Gradient Descent

Week 05 - Day 03

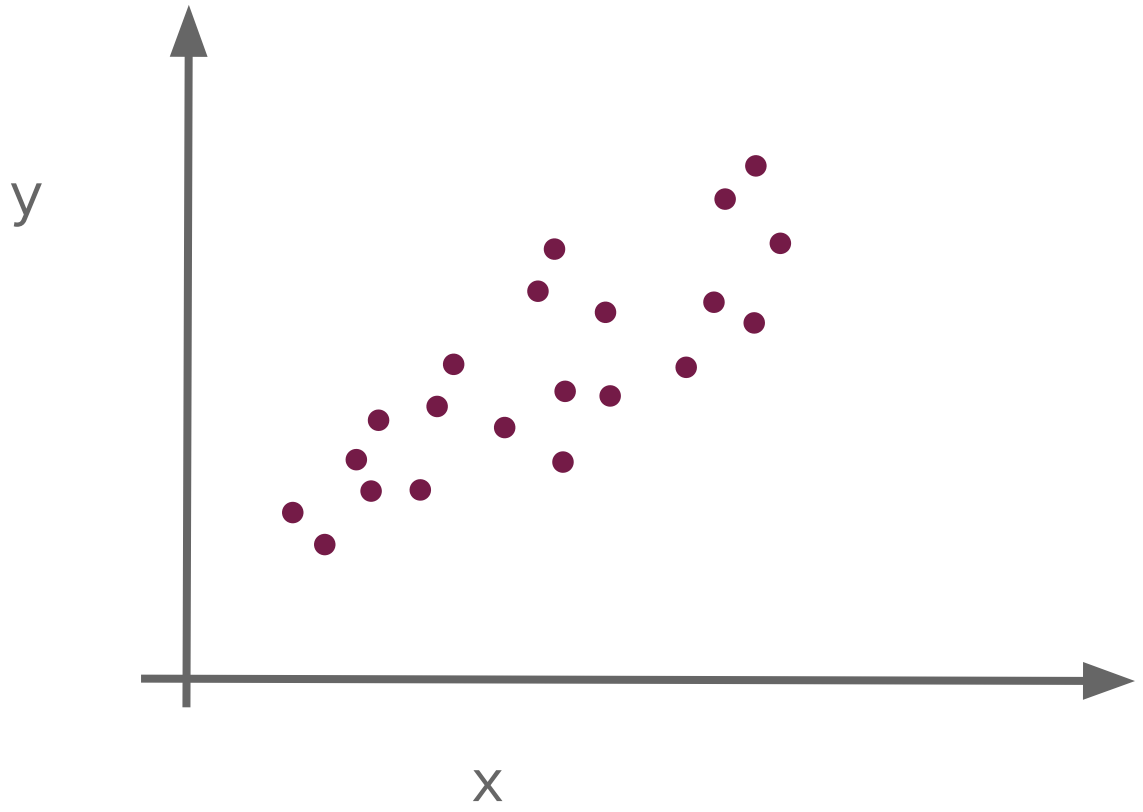
A general algorithm to optimize a loss function in ML

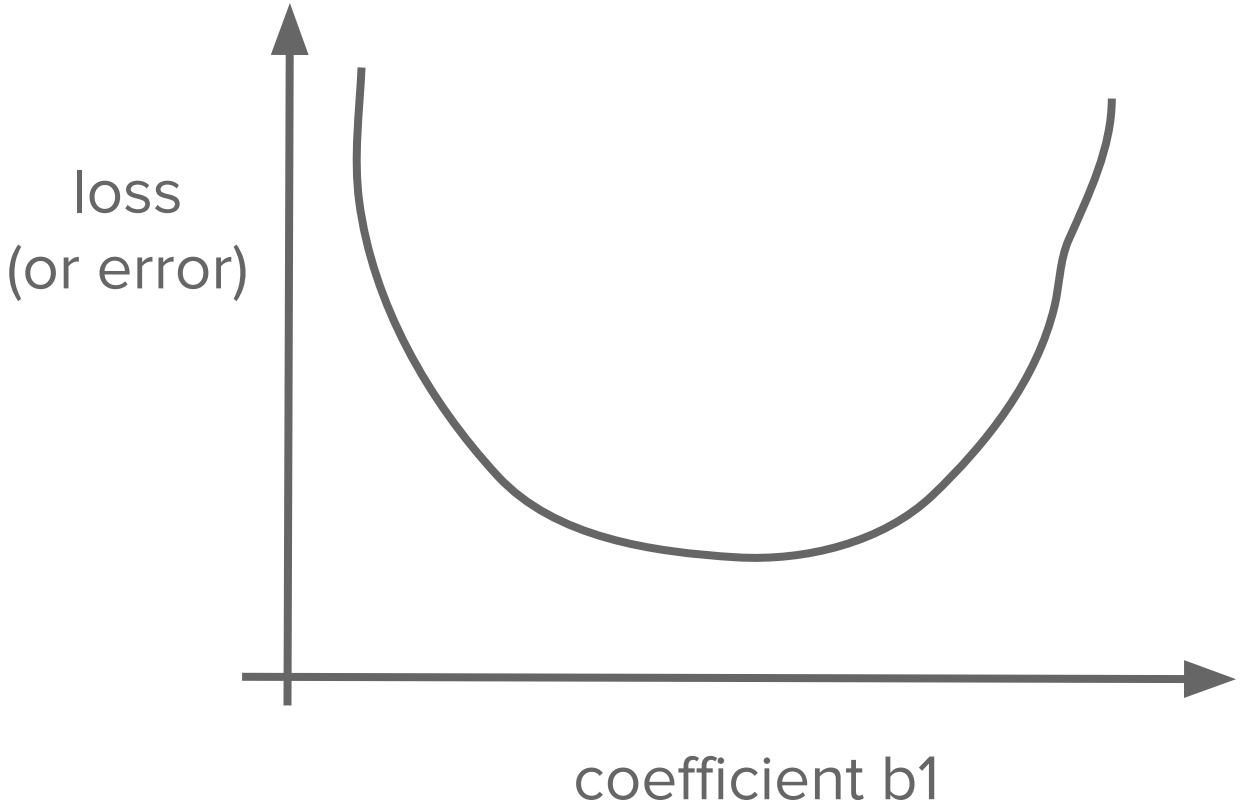
You’ll never manually use it but...

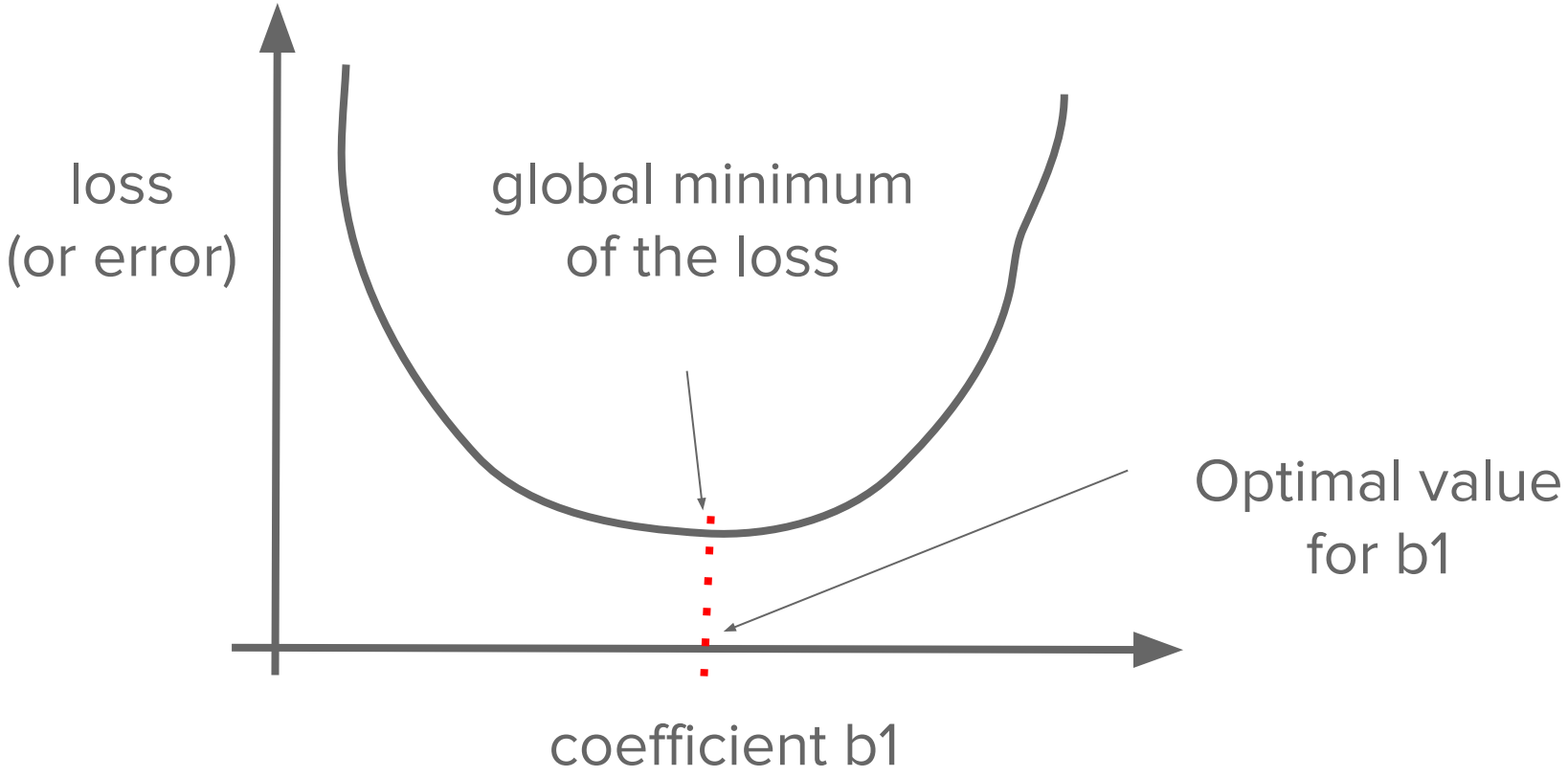
It’s a common interview question!

