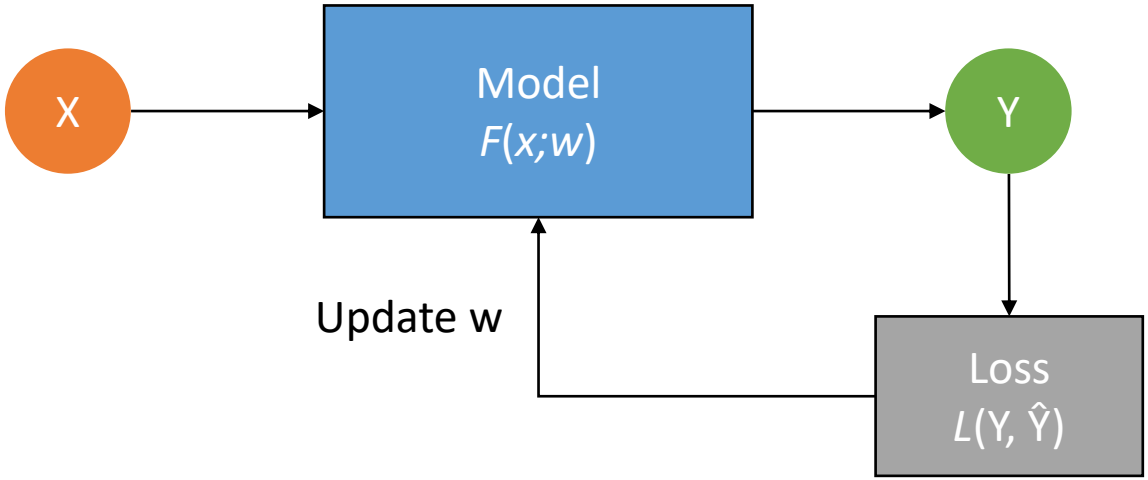
Loss function and cooperative learning

How ML works



- Statistical view
  - 우도(likelyhood)를 최대화하는 파라미터 탐색

$$arg\max_{w} P(Y|X;w)$$

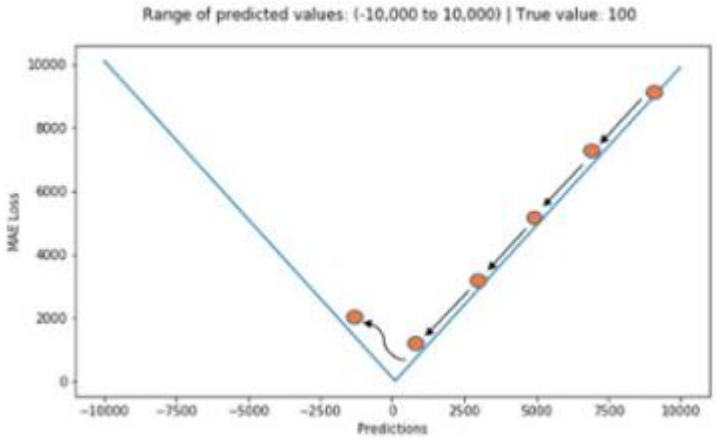
Loss function

- MAE (Mean Absolute Error)
- MSE (Mean Squared Error)
- RMSE (Rooted Mean Squared Error)
- Cross Entropy
- KLD

MAE

- Mean Absolute Error

$$L = |y - f(x)|$$

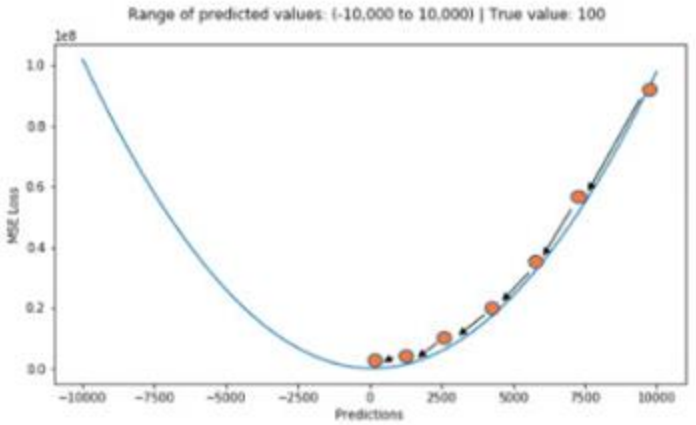


Prof. Hyerim Bae (hrbae@pusan.ac.kr)

MSE and RMSE

- Mean Squared Error

$$L = (y - f(x))^2$$



- Rooted Mean Squared Error

$$L = \sqrt{(y - f(x))^2}$$

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Huber Loss

- Combination of MSE and MAE

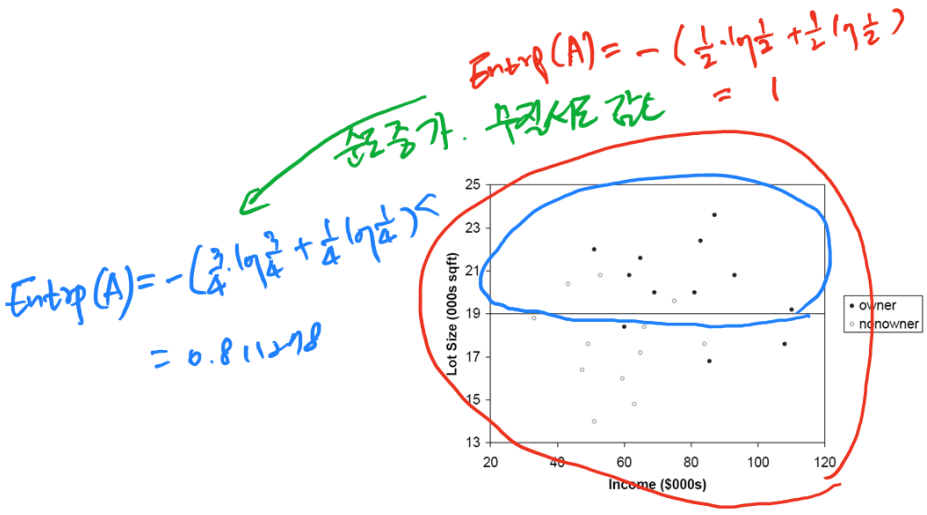
$$L = \begin{cases} \frac{1}{2}(y - f(x))^2, if\ |y - f(x)| \leq \delta \\ \delta|y - f(x)| - \frac{1}{2}\delta^2, otherwise \end{cases}$$

