Self-supervised Learning Generative or Contrastive （Outline)

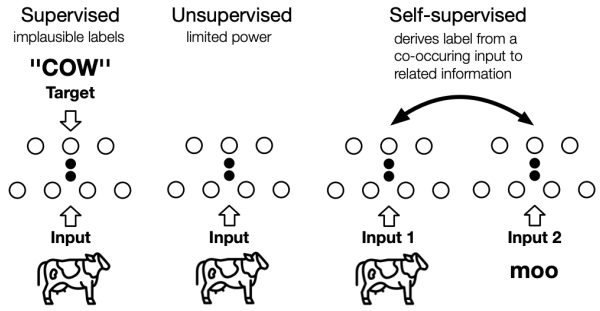
Xiao Liu, Fanjin Zhang, Zhenyu Hou, Zhaoyu Wang, Li Mian, Jing Zhang, Jie Tang*
Slide Maintain: Xiao Liu, Haoyun Hong
Knowledge Engineering Group, Tsinghua University
arXiv: https://arxiv.org/abs/2006.08218
AMiner: https://www.aminer.cn/pub/5ee8986f91e011e66831c59b/

Index

(in the rest of slides, we use ``SSL'' for ``self-supervised learning'')

1. Introduction

What is self-supervised learning



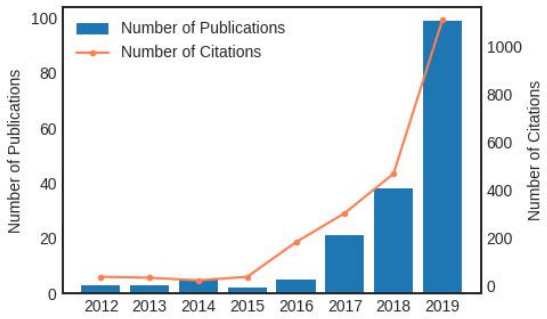
An illustration to distinguish the supervised, unsupervised and self-supervised learning framework. In self-supervised learning, the ``related information'' could be another modality, parts of inputs, or another form of the inputs.

What is self-supervised learning

- Jitendra Malik: "Supervision is the opium of the AI researcher"
- Alyosha Efros: "The AI revolution will not be supervised"
- Yann LeCun: "self-supervised learning is the cake, supervised learning is the icing on the cake, reinforcement learning is the cherry on the cake"

5

Soar in the SSL



Number of publications and citations on self-supervised learning from 2012-2019. Self-supervised learning is gaining huge attention in recent years. Data from Microsoft Academic Graph. This only includes paper containing the complete keyword ``self-supervised learning'', so the real number should be even larger.

6

2. Generative SSL

Generative SSL category

1. Auto-regressive (AR) model
2. Flow-based model
3. Auto-encoding (AE) model
4. Hybrid Generative models

Auto-regressive (AR) model

- Auto-regressive (AR) models can be viewed as “Bayes net structure” (directed graph model). The joint distribution can be factorized as a product of conditionals where the probability of each variable is dependent on the previous variables.
  - GPT / GPT-2
  - PixelCNN / PixelRNN
  - GraphRNN / MRNN / GCPN

$$\max_{\theta} p_\theta(\mathbf{x}) = \sum_{t=1}^{T} \log p_\theta(x_t | \mathbf{x}_{1:t-1})$$

9

GPT/GPT-2 (NLP)

- Improving language understanding by generative pre-training. (2018, Radford et al.)
  - Maximizing the likelihood under the forward autoregressive factorization
  - Use Transformer decoder architecture for language modeling

$$p(output|input, task)$$

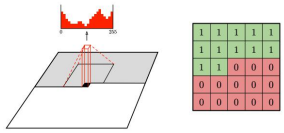
    - a translation training example can be written as the sequence
      - (“translate to French”, english text, french text).
    - a reading comprehension training example can be written as
      - (“answer the question”, document, question, answer).
  - Very good at text generation because it is a single-direction model

10

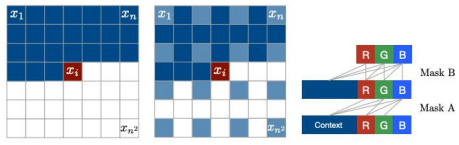
PixelCNN / PixelRNN (CV)

- Pixel recurrent neural networks. (ICML 2016, Oord et al.)
- Conditional image generation with pixelcnn decoders. (NIPS 2016, Oord et al.)
  - use auto-regressive methods to model images pixel by pixel

PixelCNN



PixelRNN



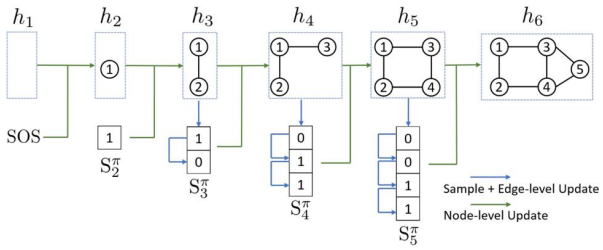
Context Multi-scale context

To generate a pixel in the multi-scale case we can also condition on the subsampled image pixels (in light blue).

11

GraphRNN (Graph)

- Graphrnn: Generating realistic graphs with deep auto-regressive models. (ICML 2018, You et al.)
  - decompose the graph generation process into a sequence generation of nodes and edges conditioned on the graph generated so far.



12

MRNN / GCPN (Graph)

- Molecularrnn: Generating realistic molecular graphs with optimized properties. (2019, Popova et al.)
- Graph convolutional policy network for goal-directed molecular graph generation. (NIPS 2018, You et al.)
  - use a reinforcement learning framework to generate molecule graphs through optimizing domain-specific rewards.
  - MRNN mainly uses RNN-based networks
  - GCPN employs GCN-based encoder networks

13

Flow-based Model

- directly formalizing high dimensional densities *p(x)* is difficult.
- we hope to generate it “step by step” by stacking a series of invertible transforming functions that describing different data characteristics respectively.
- Advantage: mapping between x and z is invertible
  - NICE / RealNVP / Glow

- Change of variable $x \rightarrow f(x)$, where f is invertible

$$p_\theta(x) = p(f_\theta(x))\left|\frac{\partial f_\theta(x)}{\partial x}\right|$$

$$\max_\theta \sum_i \log p_\theta(x^{(i)}) = \max_\theta \sum_i \log p_Z(f_\theta(x^{(i)})) + \log\left|\frac{\partial f_\theta}{\partial x}(x^{(i)})\right|$$

14

NICE / RealNVP / Glow (CV)

- Nice: Non-linear independent components estimation. (2014, Dinh et al.)
- Density estimation using real nvp (2016, Dinh et al.)
- Glow: Generative flow with invertible 1x1 convolutions. (NIPS 2018, Kingma et al.)
  - model learns disentangled transformation functions

15

Auto-encoding (AE) Model

- reconstruct (part of) inputs from (corrupted) inputs
- probably the most popular generative model due to its flexibility
  - Basic AE Model
  - Context Prediction Model (CPM)
  - Denoising AE Model (DAE)
  - Variational AE Model (VAE)

16

Basic AE Model

- first introduced for pre-training artificial neural networks
- Restricted Boltzmann Machine: a special AE
  - Undirected graphical model
  - Two layers: visible layer and hidden layer
  - minimize the difference between the marginal distribution of models and data distributions
- Basic AE
  - Could be view as ``directed graphical model''
  - Encoder and Decoder
  - linear autoencoder corresponds to the PCA method

17

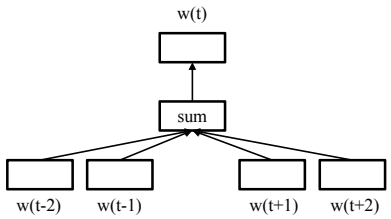
Context Prediction Model (CPM)

- predict contextual information based on inputs
  - Word2Vec (CBOW or SkipGram)
  - DeepWalk-based graph representation

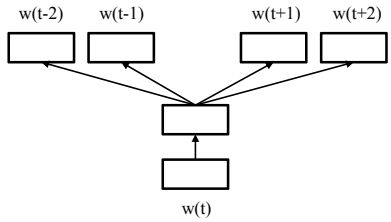
18

Word2vec: CBOW / SkipGram (NLP)

- Distributed representations of words and phrases and their compositionality (NIPS 2013, Mikolov et al.)
- CBOW
  - predict the input tokens based on context tokens
- SkipGram
  - predict context tokens based on input tokens



CBOW



SkipGram

19

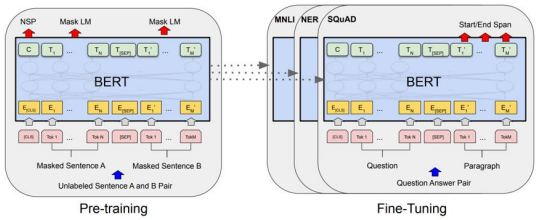
Deepwalk / LINE (Graph)

- Deepwalk: Online learning of social representations. (KDD 2014, Perozzi et al.)
  - samples truncated random walks to learn latent node embedding based on the Skip-Gram
  - treats random walks as the equivalent of sentences
- Line: Large-scale information network embedding (WWW 2015, Tang et al.)
  - generate neighbors based on current nodes

20

Denoising AE: BERT (NLP)

- Intuition: representation should be robust to the introduction of noise
  - Masked Language Model (MLM)
- BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (NAACL 2019, Devlin et al.)
  - [MASK]: a unique token introduced in the training process to mask some tokens
  - Predict masked tokens based on their context information,
  - Pre-train and fine-tune

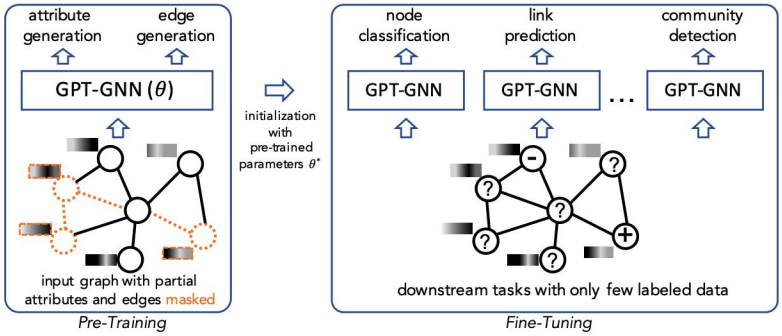


SpanBERT / ERNIE (NLP)

- Spanbert: Improving pre-training by representing and predicting spans. (TACL 2020, Joshi et al.)
  - mask continuous random spans rather than random tokens
  - trains the span boundary representations to predict the masked spans
- Ernie: Enhanced representation through knowledge integration (2019, Sun et al.)
  - masks entities or phrases to learn entity-level and phrase-level knowledge
- Ernie: Enhanced language representation with informative entities. (2019, Zhang et al.)
  - integrates knowledge (entities and relations) in knowledge graphs into language models

GPT-GNN (Graph)

- GPT-GNN: Generative Pre-Training of Graph Neural Networks (KDD 2020, Hu et al.)



Variational AE

- Assumes that data are generated from underlying latent (unobserved) representation
- Posterior distribution
  - Unobserved variables $Z = \{z_1, z_2, ..., z_n\}$
  - Estimation: $p(z|x) \approx q(z|x)$
- Evidence Lower Bound (ELBO)

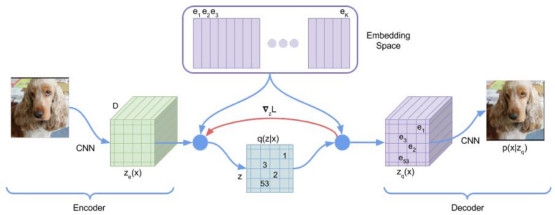
$$\log p(x) \geq -D_{KL}(q(z|x)||p(z)) + \mathbb{E}_{\sim q(z|x)}[\log p(x|z)]$$

  - VAE
  - VQ-VAE / VQ-VAE 2
  - VGAE / DVNE / vGraph

VAE / VQ-VAE (CV)

- Auto-encoding variational bayes. (2013, Kingma et al.)
  - Assume continuous latent variables
  - Assume latent distribution conforms to normal distribution
- Neural discrete representation learning (NIPS 2017, Oord et al.)
  - Learn discrete latent variables: many modalities are inherently discrete
  - Vector Quantization (VQ): the discrete latent variables are calculated by the nearest neighbor lookup using a shared, learnable embedding table

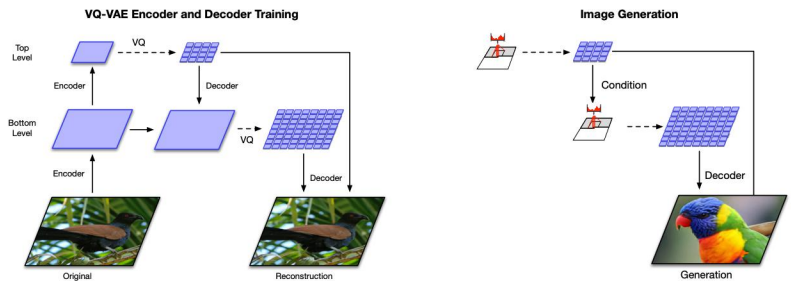
$$\mathcal{L}(x, D(e)) = \|x - D(e)\|_2^2 + \|sg[E(x)] - e\|_2^2 + \beta\|sg[e] - E(x)\|_2^2$$



25

VQ-VAE 2 (CV)

- Generating diverse high-fidelity images with vq-vae-2 (NIPS 2019, Razavi et al.)
  - Using a powerful PixelCNN prior (rather than normal distribution)
  - multi-scale hierarchical organization for local and global information



26

VGAE / DVNE / vGraph (Graph)

- VGAE: Variational graph auto-encoders (2016, Kipf et al.)
  - Using the same variational inference technique as VAE with graph convolutional networks (GCN) as the encoder
  - The objective is to reconstruct the adjacency matrix of the graph by measuring node proximity
- DVNE: Deep variational networkembedding in wasserstein space. (SIGKDD,2016, Kipf et al.)
  - Learning Gaussian node embedding to model the uncertainty of nodes
  - Using 2-Wasserstein distance to measure the similarity between the distributions for its effectiveness in preserving network transitivity
- vGraph: Deep variational networkembedding in wasserstein space. (SIGKDD,2016, Kipf et al.)
  - Performing node representation learning and community detection collaboratively through a generative variational inference framework

27

Hybrid Generative Models

- Combine the advantages of different models
  - Combining AR and AE Model
  - Combining AE and Flow-based Models

28

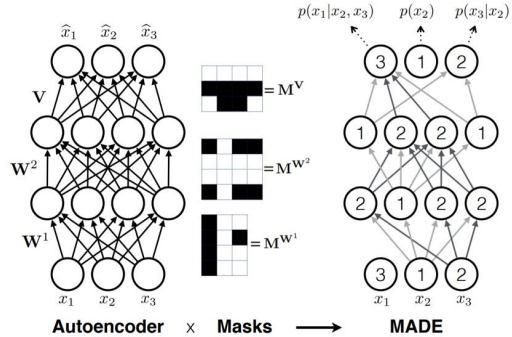
Combining AR and AE Model

- Intuition: combine the advantages of both AR and AE
  - MADE
  - XLNet

29

MADE (CV)

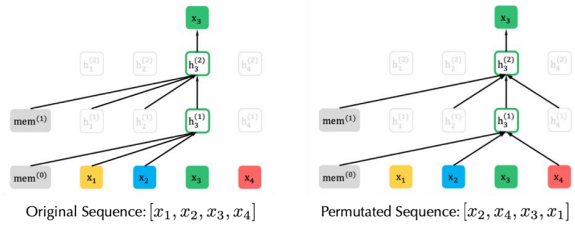
- MADE: Masked autoencoder for distribution estimation (2015, Mathieu et al.)
  - Masking the autoencoder's parameters to respect auto-regressive constraints
  - Can be easily parallelized on conditional computations, and can get direct and cheap estimates of high-dimensional joint probabilities



30

XLNet (NLP)

- XLNet: Generalized autoregressive pretraining for language understanding (NIPS 2019, Yang et al.)
  - PLM: Permutation Language Model
    - Based-on AR model
    - learning bidirectional contexts by permutation
    - reparameterization and a special two-stream self-attention for target-aware prediction
  - Transformer-XL: Extra Long Transformer
    - segment recurrence mechanism + relative encoding scheme



