COMP/EECE 7/8740 Neural Networks

Topics:

Linear regression

- Correlation and linear correlation
- Linear regression with single and multiple variables

Md Zahangir Alom
Department of Computer Science
University of Memphis, TN

Recall: Covariance and interpreting covariance

$$\operatorname{cov}(x,y) = \frac{\sum_{i=1}^{n}(x_i - \bar{X})(y_i - \bar{Y})}{n-1}$$

cov(X,Y) > 0 $\longrightarrow$ X and Y are positively correlated

cov(X,Y) < 0 $\longrightarrow$ X and Y are inversely correlated

cov(X,Y) = 0 $\longrightarrow$ X and Y are independent

Correlation coefficient

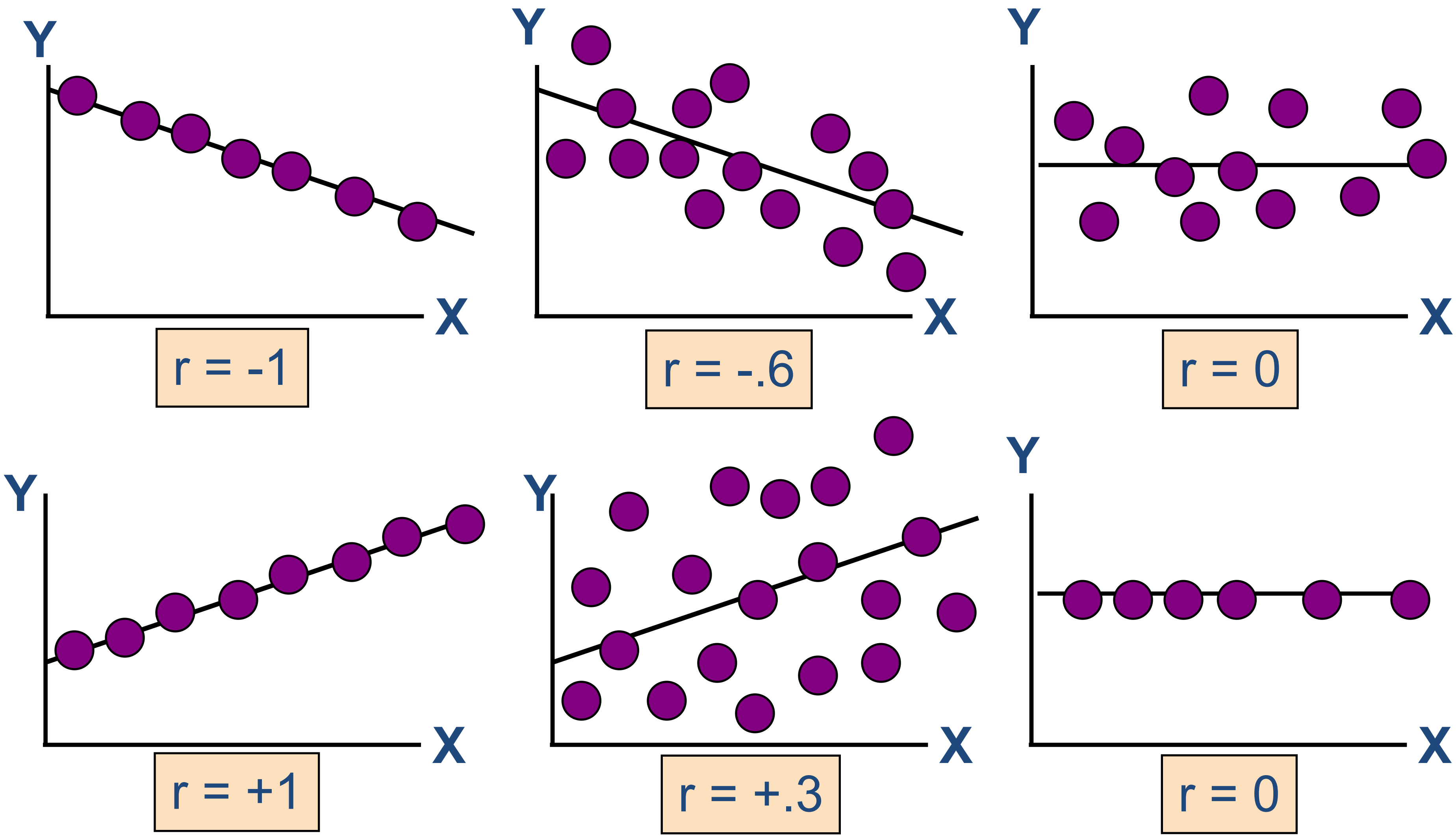
- Pearson's Correlation Coefficient is standardized covariance (unitless):

$$\boxed{r = \frac{\operatorname{cov} ariance(x, y)}{\sqrt{\operatorname{var} x}\sqrt{\operatorname{var} y}}}$$

Correlation

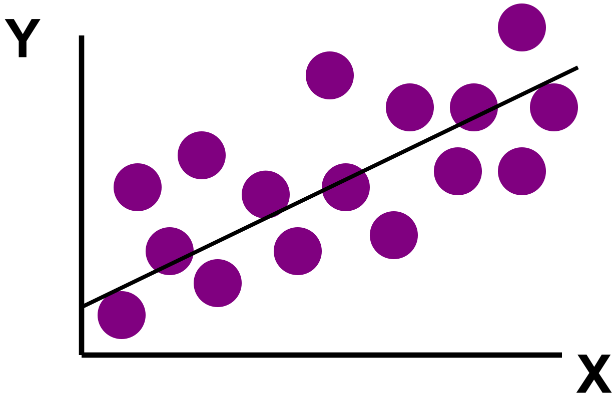
- Measures the relative strength of the *linear* relationship between two variables
  - Ranges between –1 and 1
  - The closer to –1, the stronger the negative linear relationship
  - The closer to 1, the stronger the positive linear relationship
  - The closer to 0, the weaker any positive linear relationship