  - Quadratic for smaller error
  - Linear otherwise

Prof. Hyerim Bae (hrbae@pusan.ac.kr

Cross Entropy

- Entropy: (최소)정보량의 평균
  - $entropy = -\sum_{k=1}^{m} p_k \log_2 p_k$
  - 정보량 $I(x) = -\log p(x)$

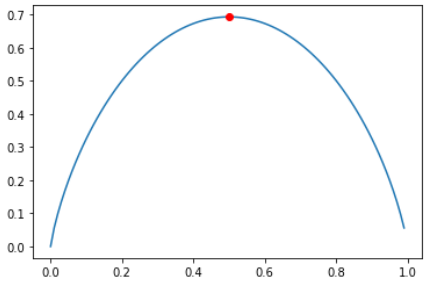


- Cross Entropy:
  - $H(P, Q) = \sum_i p_i \log_2 \frac{1}{q_i} = -\sum_i p_i \log_2 q_i$

- Binary CE

$$L = -y \cdot \log f(x) - (1 - y) \cdot \log(1 - f(x))$$

p: usually sigmoid function is used



- Multi-class CE

$$L = -\sum_i y_i \cdot \log p_i$$

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Cross entropy vs. MSE

Cross entropy	MSE
-범주형 데이터 예측 시 주로 사용 -각 클래스에 속할 확률을 의미하는 확률분포를 이용해 실제분포와의 엔트로피의 차이를 구한 것.	-수치형 데이터 예측 시 주로 사용 -정답값과 예측값의 차이를 제곱하고 평균을 내준 것.
딥러닝에서 쓰이는 대표적인 손실 함수	

Prof. Hyerim Bae (hrbae@pusan.ac.kr

KL-Divergence

- The difference between two distributions
- 어떤 이상적인 분포에 대해, 그 분포를 근사하는 다른 분포를 사용해 샘플링을 한다면 발생할 수 있는 정보 엔트로피 차이를 계산

$$D_{KL}(P||Q) = -\sum_{x}\big(P(x).logQ(x) - P(x).logP(x)\big) = H(P,Q) - H(P,P)$$

$H(P,P)$ *is the entropy of* P

$H(P,Q)$ *is the cross* $-$ *entropy of* P *and* Q

Prof. Hyerim Bae (hrbae@pusan.ac.kr

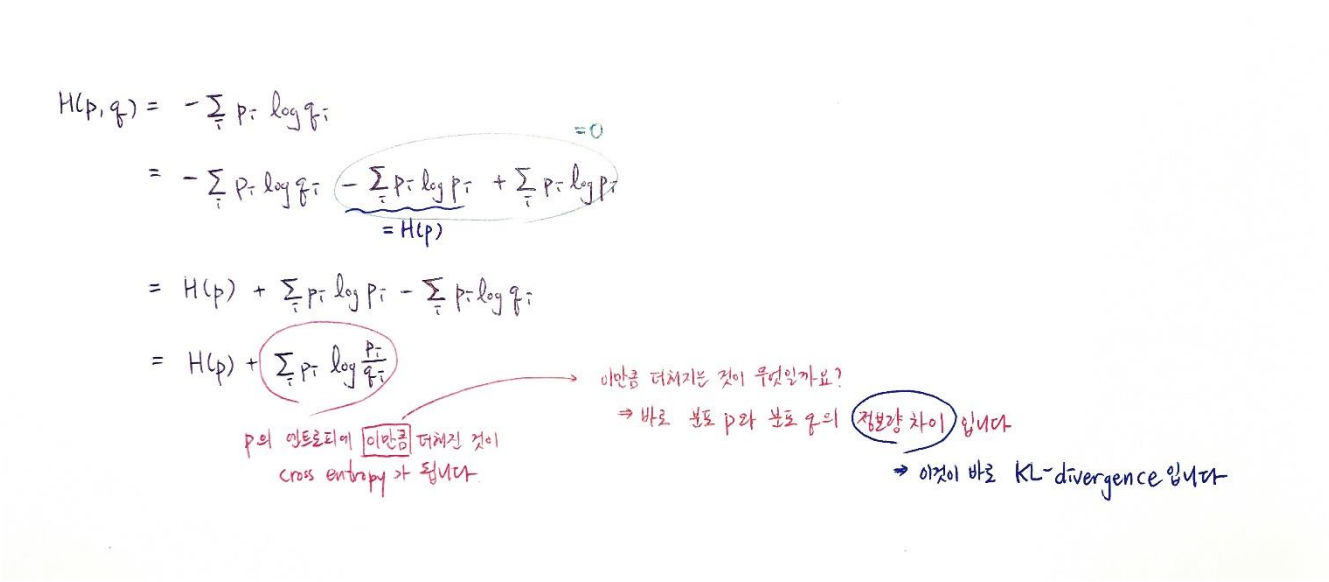
KLD(Kullback Leibler Divergence)

- In mathematical statistics, the Kullback–Leibler divergence,(also called relative entropy), is a measure of how one probability distribution is different from a second, reference probability distribution.

P: Distribution P represents the data, the observations, or a probability distribution precisely measured.

Q: Distribution Q represents instead a theory, a model, a description or an approximation of P

- The Kullback–Leibler divergence is then interpreted as the **average difference of the number of bits** required for encoding samples of P using a code optimized for Q rather than one optimized for P.



- 출처: https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

KLD(Kullback Leibler Divergence)

- $KL(p||q)$

  $= H(p,q) - H(p)$

$= -\int p(x) lnq(x) dx - (-\int p(x) lnp(x) dx) = -\int p(x) \ln\left\{\frac{q(x)}{p(x)}\right\} dx$

- $H(p,q) = -\int p(x) lnq(x) dx$
- $H(p) = -\int p(x) lnp(x) dx$
  - p : *true distribution*
  - q : *predicted distribution*
- $KL(p||q) \neq KL(q||p)$
- p와 q가 같다면 $KL(p||q) = 0$

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

example

- 실제 사면체 주사위의 확률분포 p= $(\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{1}{4})$

  추정한 사면체 주사위의 확률 분포 q = $(\frac{1}{2},\frac{1}{4},\frac{1}{8},\frac{1}{8})$라 가정.

