Applied Deep Learning

Beyond Supervised Learning

May 23rd, 2022 http://adl.miulab.tw

National Taiwan University
國立臺灣大學

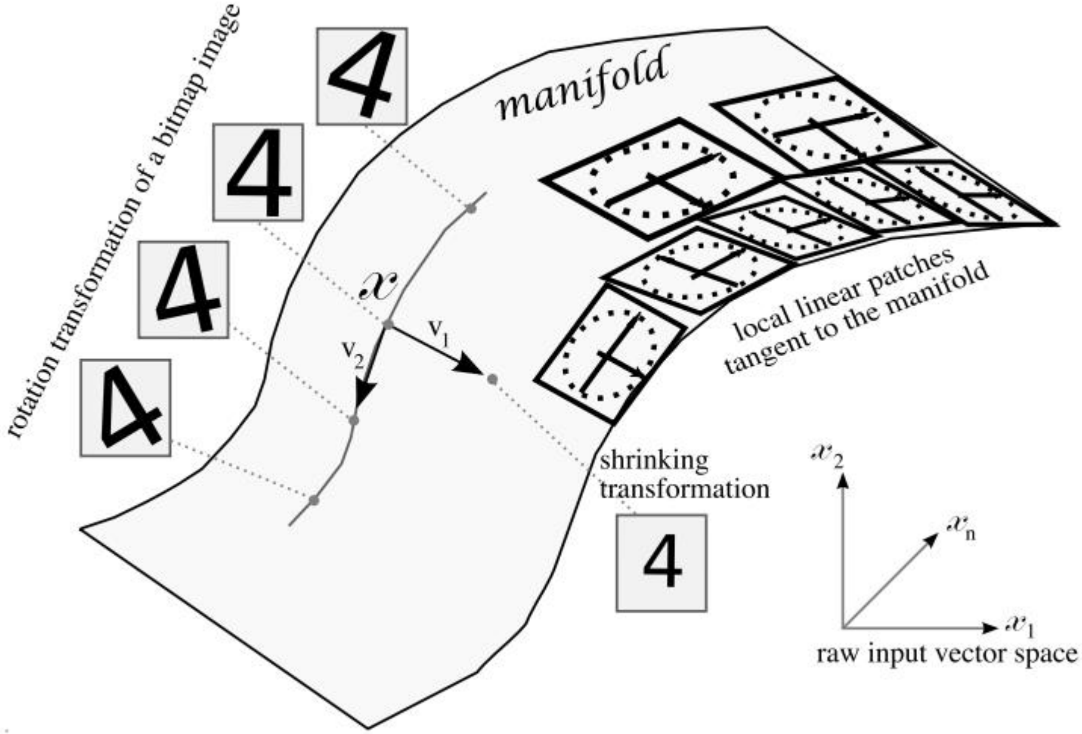
Introduction

- Big data ≠ Big annotated data
- Machine learning techniques include:
  - Supervised learning (if we have labelled data)
  - Reinforcement learning (if we have an environment for reward)
  - Unsupervised learning (if we do not have labelled data)

Why does unlabeled and unrelated data help the tasks?

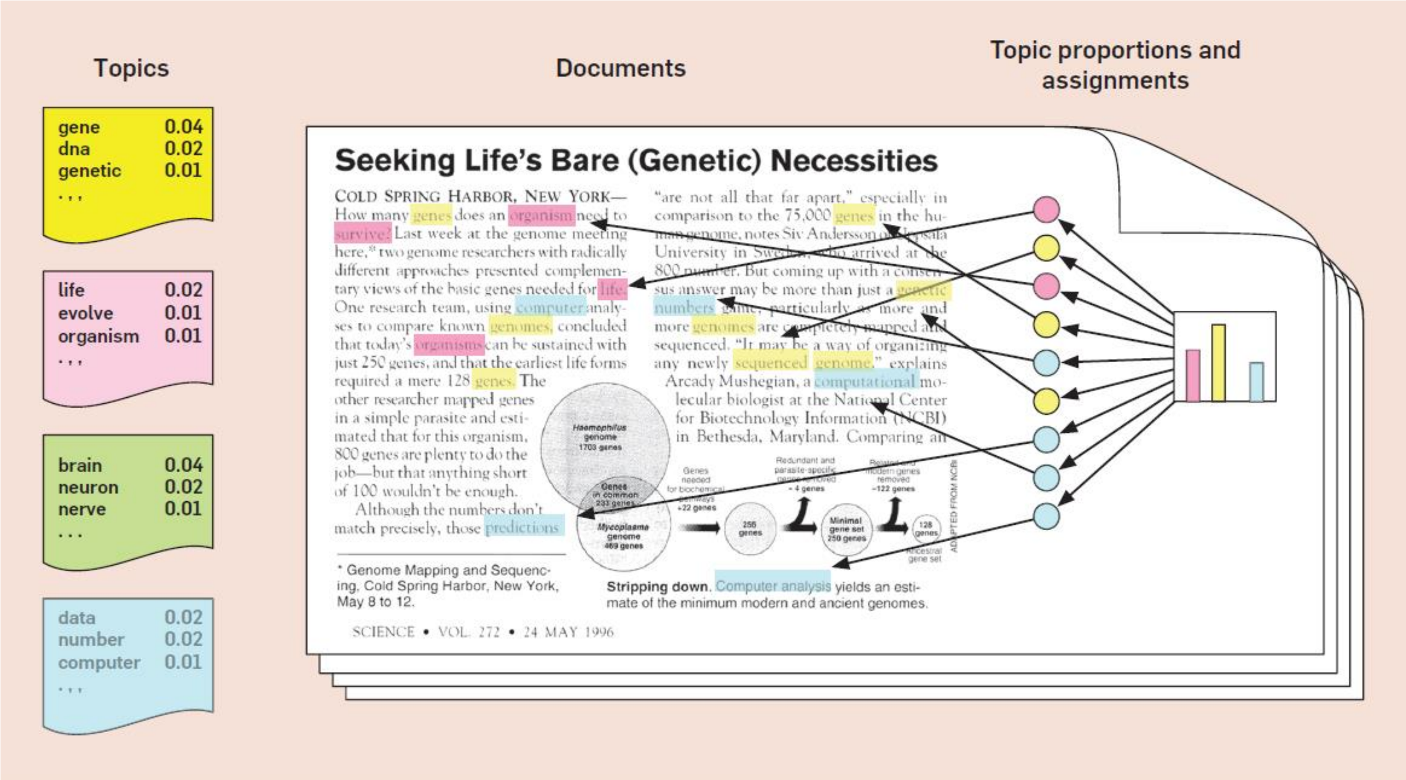
Finding latent factors that control the observations

Latent Factors for Handwritten Digits



4

Latent Factors for Documents



Latent Factors for Recommendation System

					
單純呆	A	✓	✓		
傲嬌	B			✓	✓
	C	✓	? ✓		

Latent Factor Exploitation

- Handwritten digits

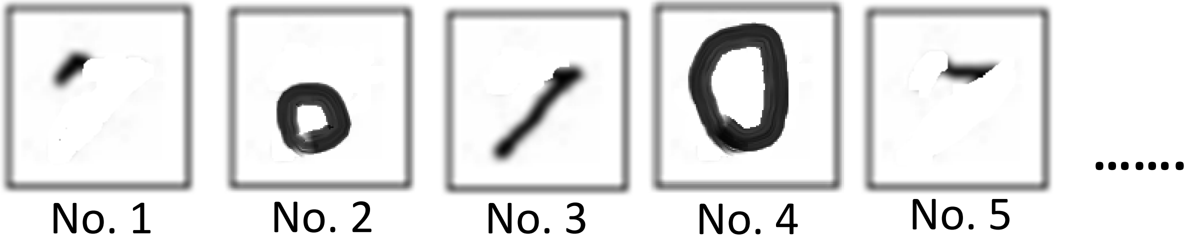
The handwritten images are composed of **strokes**

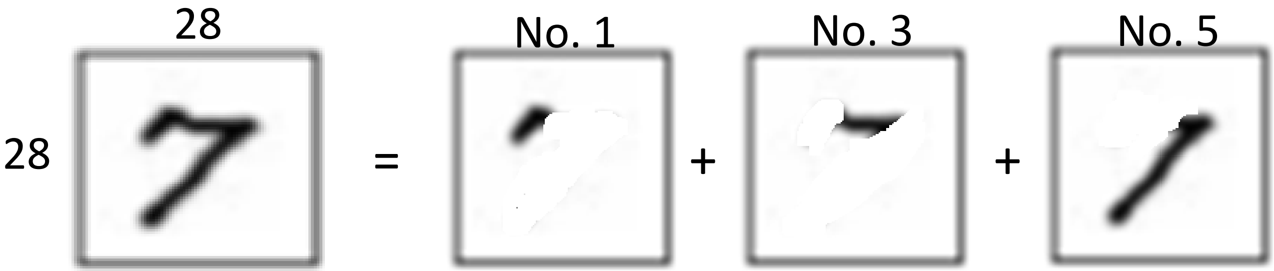
<u>Strokes (Latent Factors)</u>

No. 1　No. 2　No. 3　No. 4　No. 5　.......

Latent Factor Exploitation

Strokes (Latent Factors)





Represented by
28 X 28 = 784 pixels

[1 0 1 0 1 0 ....]
(simpler representation)

Discriminative v.s. Generative

- **Discriminative**: calculate the probability of output given input $P(Y|X)$
- **Generative**: calculate the probability of a variable $P(X)$, or multiple variables $P(X, Y)$

Variable Types

- Observed vs. Latent:
  - Observed: something we can see from our data, e.g. X or Y
  - Latent: a variable that we assume exists without a given value
- Deterministic vs. Random:
  - Deterministic: variables calculated directly via deterministic functions
  - Random (stochastic): variables obeying a probability distribution
- A latent variable model is a probability distribution over two sets of variables

$$p(\underbrace{x}_{\text{Observed}}, \underbrace{z}_{\text{Latent}}; \theta)$$

Latent Variable Types $p(x, z; \theta)$

Latent

- Latent continuous vector
  - Auto-encoder
  - Variational auto-encoder
- Latent discrete vector
  - Topic model
- Latent structure
  - HMM
  - Tree-structured model

11

Auto-Encoder

Representation Learning

Auto-Encoder

- An observed output x
- A latent variable z
- A function (network) f parameterized by θ maps from z to x

$$\boxed{x} = f(\boxed{z}; \theta)$$

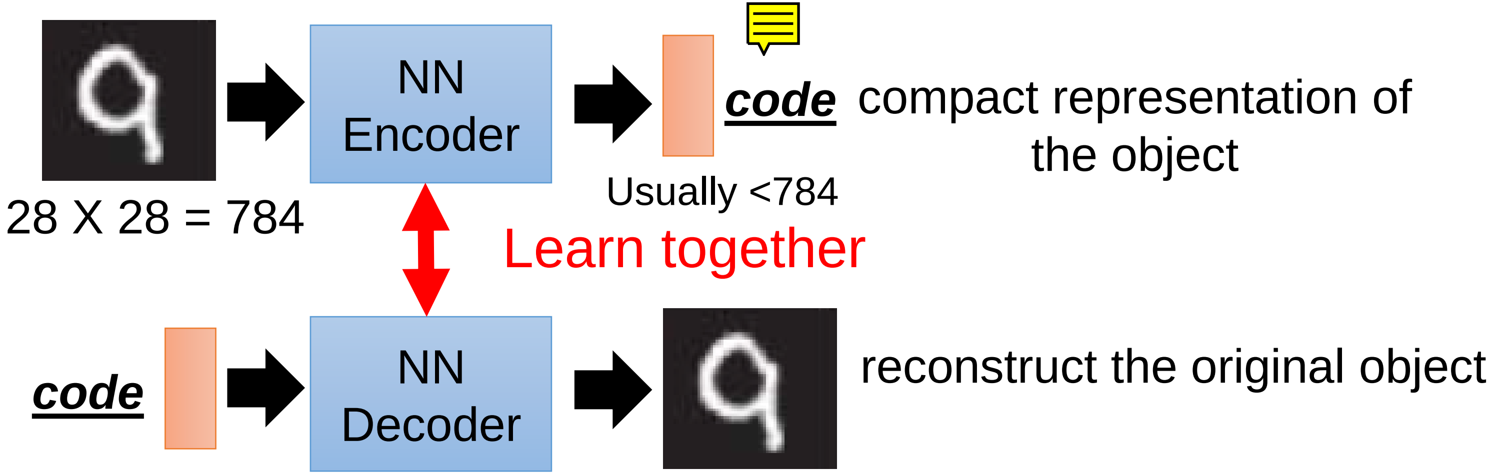
Observed Latent

Idea: represent the output in a more compact way (latent codes)

