인공지능에서 자주 등장하는 개념

인공지능을 배우는데 중간 중간
등장하는 용어와 개념들을
한번 살펴보자.

NORM이란?

Norm은 벡터의 크기(또는 길이)를 측정하는 방법이다. 두 벡터 사이의 거리를 측정하는 방법이기도 하며, '노름 ' 이라고 읽는다.

$$\|\mathbf{x}\|_p := \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}.$$

* 여기서 p 는 Norm 의 차수를 의미한다.
  (p = 1 이면 L1 Norm 이고, P = 2 이면 L2 Norm 이다.)
* n은 해당 벡터의 원소 수이다.

위키피디아 참조 : https://en.wikipedia.org/wiki/Norm_(mathematics)

L1 Norm

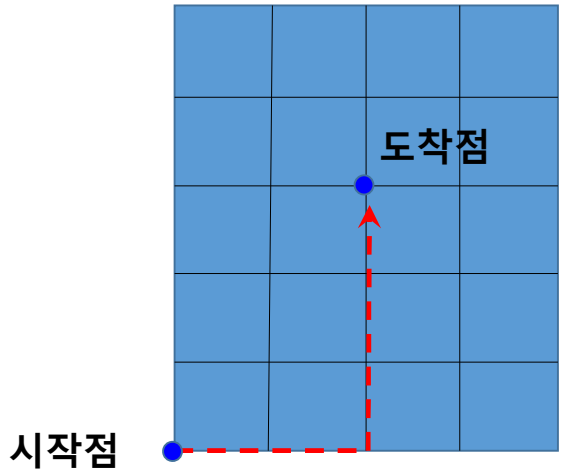
L1 Norm은 맨하튼 놈(Manhattan norm) 또는 택시캡 (Taxicab norm)이라고 불려진다.

L1 Norm은 벡터의 모든 성분의 절대값을 더한다.

만약 p가 (2 , 3)이면 $p = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$ $\|p\|_1 = |2| + |3| = 5$

L1 norm 은아래와 같이 표기 한다.

$\|p\|_1$



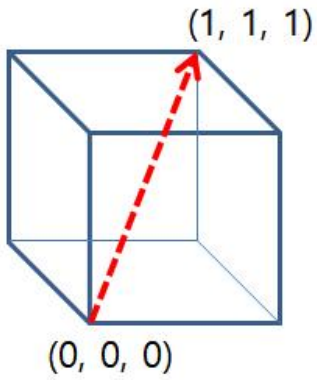
L1 Norm

L1 Norm은 벡터 p, q의 각 원소들의 차이의 절대값의 합이다.

$$d_1(\mathbf{p},\mathbf{q}) = \|\mathbf{p}-\mathbf{q}\|_1 = \sum_{i=1}^{n} |p_i - q_i|, \text{ where } (\mathbf{p},\mathbf{q}) \text{ are vectors } \mathbf{p} = (p_1, p_2, \ldots, p_n) \text{ and } \mathbf{q} = (q_1, q_2, \ldots, q_n)$$

ex) p = (4,2,3), q = (1,2,3) |4-1| + |2-2| + |3-3| = 3 이다.

p가 (1,1,1)이면, 이 벡터의 차원은 3이다. 차원이 3이면 3차원 공간에 이 벡터를 그릴 수 있다는 것을 의미한다.



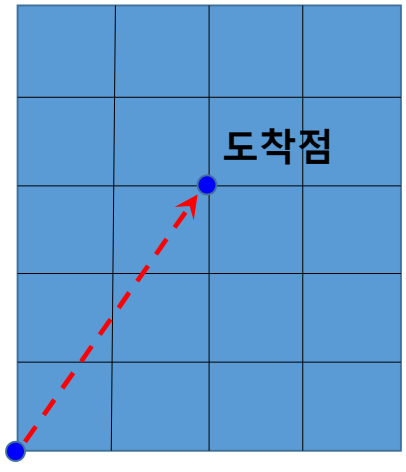
L2 Norm

L2 Norm은 출발점에서 도착점까지의 직선거리를 측정하며, 유클리디안 거리 (Euclidean distance) 라고 부른다.