31

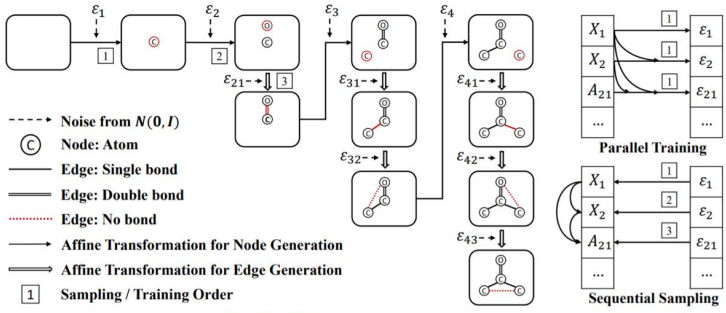
Combining AE and Flow-based Models

- Intuition: combine the advantages of both AE and Flow-based models
  - GraphAF

32

GraphAF (Graph)

- GraphAF: a flow-based autoregressive model for molecular graph generation (2020, Shi et al.)
  - a flow-based autoregressive model for the molecule graph generation
  - incorporates detailed domain knowledge into the reward design
  - defines an invertible transformation from a base distribution (e.g., multivariate Gaussian) to a molecular graph structure



(a) Sampling Phases (b) Framework

33

3. Contrastive Self-supervised Learning

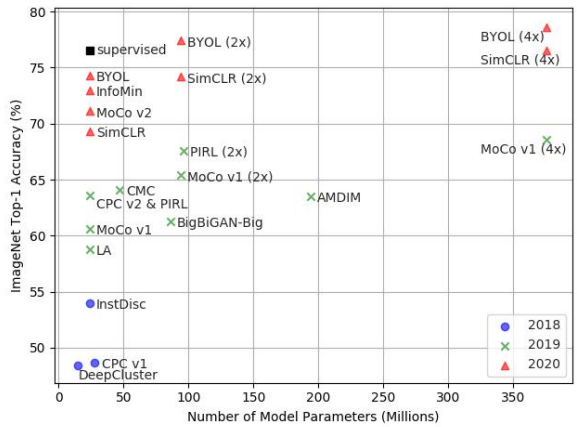
34

Contrastive Self-supervised Learning

1. Context-Instance Contrast
2. Context-Context Contrast
3. Self-supervised Contrastive Pre-training for Semi-supervised Self-training

35

Boost in contrastive pre-training



36

Context-Instance Contrast

- The context-instance contrast, or so-called *global-local* contrast, focuses on modeling the belonging relationship between the local feature of a sample and its global context representation.
  - Predict Relative Position (PRP)
  - Maximize Mutual Information (MI)

37

Predict Relative Position

- Focuses on learning relative positions between local components.
- The global context serves as an implicit requirement for predicting these relations
  - RotNet / Jigsaw / PIRL
  - ALBERT

38

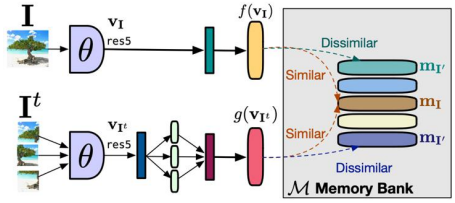
RotNET / Jigsaw / PIRL (CV)

Predict Relative Position

Rotation

Jigsaw

- Self-Supervised Learning of Pretext-Invariant Representations (CVPR 2019, Misra et al.)
  - PIRL for short
  - Use Jigsaw pretext task



39

ALBERT (NLP)

- ALBERT: A lite bert for self-supervised learning of language representations (2019, Lan et al.)
  - proposes Sentence Order Prediction (SOP) task to replace Next Sentence Prediction (NSP)
  - in NSP, the negative next sentence is sampled from other passages that may have different topics with the current one, turning the NSP into a far easier topic model problem.
  - in SOP, two sentences that exchange their position are regarded as a negative sample, making the model concentrate on the coherence of the semantic meaning.

40

Maximize Mutual Information (MI)

- focuses on learning the explicit belonging relationships between local parts and global context. The relative positions between local parts are ignored.
- Generally, this kind of models optimize

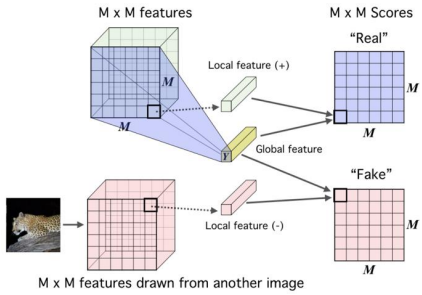
$$\max_{g_1 \in \mathcal{G}_1, g_2 \in \mathcal{G}_1} I(g_1(x_1), g_2(x_2))$$

  - Deep InfoMax / CPC
  - AMDIM / CMC
  - InfoWord
  - Deep Graph InfoMax (DGI) / InfoGraph
  - S^2GRL

41

Deep InfoMax (CV)

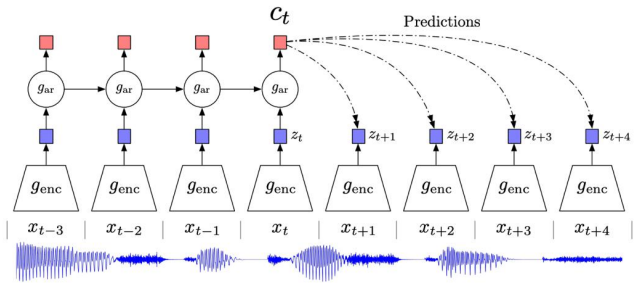
- Deep InfoMax: Learning deep representations by mutual information estimation and maximization (2018, Hjelm et al.)
  - the first one to explicitly model mutual information through a contrastive learning task, which maximize the MI between a local patch and its global context.
  - provides us with a new paradigm and boosts the development of self-supervised learning.



42

CPC (Audio / CV)

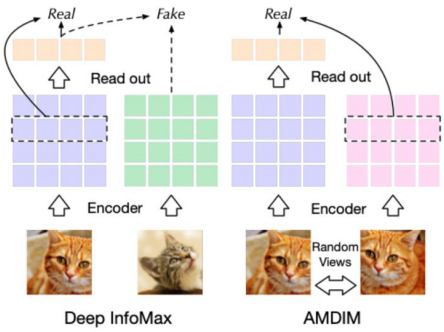
- CPC: Representation learning with contrastive predictive coding. (2018, Oord et al.)
  - a follower of Deep InfoMax for speech recognition
  - maximize the association between a segment of audio and its context audio
  - also being applied to CV



43

AMDIM (CV)

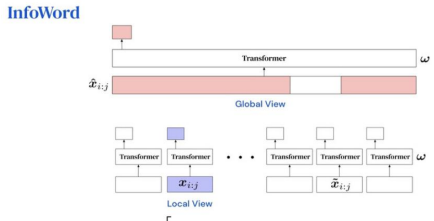
- AMDIM: Learning representations by maximizing mutual information across views (NIPS 2019, Bachman et al.)
  - enhances the positive association between a local feature and its context
- CMC: Contrastive multi-view coding (2019, Tian et al.)
  - extends single enhanced positive view to multi-view



44

InfoWord

- A mutual information maximization perspective of language representation learning (2019, Kong et al.)
  - proposes to maximize the mutual information between a global representation of a sentence and *n-grams* in it
  - The context is induced from the sentence with selected n-grams being masked, and the negative contexts are randomly picked out from the corpus



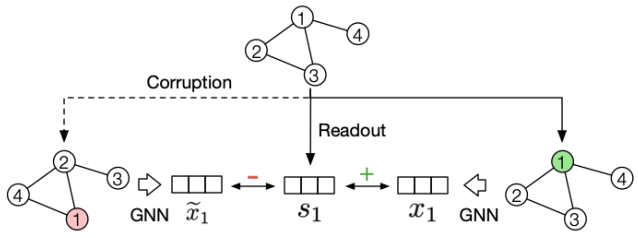
$$\mathcal{I}_{\text{DIM}} = \mathbb{E}_{p(\hat{\boldsymbol{x}}_{i:j}, \boldsymbol{x}_{i:j})} \left[g_\omega(\hat{\boldsymbol{x}}_{i:j})^\top g_\omega(\boldsymbol{x}_{i:j}) - \log \sum_{\tilde{\boldsymbol{x}}_{i:j} \in \tilde{\mathcal{S}}} \exp(g_\omega(\hat{\boldsymbol{x}}_{i:j})^\top g_\omega(\tilde{\boldsymbol{x}}_{i:j})) \right]$$

$$\mathcal{I}_{\text{INFOWORD}} = \lambda_{\text{MLM}} \mathcal{I}_{\text{MLM}} + \lambda_{\text{DIM}} \mathcal{I}_{\text{DIM}}$$

45

Deep Graph InfoMax (Graph)

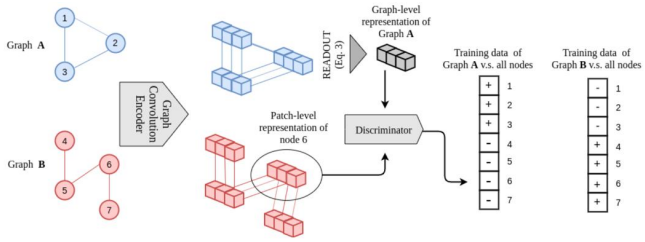
- Deep graph infomax (2018, Velickovic et al.)
  - considers a node's representation as the local feature and the average of randomly samples 2-hop neighbors as the context
  - To generate negative contexts, DGI proposes to corrupt the original context by keeping the sub-graph structure and permuting the node features



46

InfoGraph (Graph)

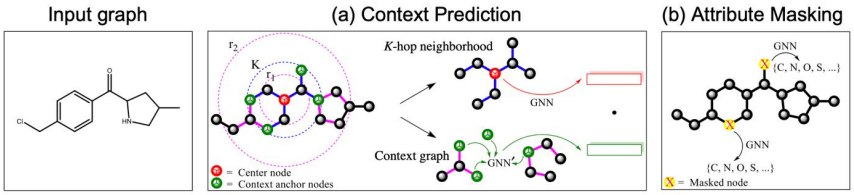
- Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization (2019, Sun et al.)
  - targets at learning graph-level representation rather than node level, maximizing the mutual information between graph-level representation and substructures at different levels



47

Strategies Pre-train GNN / S²GRL (Graph)

- Strategies for Pre-graining Graph Neural Networks (ICLR 2020, Hu et al.)
  - Context prediction: MI-based contrastive objective
  - Attribute masking: DAE-based generative objective



- Self-Supervised Graph Representation Learning via Global Context Prediction (2020, Peng et al.)
  - K-hop context prediction

48

Context-Context Contrast

- Recently, MoCo and SimCLR outperform the context-instance-based methods and achieve a competitive result to supervised methods under the linear classification protocol, through a context-to-context level direct comparison.
  - *Cluster-based Discrimination*
  - *Instance Discrimination*

49

Cluster-based Discrimination

- Context-context contrast is first studied in clustering-based methods
  - Deep Cluster
  - Local Aggregation
  - ClusterFit

50

Deep Cluster

- Deep clustering for unsupervised learning of visual features (ECCV 2018, Caron et al.)
  - Image classification asks the model to categorize images correctly, and the representation of images in the same category should be similar. Therefore, the motivation is to draw similar images near in the embedding space
  - In self-supervised learning, to solve the label problem, the model proposes to leverage clustering to yield pseudo labels and asks a discriminator to predict on images' label

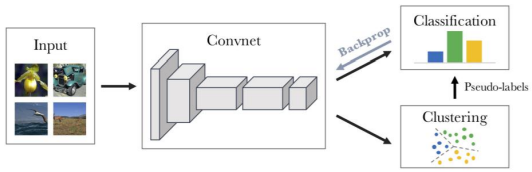
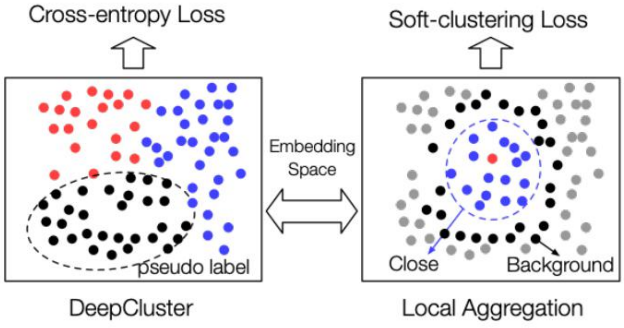


Fig. 1: Illustration of the proposed method: we iteratively cluster deep features and use the cluster assignments as pseudo-labels to learn the parameters of the convnet.

51

Local Aggregation

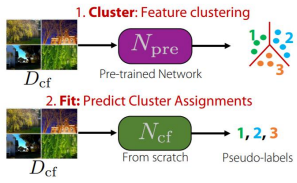
- Local aggregation for unsupervised learning of visual embeddings (IEEE ICCV 2019, Zhuang et al.)
  - Identifies neighbors separately for each example
  - employs an objective function that directly optimizes a local soft-clustering metric

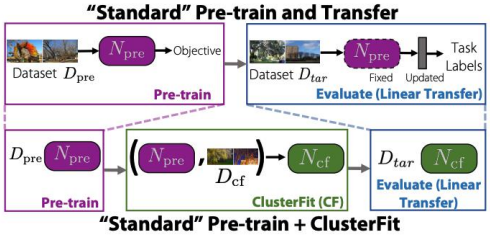


52

ClusterFit

- Clusterfit: Improving generalization of visual representations (2019, Yan et al.)
  - introduces a cluster prediction fine-tuning stage similar to DeepCluster between the above two stages, which improves the representation's performance on downstream classification evaluation





53

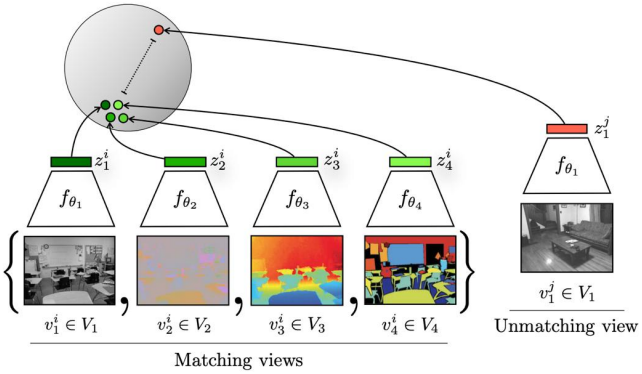
Instance Discrimination

- The prototype of leveraging instance discrimination as a pretext task is InstDisc.
  - Moco
  - SimCLR
  - InfoMin
  - BYOL
  - GCC

54

CMC (CV)

- CMC: Contrastive multi-view coding (2019, Tian et al.)
  - proposes to measure the context-context similarity rather than context-instance similarity



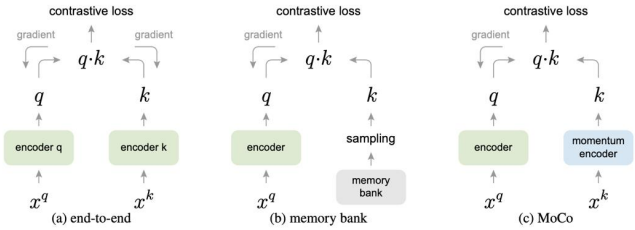
55

MoCo / MoCo v2 (CV)

- Momentum contrast for unsupervised visual representation learning (2019, He et al.)
  - leveraging instance discrimination via momentum contrast
  - optimize the following objective

$$\mathcal{L} = -\log \frac{\exp(q \cdot k_+/\tau)}{\sum_{i=0}^{K} \exp(q \cdot k_i/\tau)}$$

  - designs the momentum contrast learning with two encoders
  - employs a queue to save the recently encoded batches as negative samples

