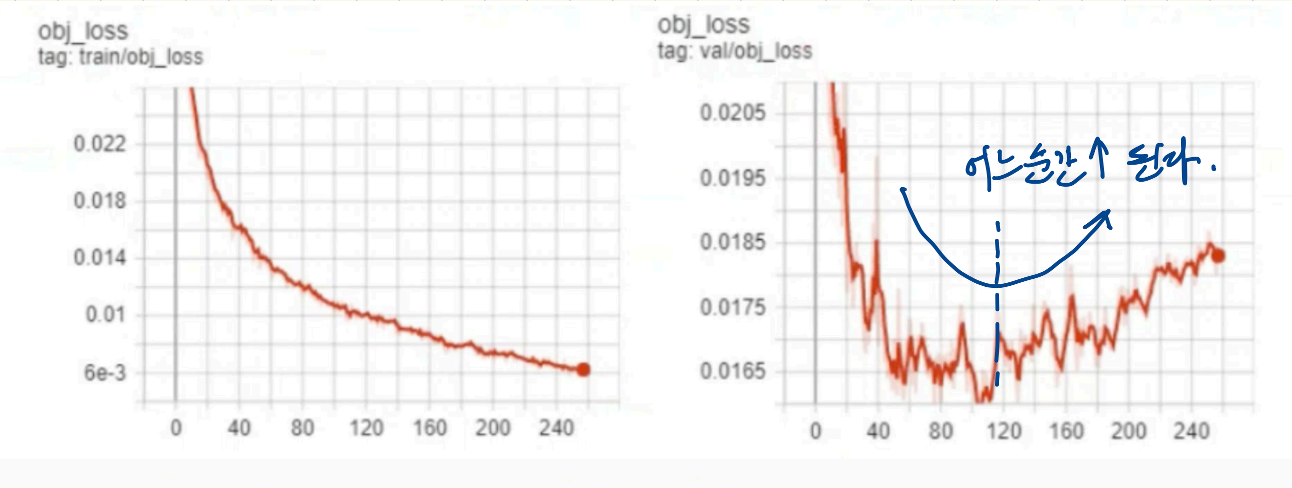
여기 Model들은 validation set을 기준으로 select한다.

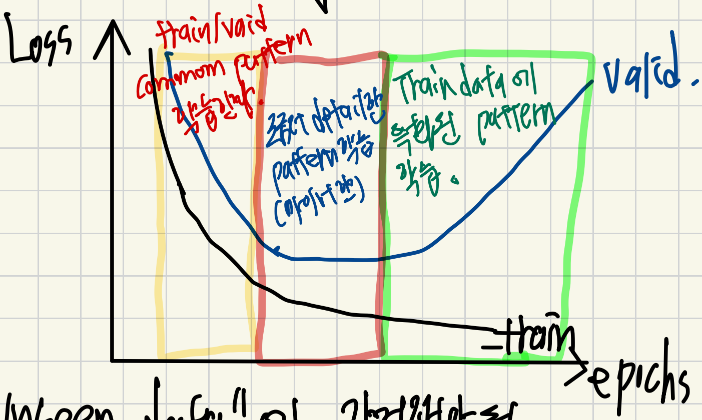
(Test는 절대 Never 사용 X)



(1) 반드시 training set은 줄어들어야한다 (Training이 ↑이면 문제이다)

(2) validation graph는 그려봐서 동향을 파악한다.

Training Loss는 계속해서 줄어들지만, 어느순간 Validation Loss가 증가하는 경우 "over fitting"이 발생했다고 한다.



Model 학습시 "unseen data"에 강건해야됨.

초기에는 train & valid의 common pattern을 학습하지만

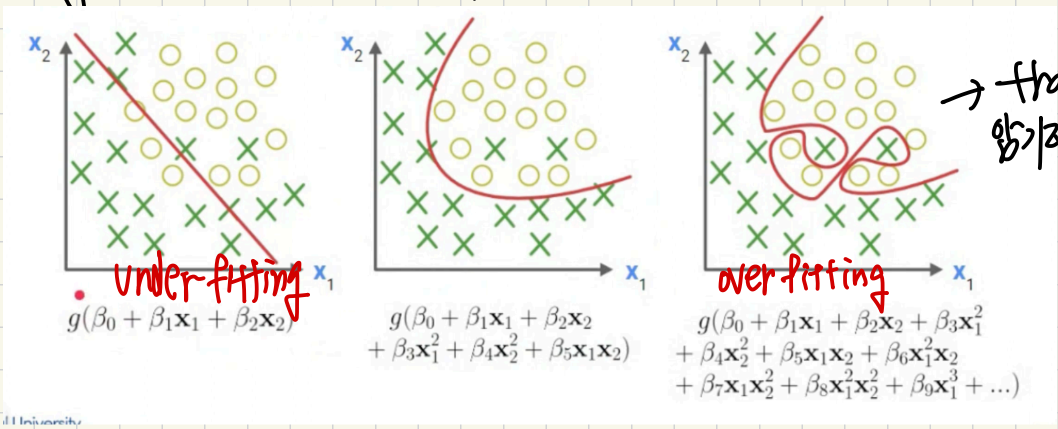
학습이 진행될수록 train data의 마이너한 pattern 까지 학습해서

valid data에 대해 Loss가 증가하게 된다. = over fitting

⇒ feature가 high & dimension이 커지는 경우

(이에 대응하는 data수가 필요하다)

overfitting은 Model의 capacity가 충분한 한계를 갖는다.



