

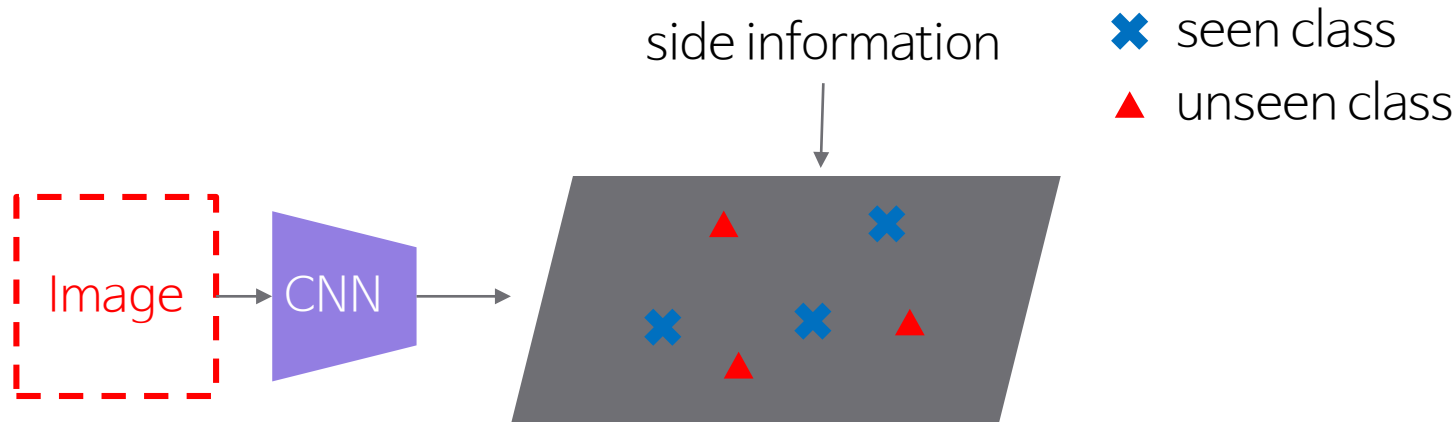
# *Zero-Shot Learning via Class-Conditioned Deep Generative Models*

2018, AAAI, W. Wang et al.

<https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16087/16709>

한양대학교  
컴퓨터 소프트웨어학과  
인공지능 연구실  
조건희

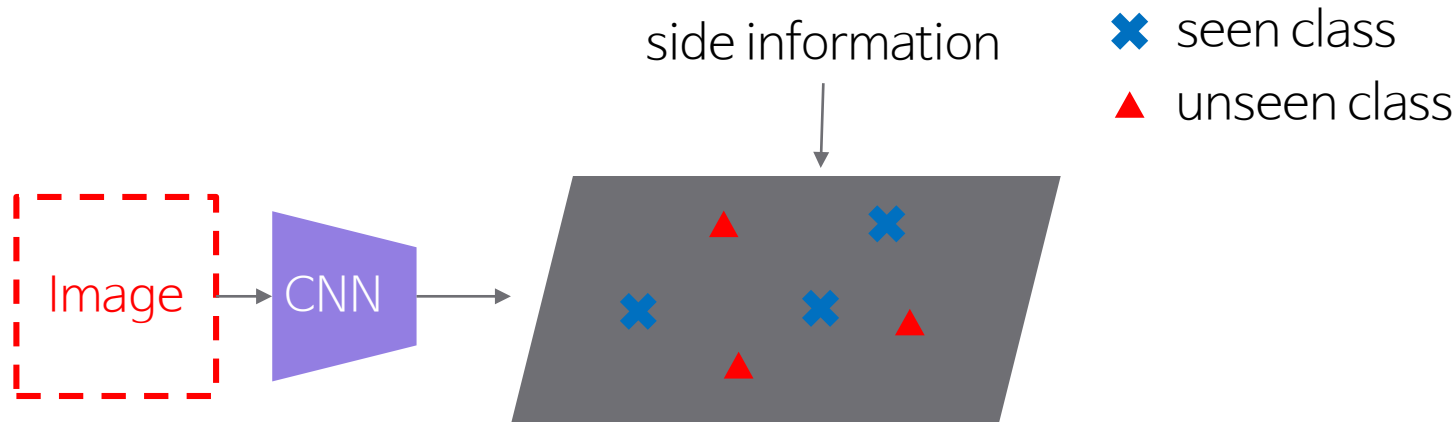
〈 Most existing zero-shot learning〉  
seen/unseen class 를  
semantic space 상의  
하나의 고정된 벡터 (fixed point) 로 임베딩



