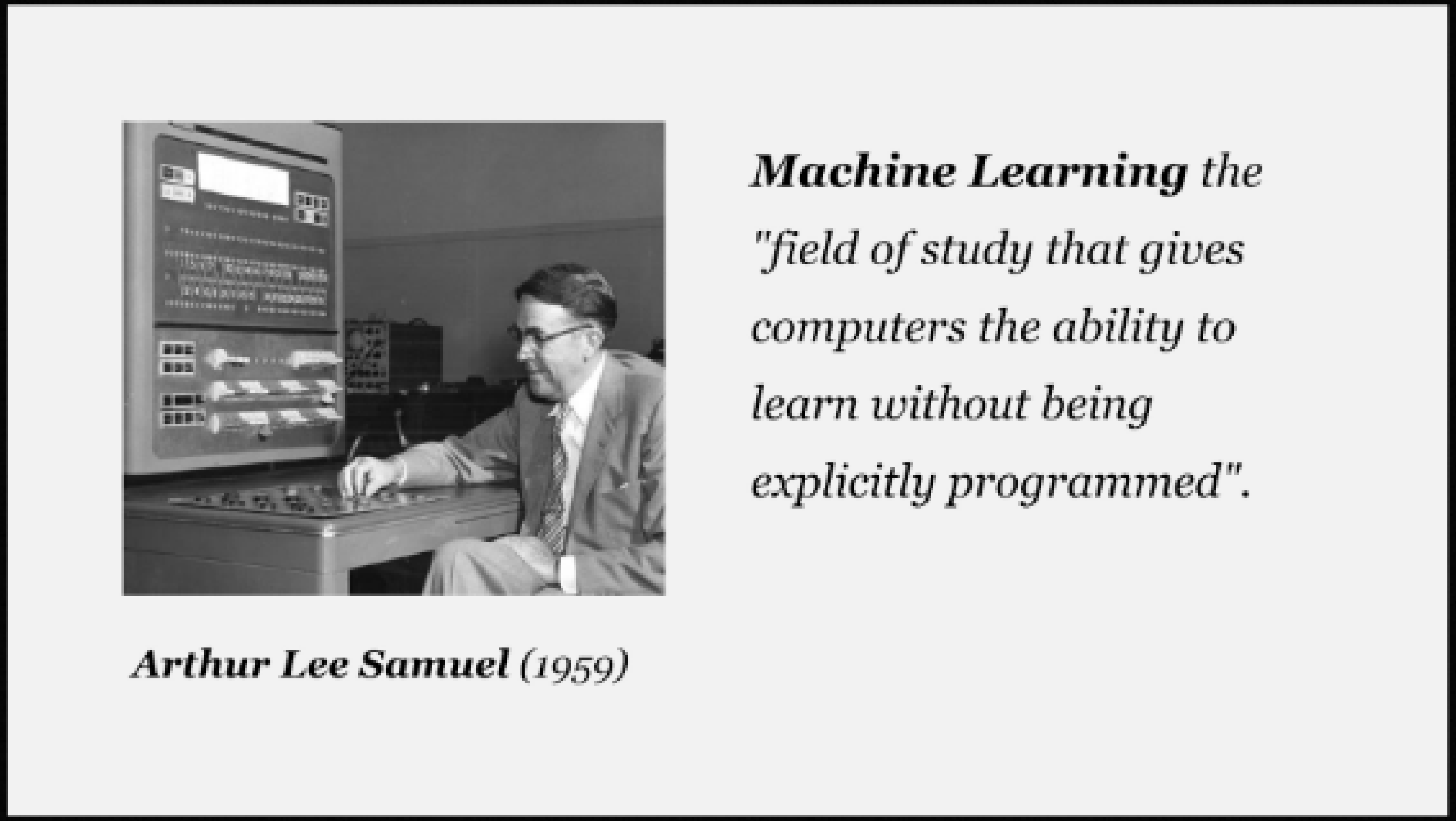
Computer → Code Execution

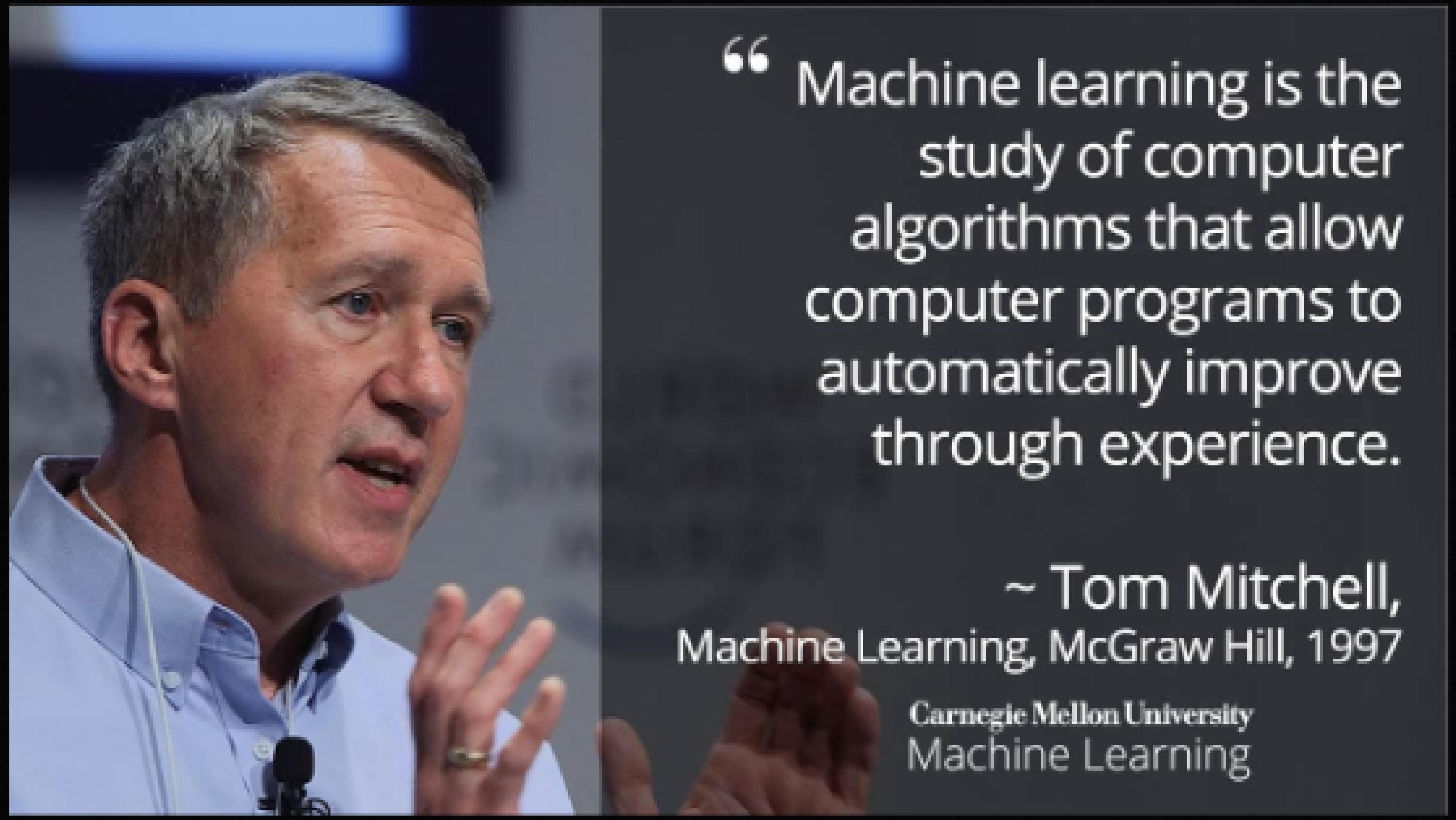
Machine Learning

Human → 1 write Algorithm 2 write Code

- Definition :
  - Learning from Data
  - Pattern Recognition
  - Learn by Experience