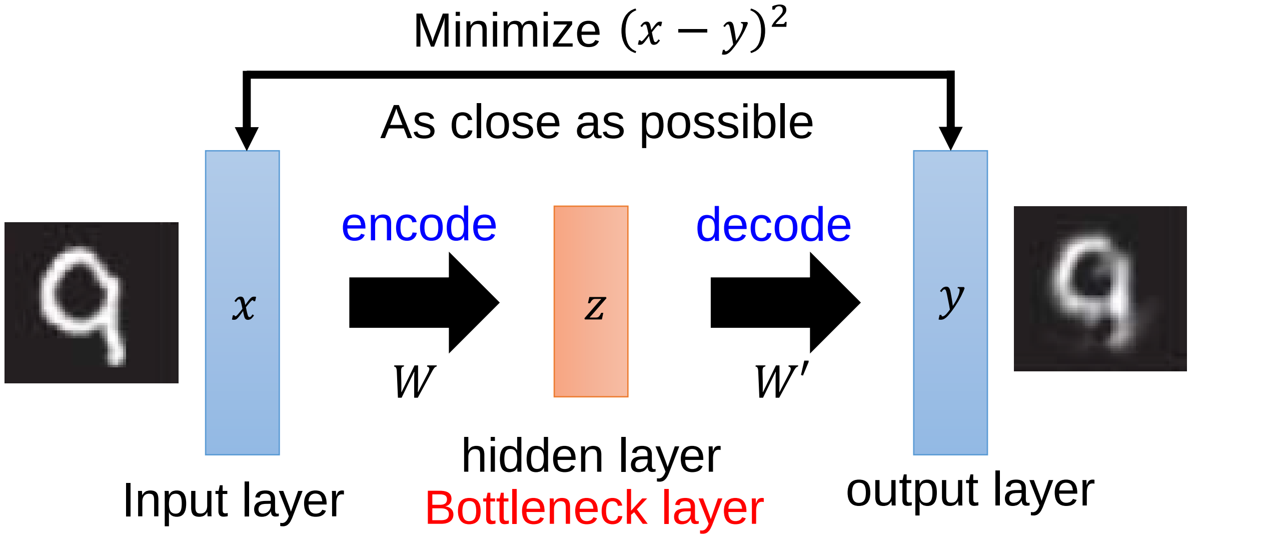
Auto-Encoder

- Represent a digit using 28 X 28 dimensions
- Not all 28 X 28 images are digits

Idea: represent the images of digits in a more compact way



Auto-Encoder

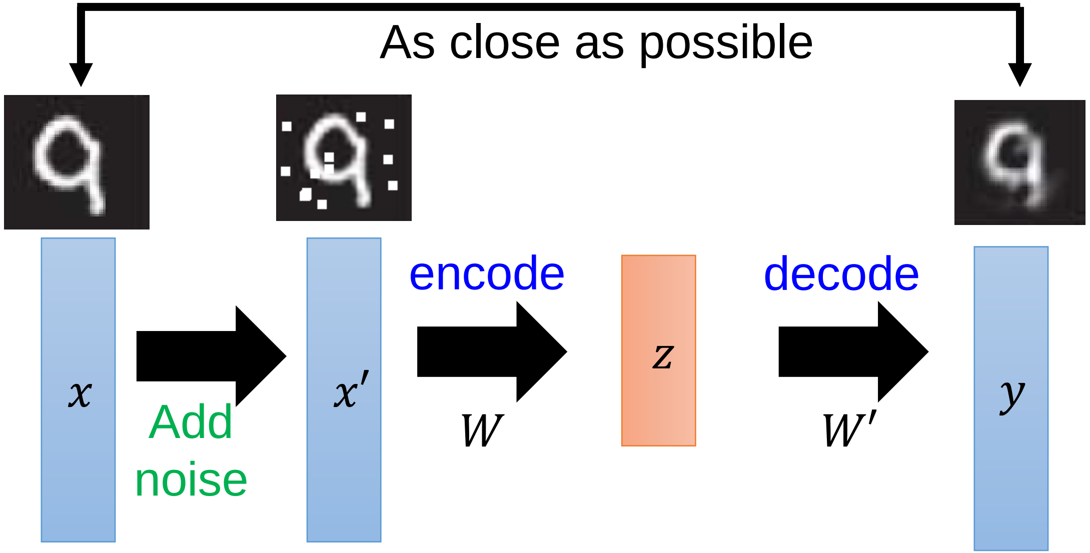


$z = \sigma(Wx + b)$ $\quad$ $y = \sigma(W'z + b')$

Output of the hidden layer is the code

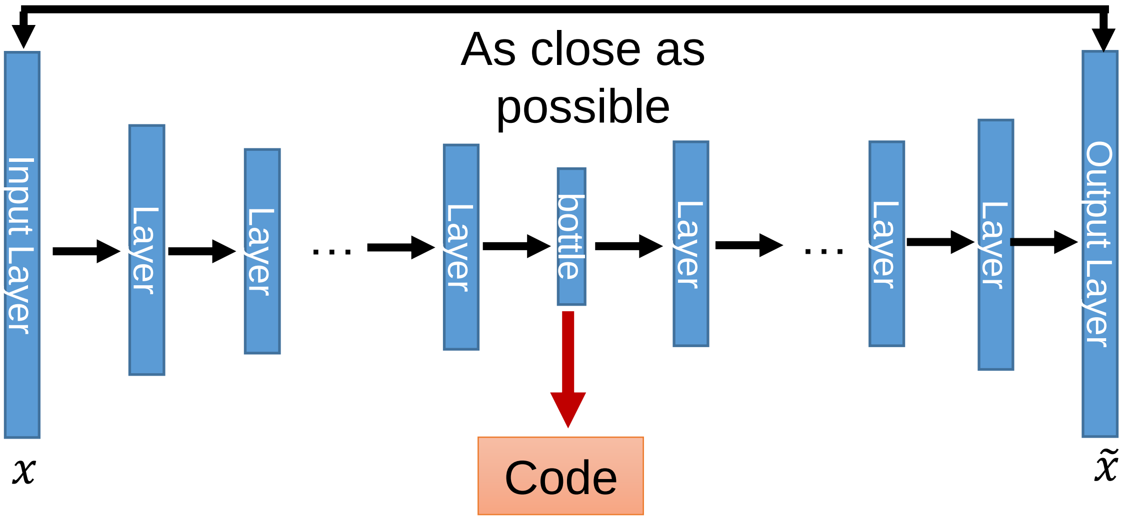
Denoising Auto-Encoder

- Improve robustness of a latent variable



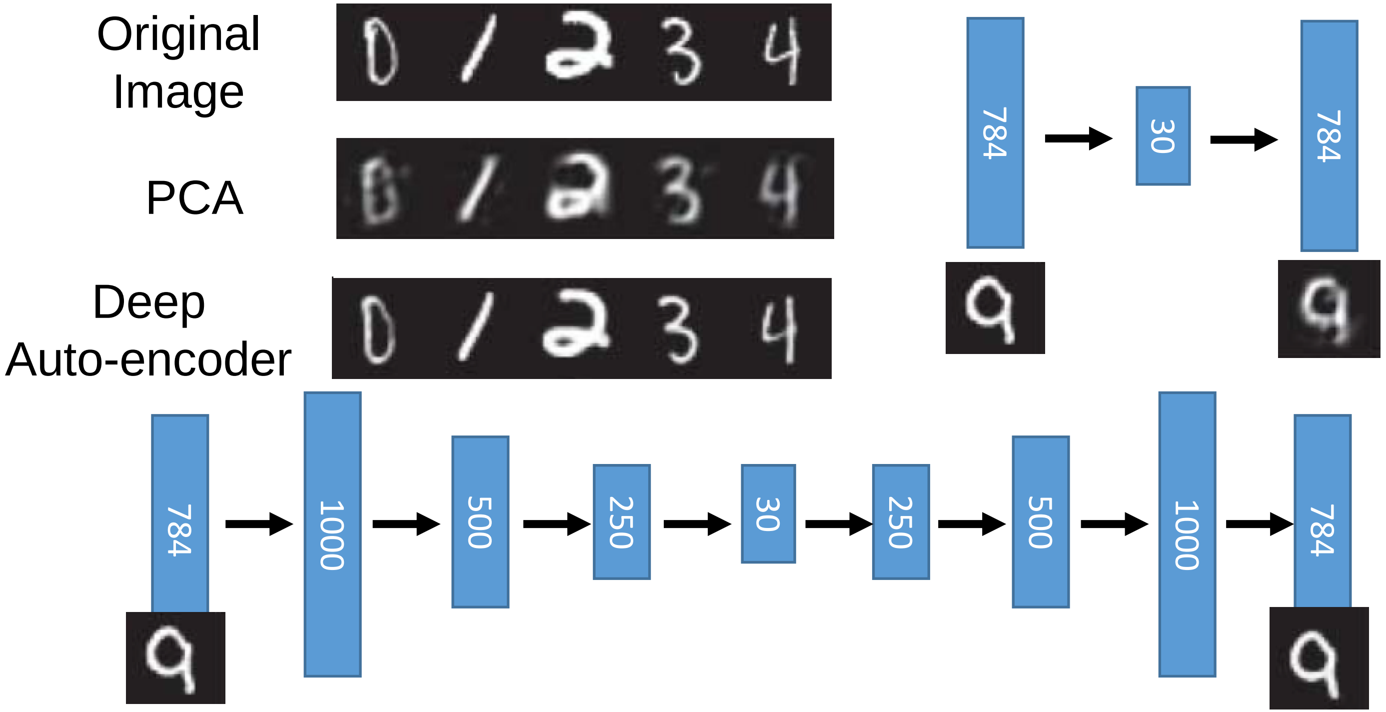
Rifai, et al. "Contractive auto-encoders: Explicit invariance during feature extraction," in *ICML*, 2011.

Deep Auto-Encoder

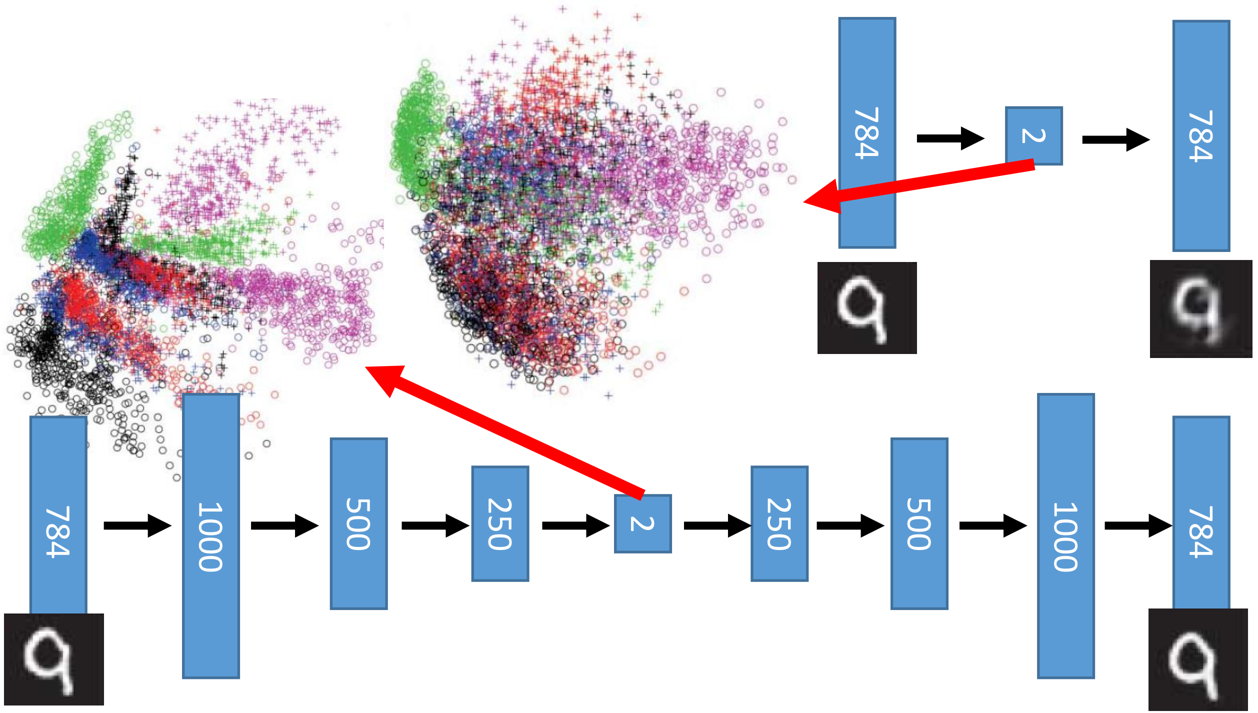


Hinton and Salakhutdinov. “Reducing the dimensionality of data with neural networks,” *Science*, 2006.

Deep Auto-Encoder



Feature Representation

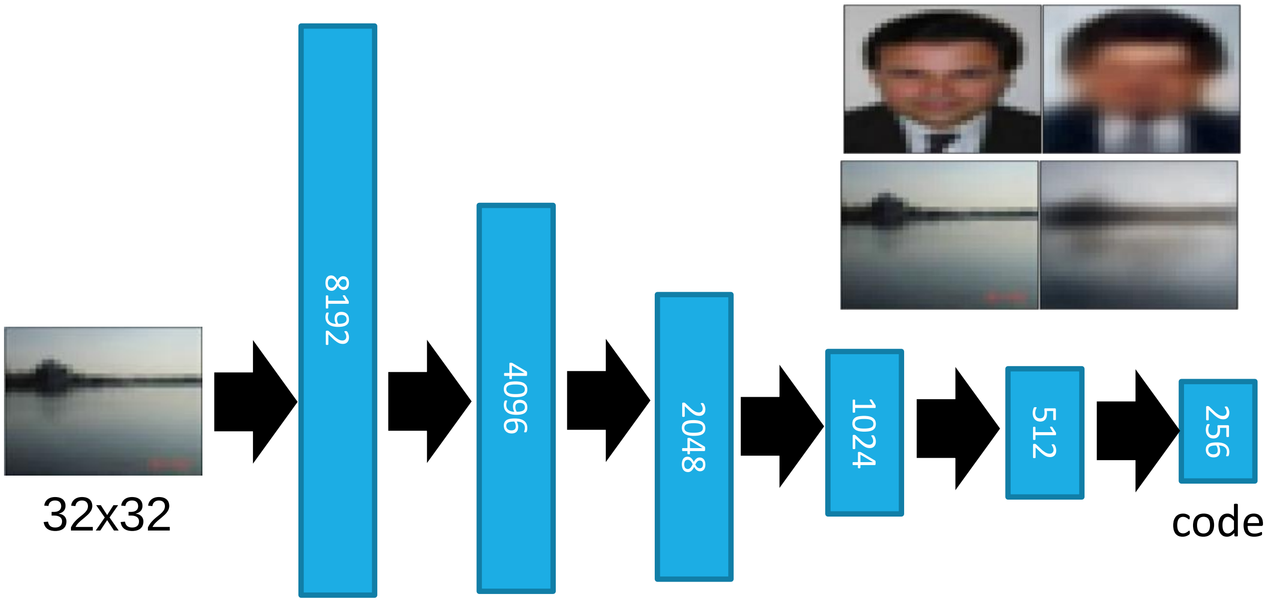


Auto-Encoder – Similar Image Retrieval

- Retrieved using Euclidean distance in pixel intensity space

Krizhevsky et al. "Using very deep autoencoders for content-based image retrieval," in *ESANN*, 2011.