- [Introduction](#)
- VAE
- VZSL
- Experiments
- Results

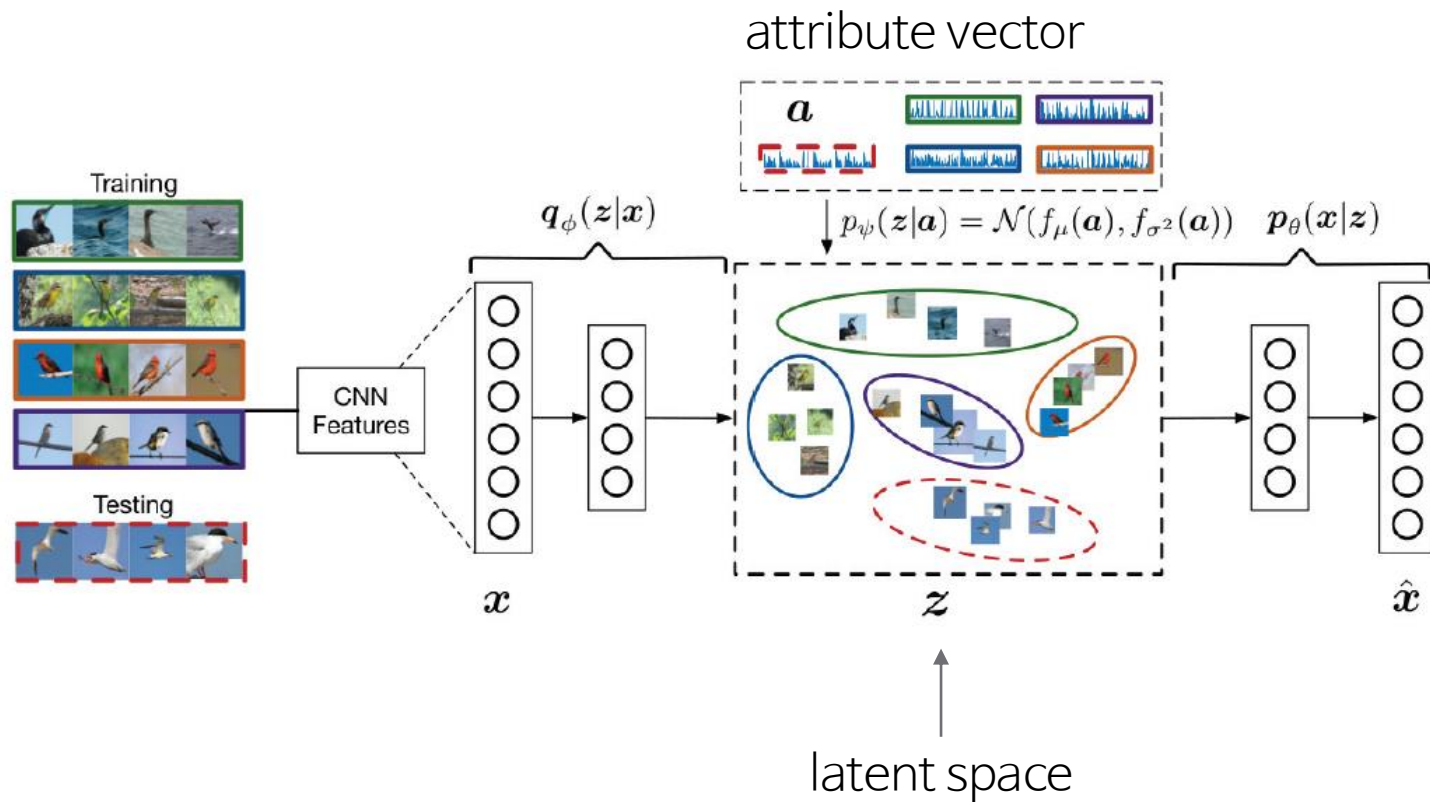
〈 Most existing zero-shot learning 〉

하나의 고정된 벡터(fixed point) 로 임베딩할 경우,  
같은 클래스 이미지 간의 다양성은 무시될 가능성이 있음



- [Introduction](#)
