
Discussion CAEsaaaaaar 

loss → MSE

$$\frac{1}{m} \cdot \sum_i (\text{pred}_i - \text{observed}_i)^2$$

$$x^n \to n(x)^{n-1}$$

$f(x) = ax + b$

$$\frac{1}{m} \sum_i^m (\text{pred}_i - \text{observed}_i)^2$$

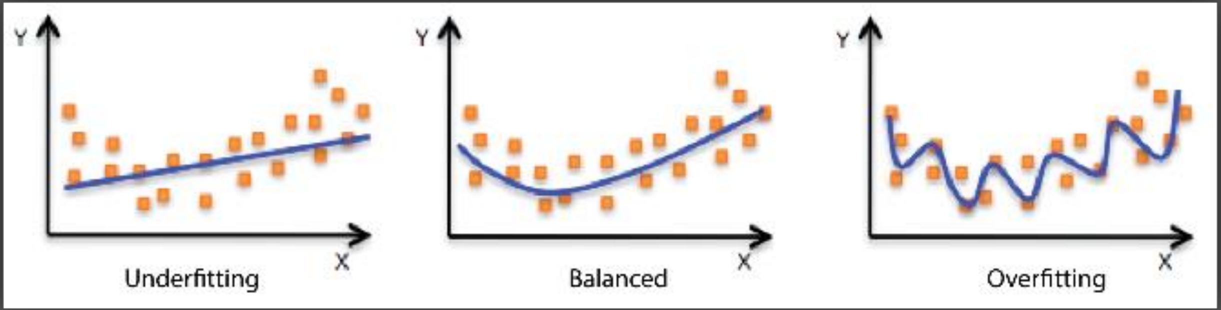
derivative w.r.t. a $\therefore = 2 \cdot \frac{1}{m} \sum_i^m (\text{pred}_i - \text{observed}_i)\, x$

Gradient Descent:

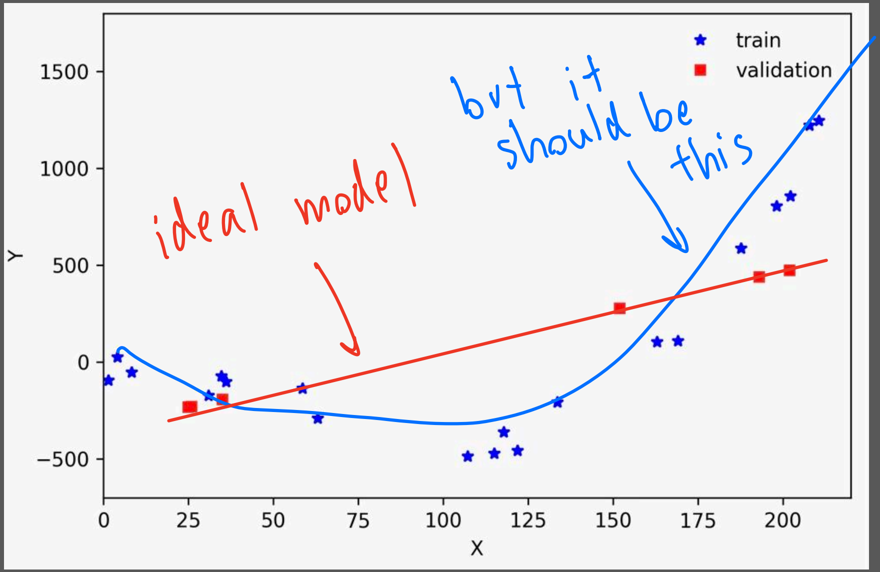
$$\theta_{0(new)} = \theta_{0(old)} - \underbrace{\lambda}_{} \, \underbrace{\frac{\partial \text{Cost}}{\partial \theta_0}}_{}$$