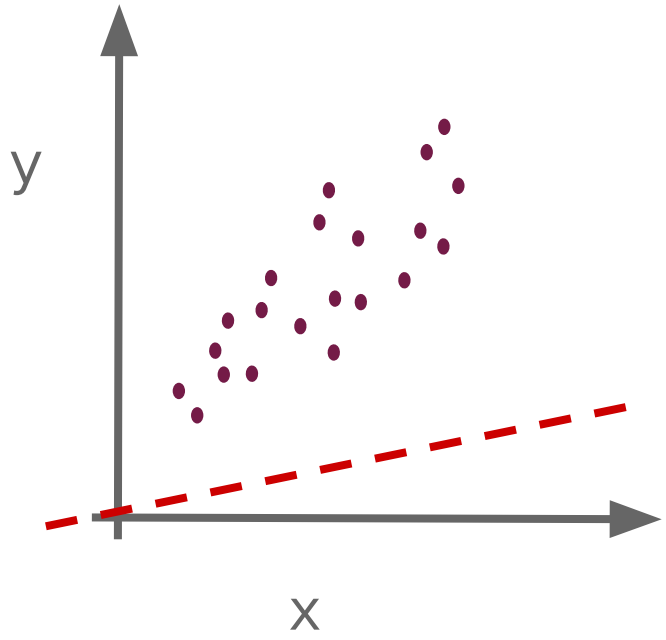
Optimizing the loss function

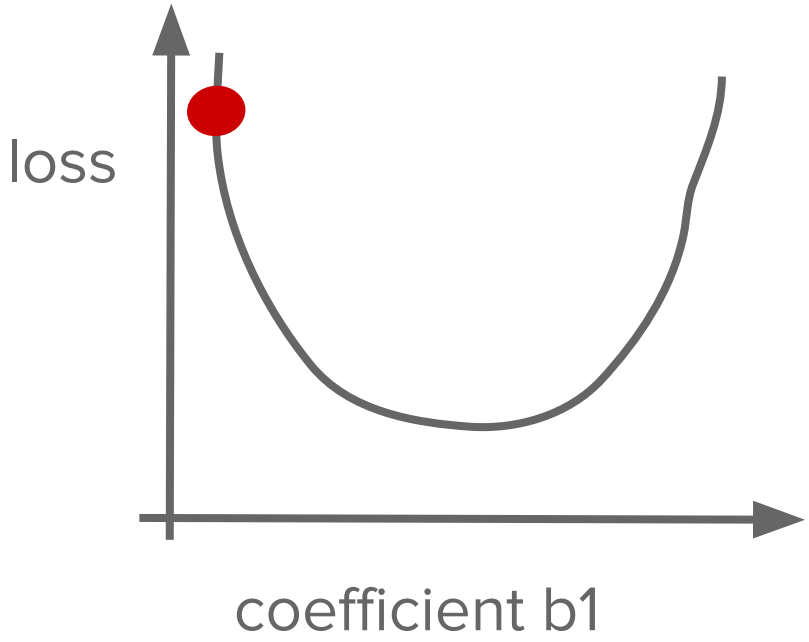
Linear regression: y = b1*x + e

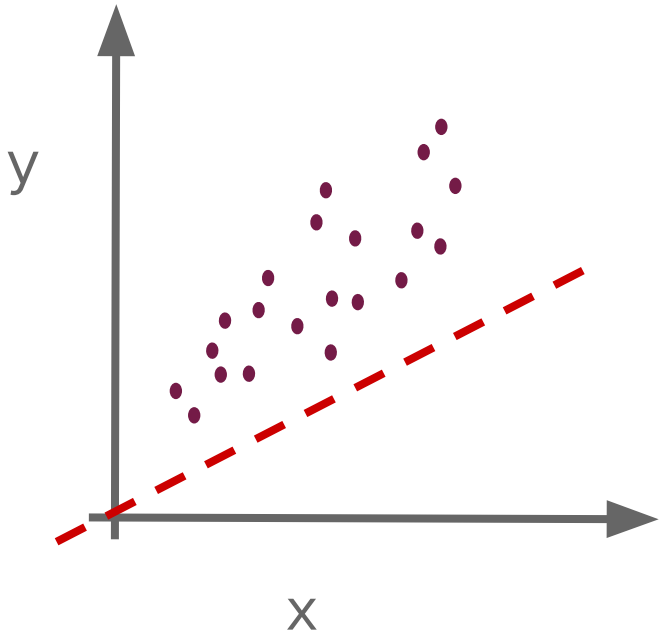


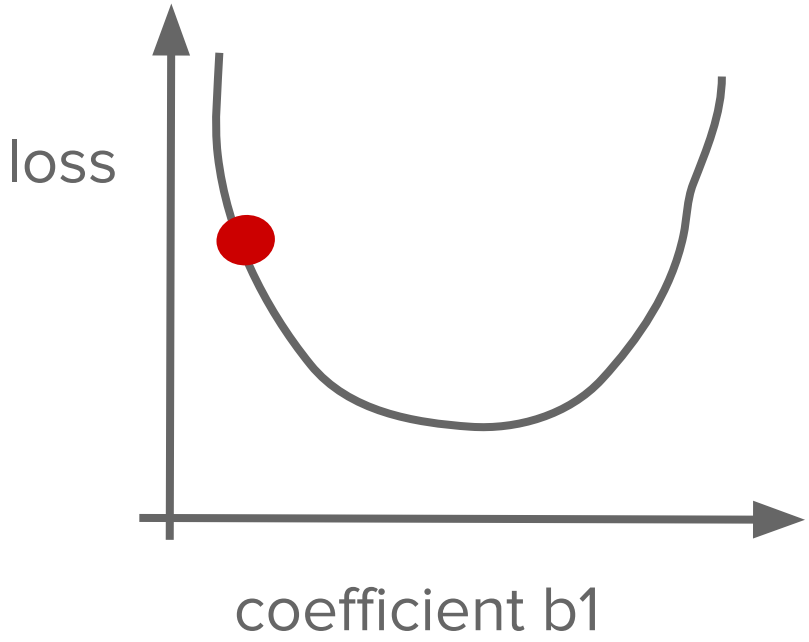

loss
(or error)
coefficient b1

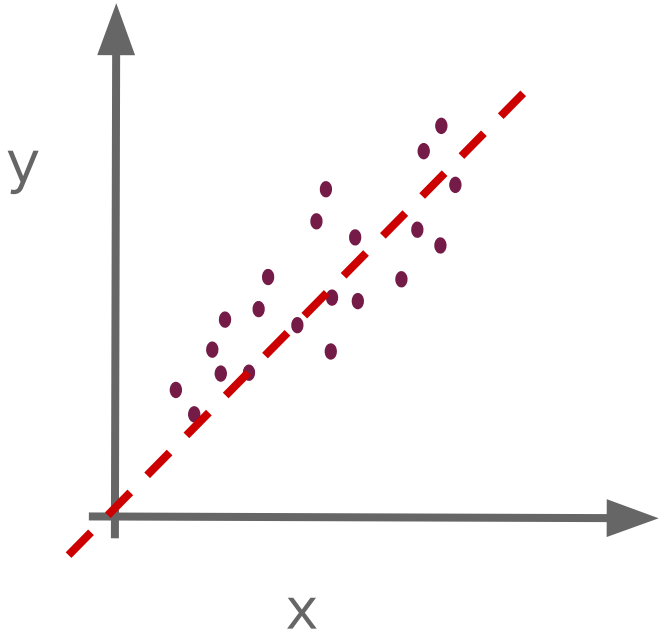

loss
(or error)
global minimum
of the loss
Optimal value
for b1
coefficient b1

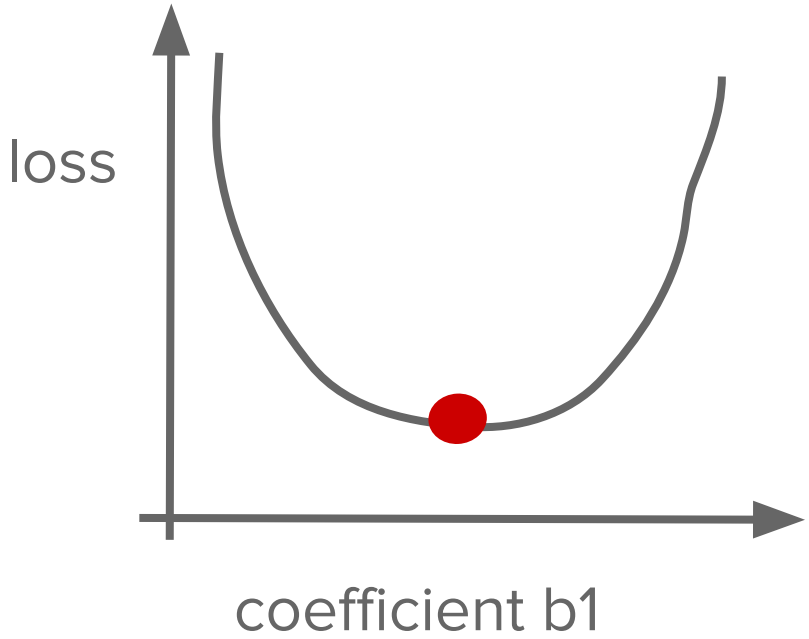

y
x

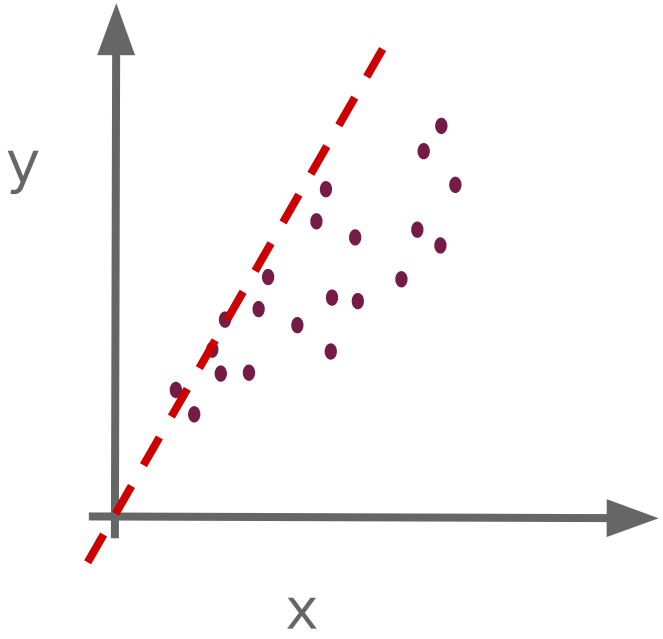