  - 추정한 확률분포 q를 통해 코딩했을 경우
    - 코딩결과: (0, 10, 110, 111)
    - Entropy = Average code length = 2.25 => **Cross entropy!**

  - 실제 분포 p를 통한 최적의 코딩을 했을 경우
    - 코딩결과: (00,01,10,11)
    - Entropy = Average code length = 2

  모델링한 p와 q가 달라서 엔트로피 차이가 발생 => **KL - divergence**

  $(-\sum_x p(x)\log_2 q(x)) - (-\sum_x p(x)\log_2 p(x)) = -\sum_x p(x)\log_2 \frac{q(x)}{p(x)}$

  = 2.25 – 2 = 0.25

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

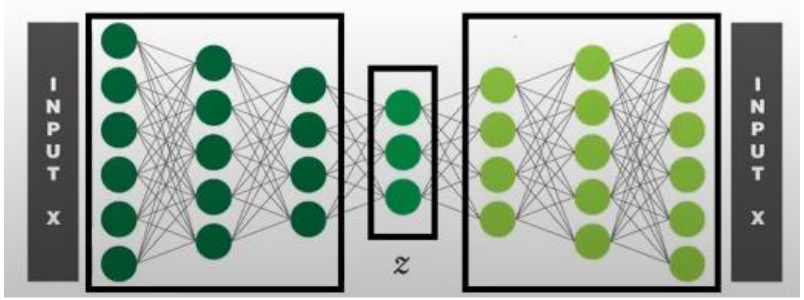
- VAE
  - The structure is the same as autoencoder
  - Finding the distribution of z

$$x = g_\theta(z)$$

$$p(x) = p(x|z)$$

$$p(x) = E_{z\sim p_\theta(x)}[p(x|z)]$$

$$p(x) = E_{z\sim p_\theta(z|x)}[p(x|z)] \cong E_{z\sim q_\emptyset(z|x)}[p(x|z)]$$



Prof. Hyerim Bae (hrbae@pusan.ac.kr)

ELBO

$$\log(p(x)) = \log\left(\int p(x,z)dz\right) = \log\left(\int p(x|z)p(z)dz\right)$$

$$\log(p(x)) = \log\left(\int p(x|z)\frac{p(z)}{q_\phi(z|x)}q_\phi(z|x)dz\right)$$

$$\log(p(x)) \geq \int \log\left(p(x|z)\frac{p(z)}{q_\phi(z|x)}\right)q_\phi(z|x)dz$$ By Jensen's Inequality

$$\log(p(x)) \geq \int \log(p(x|z))q_\phi(z|x)dz - \int \log\left(\frac{q_\phi(z|x)}{p(z)}\right)q_\phi(z|x)dz$$

$$\boxed{ELBO(\phi) = \mathbb{E}_{q_\phi(z|x)}[\log(p(x|z))] - KL(q_\phi(z|x)\|p(z))}$$

$$\int \log(p(x|z))q_\phi(z|x)dz = \mathbb{E}_{q_\phi(z|x)}[\log(p(x|z))]$$

$$\int \log\left(\frac{q_\phi(z|x)}{p(z)}\right)q_\phi(z|x)dz = KL(q_\phi(z|x)\|p(z))$$

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

KLD

$$\log(p(x)) = \int \log(p(x)) q_\phi(z|x) dz$$

$$\log(p(x)) = \int \log\left(\frac{p(x,z)}{p(z|x)}\right) q_\phi(z|x) dz$$

$$= \int \log\left(\frac{p(x,z)}{q_\phi(z|x)} \cdot \frac{q_\phi(z|x)}{p(z|x)}\right) q_\phi(z|x) dz$$

$$= \underbrace{\int \log\left(\frac{p(x,z)}{q_\phi(z|x)}\right) q_\phi(z|x) dz}_{\text{ELBO}} + \underbrace{\int \log\left(\frac{q_\phi(z|x)}{p(z|x)}\right) q_\phi(z|x) dz}_{D_{KL}}$$

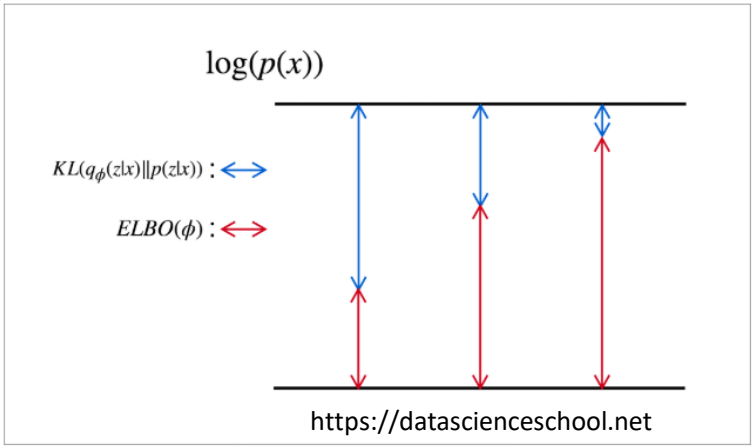
우리가 minimize하고 싶은 것

$$ELBO = \int_z \log\{ \frac{p(z,x)}{q_\phi(z|x)} \} q_\phi(z|x)\, dz = \int_z \log\{ \frac{p(z)\, p(x|z)}{q_\phi(z|x)} \} q_\phi(z|x)\, dz$$

$$= \int_z \log\{ p(x|z) \} q_\phi(z|x)\, dz - \int_z \log\{ \frac{q_\phi(z|x)}{p(z)} \} q_\phi(z|x)\, dz = \underbrace{E_{q_\phi(z|x)}[\log\{p(x|z)\}] - D_{KL}(q_\phi(z|x) \,\|\, p(z))}_{\text{ELBO; Derivation1 의 것과 동일}}$$

$$\therefore \quad \log p(x) = E_{z\sim q(z|x)}[\log p(x|z)] - D_{KL}(q(z|x)||p(z)) + D_{KL}(q(z|x)||p(z|x))$$
$$= ELBO + D_{KL}(q(z|x)||p(z|x))$$