만약 p가 (2 , 3)이면 $p = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$ $\|p\|_2 = \sqrt{2^2 + 3^2} = \sqrt{13}$

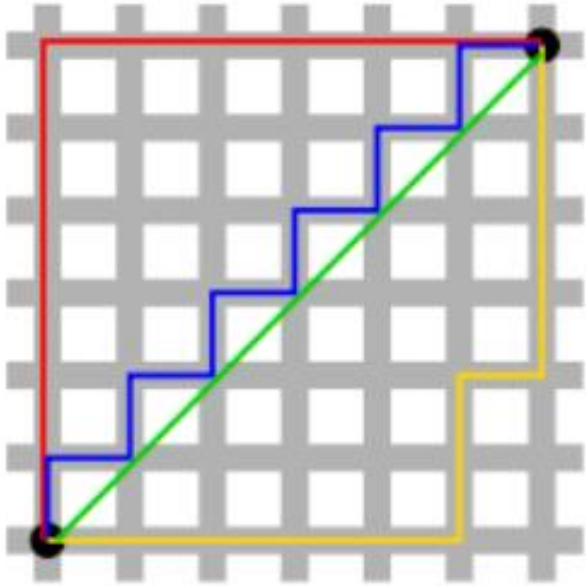
L2 norm 은아래와 같이 표기 한다.

$\|p\|_2$



$\|x\|_2 = \sqrt{x_1^2 + \cdots + x_n^2}$

L1 Norm VS L2 Norm



검정색 두 점 사이의 거리를 측정하는데
L1 Norm은 빨간색, 파란색, 노란색 선으로
표현할 수 있고, L2 Norm은 초록색 선으로만
표현될 수 있다.

즉, L1 Norm은 여러가지 Path를 갖지만,
L2 Norm은 하나의 Path만 갖는다.

L1 Loss / L2 Loss

L1 Loss

$$L = \sum_{i=1}^{n} |y_i - \hat{y}|$$

실제 값과 예측치 사이의 차이(오차)값의 절대 값을 계산하고,
그 오차 들의 합을 L1 Loss라고 한다.
Least absolute Errors(LAE)라고도 부른다.

L2 Loss

$$L = \sum_{i=1}^{n} (y_i - \hat{y})^2$$

L2 Loss는 오차의 제곱의 합을 의미한다.
Least squares error(LSE) 라고도 부른다.

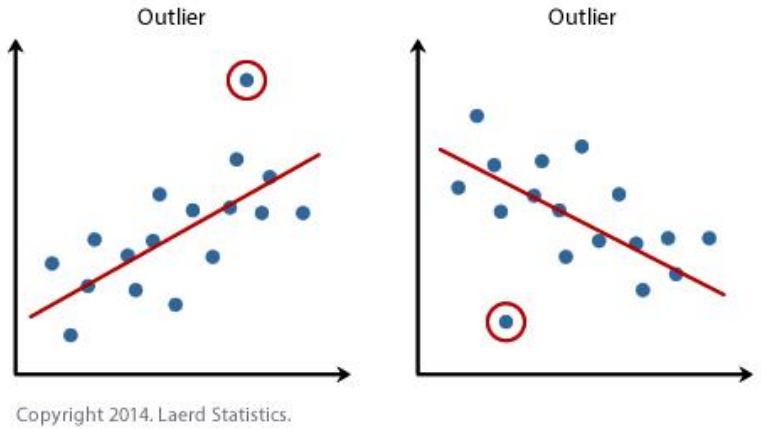
L1 Loss VS L2 Loss

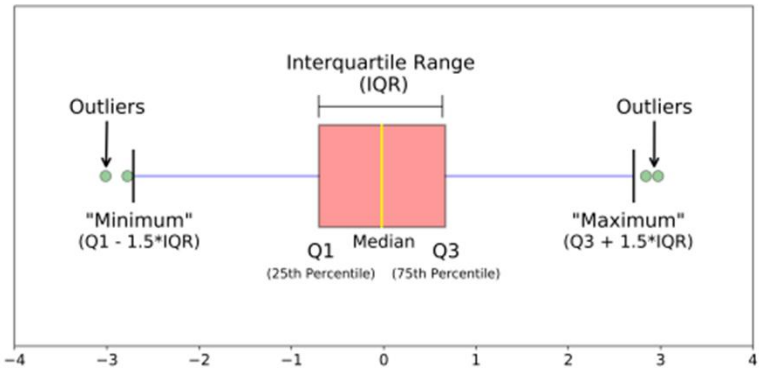
L2 Loss는 직관적으로 오차의 제곱을 더하기 때문에 이상치(Outlier)에 더 큰 영향을 받는다.

Outlier에 덜 영향을 받기 원한다면, L1 Loss를 사용하고, Outlier에 영향 받는 부분을 드러내고자 한다면 L2 Loss를 사용하는 것이 좋다.