Experiment (E)
performance (P)
Task (T)

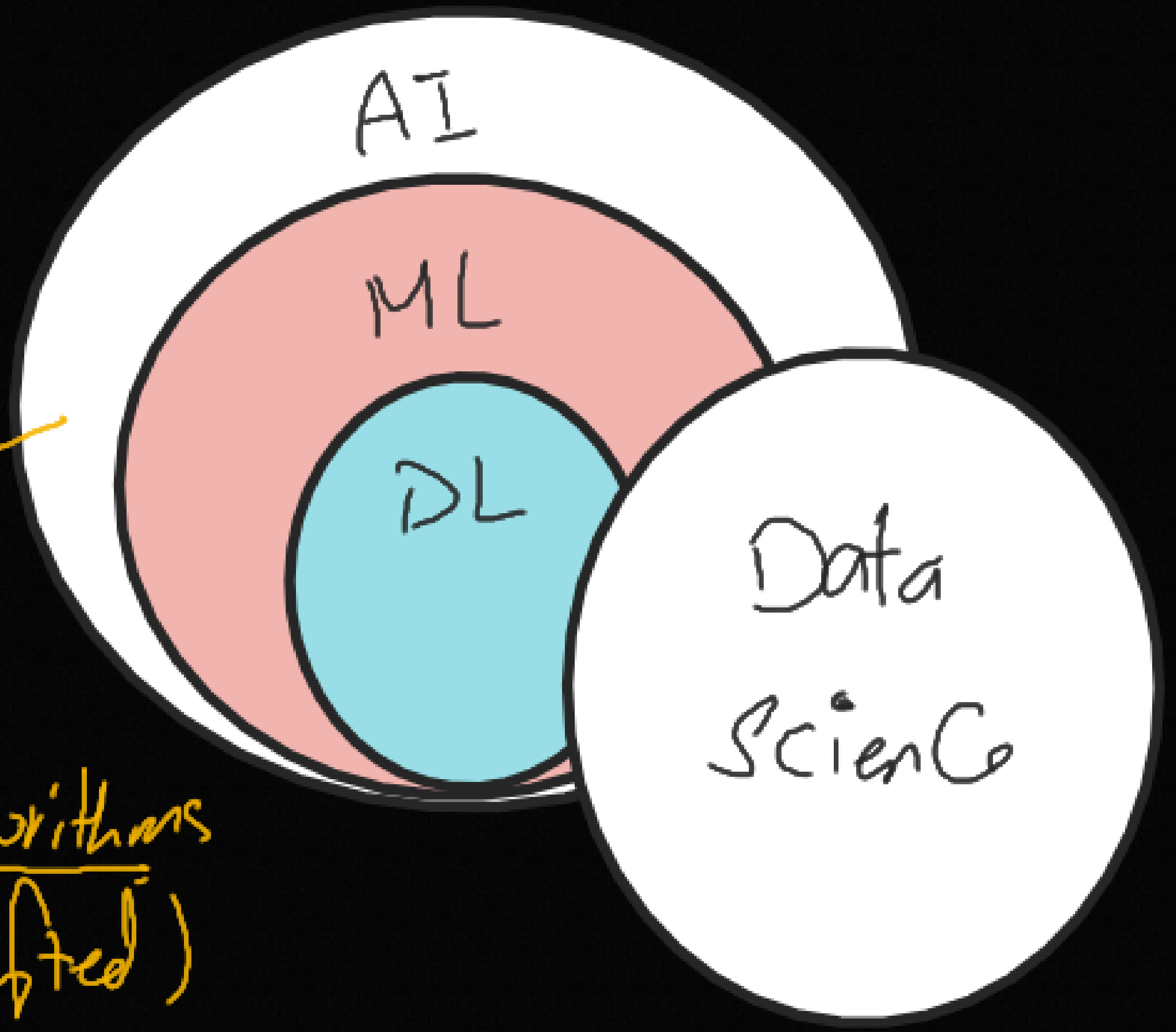


* AI : Artificial Intelligence

Simulation / Mimic of Human Intelligence

* ML → by learning from Data

* DL → learning through Artificial Neural Networks



Smart Algorithms
(Hard Crafted)
(Hard Coded)
(Traditionally)

Types of ML

Supervised

Labeled Data

Features (Attributes) — Target (Label)

X — y

Mapping function

$$y = f(x)$$

Unsupervised

Unlabeled Data

Features

X

- find pattern
- find Similarity
- Anomaly Detection

Reinforcement

Agent — Environment

Action (Policy)

Exploration (Record)

Exploitation

Feedback
- +ve (reward)
- -ve (Penalty)

Supervised ML

X, y

Numerical

* price
* Sales
* Temp
* Consumption

Regression

Categorical

* Emails clf
* purchase / Not
* Handwritten Numbers Recog [0 → 9]
* Sentiment Analysis

Classification

Unsupervised

Clustering

Grouping acc. to Similarity

Dimension Reduction

1000 features

⇓

30 features

....