$$L = -E_{z\sim q(z|x)}[\log p(x|z)] + D_{KL}(q(z|x)||p(z))$$



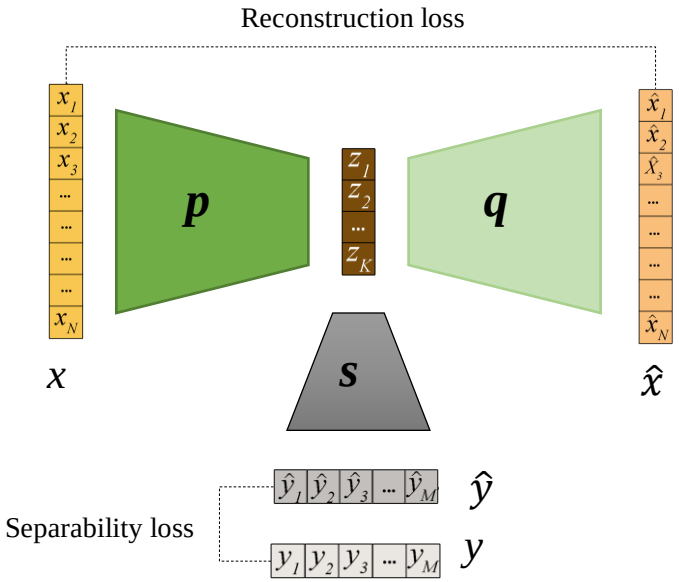
Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Cooperative and Non-cooperative Autonets: from Dimensionality Reduction to Classification

Imam Mustafa Kamal & Hyerim Bae

Super-encoder

- Super-encoder consists of encoder (***p***), separator (***s***), decoder (***q***) networks.
  The term super-encoder comes from a supervised encoder and superior dimensionality reduction.



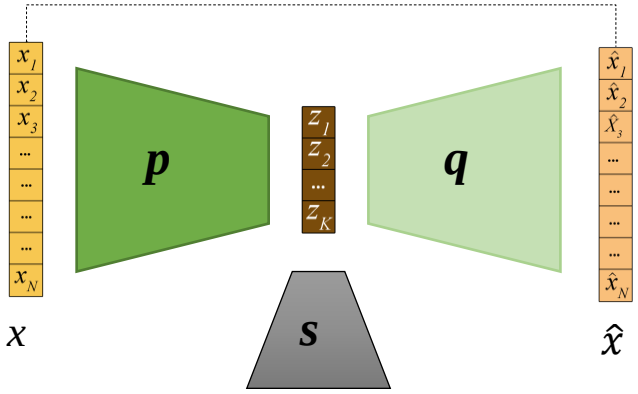
- ***p*** maps from ***x*** to latent-code ***z***
  $z = p(x)$
- ***s*** asses the ***z*** separability based on label ***y***
  $\hat{y} = s(p(x))$
- ***q*** reconstructs latent-code *z* to original data ***x***
  $\hat{x} = q(p(x|y))$

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Cooperative Super-encoder

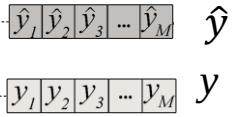
Reconstruction loss

$$\mathcal{L}_{enc}(\phi, \psi) = \mathbb{E}_{x,y}[\|y - s_\psi(q_\phi(x))\|_1]$$



Separability loss

$$\mathcal{L}_{dec}(\phi, \psi, \theta) = \mathbb{E}_{x,y}[\|x - p_\theta(q_\phi(x|y))\|_1]$$



θ Decoder parameter

ψ Separator parameter

ϕ Encoder parameter

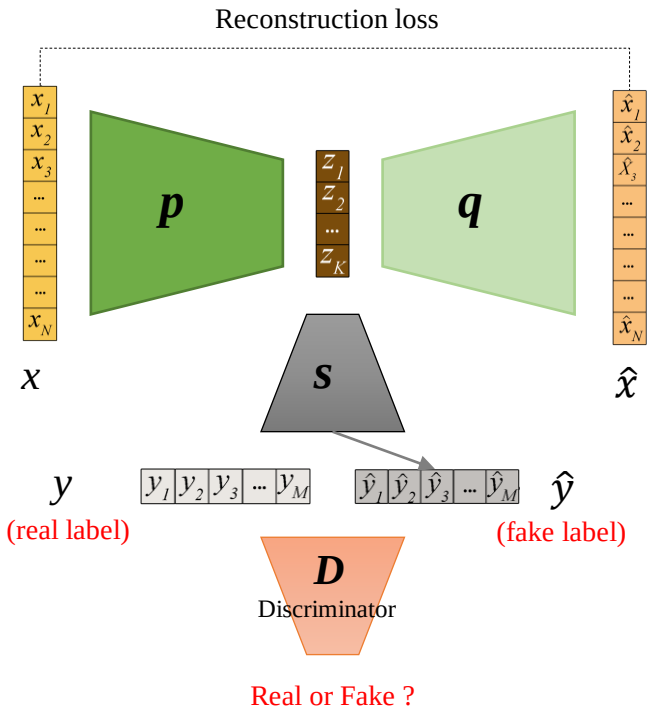
Total / cooperative loss

$$\mathcal{L}(\phi, \psi, \theta) = \arg\min_{\phi,\psi,\theta} \alpha\mathcal{L}_{enc}(\phi, \psi) + \beta\mathcal{L}_{dec}(\phi, \psi, \theta)$$

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Non-cooperative Super-encoder

$$\min_{p,q,s} \max_{D} V(\{p,q,s\},D) = \mathbb{E}_{x,y}[\log D(y)] + \mathbb{E}_{x,y}[\log(1 - D(s(q(x)))] + \mathbb{E}_{x,y}[p(q(x|y))]$$



- ***p*** and ***s*** try to minimize the discrepancy between real and fake label
- While ***D*** wants to maximize the different between real and fake label

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Super-encoder: result

Table 1. Performance of latent-code separability by encoder and reconstruction error by the decoder