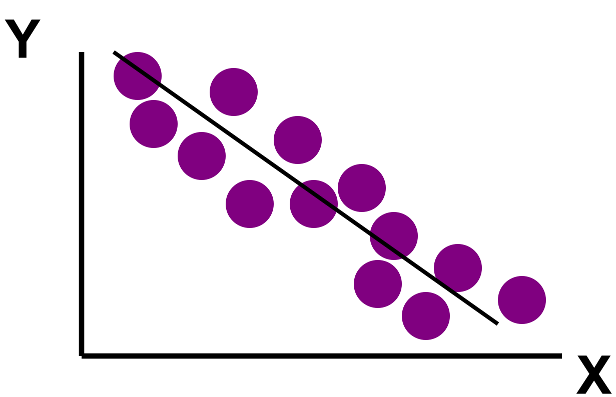
Scatter Plots of Data with Various Correlation Coefficients



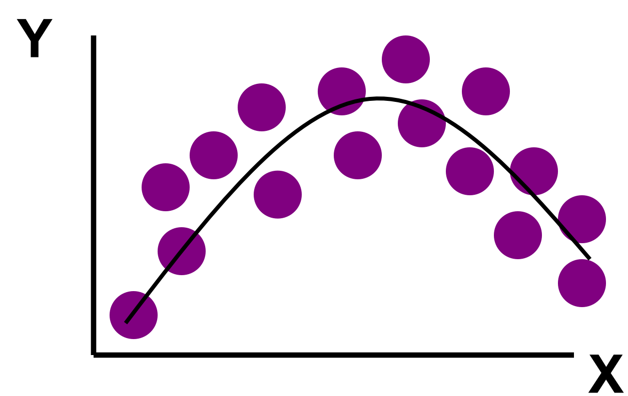
Linear Correlation

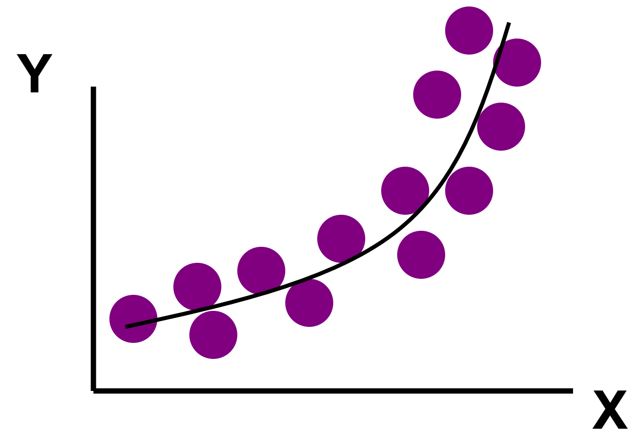
Linear relationships





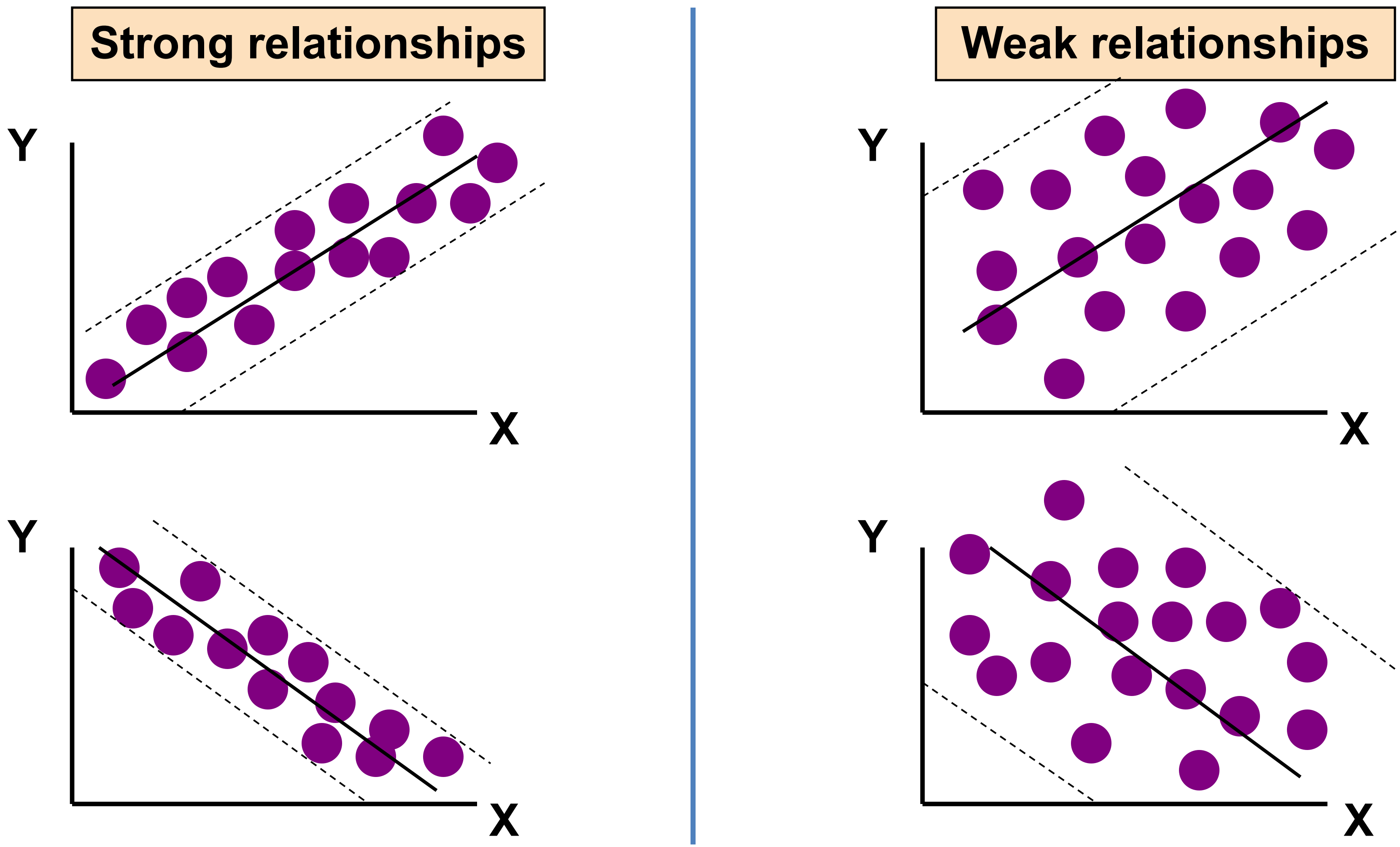
Curvilinear relationships





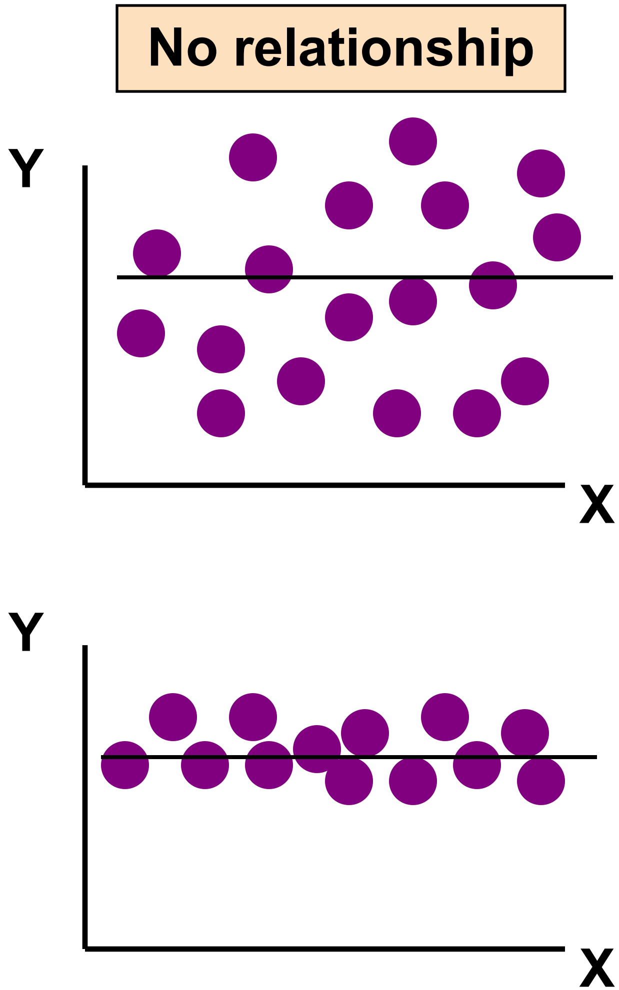
▪Slide from: Statistics for Managers Using Microsoft® Excel 4th Edition, 2004 Prentice-Hall

Linear Correlation



■Slide from: Statistics for Managers Using Microsoft® Excel 4th Edition, 2004 Prentice-Hall

Linear Correlation



■Slide from: Statistics for Managers Using Microsoft® Excel 4th Edition, 2004 Prentice-Hall

Linear regression

What is Linear Regression?

Linear regression is a **basic and commonly used** type of predictive analysis.

