# Part I: Recap of Supervised Learning and Linear Regression Setup

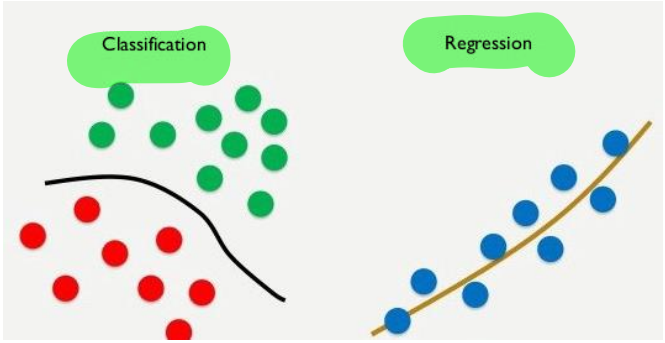
# Recap: Supervised Learning

- **Input:** $\left(x^{(1)}, y^{(1)}\right), \ldots, \left(x^{(M)}, y^{(M)}\right)$ ← Training Data
  - $x^{(m)}$ is the $m^{th}$ data item and $y^{(m)}$ is the $m^{th}$ **label**
    Feature vector — Label
- **Goal:** find a function $f$ such that $f\left(x^{(m)}\right)$ is a "good approximation" to $y^{(m)}$
  - Depends on loss fn
  - Can use it to predict $y$ values for previously unseen $x$ values

$y^m \in [0, K-1]$

$y^m \in \mathbb{R}$

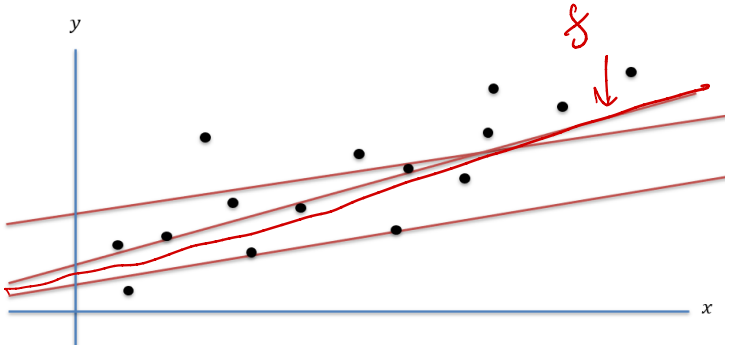
$\Rightarrow \min \sum L(f(x^m), y^m)$

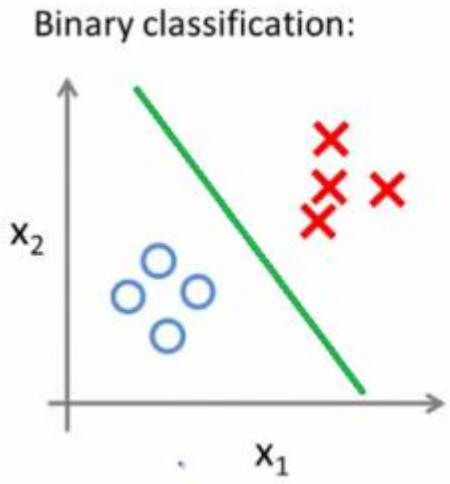


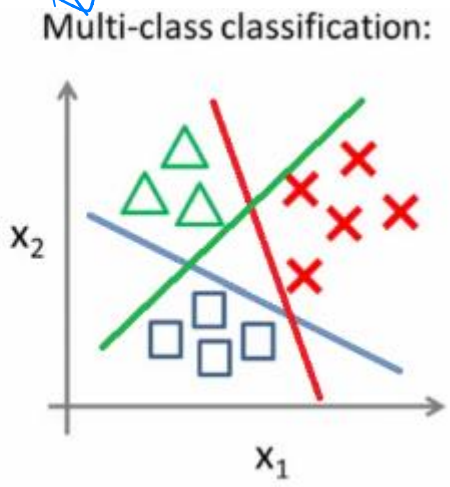
# Recap: Classification vs Regression

## Classification vs Regression

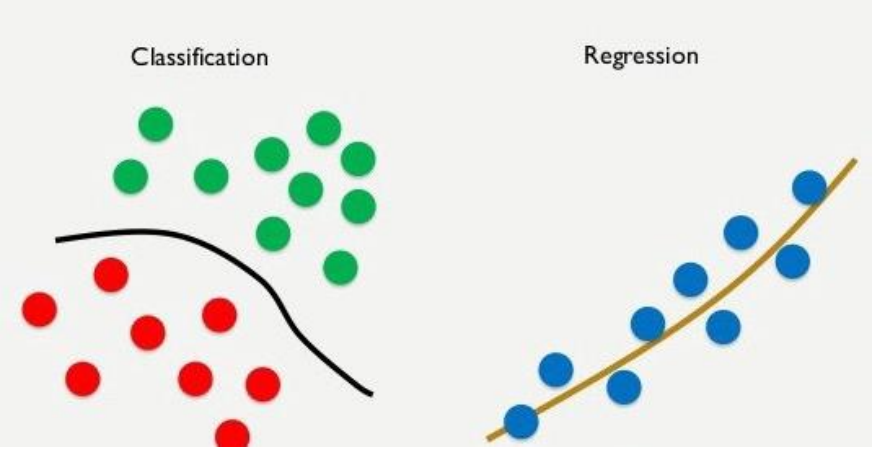
- Input: pairs of points $(x^{(1)}, y^{(1)}), \ldots, (x^{(M)}, y^{(M)})$ with $x^{(m)} \in \mathbb{R}^n$ [Feature vector]
- Regression case: $y^{(m)} \in \mathbb{R}$
- Classification case: $y^{(m)} \in [0, k-1]$ [k-class classification]
- If k = 2, we get Binary classification



# Recap: Examples of Supervised Learning




**Classification**

- Spam email detection
- Handwritten digit recognition
- Medical Diagnosis
- Fraud Detection
- Face Recognition

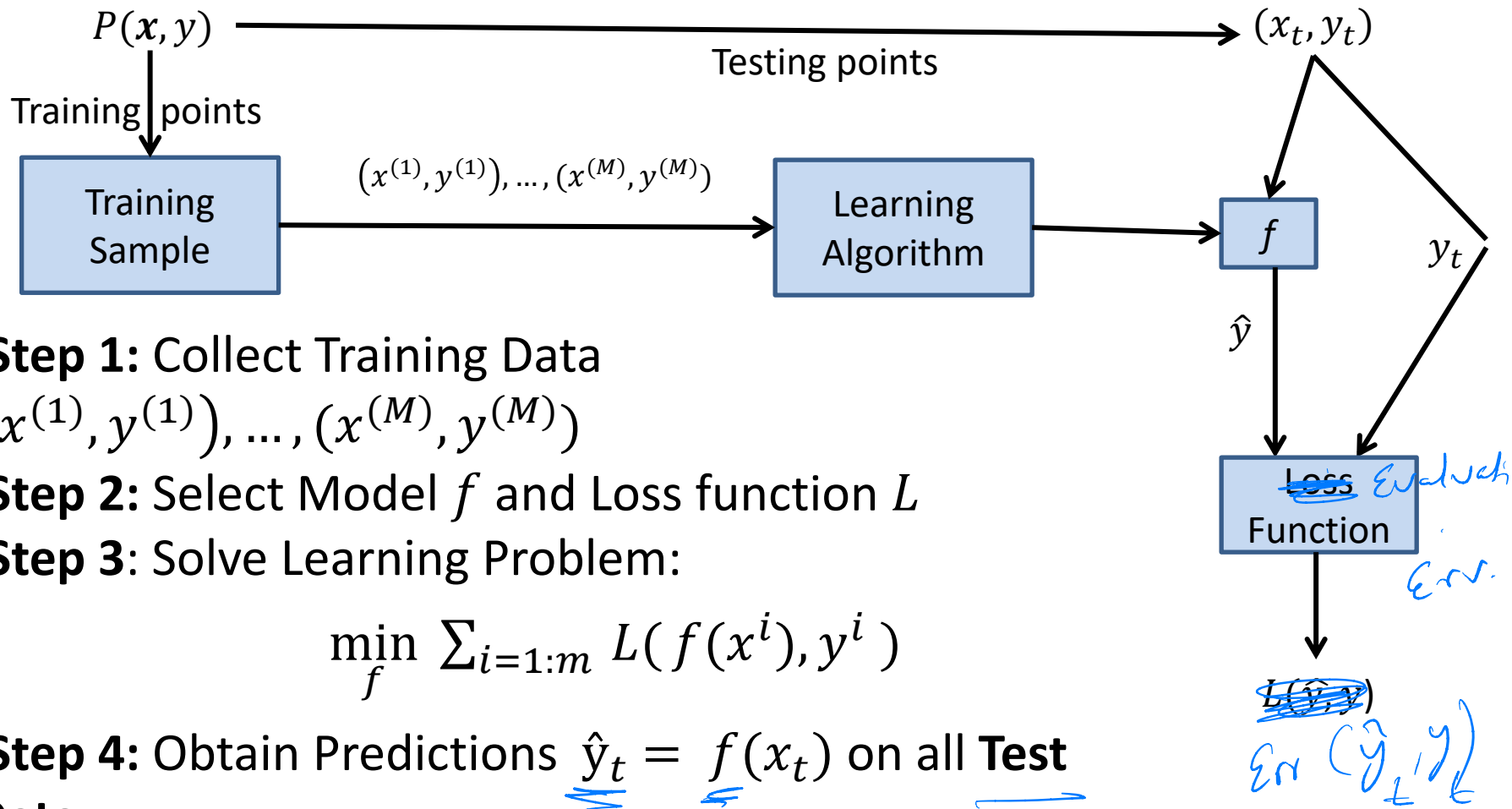
**Regression**

- Housing Price Prediction
- Stock Market Prediction
- Weather Prediction
- Market Analysis and Business Trends

# Recap: Hypothesis Space

- Hypothesis space (Aka Model): set of allowable functions $f: X \rightarrow Y$
- Goal: find the "best" element of the hypothesis space
  - How do we measure the quality of $f$?

