임상의를 위한 AI 교육 – 기초과정 2주차

딥 러닝의 개념과 실습

서울대학교병원 융합의학과 김영곤 교수

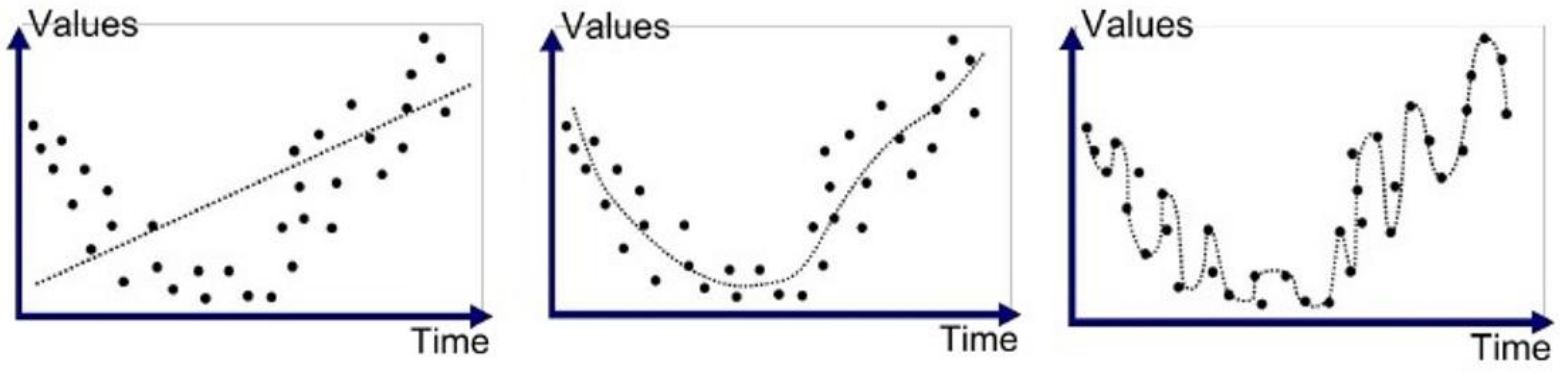
SNUH

※본 수업자료는 **“서울대학교병원 데이터사이언스연구부 AI지원실”** 학습서기반으로 제작되었습니다.

2. 학습 관련 기술 (Training Techniques)

4-6-1 **Overfitting / Underfitting**

- **과적합과 과소적합**



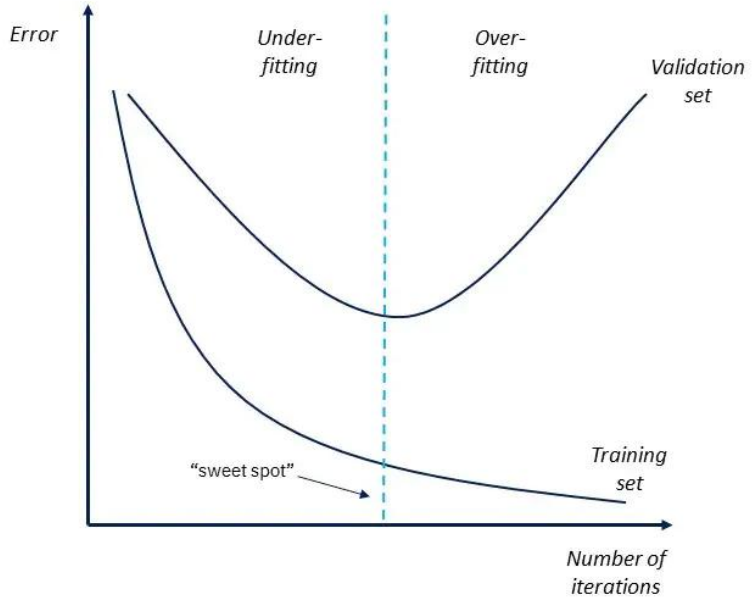
Underfitted	Good Fit/Robust	Overfitted
훈련 데이터셋을 잘 학습하지 못함 일반화도 되지 않음	훈련 데이터셋을 잘 학습함 일반화의 정도가 적합함	훈련 데이터셋을 **너무** 잘 학습함 일반화가 되지 않음

2. 학습 관련 기술 (Training Techniques)

4-6-2 **Overfitting / Underfitting**

- **과소적합 (Underfitting)의 원인과 해결방법**

  1) 모델이 충분히 학습되지 못함



∴ 학습을 더 수행하여 해결 → 에포크(Epoch)/반복(Iteration)을 증가

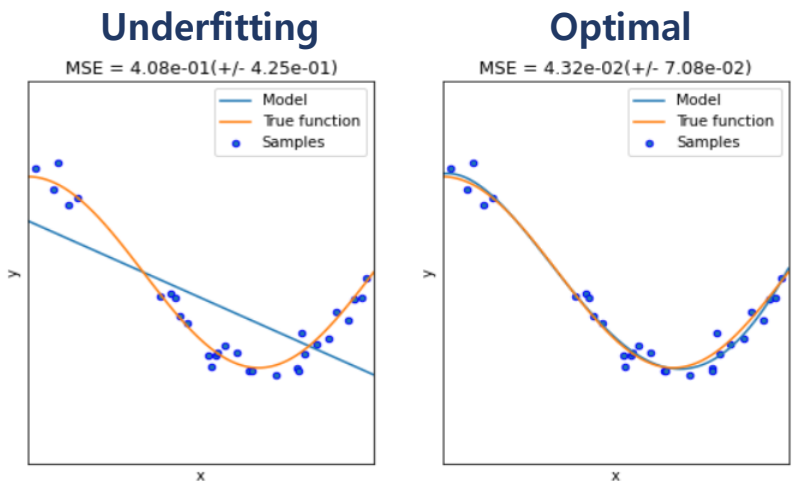
2. 학습 관련 기술 (Training Techniques)

4-6-3 Overfitting / Underfitting

- **과소적합 (Underfitting)의 원인과 해결방법**

  2) 모델의 복잡도가 너무 낮음 (= 모델이 너무 단순함)

  Ex)



