Quantile Regression

원자력 인공지능 미니석사 과정

3.28 (화)

인공지능응용연구실 류승형 선임연구원

수업 자료

- https://github.com/shryu8902/KAERI_mini_BS

목차

- 불확실성이란?
- Uncertainty Quantification 리마인드
- Quantile regression에 대해서
- 확률적 예측 결과에 대한 평가지표
- 코드로 보는 Quantile regression

설문 조사

- 내가 다루는 데이터는?

1. 이미지 2. 텍스트 3. 센서 데이터 (시계열 포함)

- 내가 다루는 문제는?

1. 분류(Classification) 2. 회귀(Regression) 또는 예측 3. 제어

불확실하다는 것?

- 결정론적으로 결과 도출이 안 되는 상황
  - 내일은 비가 온다. (??%)
  - 내일은 수요일이다. (100%)
- 결과가 확률 분포로서 존재

불확실하다는 것?

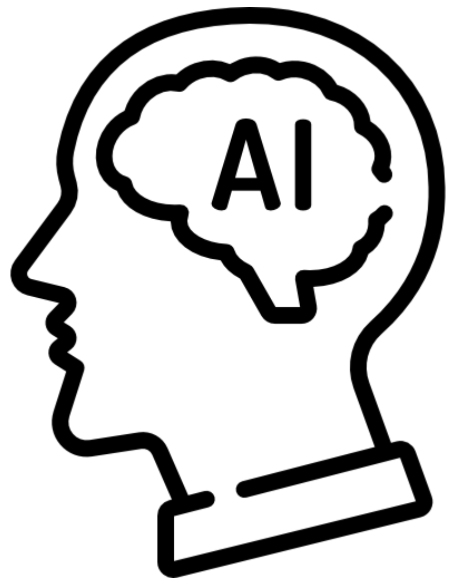
- 내일은 비가 온다. (??%)
- 내일은 수요일이다. (100%) → (??%)
    - 만약 “오늘은 화요일이다”라는 정보가 없다면?

불확실성의 발생 요소

- 대상이 불확실성을 내포하고 있을 때 >> Irreducible
- 부족한 정보에 기반 할 때 >> Reducible (if we get more data)

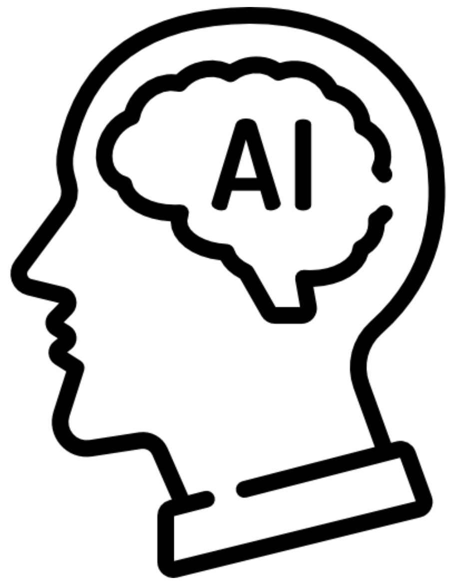
불확실성 발생상황의 예시

키 (cm)	몸무게 (kg)
170	80
170	90
170	60
170	70
180	82
180	80
180	85
180	81



모델 A

- 170 cm → 75kg
- 180 cm → 83kg

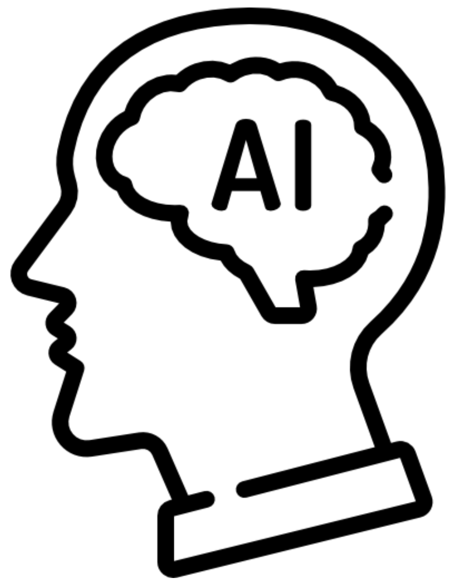


모델 B

- 170 cm → 80kg
- 180 cm → 82kg

특정 모델의 관점에서

키 (cm)	몸무게 (kg)
170	80
170	90
170	60
170	70
180	82
180	80
180	85
180	81



모델 A

- 170 cm → 75kg
- 180 cm → 83kg

- 실제 데이터와 예측값 사이의 갭.
- 데이터 자체가 불확실함.

학습 결과의 관점에서

키 (cm)	몸무게 (kg)
170	80
170	90
170	60
170	70
180	82
180	80
180	85
180	81

모델 A

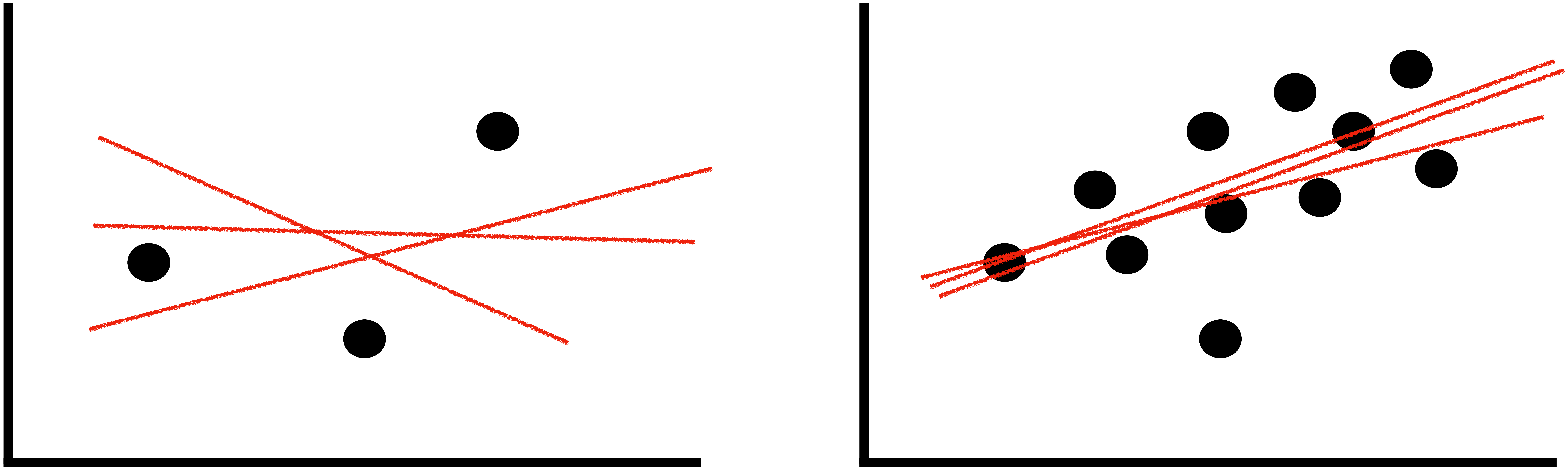
- 170 cm → 75kg
- 180 cm → 83kg

모델 B

- 170 cm → 80kg
- 180 cm → 82kg

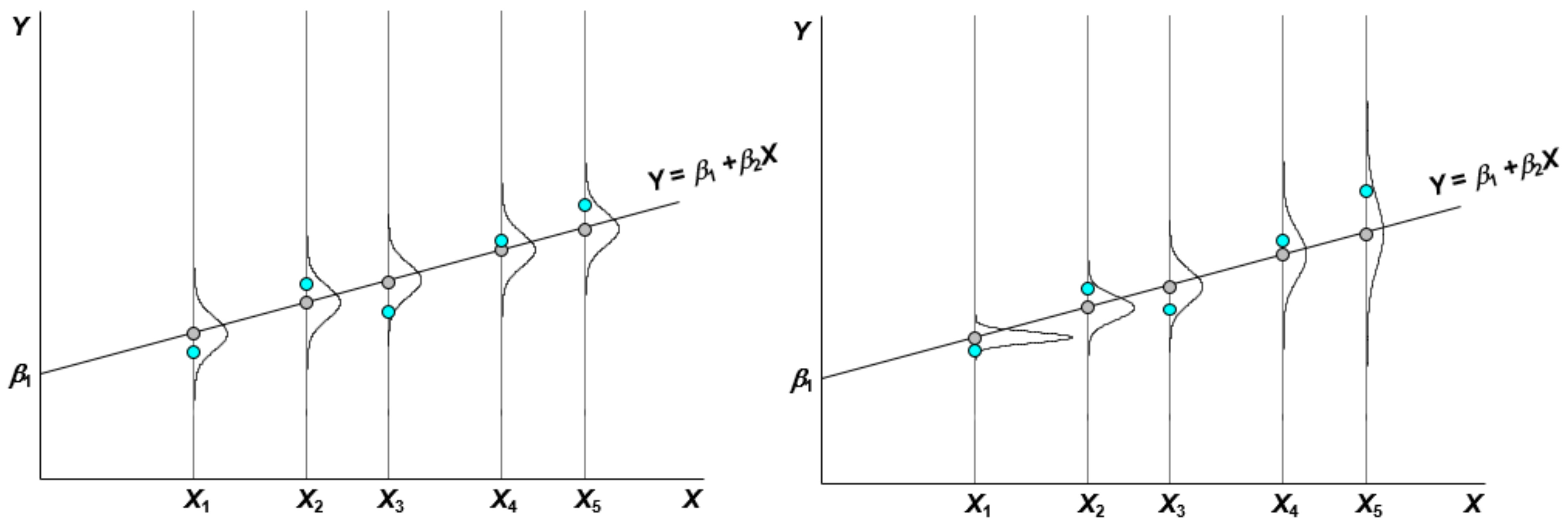
- 170 cm 입력시 : 75kg vs 80kg
- 같은 데이터 → 서로 다른 결과
- 왜 다를까? → w가 달라서 → 모델이 달라서
- 새로운 데이터가 추가되면? ← Lack of Information

Reducible?