# Recap: Supervised Learning Workflow




- **Step 1:** Collect Training Data
  $(x^{(1)}, y^{(1)}), \ldots, (x^{(M)}, y^{(M)})$
- **Step 2:** Select Model $f$ and Loss function $L$
- **Step 3**: Solve Learning Problem:

$$\min_f \sum_{i=1:m} L(f(x^i), y^i)$$

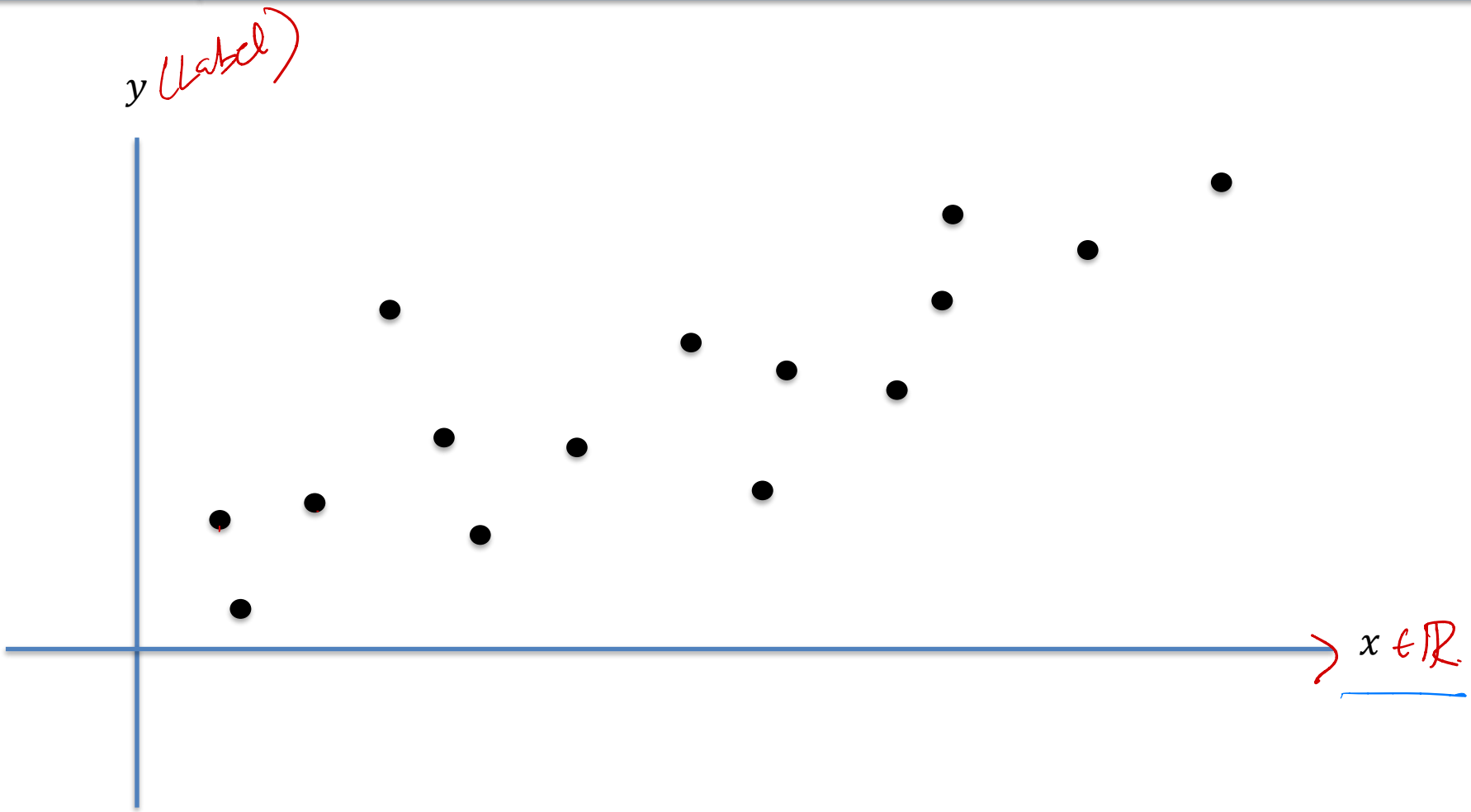
- **Step 4:** Obtain Predictions $\hat{y}_t = f(x_t)$ on all **Test Data**
- **Step 5:** Evaluation -- Measure the error $Err(\hat{y}_t, y_t)$

# Linear Regression

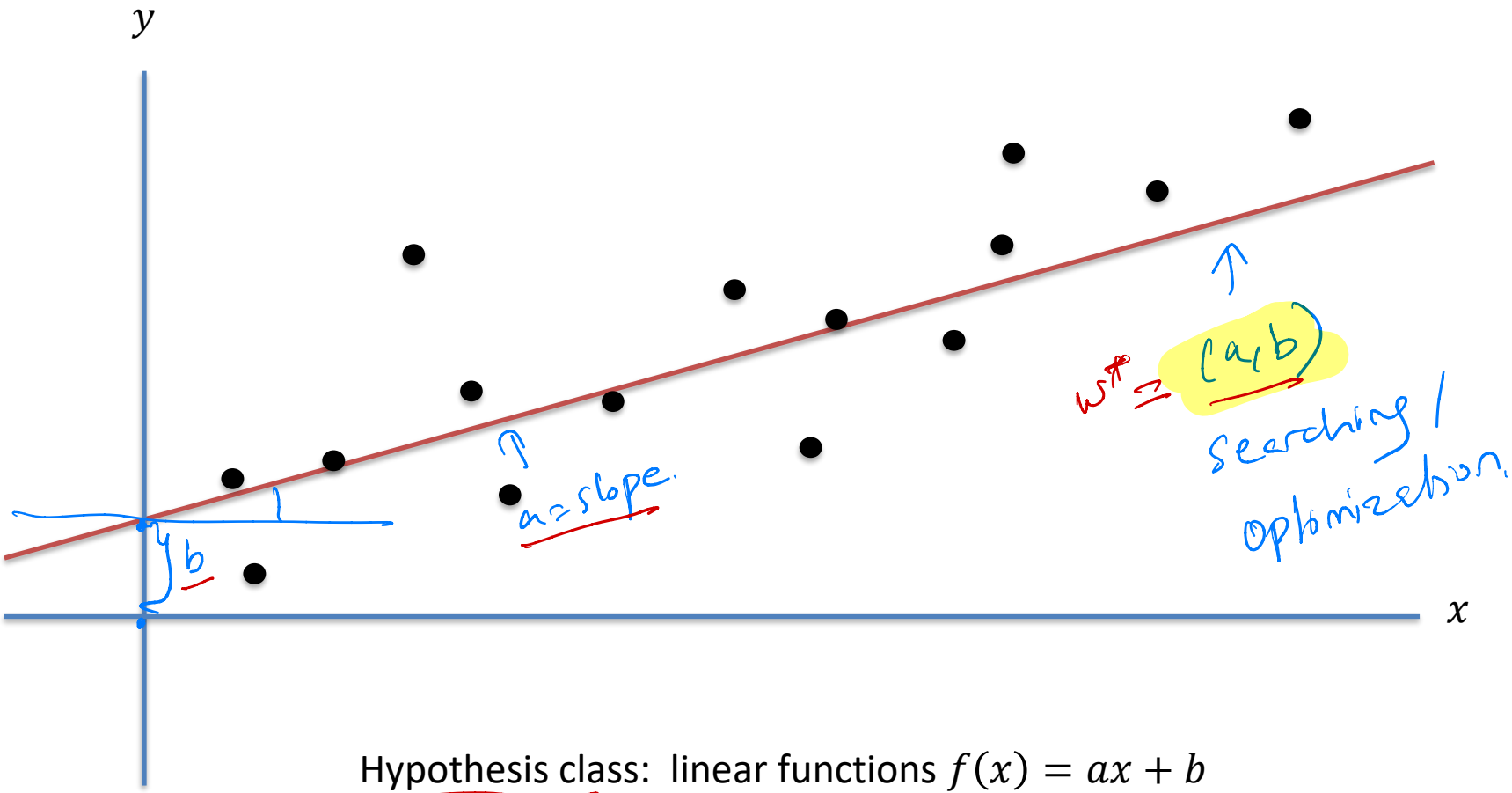
- Simple linear regression
  - Input: pairs of points $\left(x^{(1)}, y^{(1)}\right), \ldots, \left(x^{(M)}, y^{(M)}\right)$ with $x^{(m)} \in \mathbb{R}^d$ and $y^{(m)} \in \mathbb{R}$ (Regression)
  - Hypothesis space: set of linear functions $f(x) = a^T x + b$ with $a \in \mathbb{R}^n, b \in \mathbb{R}$
  - In one dimension, $a, b \in \mathbb{R}$ and $f(x) = ax + b$
  - Error metric and Loss Function: squared difference between the predicted value and the actual value

$$L(y, \hat{y}) = Err(y, \hat{y}) = [y - \hat{y}]^2$$

# Regression



# Regression




Hypothesis class: linear functions $f(x) = ax + b$

How do we compute the error of a specific hypothesis?