2~3차식으로 해결되어야 할 문제를 1차식으로 모델링

∴ **모델의 복잡도를 높여서 해결** → 가중치(Weight)의 개수를 증가, 층(Layer)의 개수를 증가하는 등의 방법

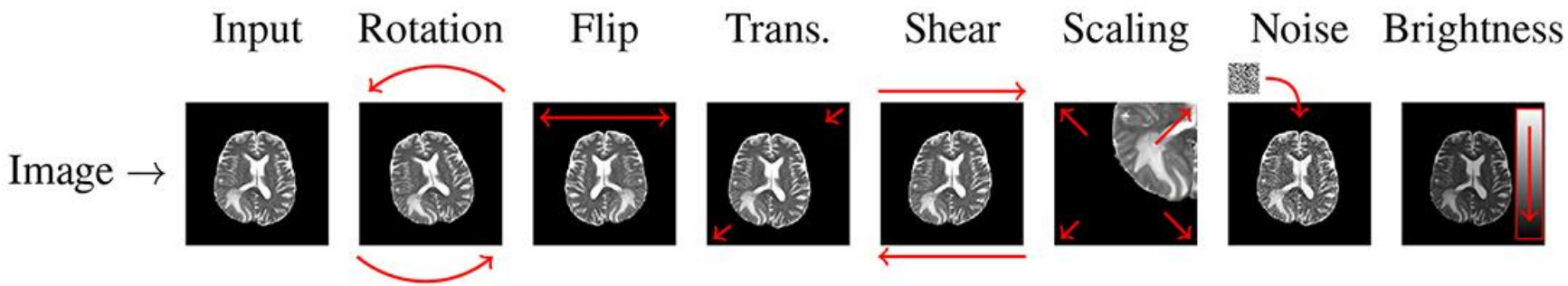
2. 학습 관련 기술 (Training Techniques)

4-6-4 **Overfitting / Underfitting**

- **과적합(Overfitting)합의 원인과 해결방법**

  1) 학습 데이터의 부족 → 데이터 증강(Data Augmentation)

  - 학습 데이터를 더 수집하는 것이 가장 이상적이지만, 현실적으로 어려운 경우가 대부분임
  - 영상 데이터의 경우, 기본적인 가공을 통해 데이터를 증가시킬 수 있음



  - 영상 데이터가 아닌 숫자 데이터(Numeric Data)의 경우에도,

    SMOTE와 같은 비지도 학습 기반의 방법이나 생성 모델인 GAN을 이용해볼 수 있음

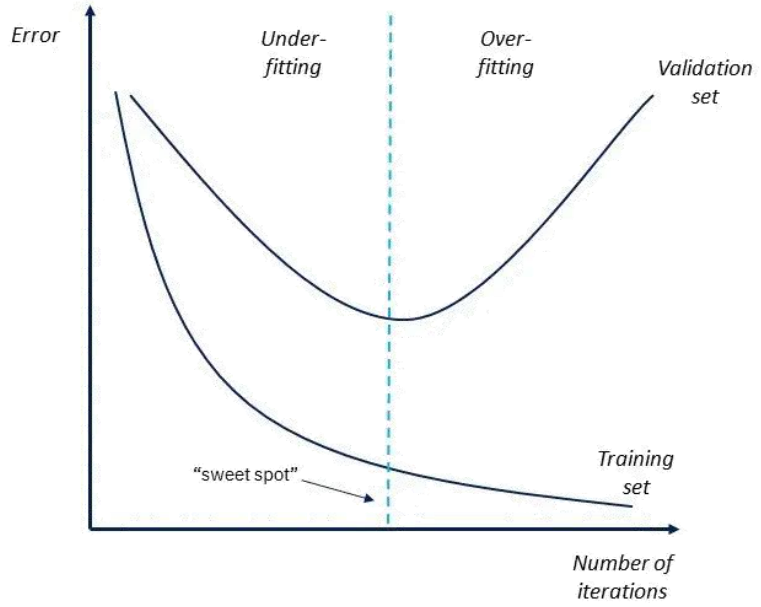
2. 학습 관련 기술 (Training Techniques)

4-6-5 **Overfitting / Underfitting**

- **과적합(Overfitting)합의 원인과 해결방법**

  2) 과도한 학습 진행 → 학습의 조기 종료(Early Stop)

  - 일반적으로, 검증 데이터(Validation Data)의 오차가 더 이상 감소하지 않는 지점을 최적의 학습 정도로 판단



2. 학습 관련 기술 (Training Techniques)

4-6-6 **Overfitting / Underfitting**

- **과적합(Overfitting)합의 원인과 해결방법**

  3) 가중치의 값들이 너무 커지는 현상 발생 → 가중치 정규화(Weight Regularization)

  - 오차 함수가 작아지는 방법으로 학습을 지속하면, 학습되는 가중치들은 값이 계속 커지게 됨
  - 오차 함수에 가중치(w)의 증가에 대해 패널티를 부여하는 항을 추가함으로써,
    단순히 오차 함수가 작아지는 방향으로만 학습되지 않도록 함

  - 기존의 오차 함수 : $C_0 = \frac{1}{n}\sum_i \left(\hat{Y}_i - Y_i\right)^2$
  - 패널티가 추가된 새로운 오차 함수 : $C = C_0 +$ (패널티항)

SNUH

2. 학습 관련 기술 (Training Techniques)

4-6-7 **Overfitting / Underfitting**

- **과적합(Overfitting)합의 원인과 해결방법**

3) 가중치 정규화(Weight Regularization)

(1) L1 Regularization (=Lasso)

$$C = C_0 + \frac{\lambda}{n}\sum |w|$$

패널티항으로 모든 가중치의 절대값의 합을 추가

→ 오차인 C_0의 값을 작게 함과 동시에, 패널티항의 값을 결정하는 w 또한 작게 하려는 방향으로 학습

2. 학습 관련 기술 (Training Techniques)

4-6-8 **Overfitting / Underfitting**

- **과적합(Overfitting)합의 원인과 해결방법**

3) 가중치 정규화(Weight Regularization)

(2) L2 Regularization (=Ridge)

$$C = C_0 + \frac{\lambda}{2n}\sum w^2$$

패널티항으로 모든 가중치의 제곱의 합을 추가

→ L1 정규화는, 영향이 적은 가중치들을 0으로 만들어버리는 반면,

L2 정규화는, 영향이 적은 가중치라고 하더라도 0에 가까운 값으로 감소시킴

혹은, L1 정규화와 L2 정규화의 패널티항을 모두 사용하는 방법도 가능(Elastic Net)

2. 학습 관련 기술 (Training Techniques)

