

## Recap:

Last lecture  $\Rightarrow$  tricks to improve performance

① Ensemble  $\rightarrow$  average over multiple models

② Dropout  $\rightarrow$  can be viewed as stochastic version of ensemble  
 $\updownarrow$   
cooperate with Neural Networks

③ Batch Normalization  $\rightarrow$  Decouple

④ Data Augmentation  $\rightarrow$  change-invariant prior  
 $\rightarrow$  more data, less generalization gap

⑤ Learning Rate Decay Scheme

$\downarrow$   
 $\left\{ \begin{array}{l} \text{High Learning rate at beginning stage} \\ \text{Then decay over training} \end{array} \right.$

## ⑥ Adversarial Training

Defn: Robust

$$\hat{f}(x') = \hat{f}(x) \quad \text{for } \forall \|x' - x\| \ll 1$$

$\downarrow$   
small perturbation

DSA5204 Lec 8

Continuous topic: (one last topic)

$\rightarrow$  Adversarial Training  $\leftrightarrow$  adversarial example

① Definition (adversarial example)

given parameter  $\hat{\theta}$ , adversarial example is the worst example

$$x' = \arg \max_{z: \|z - x\| \leq \delta} L(\hat{y}(z; \hat{\theta}), y)$$

② Definition (adversarial training)

How can we reduce the effect of "Adversarial example"?

$\downarrow$

$$\boxed{\text{1-sample}} \quad \min_{\theta} \max_{z: \|x - z\| \leq \delta} L(\hat{y}(z; \theta), y)$$

can be achieved  
by gradient ascent

multi-sample  $\min_{\theta} \frac{1}{N} \sum_{i=1}^N \max_{z_i: \|x_i - z_i\| \leq \delta} L(\hat{y}(z_i; \theta), y)$

### ③ Algorithm (Fast Gradient Sign Method [FGSM])

→ For  $k = 0, 1, \dots$

$$z_0 = x$$

find the "Adversarial Example"

gradient ascent in input space { For  $j = 0, 1, \dots, J-1$

$$z_{j+1} = z_j + \epsilon_2 \text{sign}(\nabla_z L(\hat{y}(z_j; \theta_k), y))$$

$$\theta_{k+1} = \theta_k - \epsilon_1 \nabla_{\theta} L(\hat{y}(z_j; \theta_k), y)$$

in argmax

Here,  $\delta = J \times \epsilon_2$  (we want  $\|z - x\| \leq \delta$ )

Rmk: Generally speaking, we do not need to pre-train model first, then find adversarial examples and store them, lastly re-train model.

Instead, we modify the loss function and train model directly.

$$\min_{\theta} \max_{z: \|z-x\| \leq \delta} L(\hat{y}(z; \theta), y)$$

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$$

Something like Data Augmentation

### ④ 1-D Example

Model:  $\hat{y}(x; \theta) = \theta x$

Data  $\{x=1, y=0\} = \mathcal{D}$

Loss function:  $L(\hat{y}, y) = \begin{cases} \frac{1}{2} (\hat{y} - y - \text{sign}(\hat{y} - y))^2 & \text{if } |\hat{y} - y| > 1 \\ 0 & \text{if } |\hat{y} - y| \leq 1 \end{cases}$



Then 
$$L(\hat{y}, y) = \begin{cases} \frac{1}{2} (\theta - \text{sign}(\theta))^2 & |\theta| > 1 \\ 0 & |\theta| \leq 1 \end{cases}$$

↓  
Trivial Loss, via GD,  $\hat{\theta}_{GD}^{\infty} = 1$  if we start from  $\theta^{(0)} > 1$

→ Adversarial Loss 
$$L_{\text{adv}}(\theta) = \max_{z: \|1-z\| \leq \delta} L(\hat{y}(z, \theta), y)$$

$$= \max_{z: \|1-z\| \leq \delta} \frac{1}{2} (\theta z - \text{sign}(\theta z))^2 \mathbb{1}_{\{|\theta z| \geq 1\}}$$

$$= \frac{1}{2} (\theta(1+\delta) - 1)^2 \mathbb{1}_{\{|\theta(1+\delta)| \geq 1\}}$$

Model  
 $y = \theta x$

$y=1$  and  $x \rightarrow 0$

⇒ we want  $\theta \approx 0$

⇒  $\hat{\theta}_{\text{adv}}^{\infty} = \frac{1}{1+\delta}$  is more close to 0

and  $L_{\text{adv}}(\hat{\theta}_{GD}^{\infty}) = \frac{\delta^2}{2} > 0$

Today's topic: Unsupervised Learning + Semi-supervised Learning

1. Task

a) Supervised Learning → input:  $x$       output:  $y$

GT:  $f^*$

b) Unsupervised Learning  $\rightarrow$  no output!