이상치(Outlier)

정상 범위 밖에 있는 값을 뜻한다. 잘못 입력한 값일 수도 있지만 실제 값일 수도 있다. 이런 이상치들은 전체 데이터 분포의 특성에 영향을 미친다. 품질 관리에 있어서 불량을 찾을 때 확인해 보기도 한다. 이상치를 제외하고 분석을 할지, 포함해서 분석을 할지는 상황에 따라 판단해야 한다.

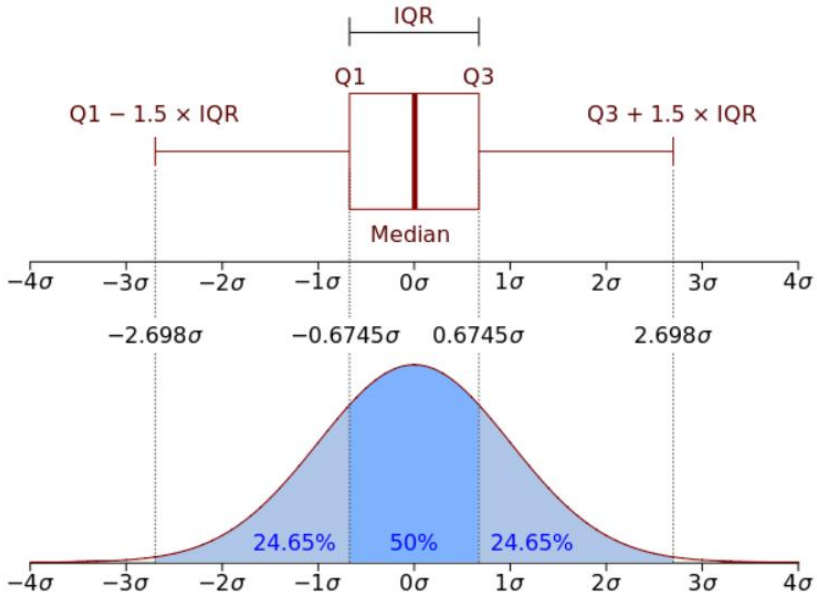




이상치(Outlier)

관측된 데이터 중 이상하게도, 특이하게도 평균 데이터 (값)와 크게 다른 데이터를 말한다. 이러한 outlier 데이터가 머신러닝/딥러닝 모델이 학습하는 과정에 입력되면 weight가 급격하게 커지거나 작아질 수 있다 (튕긴다라고도 표현한다). 그렇기에 outlier 데이터를 제거하는 과정이 필수적이다.

Outlier를 탐지하는 방법은 다양하다. 그 중 가장 널리 사용되는 방법은 IQR Rule이다. 이 방법은 이 사분범위를 바탕으로 Q3 + 1.5*IQR 이상, Q1–1.5*IQR 이하의 값을 outlier로 정의하는 것이다.

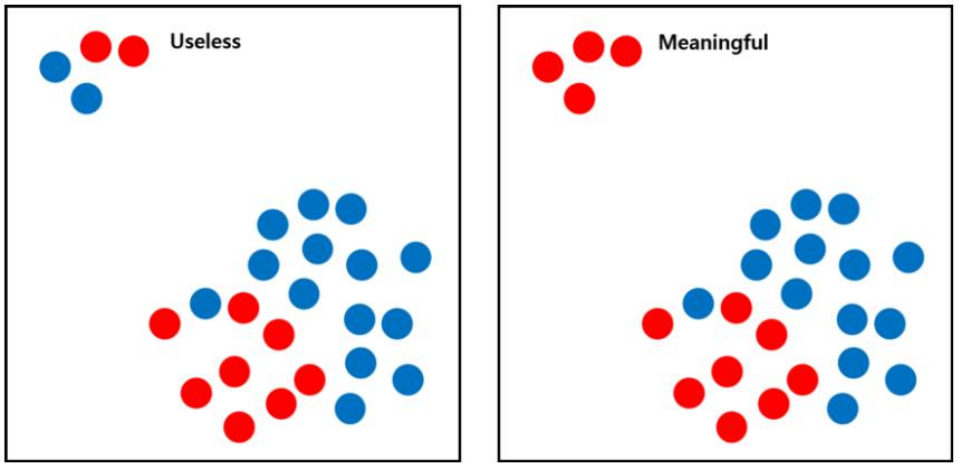


[Outlier는 모두 제거해야할까? — Outlier detection, IQR]

이상치(Outlier)

자고로 탐지된 outlier 데이터가 정말 불필요하다면 outlier 데이터의 class diversity가 아주 높아야 된다고 생각한다. 즉, outlier 데이터가 100개 탐지되었다면 class가 50:50의 비율에 가까워야 불필요한 outlier이다. 하지만 만약 class의 비율이 80:20 혹은 90:10이라면, 극단적으로 100:0이라면 이 데이터를 outlier라고 보아야할까? 아닐 것이다. 이 100개의 데이터는 의미있는 outlier 그룹일 것이다.