Model	Performance evaluation							
	Encoder: latent-code separability (accuracy)				Decoder: reconstruction error			
	MNIST	Fashion-M	SVHN	CIFAR-10	MNIST	Fashion-M	SVHN	CIFAR-10
SE (co)	**0.9893**	**0.9257**	**0.8813**	0.7883	**0.0123**	**0.0129**	0.0336	0.0162
SE*(co)	0.9786	0.8735	0.7569	0.5553	0.0198	0.0183	0.0405	0.0163
SE (nc)	0.9898	0.9161	0.8489	**0.8015**	0.0132	0.0129	0.0356	0.0163
SE*(nc)	0.9742	0.8616	0.7435	0.5551	0.0262	0.0220	0.0413	0.0164
AE	0.9262	0.8493	0.4632	0.3933	0.0243	0.0586	0.0208	**0.0148**
DAE	0.9684	0.8411	0.3892	0.3093	0.0150	0.0610	0.0404	0.0155
VAE	0.9657	0.8267	0.2868	0.3011	0.0130	0.0145	0.0297	0.0161
DVAE	0.9475	0.7866	0.3064	0.2993	0.0172	0.0204	0.0500	0.0163
β-VAE	0.9493	0.7899	0.2256	0.2895	0.0158	0.0173	0.0386	0.0164
AAE	0.9630	0.8388	0.3877	0.3181	0.0644	0.0576	0.0903	0.0161
DAAE	0.9528	0.2740	0.2156	0.2976	0.0230	0.6060	0.0144	0.0153
iDAAE	0.9588	0.2737	0.2180	0.2991	0.0928	0.6035	**0.0141**	0.0156

* Semi-supervised training with only 20% of label

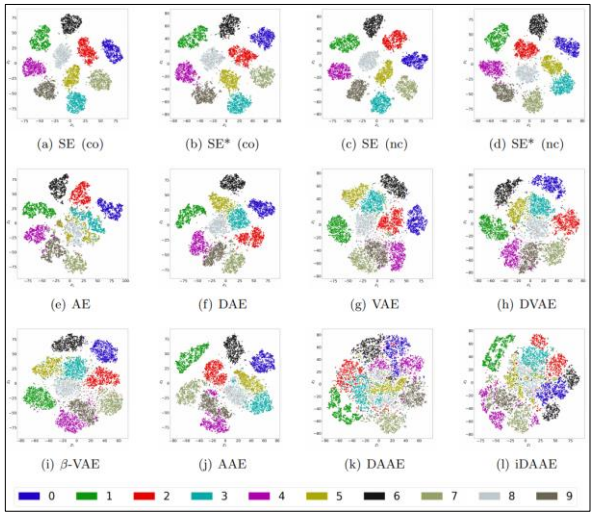
- SE = Super-encoder
- SE* = Super-encoder with semi-supervised learning
- (co) = SE with Cooperation
- (nc) = SE with Non-cooperation

Prof. Hyerim Bae (hrbae@pusan.ac.kr

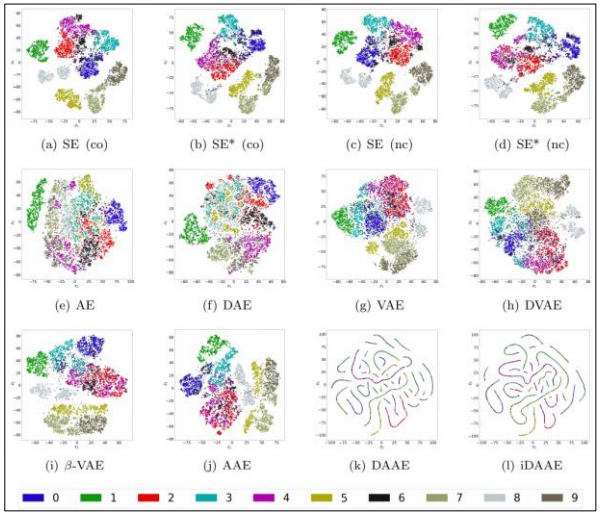
Latent-code Separability (1)