# Linear Regression

- For any data point, $x$, the learning algorithm predicts $f(x)$
- In typical regression applications, measure the fit using a squared loss function

$$L(f) = \frac{1}{M}\sum_{m=1}^{M}\left(f\left(x^{(m)}\right) - y^{(m)}\right)^2$$

$f(x) = a \cdot x + b$

$L(a,b)$

$= \frac{1}{M}\sum_{m=1}^{M}\left[a^T x^{(m)} + b - y^{(m)}\right)^2$

- Want to minimize the average loss on the training data
- The optimal linear hypothesis is then given by

$$\min_{a,b}\frac{1}{M}\sum_{m}\left(a^T x^{(m)} + b - y^{(m)}\right)^2$$

$a \in \mathbb{R}^n$, $x^{(m)} \in \mathbb{R}^n$, $b \in \mathbb{R}$

## Detour

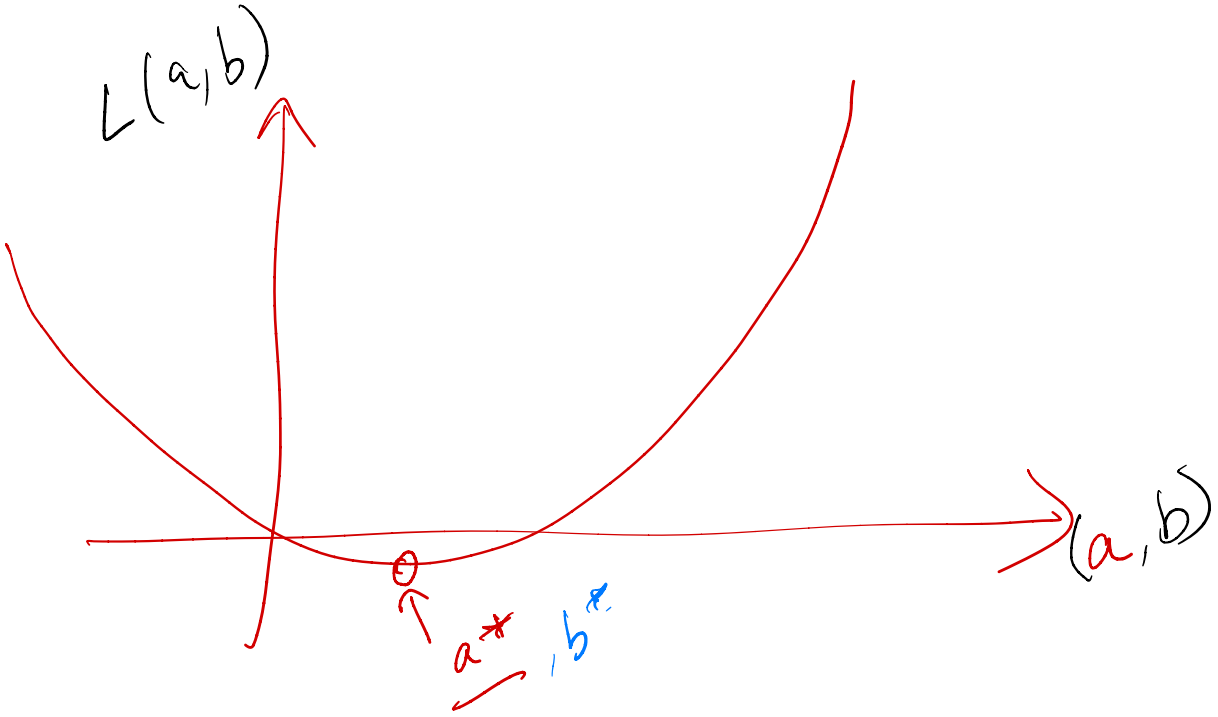
(1) $x^{(m)} \rightarrow$ $m^{th}$ training data point.

(2) $x_m \rightarrow$ $m^{th}$ co-ordinate / feature.

# Linear Regression [optimization / searching]

$$\min_{a,b} \frac{1}{M} \sum_{m} \left( ax^{(m)} + b - y^{(m)} \right)^2$$

- How do we find the optimal $a$ and $b$?



# Linear Regression

$$\min_{a,b} \frac{1}{M} \sum_{m} \left( a x^{(m)} + b - y^{(m)} \right)^2$$

$a \in \mathbb{R}^d$, $b \in \mathbb{R}$

$L(a,b)$

- How do we find the optimal $a$ and $b$?
  - Solution 1: take derivatives and solve (there is a closed form solution!)
  - Solution 2: use gradient descent

$\min_{a,b} L(a,b)$

$a^*: \nabla L(a) = 0$

$b^*: \nabla L(b) = 0$

Solve $\nabla L \begin{pmatrix} a^* \\ b \end{pmatrix} = 0$

Linear Regression Workflow

1. Model $f: a^T x + b$
2. Loss $L: [a^T x + b - y]^2$ (x: feat, y: label)
3. OPT $: \min_{a,b} \sum_m (a^T x^{(m)} + b - y^{(m)})^2$
   - ↳ opt algo: gradient descent.
4. Eval — RMSE, R2

# Linear Regression

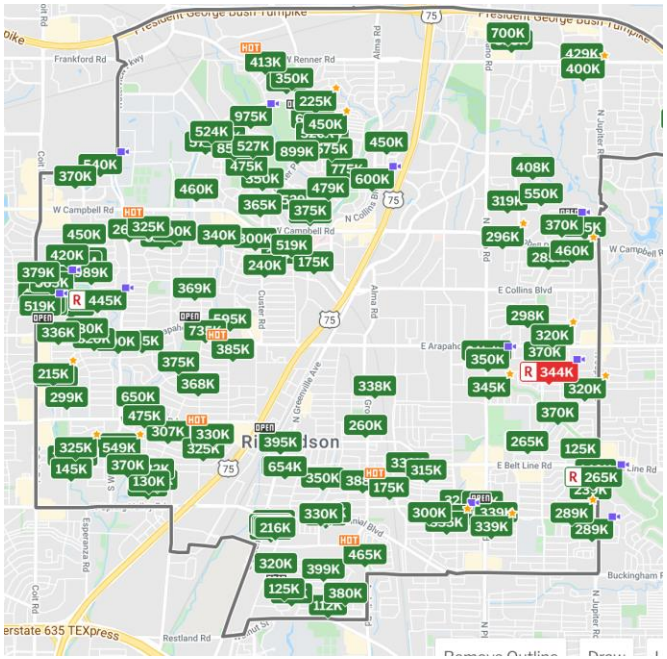
$$\min_{a,b} \frac{1}{M} \sum_{m} \left( ax^{(m)} + b - y^{(m)} \right)^2$$

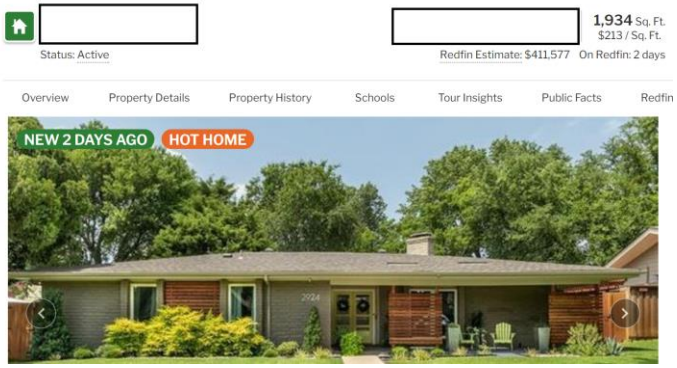
- How do we find the optimal $a$ and $b$?
  - Solution 1: take derivatives and solve (there is a closed form solution!)
  - Solution 2: use gradient descent

Part II

    - This approach is much more likely to be useful for general loss functions

# Recap – Housing Price Prediction Application




**Home Facts**

| | | | |
|---|---|---|---|
| Status | Active | Time on Redfin | 2 days |
| Property Type | Residential, Single Family | HOA Dues | $4/month |
| Year Built | 1969 | Style | Single Detached, Mid-Century Modern, Ranch, Traditional |
| Community | Canyon Creek Country Club 9 | Lot Size | 10,019 Sq. Ft. |
| MLS# | 14375892 | | |

$y$ = Price of home

$x$ = features

[Loc$^n$, Size, Bd(Bath), HOA, Prop Tax, Crime, ...]

[Zipcode]