딥러닝과 불확실성 - Data

- Data (Aleatoric) uncertainty : 입력 데이터가 같더라도 정답이 불확실한 경우
  - e.g., 기온 예측 등 대부분의 regression 문제
  - Irreducible
  - Further classified to Homoscedastic & Heteroscedastic aleatoric uncertainties.



딥러닝과 불확실성 - Model

- Model (Epistemic) uncertainty : 부족한 정보에 의해서 모델링이 불확실해지는 것
  - 결과적으로 같은 데이터로 학습하더라도 모델별로 차이가 발생
  - e.g., 같은 클래스 but 다른 성적, 신경망 가중치 초기화에 따른 변화
  - Reducible (when more data is available).

UQ for regression

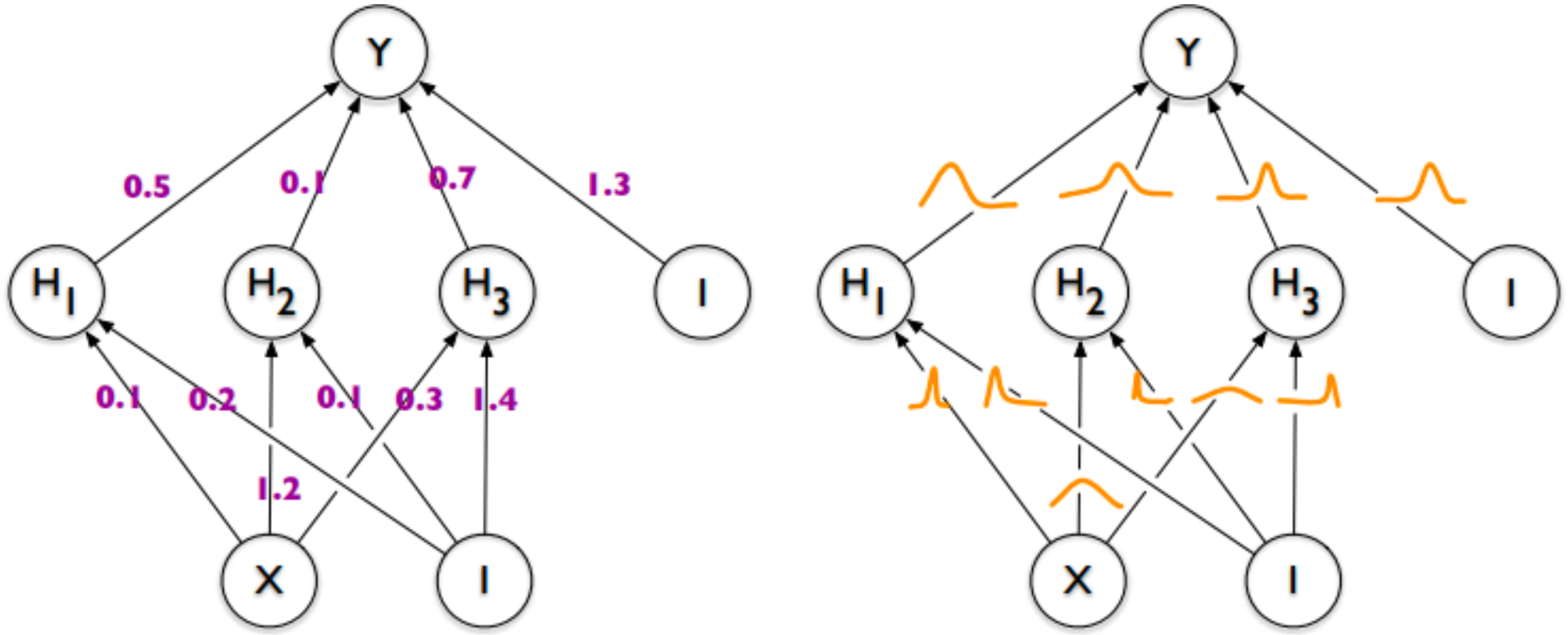
- 결과값의 확률 분포에 대한 이해
- Parametric distribution의 statistics
- “Deep learning regression model의 uncertainty quantification”

Quantifying epistemic uncertainty

- 기존의 접근 방법 : 고정된 웨이트를 갖는 인공신경망 모델
- 모델의 불확실성을 모델링하기위해서는?
  - 많은 (서로 다른) 모델이 필요 >> 같은 입력에 대해 모델의 차이에 의해 발생한 결과 분포
  - 1안 : 방법이나 모델 구조를 바꾸기 (layers, neurons, structures, etc.)
    → 효율적? 모델 간의 기본적인 차이가 큼
  - 2안 : 같은 모델 구조에서 서로다른 웨이트를 갖는 모델을 생성
    a. w의 분포를 학습해서 샘플링 : Bayesian NN
    b. Dropout으로 서브네트워크 활용 : MC Dropout
    c. 초기화를 다르게 해서 다른 네트워크를 활용 : Deep ensemble

Bayesian Neural Network

- Bayesian inference for Neural Network
  - Calculate the posterior distribution of the weights with observations *D*, $P(w \mid D)$
- Bayesian Neural Network
  - Fixed weight → Weight Distribution

