Gradient Descent in Practice I: Feature Scaling

LR with Multiple Variables

Feature Scaling

- We can speed up gradient descent by having each of our input values in roughly the same range.
- This is because θ will descend
  - quickly on small ranges and
  - slowly on large ranges,
- So it will oscillate inefficiently down to the optimum when the variables are very uneven.

Feature Scaling

Idea: Make sure features are on a similar scale.

E.g. x_1 = size (0-2000 feet^2)

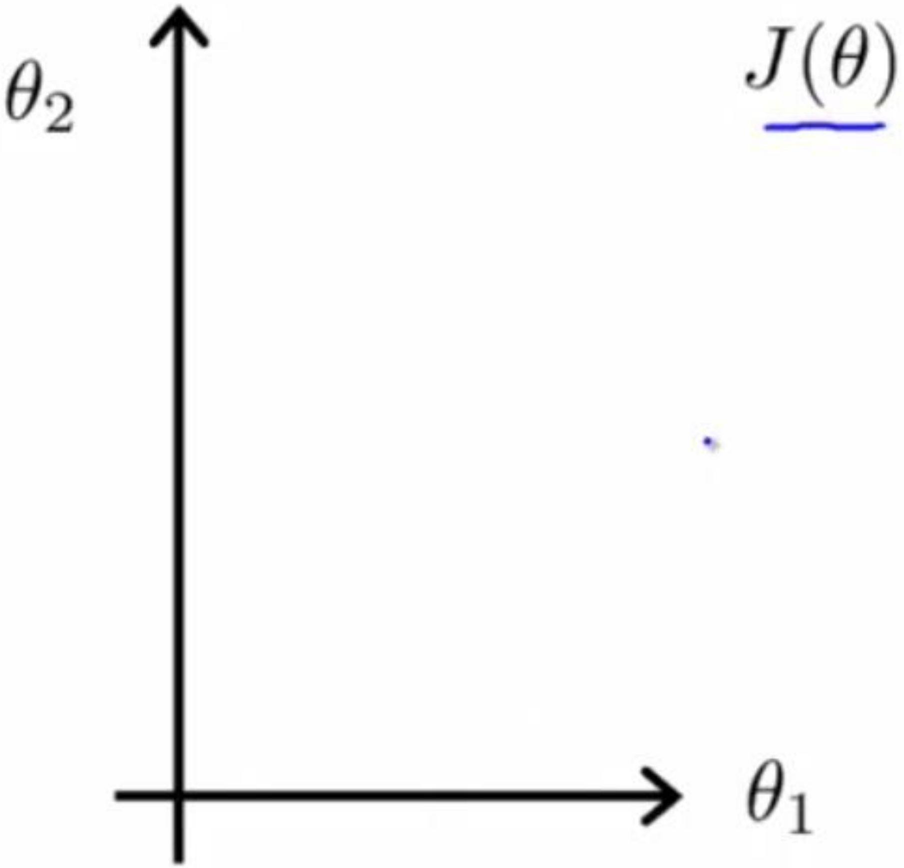
x_2 = number of bedrooms (1-5)

Feature Scaling

Idea: Make sure features are on a similar scale.

E.g. x_1 = size (0-2000 feet^2)

x_2 = number of bedrooms (1-5)