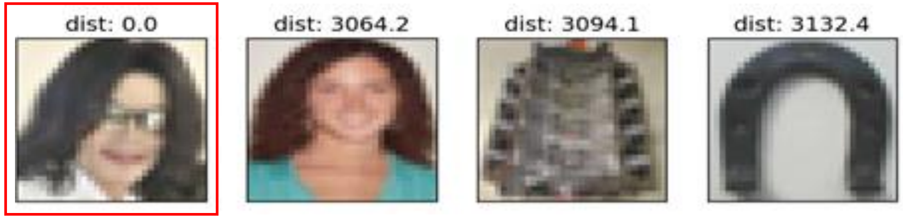
Auto-Encoder – Similar Image Retrieval



(crawl millions of images from the Internet)

Auto-Encoder – Similar Image Retrieval

- Images retrieved using Euclidean distance in pixel intensity space

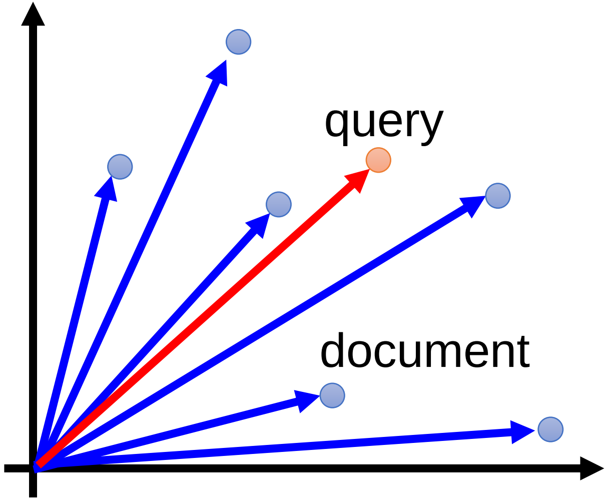


- Images retrieved using 256 codes

Learning the useful latent factors

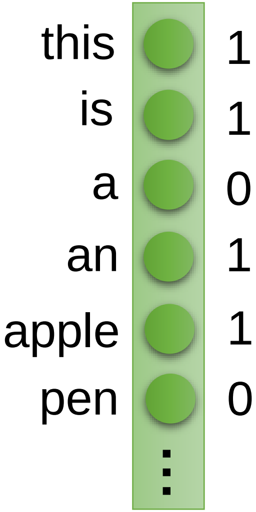
Auto-Encoder – Text Retrieval

<u>Vector Space Model</u>



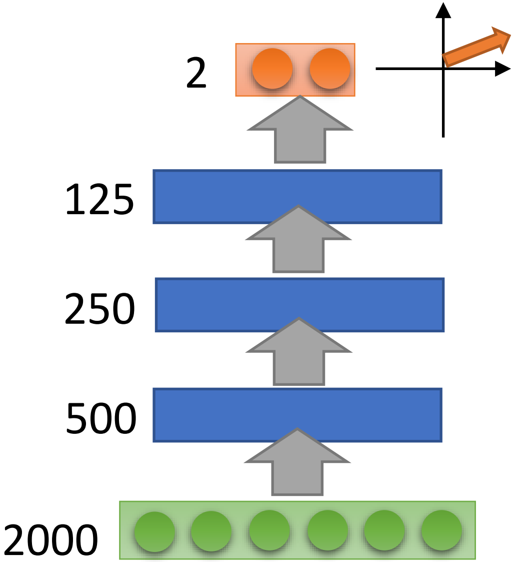
<u>Bag-of-word</u>

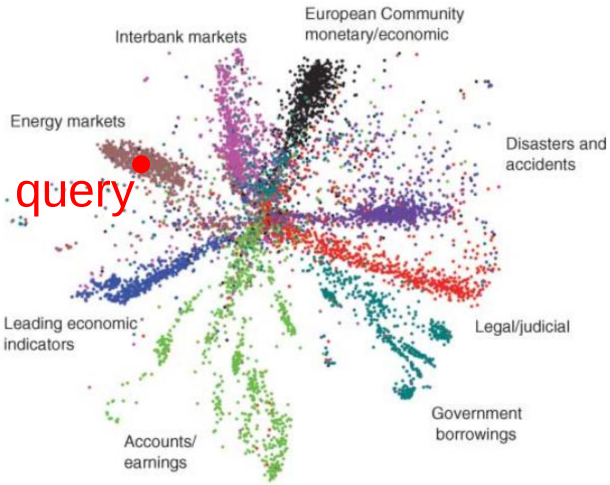
word string:
"This is an apple"



Semantics are not considered

Auto-Encoder – Text Retrieval





Bag-of-word (document or query)

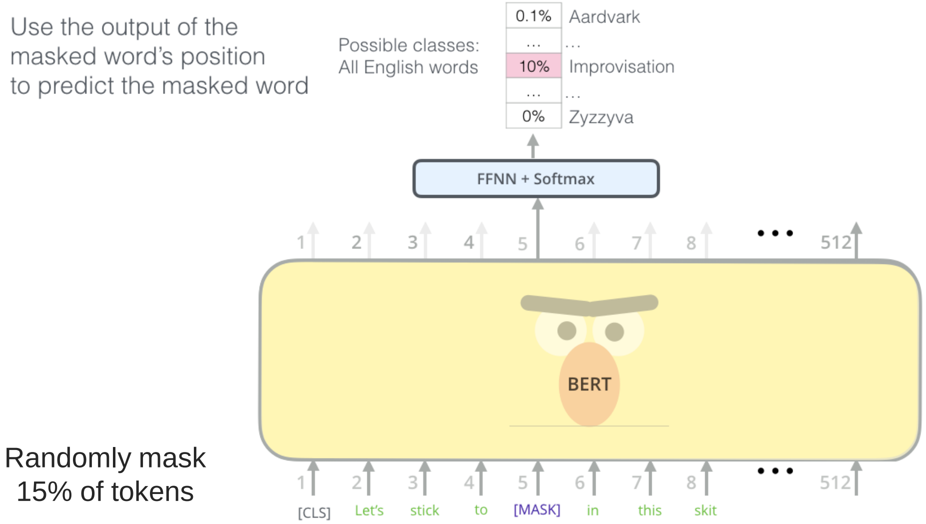
The documents talking about the same thing will have close code

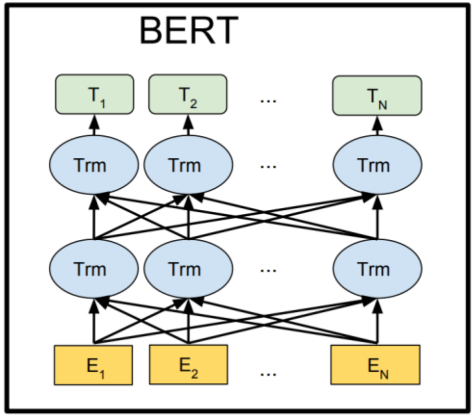
Denoising Auto-Encoding

- Objective: reconstructing $\bar{x}$ from $\hat{x}$

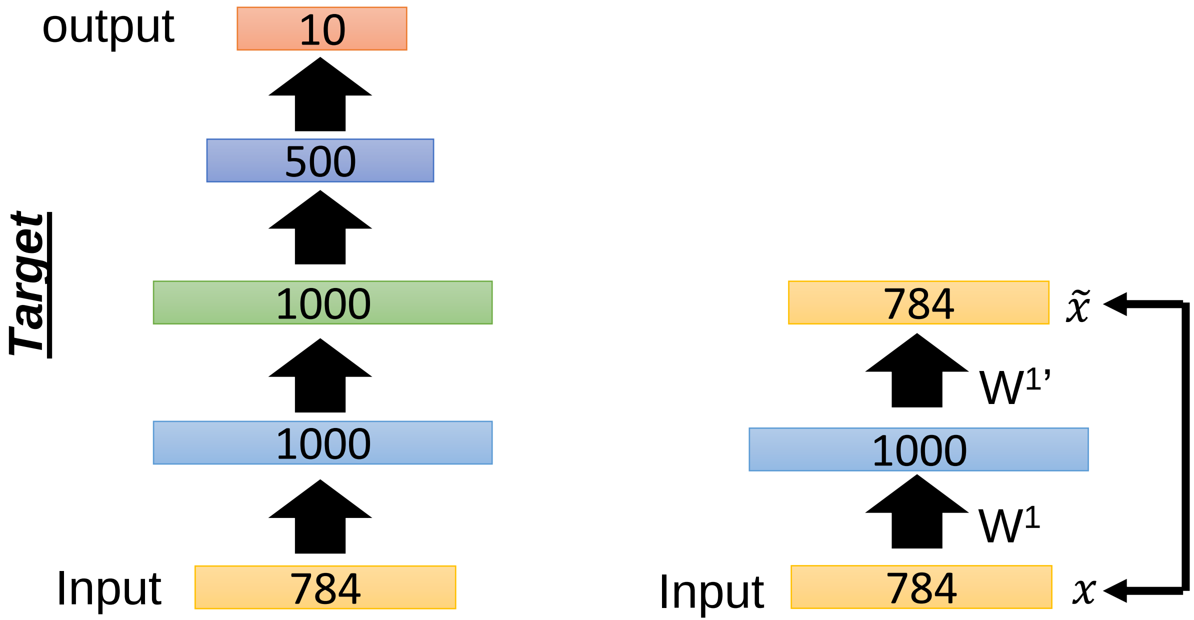
$$\max_{\theta} \quad \log p_{\theta}(\bar{\mathbf{x}} \mid \hat{\mathbf{x}}) \approx \sum_{t=1}^{T} m_t \log p_{\theta}(x_t \mid \hat{\mathbf{x}}) = \sum_{t=1}^{T} m_t \log \frac{\exp\left(H_{\theta}(\hat{\mathbf{x}})_t^{\top} e(x_t)\right)}{\sum_{x'} \exp\left(H_{\theta}(\hat{\mathbf{x}})_t^{\top}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

  - dimension reduction or denoising (masked LM)