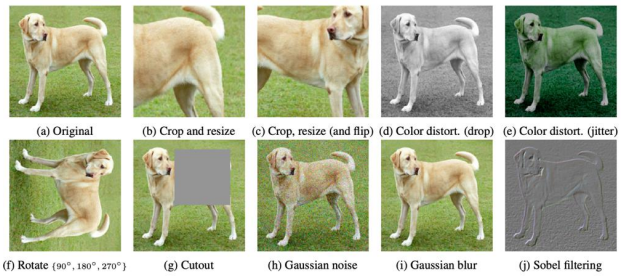


56

SimCLR

- A simple framework for contrastive learning of visual representations (2020, Chen et al.)
  - further illustrate the importance of a hard positive sample strategy by introducing data augmentation in 10 forms
  - use the pairwise contrastive loss NT-Xent loss

$$l_{i,j} = -\log \frac{\exp(\text{sim}(\hat{x}_i, \hat{x}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{I}_{[k \neq i]} \exp(\text{sim}(\hat{x}_i, \hat{x}_k)/\tau)} \qquad \mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} [l_{2i-1,2i} + l_{2i,2i-1}]$$



57

BYOL

- Bootstrap your own latent: A new approach to self-supervised learning (2020, Grill et al.)
  - discards negative sampling in self-supervised learning but achieve an even better result over InfoMin
  - regression loss rather than cross-entropy loss

$$\mathcal{L}_\theta^{\texttt{BYOL}} \triangleq \left\| \overline{q_\theta}(z_\theta) - \overline{z}'_\xi \right\|_2^2 = 2 - 2 \cdot \frac{\langle q_\theta(z_\theta), z'_\xi \rangle}{\left\| q_\theta(z_\theta) \right\|_2 \cdot \left\| z'_\xi \right\|_2}$$

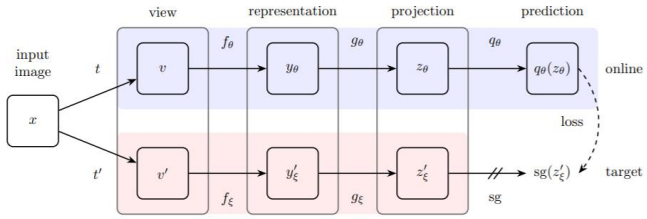
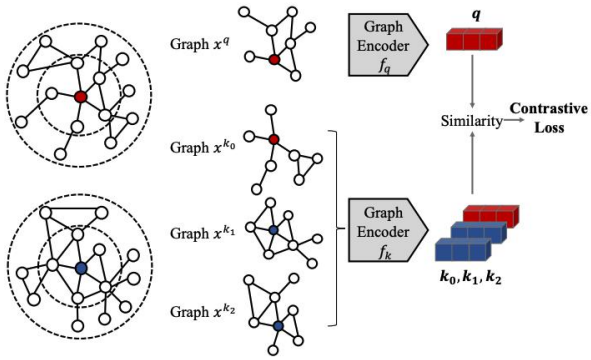


Figure 2: BYOL's architecture. BYOL minimizes a similarity loss between $q_\theta(z_\theta)$ and $\text{sg}(z'_\xi)$, where θ are the trained weights, ξ are an exponential moving average of θ and sg means stop-gradient. At the end of training, everything but f_θ is discarded, and y_θ is used as the image representation.

59

Graph Contrastive Coding (GCC)

- GCC: Graph contrastive coding for graph neural network pre-training(2020, Qiu et al.)
  - a pioneer to leverage instance discrimination as the pretext task for structural information pre-training



60

Self-supervised Contrastive Pre-training for Semi-supervised Self-training

- No matter how self-supervised learning models improve, they are still only powerful feature extractor, and to transfer to downstream task we still need abundant labels. And to bridge the gap between self pretraining and downstream task, semi-supervised learning is exactly what we are looking for.
- The success in combining self-supervised contrastive pretraining and semi-supervised self-training open up our eyes for a future data efficient deep learning paradigm. More work is expected for investigating their latent mechanisms.

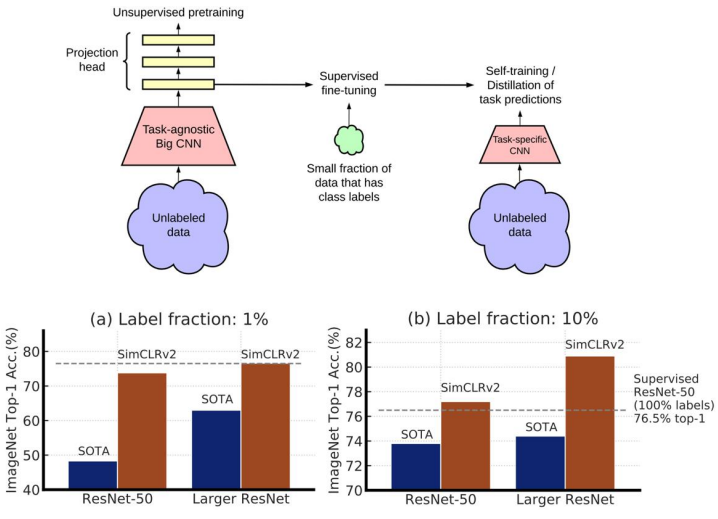
61

SimCLR v2 (CV)

- Big Self-Supervised Models are Strong Semi-Supervised Learners (2020, Ting et al)
  - Three-step framework to combine contrastive SSL and self-training semi-supervised learning
    1. Self-supervised pre-training: the same as SimCLR.
    2. Fine-tuning with 1% or 10% of ImageNet labels.
    3. Self-training: used fine-tuned model as teacher network to yield labels on unlabeled data to train a smaller student network
  - Other architecture modifications
  - Outperform supervised learning with only 10% of original labels

62

SimCLR v2 (CV)



63

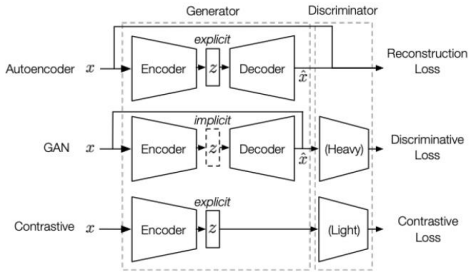
4. Generative-Contrastive (Adversarial) Self-supervised Learning

Generative-Contrastive (Adversarial) Self-supervised Learning

1. Why Generative-Contrastive (Adversarial)?
2. Generate with Complete Input
3. Recover with Partial Input
4. Pre-trained Language Model
5. Graph Learning
6. Domain Adaptation and Multi-modality Representation

Why Generative-Contrastive (Adversarial)?

- Generative SSL: maximum likelihood estimation (MLE)
  - Two fatal problems
    - sensitive and conservative distribution
    - low-level abstraction
- Discriminative and contrastive objectives can solve
  - adversarial methods absorb merits from both generative and contrastive methods together with some drawbacks.



66

Generate with Complete Input

- The inception of adversarial representation learning should be attributed to Generative Adversarial Networks (GAN), which proposes the adversarial training framework.
- Follow GAN, many variants emerge and reshape people's understanding of deep learning's potential.
  - GAN
  - AAE
  - BiGAN / ALI

67

GAN

- Unsupervised representation learning with deep convolutional generative adversarial networks (2015, Radford et al.)
  - optimize this min-max game

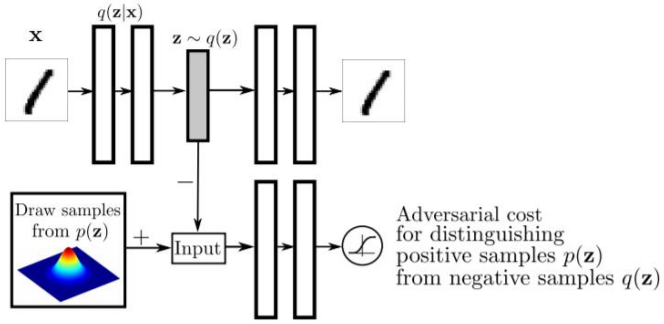
$$\min_G \max_D \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]$$

  - However, there is a gap between generation and representation. Compared to autoencoder's explicit latent sample distribution $p_z(z)$, GAN's latent distribution $p_z(z)$ is implicitly modeled. We need to extract this implicit distribution out.

68

AAE

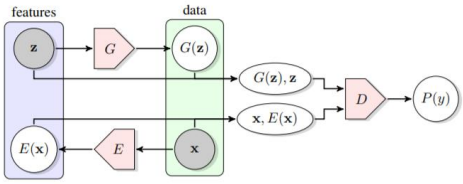
- Adversarial autoencoders. (2015, Makhzani et al.)
  - To bridge the gap of the GAN, AAE first proposes a solution to follow the natural idea of the autoencoder.
  - replace the generator with an explicit variational autoencoder (VAE).
  - substitutes the KL divergence function to a discriminative loss
  - However, AAE still preserves the reconstruction error, which contradicts with GAN's core idea.

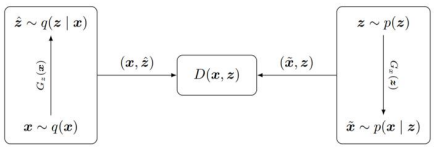


69

BiGAN / ALI

- BiGAN: Adversarial feature learning (2016, Donahue et al.)
- ALI: Adversarially learned inference (2016, Dumoulin et al.)
  - embrace adversarial learning without reservation and put forward a new framework.
  - The training goal is that encoder should learn to "convert" generator. The distribution does not make any assumption about the data itself. The distribution is shaped by the discriminator, which captures the semantic-level difference

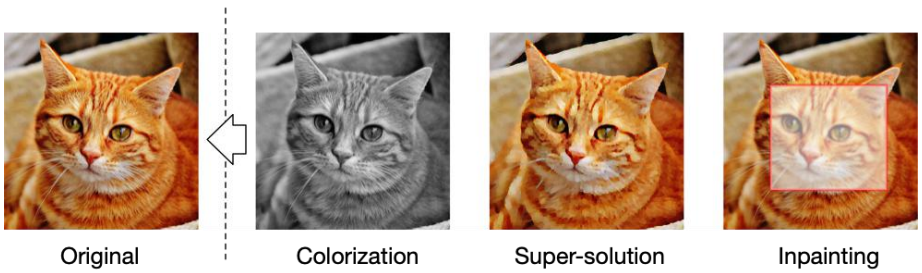




70

Recover with Partial Input

- GAN's architecture is not born for representation learning, and modification is needed to apply its framework.
  - colorization
  - inpainting
  - super-resolution



Original | Colorization | Super-solution | Inpainting

71

Colorization (CV)

- Colorful image colorization (ECCV 2016, Zhang et al.)
  - the problem can be described as given one color channel in an image and to predict the value of two other channels
  - the encoder and decoder networks can be set to any form of convolutional neural networks
  - transforms the generation task into a classification one

72

Inpainting (CV)

- Globally and locally consistent image completion (ACM ToG 2017, Iizuka et al.)
- Context encoders: Feature learning by inpainting (CVPR 2016, Pathak et al.)
  - ask the model to predict an arbitrary part of an image given the rest of it
  - a discriminator is employed to distinguish the inpainted image with the original one

73

Super-resolution (CV)

- Photo-realistic single image super-resolution using a generative adversarial network (CVPR 2017, Ledig et al.)
  - recover high-resolution images from blurred low-resolution ones in the adversarial setting

74

Pre-trained Language Model

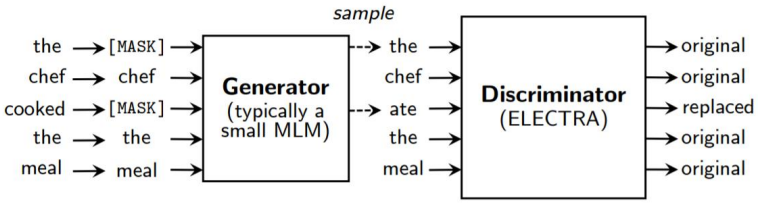
- For a long time, the pre-trained language model (PTM) focuses on maximum likelihood estimation based pretext task, because discriminative objectives are thought to be helpless due to languages' vibrant patterns. However, recently some work shows excellent performance and sheds light on contrastive objectives' potential in PTM.
  - ELECTRA
  - WKLM / REALM

75

ELECTRA

- ELECTRA: Pretraining text encoders as discriminators rather than generators (2020, Clark et al.)
  - ELECTRA proposes Replaced Token Detection (RTD) and leverages GAN's structure to pre-train a language model
  - The final objective could be written as

$$\min_{\theta_{\mathrm{G}},\theta_{\mathrm{D}}} \sum_{\boldsymbol{x}\in\mathcal{X}} \mathcal{L}_{\mathrm{MLM}}(\boldsymbol{x},\theta_{\mathrm{G}}) + \lambda\mathcal{L}_{\mathrm{Disc}}(\boldsymbol{x},\theta_{\mathrm{D}})$$



76

WKLM / REALM

- WKLM: Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model (2019, Xiong et al.)
  - For entities in Wikipedia paragraphs, WKLM replaced them with similar entities and trained the language model to distinguish them in a similar discriminative objective as ELECTRA, performing quite well in downstream tasks like question answering
- Realm: Retrieval-augmented language model pre-training (2020, Guu et al.)
  - conducts higher article-level retrieval augmentation to the language model.
  - not using the discriminative objective.

77

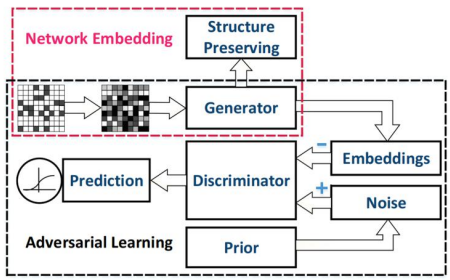
Graph Learning

- In graph learning, there are also attempts to utilize adversarial learning. Interestingly, their ideas are quite different from each other.
  - Adversarial Network Embedding (ANE)
  - GraphGAN
  - GraphSGAN

78

Adversarial Network Embedding (ANE)

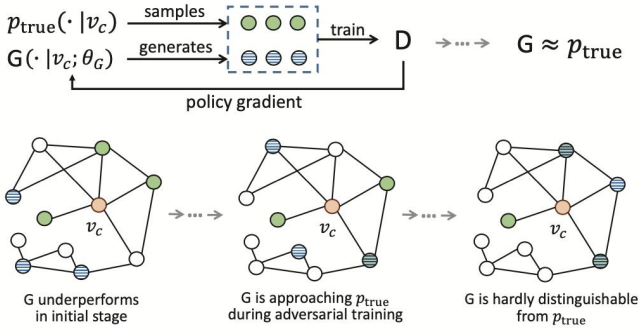
- Adversarial network embedding (AAAI 2018, Dai et al.)
  - designs a generator *G* that is updated in two stages: 1) *G* encodes sampled graph into target embedding and computes traditional NCE with a context encoder *F* like Skip-gram, 2) discriminator *D* is asked to distinguish embedding from *G* and a sampled one from a prior distribution.
  - The optimized objective is a sum of the above two objectives, and the generator *G* could yield better node representation for the classification task.



79

GraphGAN

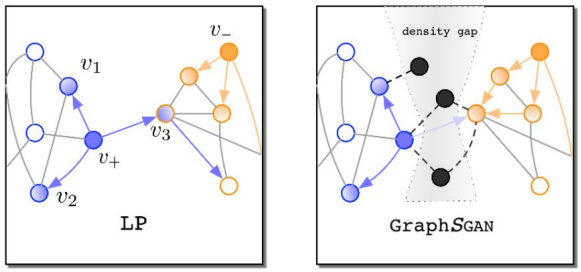
- Graphgan: Graph representation learning with generative adversarial nets (AAAI 2018, Wang et al.)
  - considers to model the link prediction task and follow the original GAN style discriminative objective to distinguish directly at node-level rather than representation-level.



80

GraphSGAN

- Semi-supervised learning on graphs with generative adversarial nets (ACM CIKM 2018, Ding et al.)
  - applies the adversarial method in semisupervised graph learning with the motivation that most classification errors in the graph are caused by marginal nodes



81

Domain Adaptation and Multi-modality Representation

- Essentially, the discriminator in adversarial learning serves to match the discrepancy between latent representation distribution and data distribution.
- This function naturally relates to domain adaptation and multi-modality representation problems, which aim at aligning different representation distribution.

