PyTorch로 딥러닝 제대로 배우기

- 중급편 -

Part5-1. 과대적합 & 과소적합

강사: 김 동 희

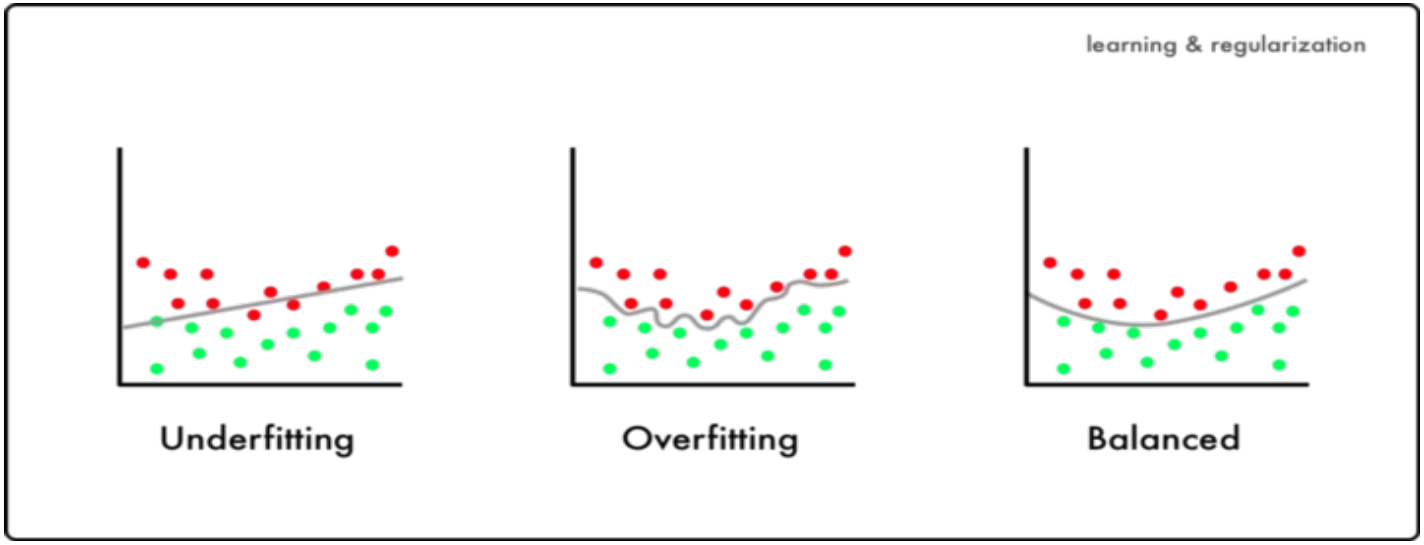
목차

I. 과대적합(Overfitting)

1. Overfitting의 정의

❑ Overfitting

- 모델이 학습 데이터(Training data)에 과하게(over) 맞춰진(fitted) 상황
- 즉, 모델이 문제를 해결하는데 있어서 **범용성(Generality)**이 떨어지는 문제



2. 원인

❑ 데이터

- 학습 데이터가 다양한 특징을 반영하지 못하고 편향되어 있는 경우
- 데이터 자체가 부족한 경우
- 데이터의 특징이 너무 다양해서 결정 경계를 만들기 어려운 경우

❑ 학습 알고리즘

- Loss 함수에 의해 오차 자체가 너무 데이터에 fit하게 모델이 학습하는 경우

❑ 모델

- 데이터에 비해 너무 복잡한 모델을 활용하는 경우
- 모델에 너무 많은 정보를 담으려고 하는 경우
- 모델을 필요 이상으로 학습을 진행하는 경우

3. 해결방안 - 데이터

❑ 데이터 증가

- 모델이 더 다양한 분포의 데이터를 학습 가능하도록 유도 해야함
- 하지만, 단순히 데이터를 증가하는 것은 효과가 크지 않고 많은 비용이 발생하여 비효율적임