# Part II: Gradient Descent and Optimization

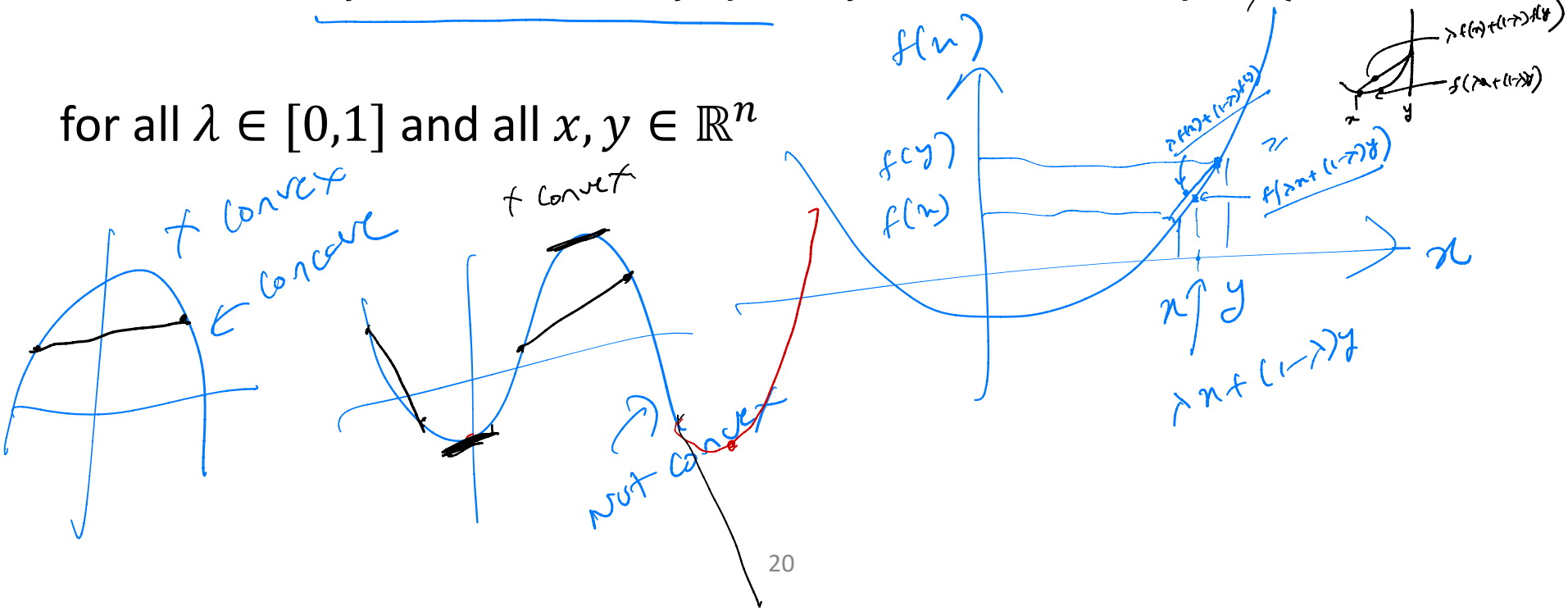
# Gradient Descent

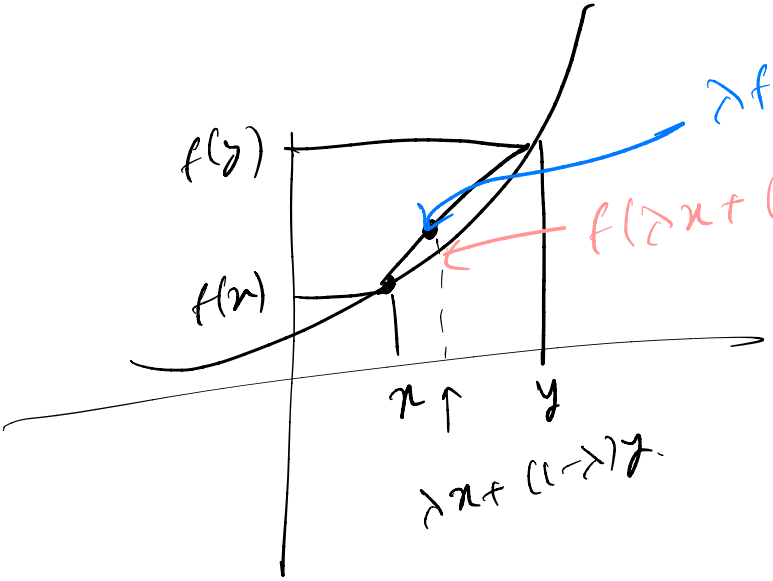
Iterative method to minimize a (convex) differentiable function $f$

A function $f: \mathbb{R}^n \to \mathbb{R}$ is convex if $x, y \in \mathbb{R}^n$

$$\lambda f(x) + (1-\lambda) f(y) \geq f(\lambda x + (1-\lambda) y)$$

for all $\lambda \in [0,1]$ and all $x, y \in \mathbb{R}^n$






$\lambda f(x) + (1-\lambda) f(y)$

$\geq$

$f(\lambda x + (1-\lambda) y)$ , $\forall \lambda \in [0,1]$

$\forall x, y.$

# Gradient Descent

Iterative method to minimize a (convex) differentiable function $f$

- Pick an initial point $x_0$
- Iterate until convergence

$$x_{t+1} = x_t - \gamma_t \nabla f(x_t)$$

where $\gamma_t$ is the $t^{th}$ step size (sometimes called learning rate)

$$L(a,b) = \sum_{m} \left[ a^T x^{(m)} + b - y^{(m)} \right]^2$$

Convex

Pick $a_0, b_0$ random / zeros

$$a_{t+1} = a_t - \gamma_t \nabla L(a_t)$$
$$b_{t+1} = b_t - \gamma_t \nabla L(b_t)$$

repeat k times until convergence.

$$\theta_{t+1} = \theta_t - \gamma_t \nabla f(\theta_t)$$

Learning Rate.

# Convergence

1. $x^t \rightarrow x^{t+1}$

$$|f(x^{t+1}) - f(x^t)| \leq \delta \quad [1e^{-5}]$$

2. Fix number of iterations

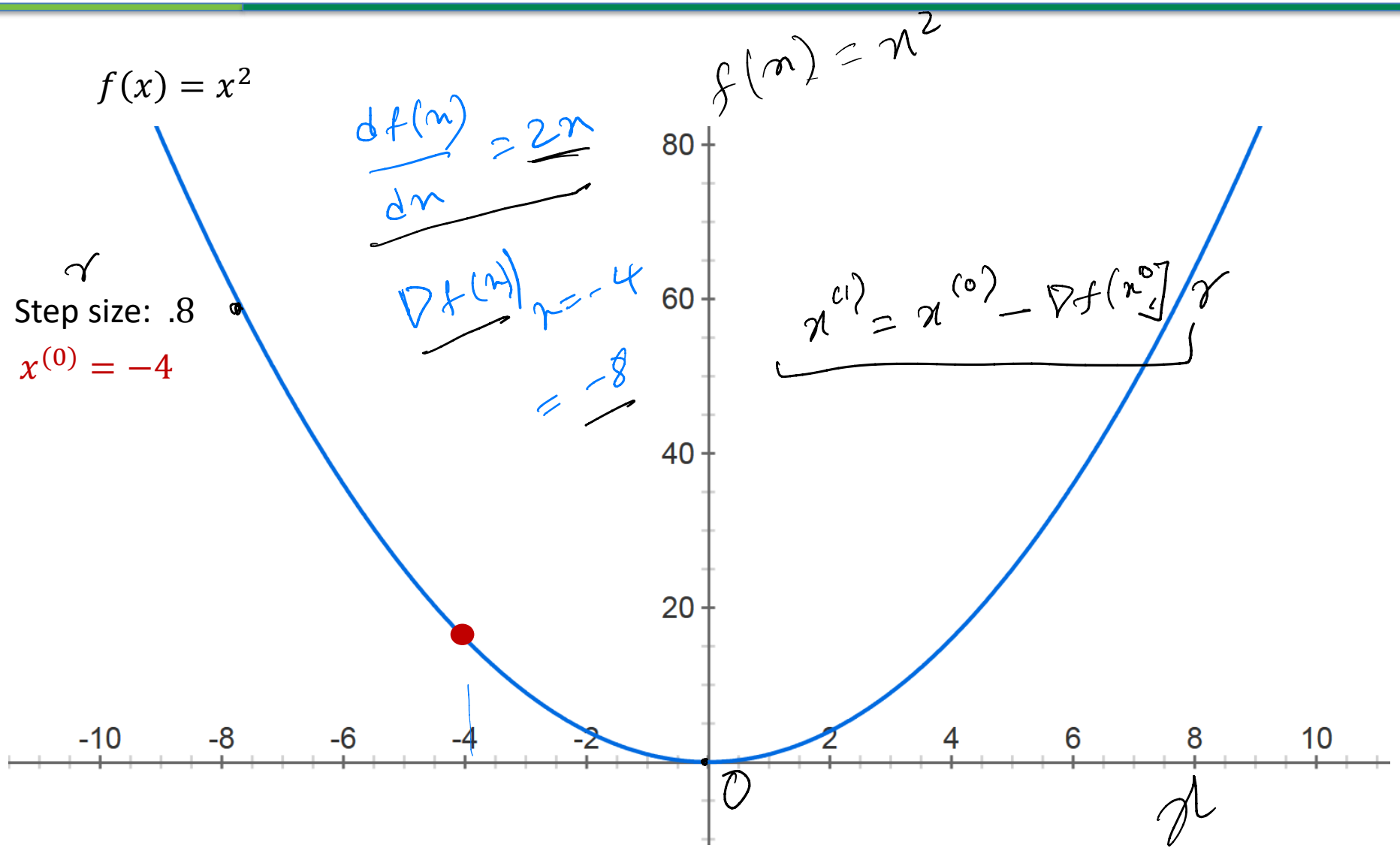
3. $\|\nabla f(x^t)\| \leq \delta \quad [1e^{-5}]$

4. $\|x^{t+1} - x^t\| \leq \delta$

# Basics of Convexity and Gradient Desc

- For additional reading, please see some of my slides from my Spring 2020 Class "Optimization in Machine Learning"
- Github Location for Lecture Notes and Slides: https://github.com/rishabhk108/OptimizationML
- Please skim through:
  - Lectures 1 and 2 for basics
  - Lectures 3-5 for convex functions
  - Lectures 6-8 on Gradient Descent
  - This includes slightly more mathematical details like convergence analysis and proofs for convergence etc.