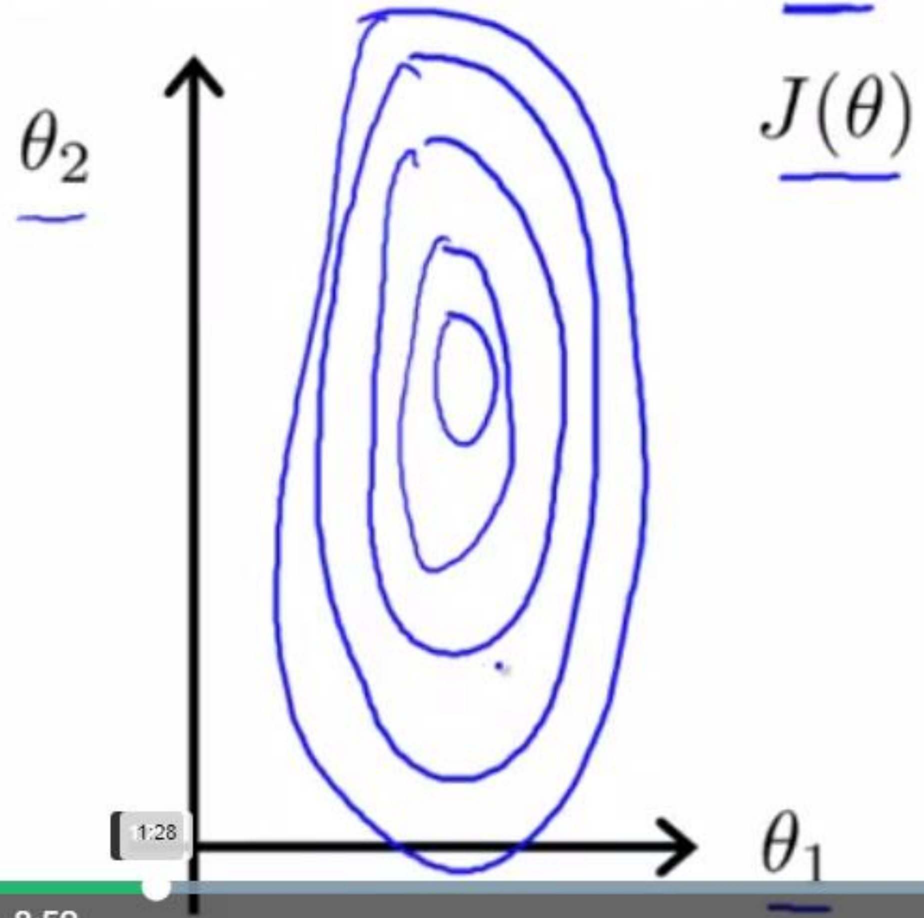


Feature Scaling

Idea: Make sure features are on a similar scale.