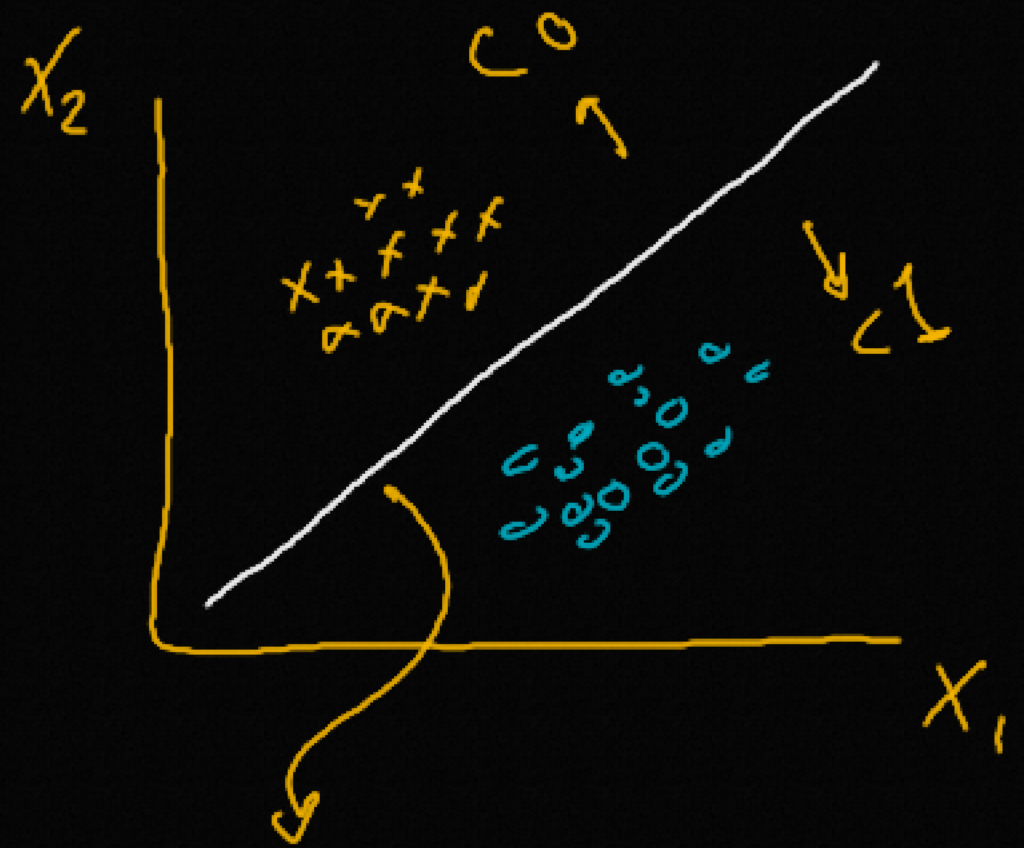
* Regression

Best fit line

$$y = w_0 + w_1 X$$

* Classification

labeled

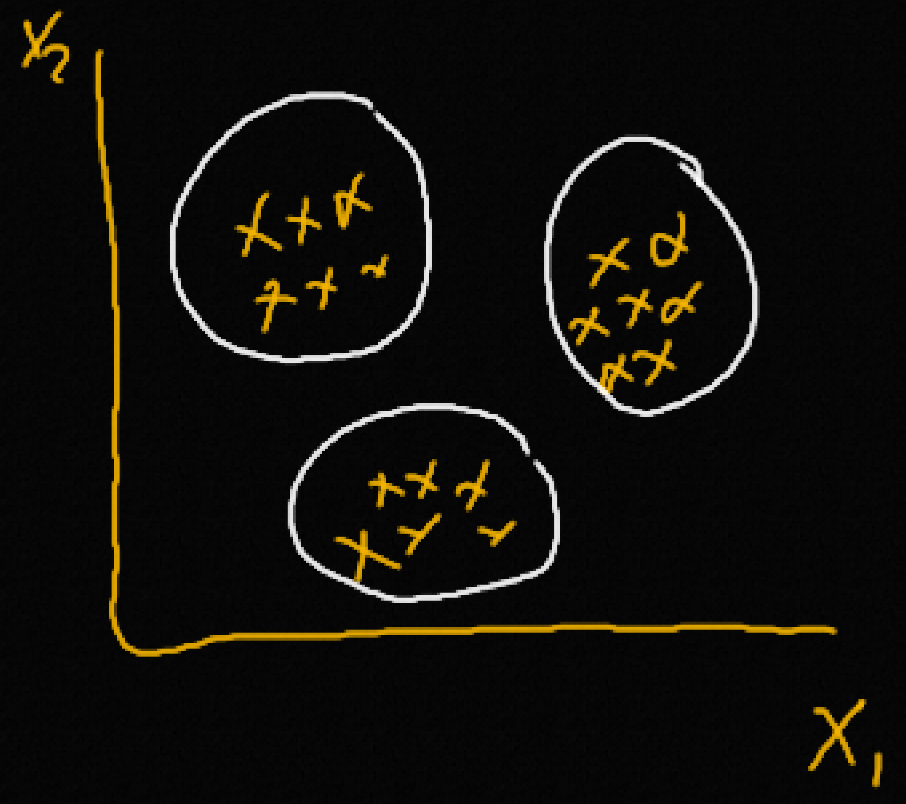


Decision Boundary

$$w_0 + w_1 X = 0$$

* Clustering

unlabeled



→ Grouping based on Similarity

Linear Regression

* Simple Linear Regression

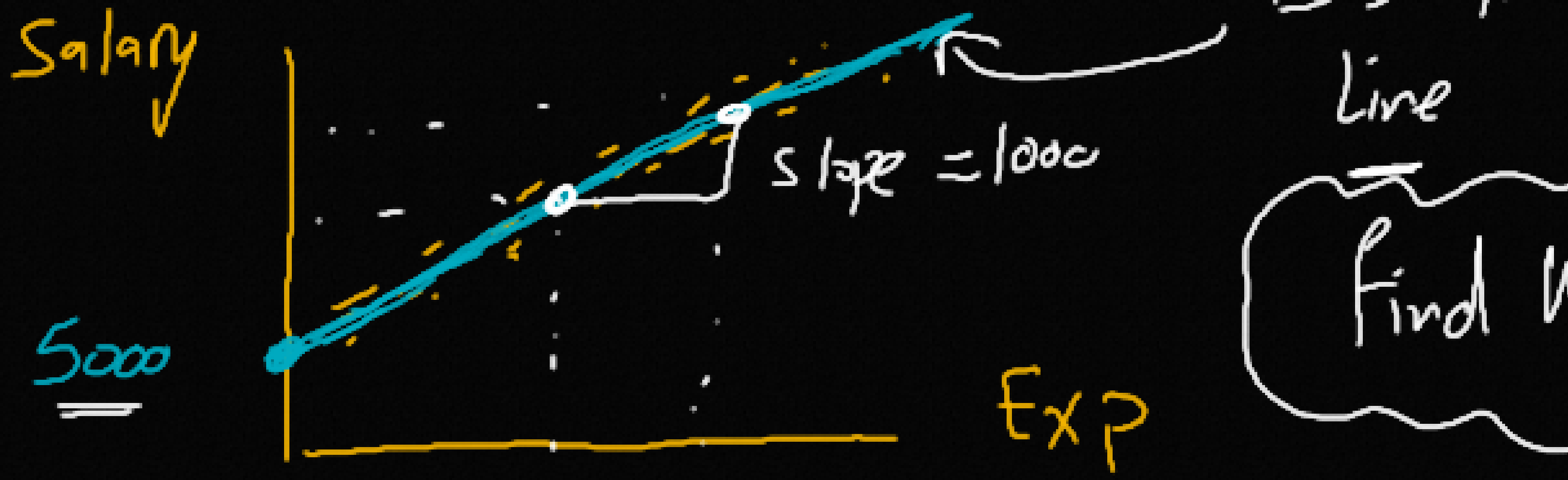
$$y = w_0 + w_1 X$$

y → Salary, w_0 → Start, w_1 → weight, X → # Experience