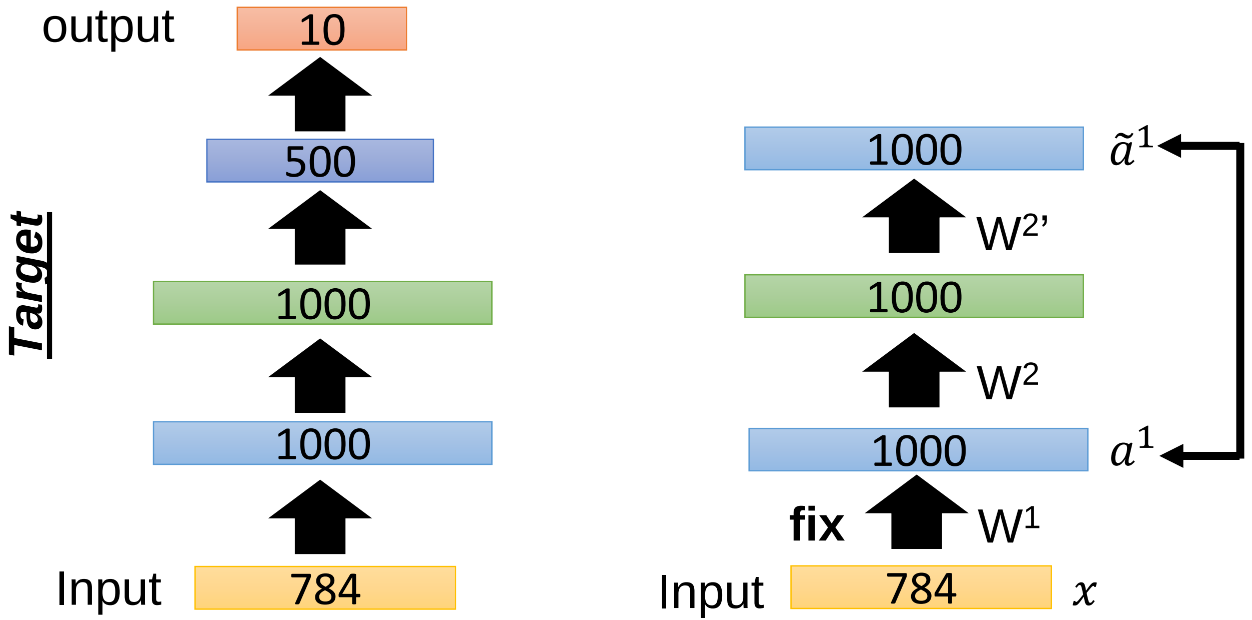




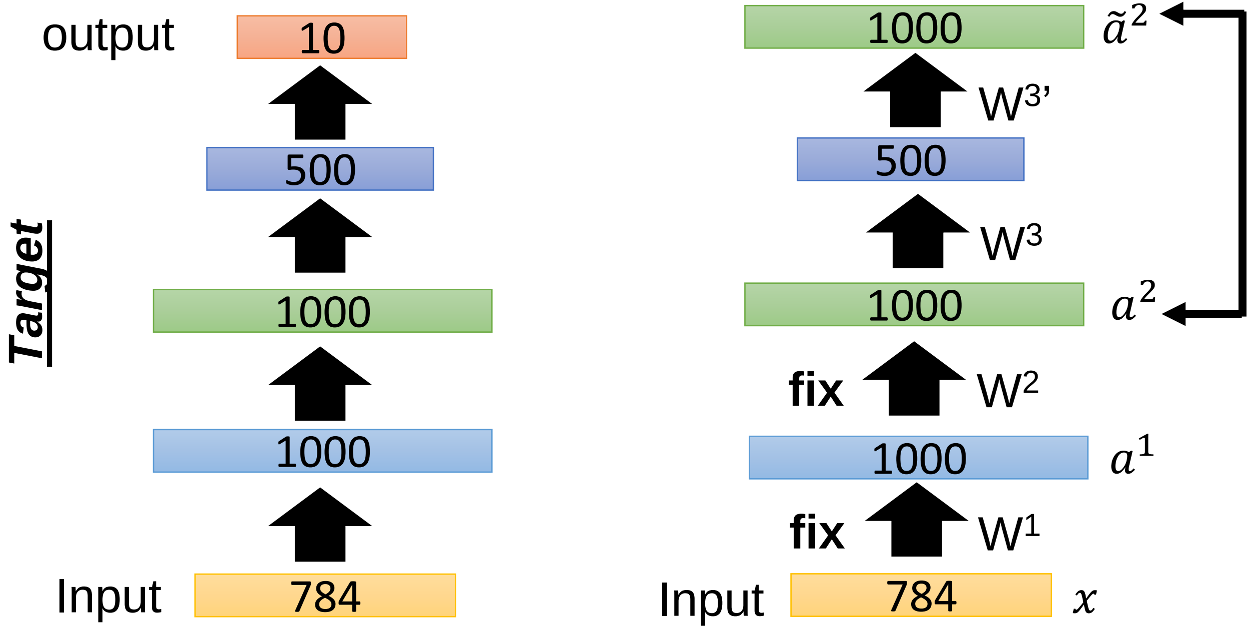
Auto-Encoder Layer-Wise Pre-Training



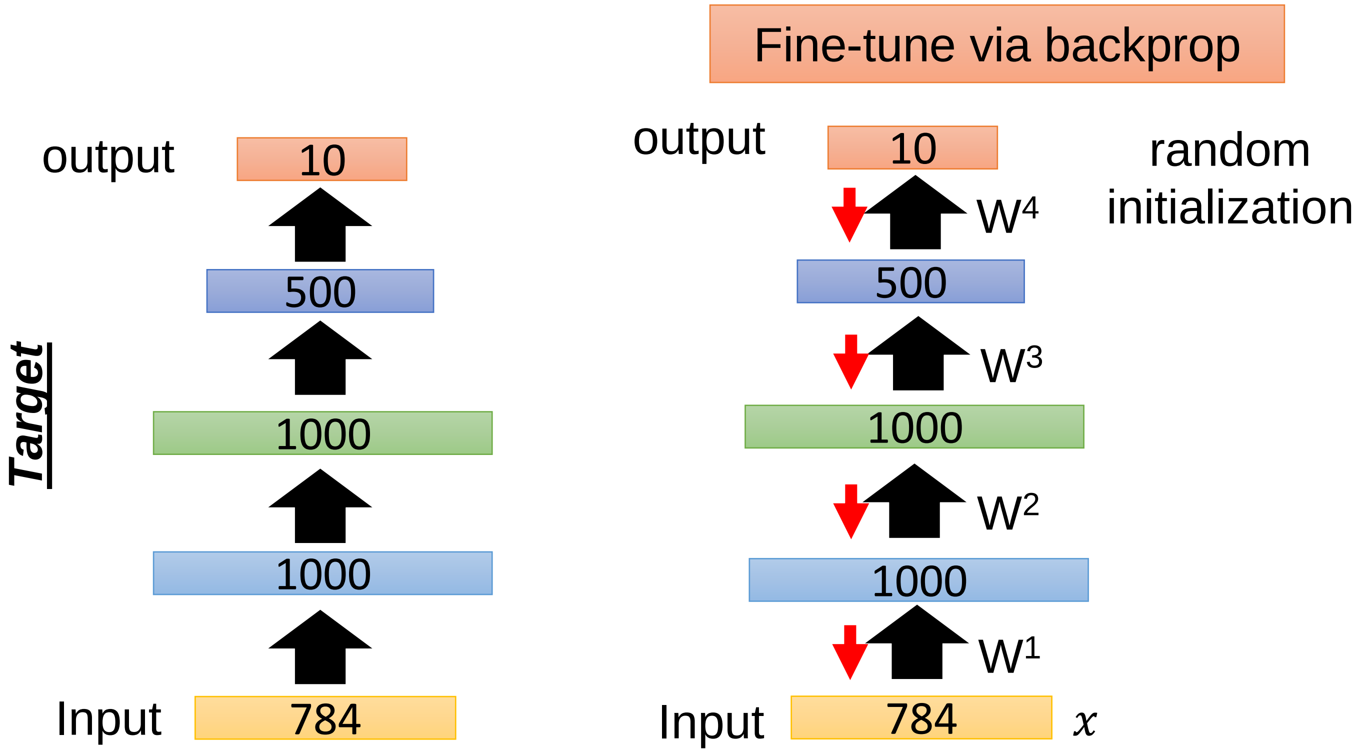
Auto-Encoder Layer-Wise Pre-Training



Auto-Encoder Layer-Wise Pre-Training

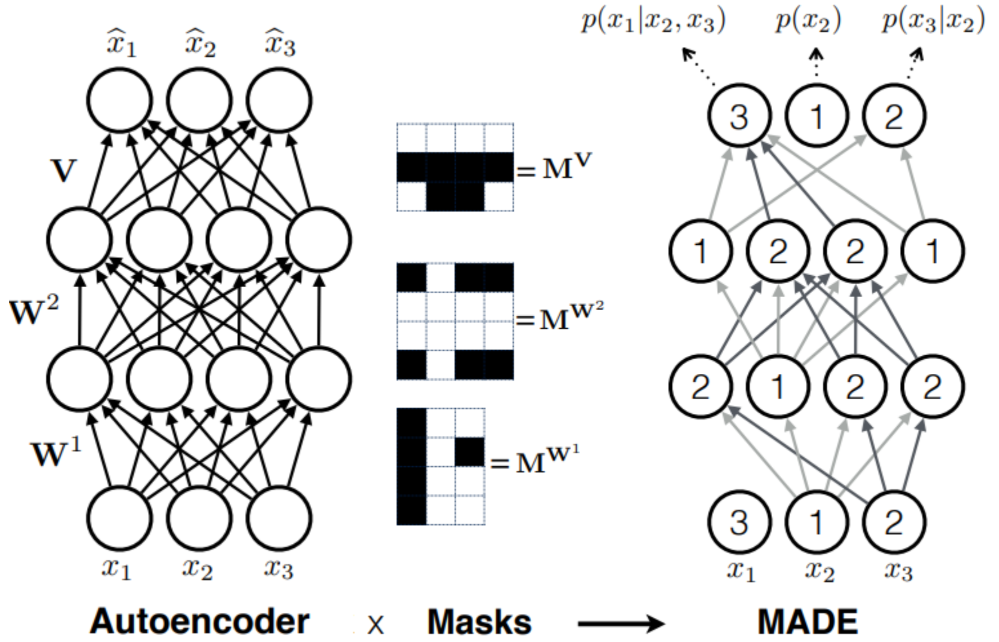


Auto-Encoder Layer-Wise Pre-Training



Masked Auto-Encoder (Germain et al., 2015)

- MADE: masked auto-encoder for distribution estimation
  - Reconstruction in a given ordering

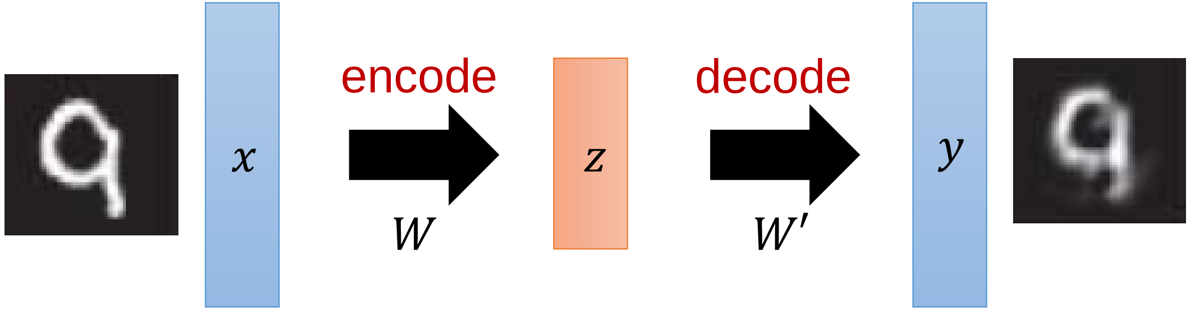


30

Variational Auto-Encoder

Representation Learning and Generation

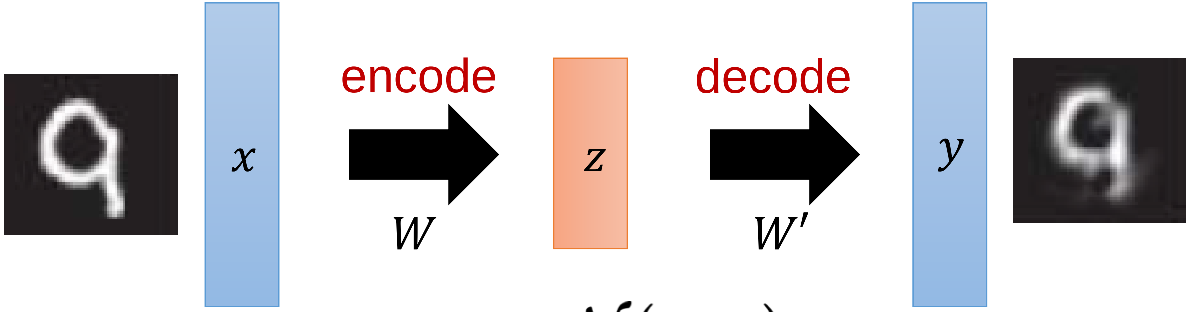
Generation from Latent Codes



How can we set a latent code for generation?

Latent Code Distribution Constraints

- Constrain the data distribution for learned latent codes
- Generate the latent code via a prior distribution

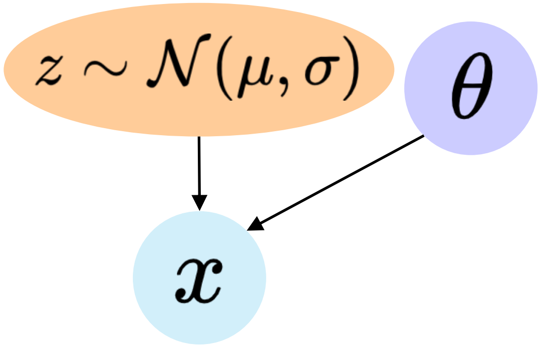


$$z \sim \mathcal{N}(\mu, \sigma)$$

Variational Auto-Encoder

- An observed output x
- A latent variable z generated from a Gaussian
- A function (network) f parameterized by θ maps from z to x

$$\underbrace{x}_{\text{Observed}} = f(\underbrace{z}_{\text{Latent}}; \theta)$$

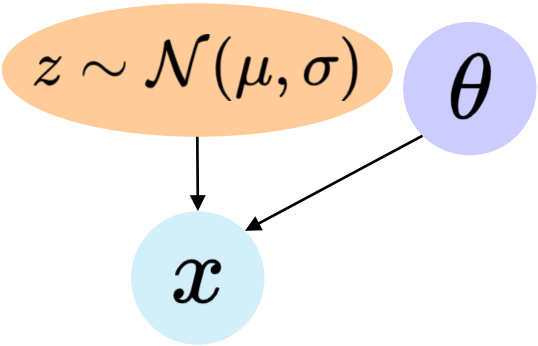


Idea: the compact representations follow a distribution

Variational Auto-Encoder $x = f(z; \theta)$

Observed Latent

- For each datapoint i
  - Draw latent variables $z_i \sim p(z)$
  - Draw a datapoint $x_i \sim p_\theta(x \mid z)$



- Joint probability distribution over data and latent variables

$$p(x, z) = \underbrace{p(z)}_{\text{prior}} \underbrace{p_\theta(x \mid z)}_{\text{posterior}}$$

- Learning objective: maximize the corpus log likelihood

$$\log P(\mathcal{X}) = \sum_{x \in \mathcal{X}} \log P(x; \theta)$$

Variational Auto-Encoder

- The marginal likelihood of a single datapoint x

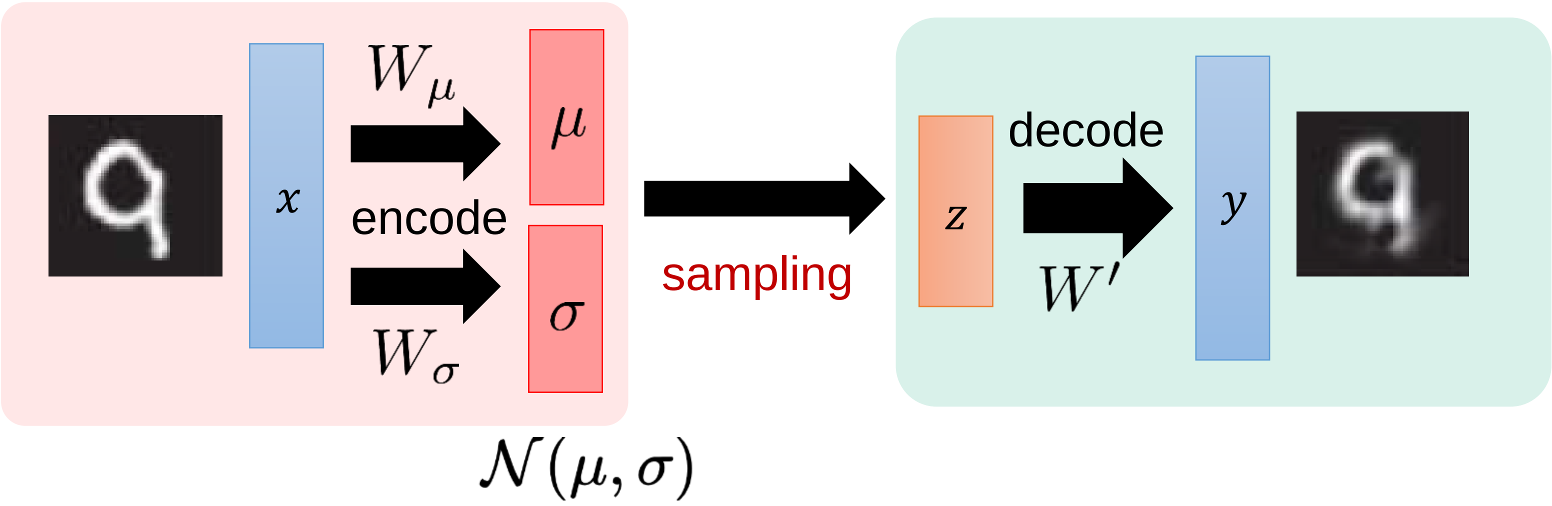
$$P(x;\theta) = \int P(x \mid z;\theta)P(z)dz$$

- Approximation by sampling z

$$P(x;\theta) \approx \sum_{z \sim P(z)} P(x \mid z;\theta)$$

Variational Auto-Encoder

- Two tasks
  - Learn to generate data from the latent code: $p_\theta(x \mid z)$
  - Learn the distribution of latent factors: $p_\theta(z \mid x)$