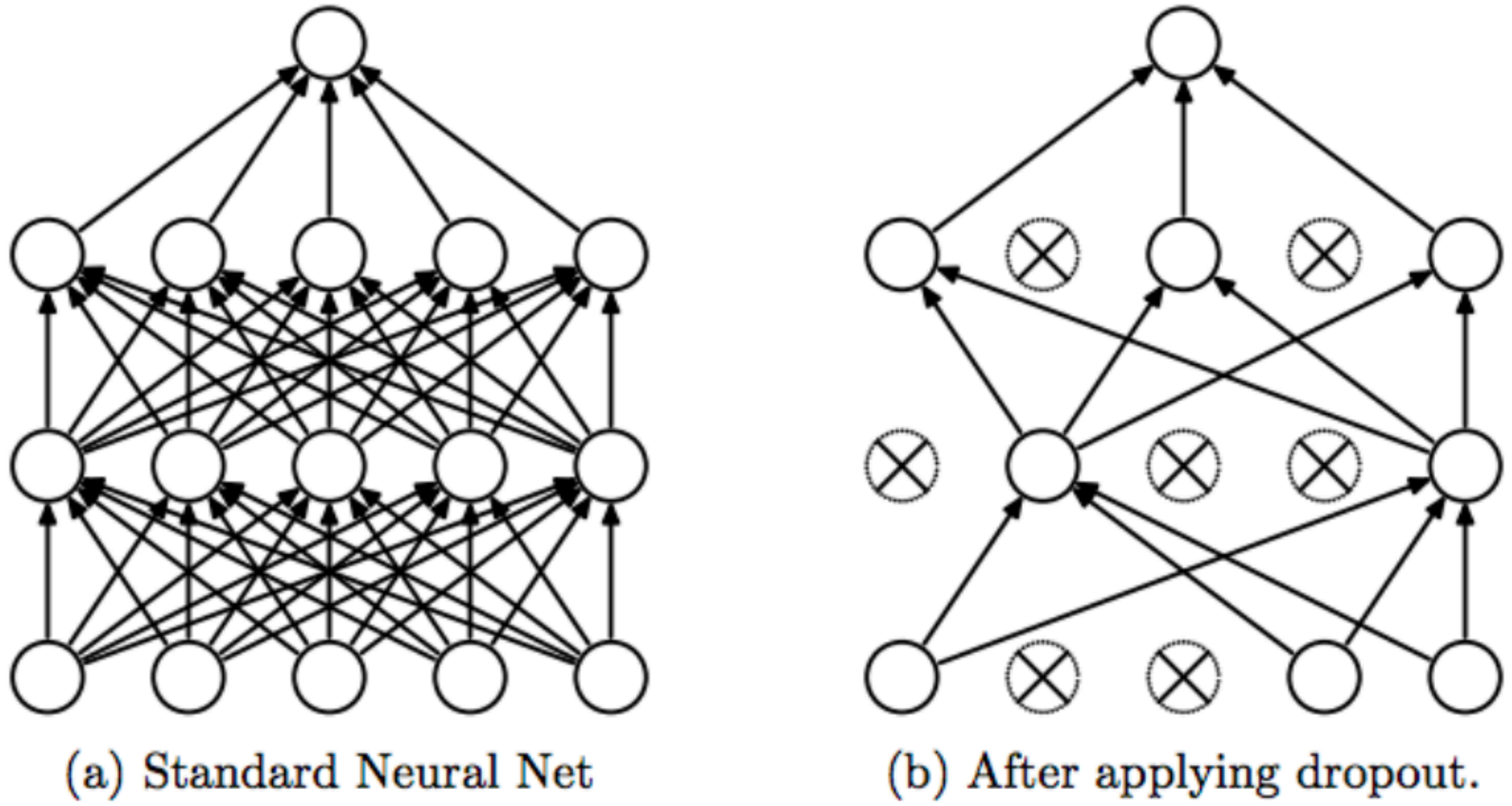


Bayesian NN

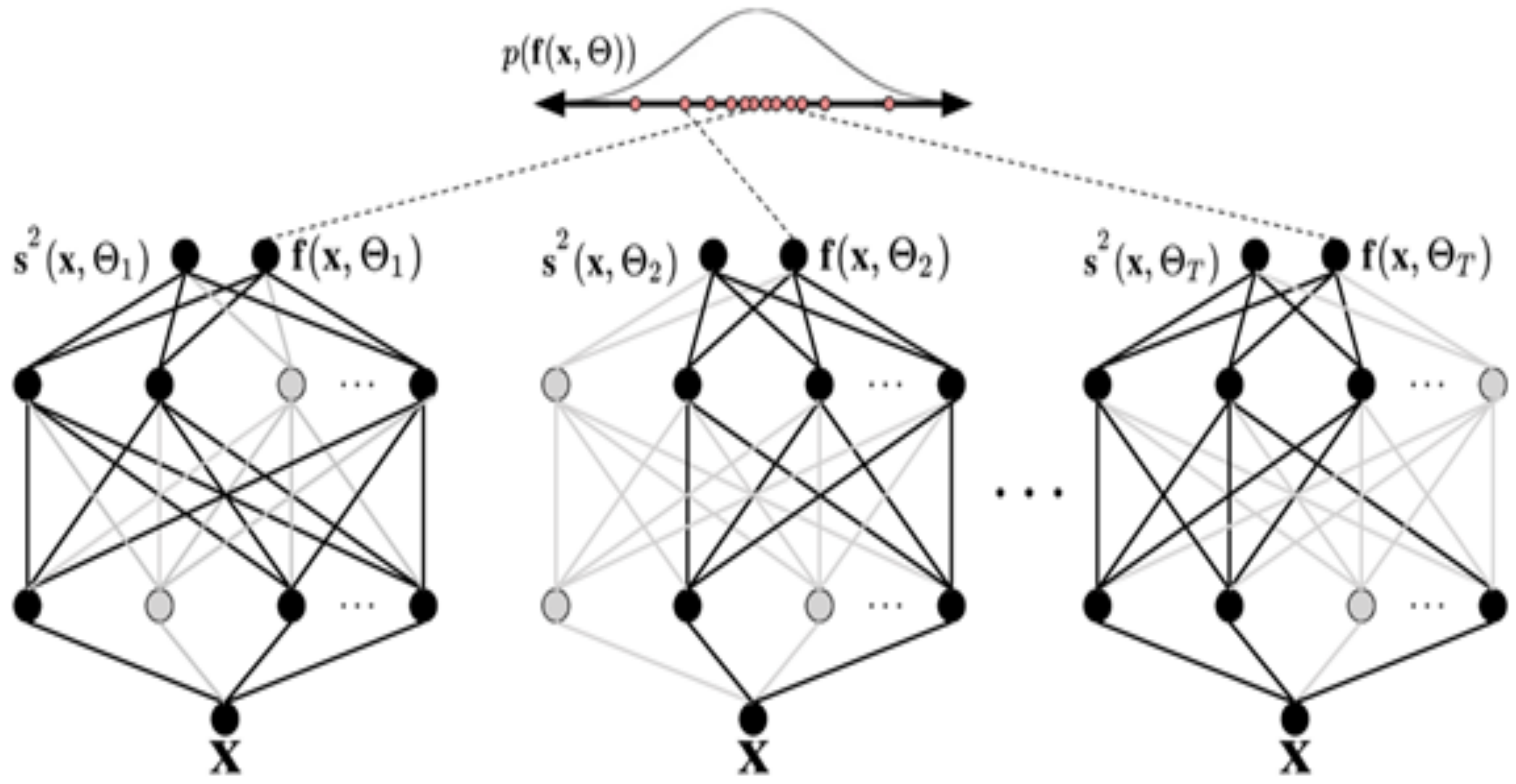
- Then predictive distribution of unknown data $P(\hat{y} \mid \hat{x})$ is …
  - $\mathbb{E}_{P(w|D)}[P(\hat{y} \mid \hat{x}, w)]$ (i.e., ensemble of differently weighted NNs)

MC dropout for Epistemic Uncertainty

- Dropout : Randomly kill neurons (during training)
- MC dropout : Randomly kill neurons (during training and inference)



Dropout



MC-dropout

Gal, Yarin, and Zoubin Ghahramani. "Dropout as a bayesian approximation: Representing model uncertainty in deep learning." *international conference on machine learning*. PMLR, 2016.

Deep Ensemble

- Simply used ensemble of networks.
- Create M models that are differently initialized.

Algorithm 1 Pseudocode of the training procedure for our method

▷ *Let each neural network parametrize a distribution over the outputs, i.e.* $p_\theta(y|\mathbf{x})$. *Use a proper scoring rule as the training criterion* $\ell(\theta, \mathbf{x}, y)$. *Recommended default values are* $M = 5$ *and* $\boldsymbol{\epsilon} = 1\%$ *of the input range of the corresponding dimension (e.g 2.55 if input range is [0,255]).*
Initialize $\theta_1, \theta_2, \ldots, \theta_M$ randomly
for $m = 1 : M$ **do** ▷ *train networks independently in parallel*
  Sample data point n_m randomly for each net ▷ *single* n_m *for clarity, minibatch in practice*
  Generate adversarial example using $\mathbf{x}'_{n_m} = \mathbf{x}_{n_m} + \boldsymbol{\epsilon}\,\mathrm{sign}\big(\nabla_{\mathbf{x}_{n_m}} \ell(\theta_m, \mathbf{x}_{n_m}, y_{n_m})\big)$
  Minimize $\ell(\theta_m, \mathbf{x}_{n_m}, y_{n_m}) + \ell(\theta_m, \mathbf{x}'_{n_m}, y_{n_m})$ w.r.t. θ_m ▷ *adversarial training (optional)*

Lakshminarayanan, Balaji, Alexander Pritzel, and Charles Blundell. "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles." *Advances in Neural Information Processing Systems* 30 (2017).

일반적인 딥러닝의 태스크

- Classification : 범주형 변수 예측
- Regression : 실수값 예측

일반적인 딥러닝의 태스크

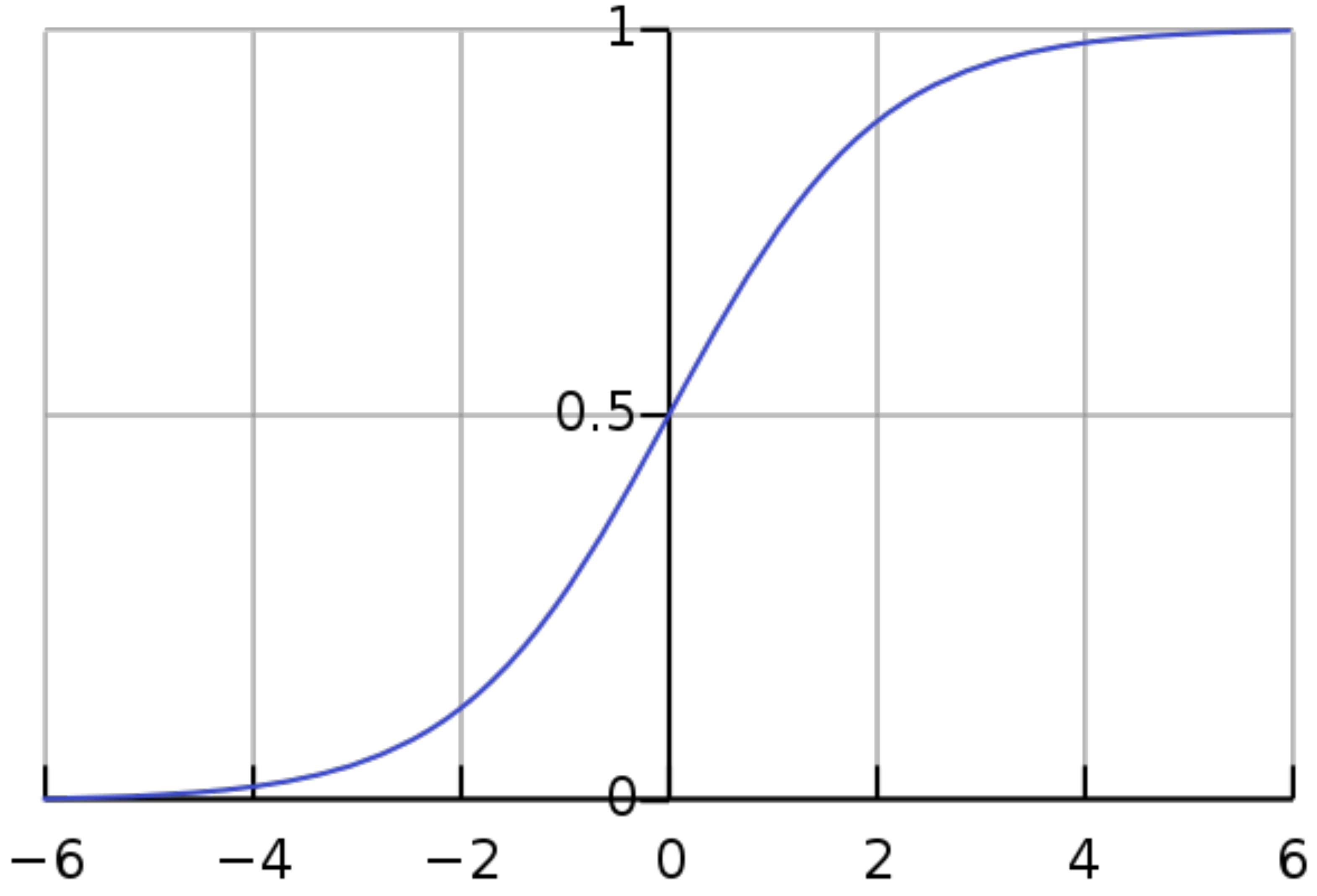
- Classification : 범주형 변수 예측 >> 이미 확률의 형태를 가지고 있음
    - [개, 고양이] >> [0.9, 0.1] >> Model uncertainty에 중점
- Regression : 실수값 예측 >> 보통 point value
    - 온도 >> 12

일반적인 딥러닝의 태스크

- Classification : 범주형 변수 예측 → Logistic Regression
- Regression : 실수값 예측

A **logistic function** or **logistic curve** is a common S-shaped curve (sigmoid curve) with equation

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}},$$



일반적인 딥러닝의 태스크

- Classification : 범주형 변수 예측 → Minimize log loss
- Regression : 실수값 예측 → How? Minimize MSE (mean squared error)

Typical regression with MSE

- y : true value
- $\hat{y}$: model output
- i : sample index
- MSE : $1/N \sum (y_i - \hat{y}_i)^2$
- 왜 MSE를 최소화하도록 학습할까? 오차가 작아야 정확하기때문에…

Using MAE?

- MAE(Mean Absolute Error) : $1/N \sum |y_i - \hat{y}_i|$
- MAE로 최소화?