In English: learning rate × slope of the cost function of parameter $\theta_0$

# Recap

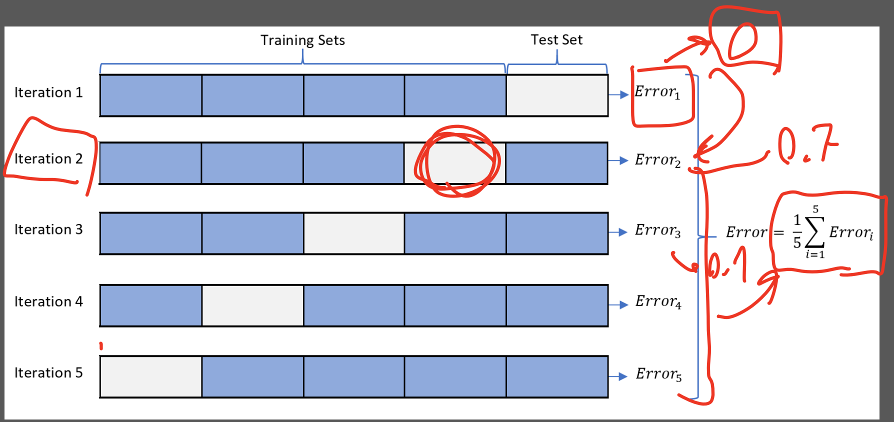


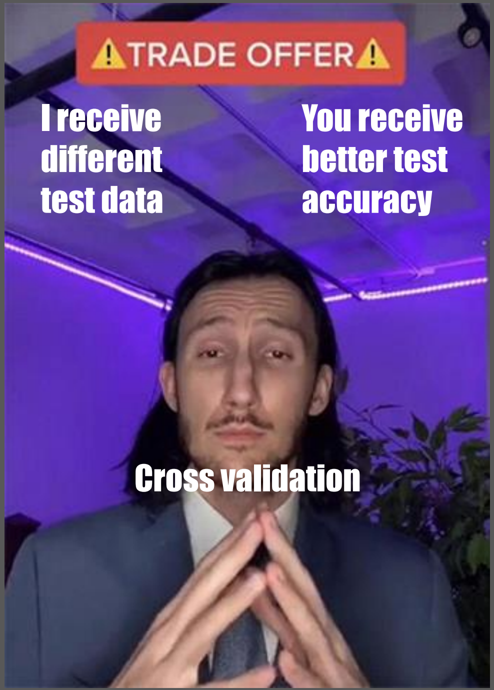
**Motivation**: we have validation dataset to measure how well the model is. But what if the validation dataset is poorly-chosen?



# [Cross Validation]

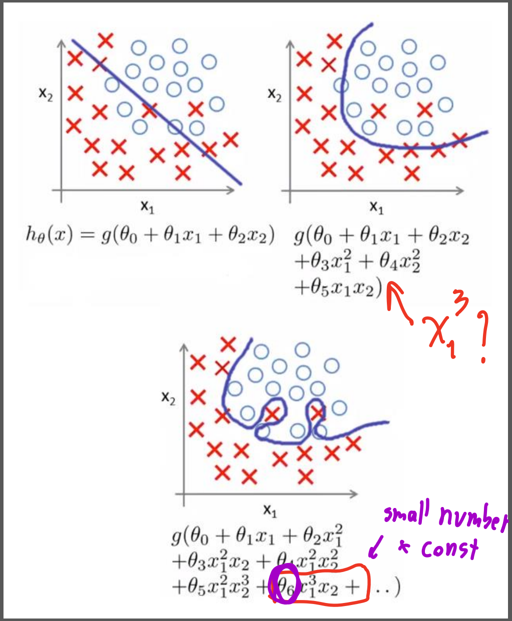
**Solution**: repeat the trials, change validation dataset, average accuracy across all trials