→ trainset을 아예 암기해버린다.

즉, 학습가능한 parameter의 수가 model의 capacity이다. (model complexity = complexity)

under fitting : Data pattern 자체를 학습할 용량 X

overfitting : Train Data에 너무 fit하게 학습

↳ "적당하게 Model 크기를 정하고, 적당하게 학습시키자"

〈Early stopping〉

validation Loss가 증가하는 경우 학습 종료!

⇒ 우리가 IID를 가정했기에

Train에서 Noise를 학습하는 time이 결국 다른 valid에서도 있을 것이다. (valid Loss가 올라가는 순간 ≈ Test Loss가 올라가는 순간 일 것이라)

∴ 일정주기로 vaidation Loss를 check해서 validation ↑이면 stop한다.

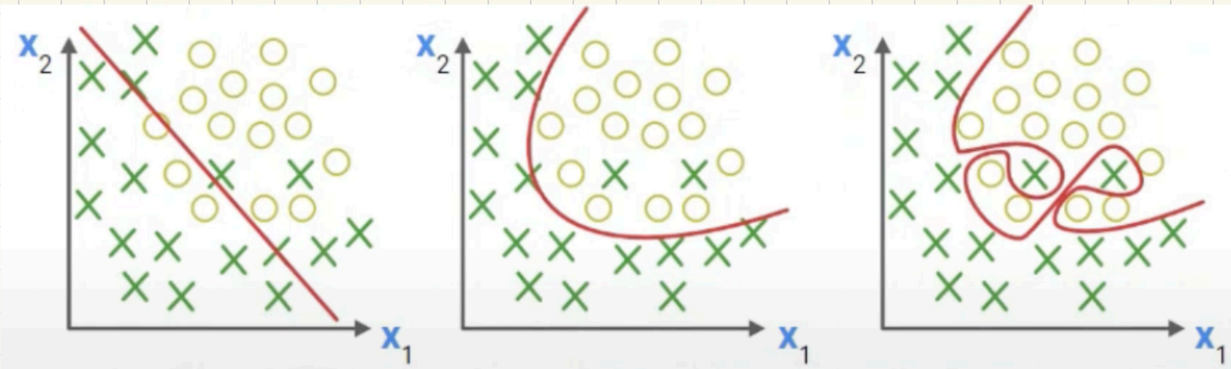
어떤 기준? ① train시 사용하는 loss function

② 실제 정확도 ---- 오히려 유리하다 (overfitting 판단에 유리)

(실제 관심있는 metric)

Metric 마다 overfitting되는 순간이 달라서, 최종으로 사용하는 metric으로 valid의 overfitting 여부를 판단하는 게 일반적이다.

⟨ Regularization ⟩



(1) $y = \beta_0 + \beta_1 x_1$ (2) $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2$ (3) $y = \beta_0 + \beta_1 x_2 + \cdots + \beta_n x_n$

위 Model 보면 (3) 모델에서 $\beta_3 x_3 \sim \beta_n x_n$ 까지가 0이 되면 (2)가 된다.

즉, $y = \beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n$ 관계를 갖게 된다.

$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2$

$y = \beta_0 + \beta_1 X_1$

∴ 가장 큰 Model을 만들고 β들을 조절해서 Complexity를 조절 가능할 수 있다. ≈ 규제항 (정말 Need한 β가 아니면 0으로 만들자!!)

Shrinkage Methods : coefficient(β)가 0에 가깝게 만들어주는 방법.

$$RSS = \sum_{i=1}^{n} (y - \beta^T x)^2 + \lambda \|\beta\|_2^2$$

↳ 양의 상수 (규제정도를 의미) λ↑이면 거의 β는 0이야.

β들이 너무 큰 값을 갖도록 하고 싶지 않다.
β가 커지면 e^2가 줄어들면 β를 키우고
e^2의 줄어든 크기 < $\|\beta\|_2^2$ 이면 그냥 β를 줄여라!!

β에 대한 Loss와
β자체의 크기 사이의 밀당주기.

★ λ↑↑일수록 β들이 0에 가까워져
model들이 simple해진다.
★ (variance ↓, bias ↑)
모두 비슷함 예측과 다름.