82

5. Theory behind Self-supervised Learning

83

Theory behind Self-supervised Learning

1. GAN
2. Maximizing Lower Bound
3. Contrastive Self-supervised Representation Learning

84

GAN

- As generative models, GANs pays attention to the difference between real data distribution and generated data distribution.
- An important drawback of supervised learning is that it easily get trapped into spurious information. As an alternative, GAN show its superior potential in learning disentangled features empirically and theoretically.
  - Divergence Matching
  - Disentangled Representation

85

Divergence Matching

- Different divergence functions leads to different GAN variants.
  - f-GAN

86

f-GAN

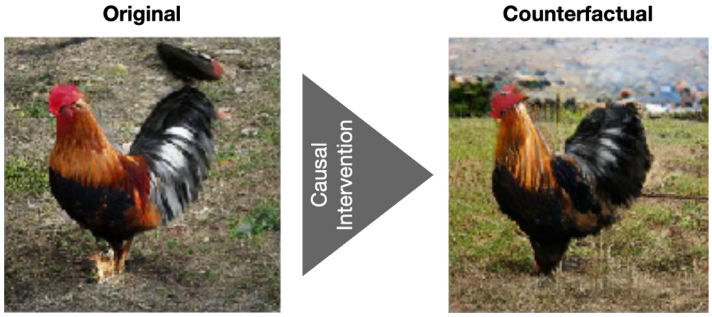
- f-gan: Training generative neural samplers using variational divergence minimization (NIPS 2016, Nowozin et al)
  - shows that the generative-adversarial approach is a special case of an exsiting more general variational divergence estimation problem and uses f-divergence to train the generative models
  - the optimization target of the minmax GAN is

$$\min_G \max_D (\mathbb{E}_{P_{data}(x)}[\log D(x)] + \mathbb{E}_{P_G(x;\theta)}[\log(1 - D(x))])$$

87

Divergence Matching

- GAN show its superior potential in learning disentangled features empirically and theoretically.
  - InfoGAN



88

InfoGAN

- Infogan: Interpretable representation learning by information maximizing generative adversarial nets (NIPS 2016, Chen et al)
  - first proposes to learn disentangled representation with DCGAN
  - Since mutual information is notoriously hard to compute, the authors leverage the variational inference approcach to estimates its lower bound, and the final objective for InfoGAN is modified as:

$$\min_G \max_D V_I(D,G) = V(D,G) - \lambda L_I(c; G(z,c))$$

  - Experiments show that InfoGAN surely learns a good disentangled representation on MNIST

89

Maximizing Lower Bound

- Maximizing Lower Bound
  - Evidence Lower Bound
  - Mutual Information

Evidence Lower Bound

- VAE learns the representation through learning a distribution $q\varphi(z|x)$ to approximate the posteriori distribution $p\theta(z|x)$:

$$\mathrm{KL}(q_\phi(z|x)||p_\theta(z|x)) = -\mathrm{ELBO}(\theta;\phi;x) + \log p_\theta(x)$$

$$\mathrm{ELBO} = E_{q_\phi(z|x)}[\log q_\phi(z|x)] - E_{p_\theta}[\log p_\theta(z,x)]$$

- VAE maximizes the *ELBO* to minimize the difference between $q\varphi(z|x)$ and $p\theta(z|x)$

$$\mathrm{ELBO}(\theta;\phi;x) = -\mathrm{KL}(q_{\phi(z|x)||p_\theta(z)}) + E_{q_\phi}[\log p_\theta(x|z)]$$

Mutual Information

- Most of current contrastive learning methods aim to maximize the MI of the input and its representation with joint density $p(x|y)$ and marginal densities $p(x)$ and $p(y)$:

$$I(X,Y) = \mathbb{E}_{p(x,y)}[\log \frac{p(x,y)}{p(y)p(x)}]$$
$$= \mathrm{KL}(p(x,y)|p(x)p(y))$$

  - Deep Infomax
  - InfoNCE

92

Deep Infomax

- Learning deep representations by mutual information estimation and maximization (2018, Hjelm et al)
  - maximizes the MI of local and global features and replaces KL-divergence with JS-divergence
  - the optimization target is

$$\max_T(\mathbb{E}_{p(x,y)}[\log(T(x,y)] + \mathbb{E}_{p(x)p(y)}[\log(1-T(x,y))])$$

  - From a probability point of view, GAN and Deep InfoMax are derived from the same process but for a different learning target

93

InfoNCE

- Representation learning with contrastive predictive coding (2018, Oord et al)
  - Instance Discrimination directly optimizes the proportion of gap of positive pairs and negative pairs. One of the commonly used estimators is InfoNCE:

$$\begin{aligned}\mathcal{L} &= \mathbb{E}_X[-\log \frac{\exp(x \cdot y/\tau)}{\Sigma_{i=0}^{K} \exp(x \cdot x^-/\tau) + \exp(x \cdot y/\tau)}] \\ &= \mathbb{E}_X[-\log \frac{p(x|y)/p(x)}{p(x|y)/p(x) + \Sigma_{x^- \in \mathbb{X}_-} p(x^-|y)/p(x^-)}] \\ &\approx \mathbb{E}_X \log[1 + \frac{p(x)}{p(x|y)}(N-1)\mathbb{E}_{x^-}\frac{p(x^-|y)}{p(x^-)}] \\ &\geq \mathbb{E}_X \log[\frac{p(x)}{p(x|y)}N] \\ &= -I(y,x) + \log(N)\end{aligned}$$

  - Therefore the MI $I(x|\ y) \geq \log(N) - L$. The approximation becomes increasingly accurate, and $I(x|\ y)$ also increases as N grows.

94

InfoNCE

- Representation learning with contrastive predictive coding (2018, Oord et al)
  - MI maximization can also be analyzed from the metric learning view. By connecting InfoNCE to the triplet (k-plet) loss in deep learning community. The InfoNCE can be rewriten as follows:

$$\begin{aligned}I_{NCE} &= \mathbb{E}[\frac{1}{k}\Sigma_{i=1}^{k} \log \frac{e^{f(x_i,y_i)}}{\frac{1}{k}\Sigma_{j=1}^{k} e^{f(x_i,y_j)}}] \\ &= \log k - \mathbb{E}[\frac{1}{k}\Sigma_{i=1}^{k} \log(1 + \Sigma_{j\neq i}^{k} e^{f(x_i,y_j)-f(x_i,y_i)})]\end{aligned}$$

  - In particular f is contrained to the form $f(x;\ y) = \varphi(x)^T \varphi(y)$ for a certain function φ. Then the InfoNCE is corresponding to the expectation of the *multi-class k-pair* loss:

$$L_{k-pair}(\phi) = \frac{1}{k}\Sigma_{i=1}^{k} \log(1 + \Sigma_{j\neq i} e^{\phi(x_i)^T\phi(y_j) - \phi(x_i)^T\phi(y_i)})$$

95

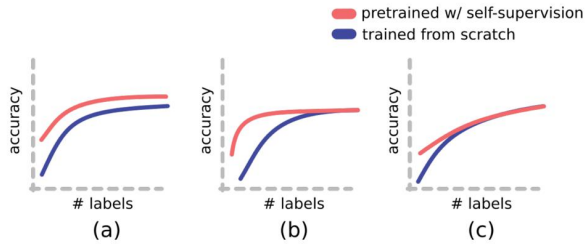
Contrastive Self-supervised Representation Learning

- Relationship with Supervised Learning
- Understand Contrastive Loss
- Generalization

96

Relationship with Supervised Learning

- reaches higher accuracy with fewer labels but plateaus to the same accuracy as baseline, and converges to baseline performance before accuracy plateaus
- cannot help on improving accuracy
- self-supervised trained neural networks are more robust and stable
- Outperform supervised distribution in out-of-distribution detection on difficult, near-distribution outliers



97

Understand Contrastive Loss

- theoretical analysis on functions of contrastive loss, and split it into two terms:

$$\mathcal{L}_{\text{contrast}} = \mathbb{E}[-\log \frac{e^{f_x^T f_y/\tau}}{e^{f_x^T f_y/\tau} + \sum_i e^{f_x^T f_{y_i^-}/\tau}}]$$
$$= \underbrace{\mathbb{E}[-f_x^T f_y/\tau]}_{\text{alignment}} + \underbrace{\mathbb{E}[\log(e^{f_x^T f_y/\tau} + \sum_i e^{f_x^T f_{y_i^-}/\tau})]}_{\text{uniformity}}$$

- directly optimize *alignment* and *uniformity* loss as:

$$\mathcal{L}_{\text{align}}(f;\alpha) \triangleq \mathbb{E}_{(x,y)\sim p_{\text{pos}}}[||f(x) - f(y)||_2^\alpha]$$
$$\mathcal{L}_{\text{uniform}}(f;t) \triangleq \log \mathbb{E}_{x,y\sim p_{\text{data}}}[e^{-t||f(x)-f(y)||_2^2}]$$

- The success of BYOL raises doubt that whether alignment and uniformity are necessarily in the form of upper two losses. This illustrates us that we may still achieve uniformity via other techniques

98

Generalization

- Contrastive learning assumes that similar data pair (x, x+) comes from a distribution D_{sim} and negative sample (x_1^-,⋯, x_k^-) from a dstribution D_{neg} that is presumably unrelated to x. Under the hypothesis that semantically similar points are sampled from the same latent class, the unsupervised loss can be expressed as:

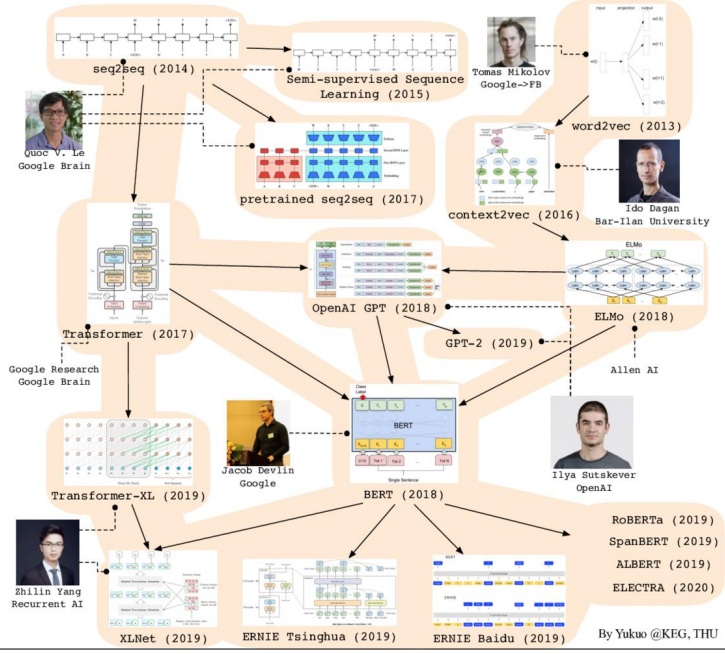
$$\mathcal{L}_{un}(f) = \mathbb{E}_{\substack{x^+\sim\mathcal{D}_{sim} \\ x^-\sim\mathcal{D}_{neg}}}[l(\{f(x)^T(f(x^+) - f(x^-))\})]$$

- to find a funtion that minimizes the empirical unsupervised loss within the capacity of the used encoder
- not always pick the best supervised representation function
- is theoretically proved to benefit the downstream classification tasks

99

6. BERTology

Overview



3 / 53

Overview

4 / 53

Background

- Language model: given an sequence of length n, it assigns a probability $p\left(x_1, x_2, x_3, \ldots, x_n\right)$ to the whole sequence. The probability of standard LM can be decomposed as $p\left(x_1, x_2, \ldots, x_n\right) = \prod_{t=1}^{n} p\left(x_t | x_1, \ldots, x_{t-1}\right)$.
- Sequence to sequence (seq2seq) learning: given an input sequence $x_1, x_2, ..., x_m$ and an output sequence $y_1, y_2, ..., y_n$, the objective of seq2seq learning is to maximize the likelihood $p(y_n, y_{n-1}, ..., y_1 | x_1, x_2, ..., x_m)$. Common seq2seq methods decompose this objective as $p\left(y_n, y_{n-1}, \ldots, y_1 | x_1, x_2, \ldots, x_m\right) = \prod_{t=1}^{n} p\left(y_t | y_{t-1}, \ldots, y_1; x_1, x_2, \ldots, x_m\right)$.

Pre-BERT Era

- Semi-supervised Sequence Learning
- context2vec: Learning Generic Context Embedding with Bidirectional LSTM
- Pre-trained seq2seq: Unsupervised Pretraining for Sequence to Sequence Learning
- ELMo: Deep contextualized word representations
- OpenAI GPT: Improving Language Understanding by Generative Pre-Training

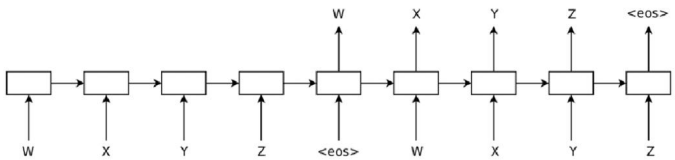
Semi-supervised Sequence Learning[1]

- This paper presents two approaches that use unlabeled data to improve sequence learning with recurrent networks.
- One is to predict what comes next in a sequence, which is a conventional language model.
- The other is to use a sequence autoencoder, which reads the input sequence into a vector and predicts the input sequence again.
- These two methods can be used as a "pretraining" step for a later supervised sequence learning algorithm.

[1] Dai, Andrew M., and Quoc V. Le. "Semi-supervised sequence learning." Advances in neural information processing systems. 2015.

Semi-supervised Sequence Learning (cont.)

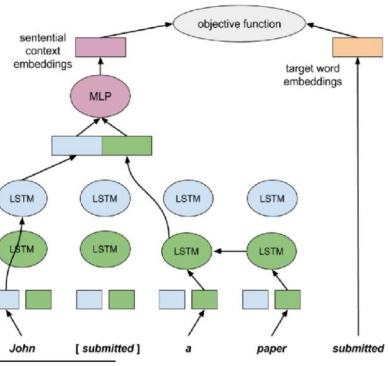
- The sequence autoencoder is inspired by seq2seq, except that it is an unsupervised learning model. The objective is to reconstruct the input sequence itself.



- The weights obtained from the sequence autoencoder can be used as an initialization of downstream LSTM networks.

context2vec[1]

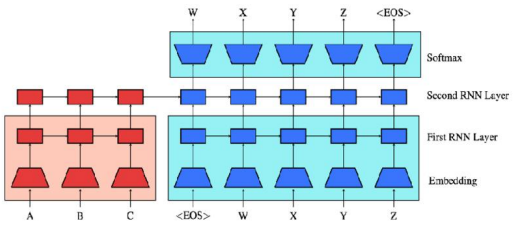
- context2vec is an unsupervised model for efficiently learning a generic context embedding function from large corpora, using bidirectional LSTM. The architecture is based on word2vec's CBOW but replaces its context modeling with LSTM.



[1]Melamud, Oren, Jacob Goldberger, and Ido Dagan. "context2vec: Learning generic context embedding with bidirectional lstm." CoNLL 2016.

Pre-trained Seq2seq[1]

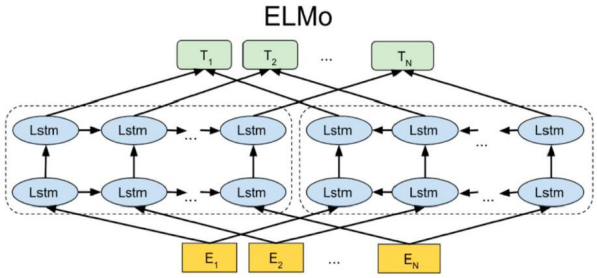
- This work presents a general unsupervised learning method to improve the accuracy of seq2seq models.
- The weights of a seq2seq model are initialized with the pretrained weights of two language models and then fine-tuned with labeled data.
- All parameters in a shaded box are pretrained from language models.



[1]Ramachandran, Prajit, Peter J. Liu, and Quoc V. Le. "Unsupervised pretraining for sequence to sequence learning." arXiv preprint arXiv:1611.02683 (2016).

ELMo[1]

- ELMo introduces a new type of deep contextualized word presentation, which are functions of the internal states of a deep bidirectional language model (biLM).
- ELMo uses the concatenation of independently trained left-to-right and right-to-left LSTMs to generate features for downstream tasks.



[1]Peters, Matthew E., et al. "Deep contextualized word representations." arXiv preprint arXiv:1802.05365 (2018).

ELMo (cont.)