Variational Auto-Encoder

- Two tasks
  - Learn to generate data from the latent code: $p_\theta(x \mid z)$
  - Learn the distribution of latent factors: $p_\theta(z \mid x)$

$$p_\theta(z \mid x) = \frac{p_\theta(x \mid z)p(z)}{p(x)}$$

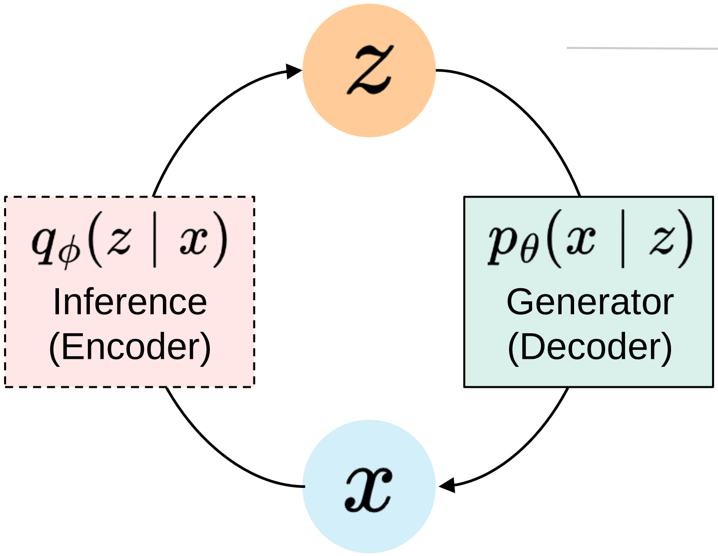
$$p(x) = \int p(z)p_\theta(x \mid z)dz \quad \text{intractable}$$

- **Variational inference** approximates the true posterior $p_\theta(z \mid x)$ with a family of distributions $q_\phi(z \mid x)$

$$\text{minimize} \quad \mathrm{KL}(q_\phi(z \mid x) \,\|\, p_\theta(z \mid x))$$

Variational Auto-Encoder

- Two tasks
  - Generator (Decoder): $p_\theta(x \mid z)$
  - Inference (Encoder) : $q_\phi(z \mid x)$

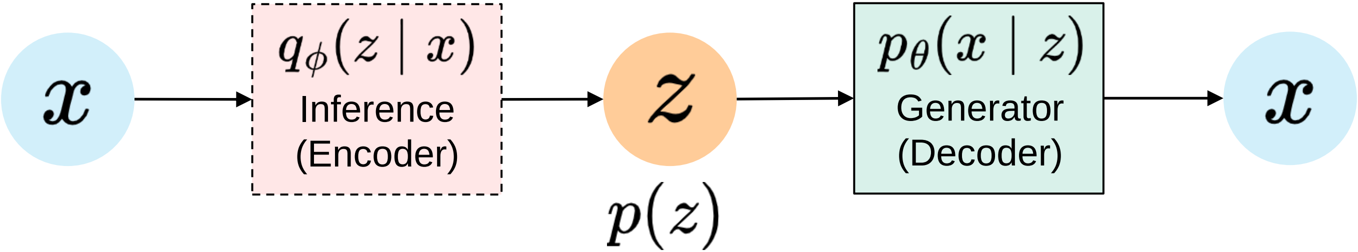


$$\underbrace{\mathbb{E}_{z \sim q_\phi(z|x)}\left[\log p_\theta(x \mid z)\right]}_{\text{reconstruction loss}} - \underbrace{D_{\mathrm{KL}}(q_\phi(z \mid x) \parallel p(z))}_{\text{KL regularizer}}$$

Regularized Auto-Encoder

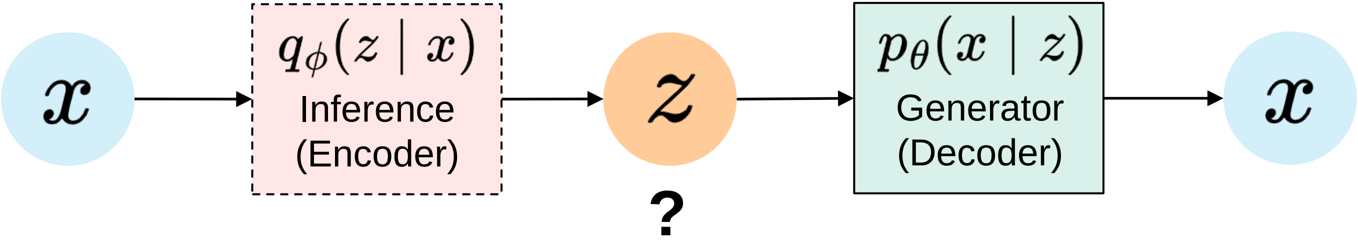
Variational Auto-Encoder

- VAE



$$\mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x \mid z)] - D_{\mathrm{KL}}(q_\phi(z \mid x) \parallel p(z))$$

- AE

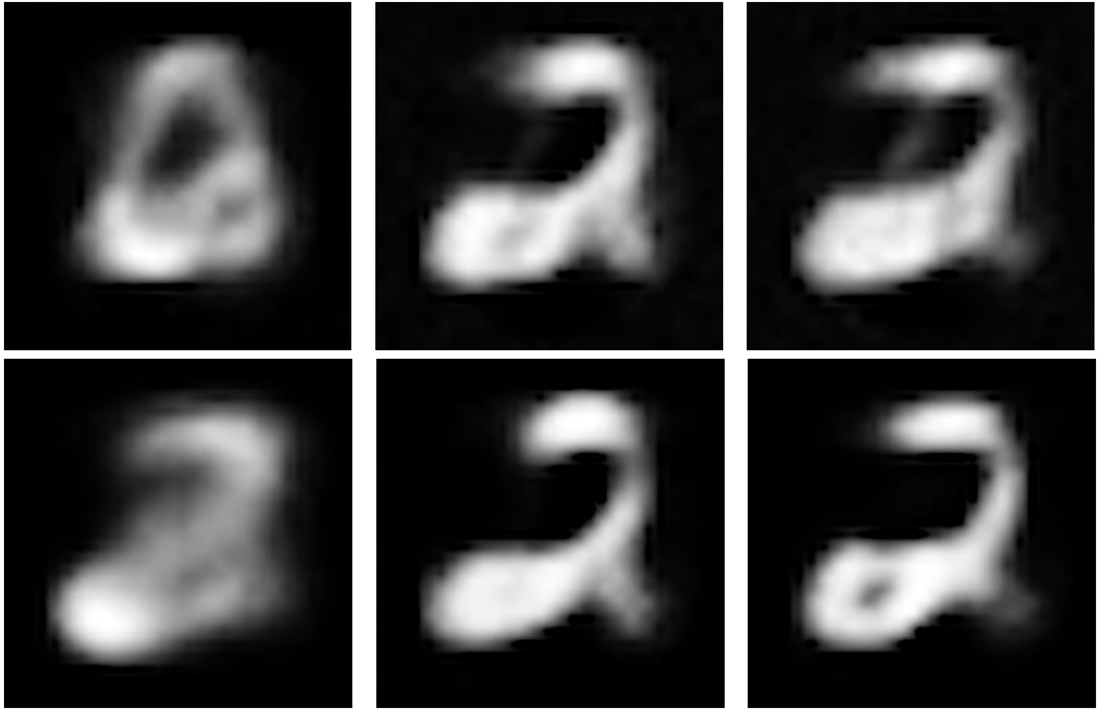


AE is not generative model: (1) Can’t sample new data from AE
(2) Can’t compute the log likelihood of data x

Image Reconstruction

AE

VAE



Text Reconstruction

- AE: standard encoder-decoder

embedding interpolation

i went to the store to buy some groceries .
i store to buy some groceries .
i were to buy any groceries .
horses are to buy any groceries .
horses are to buy any animal .
horses the favorite any animal .
horses the favorite favorite animal .
horses are my favorite animal .

- VAE

embedding interpolation

“ i want to talk to you . ”
“i want to be with you . ”
“i do n’t want to be with you . ”
i do n’t want to be with you .
she did n’t want to be with him .

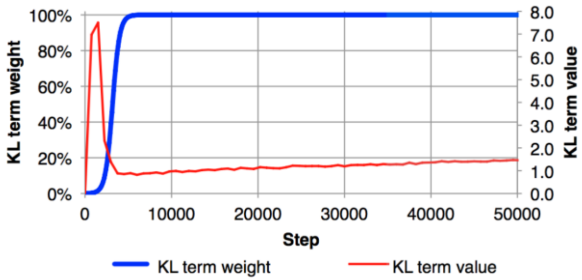
he was silent for a long moment .
he was silent for a moment .
it was quiet for a moment .
it was dark and cold .
there was a pause .
it was my turn .

VAE Training Tips

- **Posterior collapse** issue
  - KL divergence is easier to learn, so model learning relies solely on decoder and ignore latent variable

$$\underbrace{\mathbb{E}_{z \sim q_\phi(z|x)}\left[\log p_\theta(x \mid z)\right]}_{\text{requires good generative model}} - \underbrace{D_{\mathrm{KL}}(q_\phi(z \mid x) \parallel p(z))}_{\text{set the mean/variance of q to be the same as p}}$$

- Solutions
  - KL divergence annealing: an increasing constant to weight KL term
  - KL thresholding $\sum_i \max[\lambda, D_{\mathrm{KL}}(q_\phi(z_i|x)||p(z_i))]$



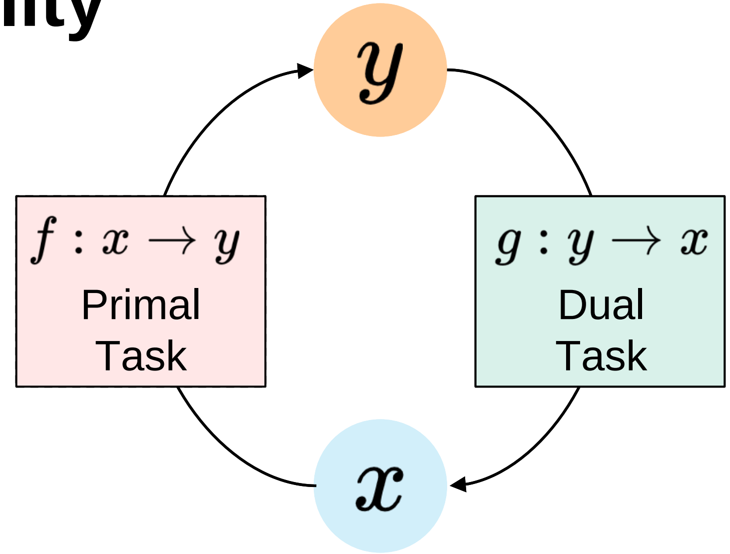
43

Dual Learning

Learning Two Tasks via Duality

Slides credited from ACML 2018 Tutorial

Task Duality



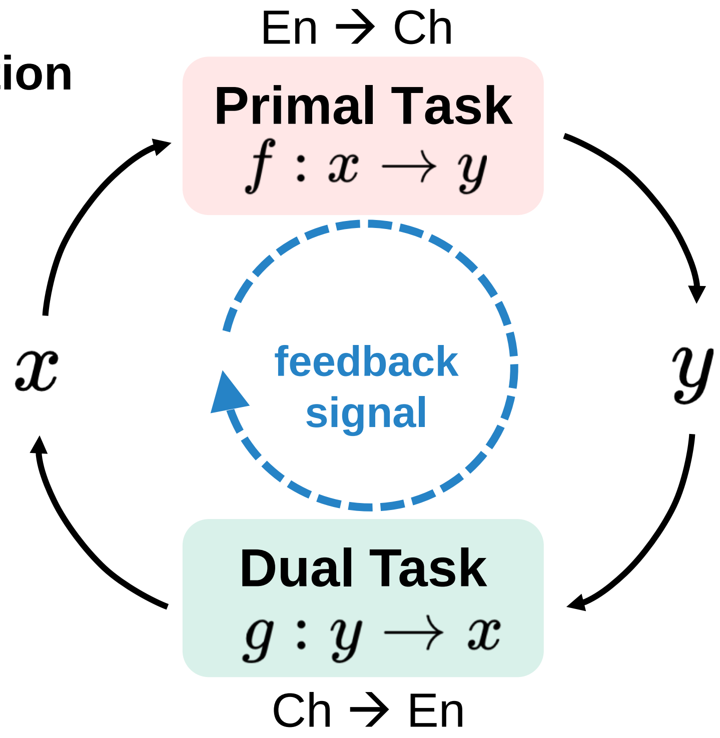
AI Tasks	$f : x \to y$	$g : y \to x$
Machine translation	EN → CH	CH → EN
Speech processing	ASR	TTS
Image understanding	captioning	Image generation
Language understanding	Language understanding	Language generation
Question answering	QA	Question generation

Dual Unsupervised Learning

- Idea: improve tasks by leveraging feedback signal via RL etc.

Machine Translation

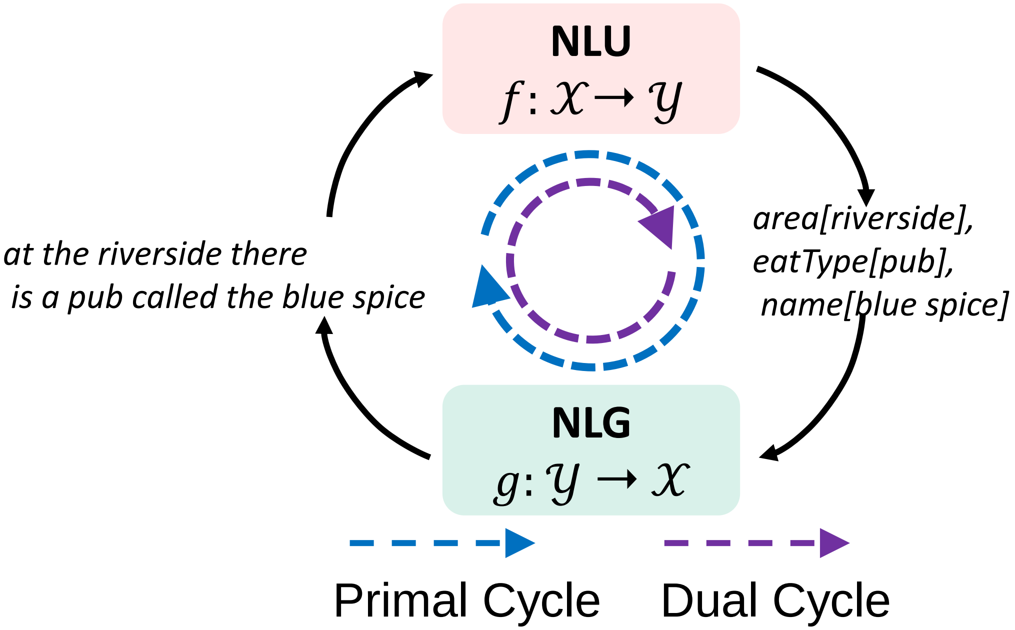
- English sentence x
- New English sentence $x' = g(y)$



- Chinese sentence $y = f(x)$

Joint Dual Learning

- Idea: perfectly *reconstructing* the input via two models



$$g(f(x)) \equiv x$$

$$f(g(y)) \equiv y$$

Shang-Yu Su, Chao-Wei Huang, and Yun-Nung Chen, "Towards Unsupervised Language Understanding and Generation by Joint Dual Learning," in *Proceedings of The 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

Joint Dual Learning Objective

- Explicit
  - Reconstruction Likelihood

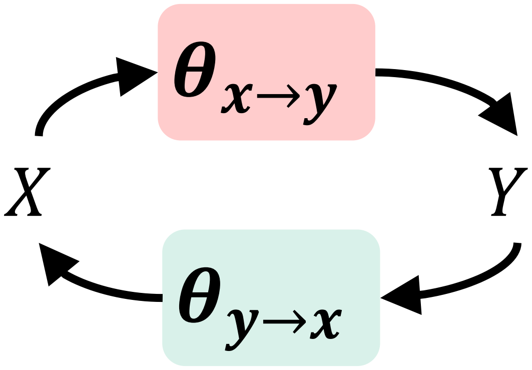
$$\begin{cases} \log p(x \mid f(x_i; \theta_{x\to y}); \theta_{y\to x}) & \textbf{Primal} \\ \log p(y \mid g(y_i; \theta_{y\to x}); \theta_{x\to y}) & \textbf{Dual} \end{cases}$$

  - Automatic Evaluation Score
    - BLEU and ROUGE for language (NLG)
    - F-score for semantic (NLU)
- Implicit
  - Model-based methods estimating data distribution
    - Language modeling (LM) for language
    - Masked autoencoder (MADE) for semantics

Shang-Yu Su, Chao-Wei Huang, and Yun-Nung Chen, "Towards Unsupervised Language Understanding and Generation by Joint Dual Learning," in *Proceedings of The 58th Annual Meeting of the Association for Computational Linguistics (ACL),* 2020.