loss
coefficient b1

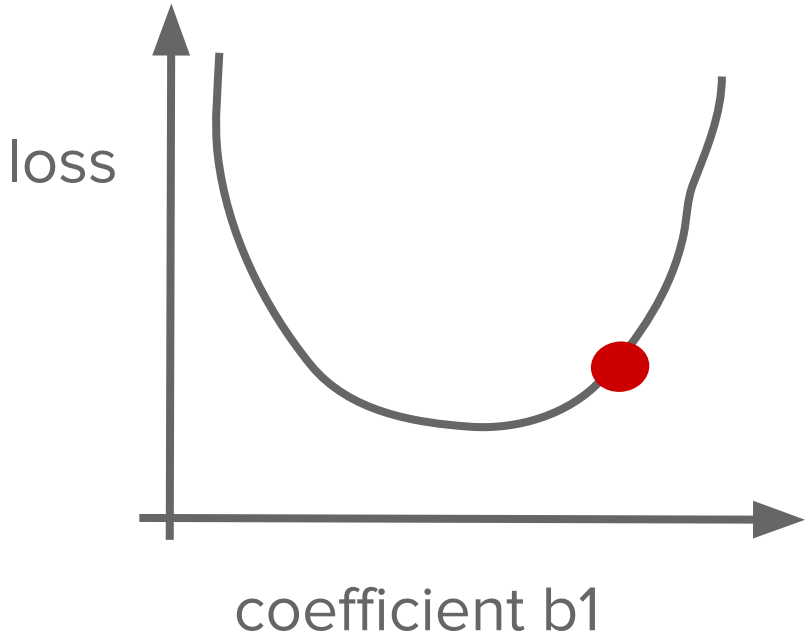

y
x

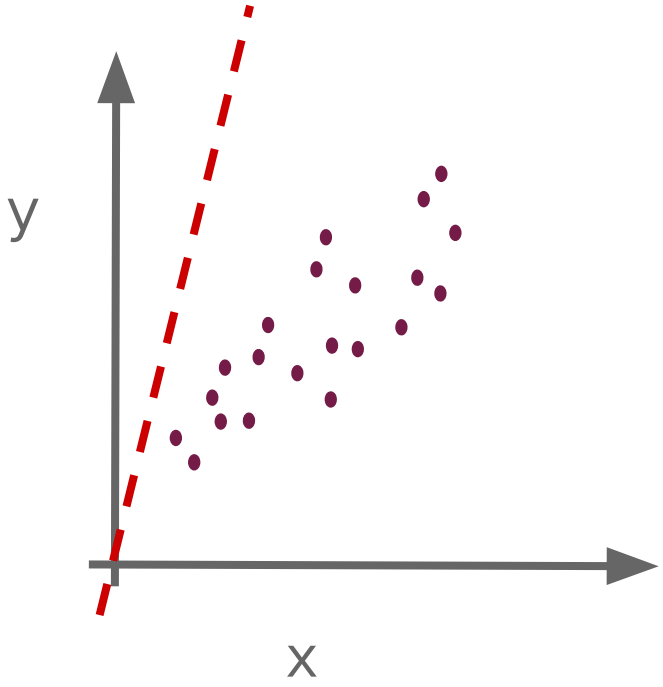

loss
coefficient b1

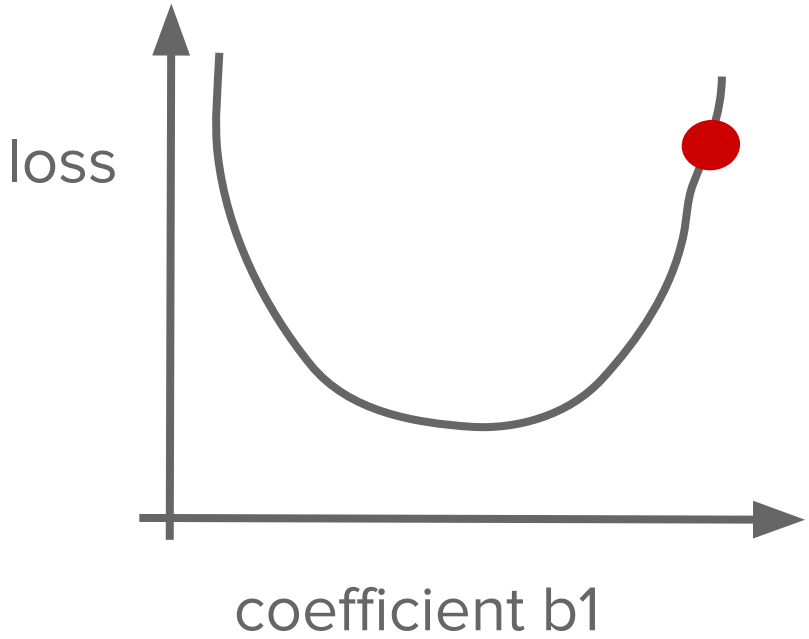

y
x


loss
coefficient b1


y
x


loss
coefficient b1


y
x


loss
coefficient b1

We used calculus to find the best point

derivative(loss_function)=0

Gradient Descent

A general optimization algorithm to find the best parameters

(i.e. the smallest error)