- ELMo formulation jointly maximizes the log likelihood of the forward and backward directions:

$$\sum_{k=1}^{N} \left(\log p\left(t_k | t_1, \ldots, t_{k-1}; \Theta_x, \vec{\Theta}_{LSTM}, \Theta_s\right) + \log p\left(t_k | t_{k+1}, \ldots, t_N; \Theta_x, \overleftarrow{\Theta}_{LSTM}, \Theta_s\right) \right) \quad (1)$$

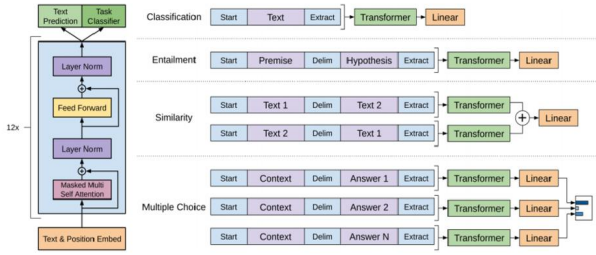
where Θ_x and Θ_s are the parameters of token representation and softmax layer.

- ELMo Results

Task	Previous SOTA		Our baseline	ELMo + baseline	Increase (absolute/ relative)
SQuAD	Liu et al. (2017)	84.4	81.1	85.8	4.7 / 24.9%
SNLI	Chen et al. (2017)	88.6	88.0	88.7 ± 0.17	0.7 / 5.8%
SRL	He et al. (2017)	81.7	81.4	84.6	3.2 / 17.2%
Coref	Lee et al. (2017)	67.2	67.2	70.4	3.2 / 9.8%
NER	Peters et al. (2017)	91.93 ± 0.19	90.15	92.22 ± 0.10	2.06 / 21%
SST-5	McCann et al. (2017)	53.7	51.4	54.7 ± 0.5	3.3 / 6.8%

OpenAI GPT[1]

- OpenAI GPT uses generative pre-training of a language model on a diverse corpus of unlabeled text, followed by discriminative fine-tuning on each specific task.
- OpenAI GPT uses a left-to-right Transformer.



[1]Radford, Alec, et al. "Improving language understanding by generative pre-training." URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf (2018).

OpenAI (cont.)

- GPT uses a standard language modeling for pre-training

$$\sum_i \log P\left(u_i | u_{i-k}, \ldots, u_{i-1}; \Theta\right)$$

where P is modeled using a multi-layer Transformer

$$h_0 = UW_e + W_p$$

$$h_l = \text{transformer\_block}\left(h_{l-1}\right) \forall i \in [1, n]$$

$$P(u) = \text{softmax}\left(h_n W_e^T\right)$$

where W_e is the token embedding and W_p is the position embedding.

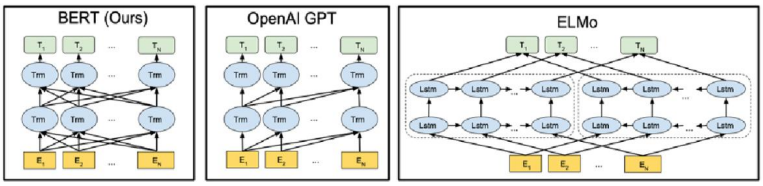
- Results on GLUE

Method	Classification		Semantic Similarity			GLUE
	CoLA (mc)	SST2 (acc)	MRPC (F1)	STSB (pc)	QQP (F1)	
Sparse byte mLSTM [16]	-	**93.2**	-	-	-	-
TF-KLD [23]	-	-	**86.0**	-	-	-
ECNU (mixed ensemble) [60]	-	-	-	81.0	-	-
Single-task BiLSTM + ELMo + Attn [64]	35.0	90.2	80.2	55.5	66.1	64.8
Multi-task BiLSTM + ELMo + Attn [64]	18.9	91.6	83.5	72.8	63.3	68.9

Overview

BERT[1]

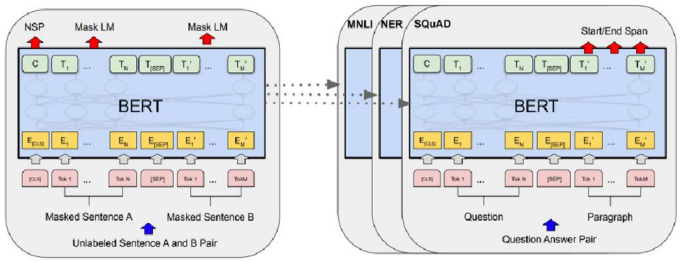
- BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers.
- The pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks. range of tasks



[1]Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

BERT (cont.)

- Overall pre-training and fine-tuning procedures for BERT.
- BERT uses two unsupervised tasks: masked language model (MLM) and next sentence prediction (NSP).



BERT Results

- Results on GLUE

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	Average -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

- Results on SQuAD

System	Dev EM	Dev F1	Test EM	Test F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	-	-	82.3	91.2
#1 Ensemble - nlnet	-	-	86.0	91.7
#2 Ensemble - QANet	-	-	84.5	90.5
Published				
BiDAF+ELMo (Single)	-	85.6	-	85.8
R.M. Reader (Ensemble)	81.2	87.9	82.3	88.5
Ours				
$\text{BERT}_{\text{BASE}}$ (Single)	80.8	88.5	-	-
$\text{BERT}_{\text{LARGE}}$ (Single)	84.1	90.9	-	-
$\text{BERT}_{\text{LARGE}}$ (Ensemble)	85.8	91.8	-	-
$\text{BERT}_{\text{LARGE}}$ (Sgl.+TriviaQA)	**84.2**	**91.1**	**85.1**	**91.8**
$\text{BERT}_{\text{LARGE}}$ (Ens.+TriviaQA)	**86.2**	**92.2**	**87.4**	**93.2**

(a) SQuAD 1.1

System	Dev EM	Dev F1	Test EM	Test F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	86.3	89.0	86.9	89.5
#1 Single - MIR-MRC (F-Net)	-	-	74.8	78.0
#2 Single - nlnet	-	-	74.2	77.1
Published				
unet (Ensemble)	-	-	71.4	74.9
SLQA+ (Single)	-		71.4	74.4
Ours				
$\text{BERT}_{\text{LARGE}}$ (Single)	78.7	81.9	80.0	83.1

(b) SQuAD 2.0

Overview

Post-BERT Era

- RoBERTa (2019)
- Transformer-XL and XLNet (2019)
- ERNIE (Tsinghua) (2019)
- ERINE (Baidu) (2019)
- ALBERT (2019)
- ELECTRA (2020)
- Pre-training with encoder-decoder architecture, BART (2019) and T5 (2019).
- Sparse Transformer (2019) and GPT-3 (2020)

RoBERTa[1]

- RoBERTa: A Robustly Optimized BERT Pretraining Approach
- RoBERTa finds that BERT is under-trained due to hyperparameters and training set.
- BERT can be improved by:
  - longer inputs (only full length sentence)
  - larger batch size and learning rate
  - larger dataset
  - No next sentence prediction
  - Dynamic masking

[1]Liu, Yinhan, et al. "Roberta: A robustly optimized bert pretraining approach." arXiv preprint arXiv:1907.11692 (2019).

RoBERTa Results

- Results on SQuAD

Model	SQuAD 1.1 EM	SQuAD 1.1 F1	SQuAD 2.0 EM	SQuAD 2.0 F1
Single models on dev, w/o data augmentation				
$\text{BERT}_{\text{LARGE}}$	84.1	90.9	79.0	81.8
$\text{XLNet}_{\text{LARGE}}$	**89.0**	94.5	86.1	88.8
RoBERTa	88.9	**94.6**	**86.5**	**89.4**
Single models on test (as of July 25, 2019)				
$\text{XLNet}_{\text{LARGE}}$			86.3†	89.1†
RoBERTa			86.8	89.8
XLNet + SG-Net Verifier			**87.0**†	**89.9**†

- Results on GLUE

	MNLI	QNLI	QQP	RTE	SST	MRPC	CoLA	STS	WNLI	Avg
Single-task single models on dev										
$\text{BERT}_{\text{LARGE}}$	86.6/-	92.3	91.3	70.4	93.2	88.0	60.6	90.0	-	-
$\text{XLNet}_{\text{LARGE}}$	89.8/-	93.9	91.8	83.8	95.6	89.2	63.6	91.8	-	-
RoBERTa	**90.2/90.2**	**94.7**	**92.2**	**86.6**	**96.4**	**90.9**	**68.0**	**92.4**	**91.3**	-
Ensembles on test (from leaderboard as of July 25, 2019)										
ALICE	88.2/87.9	95.7	**90.7**	83.5	95.2	92.6	**68.6**	91.1	80.8	86.3
MT-DNN	87.9/87.4	96.0	89.9	86.3	96.5	92.7	68.4	91.1	89.0	87.6
XLNet	90.2/89.8	98.6	90.3	86.3	**96.8**	**93.0**	67.8	91.6	**90.4**	88.4
RoBERTa	**90.8/90.2**	**98.9**	90.2	**88.2**	96.7	92.3	67.8	**92.2**	89.0	**88.5**

Transformer-XL

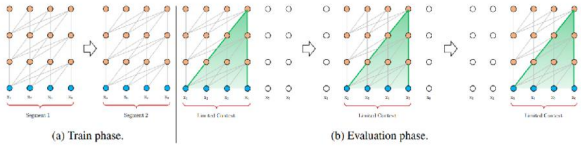


Figure 1: Limited Context of Vanilla Models

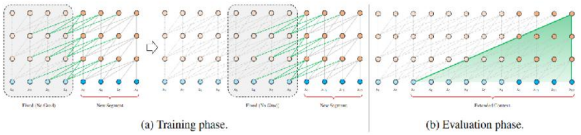


Figure 2: Extended Context

- Propose relative positional encoding. Instead of encoding absolution positions such as i and j, it encodes the relative position $i - j$

XLNet[1]

- We first review and compare the conventional auto-regressive language modeling and BERT for language pretraining.
- The objective of auto-regressive language modeling:

$$\max_{\theta} \log p_\theta(\mathbf{x}) = \sum_{t=1}^{T} \log p_\theta\left(x_t | \mathbf{x}_{<t}\right) = \sum_{t=1}^{T} \log \frac{\exp\left(h_\theta\left(\mathbf{x}_{1:t-1}\right)^\top e\left(x_t\right)\right)}{\sum_{x'} \exp\left(h_\theta\left(\mathbf{x}_{1:t-1}\right)^\top e\left(x'\right)\right)} \quad (2)$$

- The objective of BERT is to reconstruct masked tokens $\overline{\mathbf{x}}$ from a corrupted version $\hat{\mathbf{x}}$:

$$\max_{\theta} \log p_\theta(\overline{\mathbf{x}}|\hat{\mathbf{x}}) \approx \sum_{t=1}^{T} m_t \log p_\theta\left(x_t|\hat{\mathbf{x}}\right) = \sum_{t=1}^{T} m_t \log \frac{\exp\left(H_\theta(\hat{\mathbf{x}})_t^\top e\left(x_t\right)\right)}{\sum_{x'} \exp\left(H_\theta(\hat{\mathbf{x}})_t^\top e\left(x'\right)\right)} \quad (3)$$

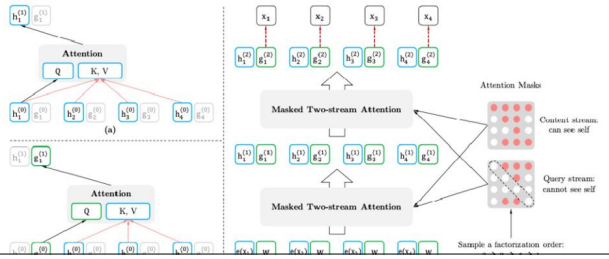
[1]Yang, Zhilin, et al. "Xlnet: Generalized autoregressive pretraining for language understanding." Advances in neural information processing systems. 2019.

XLNet (cont.)

- Instead of using a fixed forward or backward factorization order as in conventional Auto-Regressive models, XLNet maximizes the expected log likelihood of a sequence w.r.t. all possible permutations of the factorization order.

$$\max_{\theta} \quad \mathbb{E}_{\mathbf{z}\sim\mathcal{Z}_T}\left[\sum_{t=1}^{T}\log p_\theta\left(x_{z_t}|\mathbf{x}_{\mathbf{z}<t}\right)\right]$$

where $\mathcal{Z}_T$ is the set of all possible permutations of the length-T index sequence.



XLNet Results

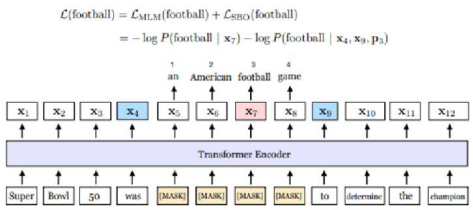
- Results on SQuAD

SQuAD2.0	EM	F1	SQuAD1.1	EM	F1
Dev set results (single model)					
BERT [10]	78.98	81.77	BERT† [10]	84.1	90.9
RoBERTa [21]	86.5	89.4	RoBERTa [21]	88.9	94.6
XLNet	**87.9**	**90.6**	XLNet	**89.7**	**95.1**
Test set results on leaderboard (single model, as of Dec 14, 2019)					
BERT [10]	80.005	83.061	BERT [10]	85.083	91.835
RoBERTa [21]	86.820	89.795	BERT* [10]	87.433	93.294
XLNet	**87.926**	**90.689**	XLNet	**89.898‡**	**95.080‡**

- Results on GLUE

Model	MNLI	QNLI	QQP	RTE	SST-2	MRPC	CoLA	STS-B	WNLI
Single-task single models on dev									
BERT [2]	86.6/-	92.3	91.3	70.4	93.2	88.0	60.6	90.0	-
RoBERTa [21]	90.2/90.2	94.7	92.2	**86.6**	96.4	**90.9**	68.0	92.4	-
XLNet	**90.8/90.8**	**94.9**	**92.3**	85.9	**97.0**	90.8	**69.0**	**92.5**	-
Multi-task ensembles on test (from leaderboard as of Oct 28, 2019)									
MT-DNN* [20]	87.9/87.4	96.0	89.9	86.3	96.5	92.7	68.4	91.1	89.0
RoBERTa* [21]	90.8/90.2	98.9	90.2	88.2	96.7	92.3	67.8	92.2	89.0
XLNet*	**90.9/90.9**†	**99.0**†	**90.4**†	**88.5**	**97.1**†	**92.9**	**70.2**	**93.0**	**92.5**

SpanBERT[1]

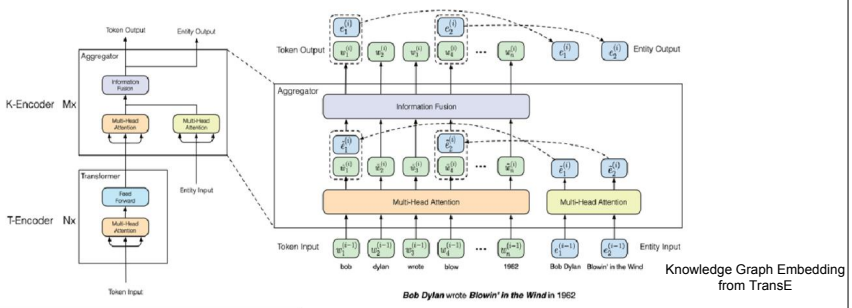


- Span Masking: Sample a span length from a geometric distribution, then randomly select the starting point for the span to be masked.
- Span Boundary Objective (SBO): using boundary tokens to predicted the masked span. Suppose the start and end of a masked span is s and e, resp, then use $\boldsymbol{x}_{s-1}, \boldsymbol{x}_{e+1}, p_i$ to predict the i-th token in the span, where $\boldsymbol{p}_i$ is a relative positional encoding.

[1] Joshi, Mandar, et al. "Spanbert: Improving pre-training by representing and predicting spans." TACL.

ERNIE (Tsinghua)[1]

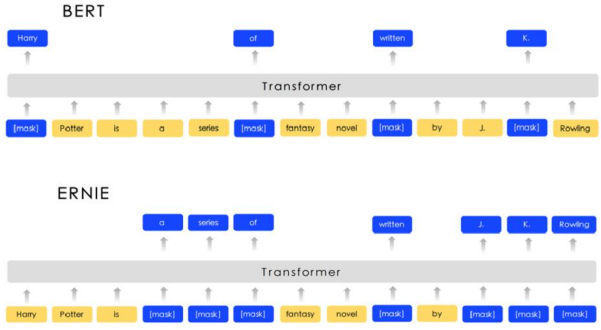
- ERNIE: Enhanced Language Representation with Informative Entities
- ERNIE utilizes both large-scale textual corpora and knowledge graphs to train an enhanced language representation model, which can take full advantage of lexical, syntactic, and knowledge information simultaneously.