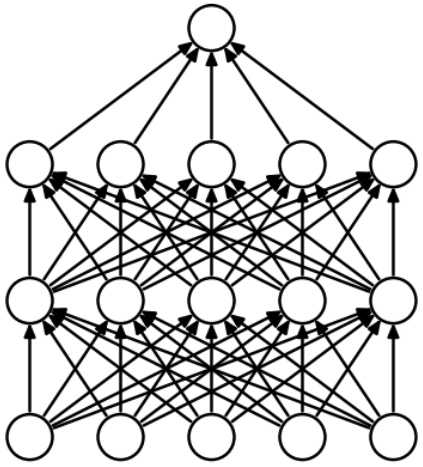
4-6-9 Overfitting / Underfitting

- **과적합(Overfitting)합의 원인과 해결방법**

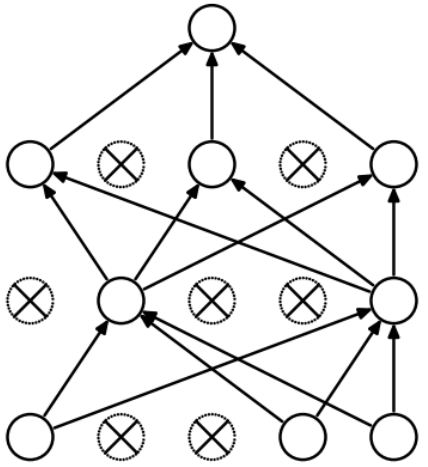
4) 모델의 복잡도가 너무 높음 → 더 단순한 모델을 설계하거나, 드롭아웃(Dropout)을 적용

- 드롭아웃 : 학습 시 모든 가중치에 대해 학습하는 대신, 특정 가중치들은 학습을 생략하는 방법

매번 임의로 선택된 가중치들만 학습함으로써 과적합을 방지할 수 있음



(a) Standard Neural Net



(b) After applying dropout.

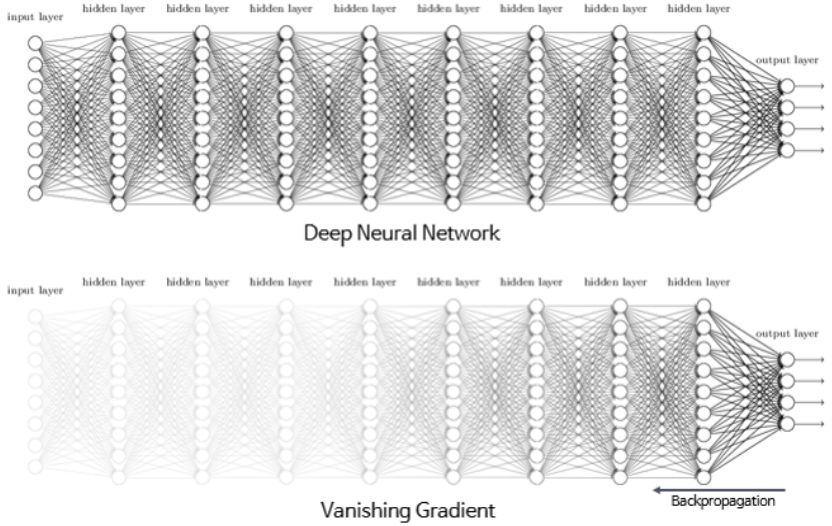
2. 학습 관련 기술 (Training Techniques)

4-7-1	Gradient Vanishing

- **기울기 소실(Gradient Vanishing)**
  - 인공 신경망을 학습시키는 과정에서, 역전파(Backpropagation)를 통해 오차를 미분해나가며 입력층까지 전달하는데,

    이 과정에서 **기울기(미분값)가 점차적으로 작아지는 현상**이 발생할 수 있음

    → 즉, 층이 깊은 인공 신경망에서 입력층에 가까워 질수록 학습이 잘 이루어지지 않는 현상



SNUH

2. 학습 관련 기술 (Training Techniques)

4-7-2 **Gradient Vanishing**

- **기울기 폭주(Gradient Exploding)**
  - 기울기 소실과 반대로, 기울기가 1보다 큰 값들이 계속 곱해지면서 가중치들이 비정상적으로 큰 값들로 학습되는 현상

    → 즉, 층이 깊은 인공 신경망에서 입력층에 가까워 질수록 비정상적으로 큰 값들로 가중치가 학습되는 현상
  - 주로, 순환신경망(RNN, Recurrent Neural Network)에서 발생함

∴ 기울기가 너무 작아지거나 너무 커지지 않도록 조절하는 방법이 필요함

→ 활성화 함수의 변경, 가중치 초기화, 정규화 등

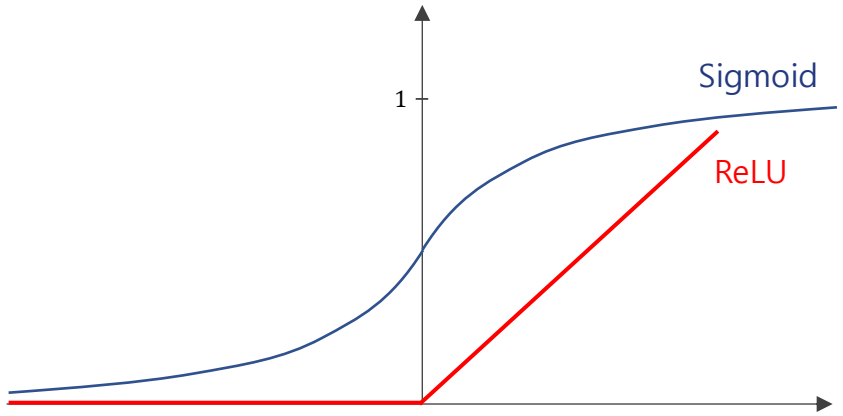
2. 학습 관련 기술 (Training Techniques)