$$w_0 = \text{Intercept}$$

$$w_1 = \text{coef (slope)} = \frac{y_2 - y_1}{x_2 - x_1}$$

$$\text{Salary} = \underline{5000} + \underline{1000} * \text{YearsExp}$$

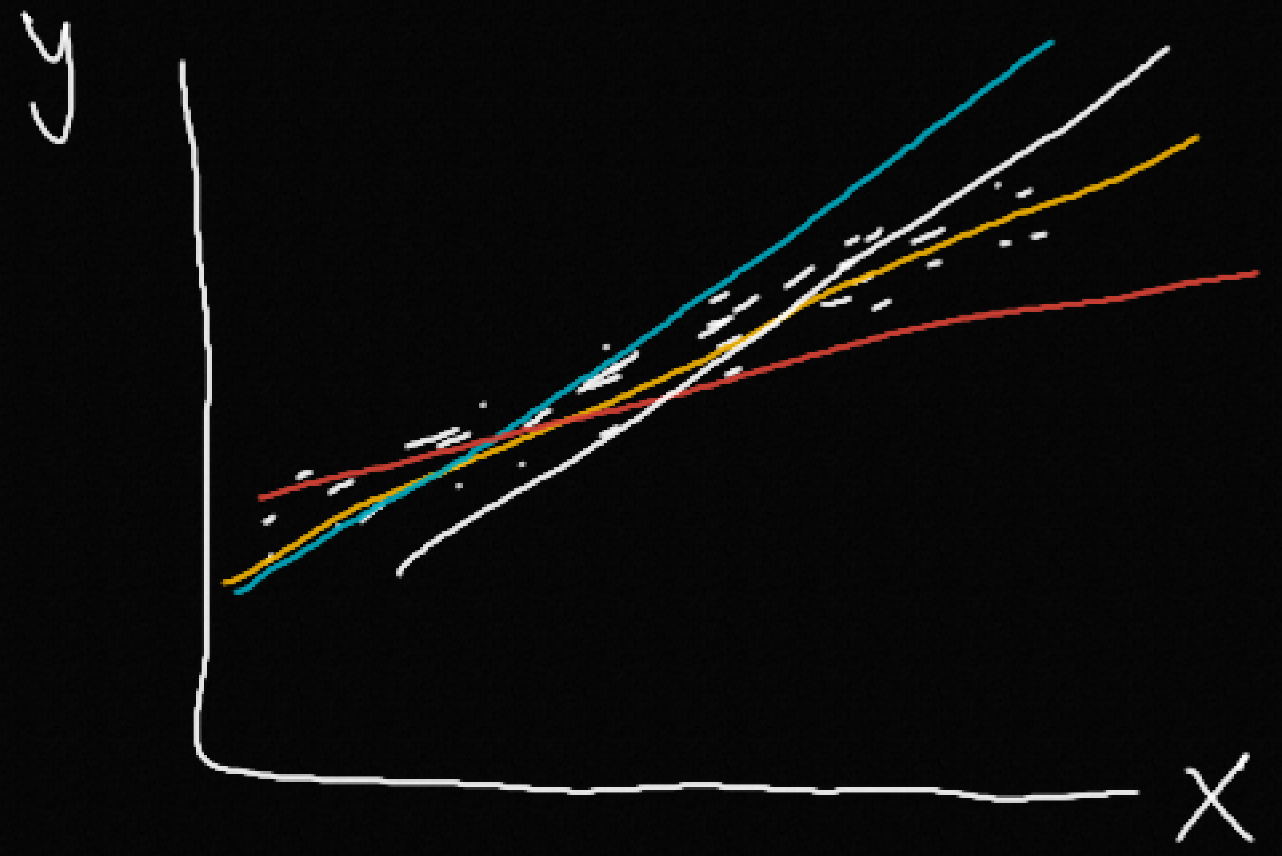


Exp	Salary
.	.
.	.
.	.
.	.
.	.

$w_0 = 5000$

$w_1 = 1000$

Find Weights ← ML Role



Best fit line?

Min Error → Error Metric?

✓ (1) Define Cost Function

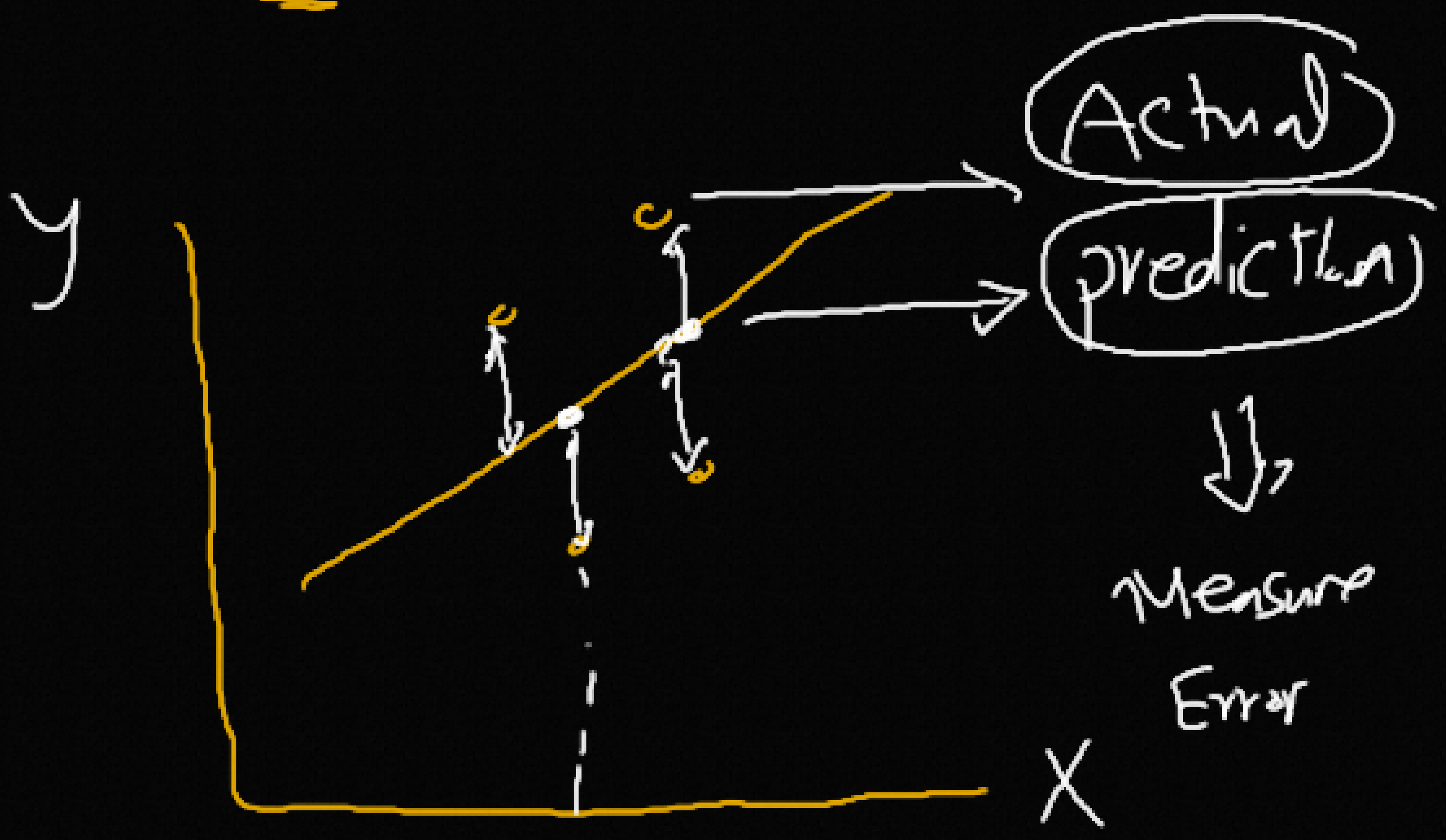
✓ (2) Find weights for Min. Cost function