Dual Supervised Learning (Xia et al., 2017)

- Proposed for machine translation
- Consider two domains X and Y, and two tasks $X \to Y$ and $Y \to X$



We have $P(x, y) = P(x \mid y)P(y) = P(y \mid x)P(x)$

Ideally $P(x, y) = P(x \mid y; \boldsymbol{\theta}_{\boldsymbol{y}\to\boldsymbol{x}})P(y) = P(y \mid x; \boldsymbol{\theta}_{\boldsymbol{x}\to\boldsymbol{y}})P(x)$

Xia, Y., Qin, T., Chen, W., Bian, J., Yu, N., & Liu, T. Y., “Dual supervised learning,” in *Proc. of the 34th International Conference on Machine Learning*, 2017.

Dual Supervised Learning

- Exploit the duality by forcing models to follow the probabilistic constraint $P(x \mid y; \boldsymbol{\theta}_{y \to x})P(y) = P(y \mid x; \boldsymbol{\theta}_{x \to y})P(x)$

Objective function

$$\begin{cases} \min_{\theta_{x \to y}} \mathbb{E}\left[l_1\left(f\left(x; \theta_{x \to y}\right), y\right)\right] + \lambda_{x \to y}\, l_{duality} \\ \min_{\theta_{y \to x}} \mathbb{E}\left[l_2\left(g\left(y; \theta_{y \to x}\right), x\right)\right] + \lambda_{y \to x}\, l_{duality} \end{cases}$$

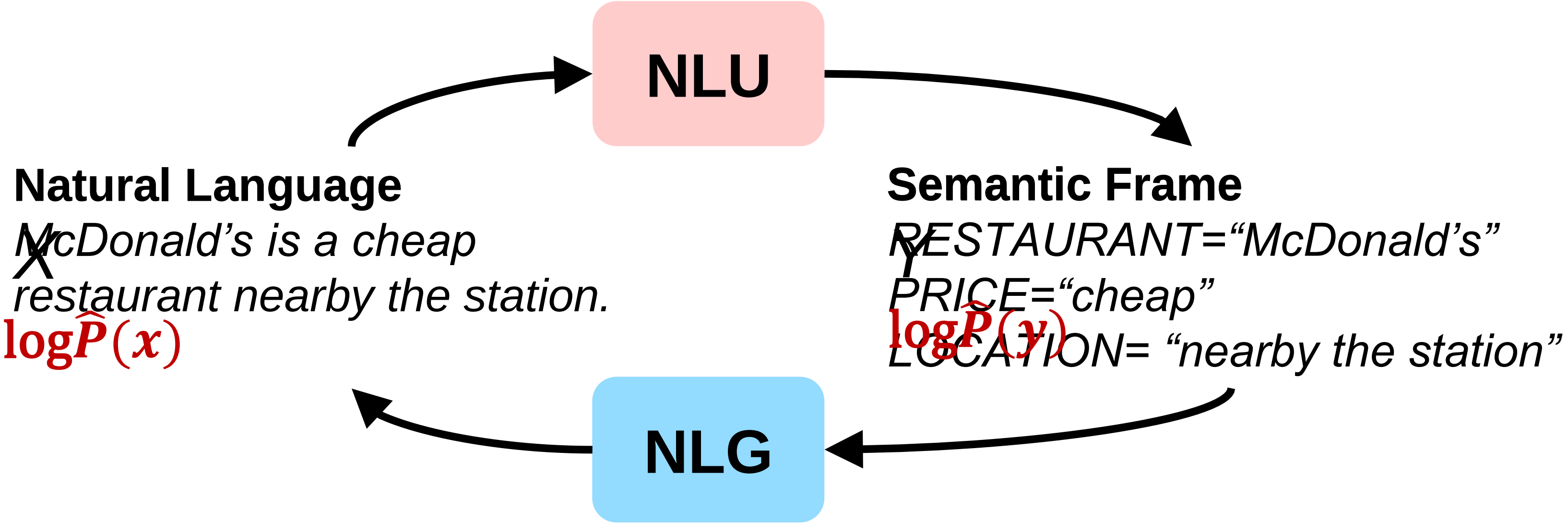
$$l_{duality} = \left(\boxed{\log \hat{P}(x)} + \log P(y \mid x; \theta_{x \to y}) - \boxed{\log \hat{P}(y)} - \log P(x \mid y; \theta_{y \to x})\right)^2$$

How to model the marginal distributions of X and Y?

Xia, Y., Qin, T., Chen, W., Bian, J., Yu, N., & Liu, T. Y., "Dual supervised learning," in *Proc. of the 34th International Conference on Machine Learning*, 2017.

Dual Supervised Learning

- Considering NLU and NLG

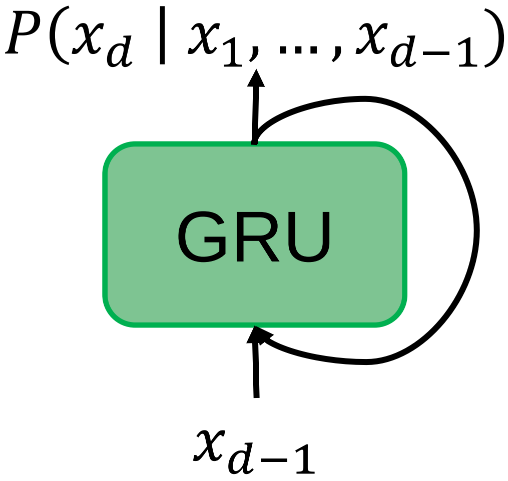


Shang-Yu Su, Chao-Wei Huang, and Yun-Nung Chen, "Dual Supervised Learning for Natual Language Understanding and Generation," in *Proceedings of The 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

Natural Language $\log \hat{P}(x)$

- Language modeling

$$p(x) = \prod_{d}^{D} p(x_d \mid x_1, \ldots, x_{d-1})$$



Shang-Yu Su, Chao-Wei Huang, and Yun-Nung Chen, "Dual Supervised Learning for Natual Language Understanding and Generation," in *Proceedings of The 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

Semantic Frame $\log \hat{P}(y)$

- We treat NLU as a multi-label classification problem
- Each label is a slot-value pair

RESTAURANT="McDonald's"
PRICE="cheap"
LOCATION= "nearby the station"

$$\rightarrow \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

How to model the marginal distributions of y?

Shang-Yu Su, Chao-Wei Huang, and Yun-Nung Chen, "Dual Supervised Learning for Natual Language Understanding and Generation," in *Proceedings of The 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

Semantic Frame $\log \hat{P}(y)$

- Naïve approach
  - Calculate prior probability for each label $\hat{P}(y_i)$ on training set.
  - $\hat{P}(y) = \prod \hat{P}(y_i)$

Assumption: labels are independent

Restaurant: “McDonald’s”	Price: “cheap”	Food: “Pizza”
Restaurant: “KFC”	Price: “expensive”	Food: “Hamburger”
Restaurant: “PizzaHut”		Food:”Chinese”

Shang-Yu Su, Chao-Wei Huang, and Yun-Nung Chen, “Dual Supervised Learning for Natual Language Understanding and Generation,” in *Proceedings of The 57th Annual Meeting of the Association for Computational Linguistics (ACL),* 2019.

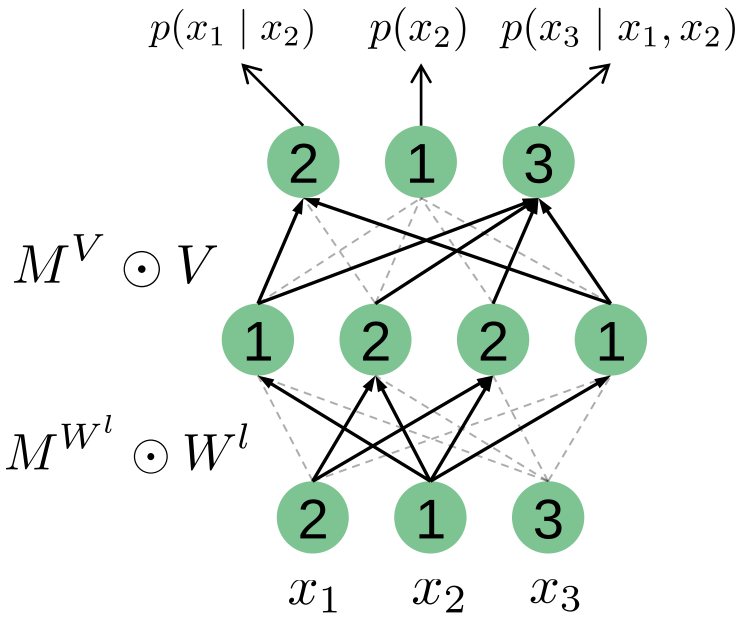
Semantic Frame $\log \hat{P}(y)$

- Masked autoencoder for distribution estimation (MADE)
  Introduce sequential dependency among labels by masking certain connections

$$M = \begin{cases} 1 & \text{if } m^l(k') \geq m^{l-1}(k) \text{ or } m^L(d) > m^{L-1}(k) \\ 0 & \text{otherwise} \end{cases}$$

$$p(x) = \prod_d^D p(x_d \mid S_d)$$

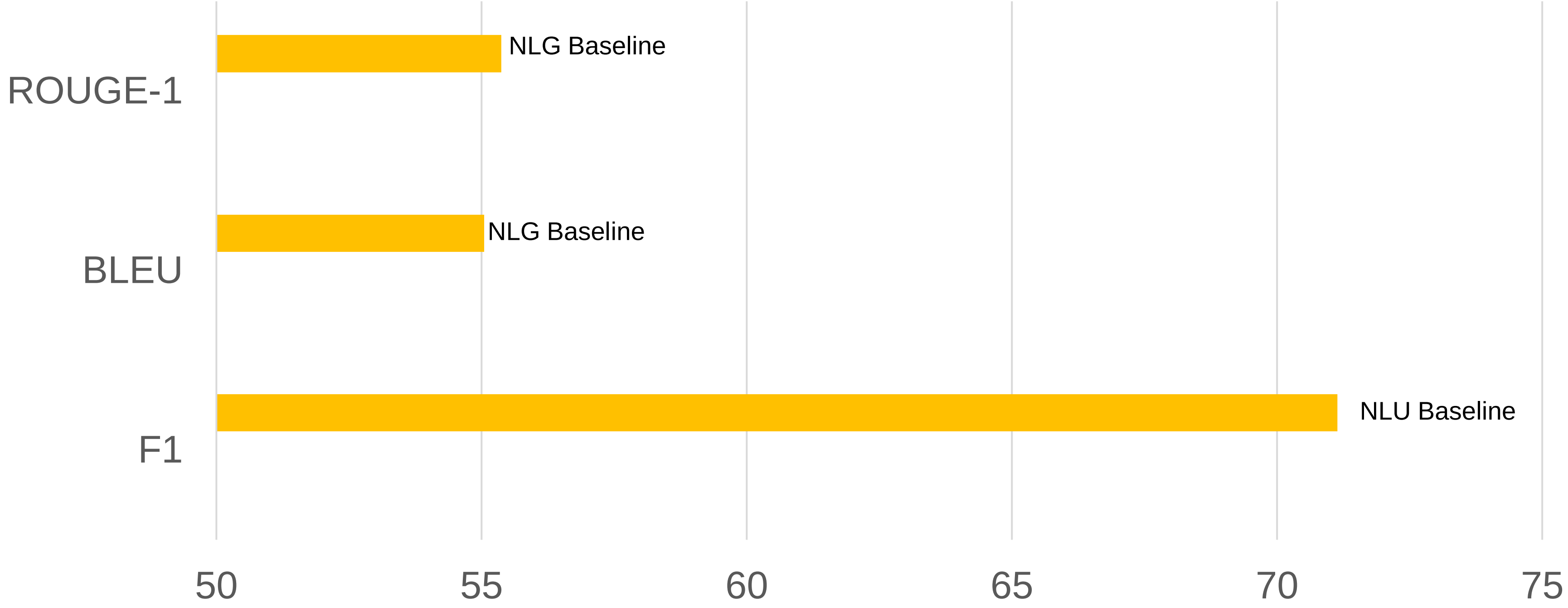
→ marginal distribution of y



Germain, M., Gregor, K., Murray, I., & Larochelle, H., "MADE: Masked autoencoder for distribution estimation," in *Proceedings of International Conference on Machine Learning*, 2015.