만약 오른쪽의 상황임에도 outlier 데이터를 제거해버린다면 큰 실수를 범하는 것이다. 이 데이터를 제거한다면 학습 모델을 테스트하는 과정에서 큰 오차를 발생시킬 수 있다. 그렇기에 outlier는 이상치를 의미하는 것이지 쓸모없는 데이터라 정의하면 오판이다. 제거하기 전에 분석을 해야할 필요가 (크게) 있다.



[Outlier는 모두 제거해야할까? — Outlier detection, IQR]

Regularization

'정규화'라고 하고, 모델 복잡도에 대한 패널티를 준다.

패널티는 weight를 조정하는데 제약을 주는 효과가 발생한다.

이는 Overfitting을 막기 위해 사용한다.

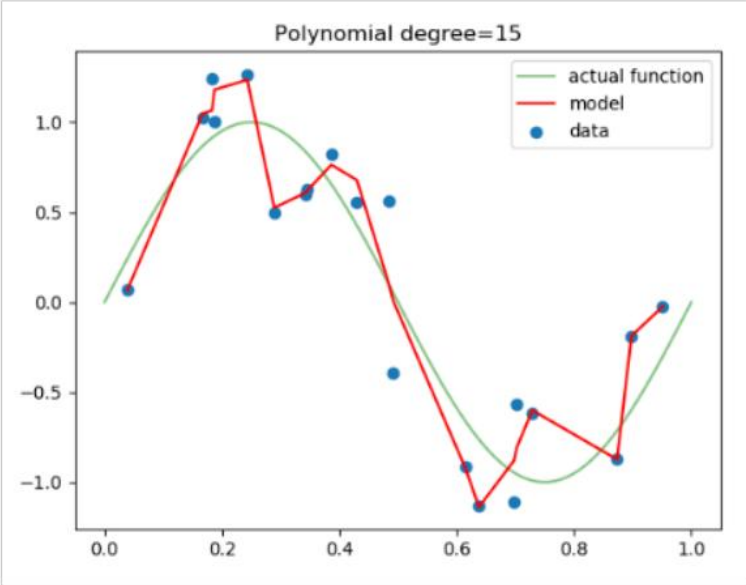
L1 regularization, L2 regularizaion 등의 종류가 있다.

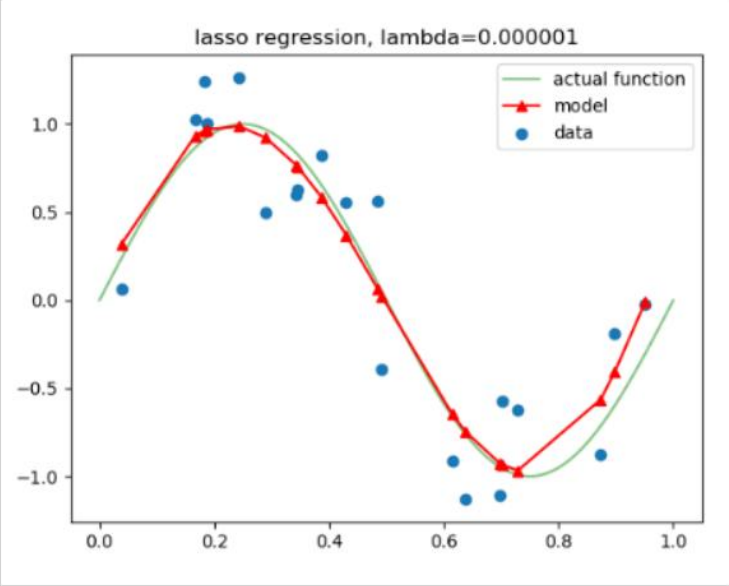
L1: LASSO(라쏘), 마름모

L2: Lidge(릿지), 원

Regularization

모델을 학습할 때 단순히 손실 함수(loss function) 값이 작아지는 방향으로만 진행하면, 특정 가중치가 큰 값을 갖기 때문에 모델의 일반화 성능이 떨어지는 현상이 발생한다. Regularization은 이와 같이 특정 가중치가 너무 과도하게 커지지 않도록 하여 모델을 아래와 같이 만들어 준다.