$w_0 = 0$

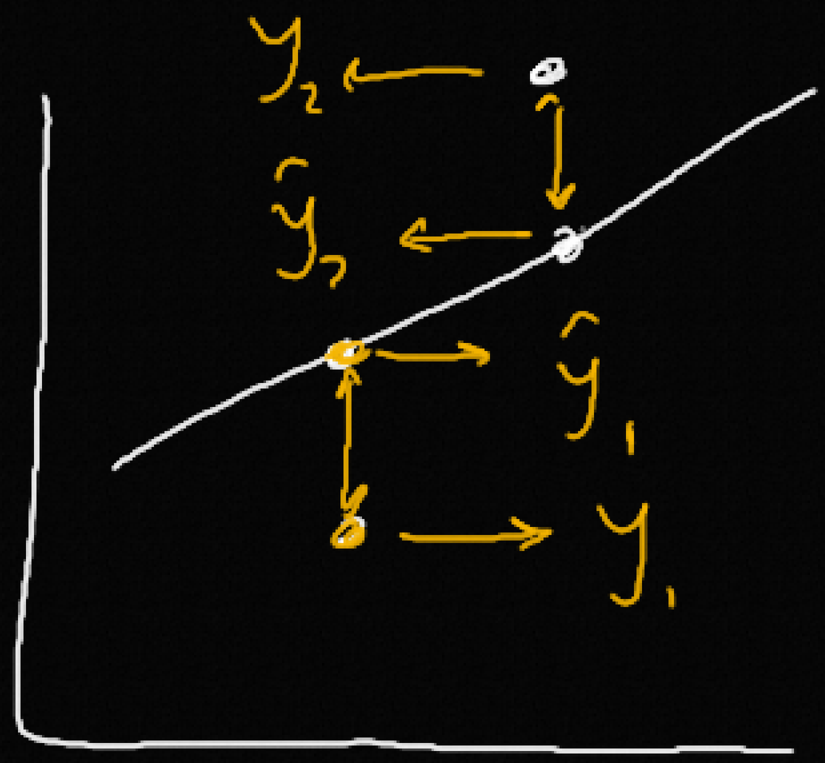
w_1	Cost fn (J)
$\boxed{5000}$	$\boxed{400}$
2000	500
7000	600
10000	700
⋮	⋮

Best weight ← 5000

400 → Min Cost fn



m-points



$$J = \frac{1}{m}\sum_{i=1}^{m}\left(y_i - \hat{y}_i\right)^2$$

→ Mean Squared Error

MSE

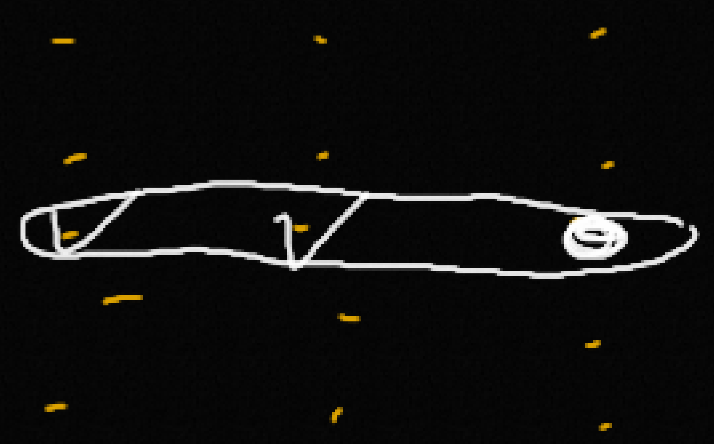
Cost Function

$$J(w) = \frac{1}{m}\sum_{i=1}^{m}\left(y_i - (w_0 + w_1 x_i)\right)^2$$

Target $\boxed{\checkmark}$

Missing

feature $\boxed{\checkmark}$



* Gradient Descent (optimizer)

1. Initialize Random Weights
2. Calculate gradients → derivatives

   → direction for next weights update (+/-)