LR examine two things:

- Does a set of predictor variables do a good job in **predicting an outcome (dependent) variable**?
- Which **variables in particular are significant predictors** of the outcome variable?

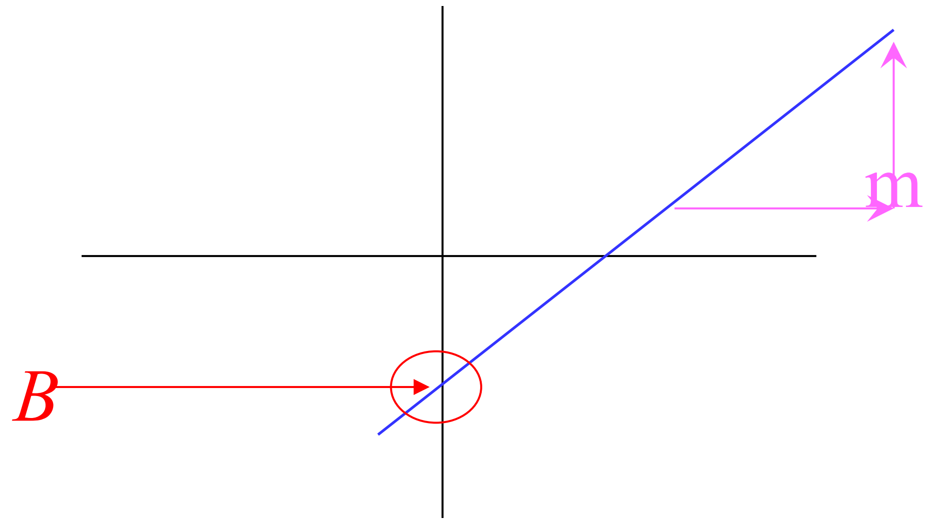
- LR use to explain the **relationship between one dependent variable and one or more independent variables**.

- Different between linear correlation and regression:
    - In correlation, the two variables **are treated as equals**.
    - In regression, one variable is considered independent (=predictor) variable (X) and the other the dependent (=outcome) variable Y.

What is "Linear"?

- Remember this:

  $Y=mX+B$?



What's Slope (*m*)?

A slope of 2 means that every 1-unit change in X yields a 2-unit change in Y.