E.g. x_1 = size (0-2000 feet^2) $\leftarrow$

x_2 = number of bedrooms (1-5) $\leftarrow$

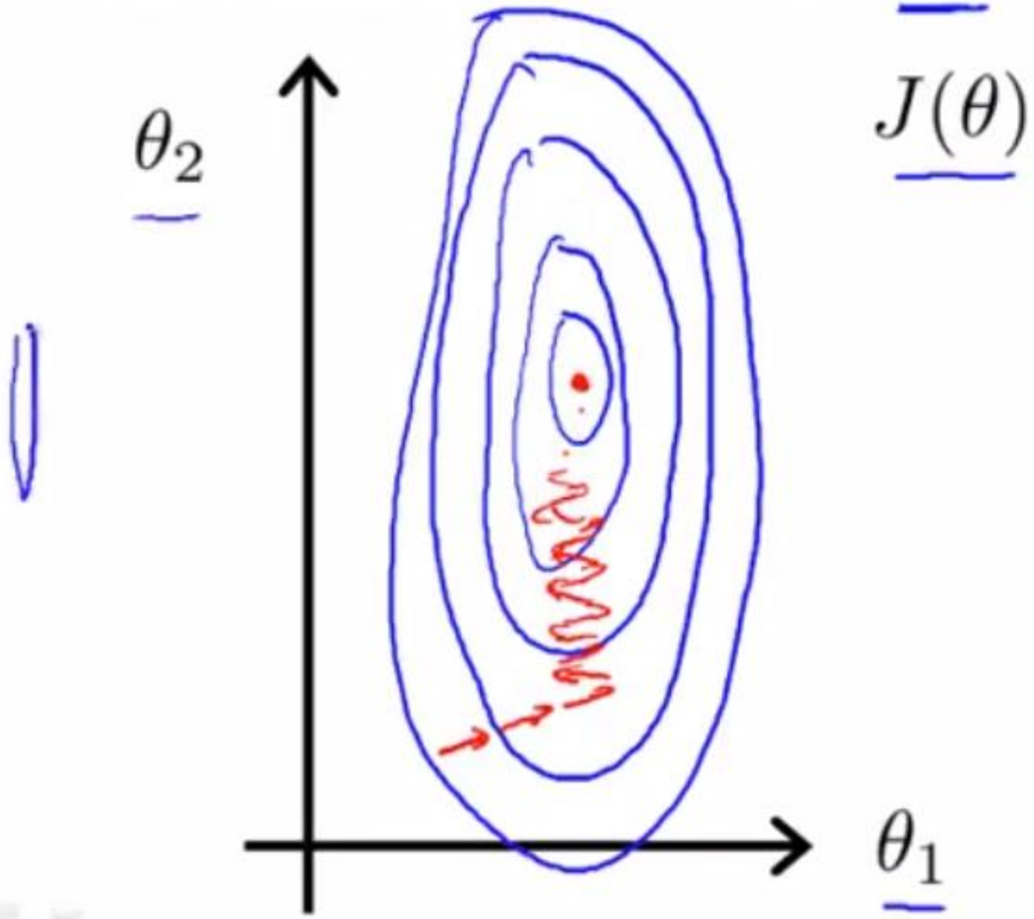


Feature Scaling

Idea: Make sure features are on a similar scale.

E.g. x_1 = size (0-2000 feet^2)

x_2 = number of bedrooms (1-5)



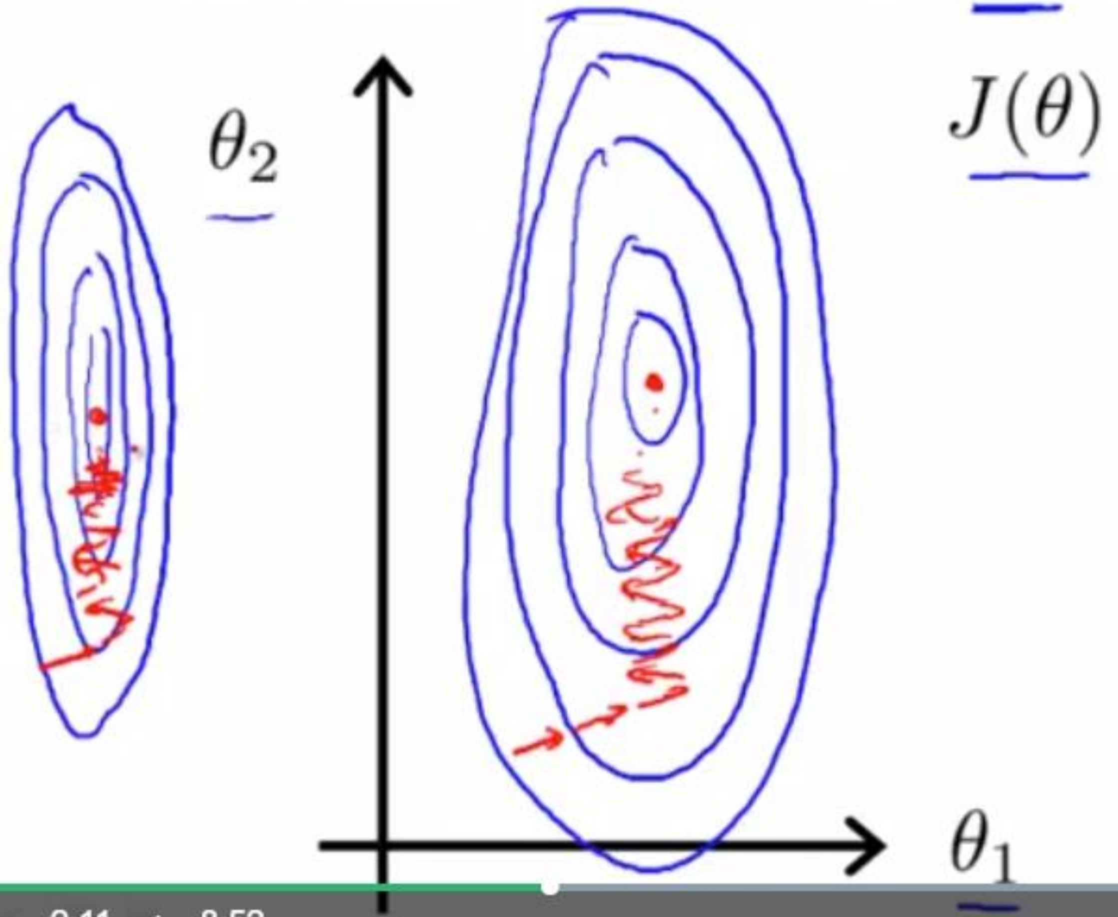
IT take a long time to find the optimal point with skinny contours.

Feature Scaling

Idea: Make sure features are on a similar scale.