- VAE
- VZSL
- Experiments
- Results

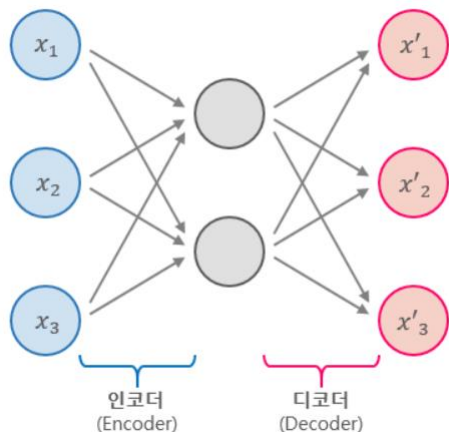
# Introduction



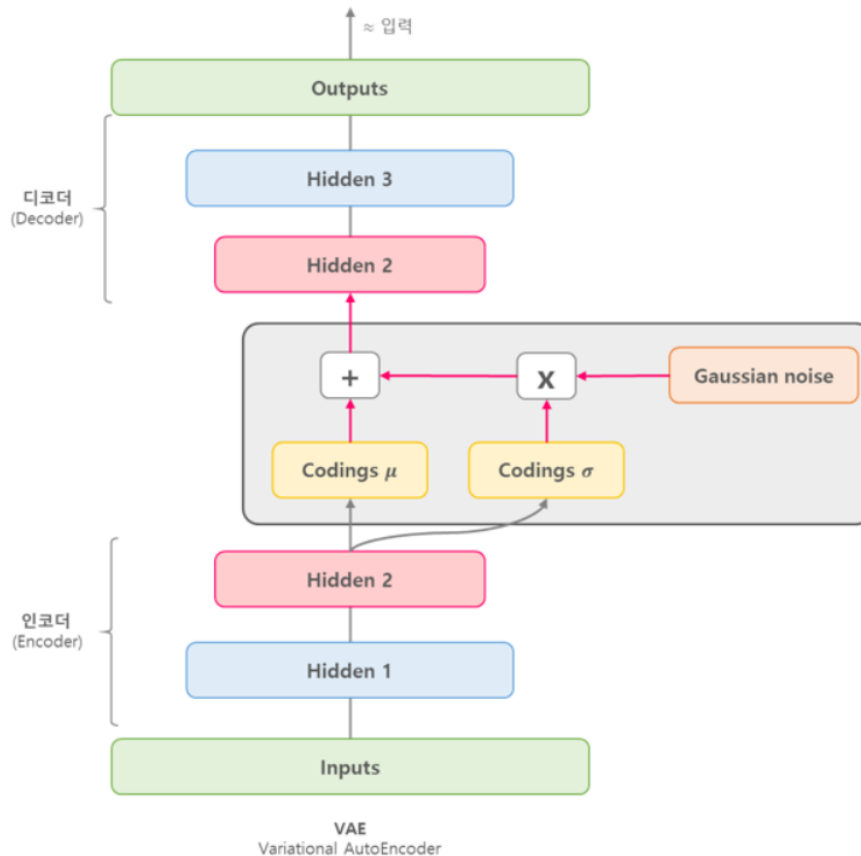
Zero-Shot  
Learning  
via Class-  
Conditioned  
Deep  
Generative  
Models