Go back to likelihood

- 확률 분포가 θ라는 변수로 모델링 될 때,
- θ 값에 대해 x가 튀어나올 확률
- Likelihood : $P(X = x | \theta)$
- Maximum likelihood : θ로 모델링이 잘된다.
- In Deep learning : $P_\theta(Y = y | x)$ → y: label, x : input, θ : network weight

Recall : 일반적인 딥러닝의 태스크

- Classification : 범주형 변수 예측 → Logistic Regression → How? Log Loss
  - Minimize negative log likelihood of binary or multi-nominal distribution
- Regression : 실수값 예측 → How? Minimize MSE (mean squared error)
  - Minimize negative log likelihood of <something>?

Recall : 일반적인 딥러닝의 태스크

- Classification : 범주형 변수 예측 → Logistic Regression → How?
    - Minimize negative log likelihood of binary or multi-nominal distribution
- Regression : 실수값 예측 → How? Minimize MSE (mean squared error)
    - Minimize negative log likelihood of <Gaussian distribution>?

Recall : Gaussian distribution

- If 측정값 (y) = 실제값 $(y*)$ + 노이즈~$N(0,\sigma^2)$ → $N(y*, \sigma^2)$

PDF of Gaussian : $\frac{1}{\sigma\sqrt{2\pi}}exp\left(-\frac{1}{2}\left(\frac{y-\mu}{\sigma}\right)^2\right)$

- Negative log likelihood of Gaussian

$log(\sigma\sqrt{2}) + \frac{1}{2}\left(\frac{y-\mu}{\sigma}\right)^2$ → μ : $\hat{y}$, σ : constant → Squared error term !!

- Mutli-variate Case? : Covariance matrix 가 끼어있음 $(y-\mu)\Sigma^{-1}(y-\mu)^T$

Modeling uncertainty by Gaussian

- In typical “point” regression…

$$log(\sigma\sqrt{2}) + \frac{1}{2}\left(\frac{y-\mu}{\sigma}\right)^2 \rightarrow \mu : \hat{y}, \sigma : \text{constant}$$

- 모델이 μ와 σ를 모두 예측하도록 만들면?
- 출력이 가우시안 분포로 모델링했을 때의 분포를 알 수 있음.
- 네트워크의 출력 뉴런 수를 두 배로 늘리고, MSE 대신 GNLL loss function을 사용.

Using MAE?

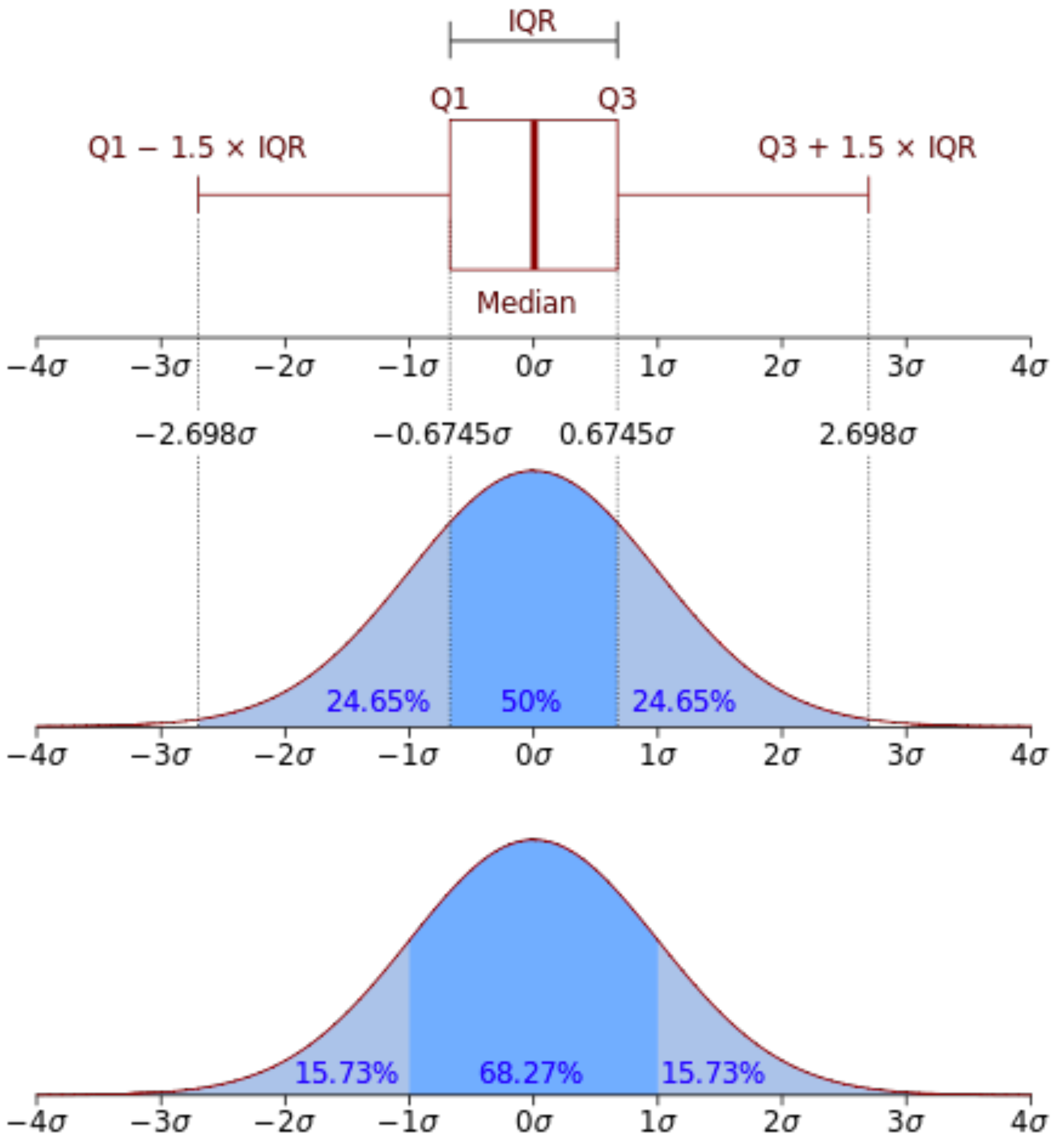
- MAE(Mean Absolute Error) : $1/N \sum |y_i - \hat{y}_i|$
- MAE로 최소화?
- Quantile regression의 특별 케이스

Quantile이란?

- **Quantiles** are **cut points** dividing the range of a probability distribution into continuous intervals with equal probabilities, or dividing the observations in a sample in the same way.

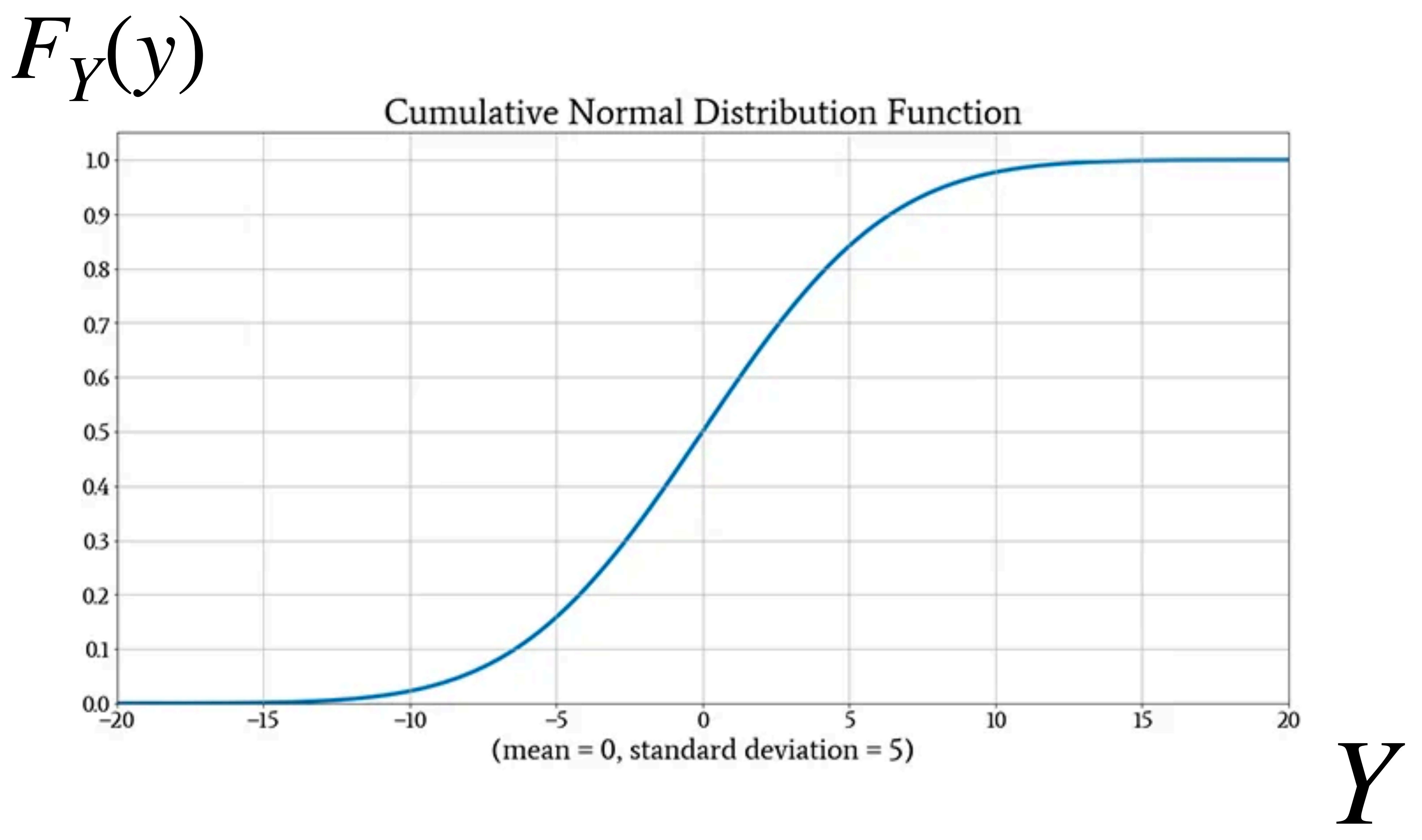
Example

- 2-quantile : 전체 분포를 동일 확률을 갖는 2 구간으로 나누는 점 = Median (중앙값)
- 4-quantile : Quartile, 박스플롯에서 활용
  - Q1, 1st quartile = 하위 25%,
  - Q2, 2nd quartile = median
  - Q3, 3rd quartile = 상위 25%