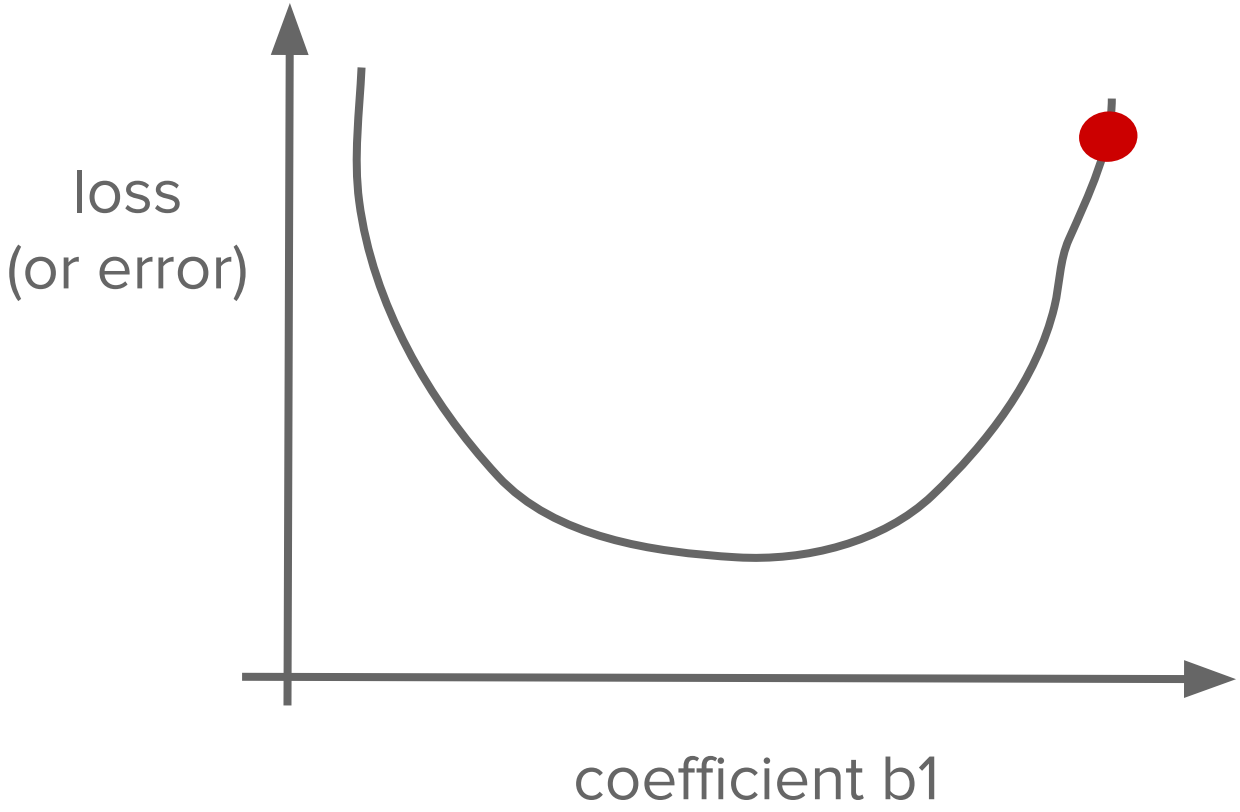
Iterative approach

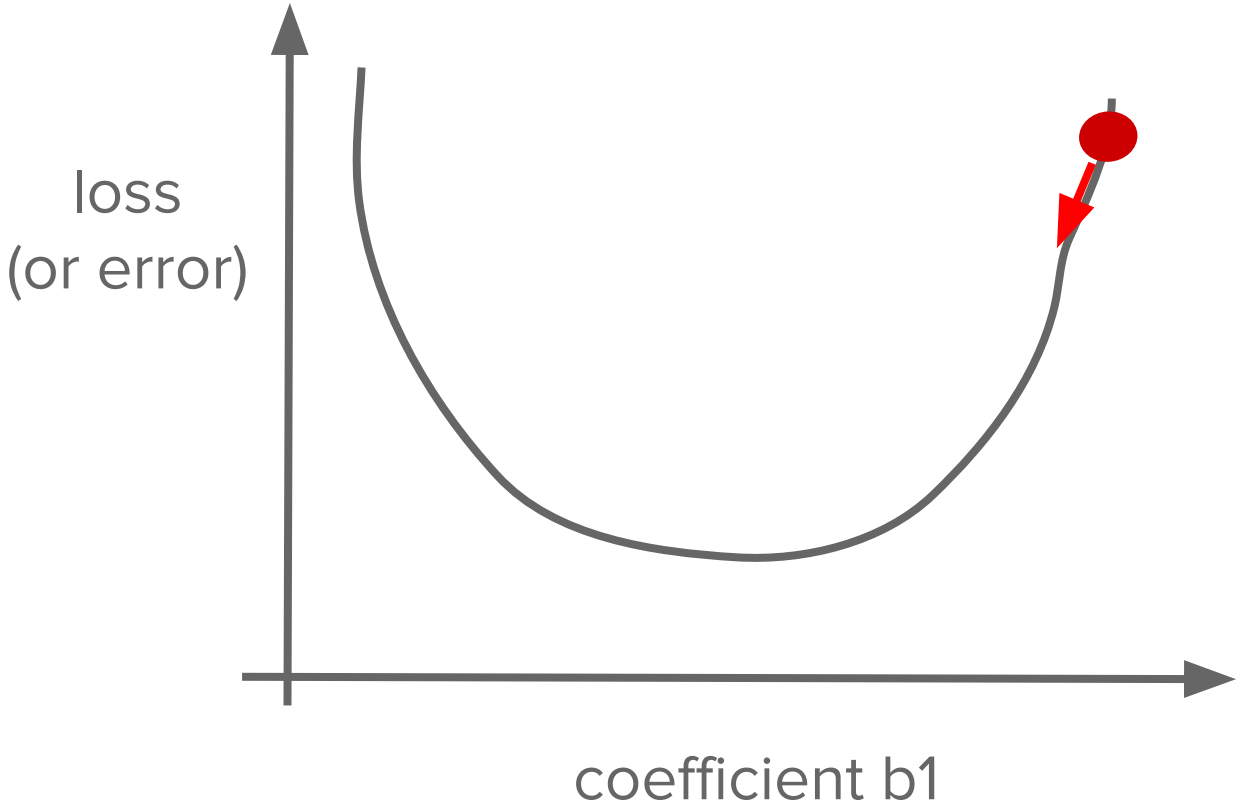
1) Start with random solution (b0,b1)

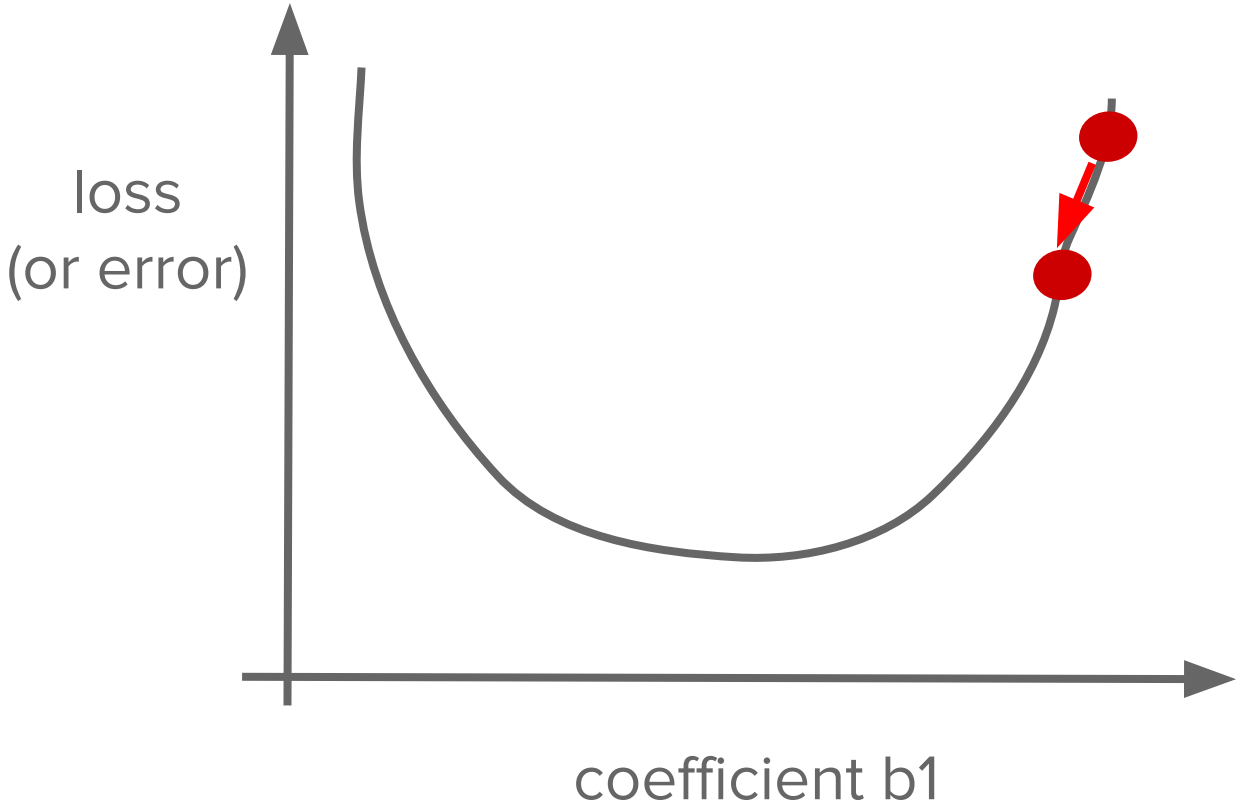
1) Start with random solution (b0,b1)
2) Find the right direction to get to a smaller error