3. update weights

$$\boxed{W^{new} = W^{old} - \eta \frac{\partial J}{\partial W}}$$

→ learning Rate

Iterations till Convergence

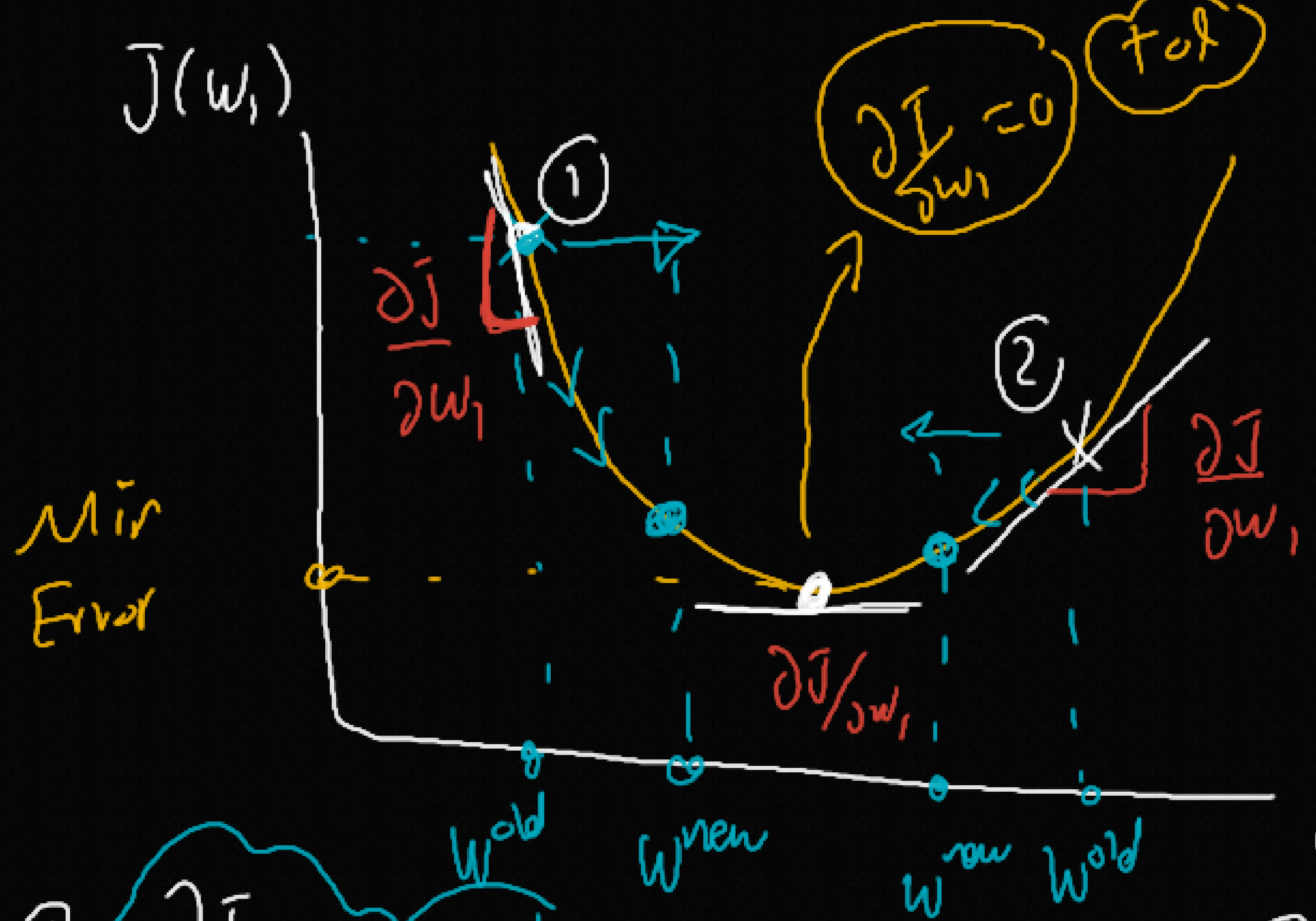
Convergence

$$W^{new} \approx W^{old}$$

$$\frac{\partial J}{\partial W} \approx 0$$

→ Best weights

$\boxed{W_0 = 0}$ $\boxed{\hat{y} = w_1 X}$ $\rightarrow$ $J(w_1) = \frac{1}{2m}\sum_{i=1}^{m}(y_i - w_1 x_i)^2$



Iterations

learning Rate

① $\frac{\partial J}{\partial w_1} = -ve$

$\rightarrow w_1^{new} = w_1^{old} - \eta \frac{\partial J}{\partial w_1}$

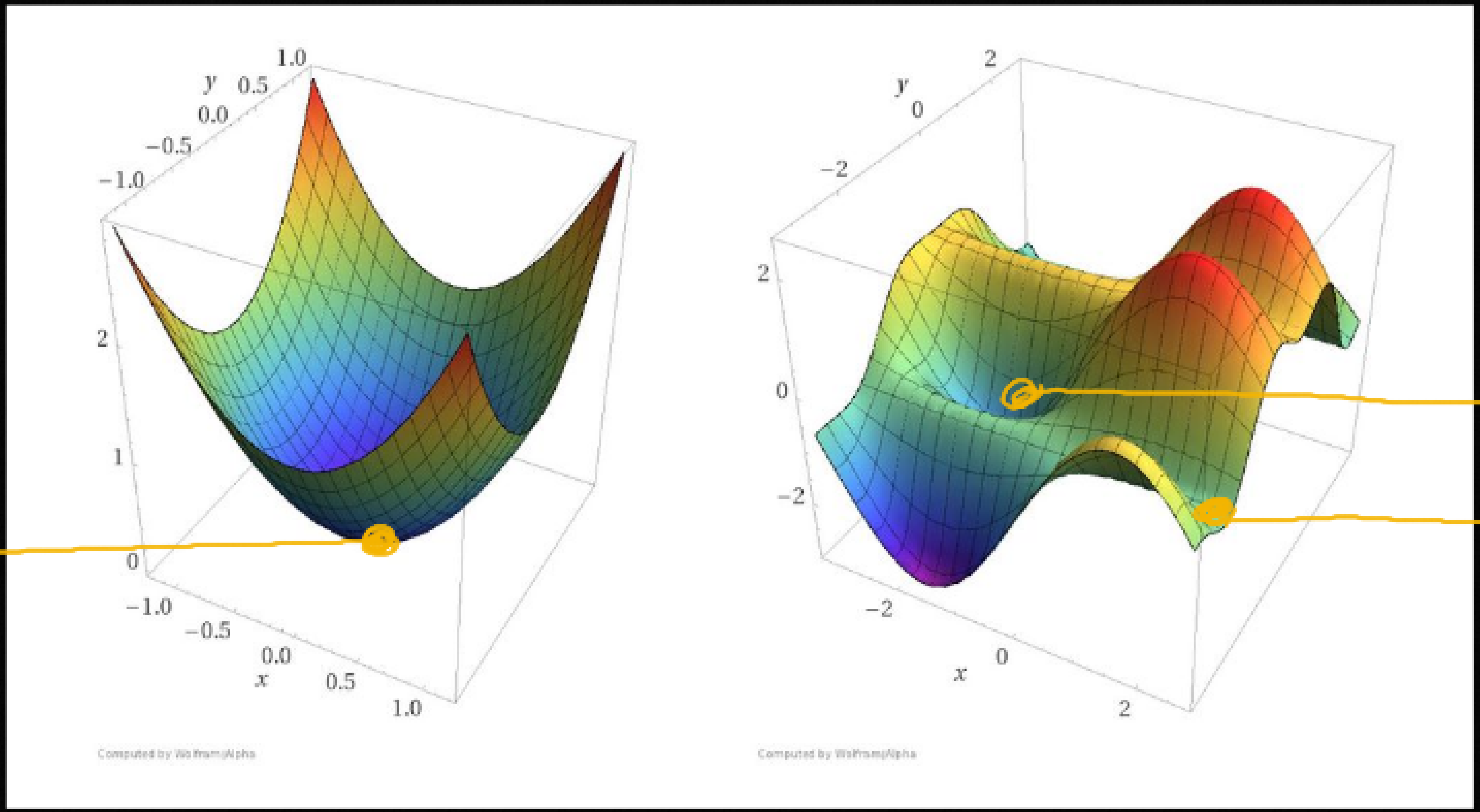
$w_1^{new} > w_1^{old}$

② $\frac{\partial J}{\partial w_1} = +ve$

$\rightarrow w_1^{new} = w_1^{old} - \eta \, \partial J / \partial w_1$

$w_1^{new} < w_1^{old}$

Min Error



Global Min

Local Min

① Initialize Random weights

② $\partial J / \partial w_0$, $\partial J / \partial w_1$ . . . . . . iters

③ update weights

Best fit line Solution

Iterative

Gradient Descent

OLS

Ordinary Least Squares

Closed Solution

$$\frac{\partial J}{\partial w} = 0$$

Eqns = # weights

https://are.berkeley.edu/courses/EEP118/current/derive_ols.pdf

* Multiple Features

$$\hat{y} = w_0 + w_1 X_1 + w_2 X_2 + \cdots + w_n X_n$$

$$\boxed{\hat{y} = w^T X}$$ vectorized

X (Matrix of features)

y (vector of labels)