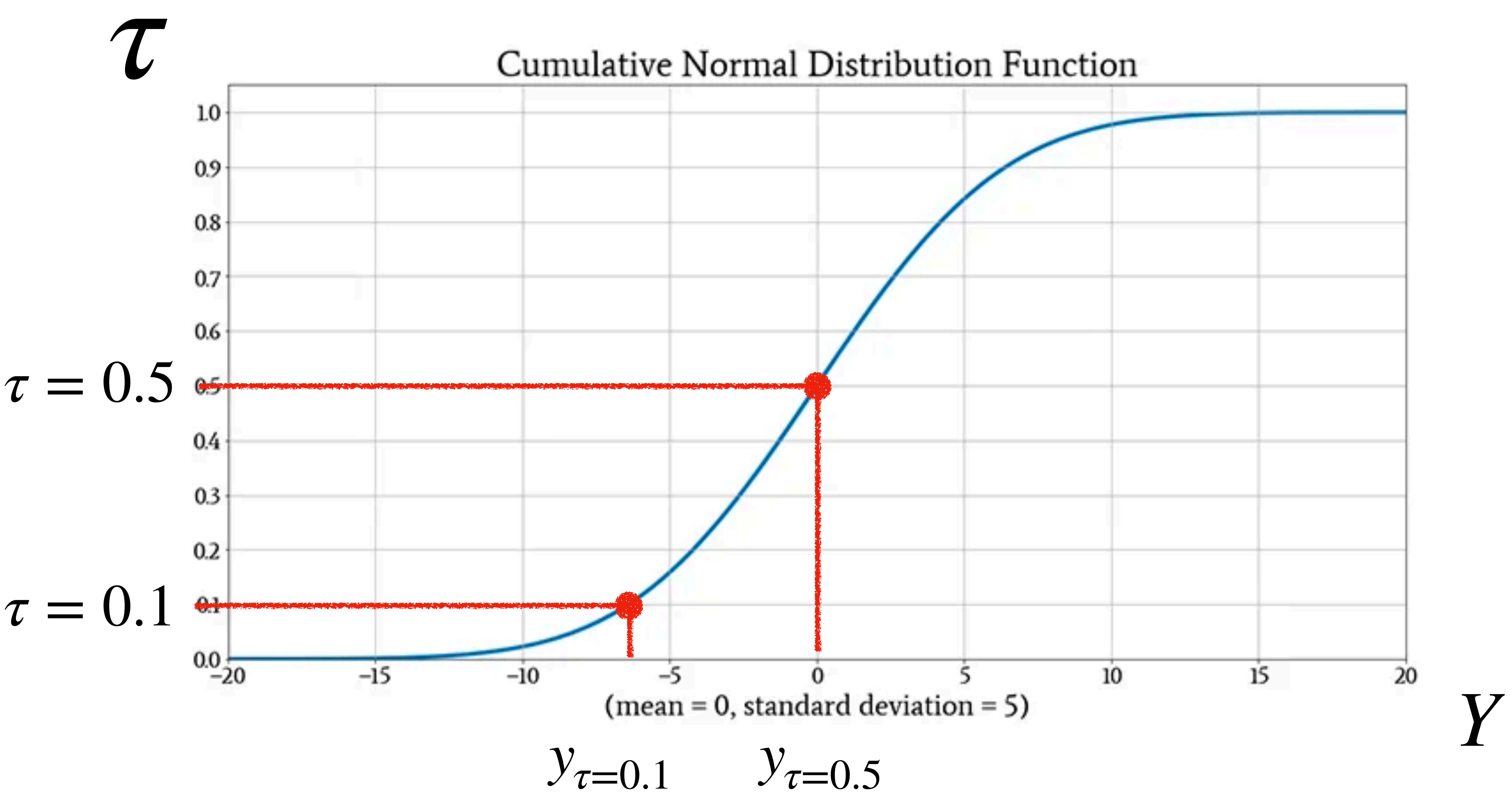


Quantile function

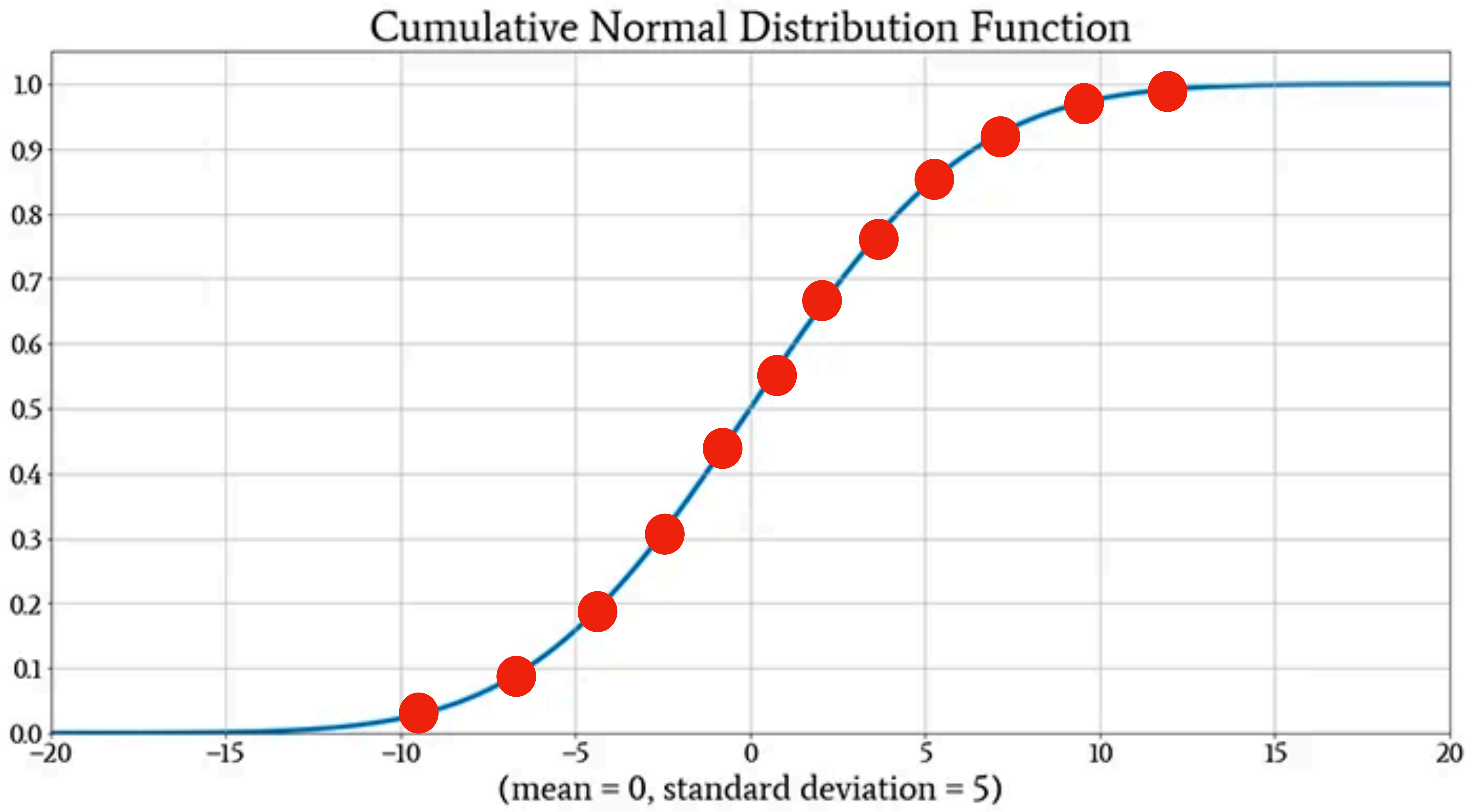
- CDF (누적분포함수)의 역함수
- Random variable Y, and It's CDF, $F_Y(y)$
- $\tau \in (0,1)$
- τ-th quantile of Y is $y_\tau = F_Y^{-1}(\tau)$
    - 즉, 누적 확률이 τ가 되는 지점의 y 값



Quantile function



Modelling uncertainty by quartile regression



GNLL 모델과의 차이점은?

- GNLL은 사전에 분포의 형태를 정해두고,
- 그 파라미터 (평균, 분산)를 출력하도록 학습.

Quantile regression via NN

- 알고자하는 τ 값을 설정해두고 신경망이 τ-th quantile 값을 출력하도록 만든다.
  - e.g., $\tau \in \{0.1, 0.5, 0.9\}$ → 신경망의 출력은 $y_{0.1}, y_{0.5}, y_{0.9}$
  - τ 별로 개별 신경망을 학습하는 방법. → Crossing quantile problem → monotonicity condition이 깨지는 경우 (τ의 대소관계가 y_τ의 대소관계로 이어짐)
  - 하나의 신경망으로 동시에 출력하도록 구성하는 방법.

학습은 어떻게?

- 지도학습에서는 label 만 있으면 학습할 수 있음.
- 기본적인 regression : $y = f(x)$, 정답 레이블 y를 알고있는 상황.
- Quantile regression : $y_\tau = f_\tau(x)$, $\{y_{0.1}, y_{0.5}, y_{0.9}\}$에 대한 정답이?
    - 정답을 안다는 것은 y에 대한 확률분포 정보를 이미 알고있다는 뜻.
    - 그러나 우리가 얻은 것은 미지의 확률 분포로 부터 얻은 몇 개의 샘플들일 뿐.
    - 샘플로부터 각 τ에 대한 예측을 수행할 수 있는 loss function이 필요.