4-7-3 **Gradient Vanishing**

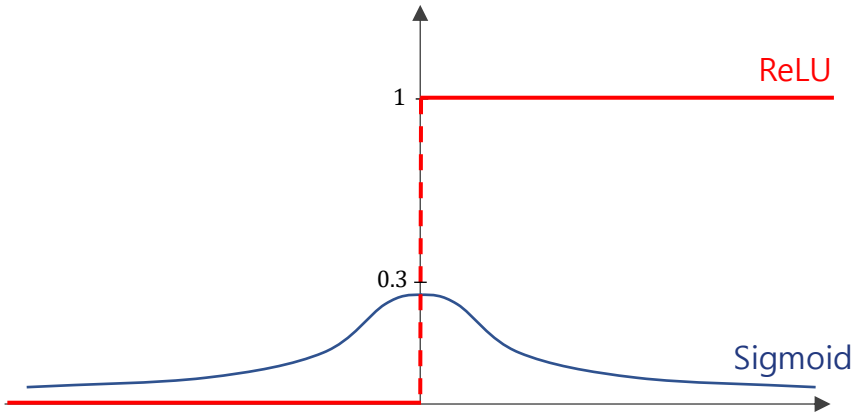
1) 활성화 함수의 변경

- 활성화 함수로 시그모이드(sigmoid) 함수나 탄젠트하이퍼볼릭(tanh) 함수 대신 **렐루(ReLU) 함수**를 이용
- Sigmoid 함수의 미분값은 최대값이 0.3을 넘지 않아, 곱 연산이 반복되면 0으로 수렴하는 반면,

  ReLU 함수의 미분값은 0 또는 1을 가지기 때문에, 곱연산이 반복되더라도 값이 0으로 수렴하지 않음



Sigmoid 함수와 ReLU 함수



Sigmoid 함수와 ReLU 함수의 미분

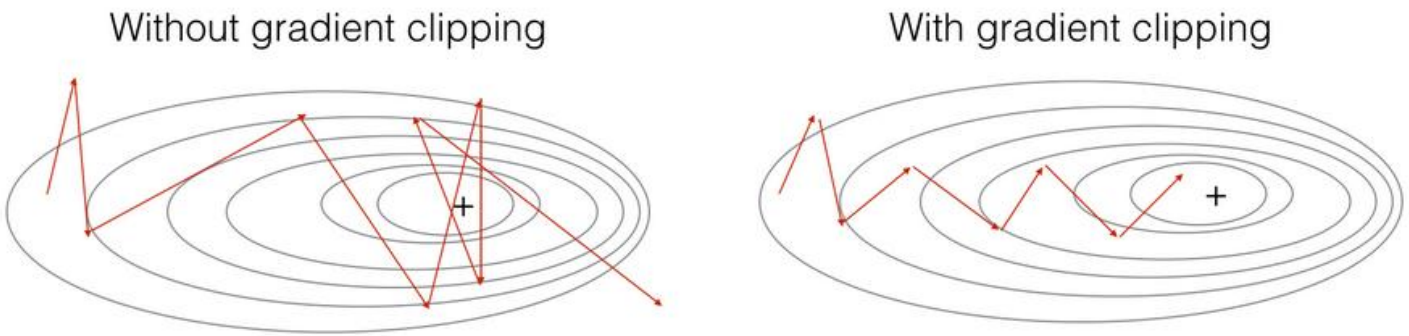
2. 학습 관련 기술 (Training Techniques)

4-7-4 Gradient Vanishing

2) 기울기 클리핑(Gradient Clipping)

- 기울기 폭주 현상을 방지하기 위하여, 기울기가 특정 값(Threshold)보다 커지면 기울기를 감소시키는 방법

$$\frac{\partial \epsilon}{\partial \theta} \leftarrow \begin{cases} \frac{threshold}{\|\hat{g}\|}\hat{g} & if \ \|\hat{g}\| \geq threshold \\ \hat{g} & otherwise \end{cases}$$



SNUH

2. 학습 관련 기술 (Training Techniques)

4-7-5 Gradient Vanishing

3) 가중치 초기화(Weight Initialization)

- 가중치의 초기값을 어떻게 설정하느냐에 따라 학습의 성능에 영향을 미칠 수 있음
- 잘못된 가중치 초기화 방법은 기울기 소실 현상을 발생시킴

(1) 모든 가중치를 0으로 초기화

- 모든 노드에 동일한 값이 역전파되어, 모든 가중치가 같은 결과를 학습
- 즉, 여러 가중치를 이용하는 의미가 없게 됨

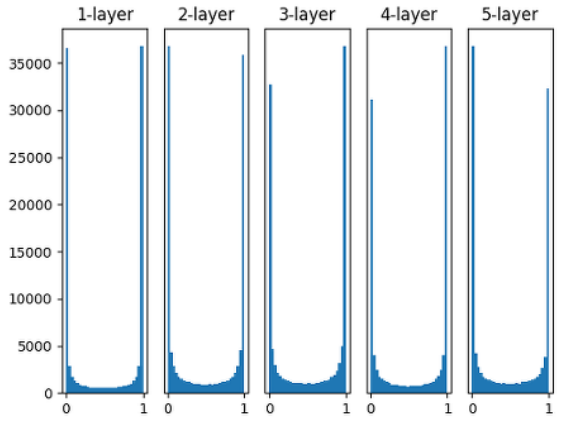
2. 학습 관련 기술 (Training Techniques)

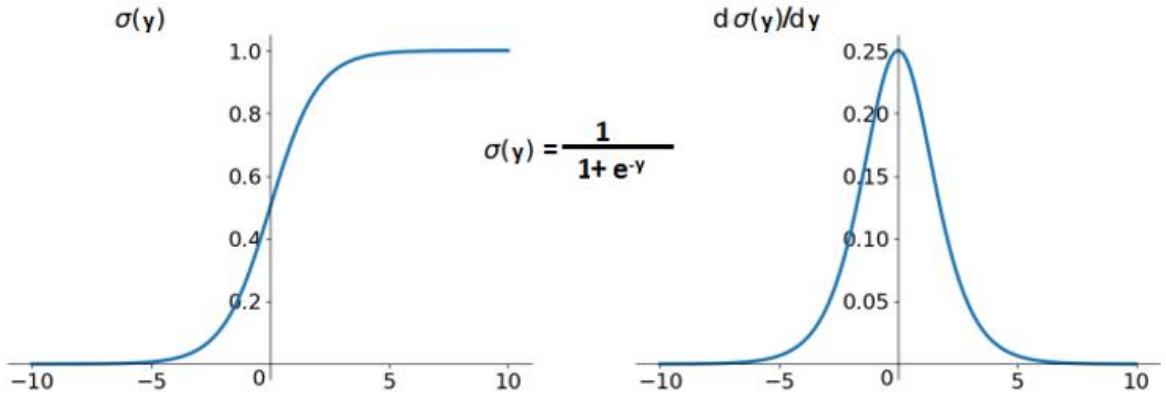
4-7-6	Gradient Vanishing

3) 가중치 초기화(Weight Initialization)