Compare with existing autoencoder models

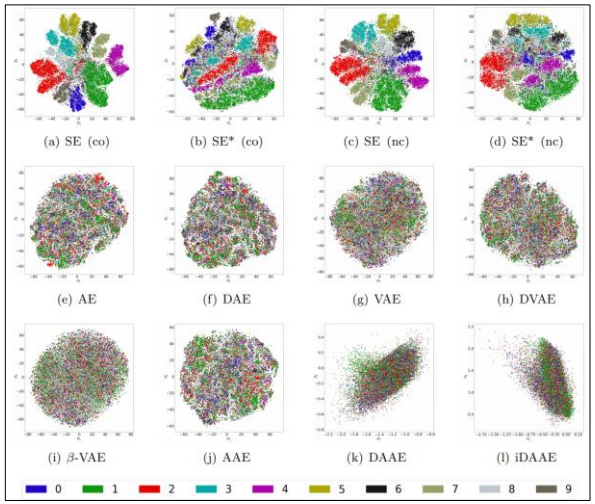
(a). MNIST



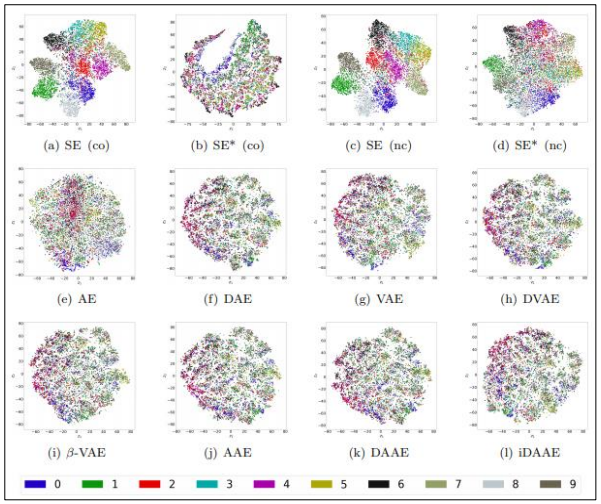
(b). Fashion-MNIST



(c). SVHN



(d). CIFAR-10

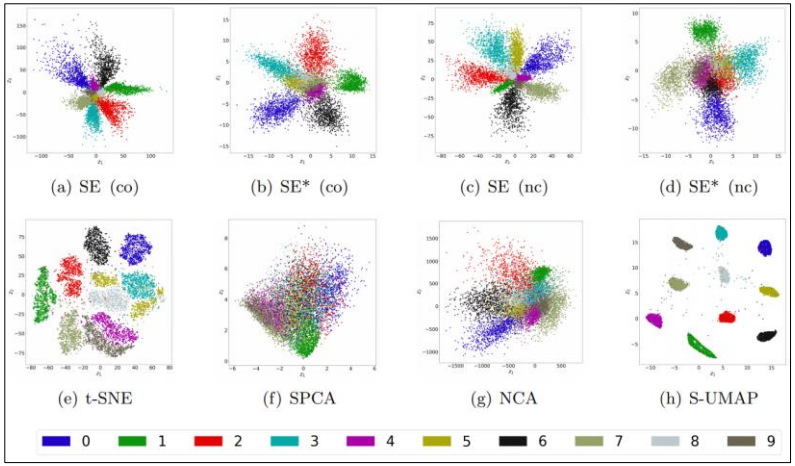


Prof. Hyerim Bae (hrbae@pusan.ac.kr)

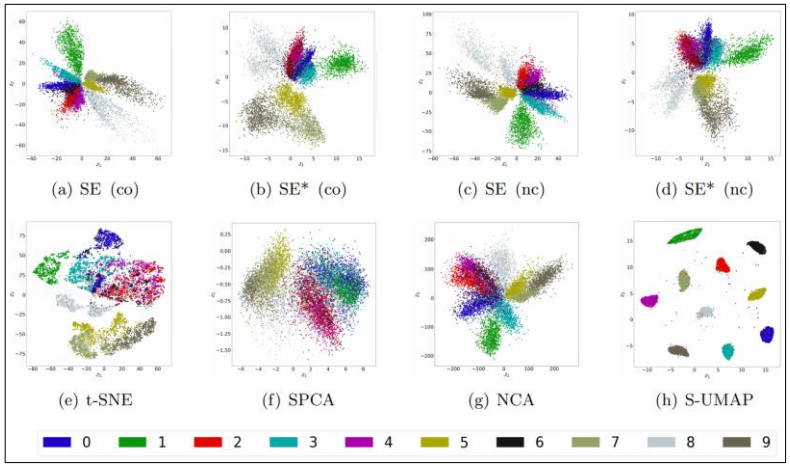
Latent-code Separability (2)

Compare with existing supervised and non-supervised dimensionality-reduction model