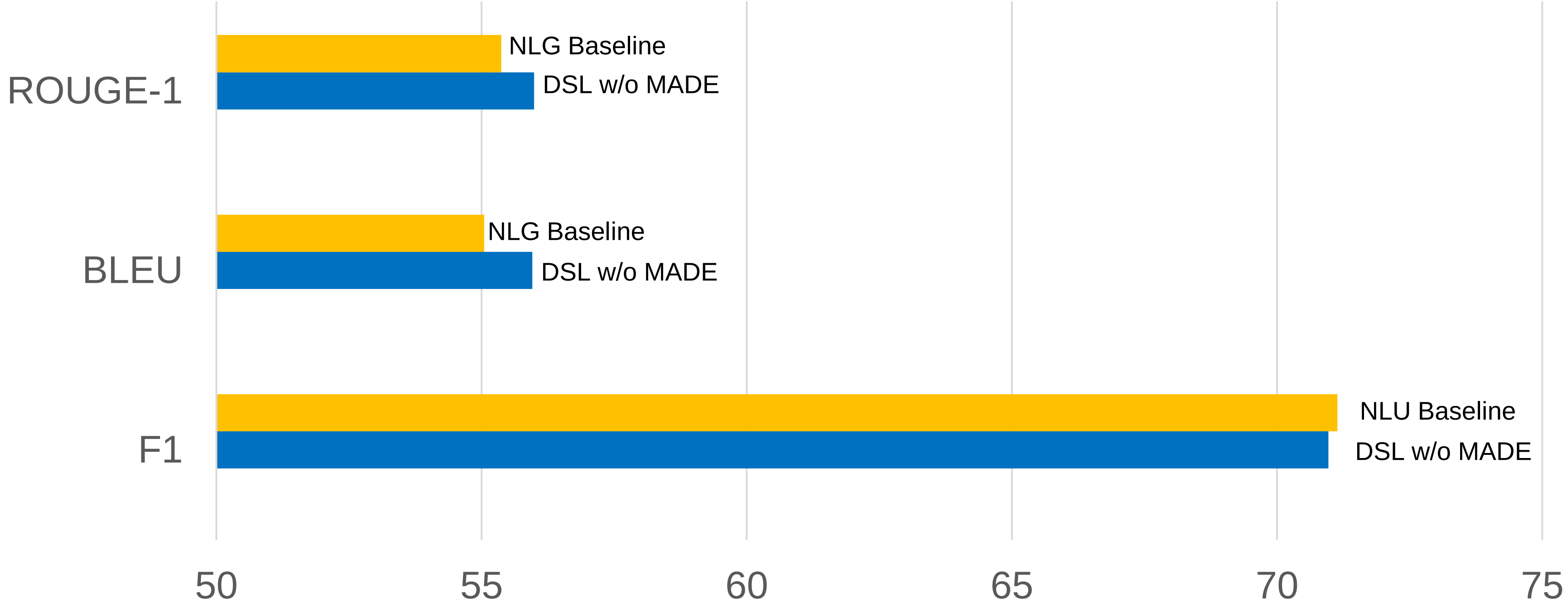
NLU/NLG Results

- E2E NLG data: 50k examples in the restaurant domain
- NLU: F-1 score; NLG: BLEU, ROUGE



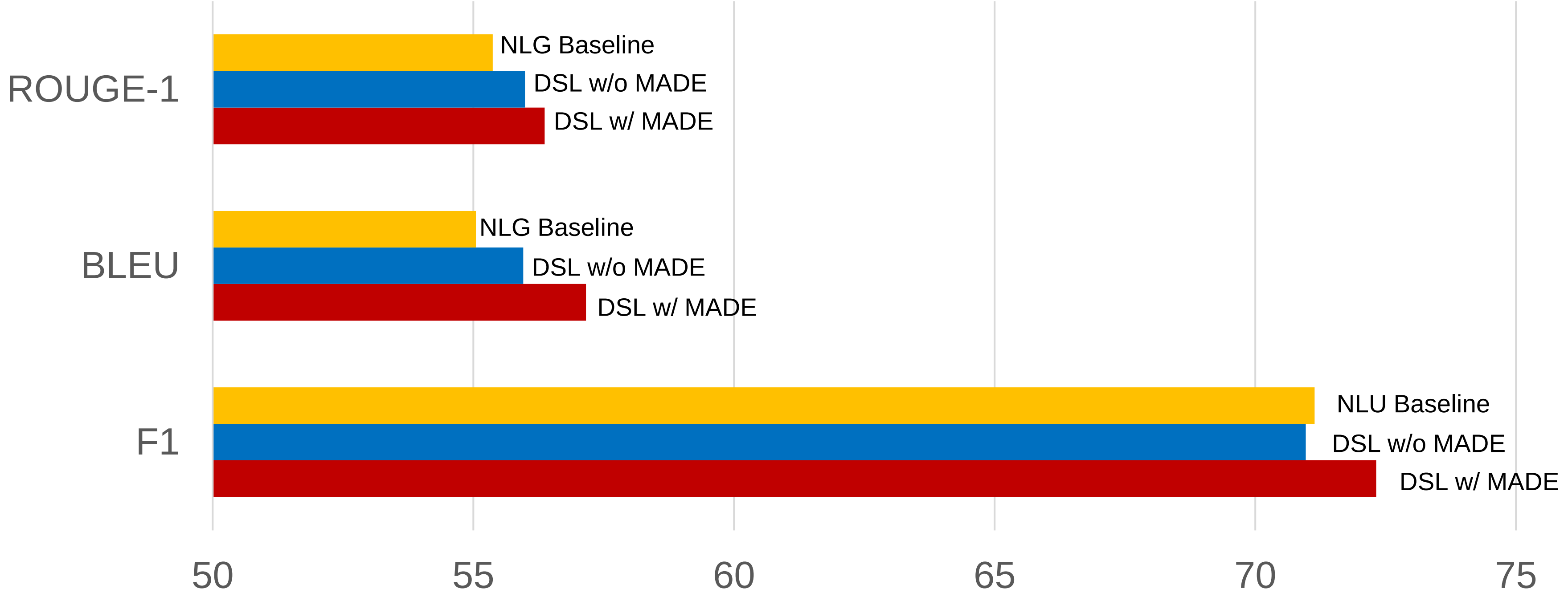
NLU/NLG Results

- E2E NLG data: 50k examples in the restaurant domain
- NLU: F-1 score; NLG: BLEU, ROUGE



NLU/NLG Results

- E2E NLG data: 50k examples in the restaurant domain
- NLU: F-1 score; NLG: BLEU, ROUGE



Comparison

Unsupervised/semi-supervised learning: only one task; no feedback signals for unlabeled data

Dual learning: multiple tasks involved; automatically generate reinforcement feedback for unlabeled data,

Co-training: only one task; different feature sets provide complementary information about the instance

Dual learning: multiple tasks involved; no assumption on feature sets

Multi-task learning: multiple tasks share the same representation

Dual learning: don’t need to share representations, only when the closed loop

Transfer learning: use auxiliary tasks to boost the target task

Dual learning: all tasks are mutually and simultaneously boosted

59

Self-Supervised Learning

Self-Prediction and Contrastive Learning

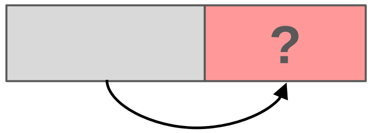
Slides credited from NeurIPS 2021 Tutorial

Self-Supervised Learning

- Self-supervised learning (SSL): a special type of representation learning via unlabeled data
- Idea: constructing supervised tasks out of unsupervised data
  - High cost of data annotation
  - Limited annotated data
  - Good representation makes it easier to transfer to diverse downstream tasks

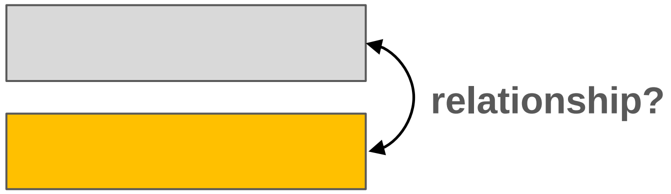
Self-Supervised Learning

- Self-Prediction
  - Given an individual data sample, the task is to predict one missing part of the sample given the other part



"intra-sample" prediction

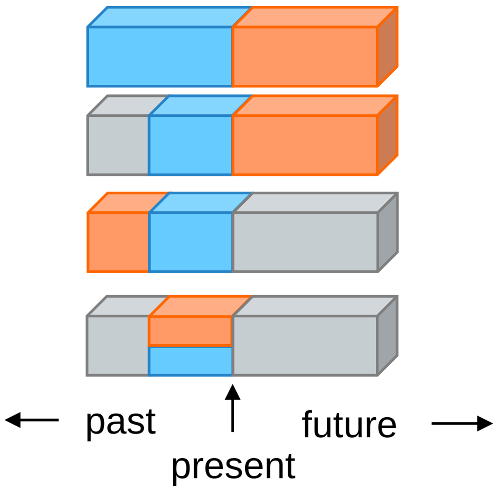
- Contrastive Learning
  - Given multiple data samples, the task is to predict their relationship



"inter-sample" prediction

Self-Prediction (illustration from Yann LeCun)

- Assume: a part of the input is unknown and predict it
  - Predict the future from the past
  - Predict the future from the recent past
  - Predict the past from the present
  - Predict the top from the bottom
  - Predict the occluded from the visible



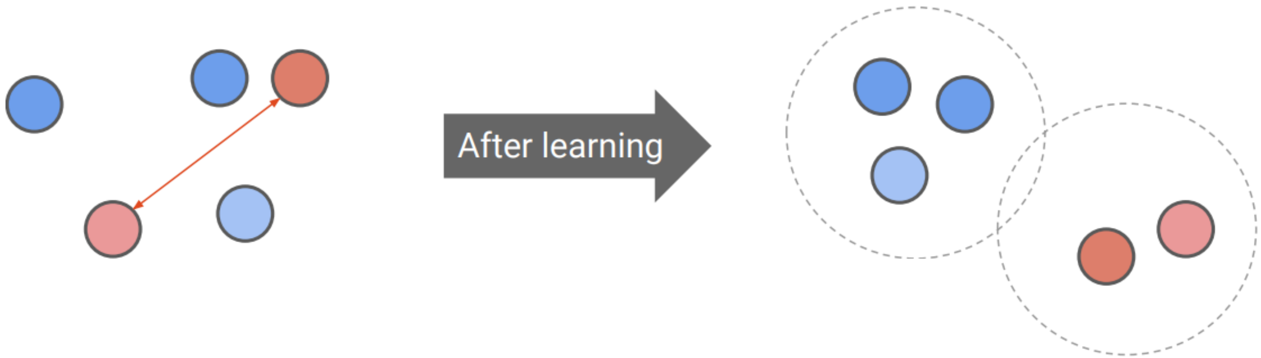
63

Contrastive Learning

Adapting Embedding Spaces

Contrastive Learning

- Idea: learn an embedding space where *similar* sample pairs stay *close* to each other while *dissimilar* ones are *far apart*
  - Inter-sample classification
  - Feature clustering
  - Multi-view coding



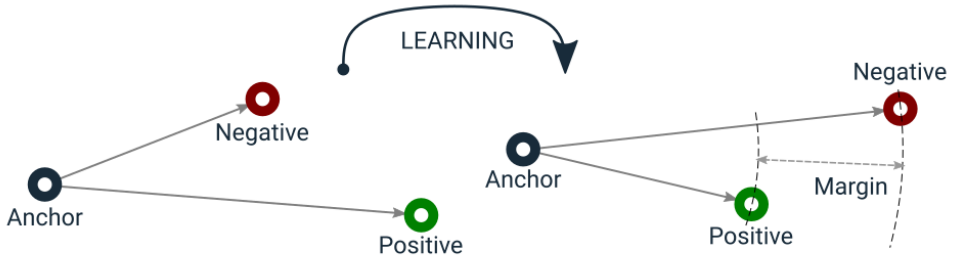
Inter-Sample Classification

- Task: given both similar (“positive”) and dissimilar (“negative”) candidates, identifying which is similar to the anchor datapoint
- Datapoint candidates
  1. The original input and its distorted version
  2. Data capturing the same target from different views

Inter-Sample Classification

- **Triplet loss** (Schroff et al., 2015)
  - minimize the distance between the anchor x and positive x^+ and maximize the distance between the anchor x and negative x^- at the same time

$$\mathcal{L}_{\text{triplet}}(x, x^+, x^-) = \sum_x \max(0, \underbrace{\|f(x) - f(x^+)\|_2^2}_{\text{as close as possible}} - \underbrace{\|f(x) - f(x^-)\|_2^2}_{\text{as far as possible}} + \epsilon)$$



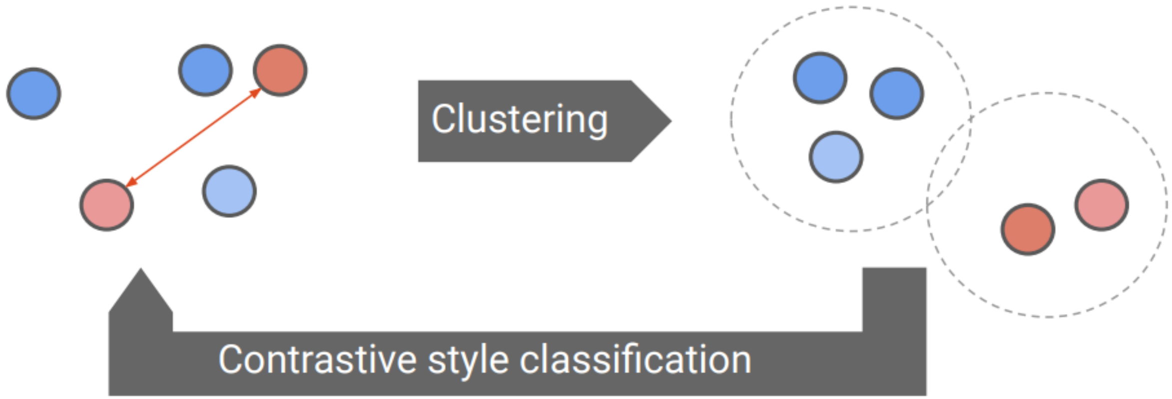
Inter-Sample Classification

- **N-pair loss** (Sohn, 2016)
  - generalizes to include comparison with multiple negative samples

$$\mathcal{L}_{\text{N-pair}}(x, x^+, \{x_i^-\}) = \log\left(1 + \sum_i \exp\left(f(x)^T f(x_i^-) - f(x)^T f(x^+)\right)\right)$$