$\Big\} \rightarrow$ W (vector of weights)

$$\begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & \cdots & x_n \end{array} \qquad y$$

$$X \qquad\qquad y$$

$$f(X)$$

$$\hat{y} = W^T X$$

$$W = (X^T X)^{-1} X^T y$$

Complex computation

Gradient Descent

iterations

learning rate

Complexity ↓

$> 10^5$

Sklearn

SGDRegressor

Normal Equation (OLS)

No iterations

No learning rate

Complexity ↑

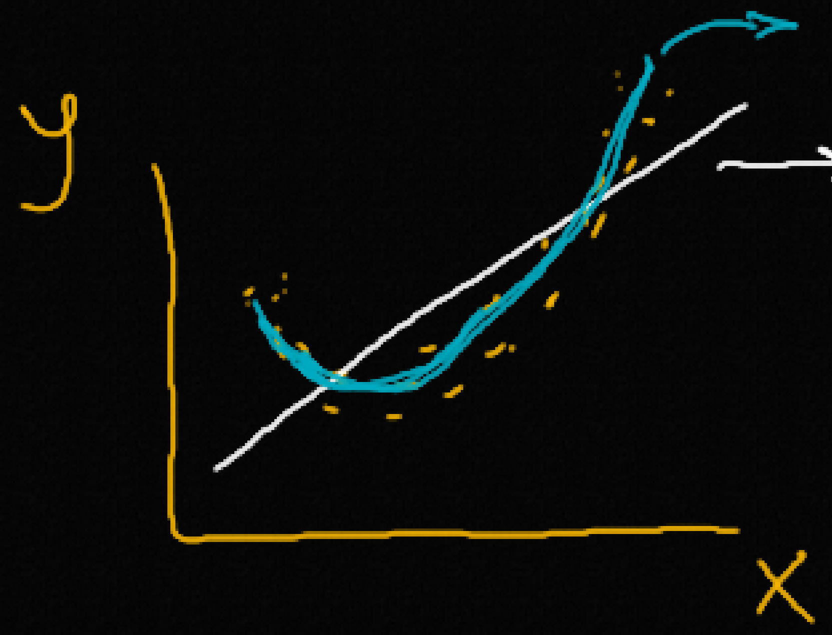
$< 10^5$

LinearRegression

$$X_{train},\ y_{train} \rightarrow \boxed{LR} \rightarrow \text{Weights}$$

Linear Regression

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + \dots + w_n x_n$$



Model 2 → Non Linear (Polynomial)

Model 1 → linear Model

$$\hat{y} = w_0 + w_1 x + w_2 x^2$$

Non linear

(Higher Degrees)

* Polynomial Regression

Polynomial features + Linear Regression

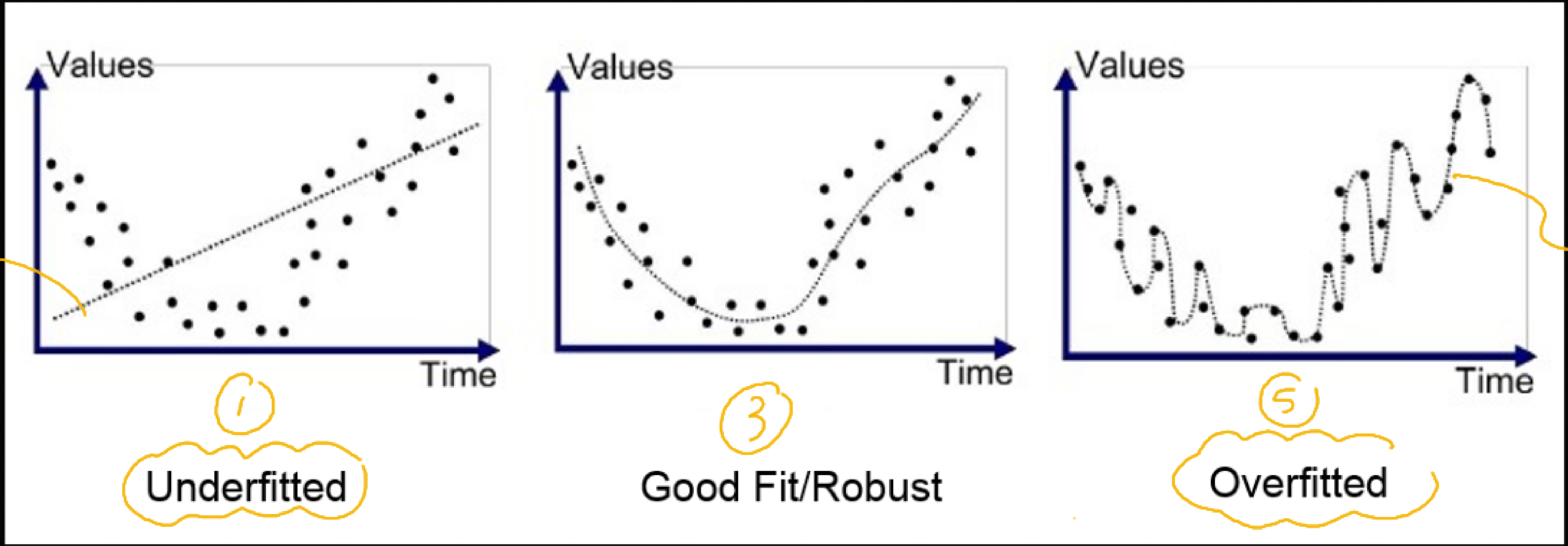
preprocessing

Exp   Salary

Exp $(Exp)^2$ Salary

Linear model $\rightarrow$ $w_0 + w_1 X_1 + w_2 X_2 = \hat{y}$

Biased Model



High variance Model

Polynomial Degree → 1 → underfitting (very simple Model)

→ 5 → overfitting (very Complex Model)