E.g. x_1 = size (0-2000 feet^2)

x_2 = number of bedrooms (1-5)



Feature Scaling

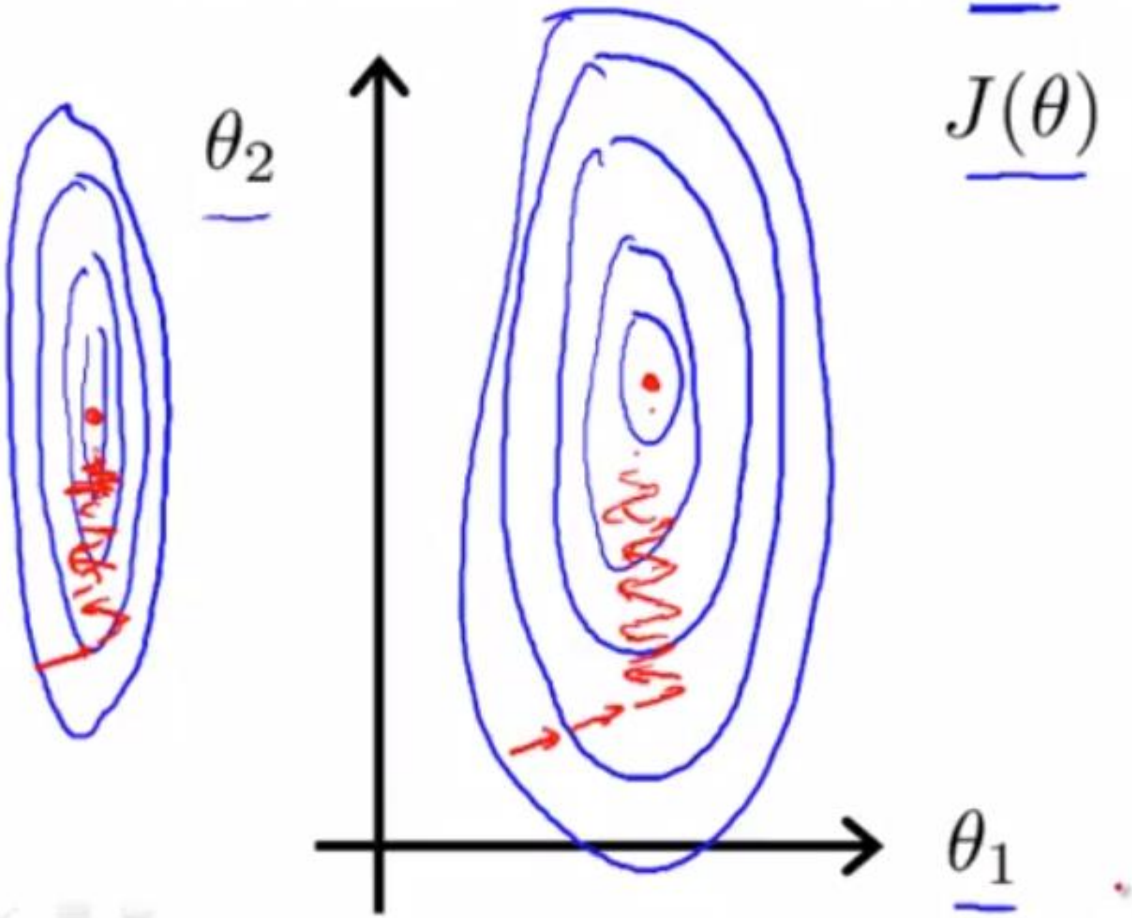
Idea: Make sure features are on a similar scale.