[1] Zhang, Zhengyan, et al. "ERNIE: Enhanced language representation with informative entities." arXiv preprint arXiv:1905.07129 (2019).

ERNIE (Baidu)[1]

- ERNIE: Enhanced Representation through Knowledge Integration
- ERNIE is designed to learn language representation enhanced by knowledge masking strategies, which includes entity-level masking and phrase-level masking.



[1]Sun, Yu, et al. "Ernie: Enhanced representation through knowledge integration." arXiv preprint arXiv:1904.09223 (2019).

ALBERT[1]

- ALBERT uses two parameter-reduction techniques to lower memory consumption and increase the training speed of BERT.
- Factorize embedding parameterization: reduce the embedding parameters from $O(V \times H)$ to $O(V \times E + E \times H)$ where $H \gg E$.
- Cross-layer parameters sharing: a default decision for ALBERT is to share all parameters across layers.
- Sentence-order prediction replaces NSP

Model		Parameters	Layers	Hidden	Embedding	Parameter-sharing
BERT	base	108M	12	768	768	False
	large	334M	24	1024	1024	False
ALBERT	base	12M	12	768	128	True
	large	18M	24	1024	128	True
	xlarge	60M	24	2048	128	True
	xxlarge	235M	12	4096	128	True

[1]Lan, Zhenzhong, et al. "Albert: A lite bert for self-supervised learning of language representations." arXiv preprint arXiv:1909.11942 (2019).

ALBERT Results

- Results on GLUE

Models	MNLI	QNLI	QQP	RTE	SST	MRPC	CoLA	STS	WNLI	Avg
Single-task single models on dev										
BERT-large	86.6	92.3	91.3	70.4	93.2	88.0	60.6	90.0	-	-
XLNet-large	89.8	93.9	91.8	83.8	95.6	89.2	63.6	91.8	-	-
RoBERTa-large	90.2	94.7	**92.2**	86.6	96.4	**90.9**	68.0	92.4	-	-
ALBERT (1M)	90.4	95.2	92.0	88.1	96.8	90.2	68.7	92.7	-	-
ALBERT (1.5M)	**90.8**	**95.3**	**92.2**	**89.2**	**96.9**	**90.9**	**71.4**	**93.0**	-	-
Ensembles on test (from leaderboard as of Sept. 16, 2019)										
ALICE	88.2	95.7	**90.7**	83.5	95.2	92.6	**69.2**	91.1	80.8	87.0
MT-DNN	87.9	96.0	89.9	86.3	96.5	92.7	68.4	91.1	89.0	87.6
XLNet	90.2	98.6	90.3	86.3	96.8	93.0	67.8	91.6	90.4	88.4
RoBERTa	90.8	98.9	90.2	88.2	96.7	92.3	67.8	92.2	89.0	88.5
Adv-RoBERTa	91.1	98.8	90.3	88.7	96.8	93.1	68.0	92.4	89.0	88.8
ALBERT	**91.3**	**99.2**	90.5	**89.2**	**97.1**	**93.4**	69.1	**92.5**	**91.8**	**89.4**

- Results on SQuAD and RACE

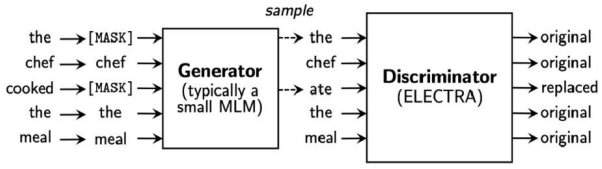
Models	SQuAD1.1 dev	SQuAD2.0 dev	SQuAD2.0 test	RACE test (Middle/High)
Single model (from leaderboard as of Sept. 23, 2019)				
BERT-large	90.9/84.1	81.8/79.0	89.1/86.3	72.0 (76.6/70.1)
XLNet	94.5/89.0	88.8/86.1	89.1/86.3	81.8 (85.5/80.2)
RoBERTa	94.6/88.9	89.4/86.5	89.8/86.8	83.2 (86.5/81.3)
UPM	-	-	89.9/87.2	-
XLNet + SG-Net Verifier++	-	-	90.1/87.2	-
ALBERT (1M)	94.8/89.2	89.9/87.2	-	86.0 (88.2/85.1)
ALBERT (1.5M)	**94.8/89.3**	**90.2/87.4**	**90.9/88.1**	**86.5 (89.0/85.5)**
Ensembles (from leaderboard as of Sept. 23, 2019)				
BERT-large	92.2/86.2	-	-	-
XLNet + SG-Net Verifier	-	-	90.7/88.2	-
UPM	-	-	90.7/88.2	
XLNet + DAAF + Verifier	-	-	90.9/88.6	-
DCMN+	-	-	-	84.1 (88.5/82.3)
ALBERT	**95.5/90.1**	**91.4/88.9**	**92.2/89.7**	**89.4 (91.2/88.6)**

ELECTRA[1]

- This paper proposes a replaced token detection task. This approach replaces some tokens with plausible alternatives sampled from a generator network, and then trains a discriminative model to predict whether each token was replaced by a generator sample or not.

$$\mathcal{L}_{\mathrm{MLM}}\left(\boldsymbol{x}, \theta_G\right) = \mathbb{E}\left(\sum_{i \in \boldsymbol{m}} -\log p_G\left(x_i | \boldsymbol{x}^{\mathrm{masked}}\right)\right) \tag{4}$$

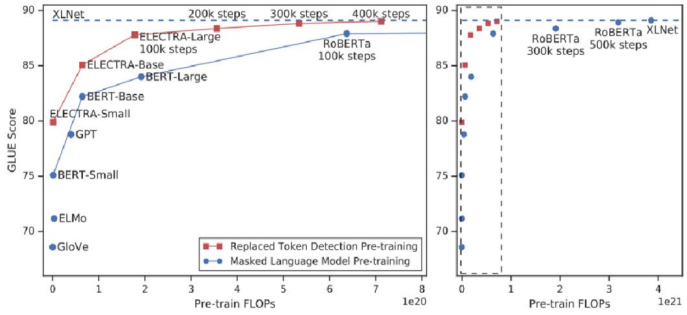
$$\min_{\theta_G, \theta_D} \sum_{\boldsymbol{x} \in \mathcal{X}} \mathcal{L}_{\mathrm{MLM}}\left(\boldsymbol{x}, \theta_G\right) + \lambda \mathcal{L}_{\mathrm{Disc}}\left(\boldsymbol{x}, \theta_D\right) \tag{5}$$



[1] Clark, Kevin, et al. "Electra: Pre-training text encoders as discriminators rather than generators." arXiv preprint arXiv:2003.10555 (2020).

ELECTRA FLOPS

- ELECTRA substantially outperforms MLM-based methods such as BERT and XLNet given the same model size, data, and computation.



ELECTRA Results

- Results on GLUE

Model	Train FLOPs	CoLA	SST	MRPC	STS	QQP	MNLI	QNLI	RTE	WNLI	Avg.*	Score
BERT	1.9e20 (0.06x)	60.5	94.9	85.4	86.5	89.3	86.7	92.7	70.1	65.1	79.8	80.5
RoBERTa	3.2e21 (1.02x)	67.8	96.7	89.8	91.9	90.2	90.8	95.4	88.2	89.0	88.1	88.1
ALBERT	3.1e22 (10x)	69.1	**97.1**	**91.2**	92.0	90.5	**91.3**	–	89.2	91.8	89.0	–
XLNet	3.9e21 (1.26x)	70.2	**97.1**	90.5	**92.6**	90.4	90.9	–	88.5	**92.5**	89.1	–
ELECTRA	3.1e21 (1x)	**71.7**	**97.1**	90.7	92.5	**90.8**	**91.3**	**95.8**	**89.8**	**92.5**	**89.5**	**89.4**

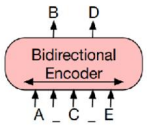
- Results on SQuAD

Model	Train FLOPs	Params	SQuAD 1.1 dev EM	SQuAD 1.1 dev F1	SQuAD 2.0 dev EM	SQuAD 2.0 dev F1	SQuAD 2.0 test EM	SQuAD 2.0 test F1
BERT-Base	6.4e19 (0.09x)	110M	80.8	88.5	–	–	–	–
BERT	1.9e20 (0.27x)	335M	84.1	90.9	79.0	81.8	80.0	83.0
SpanBERT	7.1e20 (1x)	335M	88.8	94.6	85.7	88.7	85.7	88.7
XLNet-Base	6.6e19 (0.09x)	117M	81.3	–	78.5	–	–	–
XLNet	3.9e21 (5.4x)	360M	**89.7**	**95.1**	87.9	**90.6**	87.9	90.7
RoBERTa-100K	6.4e20 (0.90x)	356M	–	94.0	–	87.7	–	–
RoBERTa-500K	3.2e21 (4.5x)	356M	88.9	94.6	86.5	89.4	86.8	89.8
ALBERT	3.1e22 (44x)	235M	89.3	94.8	87.4	90.2	88.1	90.9
BERT (ours)	7.1e20 (1x)	335M	88.0	93.7	84.7	87.5	–	–
ELECTRA-Base	6.4e19 (0.09x)	110M	84.5	90.8	80.5	83.3	–	–
ELECTRA-400K	7.1e20 (1x)	335M	88.7	94.2	86.9	89.6	–	–
ELECTRA-1.75M	3.1e21 (4.4x)	335M	**89.7**	94.9	**88.0**	**90.6**	**88.7**	**91.4**

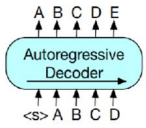
Pre-training with Encoder-Decoder Architectures

- Encoder models: such as BERT, can only work on classification tasks but not generative tasks.
- Decoder models: such as GPT-2, lose bi-directional information, leading to worse performance in classification tasks.
- Encoder-decoder models, such as BART and T5, for both generative and classification tasks, and more flexibility.

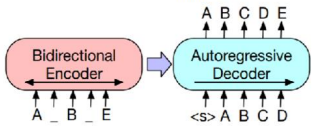
BART: Denoising Sequence-to-Sequence Pre-training



(a) BERT: Random tokens are replaced with masks, and the document is encoded bidirectionally. Missing tokens are predicted independently, so BERT cannot easily be used for generation.



(b) GPT: Tokens are predicted auto-regressively, meaning GPT can be used for generation. However words can only condition on leftward context, so it cannot learn bidirectional interactions.



(c) BART: Inputs to the encoder need not be aligned with decoder outputs, allowing arbitary noise transformations. Here, a document has been corrupted by replacing spans of text with mask symbols. The corrupted document (left) is encoded with a bidirectional model, and then the likelihood of the original document (right) is calculated with an autoregressive decoder. For fine-tuning, an uncorrupted document is input to both the encoder and decoder, and we use representations from the final hidden state of the decoder.

- Corrupting text with an arbitrary noising function
- Learning an encoder-decoder model to reconstruct the original text.

BART: Rich Types of Document Corruption

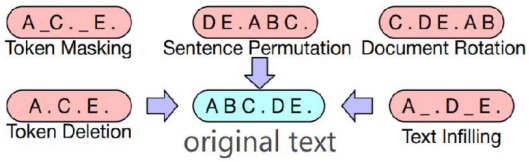
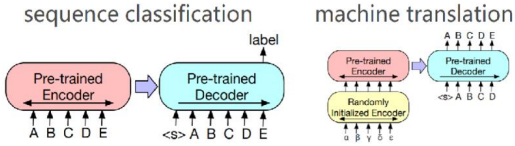


Figure 3: Original Text (Two sentences): Sentence 1: ABC; Sentence 2: DE.

- Token Masking: The same as BERT's MLM.
- Sentence Permutation: Sentences are shuffled in a random order.
- Document Rotation: A token is chosen uniformly at random, and the document is rotated so that it begins with that token.
- Token Deletion: Random tokens are deleted from the input.
- Text Infilling: A number of text spans are sampled. Each span is replaced with a single [MASK].

37 / 53

Fine-tuning BART



- Sequence Classification Tasks: the same input is fed into the encoder and decoder, and the final hidden state of the final decoder token is fed into new multi-class linear classifier.
- Token Classification Tasks (e.g. QA): feed the complete document into the encoder and decoder, and use the top hidden state of the decoder as a representation for each word.
- Sequence Generation Tasks: the encoder input is the input sequence, and the decoder generates outputs autoregressively.
- Machine Translation: replace BARTs encoder embedding layer with a new randomly initialized encoder. Trained end-to-end to map foreign words into an input that BART can de-noise to English.

38 / 53

BART Results

- Results on SQuAD and GLUE:

	SQuAD 1.1 EM/F1	SQuAD 2.0 EM/F1	MNLI m/mm	SST Acc	QQP Acc	QNLI Acc	STS-B Acc	RTE Acc	MRPC Acc	CoLA Mcc
BERT	84.1/90.9	79.0/81.8	86.6/-	93.2	91.3	92.3	90.0	70.4	88.0	60.6
UniLM	-/-	80.5/83.4	87.0/85.9	94.5	-	92.7	-	70.9	-	61.1
XLNet	**89.0**/94.5	86.1/88.8	89.8/-	95.6	91.8	93.9	91.8	83.8	89.2	63.6
RoBERTa	88.9/**94.6**	**86.5/89.4**	**90.2/90.2**	96.4	92.2	94.7	**92.4**	86.6	**90.9**	**68.0**
BART	88.8/**94.6**	86.1/89.2	89.9/90.1	**96.6**	**92.5**	**94.9**	91.2	**87.0**	90.4	62.8

- Results on summarization:

	CNN/DailyMail R1	CNN/DailyMail R2	CNN/DailyMail RL	XSum R1	XSum R2	XSum RL
Lead-3	40.42	17.62	36.67	16.30	1.60	11.95
PTGEN (See et al., 2017)	36.44	15.66	33.42	29.70	9.21	23.24
PTGEN+COV (See et al., 2017)	39.53	17.28	36.38	28.10	8.02	21.72
UniLM	43.33	20.21	40.51	-	-	-
BERTSUMABS (Liu & Lapata, 2019)	41.72	19.39	38.76	38.76	16.33	31.15
BERTSUMEXTABS (Liu & Lapata, 2019)	42.13	19.60	39.18	38.81	16.50	31.27
BART	**44.16**	**21.28**	**40.90**	**45.14**	**22.27**	**37.25**

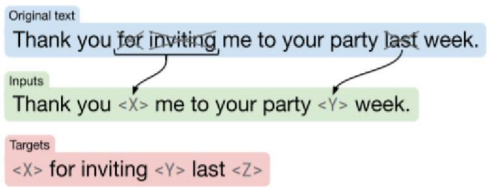
- Results on Machine Translation:

	RO-EN
Baseline	36.80
Fixed BART	36.29
Tuned BART	**37.96**

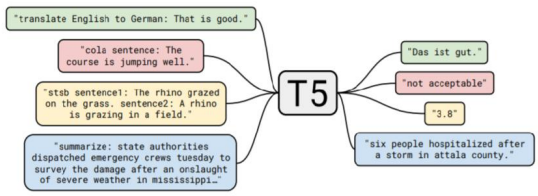
39 / 53

T5

- A similar encoder-decoder framework to BART.
- Unsupervised objective similar to the text infilling in BART.



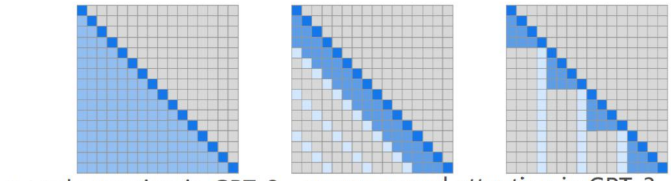
- Add a task-specific (text) prefix to the original input sequence for different down-stream tasks:



40 / 53

GPT-3: Sparse Transformer

- GPT-3 pre-trained on sequences of length (context window) $L = 2048$.
- Sparse Transformer is the architecture used in GPT-3, to reduce the $O(L^2)$ memory of multi-head attention to $O(L\sqrt{L})$.



casual attention in GPT-2 sparse casual attention in GPT-3

(a) Transformer (b) Sparse Transformer (strided) (c) Sparse Transformer (fixed)

GPT-3: Model and Data

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or "GPT-3"	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

Table 2.1: Sizes, architectures, and learning hyper-parameters (batch size in tokens and learning rate) of the models which we trained. All models were trained for a total of 300 billion tokens.