→ 3 → Good fit

Bias vs. Variance Tradeoff

Model Complexity ↓↓

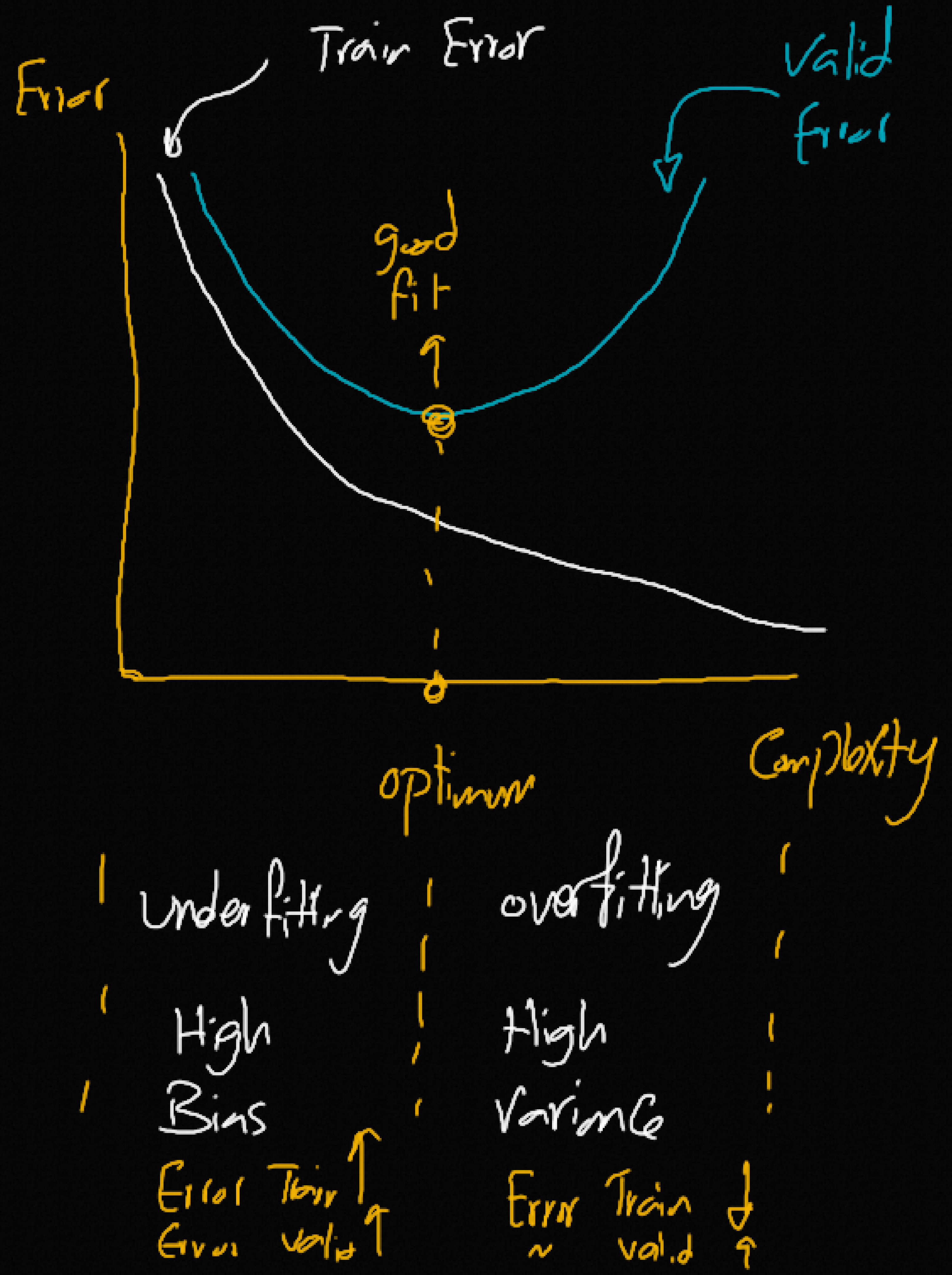
underfitting

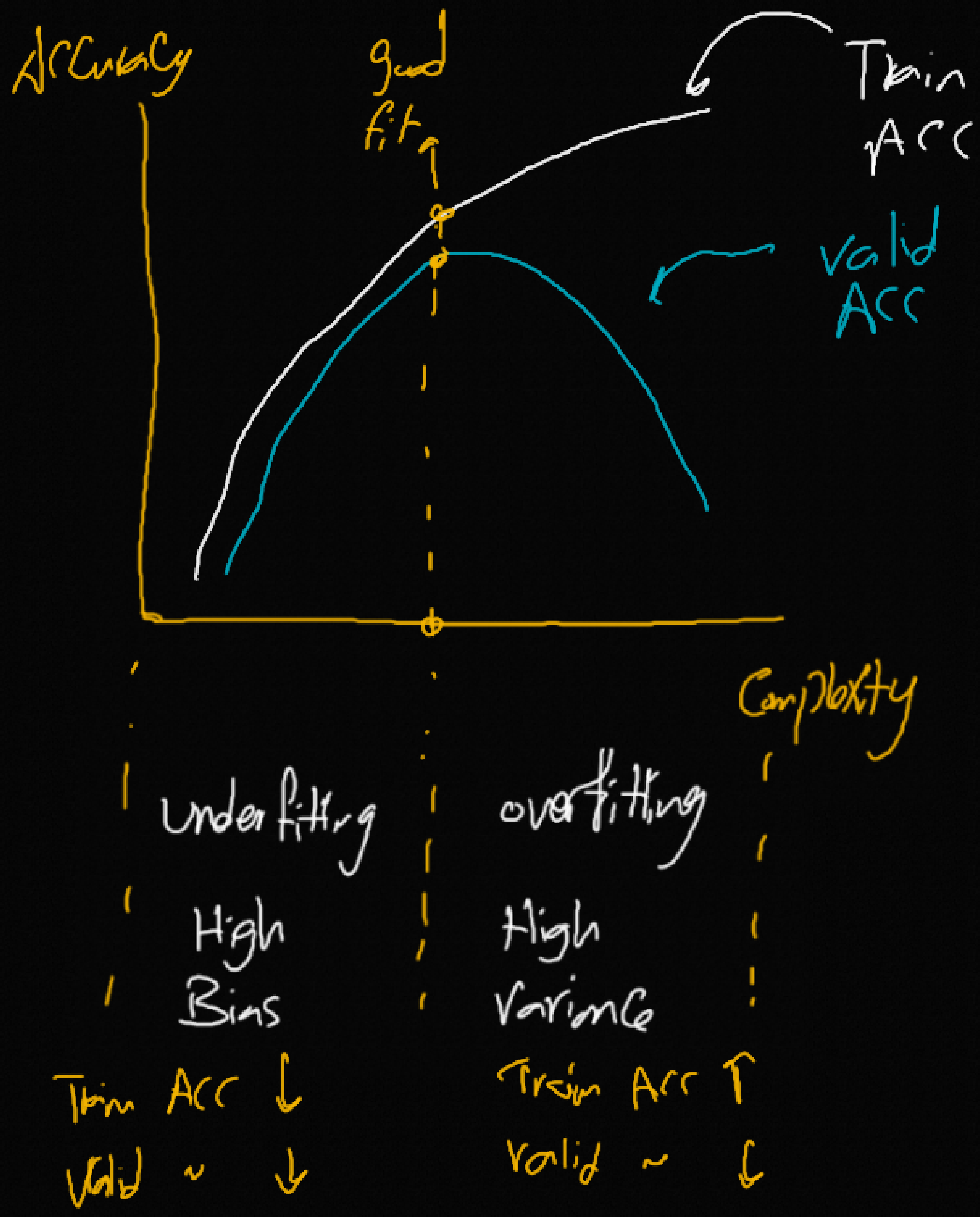
Complexity

Good fit

Model Complexity ↑↑

overfitting


Error
Train Error
Valid Error
good fit
optimum
Complexity
underfitting
High Bias
Error Train ↑
Error Valid ↑
overfitting
High Variance
Error Train ↓
~ Valid ↑


Accuracy
good fit
Train ACC
Valid ACC
Complexity
underfitting
High Bias
Train ACC ↓
Valid ~ ↓
overfitting
High Variance
Train ACC ↑
Valid ~ ↓