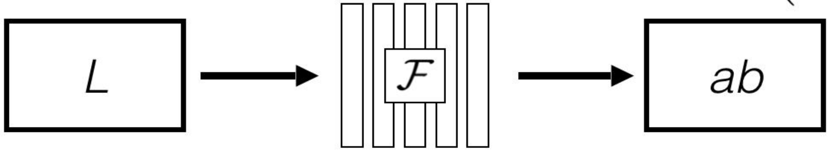


Grayscale image: *L* channel

$$\mathbf{X} \in \mathbb{R}^{H \times W \times 1}$$

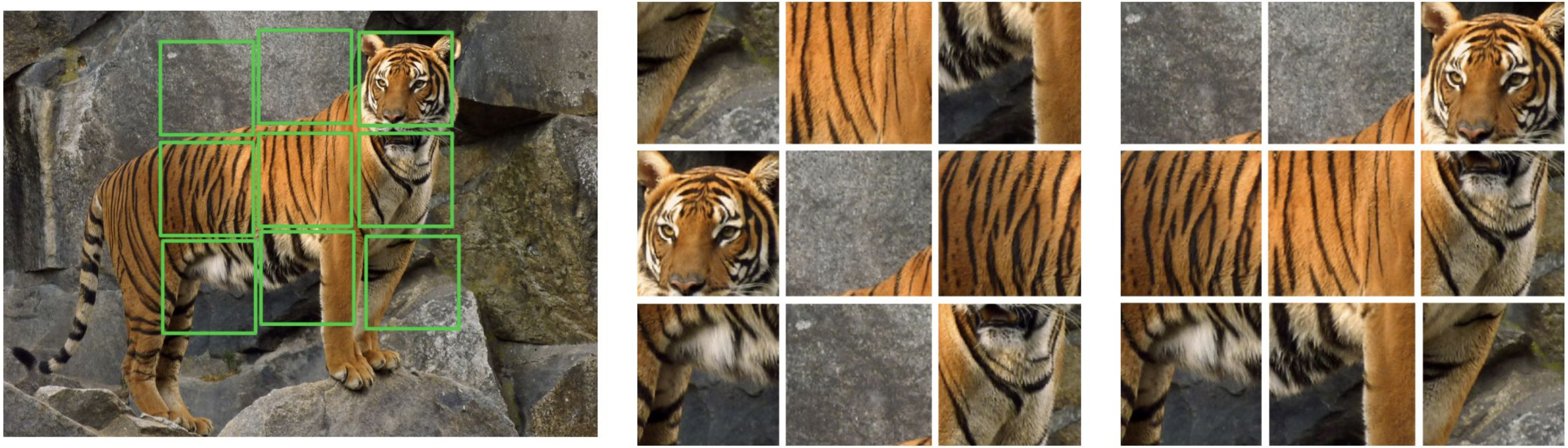
Concatenate (*L*,*ab*) channels

$$(\mathbf{X}, \widehat{\mathbf{Y}})$$