(2) 평균이 0, 표준편차가 1인 정규분포로 랜덤하게 초기화

- 시그모이드 함수(활성화 함수)의 결과값이 0 또는 1에 가깝게 분포하게 됨
- 이 경우, 기울기가 0에 가까워 학습이 잘 되지 않는 현상이 발생 → 기울기 소실





2. 학습 관련 기술 (Training Techniques)

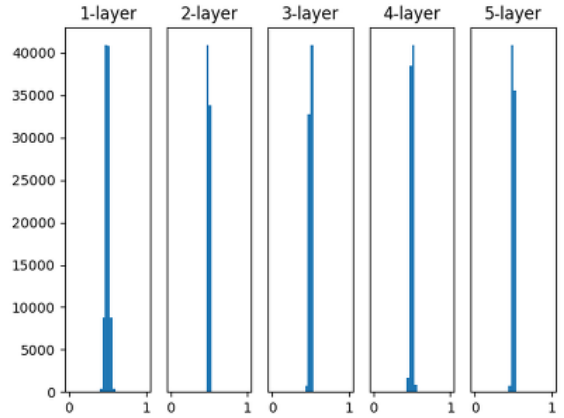
4-7-7	Gradient Vanishing

3) 가중치 초기화(Weight Initialization)

(3) 평균이 0, 표준편차가 0.01인 정규분포로 랜덤하게 초기화

- 시그모이드 함수(활성화 함수)의 결과값이 0.5에 가깝게 분포하게 됨
- 기울기 소실 현상은 발생하지 않지만, 모든 가중치가 거의 동일한 값을 출력한다는 것은,

  여러 가중치를 학습시키는 의미가 없다는 것과 동일



2. 학습 관련 기술 (Training Techniques)

4-7-8	Gradient Vanishing

3) 가중치 초기화(Weight Initialization)

(4) Xavier Initialization

- 균등 분포

$$W \sim Uniform(-\sqrt{\frac{6}{n_{in}+n_{out}}}, \sqrt{\frac{6}{n_{in}+n_{out}}})$$

(n_{in}은 입력 노드의 수, n_{out}은 출력 노드의 수)

- 정규 분포

$$W \sim N(0, \sqrt{\frac{2}{n_{in}+n_{out}}})$$

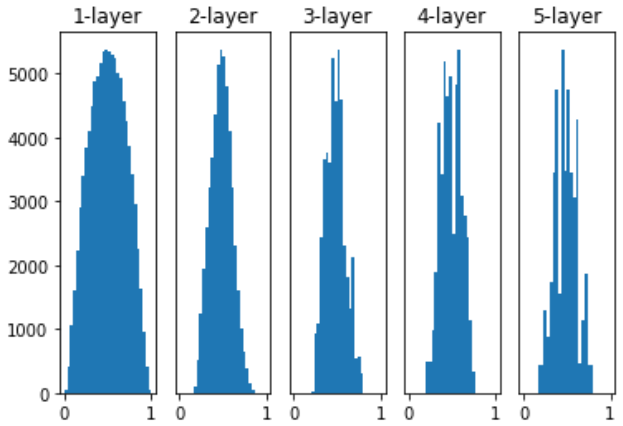
2. 학습 관련 기술 (Training Techniques)

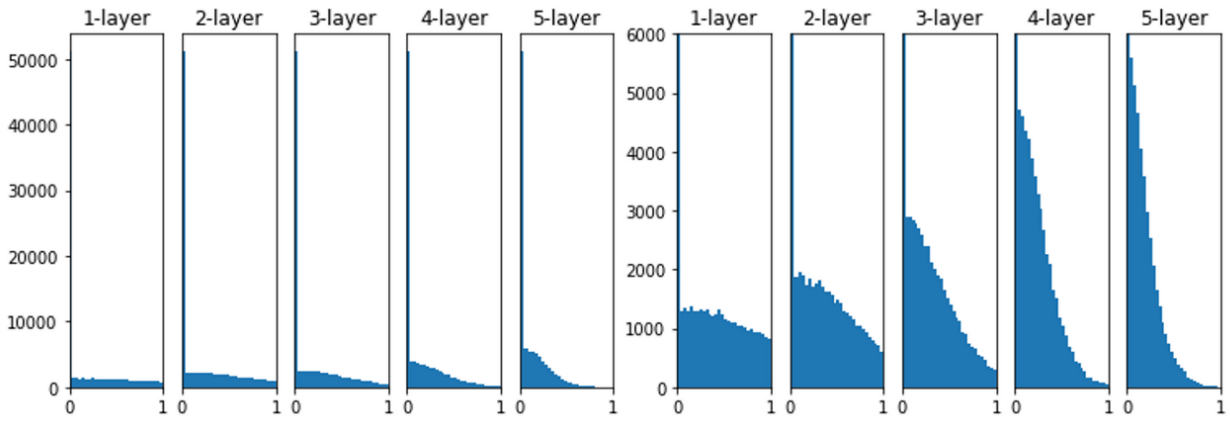
4-7-9 **Gradient Vanishing**

3) 가중치 초기화(Weight Initialization)

(4) Xavier Initialization

- 입력 노드와 출력 노드의 수를 고려하기 때문에, 각 레이어의 노드 크기에 따라 유연하게 대처 가능
- 각 레이어의 입/출력이 표준정규분포를 따르도록 조정되기 때문에, 선형 활성화 함수의 학습에 적합
- 하지만, 비선형 활성화 함수인 ReLU 함수에서는 출력값이 0으로 치우치게 됨





2. 학습 관련 기술 (Training Techniques)

4-7-10	Gradient Vanishing

3) 가중치 초기화(Weight Initialization)

(5) He Initialization

- 균등 분포

$$W \sim Uniform(-\sqrt{\frac{6}{n_{in}}}, \sqrt{\frac{6}{n_{in}}})$$

(n_{in}은 입력 노드의 수)