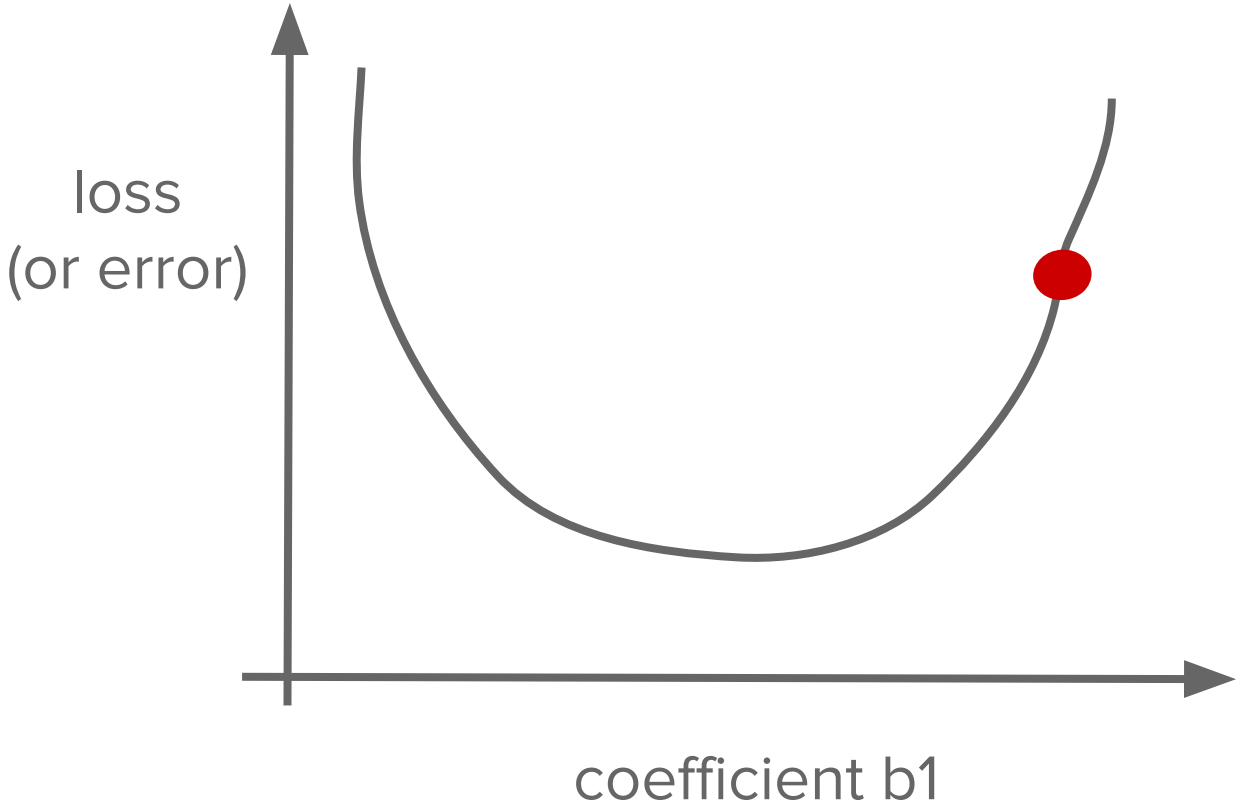
1) Start with random solution (b0,b1)
2) Find the right direction to get to a smaller error
3) Get a new better solution (b1, b2)

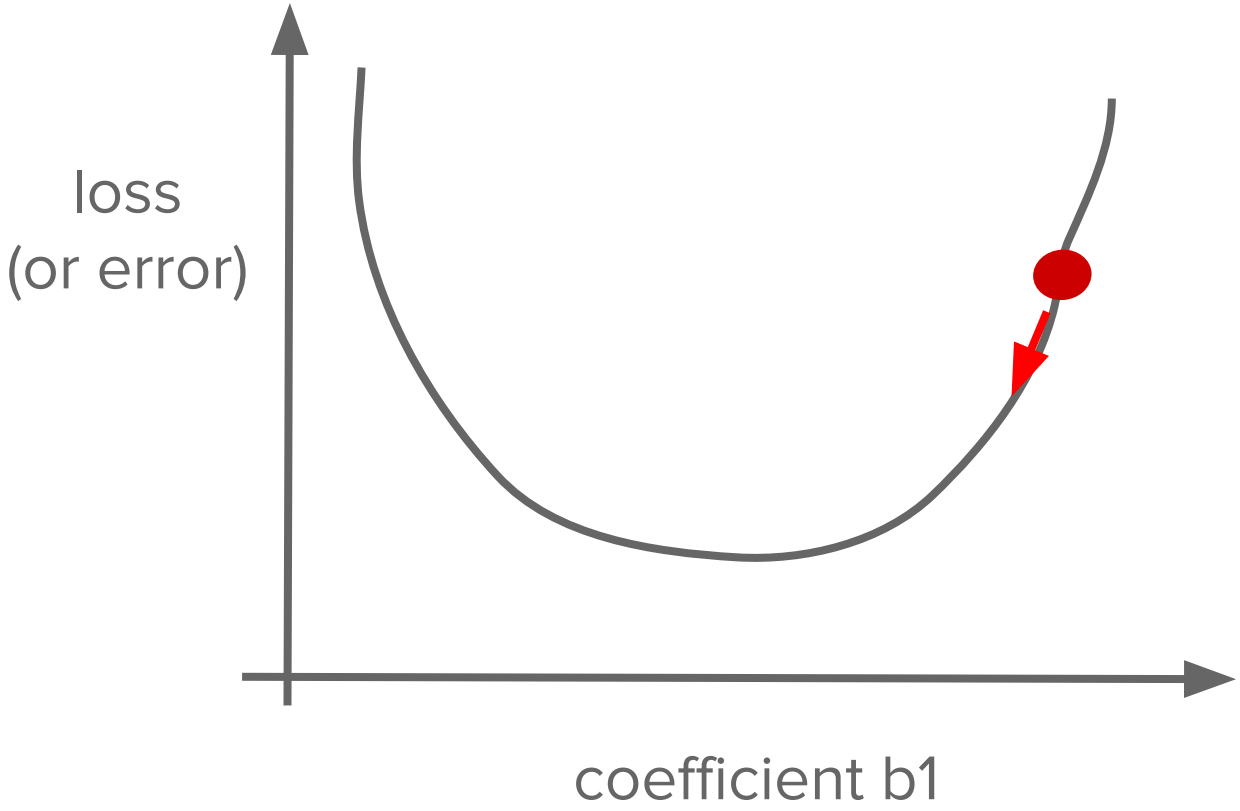
1) Start with random solution (b0,b1)
2) Find the right direction to get to a smaller error
3) Get a new better solution (b1, b2)
4) Repeat steps 2 and 3