❑ Data augmentation

- 제한된 데이터 내에서 다양성을 부여하여 일반적 특징을 학습시키기 위한 방법
- 데이터를 Rotate, Cut, Flip 등을 하여 변형 시킴

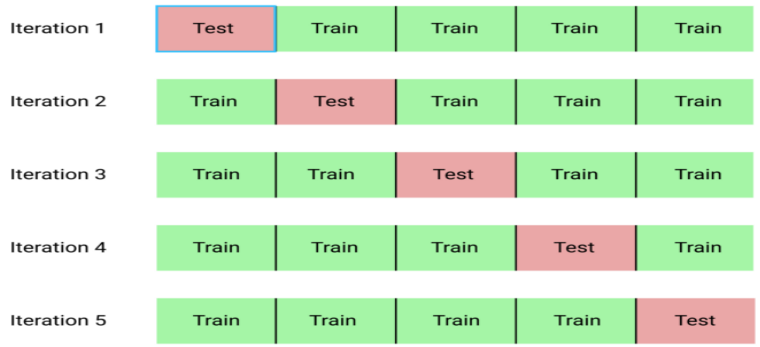
3. 해결방안 - 데이터

❑ Feature selection

- 제한된 데이터 내에서 데이터를 일반화 시키기 위한 방법
- 딥러닝에서는 잘 활용되지 않지만, 데이터를 더 늘릴 수 없는 경우 효과적

❑ Cross-validation

- 학습 데이터와 테스트 데이터를 K 그룹으로 분리하여, 각 epoch마다 Test 데이터를 교차 적용하는 법
- Test 데이터를 모두 학습에 사용
- 역시 딥러닝에서는 잘 활용되지 않지만 데이터가 적은 경우에는 활용 할 수 있음

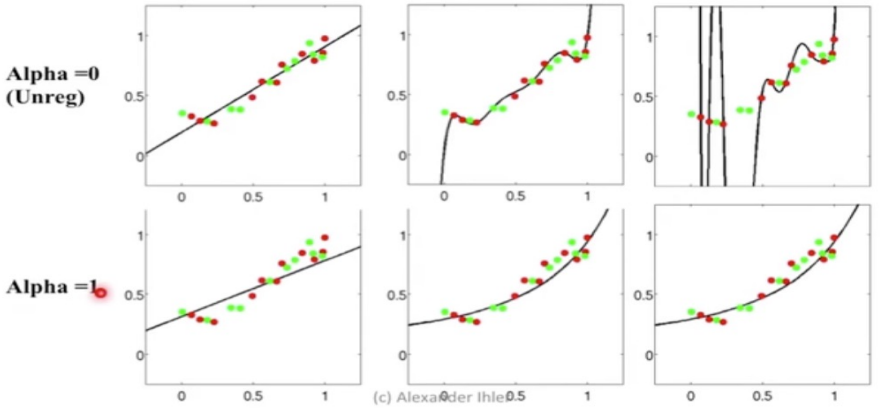


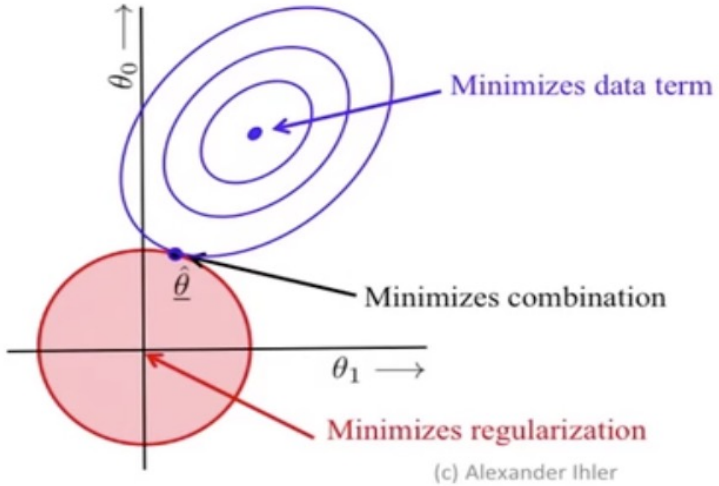
출처: https://towardsdatascience.com/8-simple-techniques-to-prevent-overfitting-4d443da2ef7d

3. 해결방안 - Regularization

❑ Regularization

- Loss 함수 결과값에 너무 fit해서 학습하는 것을 방지
- 즉, Error 값에 대한 패널티를 부여하는 것
- $L = Error(y, \hat{y}) + \lambda\, regularization$





❑ Balancing the error between regularization term

- Regularization term이 포함된 loss가 가장 저점이 되는 지점은 두 point가 맞닿는 부분
- 맞닿는 지점을 만들기 위해서는 두 값 중 하나는 증가해야 함. 증가 없이는 교차점을 만들 수 없음