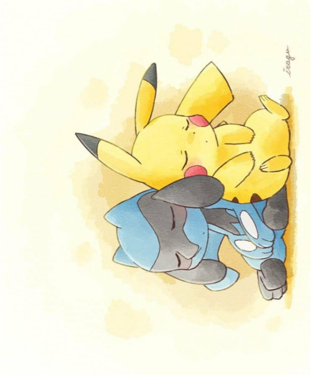
Slide: Richard Zhang

Solving Jigsaw Puzzles

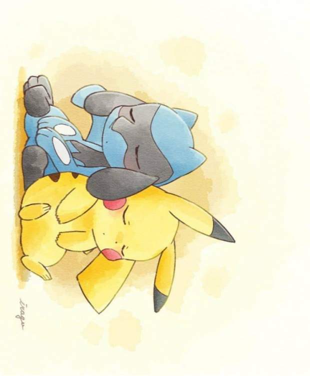


Slide: CS294-158

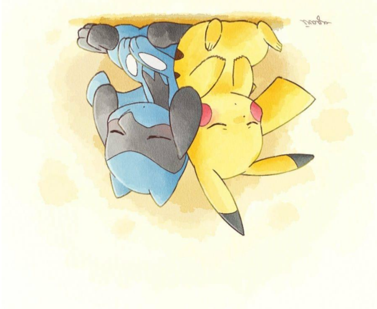
Rotation



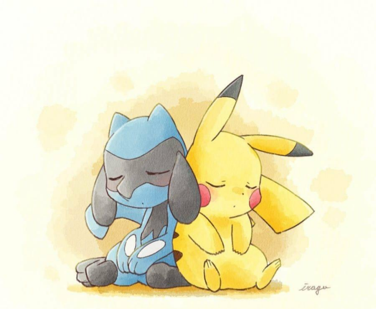
90° rotation



270° rotation