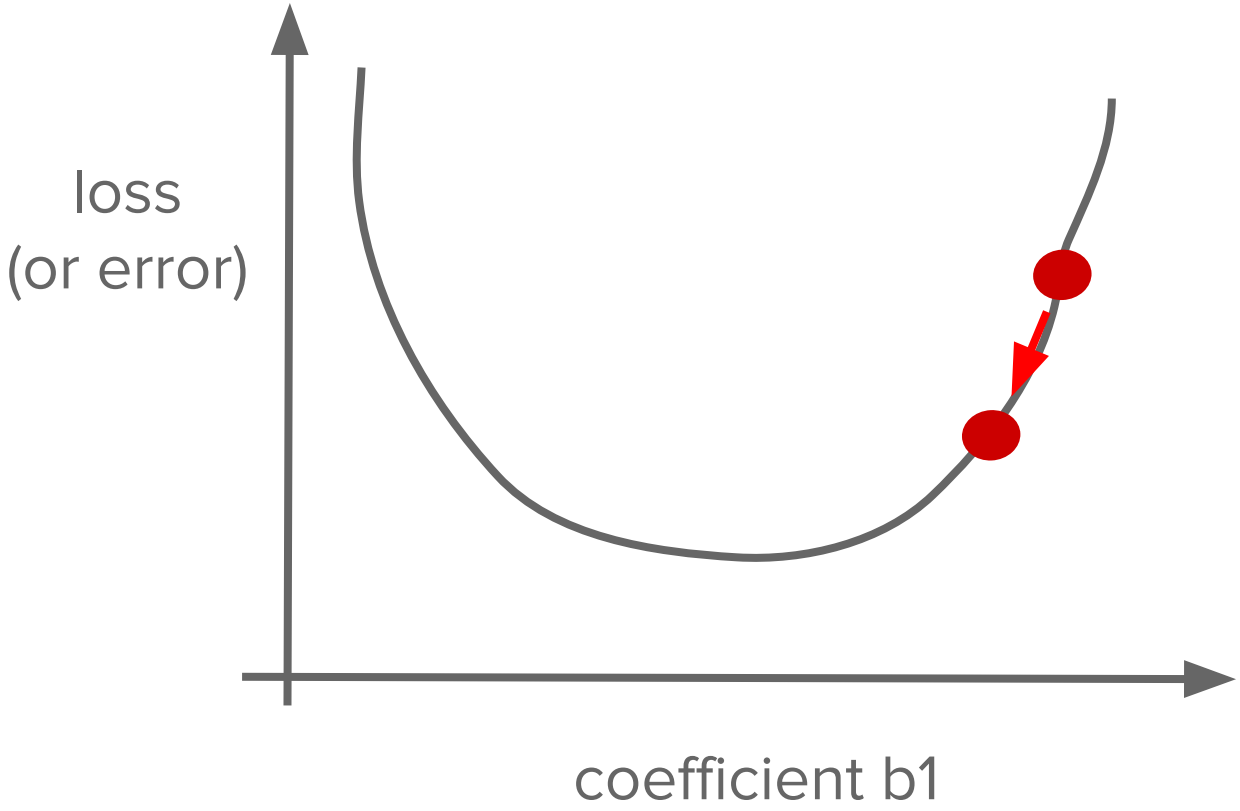

loss
(or error)
coefficient b1

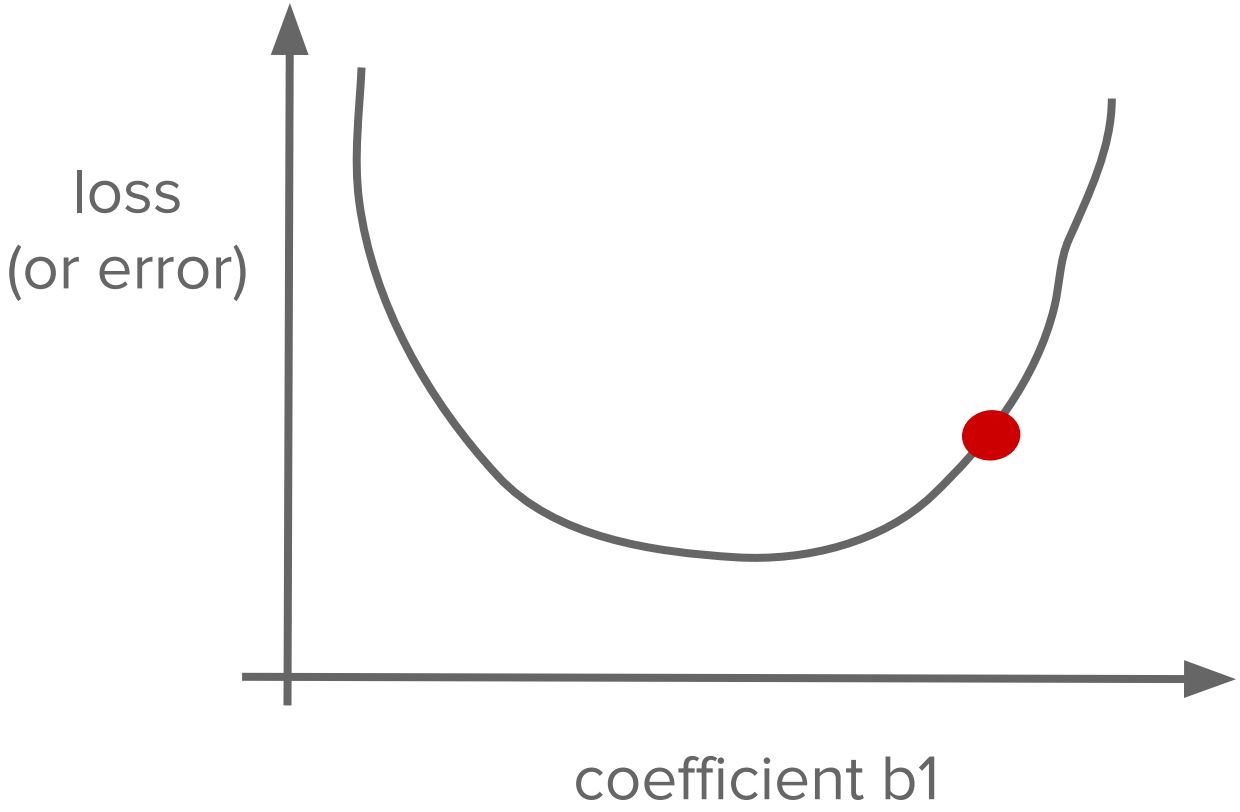

loss
(or error)
coefficient b1

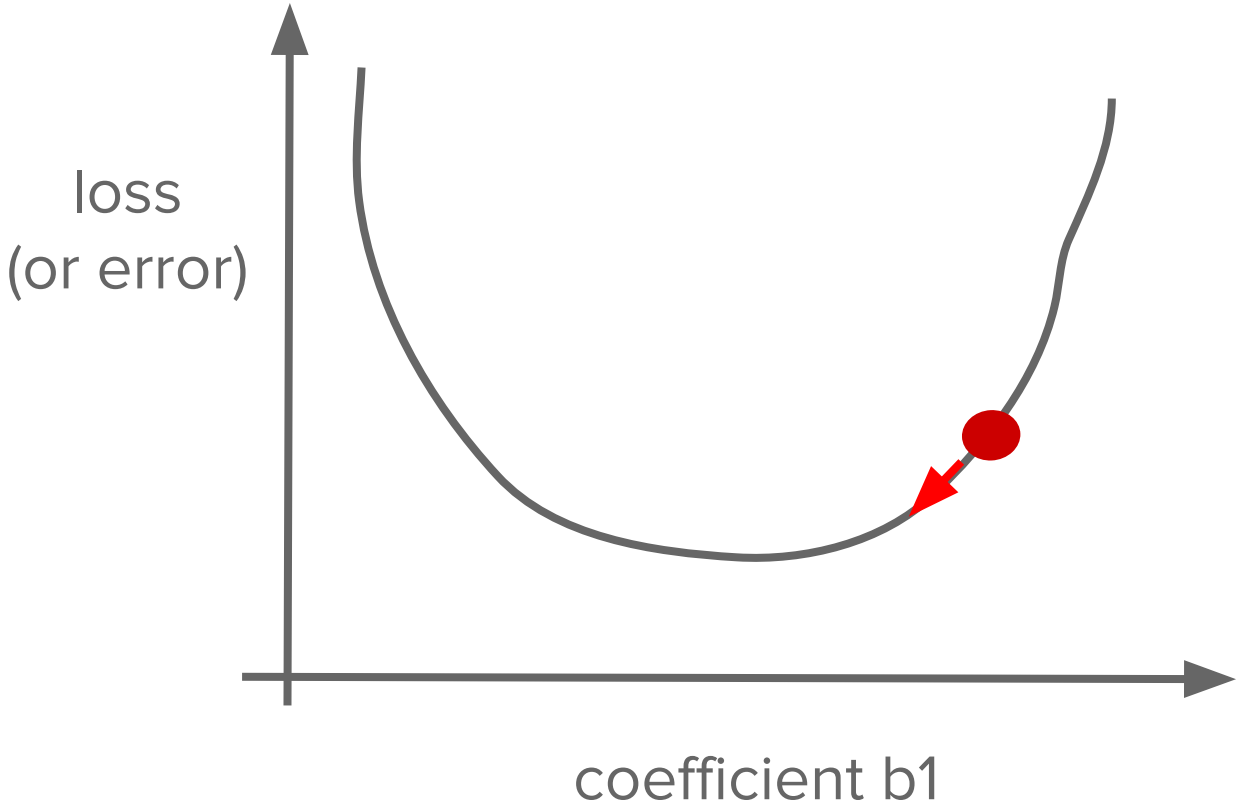

loss
(or error)
coefficient b1

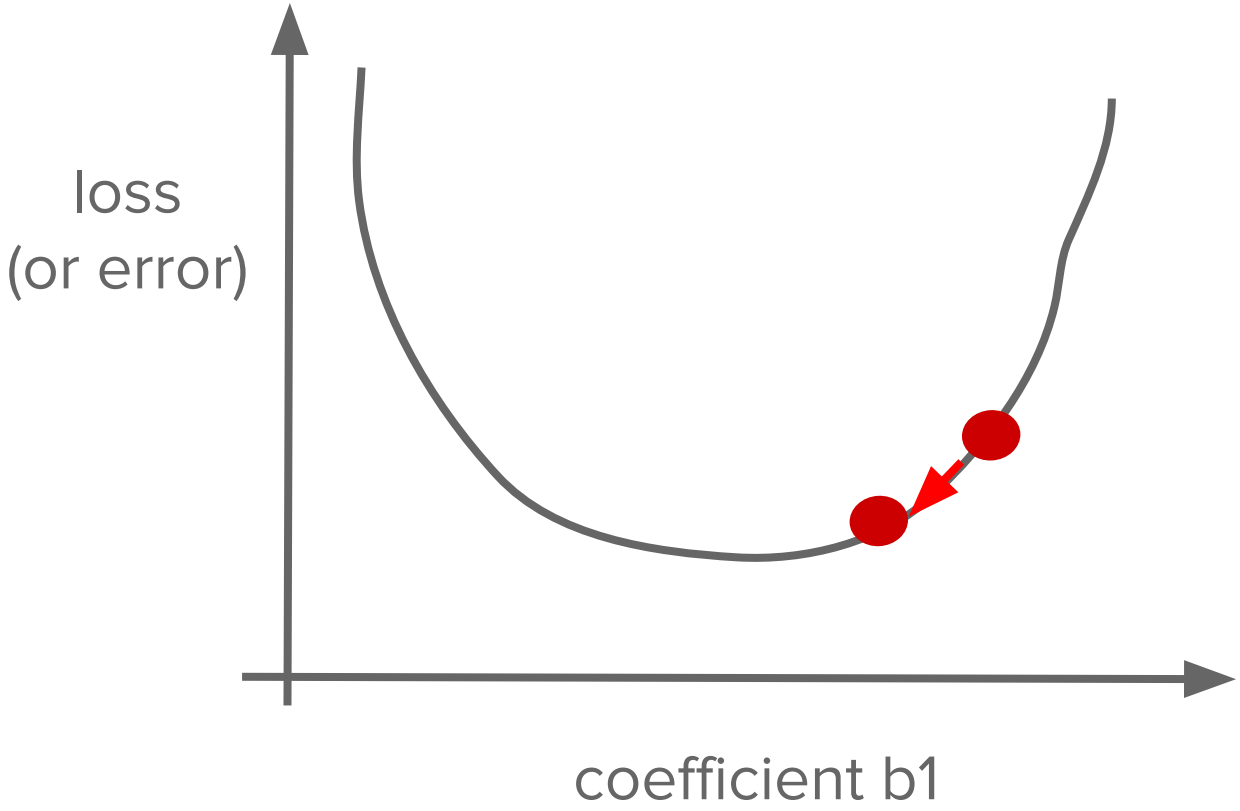

loss
(or error)
coefficient b1

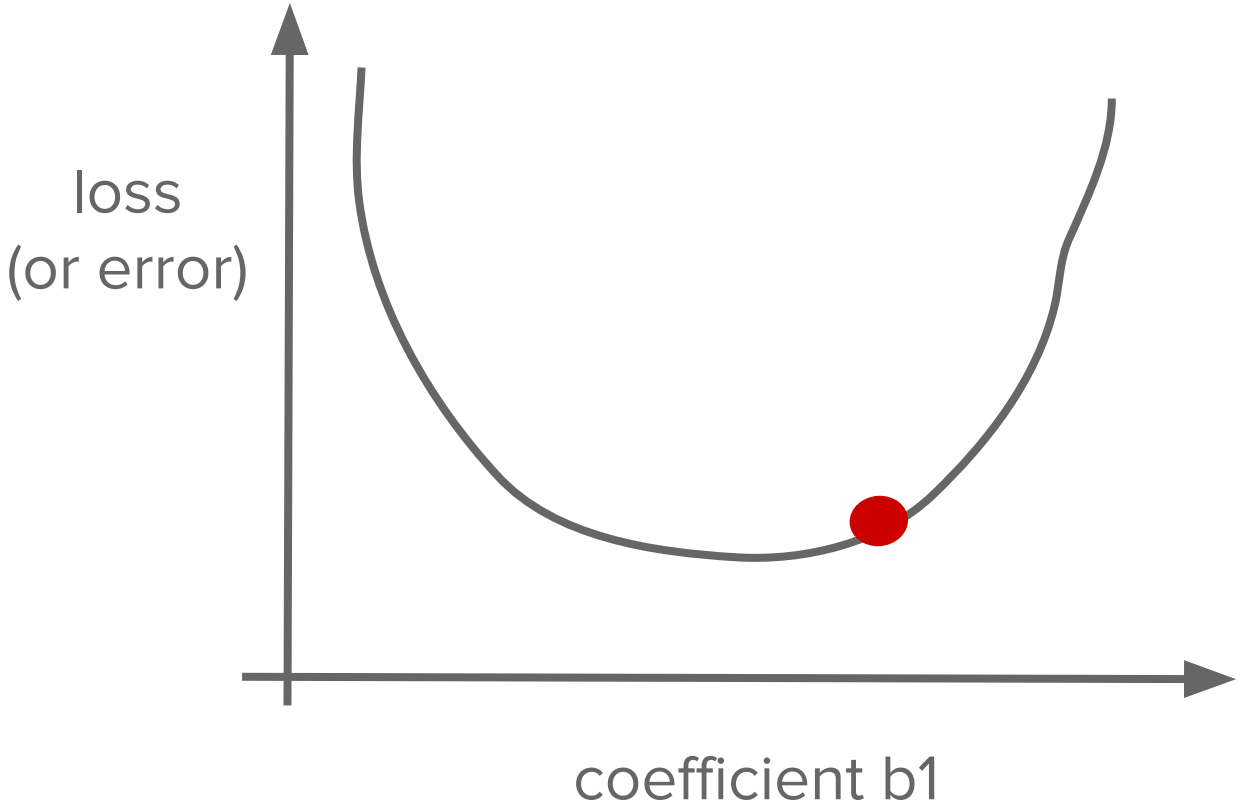

loss
(or error)
coefficient b1

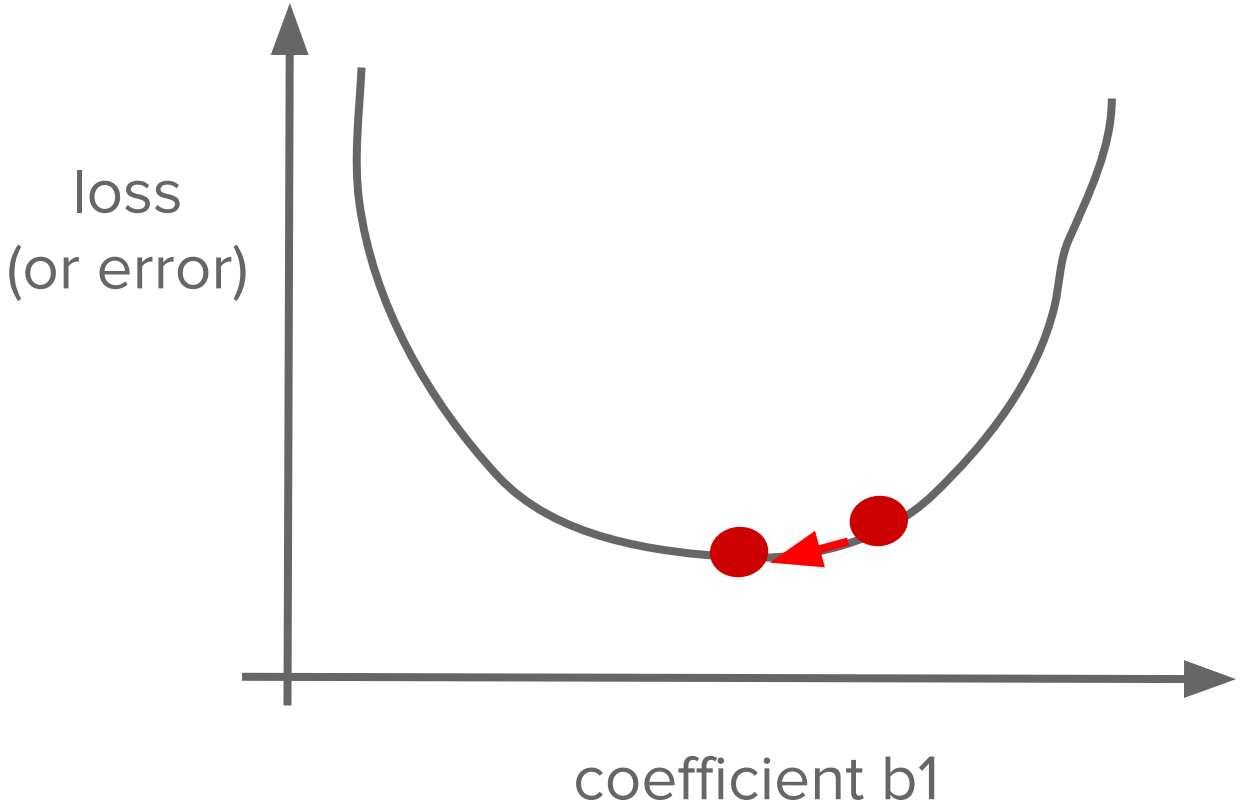

loss
(or error)
coefficient b1

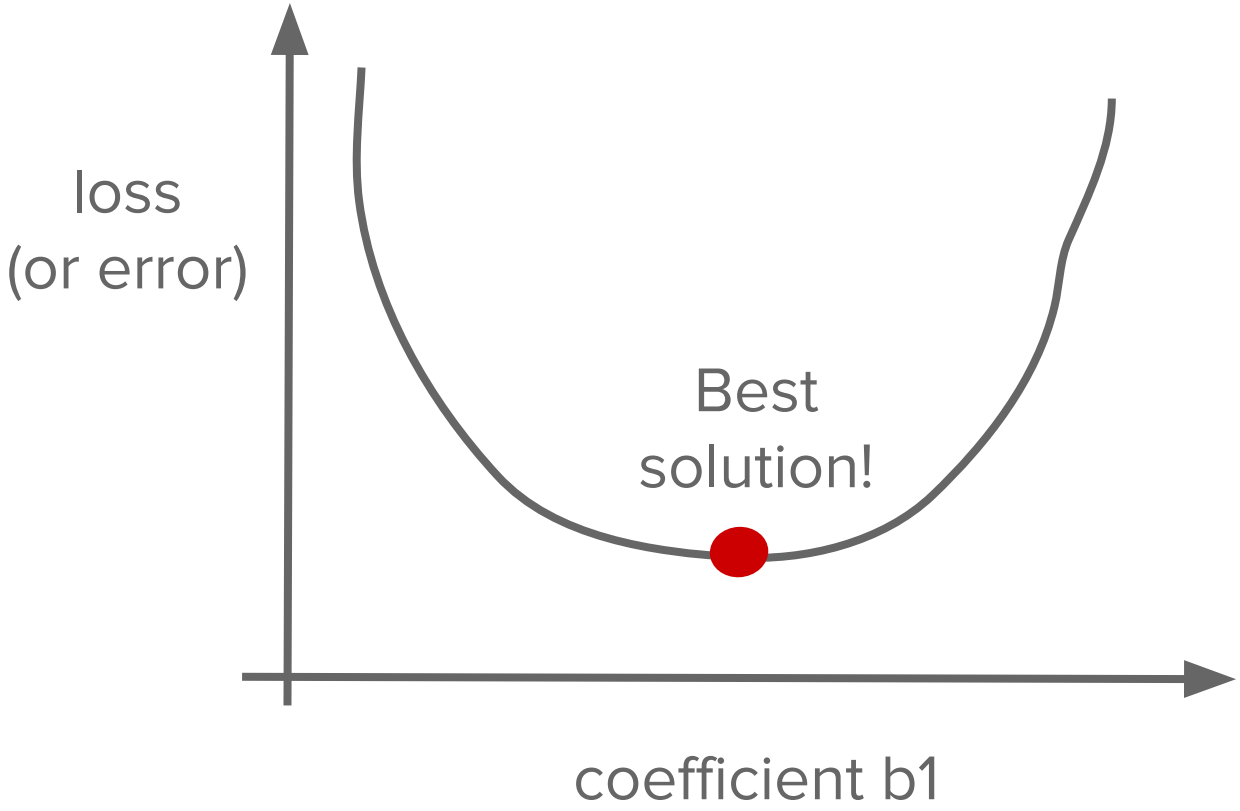

loss
(or error)
coefficient b1


loss
(or error)
coefficient b1

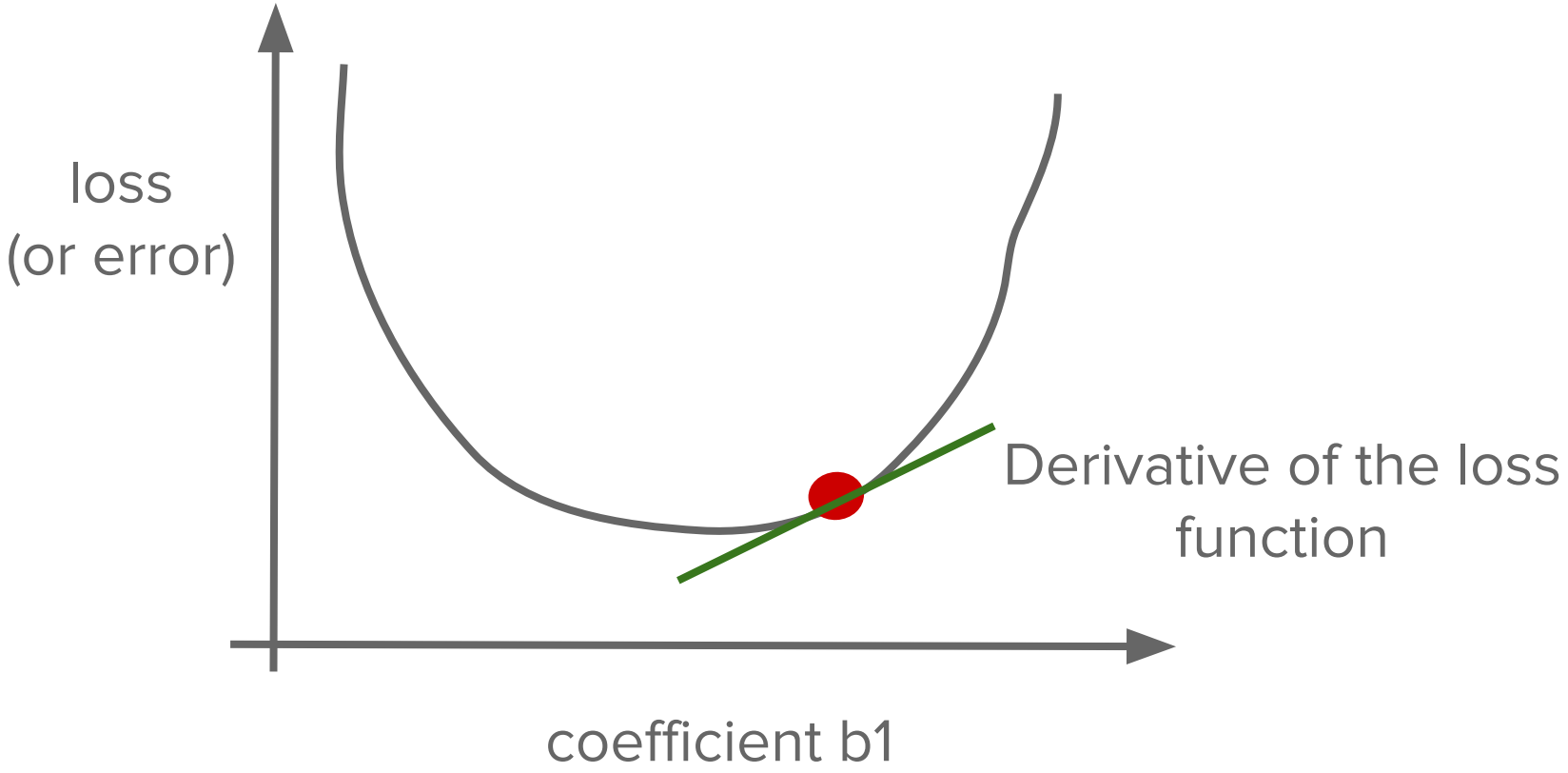

loss
(or error)
coefficient b1


loss