# Gradient Descent



# Gradient Descent

$f(x) = x^2$

$x^{(1)} = x^{(0)} - 2\gamma x^{(0)}$

$= x^0(1 - 2\gamma)$

$= -4(1 - 1.6)$

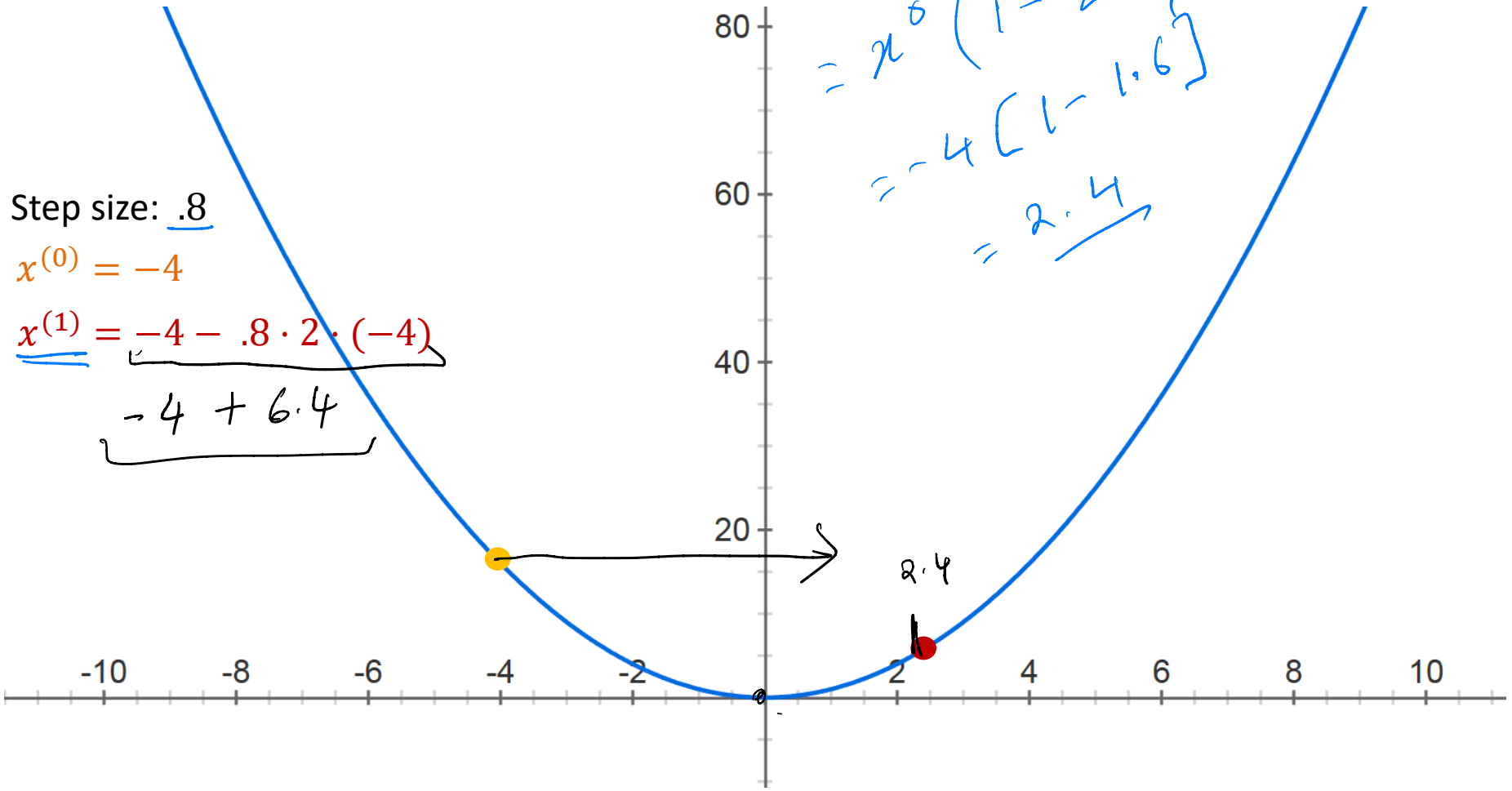
$= 2.4$

Step size: .8

$x^{(0)} = -4$

$x^{(1)} = -4 - .8 \cdot 2 \cdot (-4)$

$-4 + 6.4$



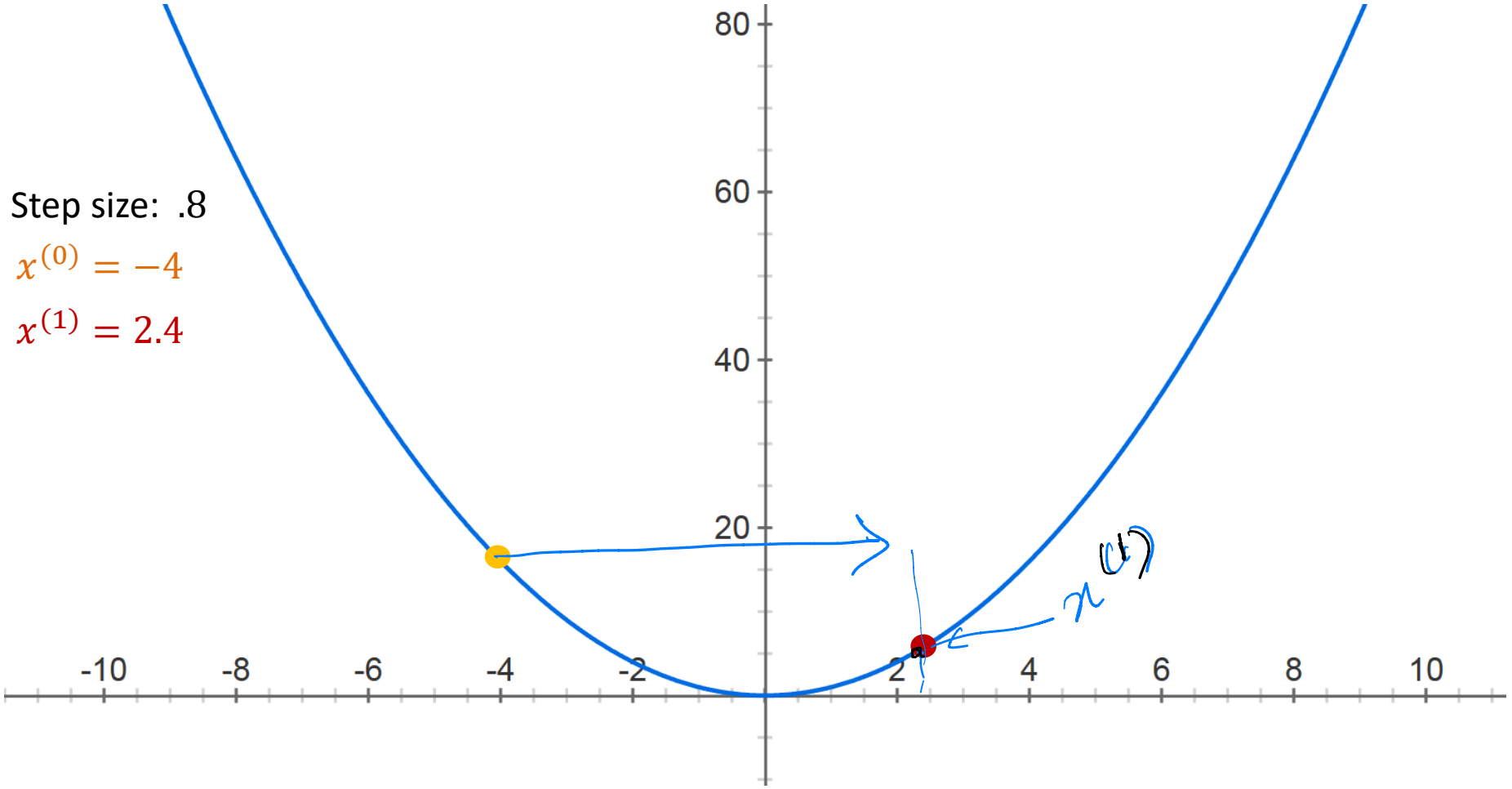
# Gradient Descent

$f(x) = x^2$

Step size: .8

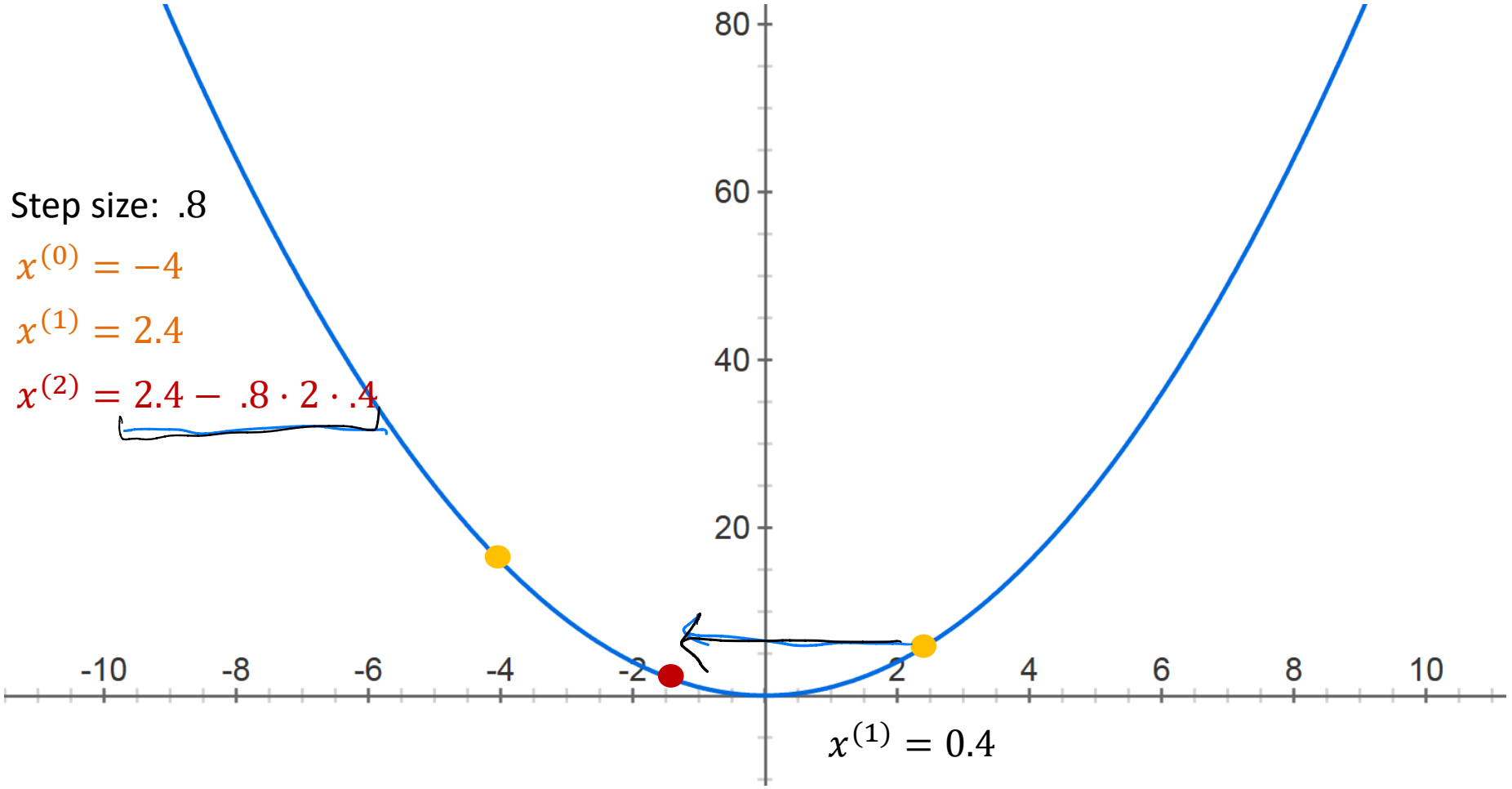
$x^{(0)} = -4$

$x^{(1)} = 2.4$



# Gradient Descent

$f(x) = x^2$




Step size: .8

$x^{(0)} = -4$

$x^{(1)} = 2.4$

$x^{(2)} = 2.4 - .8 \cdot 2 \cdot .4$

# Gradient Descent

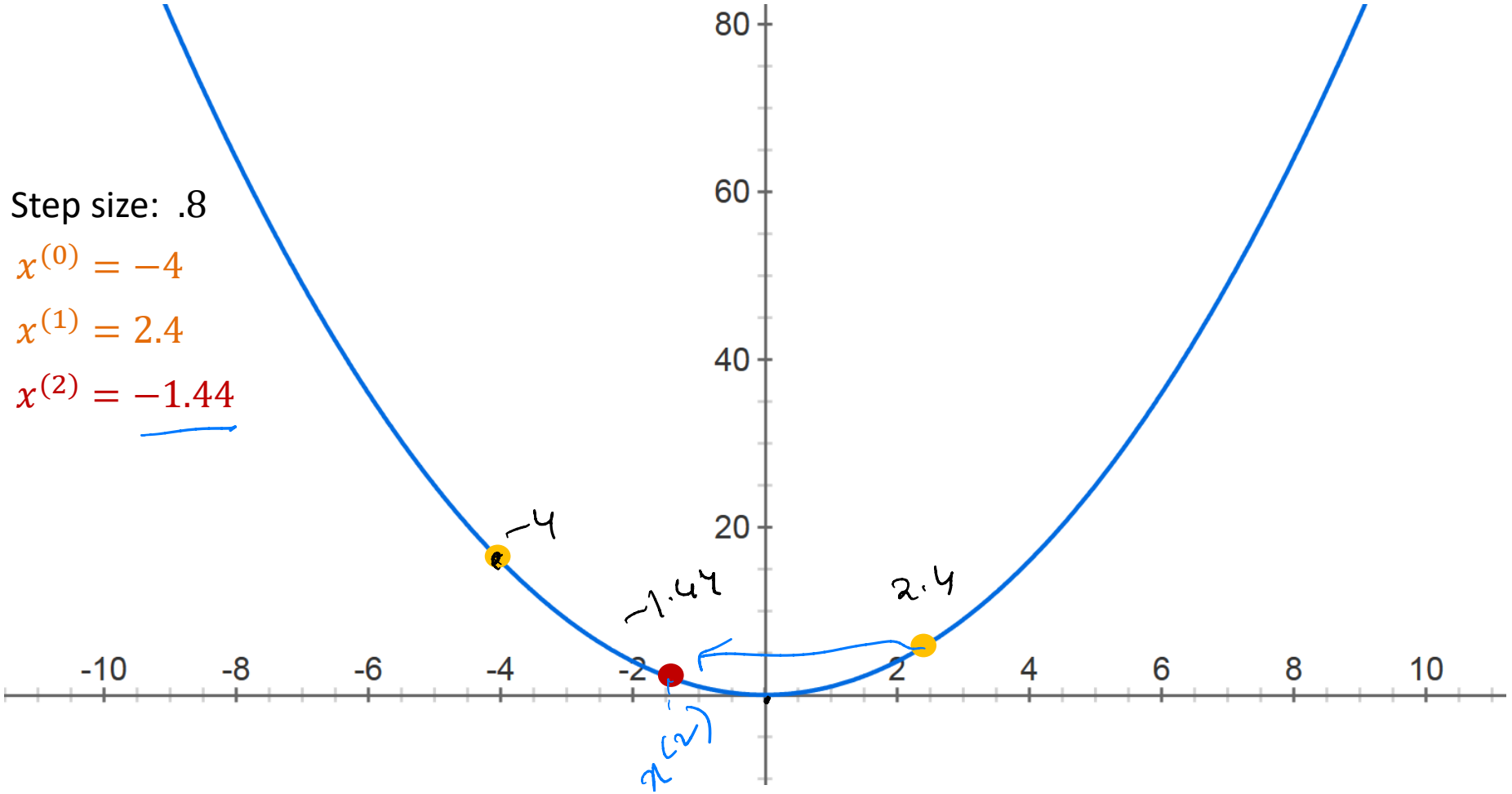
$f(x) = x^2$

Step size: .8

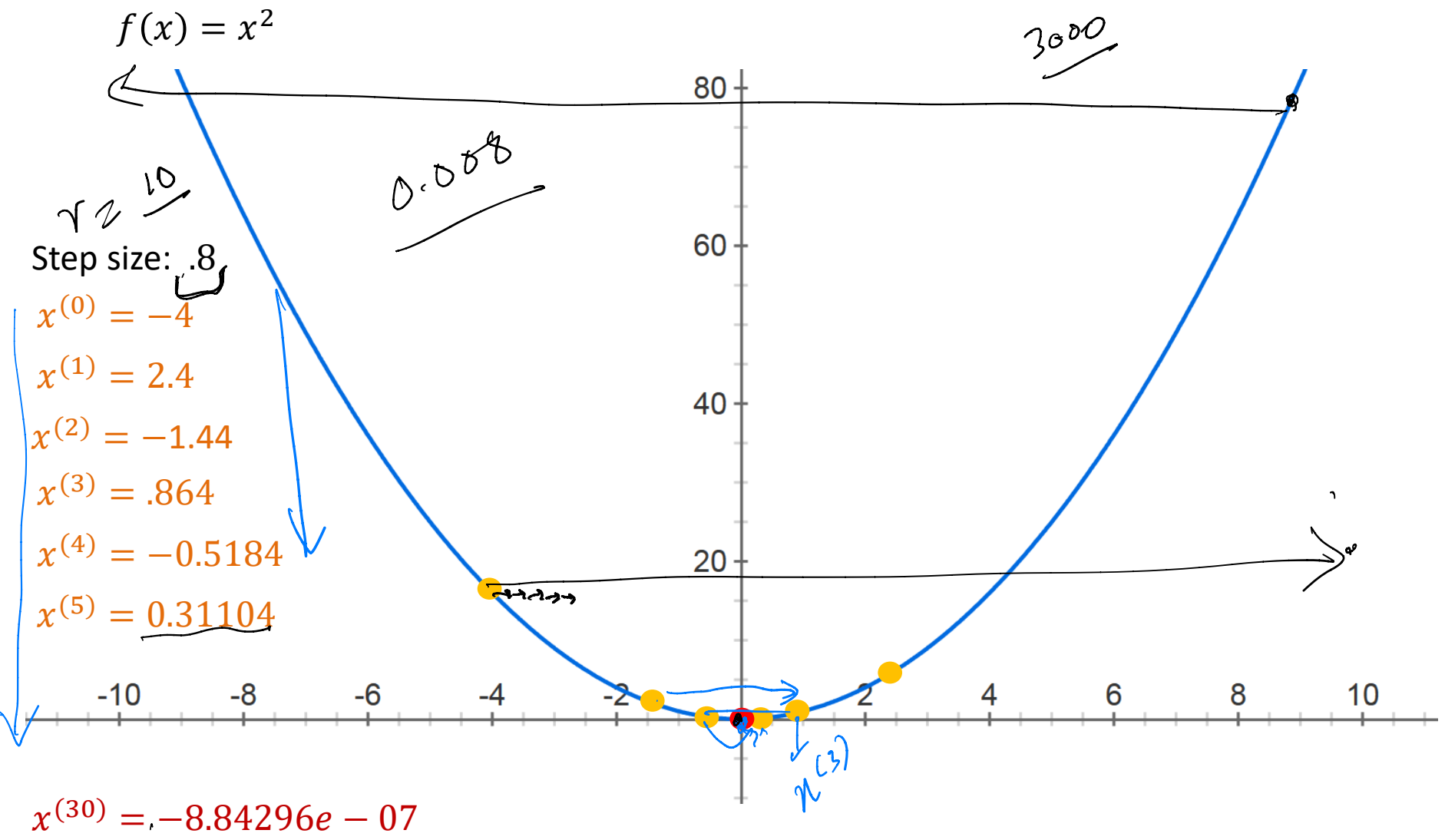
$x^{(0)} = -4$

$x^{(1)} = 2.4$

$x^{(2)} = -1.44$



# Gradient Descent (Numerical Algo)



# Gradient Descent

$a, x^{(m)} \in \mathbb{R}$

$$\min_{a,b} \frac{1}{M} \sum_{m} \underbrace{\left(ax^{(m)} + b - y^{(m)}\right)^2}_{L(a,b)}$$

- What is the gradient of this function?

- What does a gradient descent iteration look like for this simple regression problem?

$$\frac{dL}{da} = \frac{1}{M} \sum_{m} 2\left(ax^{(m)} + b - y^{(m)}\right) x^{(m)}$$