- The hidden dimension in feedforward layer $d_{\text{ff}} = 4 \times d_{\text{model}}$.
- Sequence length (context window) $L = 2048$.

Training data:

Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

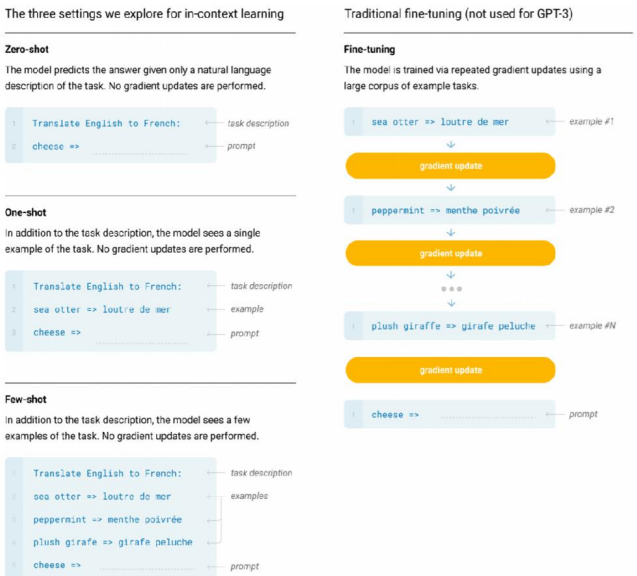
GPT-3: Few-shot Learners

Three evaluation strategies:

- Few-shot: allow as many demonstrations as will fit into the models context window (typically 10 to 100).
- One-shot: allow only one demonstration
- Zero-shot no demonstrations are allowed and only an instruction in natural language is given to the model.

Let's see some examples in machine translation.

GPT-3: Few-shot Learners



GPT-3: Results

In traditional classification tasks, GPT-3 is still not as good as bi-directional baselines, such as QA and SuperGLUE.

Setting	CoQA	DROP	QuAC	SQuADv2	RACE-h	RACE-m
Fine-tuned SOTA	**90.7**[a]	**89.1**[b]	**74.4**[c]	**93.0**[d]	**90.0**[e]	**93.1**[e]
GPT-3 Zero-Shot	81.5	23.6	41.5	59.5	45.5	58.4
GPT-3 One-Shot	84.0	34.3	43.3	65.4	45.9	57.4
GPT-3 Few-Shot	85.0	36.5	44.3	69.8	46.8	58.1

	SuperGLUE Average	BoolQ Accuracy	CB Accuracy	CB F1	COPA Accuracy	RTE Accuracy
Fine-tuned SOTA	**89.0**	**91.0**	**96.9**	**93.9**	**94.8**	**92.5**
Fine-tuned BERT-Large	69.0	77.4	83.6	75.7	70.6	71.7
GPT-3 Few-Shot	71.8	76.4	75.6	52.0	92.0	69.0

	WiC Accuracy	WSC Accuracy	MultiRC Accuracy	MultiRC F1a	ReCoRD Accuracy	ReCoRD F1
Fine-tuned SOTA	**76.1**	**93.8**	**62.3**	**88.2**	**92.5**	**93.3**
Fine-tuned BERT-Large	69.6	64.6	24.1	70.0	71.3	72.0
GPT-3 Few-Shot	49.4	80.1	30.5	75.4	90.2	91.1

GPT-3: Results

Few-shot GPT-3 achieves very good performance in machine translation.

Setting	En→Fr	Fr→En	En→De	De→En	En→Ro	Ro→En
SOTA (Supervised)	**45.6**[a]	35.0 [b]	**41.2**[c]	40.2[d]	**38.5**[e]	**39.9**[e]
XLM [LC19]	33.4	33.3	26.4	34.3	33.3	31.8
MASS [STQ+19]	37.5	34.9	28.3	35.2	35.2	33.1
mBART [LGG+20]	-	-	29.8	34.0	35.0	30.5
GPT-3 Zero-Shot	25.2	21.2	24.6	27.2	14.1	19.9
GPT-3 One-Shot	28.3	33.7	26.2	30.4	20.6	38.6
GPT-3 Few-Shot	32.6	39.2	29.7	40.6	21.0	39.5

GPT-3: Results

Arithmetic Tasks:

- 2 digit addition (2D+) The model is asked to add two integers sampled uniformly from [0, 100), phrased in the form of a question, e.g. "Q: What is 48 plus 76? A: 124."
- 2 digit subtraction (2D-) The model is asked to subtract two integers sampled uniformly from [0, 100); the answer may be negative. Example: "Q: What is 34 minus 53? A: -19".
- 2 digit multiplication (2Dx) The model is asked to multiply two integers sampled uniformly from [0, 100), e.g. "Q: What is 24 times 42? A: 1008".
- One-digit composite (1DC) The model is asked to perform a composite operation on three 1 digit numbers, with parentheses around the last two. For example, "Q: What is $6 + (4 * 8)$? A: 38". The three 1 digit numbers are selected uniformly on [0, 10) and the operations are selected uniformly from $\{+, -, *\}$.
- ...

GPT-3: Results

It is very surprising that GPT-3 can work on simple math problems.

Setting	2D+	2D-	3D+	3D-	4D+	4D-	5D+	5D-	2Dx	1DC
GPT-3 Zero-shot	76.9	58.0	34.2	48.3	4.0	7.5	0.7	0.8	19.8	9.8
GPT-3 One-shot	99.6	86.4	65.5	78.7	14.0	14.0	3.5	3.8	27.4	14.3
GPT-3 Few-shot	100.0	98.9	80.4	94.2	25.5	26.8	9.3	9.9	29.2	21.3

Figure 4: Accuracy (%).

GPT-3: Results

English Grammar Checker

few-shot examples

```
Poor English input:  I eated the purple berries.
Good English output:  I ate the purple berries.
Poor English input:  Thank you for picking me as your designer.  I'd appreciate it.
Good English output:  Thank you for choosing me as your designer.  I appreciate it.
Poor English input:  The mentioned changes have done.  or I did the alteration that you
requested.  or I changed things you wanted and did the modifications.
Good English output:  The requested changes have been made.  or I made the alteration that you
requested.  or I changed things you wanted and made the modifications.
Poor English input:  I'd be more than happy to work with you in another project.
Good English output:  I'd be more than happy to work with you on another project.
```

```
Poor English input:  Please provide me with a short brief of the design you're looking for and
that'd be nice if you could share some examples or project you did before.
Good English output:  Please provide me with a brief description of the design you're
looking for and that would be nice if you could share some examples or projects you have
done before.
```

```
Poor English input:  The patient was died.
Good English output:  The patient died.
```

```
Poor English input:  We think that Leslie likes ourselves.
Good English output:  We think that Leslie likes us.
```

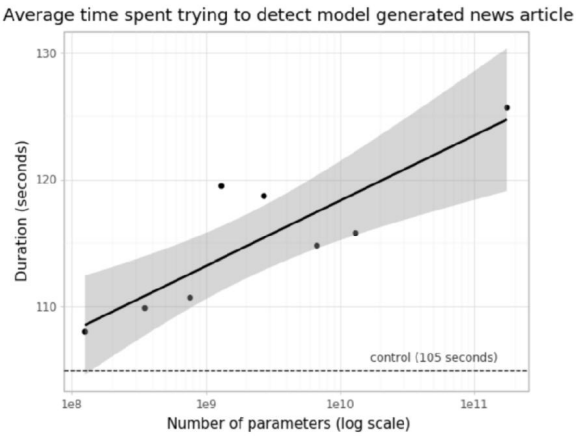
```
Poor English input:  Janet broke Bill on the finger.
Good English output:  Janet broke Bill's finger.
```

```
Poor English input:  Mary arranged for, in St.  Louis, John to rent a house cheap.
Good English output:  Mary arranged for John to rent a house in St.  Louis.
```

```
Poor English input:  Today I have went to the store to to buys some many bottle of water.
Good English output:  Today I went to the store to buy some bottles of water.
```


GPT-3: Results

OpenAI conducts experiments measuring human ability to distinguish GPT-3-generated synthetic news articles from real news articles. The average time spent increases as the model size increases.



Overview

Conclusion

- Background
  - language model
  - sequence to sequence model
- Pre-BERT Era
  - semi-supervised sequence learning
  - context2vec
  - pre-trained seq2seq
  - ELMo
  - OpenAI GPT
- BERT
- Post-BERT Era
  - RoBERTa
  - Transformer-XL and XLNet
  - ERNIE (Tsinghua)
  - ERINE (Baidu)
  - ALBERT
  - ELECTRA
  - Pre-training with encoder-decoder architecture, BART and T5
  - Sparse Transformer and GPT-3

7.Discussions and Future Directions

Discussions and Future Directions

1. Theoretical Foundation
2. Transferring to downstream tasks
3. Transferring across datasets
4. Exploring potential of sampling strategies
5. Early Degeneration for Contrastive Learning

Theoretical Foundation

- Though self-supervised learning has achieved great success, few works investigate the mechanisms behind it. In Part 5, we list several recent works on this topic and show that theoretical analysis is significant to avoid misleading empirical conclusions.

154

Transferring to downstream tasks

- There is an essential gap between pre-training and downstream tasks. Researchers design elaborate pretext tasks to help models learn some critical features of the dataset that can transfer to other jobs, but sometimes this may fail to realize.
- Besides, the process of selecting pretext tasks seems to be too heuristic and tricky without patterns to follow.

155

Transferring across datasets

- This problem is also known as how to learn inductive biases or inductive learning. Traditionally, we split a dataset into the training used for learning the model parameters and the testing part for evaluation.
- An essential prerequisite for this learning paradigm is that data in the real world conforms to the distribution in our dataset. Nevertheless, this assumption frequently fails in experiments.
- Self-supervised representation learning solves part of this problem, especially in the field of natural language processing.

156

Exploring potential of sampling strategies

- Scientists attribute one of the reasons for the success of mutual information-based methods to better sampling strategies. A series of other contrastive methods may also support this conclusion.
- They propose to leverage super large amounts of negative samples and augmented positive samples, whose effects are studied in deep metric learning.

157

Early Degeneration for Contrastive Learning

- Contrastive learning methods such as MoCo and SimCLR are rapidly approaching the performance of supervised learning for computer vision. However, their incredible performances are generally limited to the classification problem.
- Problems above are probably because the contrastive objectives often get trapped into embedding spaces' early degeneration problem.
- We expect that there would be techniques or new paradigms to solve the early degeneration problem while preserving contrastive learning's advantages.

158

8. Reference

159

[1] H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, and M. Marchand. Domain-adversarial neural networks. arXiv preprint arXiv:1412.4446, 2014.

[2] F. Alam, S. Joty, and M. Imran. Domain adaptation with adversarial training and graph embeddings. arXiv preprint arXiv:1805.05151, 2018.

[3] A. A. Alemi, B. Poole, I. Fischer, J. V. Dillon, R. A. Saurous, and K. Murphy. Fixing a broken elbo. arXiv preprint arXiv:1711.00464, 2017.

[4] S. Arora, H. Khandeparkar, M. Khodak, O. Plevrakis, and N. Saunshi. A theoretical analysis of contrastive unsupervised representation learning. arXiv preprint arXiv:1902.09229, 2019.

[5] A. Asai, K. Hashimoto, H. Hajishirzi, R. Socher, and C. Xiong. Learning to retrieve reasoning paths over wikipedia graph for question answering. arXiv preprint arXiv:1911.10470, 2019.

[6] P. Bachman, R. D. Hjelm, and W. Buchwalter. Learning representations by maximizing mutual information across views. In NIPS, pages 15509–15519, 2019.

[7] Y. Bai, H. Ding, S. Bian, T. Chen, Y. Sun, and W. Wang. Simgnn: A neural network approach to fast graph similarity computation. In WSDM, pages 384–392, 2019.

[8] D. H. Ballard. Modular learning in neural networks. In AAAI, pages 279–284, 1987.

[9] D. Bau, J.-Y. Zhu, H. Strobelt, B. Zhou, J. B. Tenenbaum, W. T. Freeman, and A. Torralba. Gan dissection: Visualizing and understanding generative adversarial networks. arXiv preprint arXiv:1811.10597, 2018.

[10] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. Journal of machine learning research, 3(Feb):1137–1155, 2003.

[11] Y. Bengio, N. Leonard, and A. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv preprint arXiv:1308.3432, 2013.

[12] Y. Bengio, P. Y. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks, 5 2:157–66, 1994.

[13] M. Besserve, R. Sun, and B. Scholkopf. Counterfactuals uncover the modular structure of deep generative models. arXiv preprint arXiv:1812.03253, 2018.

[14] Y. Blau and T. Michaeli. Rethinking lossy compression: The ratedistortion-perception tradeoff. arXiv preprint arXiv:1901.07821, 2019.

[15] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 5:135–146, 2017.

[16] A. Brock, J. Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096, 2018.

[17] L. Cai and W. Y. Wang. Kbgan: Adversarial learning for knowledge graph embeddings. arXiv preprint arXiv:1711.04071, 2017.

[18] S. Cao, W. Lu, and Q. Xu. Grarep: Learning graph representations with global structural information. In CIKM ’15, 2015.

[19] S. Cao, W. Lu, and Q. Xu. Deep neural networks for learning graph representations. In AAAI, 2016.

[20] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. In Proceedings of the ECCV (ECCV), pages 132–149, 2018.

[21] H. Chen, B. Perozzi, Y. Hu, and S. Skiena. Harp: Hierarchical representation learning for networks. In AAAI, 2018.

[22] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. arXiv preprint arXiv:2002.05709, 2020.

[23] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton. Big self-supervised models are strong semi-supervised learners. arXiv preprint arXiv:2006.10029, 2020.

[24] T. Chen, Y. Sun, Y. Shi, and L. Hong. On sampling strategies for neural network-based collaborative filtering. In SIGKDD, pages 767–776, 2017.

[25] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In NIPS, pages 2172–2180, 2016.

164

[26] X. Chen, H. Fan, R. Girshick, and K. He. Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297, 2020.

[27] L. Chongxuan, T. Xu, J. Zhu, and B. Zhang. Triple generative adversarial nets. In NIPS, pages 4088–4098, 2017.

[28] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning. Electra: Pretraining text encoders as discriminators rather than generators. arXiv preprint arXiv:2003.10555, 2020.

[29] A. Conneau, G. Lample, M. Ranzato, L. Denoyer, and H. Jegou. Word translation without parallel data. arXiv preprint arXiv:1710.04087, 2017.

[30] Q. Dai, Q. Li, J. Tang, and D. Wang. Adversarial network embedding. In AAAI, 2018.

165

[31] Z. Dai, Z. Yang, Y. Yang, J. G. Carbonell, Q. Le, and R. Salakhutdinov. Transformer-xl: Attentive language models beyond a fixedlength context. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 2978–2988, 2019.

[32] V. R. de Sa. Learning classification with unlabeled data. In NIPS, pages 112–119, 1994.

[33] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In CVPR, pages 248–255. Ieee, 2009.

[34] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pretraining of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, 2019.

[35] M. Ding, J. Tang, and J. Zhang. Semi-supervised learning on graphs with generative adversarial nets. In Proceedings of the 27th ACM CIKM, pages 913–922, 2018.

166

[36] M. Ding, C. Zhou, Q. Chen, H. Yang, and J. Tang. Cognitive graph for multi-hop reading comprehension at scale. arXiv preprint arXiv:1905.05460, 2019.

[37] L. Dinh, D. Krueger, and Y. Bengio. Nice: Non-linear independent components estimation. arXiv preprint arXiv:1410.8516, 2014.

[38] L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using real nvp. arXiv preprint arXiv:1605.08803, 2016.

[39] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In Proceedings of the IEEE ICCV, pages 1422–1430, 2015.

[40] J. Donahue, P. Krahenb ¨ uhl, and T. Darrell. Adversarial feature learning. arXiv preprint arXiv:1605.09782, 2016.20

167

[41] J. Donahue and K. Simonyan. Large scale adversarial representation learning. In NIPS, pages 10541–10551, 2019.

[42] C. Donnat, M. Zitnik, D. Hallac, and J. Leskovec. Learning structural node embeddings via diffusion wavelets. In SIGKDD, pages 1320–1329, 2018.

