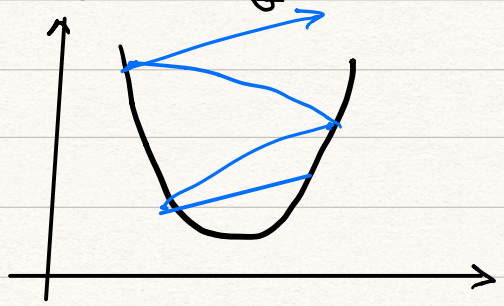
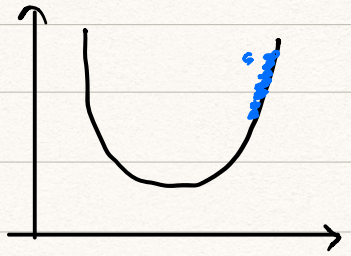
1-1. Large learning rate : overshooting

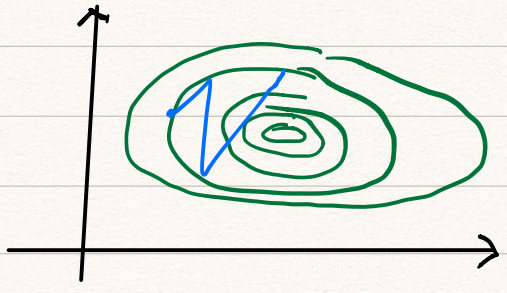


1-2. Small learning rate : ~~takes too long~~, stops at local minimum



=> Try several learning rates

2. Data (X) preprocessing for gradient descent



x_1	x_2	y
1	9000	A
2	-7000	A
4	-2000	B
6	8000	B
9	9000	C

데이터 값의 큰 차이가 있을경우엔

'normalize' 해줄 필요가 있음.

많이 쓰는 방법은 { zero-centered data 나
normalized data 도!.

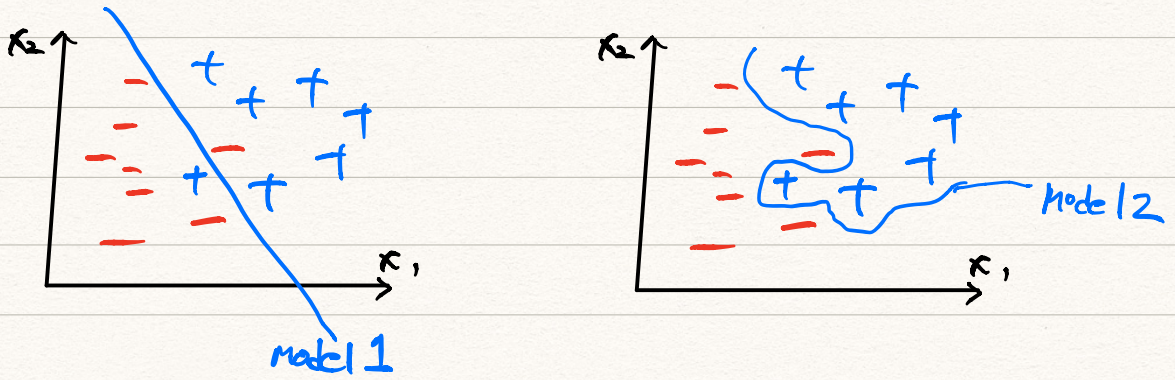
⊛ Standardization

$$x'_j = \frac{x_j - \mu_j}{\sigma_j}$$] 처리.

```
x_std[:,0] = (x[:,0] - x[:,0].mean()) / X[:,0].std()
```

⊛ Overfitting

: 테스트에선 잘 맞으나, 실제에선 잘 안 맞는 모델.



⇒ model 2는 저 상황에선 더 잘 맞지만, 실제 일반적인 상황에서는 잘 안 맞는 모델.

= overfitting.

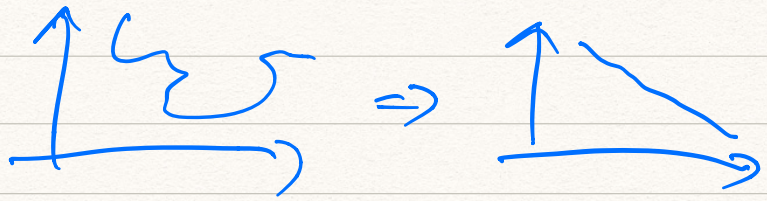
↳ training data가 많을수록 줄일 수 있어

{ features 수를 줄이자!
Regularization ★

⊛ Regularization

=> 너무 특정 상황에만 맞게 훈련하지 말자.

- " 구부러지 말고 펴자.



$$L = \frac{1}{N}\sum_i D(S(Wx_i + b), L_i) + \lambda \sum W^2$$

regularization strength

```
l2reg = 0.001 * tf.reduce_sum(tf.square(W))
```

상수