learn some task-agnostic patterns

① Dimensionality Reduction

② Generative Model

③ Clustering

④ Density Estimation

$$z \rightarrow \boxed{f(z)}$$

c) Semi-supervised Learning

## 2. PCA

- ① Design Matrix  $X \in \mathbb{R}^{N \times d}$
- ② Covariance Matrix  $S = \frac{1}{N} X^T X$
- ③ Eigenvalue Decomp.  $S = U \Sigma U^T$



$$SU = U \Sigma$$

$$\boxed{U = (u_1, \dots, u_d)}$$

$$\underline{u_i \in \mathbb{R}^d}$$

$\rightarrow$  m Principle component direction

$$\underline{U_m = (u_1, \dots, u_m)} \quad m < d$$



$$\boxed{Z_m = X U_m \in \mathbb{R}^{N \times m}} \rightarrow \underline{\text{Principle Scoring}}$$

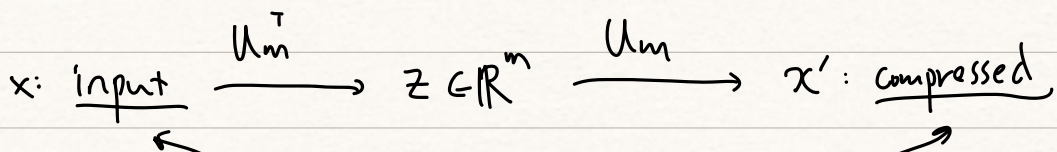
Reconstruction : 
$$\underline{X \approx X U_m U_m^T = Z_m \cdot U_m^T \in \mathbb{R}^{N \times d}}$$
$$= Z_m \begin{pmatrix} u_1^T \\ \vdots \\ u_m^T \end{pmatrix}$$

explained variance : 
$$\boxed{\sum_{i=1}^m \lambda_i}$$

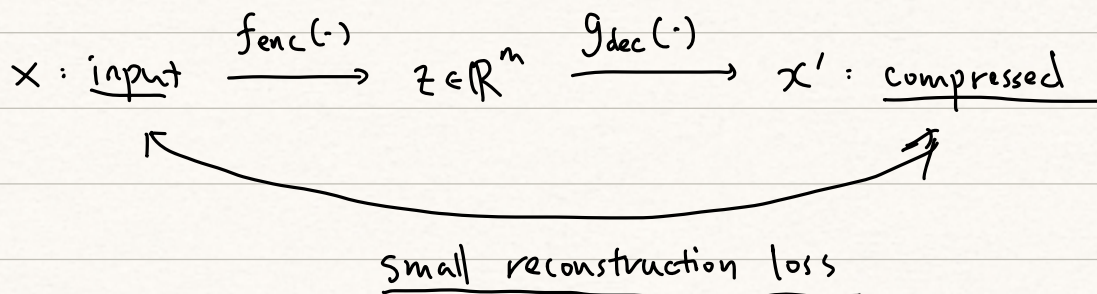


### 3. Auto-Encoder (AE)

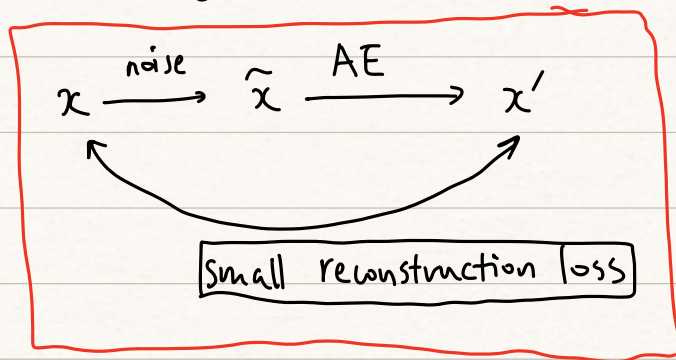
Re-cap: PCA framework



$\Rightarrow$  Generalization (AE): small reconstruction loss



"Denoising AE"



### 4. Semi-supervised Learning

a)  $\left\{ \begin{array}{l} \text{transductive} \\ \text{inductive} \end{array} \right.$  task

b) Method to label unlabelled data

① Naive approach (Self-learning)

train model  $\rightarrow$  label unlabelled  $\rightarrow$  re-train

## ② Label propagation

Rnk: different from some clustering algorithms like K-means etc.