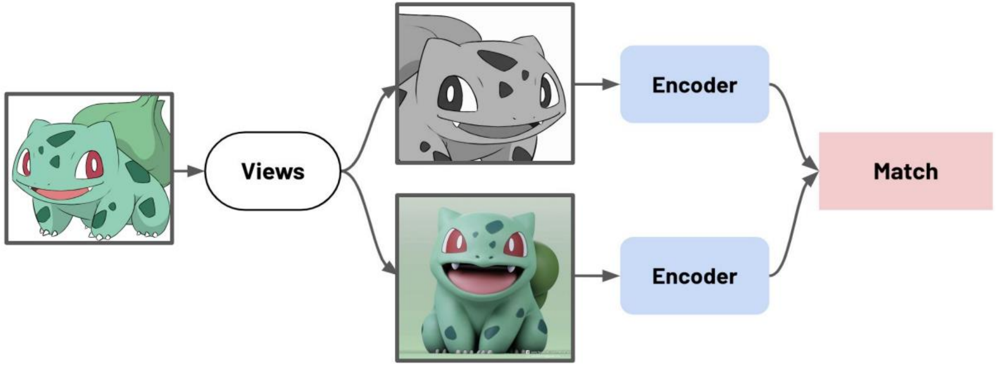
Feature Clustering

- Idea: cluster similar datapoints based on learned features
  → assign pseudo labels to samples for intra-sample classification



Multiview Coding

- Idea: apply the InfoNCE objective to different views of input
  - Data augmentation is adopted for generating different views
  - “views” can come from different modalities

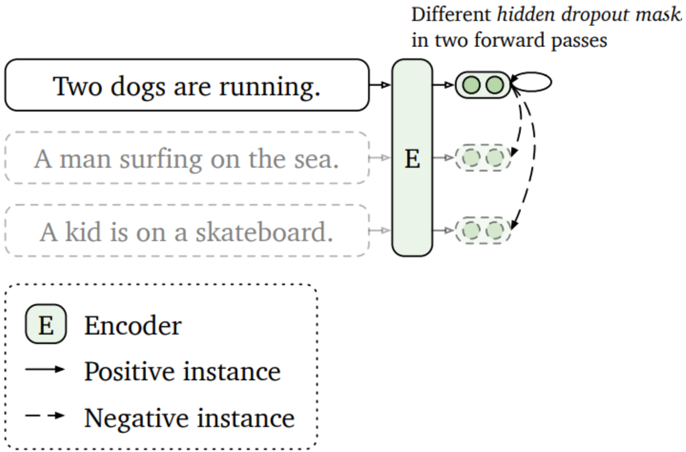


mainstream approaches for contrastive learning

Contrastive Learning in NLP

- **SimCSE** (Gao et al., 2021): simple contrastive learning of sentence embeddings
  - *Unsupervised*: predict a sentence from itself with only dropout noise

(a) Unsupervised SimCSE



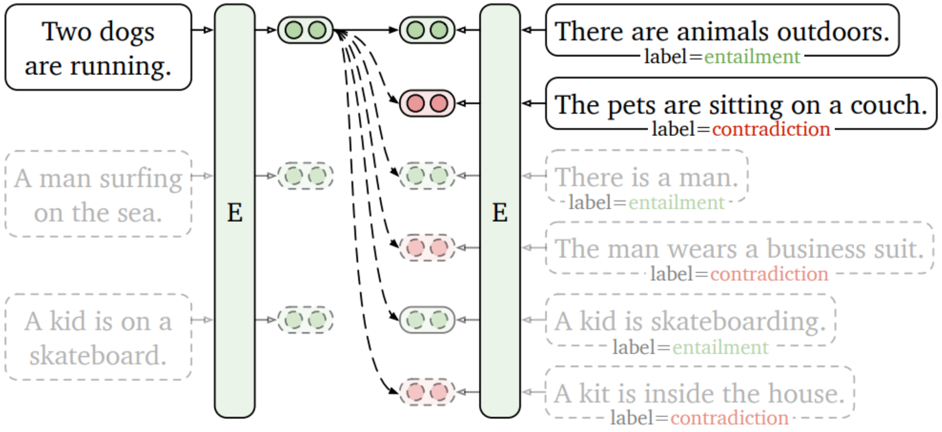
Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
Unsupervised models								
GloVe embeddings (avg.)♣	55.14	70.66	59.73	68.25	63.66	58.02	53.76	61.32
BERT$_{base}$ (first-last avg.)	39.70	59.38	49.67	66.03	66.19	53.87	62.06	56.70
BERT$_{base}$-flow	58.40	67.10	60.85	75.16	71.22	68.66	64.47	66.55
BERT$_{base}$-whitening	57.83	66.90	60.90	75.08	71.31	68.24	63.73	66.28
IS-BERT$_{base}$♡	56.77	69.24	61.21	75.23	70.16	69.21	64.25	66.58
CT-BERT$_{base}$	61.63	76.80	68.47	77.50	76.48	74.31	69.19	72.05
∗ SimCSE-BERT$_{base}$	**68.40**	**82.41**	**74.38**	**80.91**	**78.56**	**76.85**	**72.23**	**76.25**
RoBERTa$_{base}$ (first-last avg.)	40.88	58.74	49.07	65.63	61.48	58.55	61.63	56.57
RoBERTa$_{base}$-whitening	46.99	63.24	57.23	71.36	68.99	61.36	62.91	61.73
DeCLUTR-RoBERTa$_{base}$	52.41	75.19	65.52	77.12	78.63	72.41	**68.62**	69.99
∗ SimCSE-RoBERTa$_{base}$	**70.16**	**81.77**	**73.24**	**81.36**	**80.65**	**80.22**	68.56	**76.57**
∗ SimCSE-RoBERTa$_{large}$	**72.86**	**83.99**	**75.62**	**84.77**	**81.80**	**81.98**	**71.26**	**78.90**

Gao, Tianyu, Xingcheng Yao, and Danqi Chen. "SimCSE: Simple Contrastive Learning of Sentence Embeddings." in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021.

Contrastive Learning in NLP

- **SimCSE** (Gao et al., 2021): simple contrastive learning of sentence embeddings
  - *Supervised*: further adapt embeddings based on labels

(b) Supervised SimCSE

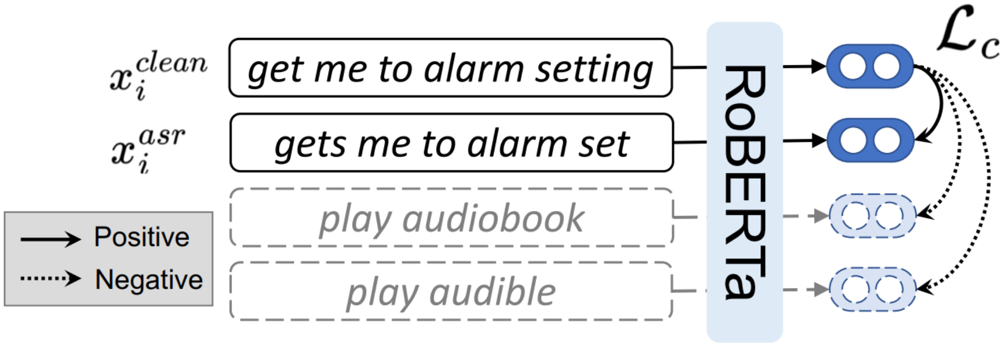


Supervised models								
InferSent-GloVe♣	52.86	66.75	62.15	72.77	66.87	68.03	65.65	65.01
Universal Sentence Encoder♣	64.49	67.80	64.61	76.83	73.18	74.92	76.69	71.22
$\text{SBERT}_{\text{base}}$♣	70.97	76.53	73.19	79.09	74.30	77.03	72.91	74.89
$\text{SBERT}_{\text{base}}$-flow	69.78	77.27	74.35	82.01	77.46	79.12	76.21	76.60
$\text{SBERT}_{\text{base}}$-whitening	69.65	77.57	74.66	82.27	78.39	79.52	76.91	77.00
$\text{CT-SBERT}_{\text{base}}$	74.84	83.20	78.07	83.84	77.93	81.46	76.42	79.39
∗ $\text{SimCSE-BERT}_{\text{base}}$	**75.30**	**84.67**	**80.19**	**85.40**	**80.82**	**84.25**	**80.39**	**81.57**
$\text{SRoBERTa}_{\text{base}}$♣	71.54	72.49	70.80	78.74	73.69	77.77	74.46	74.21
$\text{SRoBERTa}_{\text{base}}$-whitening	70.46	77.07	74.46	81.64	76.43	79.49	76.65	76.60
∗ $\text{SimCSE-RoBERTa}_{\text{base}}$	**76.53**	**85.21**	**80.95**	**86.03**	**82.57**	**85.83**	**80.50**	**82.52**
∗ $\text{SimCSE-RoBERTa}_{\text{large}}$	**77.46**	**87.27**	**82.36**	**86.66**	**83.93**	**86.70**	**81.95**	**83.76**

Gao, Tianyu, Xingcheng Yao, and Danqi Chen. "SimCSE: Simple Contrastive Learning of Sentence Embeddings." in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021.

Contrastive Learning in NLP

- **SpokenCSE** (Chang & Chen, 2022): improve ASR robustness
    - *Unsupervised*: learning with the paired clean/noisy sentences

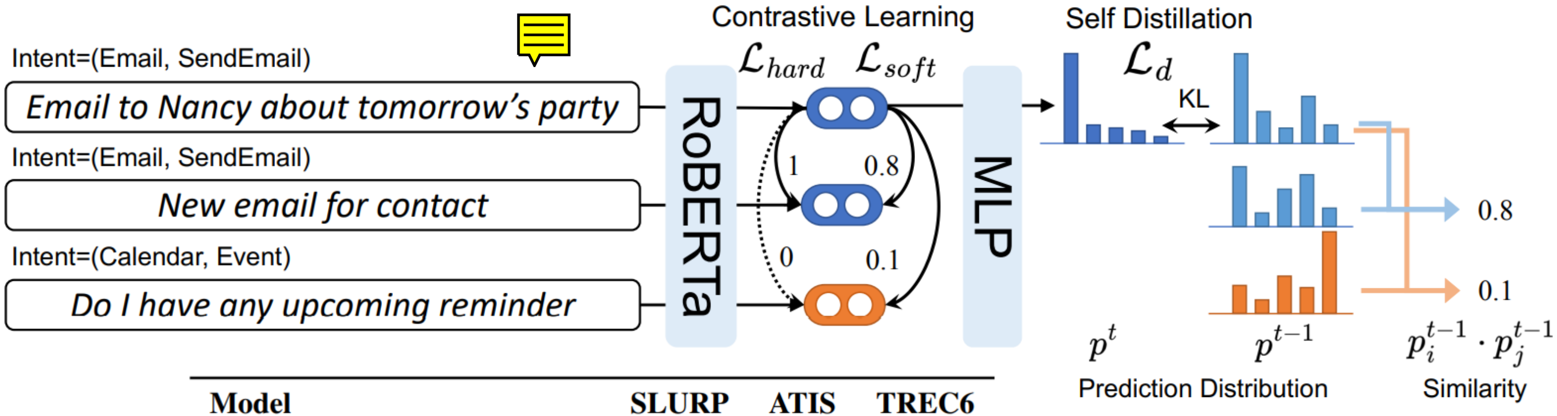


Model	SLURP	ATIS	TREC6
RoBERTa	83.97	94.53	84.08
Phoneme-BERT†	83.78	94.83	85.96
SimCSE	84.47	94.07	84.92
Proposed (pre-train only)	84.51	95.02	85.20

Ya-Hsin Chang and Yun-Nung Chen, "Contrastive Learning for Improving ASR Robustness in Spoken Language Understanding," in arXiv:2205.00693, 2022.

Contrastive Learning in NLP

- **SpokenCSE** (Chang & Chen, 2022): improve ASR robustness
  - *Supervised*: learning with self-distillation

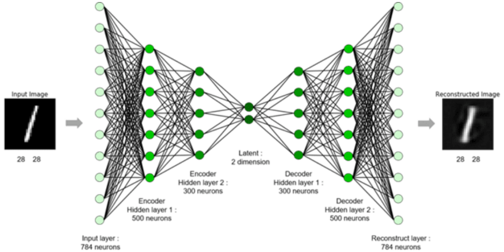


Model	SLURP	ATIS	TREC6
RoBERTa	83.97	94.53	84.08
Phoneme-BERT†	83.78	94.83	85.96
SimCSE	84.47	94.07	84.92
Proposed (pre-train only)	84.51	95.02	85.20
Proposed (pre-train + fine-tune)	**85.26**	**95.10**	**86.36**

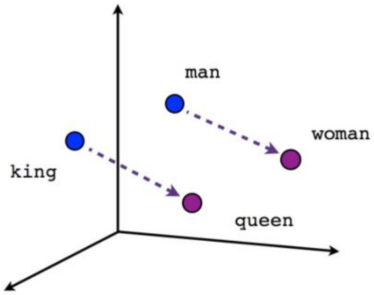
Ya-Hsin Chang and Yun-Nung Chen, "Contrastive Learning for Improving ASR Robustness in Spoken Language Understanding," in arXiv:2205.00693, 2022.

Diverse Approaches and Applications

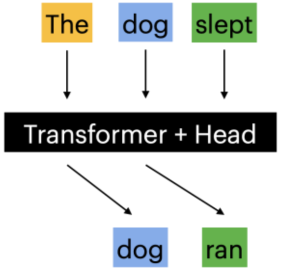
Auto-Encoders



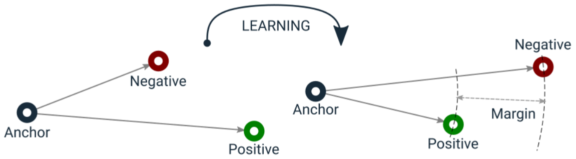
word2vec



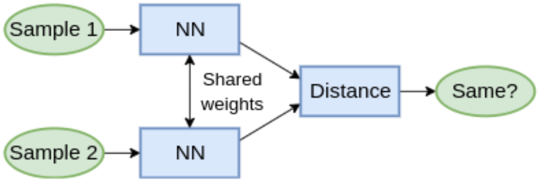
Autoregressive Language Modeling



Contrastive Learning



Siamese Networks



Concluding Remarks

- Labeling data is expensive, but we have large unlabeled data
- AE / VAE
  - exploits unlabeled data to learn latent factors as representations
  - learned representations can be transfer to other tasks
- Dual Learning
  - utilize the duality of two tasks
  - towards semi-supervised learning / unsupervised learning
- Self-Prediction
  - predict one missing part of the sample given the other part
- Contrastive Learning
  - positive pairs have similar embeddings