Prediction : If you know something about X, this knowledge helps you predict something about Y. (sound like conditional probabilities?)

Regression equation…

- Expected value of y at a given level of *x*=

$$\boxed{E(y_i / x_i) = \alpha + \beta x_i}$$

- Predicted value for an individual…

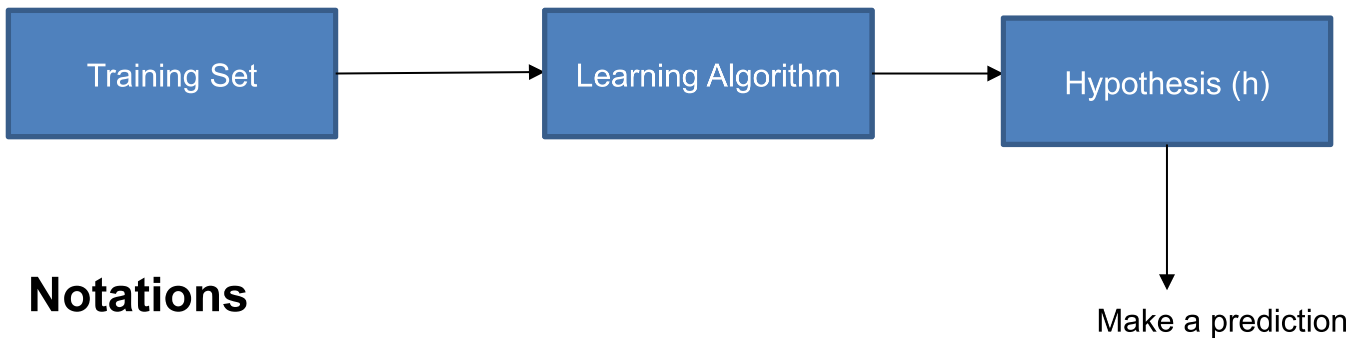
$$y_i = \underbrace{\hat{\alpha} + \beta * x_i}_{\text{Fixed – exactly on the line}} + \boxed{\text{random error}_i}$$

Follows a normal distribution

Linear regression with one variable

X	Y
0.4	11
0.56	14
0.6	15
0.8	22
1	26

Processing pipeline



Notations

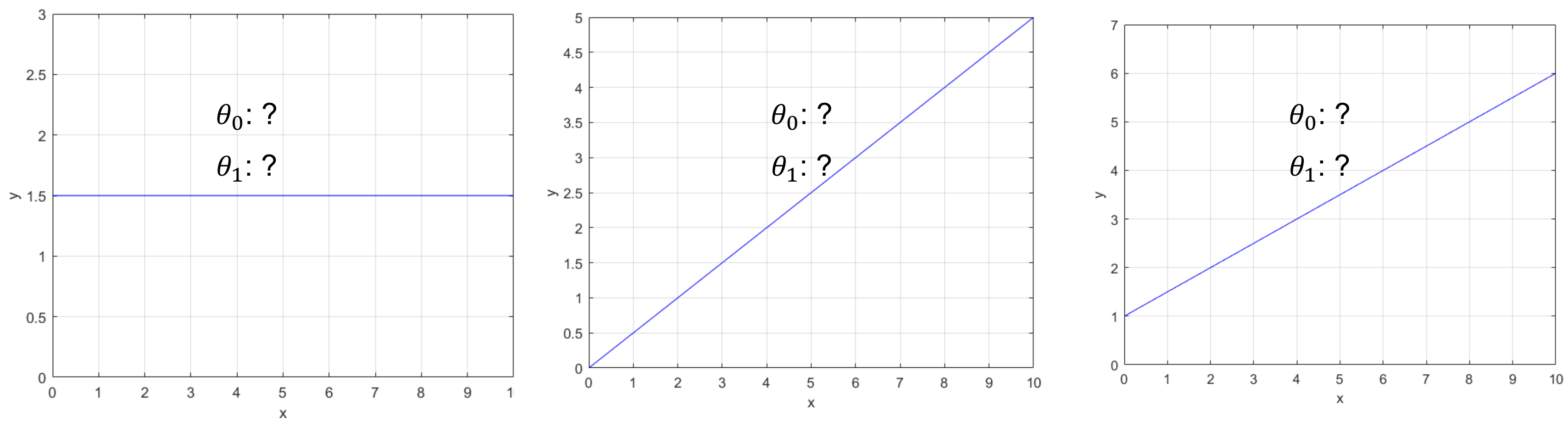
- M – Number of Training Examples
- x's – Input Variables / Features
- y's – Output variable / "target" variable
- h – Hypothesis "Proposal"

Each training example – (x,y)

'h' is a function that maps 'x' to 'y'