출처: https://youtu.be/sO4ZirJh9ds

3. 해결방안 - Regularization

❑ Norm

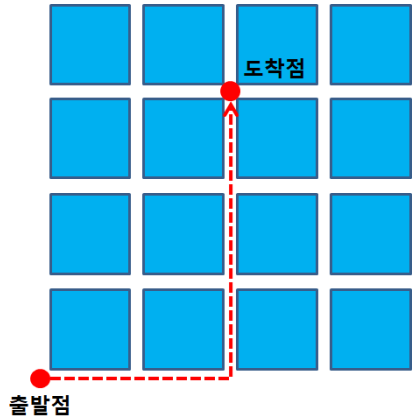
- 벡터의 크기를 나타내는 값

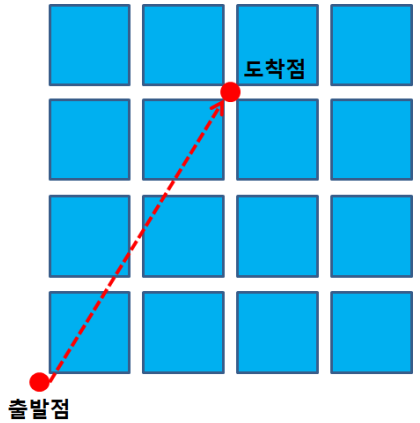
❑ L1 Norm

- 흔히 Manhattan distance라고 말함
- 두 벡터의 최단 거리가 아닌 벡터의 모든 성분의 절대값을 더한 값
- $\|\boldsymbol{x}\|_1 = \sum_{i=1}^{n} |x_i|$

❑ L2 Norm

- 벡터간의 직선 거리를 나타냄
- $\|\boldsymbol{x}\|_2 = \left(\sum_{i=1}^{n} |\boldsymbol{x_i}|^2\right)^{\frac{1}{2}} = \sqrt{\sum_{i=1}^{n} |\boldsymbol{x_i}|^2}$

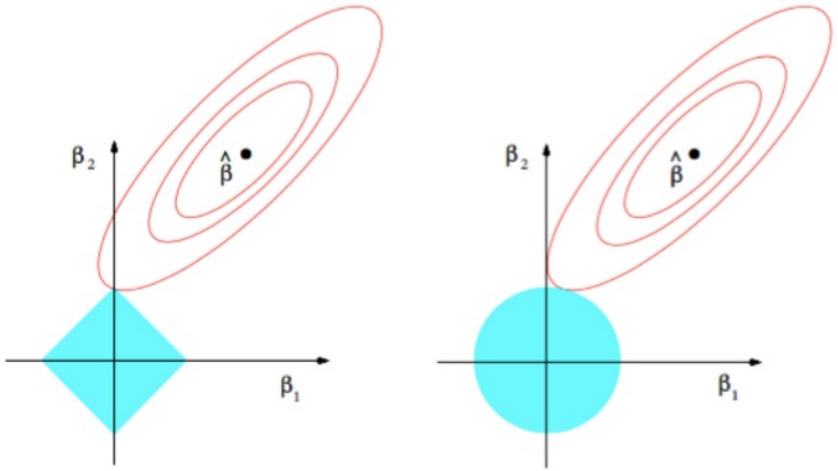


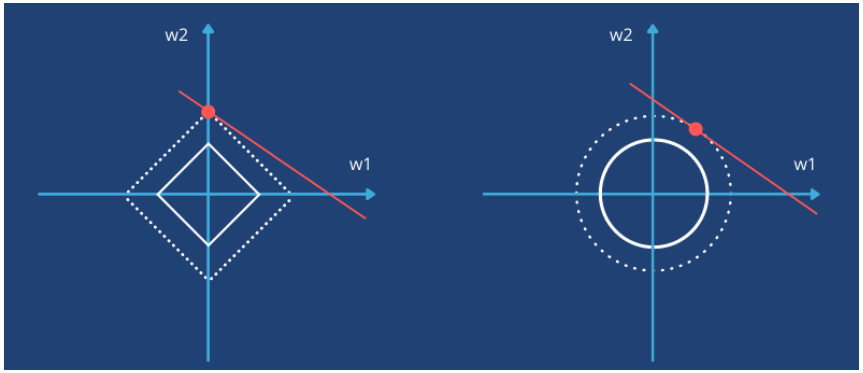


3. 해결방안 - Regularization

❑ L1 vs L2 Norm

- L1 regularization은 Sparsity 부여 가능
  - ➢ Loss 값과 Regularization 값이 교차하는 지점이 Axis인 경우
- L2 regularization은 모양의 한계 때문에 Axis위에서 교차점을 찾기 어려움





출처1: https://satishkumarmoparthi.medium.com/why-l1-norm-creates-sparsity-compared-with-l2-norm-3c6fa9c607f4
출처2: https://youtu.be/76B5cMEZA4Y