- 정규 분포

$$W \sim N(0, \sqrt{\frac{2}{n_{in}}})$$

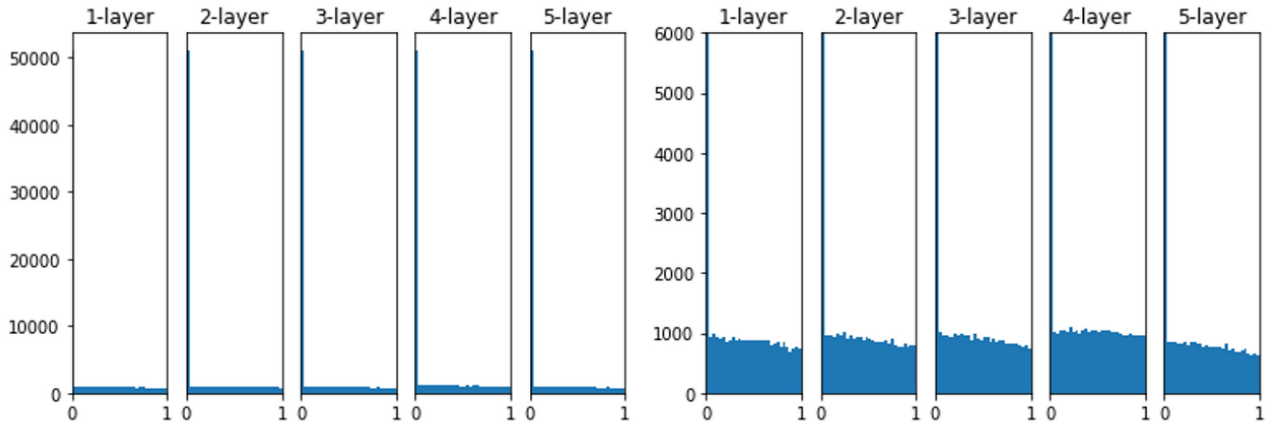
2. 학습 관련 기술 (Training Techniques)

4-7-11	Gradient Vanishing

3) 가중치 초기화(Weight Initialization)

(5) He Initialization

- ReLU 함수에 He Initialization을 적용한 결과, 출력값이 골고루 분포하는 것을 확인할 수 있음

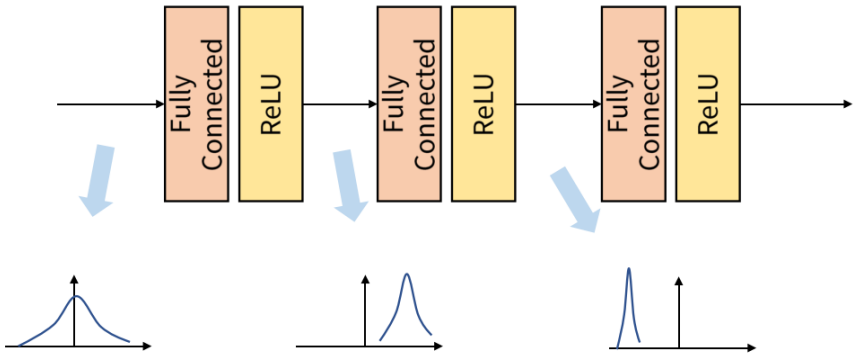


2. 학습 관련 기술 (Training Techniques)

4-7-12	Gradient Vanishing

4) 배치 정규화(Batch Normalization)

- 각 층에 들어가는 입력을 배치 단위로 정규화하여 학습의 효율성을 향상
- 내부 공변량 변화(Internal Covariate Shift)



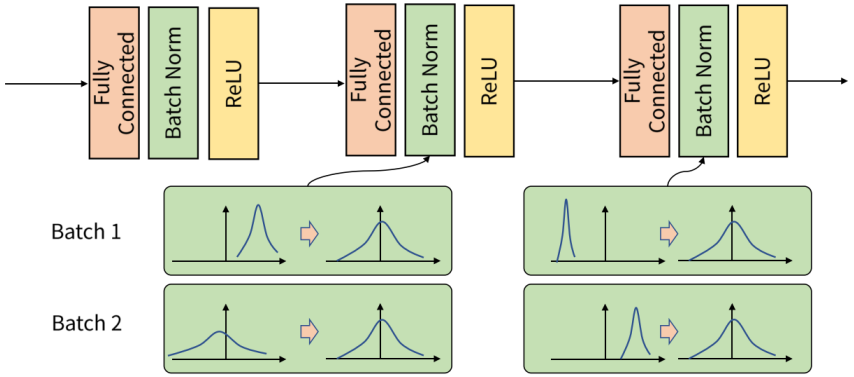
- 각 층마다 입력되는 값의 분포가 서로 달라지는 현상 → 기울기 소실 현상을 발생시킬 수 있음

2. 학습 관련 기술 (Training Techniques)

4-7-13	Gradient Vanishing

4) 배치 정규화(Batch Normalization)

- 각 층에 들어가는 입력을 배치 단위로 정규화하여, 내부 공변량 변화를 완화할 수 있음



$$BN(X) = \gamma\left(\frac{X - \mu_{batch}}{\sigma_{batch}}\right) + \beta$$

- γ와 β는 학습되어, 신경망을 통한 예측 시 사용됨

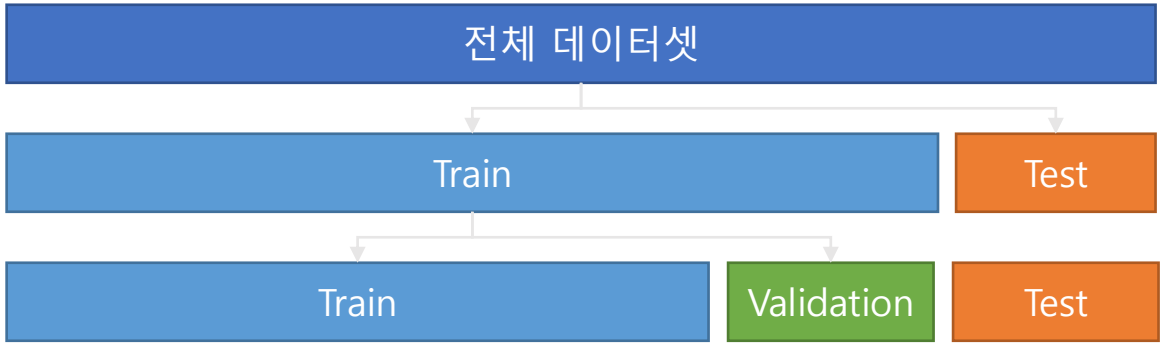
SNUH

3. 모델 평가 (Model Test)

4-8-1 Validation and Test sets

- **Train vs. Validation vs. Test**
  - 전체 데이터셋을 훈련(Train) 데이터셋, 검증(Validation) 데이터셋, 평가(Test) 데이터셋으로 나누어 사용하여야 함
  - 전체 데이터셋을 사용하여 훈련하고 이 중 일부로 평가를 할 수도 있지만,