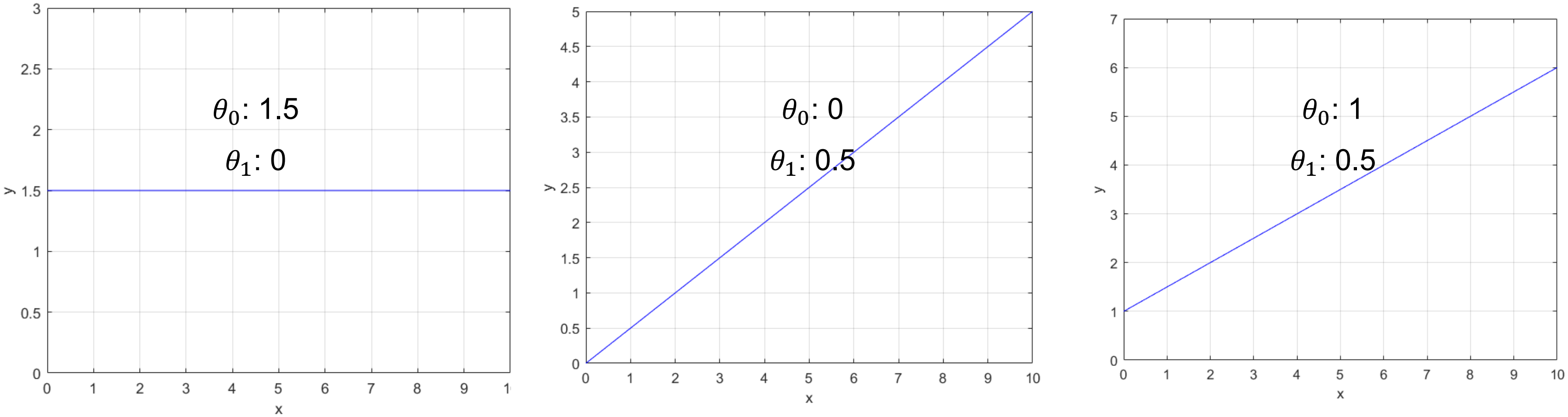
Linear Regression with one variable

- $h_\theta(x) = \theta_0 + \theta_1 x$



Linear Regression with one variable

- $h_\theta(x) = \theta_0 + \theta_1 x$



- We have to choose θ_0 and θ_1 such that $h_\theta(x) \cong y$
- Minimize $h_\theta(x) - y$ or $y - h_\theta(x)$
- $(h_\theta(x) - y)^2$is a good function to minimize – Loss Function

Linear regression with one variable

- We have to optimize the values across the entire dataset: i.e., **every sample needs to contribute to loss function**
- Cost function: J

$$J = minimize_{\theta_0,\theta_1} \sum_{i=1}^{M} \left(h_\theta\left(x^i\right) - y^i\right)^2$$

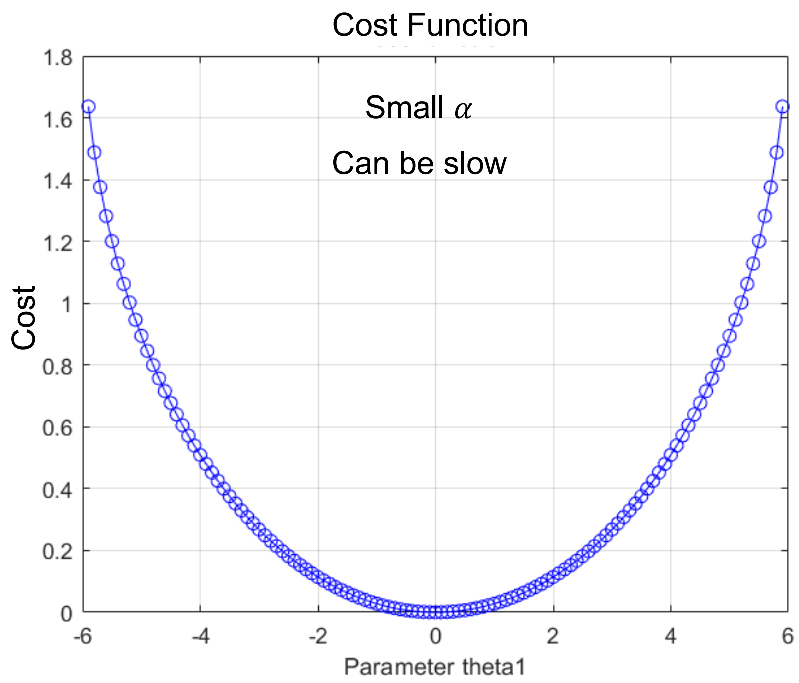
$$\Downarrow$$

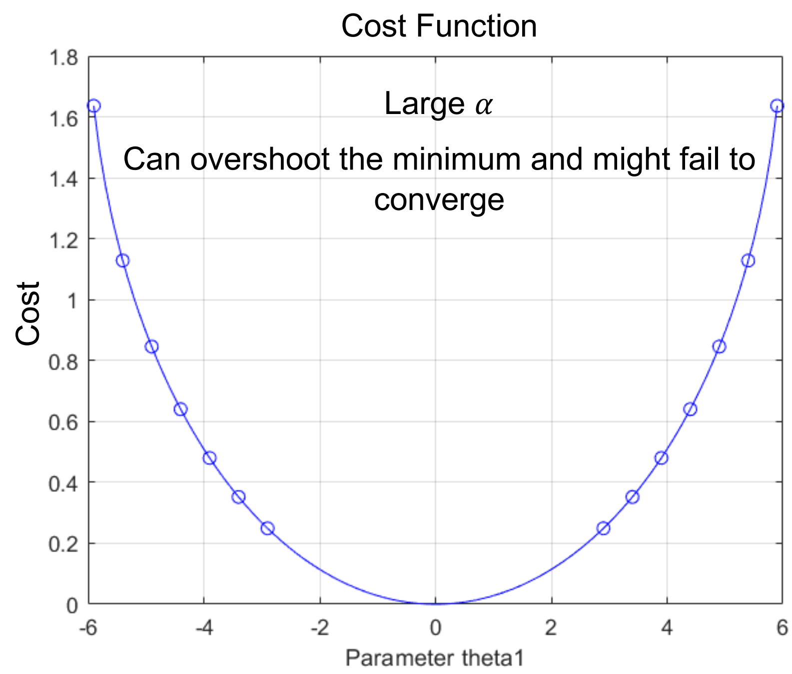
$$J = minimize_{\theta_0,\theta_1} \frac{1}{\mathbf{2}m} \sum_{i=1}^{M} \left(h_\theta\left(x^i\right) - y^i\right)^2$$