- Introduction
- VAE
- VZSL
- Experiments
- Results

# Variational Auto Encoder

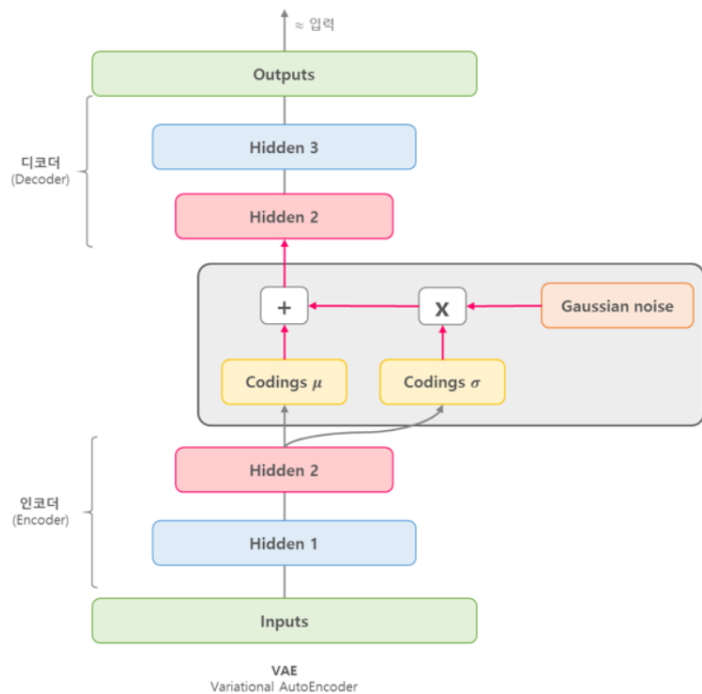


Auto encoder



Zero-Shot  
Learning  
via Class-  
Conditioned  
Deep  
Generative  
Models

- Introduction
- VAE
- VZSL
- Experiments
- Results

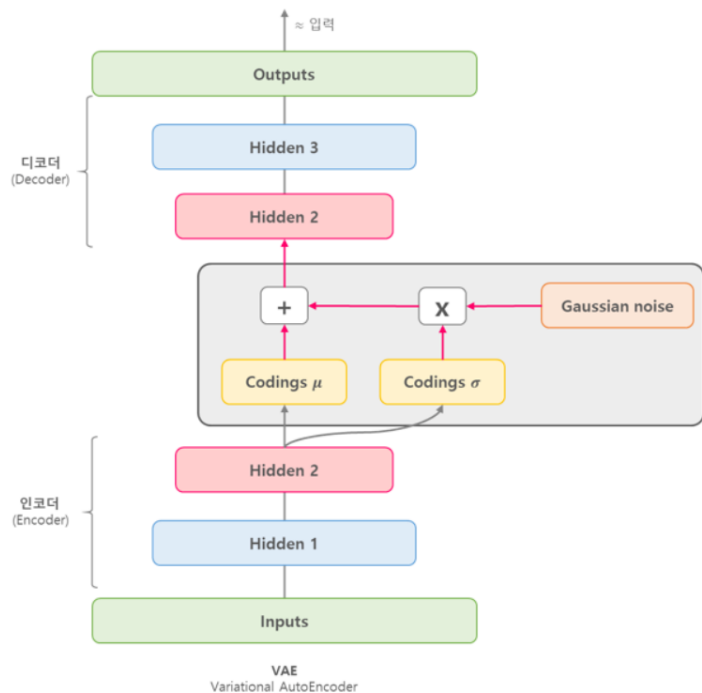


- 인코더 부분이 주어진 입력에 대한 코딩을 바로 만드는 방법이 아니라 평균 코딩( $\mu$ )과 표준편차 코딩( $\sigma$ )을 만들고 가우시안 분포에서 샘플링하는 방식.
- 학습하는 동안 손실함수가 가우시안 샘플의 형태를 가진 잠재변수 공간(latent space)으로 코딩을 이동시킴.

$$\mathcal{L}_{\theta, \phi}^V(x) = \mathbb{E}_{q_{\phi}(z|x)}[\log p_{\theta}(x|z)] - KL(q_{\phi}(z|x)||p_0(z))$$

- Introduction
- VAE
- VZSL
- Experiments
- Results

# Variational Auto Encoder



- VAE는 학습이 끝난 후 새로운 샘플을 가우시안 분포로부터 랜덤한 코딩을 샘플링해서 디코딩해서 생성할 수 있음.

prior

$$p_0(z) = \mathcal{N}(0, \mathbf{I})$$



$$\mathcal{L}_{\theta, \phi}^V(x) = \mathbb{E}_{q_{\phi}(z|x)}[\log p_{\theta}(x|z)] - KL(q_{\phi}(z|x) || p_0(z))$$

Zero-Shot Learning via Class-Conditioned Deep Generative Models

- Introduction
- VAE
- VZSL
- Experiments
- Results

- *attribute-specific* latent Gaussian

$$p_{\psi}(z_n|a_n) = \mathcal{N}(\mu(a_n), \Sigma(a_n))$$

$$\mu(a_n) = f_{\mu}(a_n) = W_{\mu}a_n$$

$$\Sigma(a_n) = \text{diag}(\exp(f_{\sigma}(a_n)))$$

$$f_{\sigma}(a_n) = W_{\sigma}a_n$$

- $W_{\mu}$ ,  $W_{\sigma}$  가 학습 완료되면, unseen class 의 attribute vector로 *attribute-specific* latent Gaussian 을 구할 수 있다.
- 이것을 이용하여 unseen class inference.

- Introduction
- VAE
- VZSL
- Experiments
- Results

$$\mathcal{L}_{\theta, \phi, \psi}(x, a) = \mathbb{E}_{q_{\phi}(z|x)}[\log p_{\theta}(x|z)] - KL(q_{\phi}(z|x) || p_{\psi}(z|a))$$



$$\mathcal{L}_{\theta,\phi,\psi}(x, a) = \mathbb{E}_{q_{\phi}(z|x)}[\log p_{\theta}(x|z)] - KL(q_{\phi}(z|x)||p_{\psi}(z|a))$$