L1 Regularization

모델을 학습할 때 단순히 손실 함수(loss function) 값이 작아지는 방향으로만 진행하면, 특정 가중치가 큰 값을 갖기 때문에 모델의 일반화 성능이 떨어지는 현상이 발생한다. Regularization은 이와 같이 특정 가중치가 너무 과도하게 커지지 않도록 하여 모델을 아래와 같이 만들어 준다.

$$Cost = \frac{1}{n}\sum_{i=1}^{n}\{L(y_i, \hat{y}_i) + \frac{\lambda}{2}|w|\}$$

$L(y_i, \hat{y}_i)$: 기존의 Cost function

L1 Regularization 은 위 수식처럼 표현할 수 있다. 논문에 따라서 앞에 분수로 붙는 1/n 이나 1/2 가 달라지는 경우가 있는데 L1 Regularization 의 개념에서 가장 중요한 것은 cost function 에 가중치의 절대값을 더해준다는 것이다. 기존의 cost function 에 가중치의 크기가 포함되면서 가중치가 너무 크지 않은 방향으로 학습 되도록 합니다. 이때 λ 는 learning rate(학습률) 같은 상수로 0에 가까울 수록 정규화의 효과는 없어진다.

L1 Regularization 을 사용하는 Regression model 을 **L**east **A**bsolute **S**hrinkage and **S**election **O**perater(Lasso) Regression 이라고 부른다.

L2 Regularization

모델을 학습할 때 단순히 손실 함수(loss function) 값이 작아지는 방향으로만 진행하면, 특정 가중치가 큰 값을 갖기 때문에 모델의 일반화 성능이 떨어지는 현상이 발생한다. Regularization은 이와 같이 특정 가중치가 너무 과도하게 커지지 않도록 하여 모델을 아래와 같이 만들어 준다.

$$Cost = \frac{1}{n}\sum_{i=1}^{n}\{L(y_i, \hat{y}_i) + \frac{\lambda}{2}|w|^2\}$$

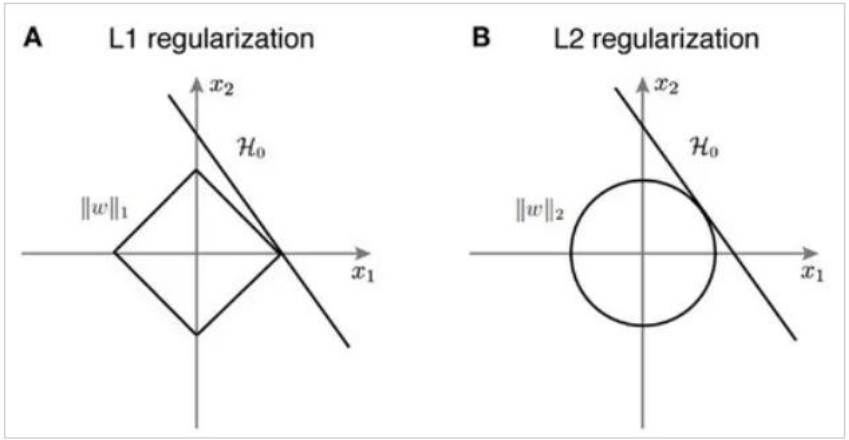
기존의 cost function 에 가중치의 제곱을 포함하여 더함으로써 L1 Regularization 과 마찬가지로 가중치가 너무 크지 않은 방향으로 학습되게 된다. 이를 Weight decay 라고도 한다.
L2 Regularization을 사용하는 Regression model 을 Ridge Regression이라고 부른다.

L1 Regularization VS L2 Regularization

Regularization 의 의미를 다시 한번 생각해보면, 가중치 w 가 작아지도록 학습한 다는 것은 결국 Local noise 에 영향을 덜 받도록 하겠다는 것이며, 이는 Outlier 의 영향을 더 적게 받도록 하겠다는 것이다.