(or error)
coefficient b1


loss
(or error)
coefficient b1


loss
(or error)
Best
solution!
coefficient b1

How do we find the best right direction?


loss
(or error)
Derivative of the loss
function
coefficient b1

$$prediction = b0 + b1*x1$$

$$error = (y - prediction)**2$$

$$error = (y - (b0 + b1*x1))**2$$

$$partial\_derivative(error,b0,b1) = (7,5)$$

$$b0 = b0 - 7$$

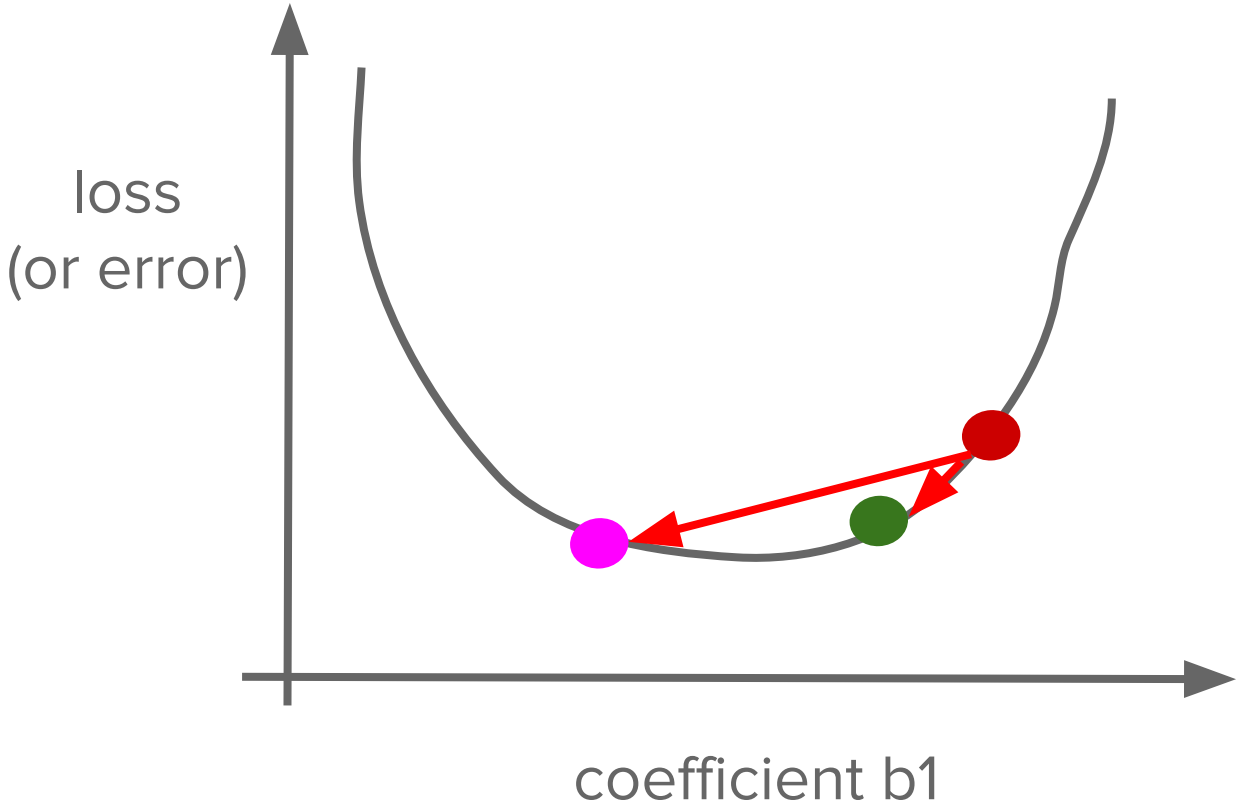
$$b1 = b1 - 5$$

$$b0 = b0 - \mathbf{0.01} * 7$$

$$b1 = b1 - \mathbf{0.01} * 5$$

Size of the step

Learning rate


loss
(or error)
coefficient b1

Small learning rate

++ faster

-- may no convergence

Small learning rate

-- slower

++ convergence

Recap

1. gradient descent = optimization process

1. gradient descent = optimization process
2. iterative approach

1. gradient descent = optimization process
2. iterative approach
3. small updates of the coefficients

1. gradient descent = optimization process
2. iterative approach
3. small updates of the coefficients
4. direction = derivative

1. gradient descent = optimization process
2. iterative approach
3. small updates of the coefficients
4. direction = derivative
5. learning rate is important

1. Andrew Ng - gradient descent
2. Siraj - gradient descent
3. Derivatives
4. Partial Derivatives