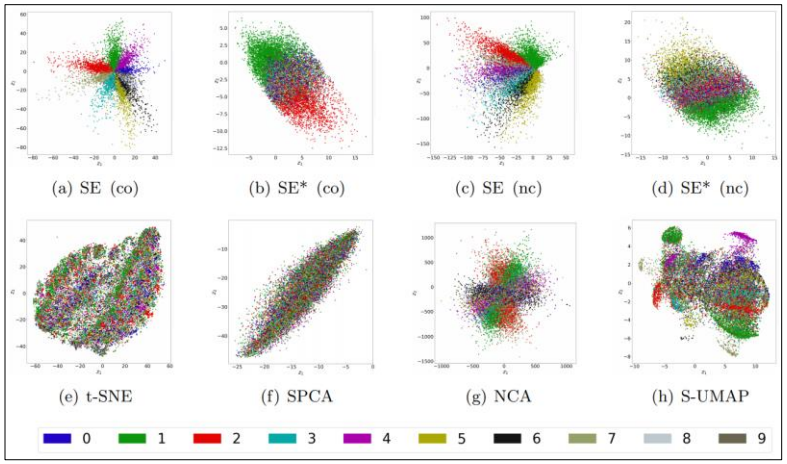
(a). MNIST



(b). Fashion-MNIST



(c). SVHN



(d). CIFAR-10

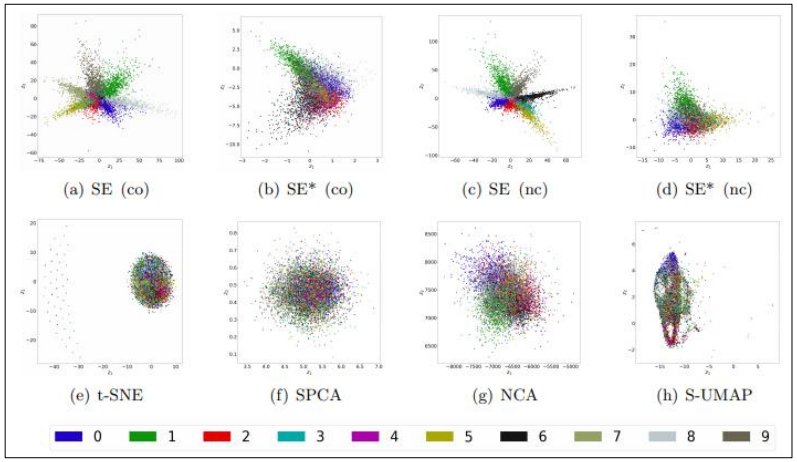
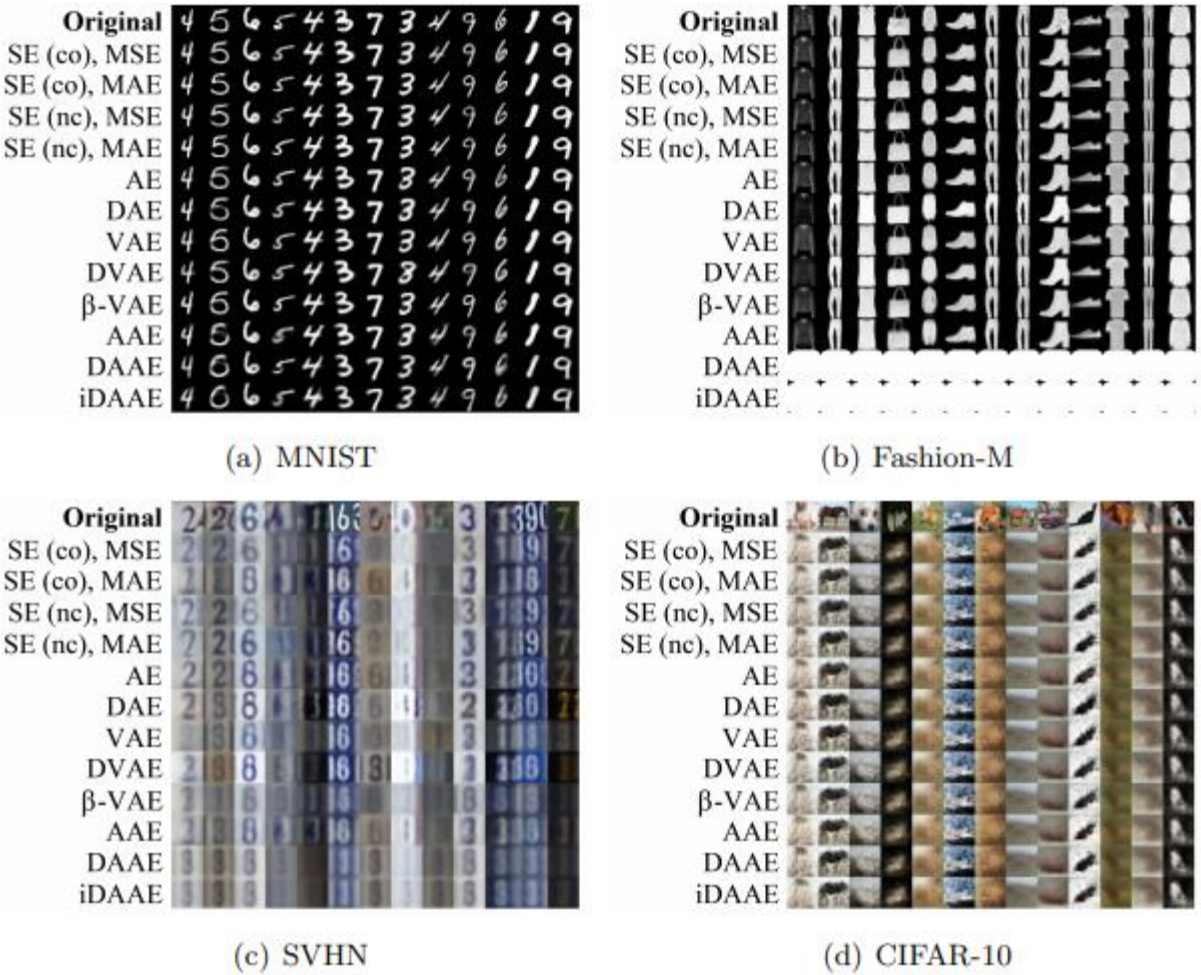


Image Reconstruction (Generation)

Compare with existing autoencoder models



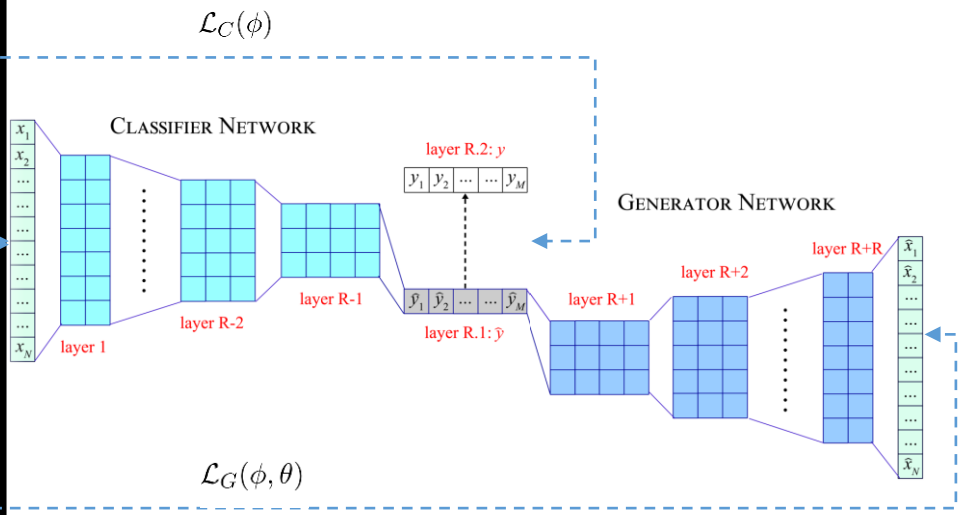
(a) MNIST (b) Fashion-M

(c) SVHN (d) CIFAR-10

Cooperative Auto-classifier

- Auto-classifier comprises classifier (***C***) and generator (***G***) networks

The term auto-classifier stands for an automatic classifier,
where the classifier is devised to automatically deal with or exploit the dimensionality reduction.



- ***C*** estimates label (***y***) from input ***x***,
- ***G*** must reconstruct label ***y*** to original data ***x***,
- Implicitly, label ***y*** also represents the latent representation of original data ***x***.
- So, ***C*** is two-fold model: both as a classifier and dimensionality-reduction model.

- Strict cooperation (AC (co) v1) :

$$\mathcal{L}(\phi,\theta) = \arg\min_{\phi,\theta} \mathcal{L}_C(\phi) + \mathcal{L}_G(\phi,\theta) + |\mathcal{L}_C(\phi) - \mathcal{L}_G(\phi,\theta)|$$

- Adjusted cooperation (AC (co) v2):

$$\mathcal{L}(\phi,\theta) = \arg\min_{\phi,\theta} \lambda(\mathcal{L}_C(\phi)) + \mathcal{L}_G(\phi,\theta)$$

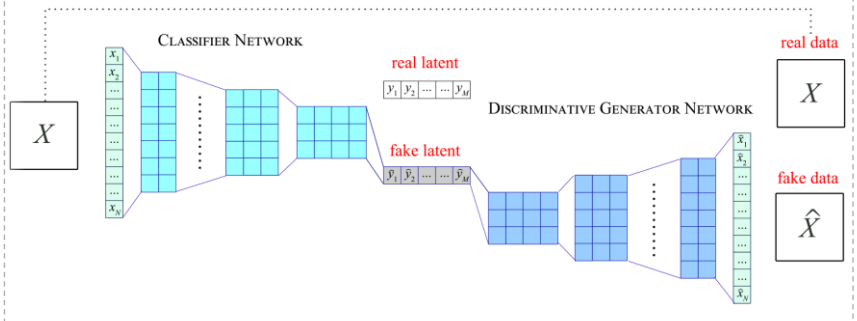
Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Non-cooperative Auto-classifier