Gradient Descent

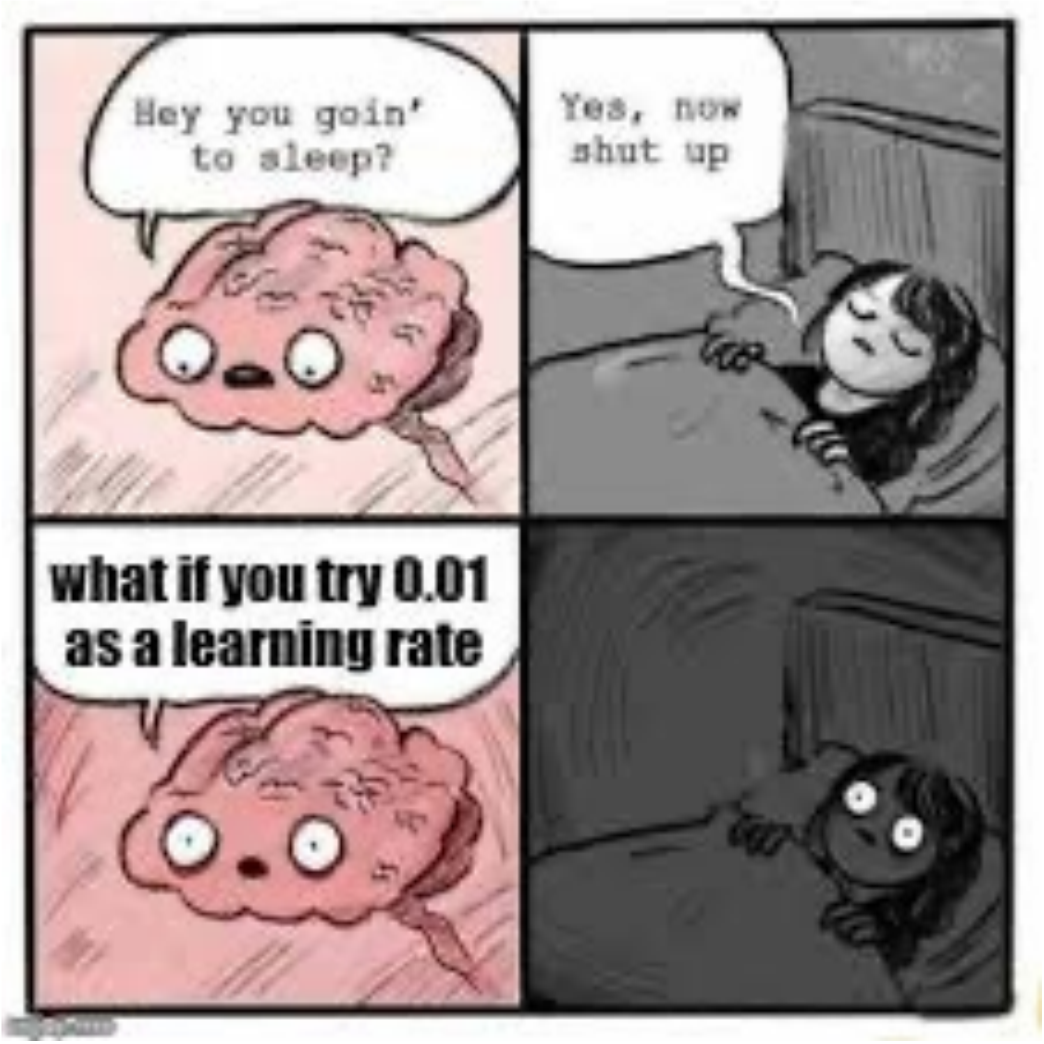
- Minimize cost function:
  - Start with a random
  - Keep updating θ_0 & θ_1 to reduce J
  - Find the global minimum
  - Find the values of θ_0, θ_1 at which J is minimum
  - Differentiation is the way to go!
    - » $\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} \boldsymbol{J}(\theta_0, \theta_1)$
  - α – Learning rate – Big steps (Large value), Baby Steps (Small value)

Learning rate example





Learning rate example



Update the parameters : θ_0, θ_1

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} \boldsymbol{J}(\theta_0, \theta_1)$$

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} \frac{1}{\mathbf{2}m} \sum_{i=1}^{M} \left(h_\theta\left(x^i\right) - y^i\right)^2$$

$$\theta_0 = \theta_0 - \alpha \frac{\partial}{\partial \theta_0} \frac{1}{\mathbf{2}m} \sum_{i=1}^{M} \left(h_\theta\left(x^i\right) - y^i\right)^2$$ and $$\theta_1 = \theta_1 - \alpha \frac{\partial}{\partial \theta_1} \frac{1}{\mathbf{2}m} \sum_{i=1}^{M} \left(h_\theta\left(x^i\right) - y^i\right)^2$$

$$\frac{\partial}{\partial \theta_o} \boldsymbol{J}(\theta_0, \theta_1) = \frac{\partial}{\partial \theta_0} \frac{1}{2m} \sum_{i=1}^{M} \left(\theta_0 + \theta_1 x^i - y^i\right)^2$$

$$\frac{\partial}{\partial \theta_1} \boldsymbol{J}(\theta_0, \theta_1) = \frac{\partial}{\partial \theta_1} \frac{1}{2m} \sum_{i=1}^{M} \left(\theta_0 + \theta_1 x^i - y^i\right)^2$$