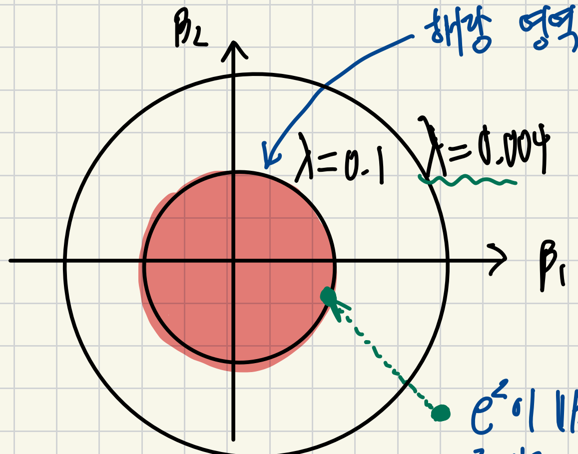
Regularization에서 data들의 기본 단위가 다르면 단위가 작은 data는 penalty로 부터 자유롭다.
이는 Model이 제대로 학습X 이 된다.

" 그래서 data들은 $\tilde{x}_{ij} = \dfrac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij}-\bar{x}_j)^2}}$ 로 정규화하여 사용한다"

Linear Regression 풀면 $\beta = (X^TX + \lambda I)^{-1}X^Ty$ 가 된다.

<Ridge (L2 - regularization)>

해당 영역 밖은 $\|\beta\|_2^2 < e^2$ 이다.



(λ↑이면 β가 더 작은 원을 그린다)

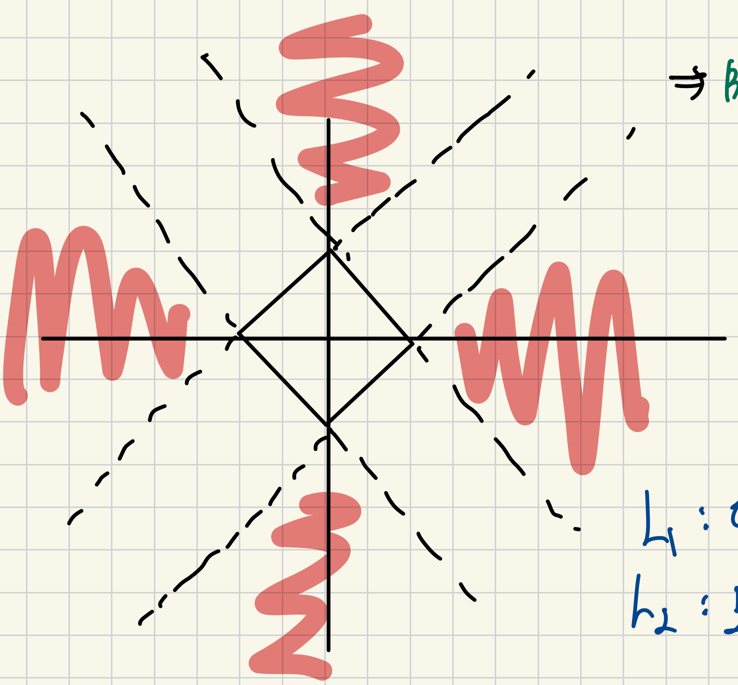
$\lambda=\infty$이면 영역X
$\lambda=0$ 이면 모든 $\beta=0$ 이 된다.

e^2이 $\|\beta\|_2^2$보다 크기에 규제로 인해 $\|\beta\|_2^2$로 이동.

<Lasso (L1 - Regularization)>

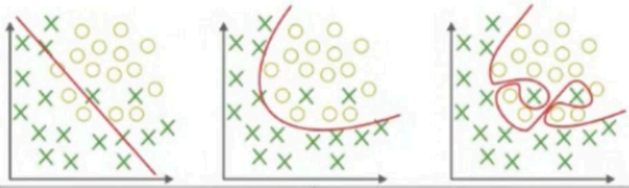
penalty term이 $\|\beta\|_1$이 된다. L1 Loss는 더욱 sparse한 model을 유도 ⇒ β를 0으로 만들어준다.
L2는 0에 가까워지지만 0이 되진 않는다.

⇒ β나 β가 몇개 살아남을지 알수X (λ로 실험적으로 찾는다)



: 해당 영역들과 가장 가까운 point가 꼭짓점이다. 즉, 특성 $\beta=0$을 만족하는 위치이다.
($\beta=0$인 영역 > $\beta\neq0$인 영역)

L1 : 덜 need한 β들부터 0이 됨
L2 : 모든 β가 동일한 속도로 0으로 수렴한다.



# of parameters	Small	Large
Model rigidity	Rigid	Flexible
Model complexity	Simple	Complex
Decision boundary	Smooth	Irregular
Regularization	High λ	Low λ
Extreme problem	Underfitting	Overfitting
Bias	High	Low
Variance	Low	High