- Two types of games:

Classifier vs. Generator (AC (nc) v1)

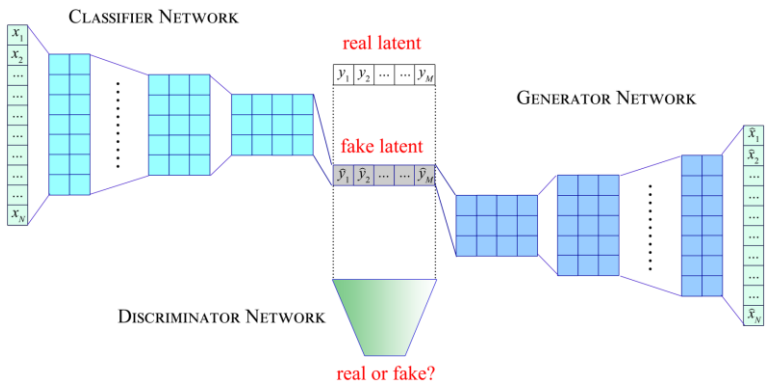
$$\min_{C} \max_{G^+} V(C, G^+) = \mathbb{E}_{x,y}[\log G^+(y)] + \mathbb{E}_{x,y}[\log(1 - G^+(C(x)))]$$



- Discriminative Generator: distinguish the real and fake image
- The real latent must generate a real image
- The fake latent must generate a fake image

Classifier vs. Discriminator (AC (nc) v2)

$$\min_{C,G} \max_{D} V(\{C, G\}, D) = \mathbb{E}_{x,y}[\log D(y)] + \mathbb{E}_{x,y}[\log(1 - D(C(x)))] + \mathbb{E}_{x,y}[G(C(x))]$$



- Discriminator: distinguish the real and fake latent-code

Auto-classifier: Result (1)

- AC = auto-classifier
- (co) = cooperative learning
- (nc) = non-cooperative learning
- AC(co) v1 = strict losss
- AC (co) v2 = adjusted loss
- AC (nc) v1 = via generator
- AC (nc) v2 = via discriminator

Table 2. Classification performance comparison of auto-classifier and some outstanding classifier models

Model	Classification accuracy (MSE)			
	MNIST	Chest X-Ray	Fashion-M	CIFAR-10
AC (co) v1, MSE	**0.9962**	**0.8654**	**0.9408**	0.8201
AC (co) v1, MAE	0.9944	0.8445	0.9218	0.8014
AC (co) v2, MSE	0.9959	0.8590	0.9392	0.8192
AC (co) v2, MAE	0.9939	0.8654	0.9235	0.8021
AC (nc) v1, MSE	0.9958	0.8718	0.9378	0.8208
AC (nc) v1, MAE	0.9955	0.8670	0.9181	0.8091
AC (nc) v2, MSE	0.9958	0.8718	0.9390	0.8238
AC (nc) v2, MAE	0.9933	0.9933	0.9241	0.8066
Independent CNN	0.9951	0.8429	0.9360	0.8017
Maxout Networks (Goodfellow et al. [109])	0.9400	N/A	N/A	**0.8600**
VAE (M1 + M2) (Makhzani et al. [40])	0.9904	N/A	N/A	N/A
VAT (Miyato et al. [110])	0.9936	N/A	N/A	N/A
CatGAN (Springenberg[111])	0.9909	N/A	N/A	0.8042
Task-Oriented GAN [90]	0.9312	N/A	N/A	N/A
Ladder Networks (Rasmus et al. [112])	0.9943	N/A	N/A	0.7960
SRC (Xie et al. [113])	0.8595	N/A	N/A	N/A
CRC (Xie et al. [113])	0.8325	N/A	N/A	N/A
KENRR (Linear) (Xie et al. [113])	0.8770	N/A	N/A	N/A
KENRR (POLY) (Xie et al. [113])	0.8795	N/A	N/A	N/A
KENRR (RBF) (Xie et al. [113])	0.8795	N/A	N/A	N/A
KENRR (Hellinger) (Xie et al. [113])	0.8845	N/A	N/A	N/A

Auto-classifier: Result (2)

Table 3. Performance measures of classification comparison among independent classifier, cooperative and non-cooperative classifier

Dataset	F1 score				
	Independent CNN	AC (co) v1	AC (co) v2	AC (nc) v1	AC (nc) v2
MNIST	0.9950	**0.9962**	0.9959	0.9957	0.9957
Chest X-Ray	0.8233	**0.8506**	0.8449	0.8615	0.8587
Fashion-M	0.9360	**0.9409**	0.9391	0.9376	0.9389
CIFAR-10	0.8012	0.8186	0.8185	0.8215	**0.8222**
Dataset	Precision				
	Independent CNN	AC (co) v1	AC (co) v2	AC (nc) v1	AC (nc) v2
MNIST	0.9950	**0.9962**	0.9958	0.9957	0.9957
Chest X-Ray	0.8102	**0.8393**	0.8359	0.8573	0.8487
Fashion-M	0.9360	**0.9408**	0.9392	0.9378	0.9390
CIFAR-10	0.8017	0.8201	0.8192	0.8208	**0.8238**
Dataset	Recall				
	Independent CNN	AC (co) v1	AC (co) v2	AC (nc) v1	AC (nc) v2
MNIST	0.9951	**0.9962**	0.9960	0.9958	0.9958
Chest X-Ray	0.8520	**0.8712**	0.8596	0.8668	0.8753
Fashion-M	0.9362	**0.9412**	0.9392	0.9378	0.9391
CIFAR-10	0.8100	0.8207	0.8208	**0.8269**	0.8222

Cooperative and non-cooperative networks are more robust than standalone CNN (classifier)

Prof. Hyerim Bae (hrbae@pusan.ac.kr

Reference

- https://ratsgo.github.io/deep%20learning/2017/09/24/loss/
- https://www.analyticsvidhya.com/blog/2019/08/detailed-guide-7-loss-functions-machine-learning-python-code/