Partial differentiation

$$\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{\partial}{\partial \theta_0} \frac{1}{2m} \sum_{i=1}^{m} \left(\theta_0 + \theta_1 x^i - y^i\right)^2$$

$$= \frac{\partial}{\partial \theta_0} \frac{1}{2m} \sum_{i=1}^{m} \theta_0^2 + \theta_1^2 x^{i^2} + 2\theta_0\theta_1 x^i + y^{i^2} - 2(\theta_0 + \theta_1 x^i)\, y^{(i)}$$

$$= \frac{1}{\cancel{2}m} \sum_{i=1}^{m} \cancel{2}\theta_0 + 0 + \cancel{2}\theta_1 x^i + 0 - \cancel{2} y^i$$

$$\frac{\partial J(\theta_0, \theta_1)}{\partial \theta_0} = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^i\right)$$

$$\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{\partial}{\partial \theta_1} \frac{1}{2m} \sum_{i=1}^{m} \theta_0^2 + \theta_1^2 x^{i^2} + 2\theta_0\theta_1 x^i + y^{i^2} - 2(\theta_0 + \theta_1 x^i)\, y^{(i)}$$

$$= \frac{1}{\cancel{2}m} \sum_{i=1}^{m} 0 + \cancel{2}\theta_1 x^{(i)2} + \cancel{2}\theta_0 x^{(i)} + 0 - \cancel{2} x^{(i)} y^{(i)}$$

$$= \frac{1}{m} \sum_{i=1}^{m} \left(\theta_0 + \theta_1 x^{(i)} - y^{(i)}\right) x^{(i)}$$

$$\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right) x^{(i)}$$

Update the parameters : θ_0, θ_1

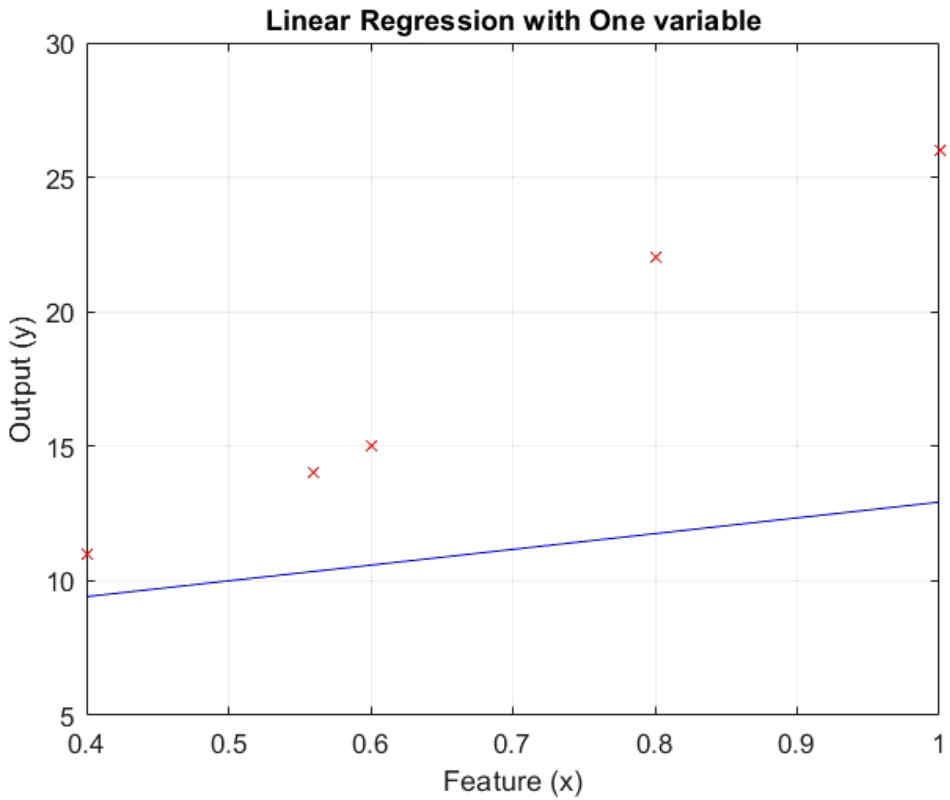
$$\frac{\partial}{\partial\theta_o}\boldsymbol{J}(\theta_0, \theta_1) = \frac{1}{m}\sum_{i=1}^{M}\left(h_\theta\left(x^i\right) - y^i\right),$$

$$\frac{\partial}{\partial\theta_1}\boldsymbol{J}(\theta_0, \theta_1) = \frac{1}{m}\sum_{i=1}^{M}\left(h_\theta\left(x^i\right) - y^i\right) x^i$$

$$\theta_0 = \theta_0 - \alpha\frac{1}{m}\sum_{i=1}^{M}\left(h_\theta\left(x^i\right) - y^i\right), \qquad \theta_1 = \theta_1 - \alpha\frac{1}{m}\sum_{i=1}^{M}\left(h_\theta\left(x^i\right) - y^i\right) x^i$$

Repeat this process for ‘N’ iterations

Linear Regression Demo! (Video)



Feature engineering?

Feature engineering

- Size in square feet
- # Bedrooms
- # Restrooms
- School district
- School quality
- Crime rate
- Highway accessibility

And so on…

We can engineer ‘n’ number of features and so on.

Linear regression with multiple variables

$$h_\theta(x) = \theta_0 + \theta_1 x_1$$

$$\downarrow$$

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 \ldots \ldots \ldots \theta_n x_n$$

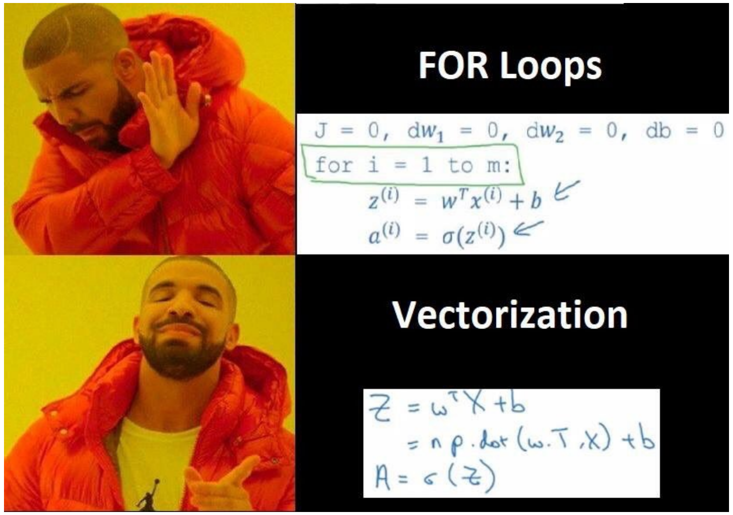
- It doesn't make sense to type out this equation manually for every variable,
- You'll have to **modify the equation every time a feature is added** especially in problems where <u>you</u> are engineering features
- Moreover, the **gradient descent function would also** vary as you add more variables