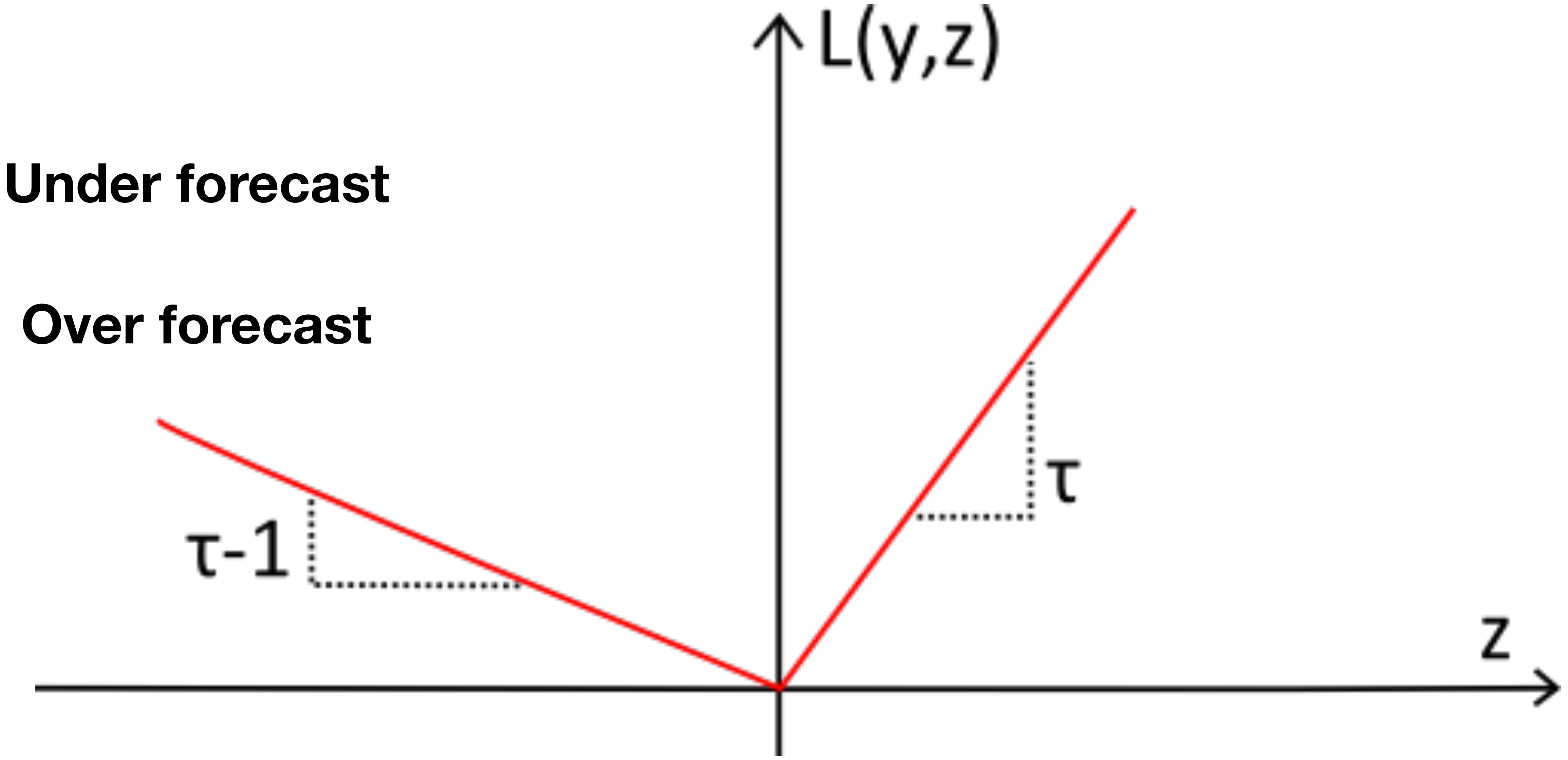
Pinball loss

- y : 정답 레이블, $\hat{y}_\tau$: τ-th quantile에 대한 신경망 출력
- Error = $y - \hat{y}_\tau$
- Pinball loss
- $L_\tau(y, \hat{y}) = \begin{cases} \tau(y - \hat{y}) & \text{if } y \geq \hat{y} \\ (1-\tau)(\hat{y} - y) & \text{if } y < \hat{y} \end{cases}$

Under forecast

Over forecast

- 오차가 양수인 구간의 기울기 = τ
- 오차가 음수인 구간의 기울기 = $\tau - 1$



동작 예시 (Lower quantile case)

- $y = 0.5, \tau = 0.3$ 인 경우 (lower quantile case)

$$L_\tau = \begin{cases} \tau(y - \hat{y}) & \text{if } y \geq \hat{y}(UF) \\ (1-\tau)(\hat{y} - y) & \text{if } y < \hat{y}(OF) \end{cases}$$

- Case 1 : $\hat{y} = 0.3$, (under-forecast)
  - 0.3 * (0.5-0.3) = 0.06 / (1-0.3)(0.3-0.5) < 0
- Case 2 : $\hat{y} = 0.7$ (over-forecast)
  - 0.7* (0.7-0.5) = 0.14 / 0.7(0.5-0.7) < 0
- 오차의 절대값이 같더라도 τ가 0.5 보다 작은 경우 over-forecast일 때 더 큰 loss값을 갖음 >> under-forecast 하도록 유도

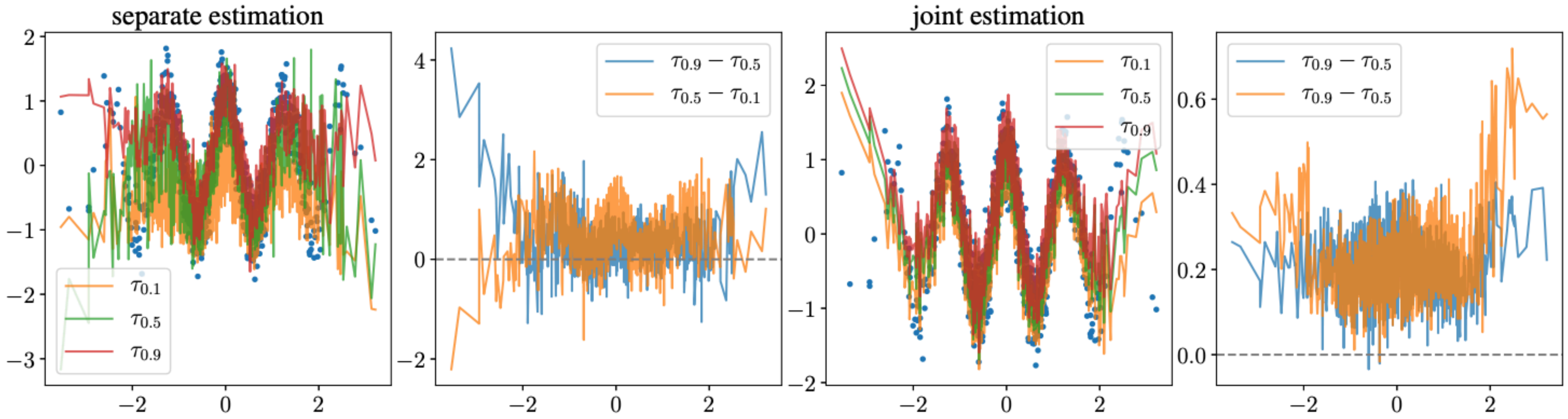
동작 예시 (Upper quantile case)

- $y = 0.5, \tau = 0.7$ 인 경우 (upper quantile case)
- Case 1 : $\hat{y} = 0.3$, (under-forecast)
  - 0.7 * (0.5-0.3) = 0.14 / (1-0.3)(0.3-0.5) < 0
- Case 2 : $\hat{y} = 0.7$, (over-forecast)
  - 0.3* (0.7-0.5) = 0.06 / (0.7)(0.5-0.7) < 0
- 오차의 절대값이 같더라도 τ가 0.5 보다 큰 경우는 under-forecast일 때 더 큰 loss 값을 갖음 >> over-forecast하도록 유도

$$L_\tau = \begin{cases} \tau(y - \hat{y}) & \text{if } y \geq \hat{y}(UF) \\ (1-\tau)(\hat{y} - y) & \text{if } y < \hat{y}(OF) \end{cases}$$

Crossing quantiles

- 모델을 개별적으로 학습시키면 τ 사이의 관계를 학습할 수 없음.
- If $\tau_1 < \tau_2$ then $y_{\tau_1} < y_{\tau_2}$.



- Pinball score 가 τ 사이의 관계를 학습하는 방법은?

Crossing quantile with loss

- Loss function 계산시 다양한 τ 값이 고려되어야 함.

$$L = \sum_{\tau \in Q} L_\tau(y, \hat{y}_\tau)$$

동작 예시 (lower quantile case)

- 두 개의 τ를 가정 : $\tau_1 = 0.1, \tau_2 = 0.2$
- $y_{\tau_1} < y_{\tau_2}$가 성립되어야 함.
- Let $\hat{y}_{\tau_2} = \hat{y}_{\tau_1} + \alpha$, and $\hat{y}_{\tau_1}, \hat{y}_{\tau_2} < y$
- $$\begin{aligned} L_{\tau_1} + L_{\tau_2} &= \tau_1(y - \hat{y}_{\tau_1}) + \tau_2(y - \hat{y}_{\tau_2}) \\ &= (\tau_1 + \tau_2)y - (\tau_1 + \tau_2)\hat{y}_{\tau_1} - \tau_2\alpha \\ &= (\tau_1 + \tau_2)(y - \hat{y}_{\tau_1}) - \tau_2\alpha \end{aligned}$$