    이 경우 평가에 사용되는 데이터는 '이미 학습한' 데이터이기 때문에 유의미한 평가라고 할 수 없음

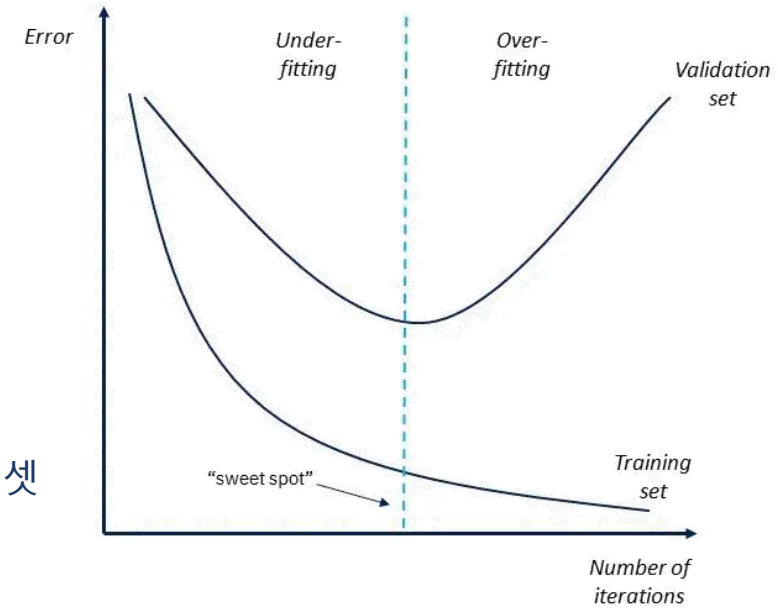


  - 일반적으로, Train : Validation : Test 데이터의 비율을 6:2:2 또는 8:1:1 등으로 나누어 사용함

3. 모델 평가 (Model Test)

4-8-2 **Validation and Test sets**

- **Train vs. Validation vs. Test**
  - 훈련(Train) 데이터셋
    - 실제로 모델을 학습시키기 위하여 사용되는 데이터
  - 검증(Validation) 데이터셋
    - Train 데이터셋으로 학습된 모델의 성능을 '측정'하기 위하여 사용되는 데이터
    - Validation 데이터셋은 학습의 정도 혹은 모델의 파라미터들을 결정하는 데에 사용되어, 학습 과정에 영향을 줌
    - 즉, Validation 데이터셋은 학습에 직접 사용되지는 않지만, 학습에 관여되는 데이터셋
  - 평가(Test) 데이터셋
    - 학습이 완료된 모델의 성능을 최종적으로 평가하기 위하여 사용되는 데이터



3. 모델 평가 (Model Test)

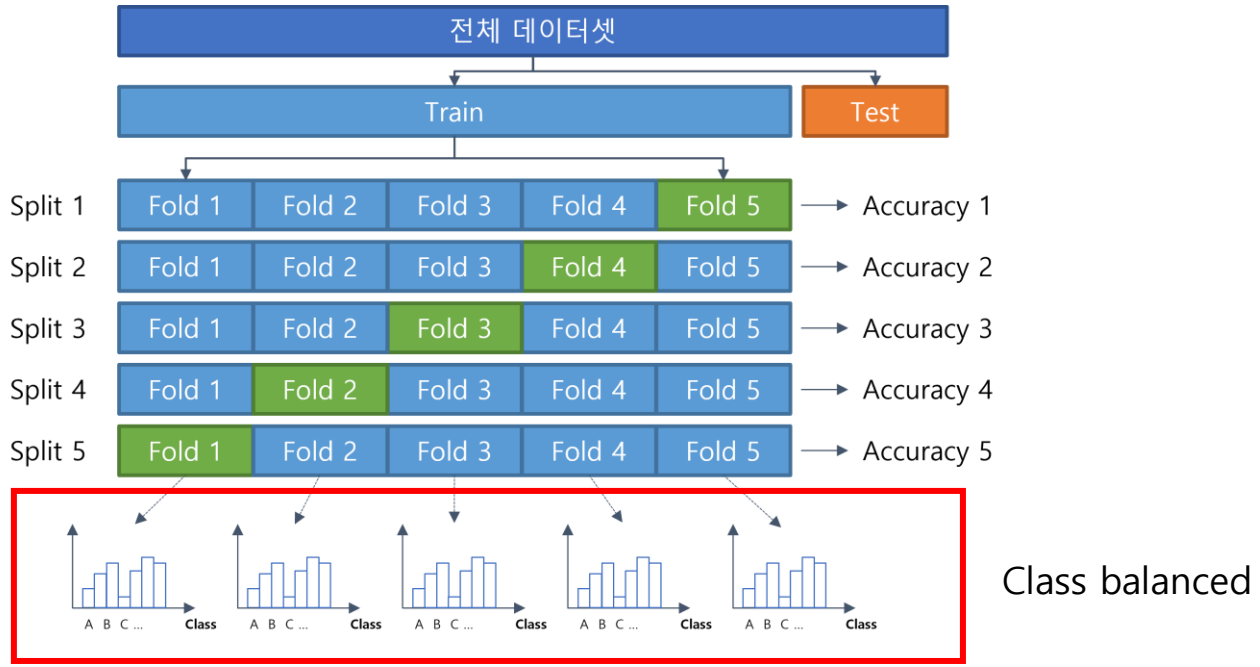
4-8-3 **Validation and Test sets**

- **k-Fold Cross Validation (k-겹 교차 검증)**
  - 단순히 Train 데이터셋과 Validation 데이터셋을 나누게 되면, Validation 데이터셋의 편향에 따라 학습이 과적합(Overfitting)될 수 있음. 즉, Validation에만 적합하도록 파라미터 등의 모델을 튜닝하게 됨
  - 이를 방지하기 위하여, Validation 데이터셋을 변경해가며 검증하는 교차 검증 방법이 사용됨

3. 모델 평가 (Model Test)

