→ Learning Rate (α)

- too small: gradient descent is too slow
- too big: "big steps", may lead to overshooting and failure to converge

→ Gradient Descent for Linear Regression

model
$$\begin{cases} h_\theta(x) = \theta_0 + \theta_1 x \\ J(\theta_0,\theta_1) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 \end{cases}$$
→ apply GD to get lowest cost parameters

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

→ $j=0$ → $\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})$

→ $j=1$ → $\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)}) \cdot x^{(i)}$

$$\begin{cases} \theta_0 := \theta_0 - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)}) \\ \theta_1 := \theta_1 - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)} - y^{(i)})x^{(i)} \end{cases}$$

* "Batch" Gradient Descent: each iteration goes through the entire training set

Week 2

→ Multivariate Linear Regression

- X_n feature contribute to output value Y
- Notation: $x^{(i)}$ is vector containing all features in i-eth example
- Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n$
  - → $x_0 = 1$ → $h_\theta(x) = \sum_{i=0}^{n} x_i\theta_i = \theta^T x$
    - vector with all θ_i
    - → vector with all x_s
- Gradient Descent:
  $$\theta_j := \theta_j - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

→ Feature Scaling

- Features being on a similar scale can make gradient descent work faster
- get every feature in range $[-1, 1]$ → divide by maximum value
- Mean Normalization: $x_i = \frac{x_i - \mu_i}{\text{range}}$ → average value

→ Convergence method

- establish a threshold E and if $J(\theta)$ decreases by less than E in one iteration it has converged * hard to determine E
- plot nº iterations x $J(\theta)$ → helpful when choosing α

* It's possible to create new features that enable the fitting of polynomial functions to the data

Normal Equations

- Solves for optimum value directly
- Take derivative and equal to zero
- Defining X as a matrix containing all features and Y as the solution ve

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

- No need to choose α or to iterate
- Does not scale well to a large number of features