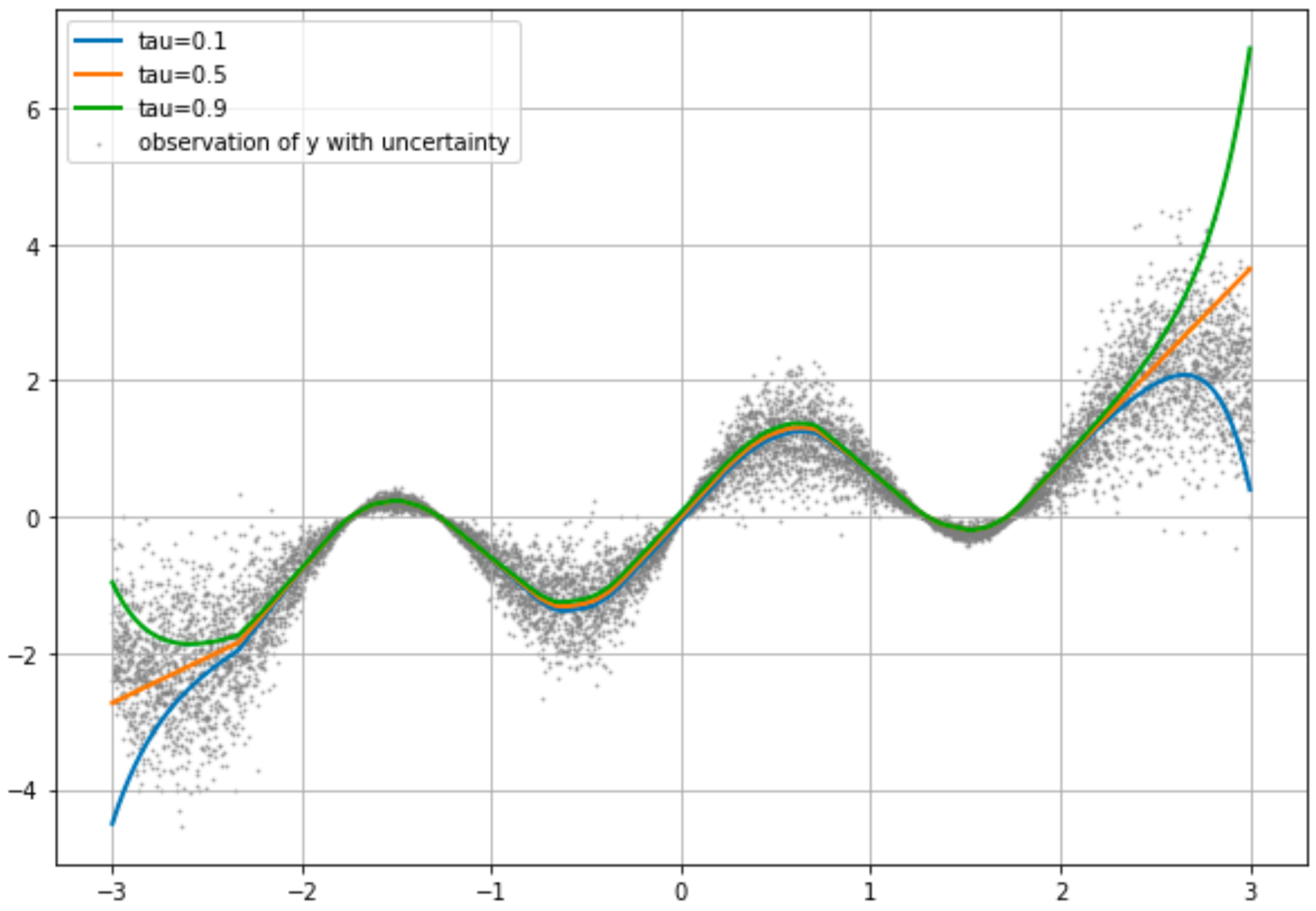
$$L_\tau = \begin{cases} \tau(y - \hat{y}) & \text{if } y \geq \hat{y}(UF) \\ (1 - \tau)(\hat{y} - y) & \text{if } y < \hat{y}(OF) \end{cases}$$

$$L = \sum_{\tau \in Q} L_\tau(y, \hat{y}_\tau)$$

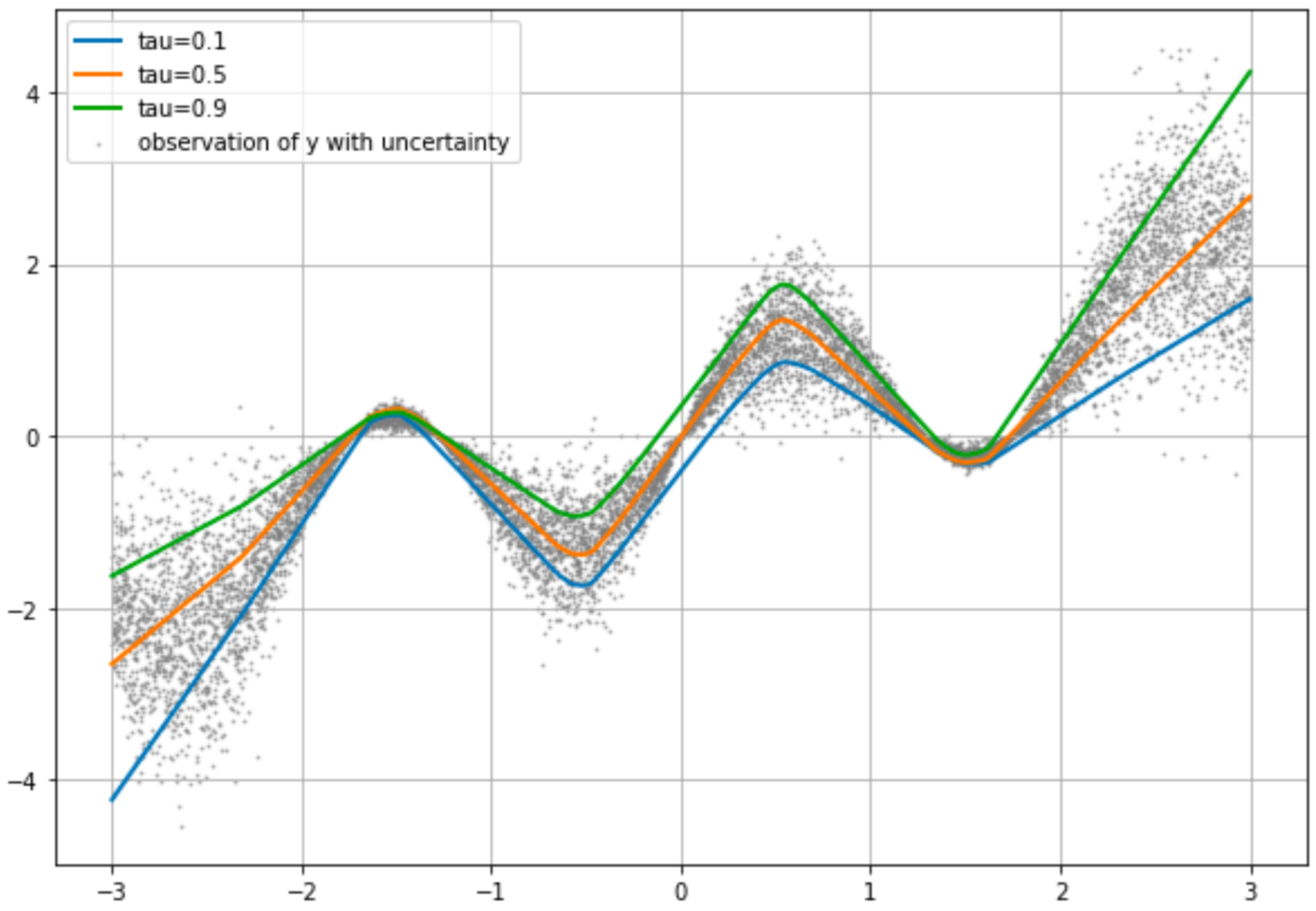
동작 예시 (lower quantile case)

- $(\tau_1 + \tau_2)(y - \hat{y}_{\tau_1}) - \tau_2\alpha$ >> (+)(+) - (?)
- loss는 두가지 경우에 감소함
  - $\hat{y}_{\tau_1}$이 y에 가까워질 때 (예측을 잘 할 때)
  - $\alpha > 0$ 일 때 ($y_{\tau_1} < y_{\tau_2}$ 조건을 만족할 때)
- Upper quantile case? >> $y < \hat{y}_{\tau_1} < \hat{y}_{\tau_2}$

GNLL과 Pinball loss 사용시 학습 결과예시



GNLL with standard deviation



Pinball loss with 0.1, 0.5, 0.9 quantile

확률적 예측 모델의 모델 평가 방법은?

- 점예측 (point prediction) 에서는 해당 포인트의 값과 정답값 사이의 오차로 평가
  - MSE, MAPE, RMSE, etc
- 확률적예측 (probabalilistic prediction)에서는 정답값을 바탕으로 분포의 정확도를 평가할 수 있는 메트릭이 필요
  - Pinball loss, Winkler score, Prediction Interval Normalized Average Width(PINAW)

Pinball loss for scoring

- Multiple τ에 대한 pinball loss의 합을 사용.
- $Q = \{0.1, 0.3, 0.5, 0.7, 0.9\}$ (신경망이 출력하는 τ 값들)
- i : sample index for test set

$$\sum_{i} \sum_{\tau \in Q} L_\tau(y_i, \hat{y}_{i,\tau})$$

Winkler score

- Winkler score는 $(1-\alpha)\%$ prediction interval의 길이를 평가
    - e.g., For 90% PI, $\alpha = 0.1$, For 60% PI, $\alpha = 0.4$
- $[\tau_{\alpha/2}, \tau_{1-\alpha/2}]$
    - e.g., $\tau_{.05}$ & $\tau_{.95}$ for $\alpha = 0.1$, $\tau_{.2}$ & $\tau_{.8}$ for $\alpha = 0.4$