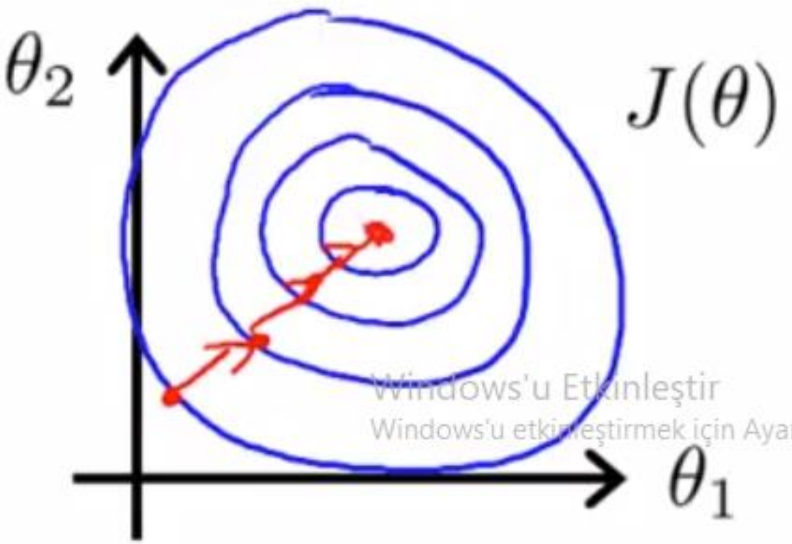
E.g. x_1 = size (0-2000 feet^2)

x_2 = number of bedrooms (1-5)

$$\rightarrow x_1 = \frac{\text{size (feet}^2\text{)}}{2000}$$

$$\rightarrow x_2 = \frac{\text{number of bedrooms}}{5}$$





Feature Scaling

Get every feature into approximately a $-1 \leq x_i \leq 1$ range.

Feature Scaling

Get every feature into approximately a $-1 \le x_i \le 1$ range.

$x_0 = 1$

$0 \le x_1 \le 3$ ✓

$-2 \le x_2 \le 0.5$ ✓

$-100 \le x_3$ 100 ✗

$-0.0001 \le x_4 \le 0.0001$ ✗

Feature Scaling

Get every feature into approximately a $-1 \le x_i \le 1$ range.

$x_0 = 1$

$0 \le x_1 \le 3$ ✓

$-2 \le x_2 \le 0.5$ ✓

$-100 \le x_3 \; \boxed{100}$ ✗

$-0.0001 \le x_4 \le \boxed{0.0001}$ ✗

-3 to 3 ✓

$-\frac{1}{3}$ to $\frac{1}{3}$ ✓

Mean normalization

Replace x_i with $x_i - \mu_i$ to make features have approximately zero mean (Do not apply to $x_0 = 1$).

E.g. $x_1 = \frac{size - 1000}{2000}$

$x_2 = \frac{\#bedrooms - 2}{5}$

$$-0.5 \le x_1 \le 0.5, -0.5 \le x_2 \le 0.5$$

Mean normalization

Replace x_i with $x_i - \mu_i$ to make features have approximately zero mean
(Do not apply to $x_0 = 1$).

E.g. $\boxed{x_1 = \frac{size-1000}{2000}}$

Average size = 100

$\boxed{x_2 = \frac{\#bedrooms-2}{5}}$

1-5 bedrooms

$$-0.5 \le x_1 \le 0.5, \quad -0.5 \le x_2 \le 0.5$$

Mean normalization

Replace x_i with $x_i - \mu_i$ to make features have approximately zero mean (Do not apply to $x_0 = 1$).

E.g. $\boxed{x_1 = \frac{size - 1000}{2000}}$

Average size = 100

$\boxed{x_2 = \frac{\#bedrooms - 2}{5}}$

1-5 bedrooms

$$-0.5 \leq x_1 \leq 0.5, \; -0.5 \leq x_2 \leq 0.5$$

$$x_1 \leftarrow \frac{x_1 - \mu_1}{s_1}$$

avg value of x_1 in training set

range (max - min)

(or standard deviation)

Mean normalization

Replace x_i with $x_i - \mu_i$ to make features have approximately zero mean
(Do not apply to $x_0 = 1$).

E.g. $\boxed{x_1 = \frac{size - 1000}{2000}}$

Average size = 100

$\boxed{x_2 = \frac{\#bedrooms - 2}{5}}$

1-5 bedrooms

$\boxed{-0.5 \le x_1 \le 0.5}$, $\boxed{-0.5 \le x_2 \le 0.5}$

$$x_1 \leftarrow \frac{x_1 - \mu_1}{s_1}$$

avg value of x_1 in training set

range (max - min)
(or standard deviation)

$$x_2 \leftarrow \frac{x_2 - \mu_1}{s_2}$$

Exercise

- Suppose you are using a learning algorithm to estimate the price of houses in a city. You want one of your features x_i to capture the age of the house.
- In your training set, all of your houses have an age between 30 and 50 years, with an average age of 38 years.
- How do you normalize your data using mean normalization?