Linear regression with multiple variables

$$h_\theta(\mathrm{x}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 \ldots\ldots\ldots \theta_n x_n$$

Let's resort to vectorization:

$$h_\theta(\mathrm{x}) = \theta x^T$$



- However, size of θ is (1 x n+1) and x is (M x n)
- Note that, **x is a feature matrix** instead of an array

Hypothesis

- Let's introduce x_0 whose value is 1 to make sure the dimensions match – [x_0 = 1 for all samples present in the dataset]

$$h_\theta(\mathrm{x}) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 \ldots \ldots \ldots \theta_n x_n$$

- Now, size of $\boldsymbol{\theta}$ **is (1 x n+1)** and **x is (M x n+1)**
- $h_\theta(\mathrm{x}) = \theta x^T$ should be possible now

- $h_\theta(\mathrm{x})$ - Size would be **(1 x M)**

Hypothesis can be computed for multiple variables now!

Cost function

- $J = \frac{1}{\mathbf{2}m} \sum_{i=1}^{M} \left(h_\theta(x^i) - y^i \right)^2$ - from previous slide
- y is **1 x M** and $h_\theta(x^i)$ is **1 x M**

Do we need to make any changes to this equation?

- No need of for-loops for cost calculation

Gradient Descent

From previous slides,

- $\theta_0 = \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{M} \left(h_\theta\left(x^i\right) - y^i \right)$
- $\theta_1 = \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{M} \left(h_\theta\left(x^i\right) - y^i \right) x^i$

Is there any generic way we can write such that we can update all weights simultaneously?

Gradient Descent

- However, now we have until θ_n and x is (M x N+1)
- Instead of updating parameters one by one, it's good to update them all simultaneously

- $\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} \boldsymbol{J}(\theta)$
- $\delta = \frac{\partial}{\partial \boldsymbol{\theta}} \boldsymbol{J}(\theta)$

Vectorized implementation of gradient descent

- Generic formula for simultaneously updating all θ_j
- $\delta = \frac{1}{m}\sum_{i=1}^{M}\left(h_\theta\left(x^i\right) - y^i\right) x^i$

Notes for programming:

- Dimensions of $(h_\theta(x) - y)$ - (1 x M)
- Dimensions of x – (M x n+1)
- $\delta = (h_\theta(x) - y)\ x$ => Note that, it is matrix multiplication
- Dimensions of δ - (1 x n+1)

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \boldsymbol{\alpha\delta}$$

Gradient descent for multiple variables is complete now!

One extra step to be completed!

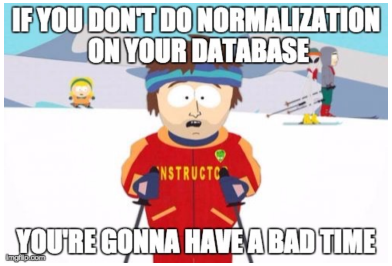
- x_1 - Size in square feet – **Range of values: (100-10,000)**
- x_2 - # Bedrooms – **Range of values: (0-6)**
- x_3 - # Restrooms – **Range of values: (0-5)**
- x_4 - School quality – **Range of values: (1-100)**

- Indifferent range for every feature – Make it forever to converge or even might make it biased to certain features
- It’s important to normalize the data

Data normalization

- There are multiple ways to normalize the data
- Let's adopt to the one where we can zero center the values

$$\hat{x} = \frac{x \; - \mu}{\sigma}$$



μ – mean for each feature, σ – standard deviation for each feature

- No need of for-loops for this as well.

Note that, please do this normalization step before adding $\boldsymbol{x_0}$

Steps for linear regression with multiple variables

- Read the dataset : (Input features and Output values)
- Normalize features
  - $\hat{x} = \frac{x - \mu}{\sigma}$
  - Add $\boldsymbol{x_0} = 1$ to all samples (Normalized $\hat{x}$)
  - new Dimensions of $\boldsymbol{x}$ – M x (n+1)
- Implement cost function after computing hypothesis
  - $h_\theta(\mathrm{x}) = \theta x^T$
  - $J = \frac{1}{2m} \sum_{i=1}^{M} \left(h_\theta\left(x^i\right) - y^i\right)^2$
- Implement gradient descent function for multiple variables
  - $\boldsymbol{\theta} = \theta - \alpha\delta$
  - $\delta = \frac{1}{m} \sum_{i=1}^{M} \left(h_\theta\left(x^i\right) - y^i\right) x^i$

No need any for-loops other than one for iteration

Assignment 1 for practice at home

- Compute cost Function
  - Determine the number of samples
  - Compute $h_\theta(x) = \theta_0 + \theta_1 x$
  - Compute J = $\frac{1}{2m}\sum_{i=1}^{M}\left(h_\theta\left(x^i\right) - y^i\right)^2$

- Gradient Descent Function
  - Determine the number of samples
  - Loop for iterations
  - Compute $h_\theta(x) = \theta_0 + \theta_1 x$
  - Compute cost using your own “compute cost” function
  - Calculate updated θ_0 and θ_1
  - Repeat for all iterations!

Summary

- Linear correlation
- Linear regression with and single and multiple variables
- What’s next ?
  - Logistic Regression (LR)
  - Variants…