- *attribute-specific* latent Gaussian 을 이용한 objective는  $q_{\phi}(z|x)$  이 분포가 class-specific한 분포인  $p_{\psi}(z|a)$ 에 다가가는 방향으로 유도.
- 하지만, ZSL에서 원하는 것은 classification 이므로 학습된  $q_{\phi}(z|x)$  분포끼리 서로 다른 클래스끼리는 일정 margin 만큼 떨어져있도록 학습하면 더 좋은 결과를 얻을 수 있음.
- Margin regularizer

$$\mathcal{L}_{\theta,\phi,\psi}(x, a) = \mathbb{E}_{q_{\phi}(z|x)}[\log p_{\theta}(x|z)] - \left[ KL(q_{\phi}(z|x)||p_{\psi}(z|a)) - R^* \right]$$

- $R^*$ 은 margin regularizer term 으로  $q_{\phi}(z|x)$  와 다른 모든 클래스의 latent space distribution 과의 minimum KL divergence 로 정의.

$$\begin{aligned} R^* &= \min_c \left\{ \text{KL} \left( q_\phi(z|x) \| p_\psi(z|\mathbf{A}_c) \right) \right\} \\ &= - \max_c \left\{ -\text{KL} \left( q_\phi(z|x) \| p_\psi(z|\mathbf{A}_c) \right) \right\} \end{aligned}$$

- 여기서  $c$  는 다른 모든 seen class.
- 이렇게 학습하면, true class 와 next best class 의 분포가 최대한 떨어져 있게 됨.
- 위 식은 미분 불가능이라 아래 근사식을 사용.

$$R = -\log \sum_{c=1}^S \exp \left( -\text{KL} \left( q_\phi(z|x) \| p_\psi(z|\mathbf{A}_c) \right) \right)$$

- Introduction
- VAE
- VZSL
- Experiments
- Results

- 최종 objective

$$\begin{aligned} \hat{\mathcal{L}}_{\theta, \phi, \psi}(x, a) \\ &= \mathbb{E}_{q_{\phi}(z|x)}[\log p_{\theta}(x|z)] - KL\left(q_{\phi}(z|x) \| p_{\psi}(z|a)\right) \\ &\quad - \lambda \log \sum_{c=1}^s \exp\left(-KL\left(q_{\phi}(z|x) \| p_{\psi}(z|\mathbf{A}_c)\right)\right) \end{aligned}$$

- Introduction
- VAE
- VZSL
- Experiments
- Results

- Dataset
  - AWA, CUB, SUN, ImageNet
- image feature extraction
  - VGG-19
  - 4096-dim vector
- class attribute vector
  - AWA, CUB, SUN 은 데이터셋에서 제공해주는 벡터.
  - ImageNet은 word2vec 임베딩 벡터.
- VAE model(encoder, decoder)
  - 2 layer MLP

- Introduction
- VAE
- VZSL
- [Experiments](#)
- Results

# Results

Method	AwA	CUB-200	SUN	Average	Method	ImageNet
(Lampert et al. 2014)	57.23	—	72.00	—	DeViSE (Frome et al. 2013)	12.8
ESZSL (Romera-Paredes et al. 2015)	75.32 ± 2.28	—	82.10 ± 0.32	—	ConSE (Norouzi et al. 2013)	15.5
MLZSC (Bucher et al. 2016)	77.32 ± 1.03	43.29 ± 0.38	84.41 ± 0.71	68.34	AMP (Fu et al. 2015b)	13.1
SDL (Zhang et al. 2016b)	80.46 ± 0.53	42.11 ± 0.55	83.83 ± 0.29	68.80	SS-Voc (Fu et al. 2016)	16.8
BiDiLEL (Wang et al. 2016a)	79.20	46.70	—	—		
SSE-ReLU (Zhang et al. 2015)	76.33 ± 0.83	30.41 ± 0.20	82.50 ± 1.32	63.08		
JFA (Zhang et al. 2016a)	81.03 ± 0.88	46.48 ± 1.67	84.10 ± 1.51	70.53		
SAE (Kodirov et al. 2017)	83.40	56.60	84.50	74.83		
GFZSL (Verma et al. 2017)	80.83	56.53	86.50	74.59		
VZSL#	84.45 ± 0.74	55.37 ± 0.59	85.75 ± 1.93	74.52	-	22.88
VZSL	<b>85.28 ± 0.76</b>	<b>57.42 ± 0.63</b>	<b>86.75 ± 2.02</b>	<b>76.48</b>	-	<b>23.08</b>

- AWA, CUB, SUN dataset 모두에서 성능 증가, 특히 ImageNet dataset 에서는 증가폭이 큼. (37.4 % 증가)
- most similar class 를 찾을 때 가장 가까운 single fixed point 를 찾는 것이 아니라 class distribution 으로부터 “most probable class” 를 찾기 때문에 성능이 높아진 것으로 보임.

- Introduction
- VAE
- VZSL
- Experiments
- Results

감사합니다