# Derivation of the Gradient (one dim)

$$f(a,b) = \frac{1}{M}\sum_{m=1}^{M}\left[a\cdot x^{(m)} + b - y^{(m)}\right]^2$$

$a \in \mathbb{R}$, $b \in \mathbb{R}$

$$\frac{df}{da} = \nabla f(a) = \frac{1}{M}\sum_{m=1}^{M} 2\left[a\cdot x^{(m)} + b - y^{(m)}\right]x^{(m)}$$

$$\nabla f(b) = \frac{1}{M}\sum_{m=1}^{M} 2\left(a x^{(m)} + b - y^{(m)}\right]$$

$$a^{(t+1)} = a^{(t)} - \gamma_t \nabla f(a)\Big|_{a=a^t}$$

$$b^{(t+1)} = b^{(t)} - \gamma_t \nabla f(b)\Big|_{b=b^t}$$

# Linear Regression

- In higher dimensions, the linear regression problem is essentially the same with $x^{(m)} \in \mathbb{R}^n$

$$\min_{a \in \mathbb{R}^n, b} \frac{1}{M} \sum_{m} \left( a^T x^{(m)} + b - y^{(m)} \right)^2$$

- Can still use gradient descent to minimize this
  - Not much more difficult than the $n = 1$ case

# Derivation of the Gradient (> 1 dim)

$$f(a,b) = \frac{1}{M}\sum_{m=1}^{M}\left[a^T x^{(m)} + b - y^{(m)}\right]^2$$

$$\underbrace{\nabla f(a)}_{\in \mathbb{R}^n} = \frac{1}{M}\sum_{m=1}^{M} \underbrace{2\left[a^T x^{(m)} + b - y^{(m)}\right]}_{\text{scalar}} \underbrace{x^{(m)}}_{\mathbb{R}^n}$$

$\nabla f(b)$: same as 1d case.