Winkler score

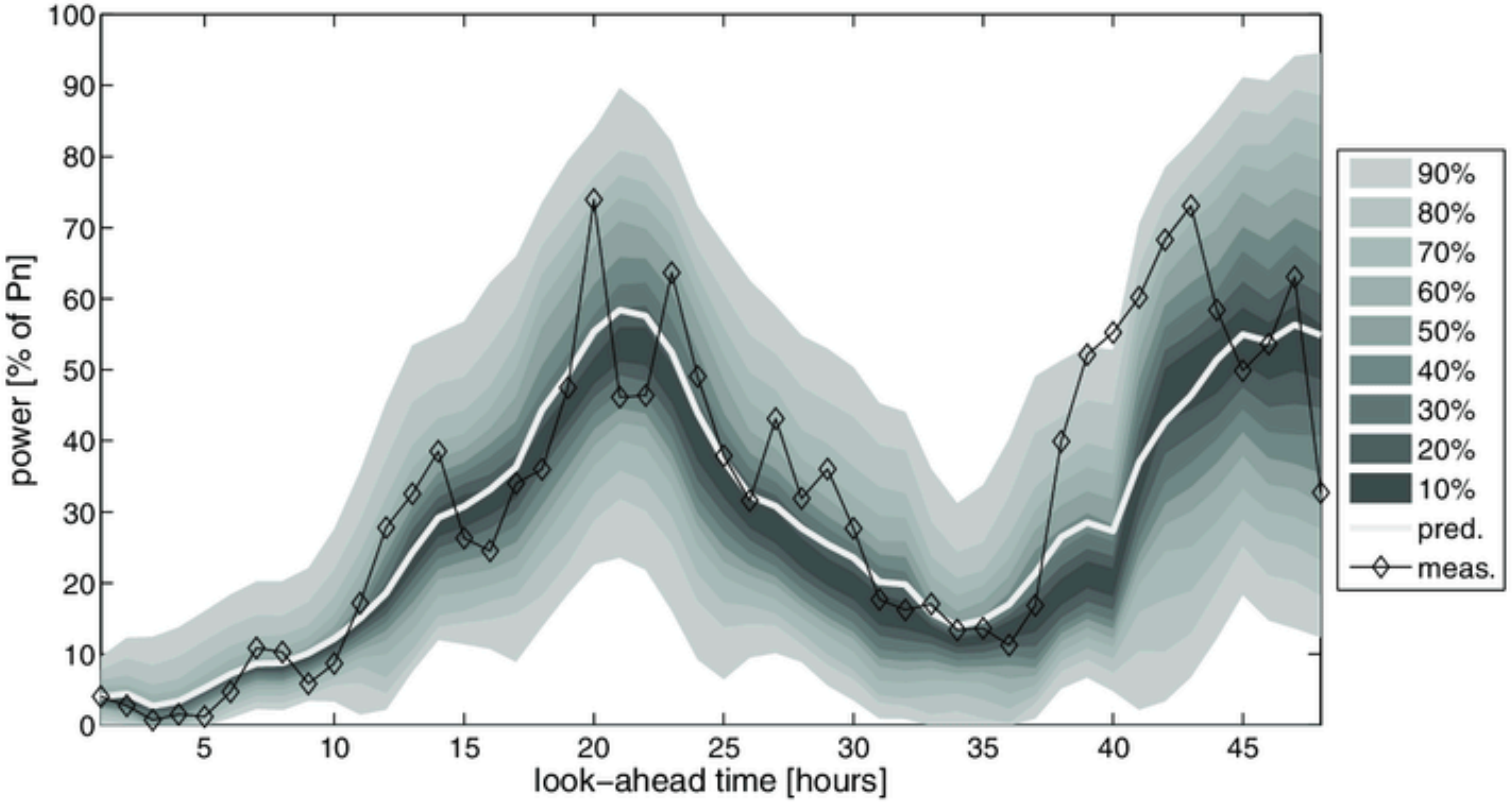
- $[\tau_{\alpha/2}, \tau_{1-\alpha/2}] \rightarrow \hat{x}_l, \hat{x}_u$
- WKL $(\hat{x}_u, \hat{x}_l, x)$ =
  - $\hat{x}_u - \hat{x}_l + 2(\hat{x}_l - x)/\alpha$ if $x < \hat{x}_l$
  - $\hat{x}_u - \hat{x}_l$ if $\hat{x}_l \leq x \leq \hat{x}_u$
  - $\hat{x}_u - \hat{x}_l + 2(x - \hat{x}_u)/\alpha$ if $\hat{x}_u < x$

불확실성 활용 예. 이상탐지

- 이상 탐지의 핵심
- → 정상과 비정상 데이터로부터 상이한 특성을 갖는 feature들 모으기
  → 정상 데이터의 분포를 모델링하기
- 오토인코더
  - 정상 데이터로 모델 학습
  - 정상 데이터는 작은 복원오차 (학습되었기때문에) → 비정상 데이터는 큰 복원오차 (학습이 안되어있기 때문에)

불확실성 활용 예. Forecasting

- 시계열 예측에서 활용
- Point vs Probabilistic



불확실성 활용 예. 이상탐지

- Prediction interval을 활용해보자
- 상/하위 분위수의 차이를 고려
- 정상 데이터의 PI와
  비정상 데이터 PI의 분포차를 이용

Quantile Autoencoder With Abnormality Accumulation for Anomaly Detection of Multivariate Sensor Data

SEUNGHYOUNG RYU[1], JIYEON YIM[1,2], JUNGHOON SEO[3], YONGGYUN YU[1], AND HOGEON SEO[1]
[1]Artificial Intelligence Application & Strategy Team, Korea Atomic Energy Research Institute, Daejeon 34507, South Korea
[2]Department of Nuclear Engineering, Ulsan National Institute of Science and Technology, Ulsan 44919, South Korea
[3]SI Analytics Company, Daejeon 34047, South Korea
Corresponding author: Hogeon Seo (hogeony@kaeri.re.kr)

This work was supported in part by the Korea Atomic Energy Research Institute (KAERI) Research and Development Program under Grant KAERI-524450-22, and in part by the National Research Foundation of Korea (NRF) funded by the Korean Government [Ministry of Science and ICT (MSIT)] under Grant NRF-2021R1F1A1051290.

불확실성 활용 예. Adaptive Samplling

Adaptive Sampling of Dynamic Scenarios close to the Limit Surface using Deep Neural Network and Monte Carlo Dropout

Junyong Bae, Jong Woo Park, and Seung Jun Lee*
Ulsan National Institute of Science and Technology, 50 UNIST-gil, Ulju-gun, Ulsan, 44919, Republic of Korea
*junyong8090@unist.ac.kr, jonwoo822@unist.ac.kr, sjlee420@unist.ac.kr**

- Surrogate 모델 개발시

- 불확실한 부분을 더 많이 학습

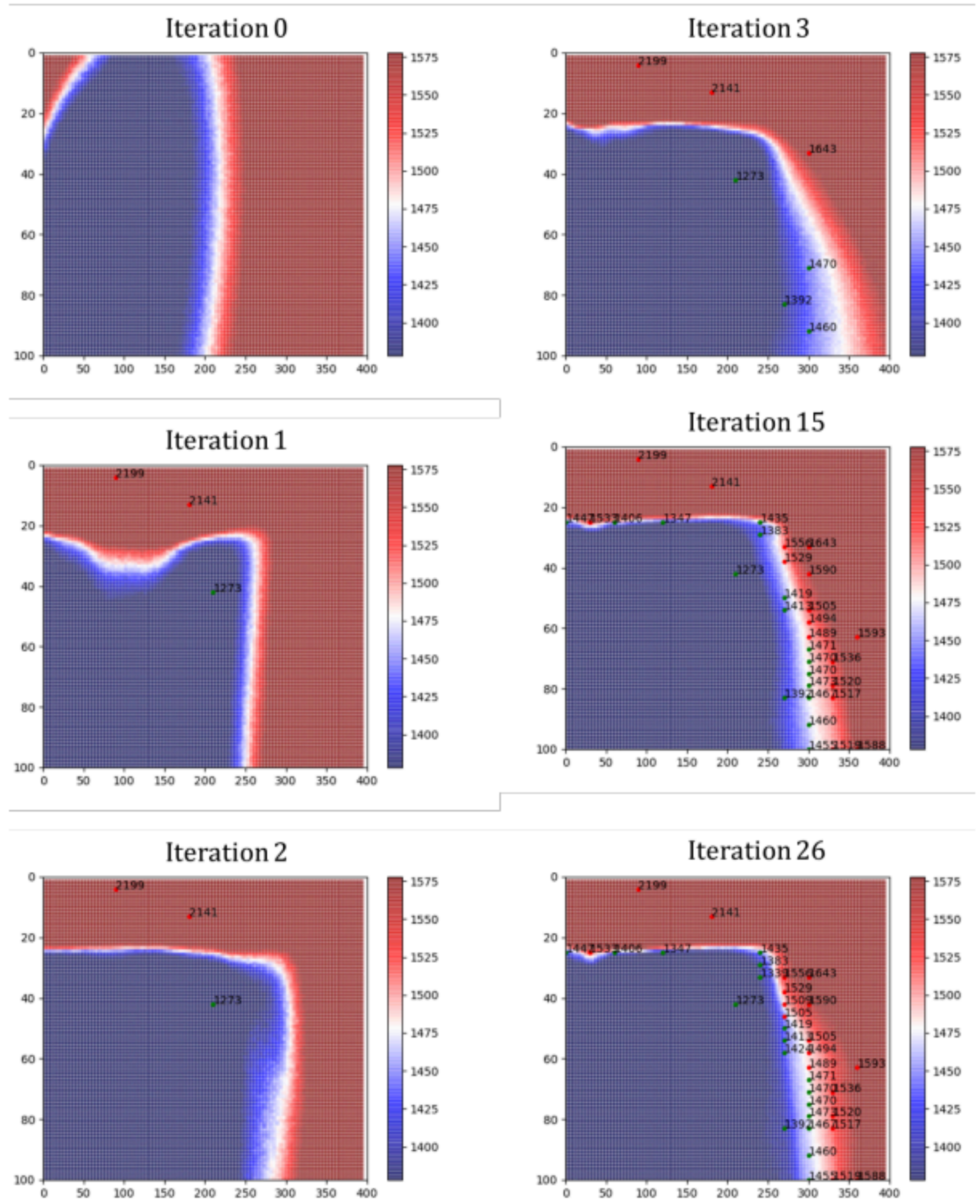


Fig. 3. The change of metamodel predictions when SITs performances are (SIT1, SIT2, SIT3 = 50%, 50%, 0)

코드로 보는 Quantile regression

- Pytorch-lightning으로 간단한 모델 만들기
- GNLL 활용한 모델 만들기
- Pinball loss function 만들기
- Quantile regression 모델 만들기