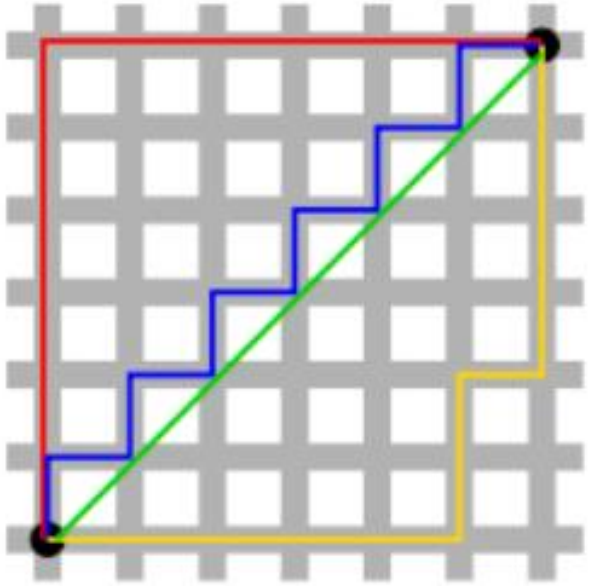
$$a = (0.3, -0.3, 0.4) \qquad ||a||_1 = |0.3| + |-0.3| + |0.4| = 1$$
$$b = (0.5, -0.5, 0) \qquad ||b||_1 = |0.5| + |-0.5| + |0| = 1$$

$$||a||_2 = \sqrt{0.3^2 + (-0.3^2) + 0.4^2} = 0.583095$$
$$||b||_2 = \sqrt{0.5^2 + (-0.5^2) + 0^2} = 0.707107$$



L1 Regularization VS L2 Regularization

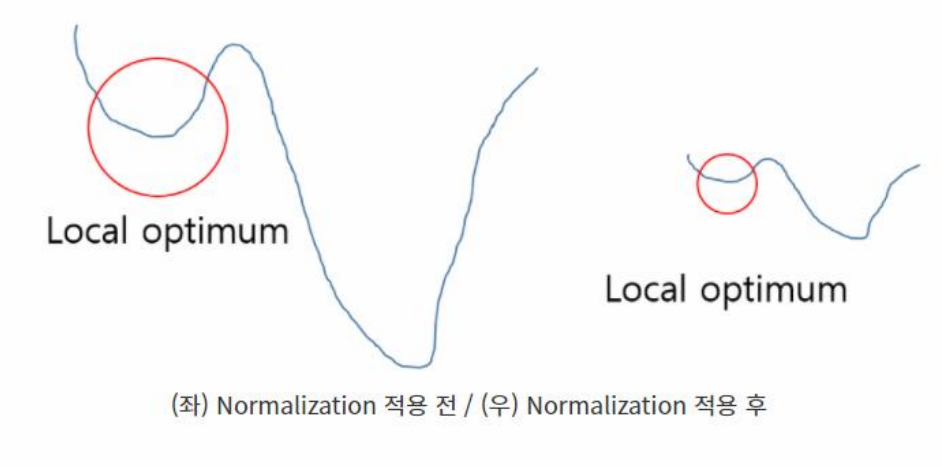
L2 Norm 은 각각의 벡터에 대해 항상 Unique 한 값을 내지만, L1 Norm 은 다르게 표현할 수 있다.
L1 Norm은 특정 Feature를 빼는 작업이 쉽고, L2 Norm은 특정 Feature를 빼는 작업이 상대적으로 어렵다고 할 수 있다.



Normalization

- 값의 범위(Scale)를 0~1 사이의 값으로 바꾸는 것이다.
- 학습 전에 scaling한다.
- scale이 큰 feature의 영향이 비대해지는 것을 방지하는 목적이 있다.
- Local Minima에 빠질 위험이 감소한다. (학습 속도의 향상)
- Scikit-learn에서는 MinMaxScaler를 사용하면 된다.

$$\frac{x - x_{min}}{x_{max} - x_{min}}$$



(좌) Normalization 적용 전 / (우) Normalization 적용 후

Normalization을 하게 되면 그래프를 다음과 같이 바꿔주는 효과가 있다.

Batch Normalization 참조

Standardization

- 값의 범위(scale)를 평균 0, 분산 1이 되도록 변환
- 학습 전에 scaling하는 것
- scale이 큰 feature의 영향이 비대해지는 것을 방지
- Local Minima에 빠질 위험 감소(학습 속도 향상)
- 정규분포를 표준정규분포로 변환하는 것과 같음
- Z-score(표준 점수)
- -1~1 사이에 68%가 있고, -2~2 사이에 95%가 있고, -3~3 사이에 99%가 있음
- -3~3의 범위를 벗어나면 outlier일 확률이 높음
- Scikit-learn에서 StandardScaler를 사용하면 된다.

$$\frac{x-\mu}{\sigma} \quad (\mu : 평균, \sigma : 표준편차)$$