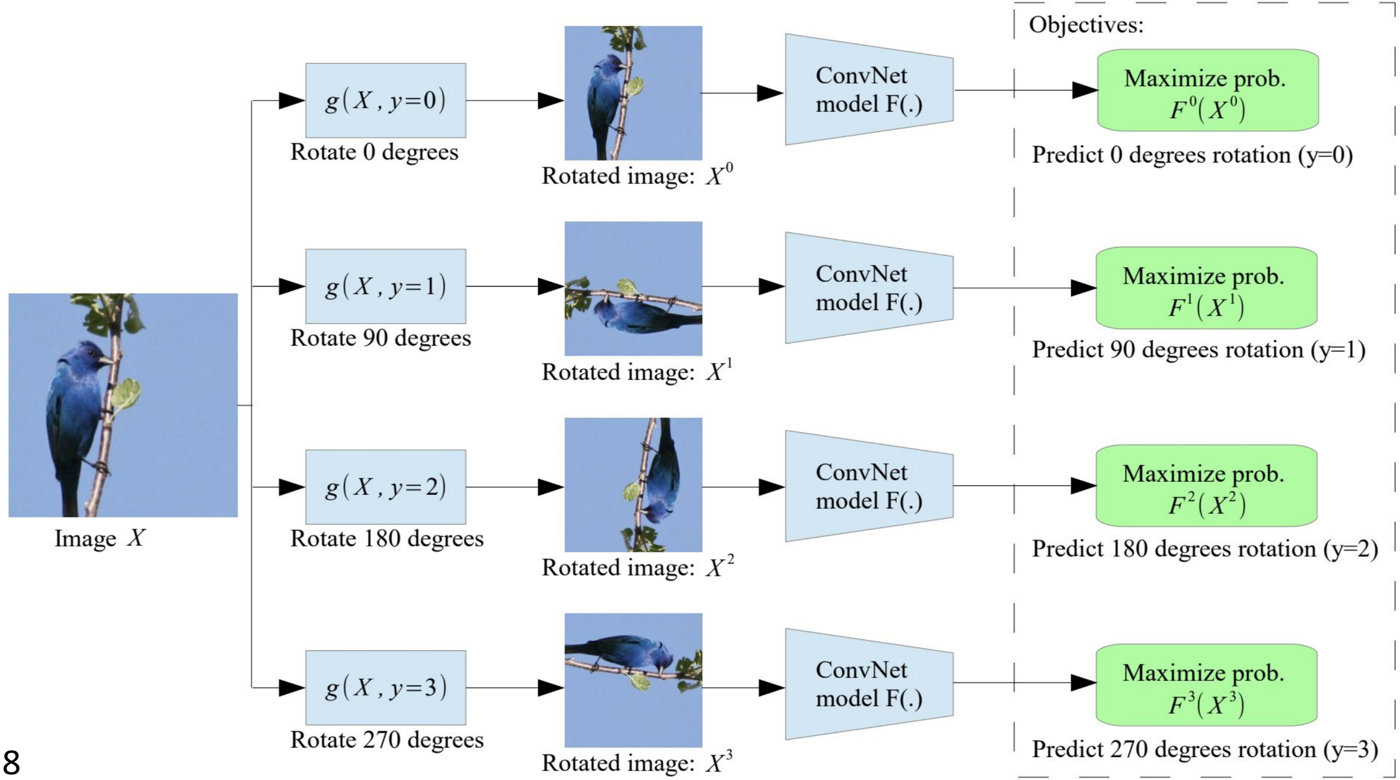


180° rotation



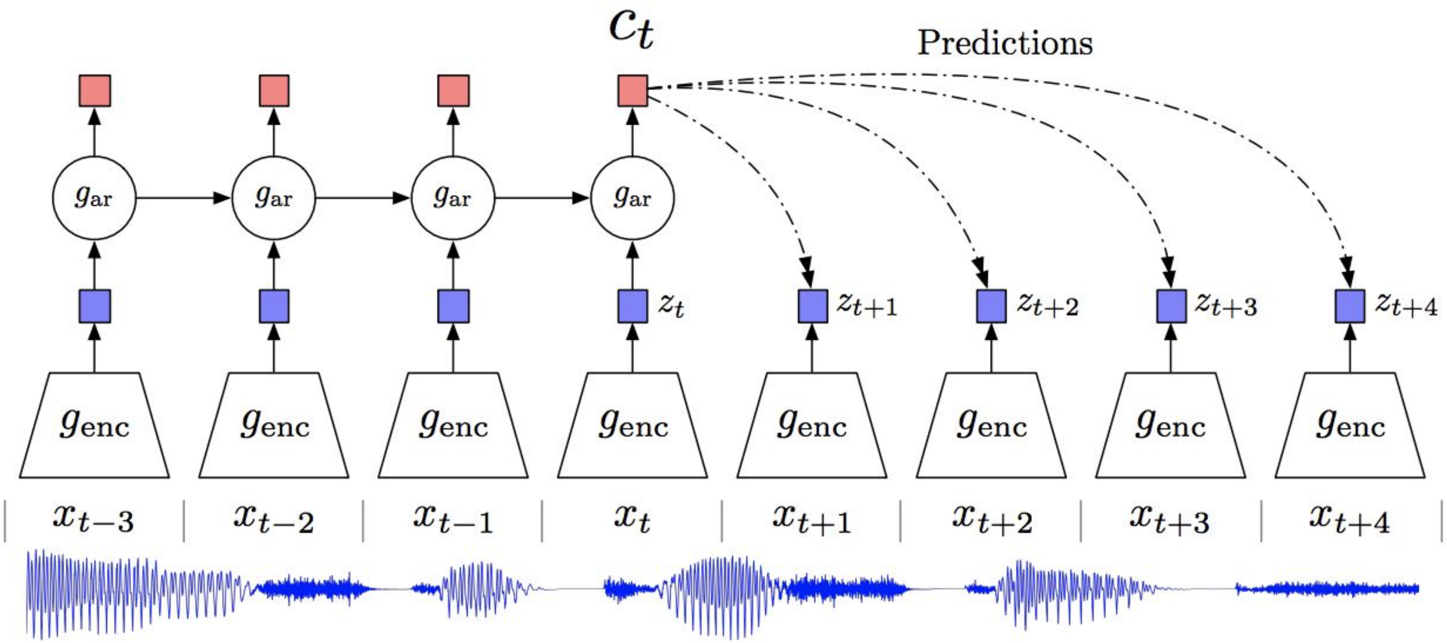
0° rotation

Rotation



Slide: CS294-158

Contrastive Predictive Coding

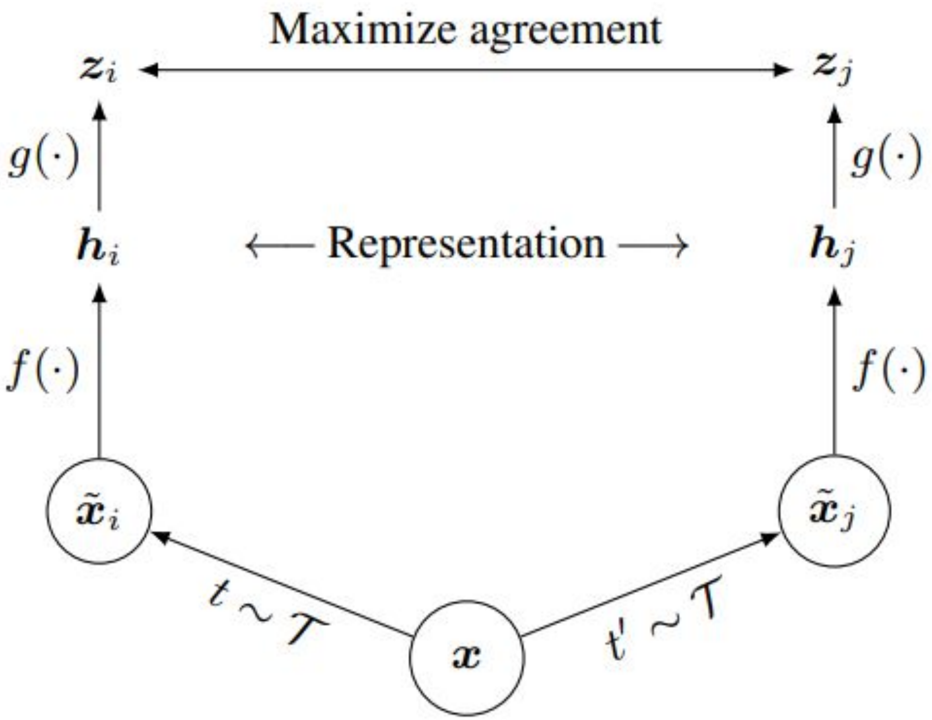


Remember Word2Vec? They are almost the same idea.