4-8-4 Validation and Test sets

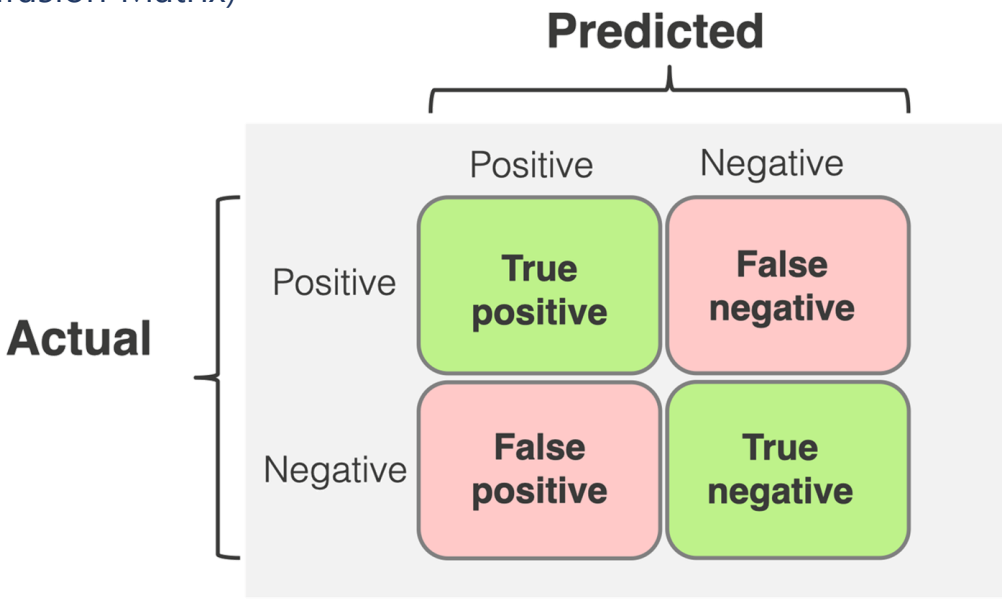
- **Stratified Cross Validation (계층별 교차 검증)**
  - 분류(Classification)을 위한 다중 클래스(Multi-class) 데이터셋에서, **전체 데이터셋의 클래스별 분포**를 고려하여 교차 검증하는 방법



3. 모델 평가 (Model Test)

4-8-5 Validation and Test sets

- **분류(Classification) 모델의 평가**
  - 오차 행렬(Confusion Matrix)



- "True" : 옳게 예측
- "False" : 틀리게 예측
- "Positive" : "Positive"로 예측
- "Negative" : "Negative"로 예측

3. 모델 평가 (Model Test)

4-8-6 Validation and Test sets

- **분류(Classification) 모델의 평가**
  - 정확도(Accuracy) – 전체 데이터 중 옳게 예측한 데이터 $\frac{TP + TN}{TP + TN + FP + FN}$
  - 재현율(Recall) – 실제 Positive 중 옳게 예측한 데이터 $\frac{TP}{TP + FN}$
  - 정밀도(Precision) – Positive로 예측한 데이터 중 실제 Positive인 데이터 $\frac{TP}{TP + FP}$
  - 특이도(Specificity) – 실제 Negative 중 옳게 예측한 데이터 $\frac{TN}{TN + FP}$

Predicted Values \ Actual Values	Positive (1)	Negative (0)
Positive (1)	TP	FP
Negative (0)	FN	TN

3. 모델 평가 (Model Test)

4-8-7 Validation and Test sets

- **분류(Classification) 모델의 평가**
  - 정밀도와 재현율은, 분류의 임계치(Threshold)를 조정함으로써 어느 정도의 임의로 조정이 가능함
  - 따라서, 두 지표 중 하나만 이용하는 것은 신뢰하기 어려움

  - F1 점수(F1 Score) – 정밀도와 재현율을 모두 이용한 지표

$$F_1 = \frac{2}{\frac{1}{recall} + \frac{1}{precision}} = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

  - 정밀도와 재현율의 조화평균을 이용하기 때문에, 데이터의 불균형이 심한 경우에도 모델을 잘 평가한다고 할 수 있음

3. 모델 평가 (Model Test)

4-8-8 Validation and Test sets

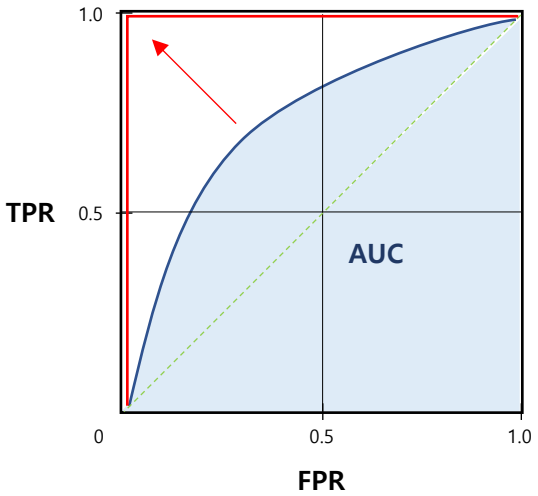
- **분류(Classification) 모델의 평가**
  - ROC Curve
    - 가로축을 FPR(False Positive Rate), 세로축을 TPR(True Positive Rate)으로 표현한 그래프

$$TPR = \frac{TP}{TP + FN} = Recall$$

$$FPR = \frac{FP}{FP + TN} = 1 - Specificity$$

    - ROC 곡선 아래의 면적을 AUC(Area Under Curve)라고 하며,

      AUC가 높을수록 모델의 성능이 높음을 의미
    - 즉, ROC 커브가 좌측 상단에 가까울수록 모델의 성능이 높음



SNUH

3. 모델 평가 (Model Test)

4-8-9 **Validation and Test sets**

- **회귀(Regression) 모델의 평가**
  - MAE(Mean Absolute Error, 평균 절대 오차) $MAE = \frac{\sum|y - \hat{y}|}{n}$
  - MSE(Mean Squared Error, 평균 제곱 오차), RMSE(Root Mean Squared Error, 평균 제곱근 오차)

$$MSE = \frac{\sum(y - \hat{y})^2}{n}, \qquad RMSE = \sqrt{\frac{\sum(y - \hat{y})^2}{n}}$$

  - MSLE(Mean Squared Log Error, 평균 제곱 로그 오차), RMSLE(Root Mean Squared Log Error, 평균 제곱근 로그 오차)

$$MSLE = \frac{\sum(\log(y+1) - \log(\hat{y}+1))^2}{n}, \qquad RMSLE = \sqrt{\frac{\sum(\log(y+1) - \log(\hat{y}+1))^2}{n}}$$