[43] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville. Adversarially learned inference. arXiv preprint arXiv:1606.00704, 2016.

[44] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. The Journal of Machine Learning Research, 17(1):2096–2030, 2016.

[45] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. arXiv preprint arXiv:1803.07728, 2018.

[46] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, pages 580–587, 2014.

[47] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In NIPS, pages 2672–2680, 2014.

[48] J.-B. Grill, F. Strub, F. Altche, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. arXiv preprint arXiv:2006.07733, 2020.

[49] A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. In SIGKDD, pages 855–864, 2016.

[50] A. Grover, A. Zweig, and S. Ermon. Graphite: Iterative generative modeling of graphs. In ICML, 2018.

[51] M. Gutmann and A. Hyvarinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pages 297–304, 2010.

[52] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang. Realm: Retrieval-augmented language model pre-training. arXiv preprint arXiv:2002.08909, 2020.

[53] W. L. Hamilton, R. Ying, and J. Leskovec. Representation learning on graphs: Methods and applications. IEEE Data Eng. Bull., 40:52–74, 2017.

[54] W. L. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. In NIPS, 2017.

[55] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. arXiv preprint arXiv:1911.05722, 2019.

170

[56] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In CVPR, pages 770–778, 2016.

[57] D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song. Using self-supervised learning can improve model robustness and uncertainty. In NeurIPS, pages 15663–15674, 2019.

[58] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio. Learning deep representations by mutual information estimation and maximization. arXiv preprint arXiv:1808.06670, 2018.

[59] J. Ho, X. Chen, A. Srinivas, Y. Duan, and P. Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In ICML, pages 2722–2730, 2019.

[60] W. Hu, B. Liu, J. Gomes, M. Zitnik, P. Liang, V. Pande, and J. Leskovec. Strategies for pre-training graph neural networks. In ICLR, 2019.

171

[61] Z. Hu, Y. Dong, K. Wang, and Y. Sun. Heterogeneous graph transformer. arXiv preprint arXiv:2003.01332, 2020.

[62] G. Huang, Z. Liu, and K. Q. Weinberger. Densely connected convolutional networks. 2017 IEEE CVPR, pages 2261–2269, 2017.

[63] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and locally consistent image completion. ACM Transactions on Graphics (ToG), 36(4):1–14, 2017.

[64] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. ArXiv, abs/1502.03167, 2015.

[65] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In CVPR, pages 1125–1134, 2017.

[66] L. Jing and Y. Tian. Self-supervised visual feature learning with deep neural networks: A survey. arXiv preprint arXiv:1902.06162, 2019.

[67] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy. Spanbert: Improving pre-training by representing and predicting spans. Transactions of the Association for Computational Linguistics, 8:64–77, 2020.

[68] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In CVPR, pages 4401–4410, 2019.

[69] D. Kim, D. Cho, D. Yoo, and I. S. Kweon. Learning image representations by completing damaged jigsaw puzzles. In 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pages 793–802. IEEE, 2018.

[70] D. P. Kingma and P. Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In NIPS, pages 10215–10224, 2018.

[71] D. P. Kingma and M. Welling. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013.

[72] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907, 2016.

[73] T. N. Kipf and M. Welling. Variational graph auto-encoders. arXiv preprint arXiv:1611.07308, 2016.

[74] L. Kong, C. d. M. d'Autume, W. Ling, L. Yu, Z. Dai, and D. Yogatama. A mutual information maximization perspective of language representation learning. arXiv preprint arXiv:1910.08350, 2019.

[75] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, pages 1097–1105, 2012.

174

[76] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942, 2019.

[77] G. Larsson, M. Maire, and G. Shakhnarovich. Learning representations for automatic colorization. In ECCV, pages 577–593. Springer, 2016.

[78] G. Larsson, M. Maire, and G. Shakhnarovich. Colorization as a proxy task for visual understanding. In CVPR, pages 6874–6883, 2017.

[79] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. nature, 521(7553):436–444, 2015.

[80] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. Neural Comput., 1(4):541–551, Dec. 1989.

175

[81] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Muller. Efficient backprop. In Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop, page 9–50, Berlin, Heidelberg, 1998. Springer-Verlag.

[82] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photorealistic single image super resolution using a generative adversarial network. In CVPR, pages 4681–4690, 2017.

[83] D. Li, W.-C. Hung, J.-B. Huang, S. Wang, N. Ahuja, and M.-H. Yang. Unsupervised visual representation learning by graphbased consistent constraints. In ECCV, pages 678–694. Springer, 2016.

[84] R. Li, S. Wang, F. Zhu, and J. Huang. Adaptive graph convolutional neural networks. ArXiv, abs/1801.03226, 2018.

[85] B. Liu. Sentiment analysis and opinion mining. Synthesis lectures on human language technologies, 5(1):1–167, 2012.

[86] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692, 2019.

[87] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In CVPR, pages 3431–3440, 2015.

[88] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. Adversarial autoencoders. arXiv preprint arXiv:1511.05644, 2015.

[89] M. Mathieu. Masked autoencoder for distribution estimation. 2015.

[90] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean. Efficient estimation of word representations in vector space. CoRR, abs/1301.3781, 2013.

[91] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In NIPS'13, pages 3111–3119, 2013.

[92] I. Misra and L. van der Maaten. Self-supervised learning of pretext-invariant representations. arXiv preprint arXiv:1912.01991, 2019.21

[93] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In ICML, 2010.

[94] A. Newell and J. Deng. How useful is self-supervised pretraining for visual tasks? In CVPR, pages 7345–7354, 2020.

[95] A. Ng et al. Sparse autoencoder. CS294A Lecture notes, 72(2011):1–19, 2011.

178

[96] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In ECCV, pages 69–84. Springer, 2016.

[97] M. Noroozi, A. Vinjimoor, P. Favaro, and H. Pirsiavash. Boosting self-supervised learning via knowledge transfer. In CVPR, pages 9359–9367, 2018.

[98] S. Nowozin, B. Cseke, and R. Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In NIPS, pages 271–279, 2016.

[99] A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748, 2018.

[100] M. Ou, P. Cui, J. Pei, Z. Zhang, and W. Zhu. Asymmetric transitivity preserving graph embedding. In KDD '16, 2016.

179

[101] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In CVPR, pages 2536–2544, 2016.

[102] Z. Peng, Y. Dong, M. Luo, X. ming Wu, and Q. Zheng. Selfsupervised graph representation learning via global context prediction. ArXiv, abs/2003.01604, 2020.

[103] Z. Peng, Y. Dong, M. Luo, X.-M. Wu, and Q. Zheng. Selfsupervised graph representation learning via global context prediction. arXiv preprint arXiv:2003.01604, 2020.

[104] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In SIGKDD, pages 701–710, 2014.

[105] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. arXiv preprint arXiv:1802.05365, 2018.

180

[106] M. Popova, M. Shvets, J. Oliva, and O. Isayev. Molecularrnn: Generating realistic molecular graphs with optimized properties. arXiv preprint arXiv:1905.13372, 2019.

[107] J. Qiu, Q. Chen, Y. Dong, J. Zhang, H. Yang, M. Ding, K. Wang, and J. Tang. Gcc: Graph contrastive coding for graph neural network pre-training. arXiv preprint arXiv:2006.09963, 2020.

[108] J. Qiu, Y. Dong, H. Ma, J. Li, K. Wang, and J. Tang. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In WSDM '18, 2018.

[109] J. Qiu, Y. Dong, H. Ma, J. Li, K. Wang, and J. Tang. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In WSDM, pages 459–467, 2018.

[110] J. Qiu, J. Tang, H. Ma, Y. Dong, K. Wang, and J. Tang. Deepinf: Social influence prediction with deep learning. In KDD'18, pages 2110–2119. ACM, 2018.

181

[111] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang. Pre-trained models for natural language processing: A survey. arXiv preprint arXiv:2003.08271, 2020.

[112] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434, 2015.

[113] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training.

[114] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. OpenAI Blog, 1(8):9, 2019.

[115] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. Squad: 100,000+ questions for machine comprehension of text. arXiv preprint arXiv:1606.05250, 2016.

[116] A. Razavi, A. van den Oord, and O. Vinyals. Generating diverse high-fidelity images with vq-vae-2. In NIPS, pages 14837–14847, 2019.

[117] L. F. Ribeiro, P. H. Saverese, and D. R. Figueiredo. struc2vec: Learning node representations from structural identity. In SIGKDD, pages 385–394, 2017.

[118] M. T. Ribeiro, S. Singh, and C. Guestrin. " why should i trust you?" explaining the predictions of any classifier. In SIGKDD, pages 1135–1144, 2016.

[119] N. Sarafianos, X. Xu, and I. A. Kakadiaris. Adversarial representation learning for text-to-image matching. In Proceedings of the IEEE ICCV, pages 5814–5824, 2019.

[120] A. M. Saxe, J. L. McClelland, and S. Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. CoRR, abs/1312.6120, 2013.

[121] J. Shen, Y. Qu, W. Zhang, and Y. Yu. Adversarial representation learning for domain adaptation. stat, 1050:5, 2017.

[122] T. Shen, T. Lei, R. Barzilay, and T. Jaakkola. Style transfer from non-parallel text by cross-alignment. In NIPS, pages 6830–6841, 2017.

[123] C. Shi, M. Xu, Z. Zhu, W. Zhang, M. Zhang, and J. Tang. Graphaf: a flow-based autoregressive model for molecular graph generation. arXiv preprint arXiv:2001.09382, 2020.

[124] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[125] A. Sinha, Z. Shen, Y. Song, H. Ma, D. Eide, B.-j. P. Hsu, and K. Wang. An overview of microsoft academic service (mas) and applications. In WWW'15, pages 243–246, 2015.

[126] P. Smolensky. Information processing in dynamical systems: Foundations of harmony theory. Technical report, Colorado Univ at Boulder Dept of Computer Science, 1986.

[127] F.-Y. Sun, J. Hoffmann, and J. Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. arXiv preprint arXiv:1908.01000, 2019.

[128] F.-Y. Sun, M. Qu, J. Hoffmann, C.-W. Huang, and J. Tang. vgraph: A generative model for joint community detection and node representation learning. In NIPS, pages 512–522, 2019.

[129] K. Sun, Z. Zhu, and Z. Lin. Multi-stage self-supervised learning for graph convolutional networks. ArXiv, abs/1902.11038, 2020.

[130] Y. Sun, S. Wang, Y. Li, S. Feng, X. Chen, H. Zhang, X. Tian, D. Zhu, H. Tian, and H. Wu. Ernie: Enhanced representation through knowledge integration. arXiv preprint arXiv:1904.09223, 2019.

[131] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. 2015 IEEE CVPR, pages 1–9, 2015.

[132] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. Line: Large-scale information network embedding. In WWW'15, pages 1067–1077, 2015.

[133] W. L. Taylor. "cloze procedure": A new tool for measuring readability. Journalism quarterly, 30(4):415–433, 1953.

[134] Y. Tian, D. Krishnan, and P. Isola. Contrastive multiview coding. arXiv preprint arXiv:1906.05849, 2019.

[135] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola. What makes for good views for contrastive learning. arXiv preprint arXiv:2005.10243, 2020.

186

[136] M. Tschannen, O. Bachem, and M. Lucic. Recent advances in autoencoder-based representation learning. arXiv preprint arXiv:1812.05069, 2018.

[137] M. Tschannen, J. Djolonga, P. K. Rubenstein, S. Gelly, and M. Lucic. On mutual information maximization for representation learning. arXiv preprint arXiv:1907.13625, 2019.

[138] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. In 9th ISCA Speech Synthesis Workshop, pages 125–125.

[139] A. Van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, et al. Conditional image generation with pixelcnn decoders. In NIPS, pages 4790–4798, 2016.

[140] A. van den Oord, O. Vinyals, et al. Neural discrete representation learning. In NIPS, pages 6306–6315, 2017.

187

[141] A. Van Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. In ICML, pages 1747–1756, 2016.

[142] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In NIPS, pages 5998–6008, 2017.

[143] P. Velickovi ˇ c, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. Graph attention networks. arXiv preprint arXiv:1710.10903, 2017.

[144] P. Velickovi ˇ c, W. Fedus, W. L. Hamilton, P. Li ´ o, Y. Bengio, and R. D. Hjelm. Deep graph infomax. arXiv preprint arXiv:1809.10341, 2018.

[145] H. Wang, J. Wang, J. Wang, M. Zhao, W. Zhang, F. Zhang, X. Xie, and M. Guo. Graphgan: Graph representation learning with generative adversarial nets. In AAAI, 2018.

188

[146] P. Wang, S. Li, and R. Pan. Incorporating gan for negative sampling in knowledge representation learning. In AAAI, 2018.22

[147] T. Wang and P. Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. arXiv preprint arXiv:2005.10242, 2020.

[148] Z. Wang, Q. She, and T. E. Ward. Generative adversarial networks: A survey and taxonomy. arXiv preprint arXiv:1906.01529, 2019.

[149] C. Wei, L. Xie, X. Ren, Y. Xia, C. Su, J. Liu, Q. Tian, and A. L. Yuille. Iterative reorganization with weak spatial constraints: Solving arbitrary jigsaw puzzles for unsupervised representation learning. In CVPR, pages 1910–1919, 2019.

[150] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin. Unsupervised feature learning via non-parametric instance discrimination. In CVPR, pages 3733–3742, 2018.

189

[151] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le. Self-training with noisy student improves imagenet classification. In CVPR, pages 10687–10698, 2020.

[152] S. Xie, R. B. Girshick, P. Dollar, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. 2017 IEEE CVPR, pages 5987–5995, 2017.

[153] W. Xiong, J. Du, W. Y. Wang, and V. Stoyanov. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. arXiv preprint arXiv:1912.09637, 2019.

[154] X. Yan, I. Misra, A. Gupta, D. Ghadiyaram, and D. Mahajan. Clusterfit: Improving generalization of visual representations. arXiv preprint arXiv:1912.03330, 2019.

[155] J. Yang, D. Parikh, and D. Batra. Joint unsupervised learning of deep representations and image clusters. In CVPR, pages 5147–5156, 2016.

190

[156] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In NIPS, pages 5754–5764, 2019.

[157] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. arXiv preprint arXiv:1809.09600, 2018.

[158] J. You, B. Liu, Z. Ying, V. Pande, and J. Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. In NIPS, pages 6410–6421, 2018.

[159] J. You, R. Ying, X. Ren, W. Hamilton, and J. Leskovec. Graphrnn: Generating realistic graphs with deep auto-regressive models. In ICML, pages 5708–5717, 2018.

[160] S. Zagoruyko and N. Komodakis. Wide residual networks. ArXiv, abs/1605.07146, 2016.

191

[161] J. Zhang, Y. Dong, Y. Wang, J. Tang, and M. Ding. Prone: fast and scalable network representation learning. In IJCAI, pages 4278–4284, 2019.

[162] M. Zhang, Z. Cui, M. Neumann, and Y. Chen. An end-to-end deep learning architecture for graph classification. In AAAI, 2018.

[163] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In ECCV, pages 649–666. Springer, 2016.

[164] R. Zhang, P. Isola, and A. A. Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In CVPR, pages 1058–1067, 2017.

[165] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu. Ernie: Enhanced language representation with informative entities. arXiv preprint arXiv:1905.07129, 2019.

192

[166] D. Zhu, P. Cui, D. Wang, and W. Zhu. Deep variational network embedding in wasserstein space. In SIGKDD, pages 2827–2836, 2018.

[167] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman. Toward multimodal image-to-image translation. In NIPS, pages 465–476, 2017.

[168] C. Zhuang, A. L. Zhai, and D. Yamins. Local aggregation for unsupervised learning of visual embeddings. In Proceedings of the IEEE ICCV, pages 6002–6012, 2019.

[169] B. Zoph, G. Ghiasi, T.-Y. Lin, Y. Cui, H. Liu, E. D. Cubuk, and Q. V. Le. Rethinking pre-training and self-training. arXiv preprint arXiv:2006.06882, 2020.

[170] B. Zoph and Q. V. Le. Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578, 2016.

193

Thank you !

Jie Tang, KEG, Tsinghua U
http://keg.cs.tsinghua.edu.cn/jietang
Download all data & Codes
https://keg.cs.tsinghua.edu.cn/cogdl/

194