**Recap**: We can make the model more complex to capture non-linear data

**Problem**: What is the right degree complexity?



**Solution:** Re  rization

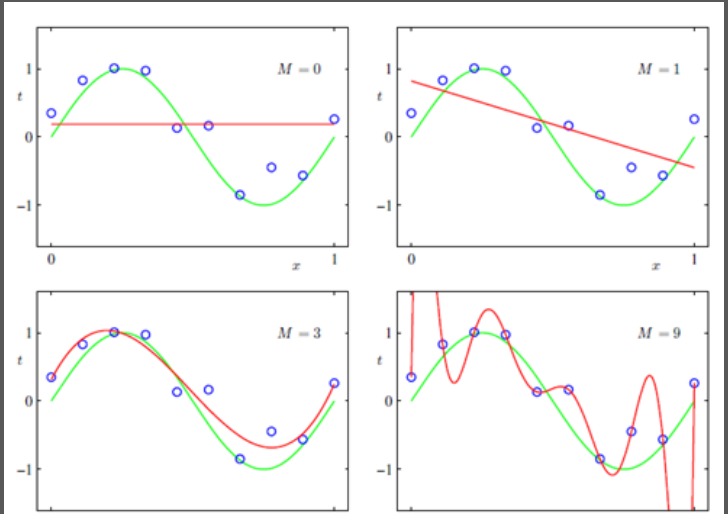
gura

# Regularization

"i" is your $i^{th}$ data from the dataset

$$y = \beta_0 + \beta_1 x$$

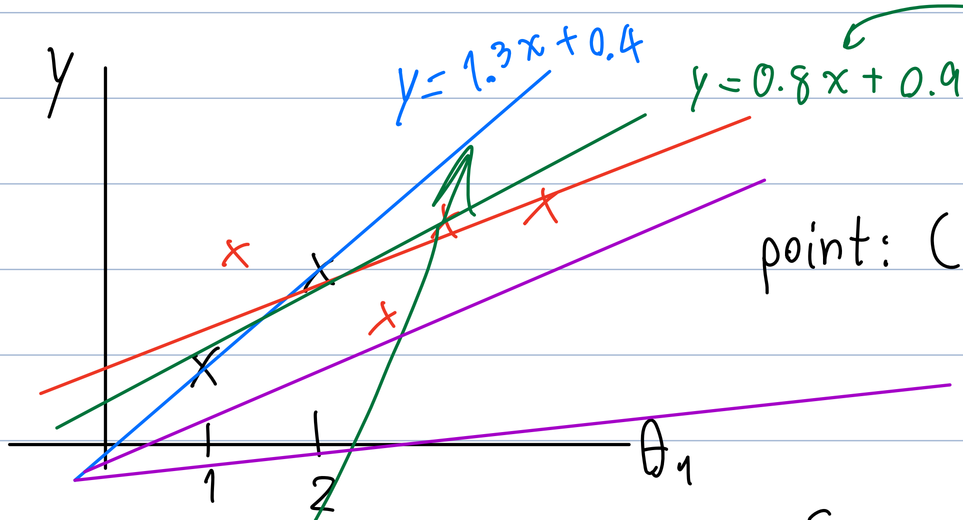
$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$$




$$y = \beta_0 + \beta_1 x + \dots + \beta_9 x^9$$

Reminder : $\mathrm{Cost}(\beta) = \frac{1}{m} \sum_{i=1}^{m} \left[ \mathrm{predicted}^{(i)} - \mathrm{observed}^{(i)} \right]^2$

Ex




point: $(1, \boxed{1.7})$, $(2, 3)$

Modify $\text{Cost}(\theta_1)$ to be $= \frac{1}{m} \sum_i \left[\text{predicted}^{(i)} - \text{observed}^{(i)}\right]^2$

technically parameter

$+ \underbrace{\lambda \times (\text{slope})^2}$ ↳ penalty term

$\lambda = 1$ ~~100~~ 100,000

Ex Blue: Old cost $+ \lambda \times \text{slope}^2$

$= 0 + 1 \times (1.3)^2 = 1.69$

Green: $\frac{1}{2} \times \left[(1.7 - 1.7)^2 + (2.5 - 3)^2\right]$

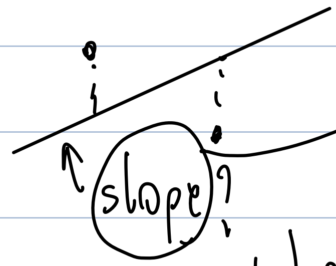
$+ (1)(0.8)^2 = 0.765$

we can see green's cost is less than blue

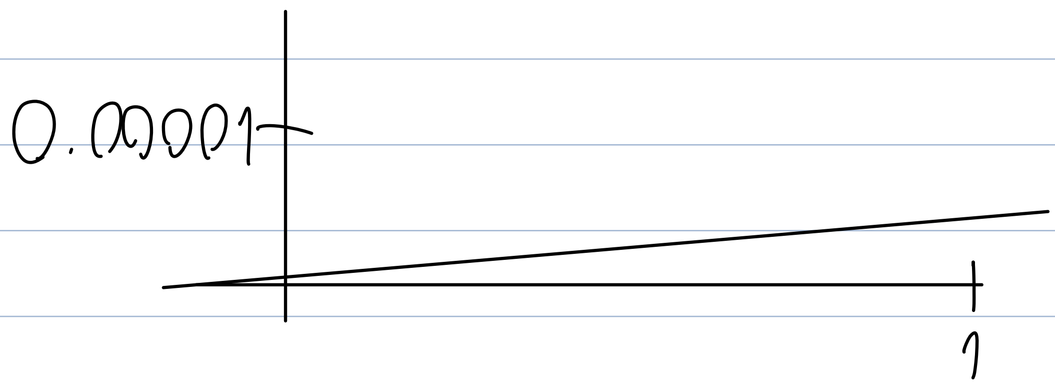
Goal: we want to minimize the cost value

$$\min \left[ \frac{1}{m} \sum_i \left[ \text{predicted}^{(i)} - \text{observed}^{(i)} \right]^2 + \lambda \, (\text{slope})^2 \right]$$

$10^6$      $1/10^6$




what's the value of slope to make cost() = 1 ?