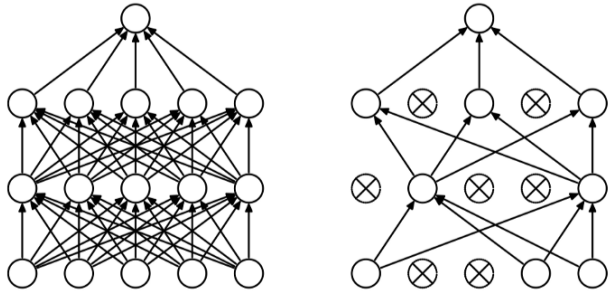
3. 해결방안 - 모델

❑ Remove Hidden units or Layers

- 데이터에 비해 모델이 너무 복잡한 경우, 모델을 간소화 시킬 수 있음
- 모델은 간소화 시킨다는 것은 모델의 크기(깊이 또는 넓이)를 줄이는 것을 의미

❑ Dropout

- Dropout은 학습하는 단계에서 몇 개의 neuron은 비활성화 상태로 만들고, Test 단계에서 모두 활성화 하는 방법
- Hidden Unit을 줄이는 효과와 비슷함



(a) Standard Neural Net (b) After applying dropout.

❑ Early stopping

- 모델 학습에서 Validation set의 오류가 증가하는 현상에서 학습을 중지하는 방법
- Overfitting을 예방하는 가장 쉬운 방법이나, 아직 학습이 덜 되었을 가능성에서 좋은 방법은 아님

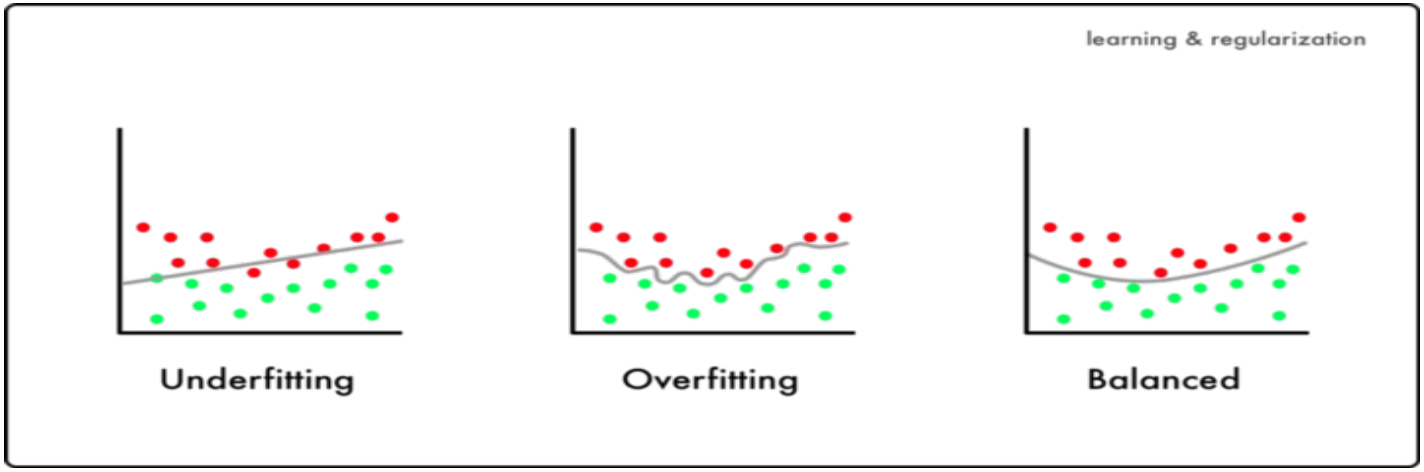
출처: https://towardsdatascience.com/8-simple-techniques-to-prevent-overfitting-4d443da2ef7d

II. 과소적합(Underfitting)

1. Underfitting의 정의

❑ Underfitting

- 모델이 학습 데이터(Training data)에서 입력 데이터와 출력 결과에서 관계를 잘 못 찾아 내는 현상
- 학습 데이터와 테스트 데이터 모두에서 정확도가 낮게 측정될 때 underfitting 현상이 발생하고 있음을 추정
- 데이터에 비해 모델이 너무 간단하거나, 학습 시간이 짧거나, learning rate가 적합하지 않을 때 발생



2. 해결방안

❑ Decrease regularization

- 정규화는 overfitting에서 확인했듯, 모델의 복잡도를 낮추기 위해 사용
- 이를 반대로 적용하여, regularization을 낮추는 것이 underfitting에서 도움이 될 수 있음

❑ Increase the duration of training

- 학습이 아직 되고 있지 않은 시점일 수 있음
- 따라서 epoch을 더 늘리거나, learning rate를 조절하는 것이 도움이 됨

❑ Feature selection

- 모델에 비해 데이터가 너무 복잡 하면서 발생 가능 할 수 있음
- 데이터의 양이 적고, feature가 많은 경우 불필요한 정보를 삭제하는 것이 도움 됨

감사합니다.