$g(a) = a^T x^{(m)}$

$\nabla g(a) = x^{(m)}$

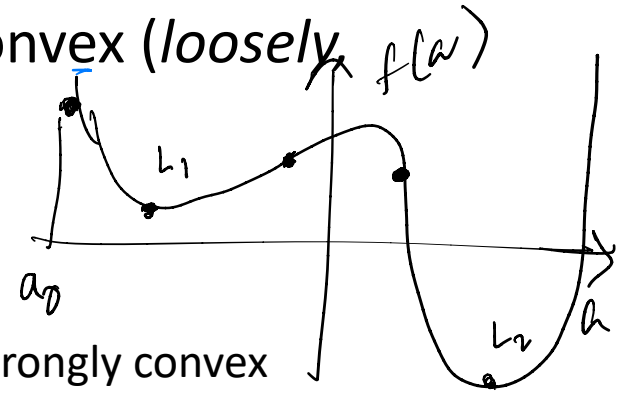
$$\nabla g(a) = \begin{bmatrix} \partial g/\partial a_1 \\ \partial g/\partial a_2 \\ \vdots \\ \partial g/\partial a_n \end{bmatrix} = x^{(m)}$$

$$g(a) = \sum_{i=1}^{n} a_i x_i^{(m)}$$

$$\frac{\partial g}{\partial a_i} = x_i^{(m)}$$

# Gradient Descent

- Gradient descent converges under certain technical conditions on the function $f$ and the step size $\gamma_t$
  - If $f$ is convex, then any fixed point of gradient descent must correspond to a global minimum of $f$
  - In general, for a nonconvex function, may only converge to a local optimum
- Very fast convergence because the Linear Regression is *smooth* (loosely, think *differentiable*) and strongly convex (*loosely bounded below by a quadratic function*)*



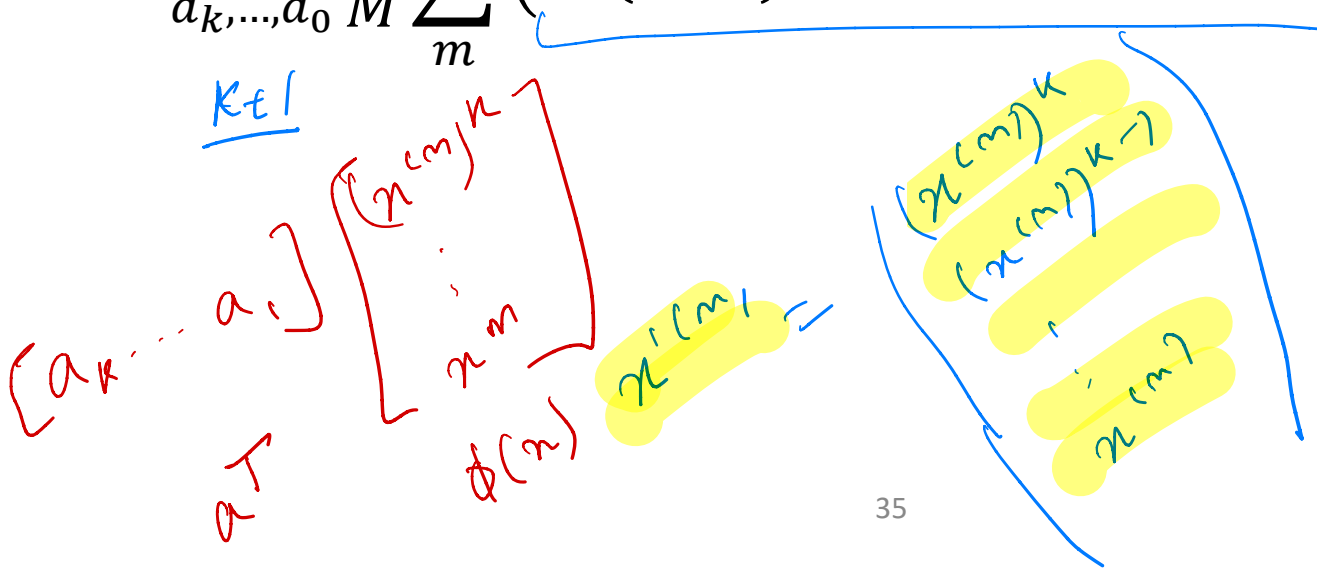
* See Lectures 6-8 for better understanding of smooth and strongly convex

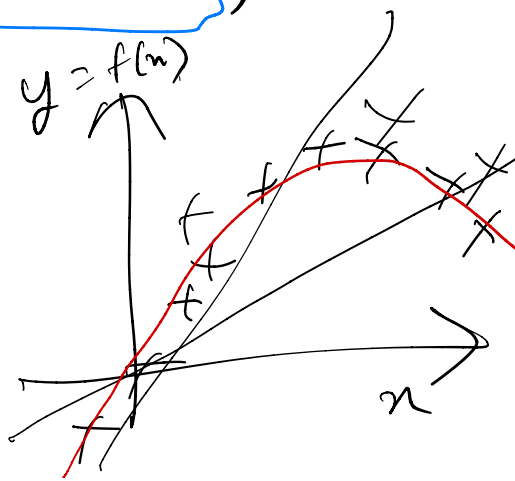
# Part III: Polynomial Regression

# Polynomial Regression

- What if we enlarge the hypothesis class?
  - Quadratic functions: $ax^2 + bx + c$
  - $k$-degree polynomials: $a_k x^k + a_{k-1} x^{k-1} + \cdots + a_1 x + a_0$

$$\min_{a_k,\ldots,a_0} \frac{1}{M} \sum_m \left( a_k \left(x^{(m)}\right)^k + \cdots + a_1 x^{(m)} + a_0 - y^{(m)} \right)^2$$





# Polynomial Regression as Linear regression

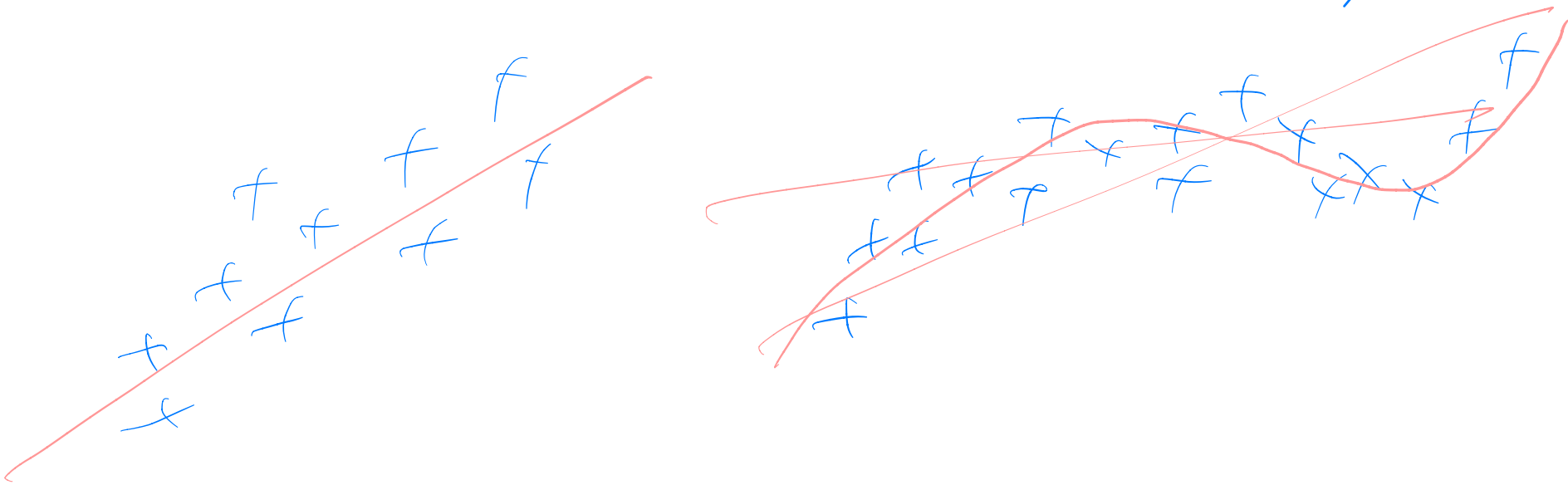
$$L(a_0, \dots a_k) = \sum_m \left[ a_k \left[x^{(m)}\right]^k + a_{k-1} \left[x^{(m)}\right]^{k-1} + \dots + a_0 - y \right]^2$$

$$\left( \begin{bmatrix} a_k, a_{k-1}, \dots, a_0 \end{bmatrix} \begin{bmatrix} \left(x^{(m)}\right)^k \\ \left(x^{(m)}\right)^{k-1} \\ \vdots \\ x^{(m)} \\ 1 \end{bmatrix} - y \right)^2$$

1D Poly-Regression $\equiv$ k-Dim Lin Regression with poly features

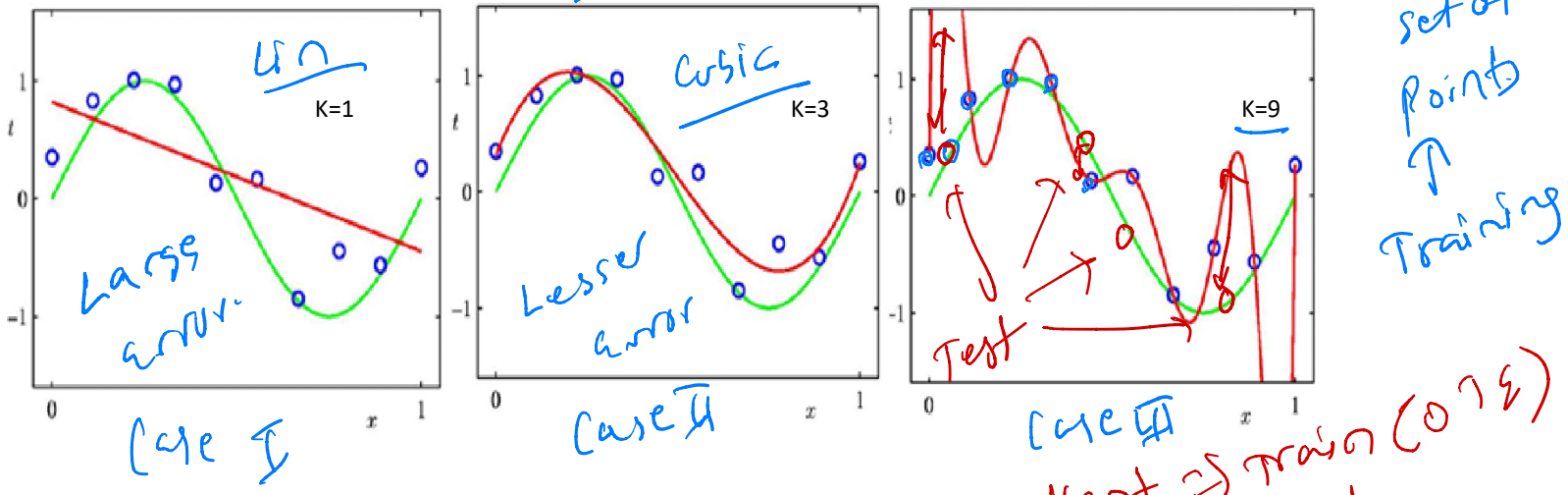
# Polynomial Regression

- What if we enlarge the hypothesis class?
  - Quadratic functions: $ax^2 + bx + c$
  - $k$-degree polynomials: $a_k x^k + a_{k-1} x^{k-1} + \cdots + a_1 x + a_0$
- Can we always learn "better" with a larger hypothesis class?