Ridge regression : $Cost(\beta) = MSE + \text{penalty term}$

$$= \frac{1}{m} \sum_i (\text{pred}^{(i)} - \text{obs}^{(i)})^2 + \lambda (\beta_0^2 + \beta_1^2 + \dots \beta_n^2)$$

$$= \boxed{\frac{1}{m} \sum_i (\text{pred}^{(i)} - \text{obs}^{(i)})^2 + \lambda \cdot \sum_i (\beta_i)^2}$$

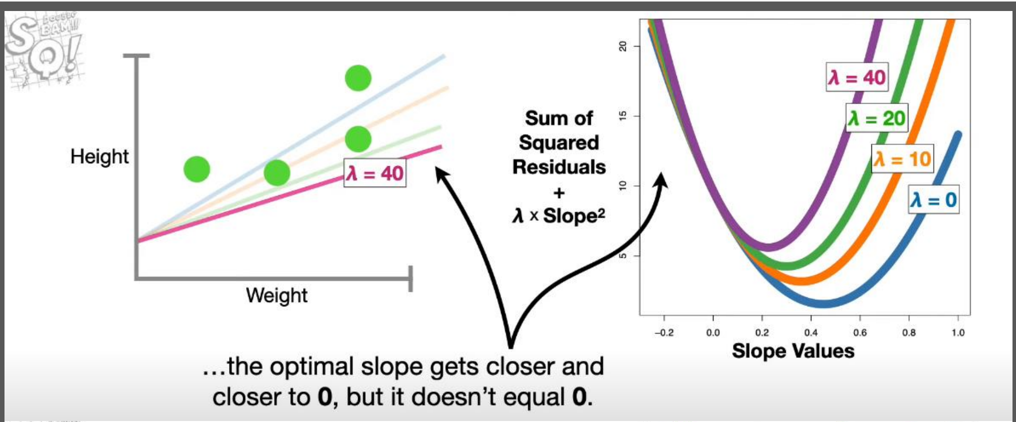
Key takeaway: We find a way to make the model underfits, so we can get higher testing accuracy even if training accuracy is low

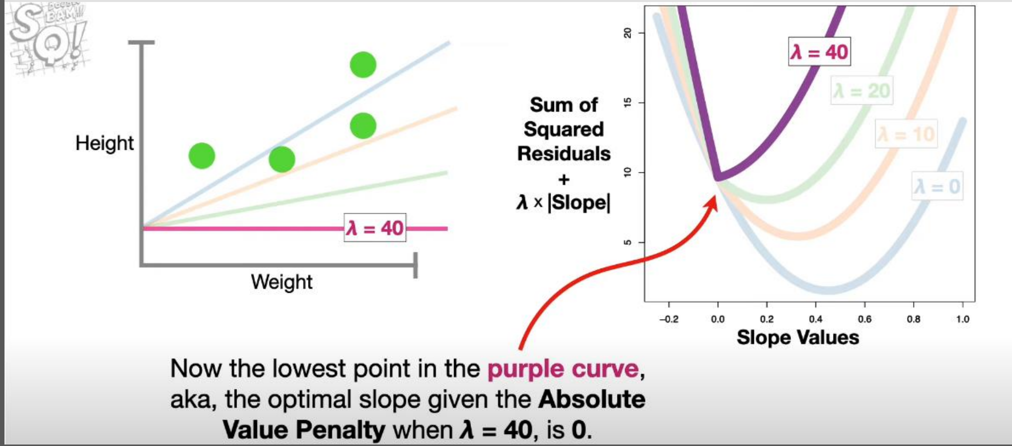
# Note: there is another type of regularization, which is called **Lasso**

The difference is that the penalty term, we use **absolute** instead of squaring the parameters

| Ridge | Lasso |
| --- | --- |
| Squared the parameters | Take absolute of the parameters |
| Parameters get close to zero | Parameters can reach zero |
| Better when we believe every parameters are useful | Can exclude useless parameters |

close to zero, but not




Statquest Youtube video: Ridge vs Lasso Regression, Visualized!!!

https://www.youtube.com/watch?v=Xm2C_gTAI8c

# But you know, I learned something today

Back at 2:25

Ex ridge reg:

$y = 5x_1 + 3x_2 + 0.008x_3$

lasso reg:

$y = 5x_1 + 3x_2 + 0(x_3)$

- We use cross validation to average models across all trials, instead of accidentally pick the invalid test data
- We use ridge/lasso regression to lower training accuracy, but get higher test accuracy