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

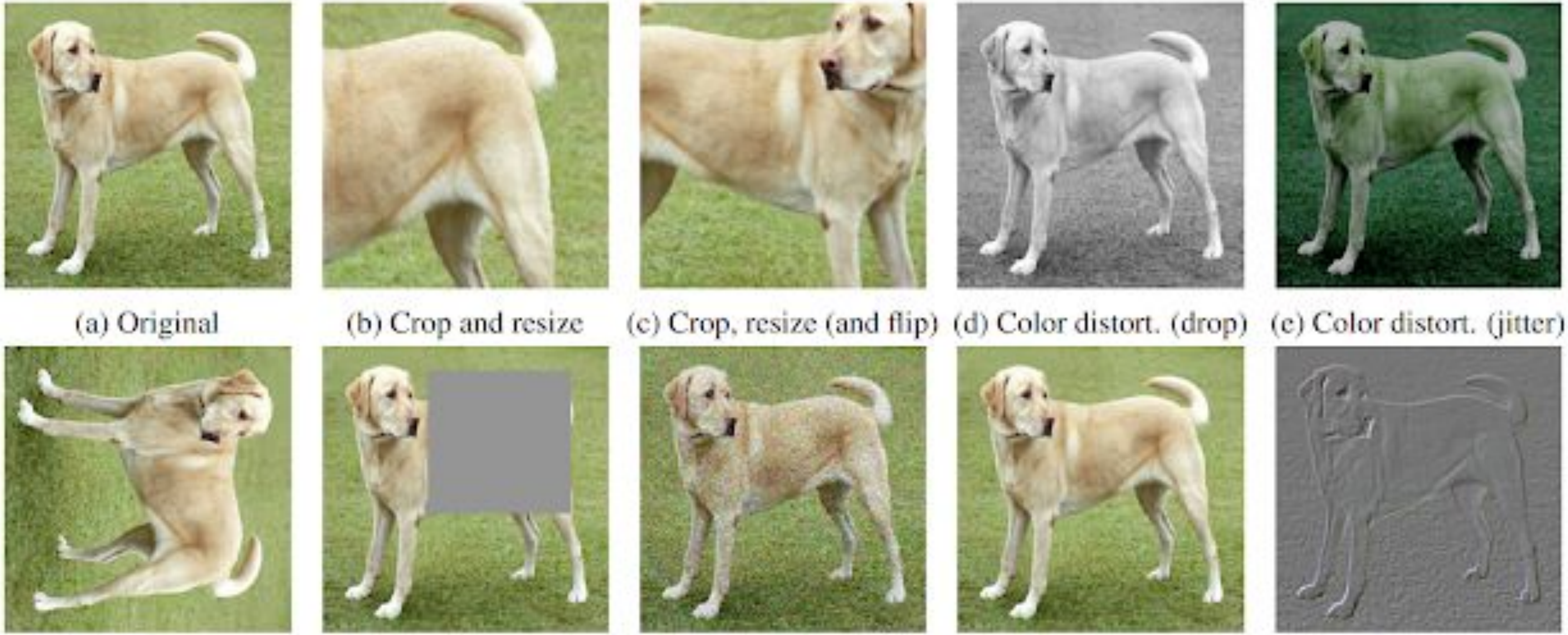
van den Oord et al. 2020

SimCLR



Chen et al. 2020

SimCLR



(a) Original

(b) Crop and resize

(c) Crop, resize (and flip)

(d) Color distort. (drop)

(e) Color distort. (jitter)

(f) Rotate $\{90^\circ, 180^\circ, 270^\circ\}$

(g) Cutout

(h) Gaussian noise

(i) Gaussian blur

(j) Sobel filtering

Reference

- CS294-158 Deep Unsupervised Learning Lecture 7
- AAAI 2020 Keynotes Turing Award Winners Event
- Learning From Text - OpenAI
- Learning from Unlabeled Data - Thang Luong