# Polynomial Regression

- What if we enlarge the hypothesis class?
  - Quadratic functions: $ax^2 + bx + c$
  - $k$-degree polynomials: $a_k x^k + a_{k-1} x^{k-1} + \cdots + a_1 x + a_0$
- Can we always learn "better" with a larger hypothesis class?

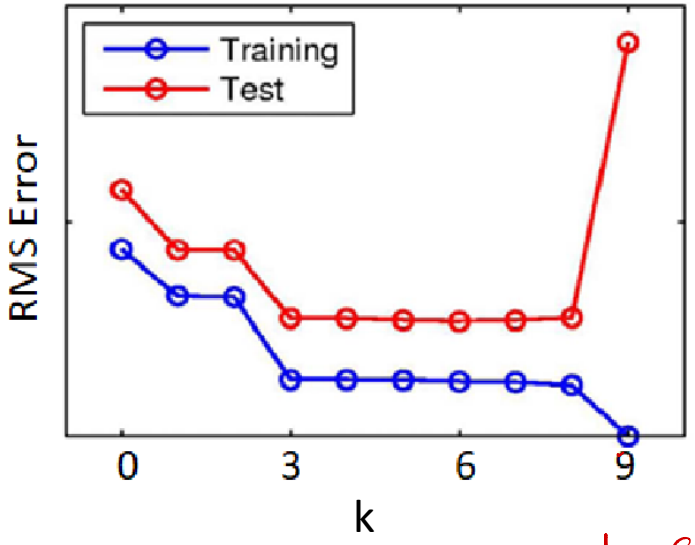


# Caveats with Polynomial Regression

- Larger hypothesis space always decreases the cost function, but this does NOT necessarily mean better predictive performance
  - This phenomenon is known as overfitting
  - Ideally, we would select the simplest hypothesis consistent with the observed data
- In practice, we cannot simply evaluate our learned hypothesis on the training data, we want it to perform well on unseen data (otherwise, we can just memorize the training data!)
  - Report the loss on some held-out test data (i.e., data not used as part of the training process)

# Overfitting

- As *the degree of the polynomial (k)* increases, training error decreases monotonically
- As *k* increases test error can increase
- Test error can decrease at first, but increases
- Overfitting can occur
  - When the model is too complex and trivially fits the data (i.e., too many parameters)
  - When the data is not enough to estimate the parameters
  - Model captures the noise (or the chance)




Too many parameters

"Regularization"

# Part IV: Hands On

# House Price Prediction

- Boston House Price Dataset

UCI Repo

**CRIM:** Per capita crime rate by town (crime)
**ZN:** Proportion of residential land zoned for lots over 25,000 sq. ft
**INDUS:** Proportion of non-retail business acres per town
**CHAS:** Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
**NOX:** Nitric oxide concentration (parts per 10 million)
**RM:** Average number of rooms per dwelling
**AGE:** Proportion of owner-occupied units built prior to 1940
**DIS:** Weighted distances to five Boston employment centers
**RAD:** Index of accessibility to radial highways
**TAX:** Full-value property tax rate per $10,000
**PTRATIO:** Pupil-teacher ratio by town
**B:** $1000(Bk - 0.63)^2$, where Bk is the proportion of [people of African American descent] by town
**LSTAT:** Percentage of lower status of the population
**MEDV:** Median value of owner-occupied homes in $1000s → Label / Target

Feat

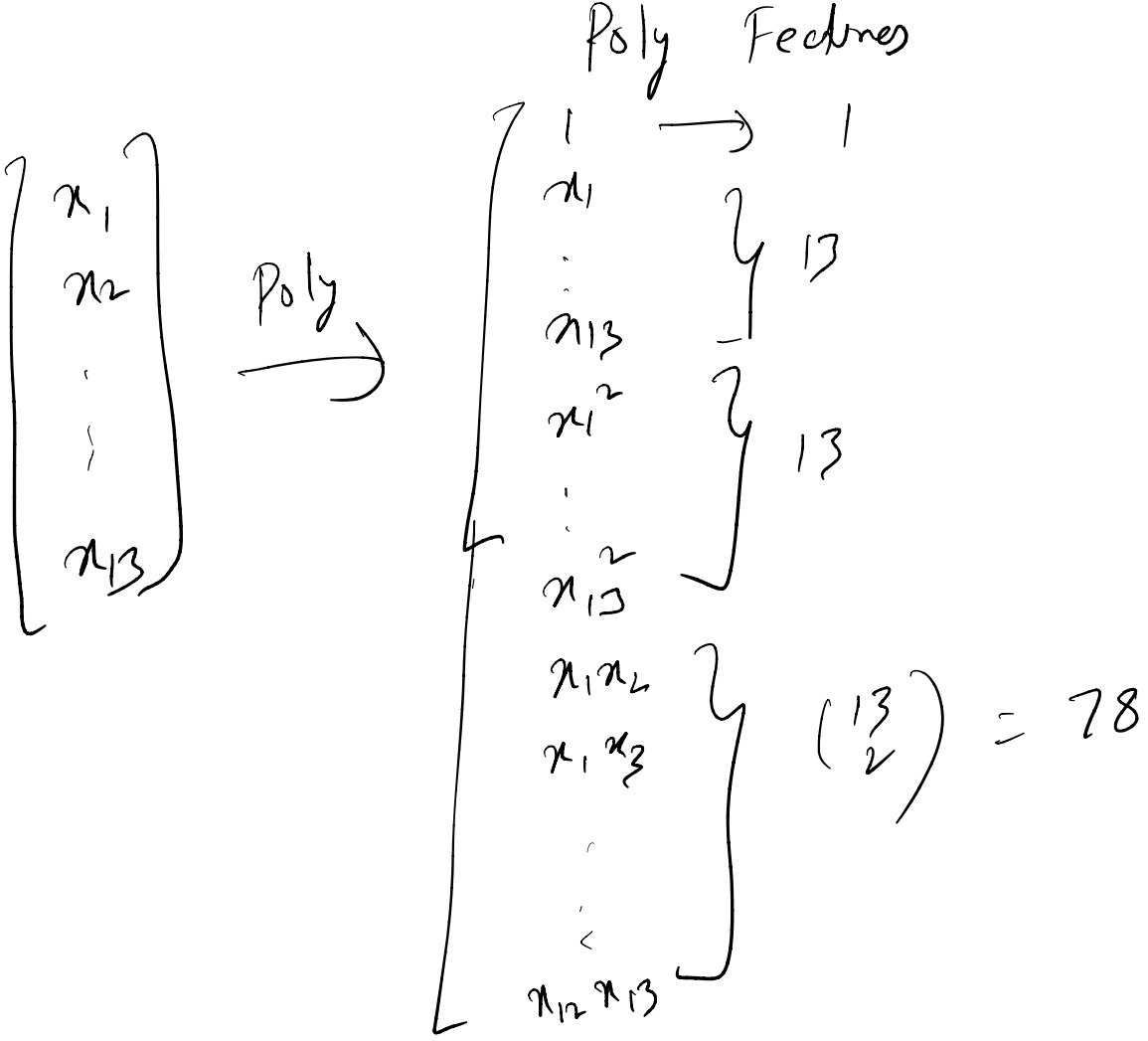
# Summary of the Hands On Portion

- Load the Dataset (sk-learn)
- Exploratory Data Analysis (numpy)
- Training a Linear Regression Model
- Training a Polynomial Regression Model

→ inbuilt LR in sk-learn

- Training a Linear/Poly Regression from scratch using gradient descent

Contrast to inbuilt LR


Poly Features
$x_1$
$x_2$
$x_{13}$
Poly
1 → 1
$x_1$
$x_{13}$
13
$x_1^2$
$x_{13}^2$
13
$x_1x_2$
$x_1x_3$
$x_{12}x_{13}$
$\binom{13}{2} = 78$

# Linear Regression Closed form

$$L(w,b) = \sum_{i=1}^{m} \left[ w^T x_i + b - y_i \right]^2$$

$$= \| Xw + b1 - y \|^2 \quad \leftarrow X = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_M^T \end{bmatrix} \quad y = \begin{bmatrix} y_1 \\ \vdots \\ y_M \end{bmatrix}$$

$$L(w') = \| X'w' - y \|^2 \quad \text{where} \quad X' = [X \; 1], \; w' = [w, b]^T$$

$$\frac{\partial L}{\partial w'} = 2(X')^T (X'w' - y) = 0$$

$$\Rightarrow (X')^T X' w' = X'y$$

$$\Rightarrow w' = \left[ (X')^T X' \right]^{-1} X'y$$

$(d+1) \times M \quad M \times (d+1) \Rightarrow (d+1) \times (d+1)$

$X \in \mathbb{R}^{M \times d}$

$X^T \in \mathbb{R}^{d \times M}$

$O(M d^2) + O(d^3)$