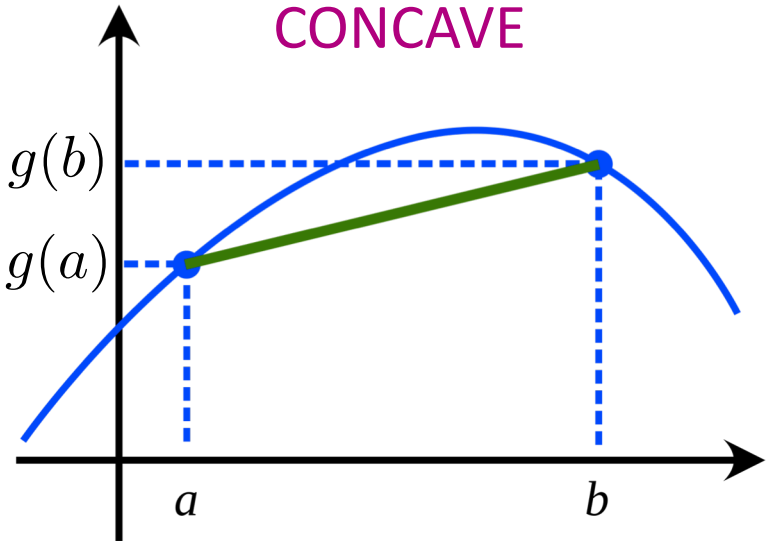
Gradient descent

Convex/concave functions

CONCAVE

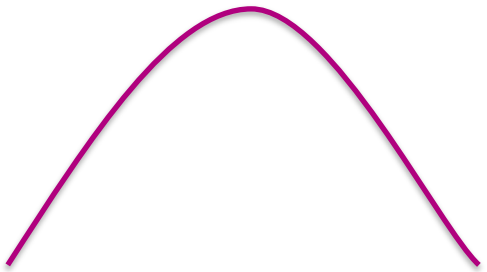


CONVEX

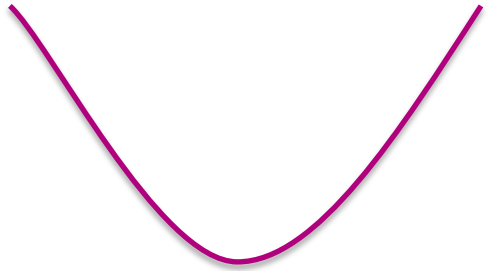
NEITHER

Finding the max or min analytically

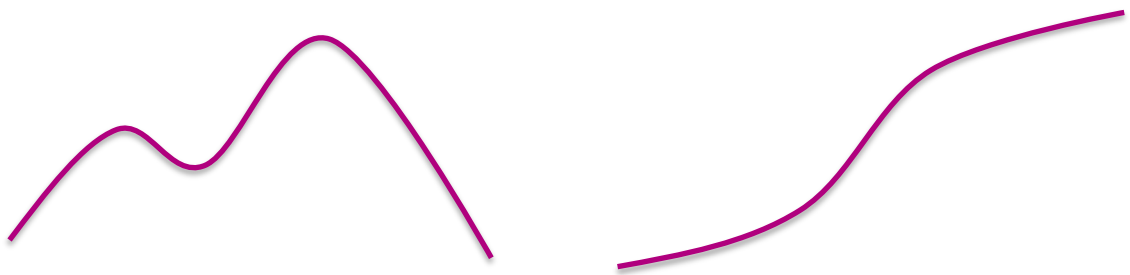
CONCAVE



CONVEX



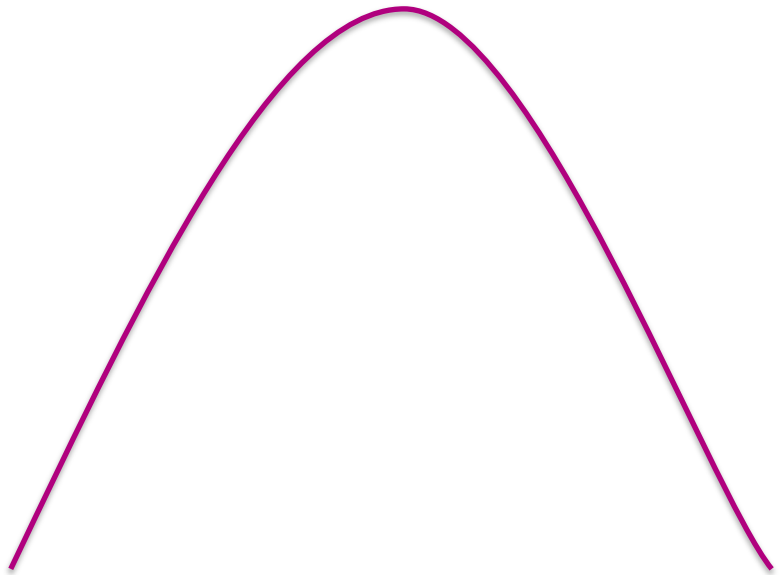
NEITHER



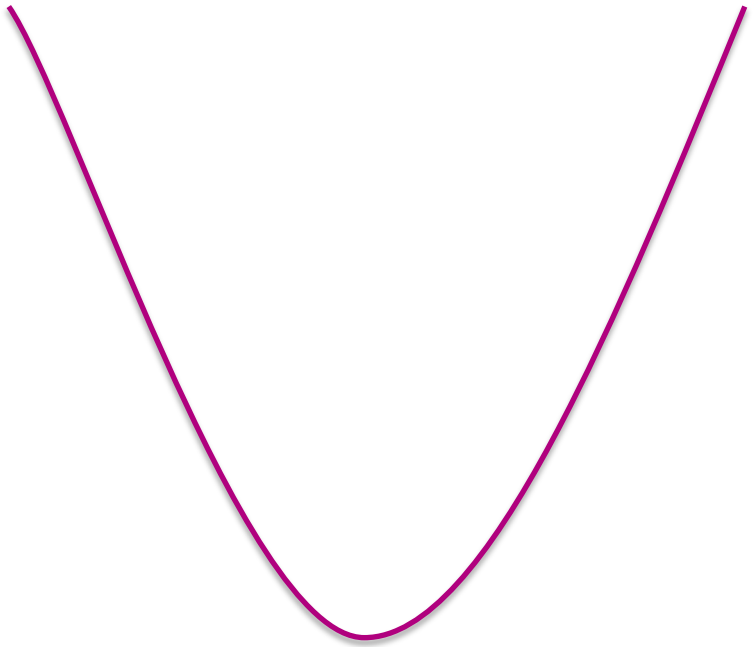
Example:

$g(w) = 5-(w-10)^2$

Finding the max via hill climbing



Finding the min via hill descent



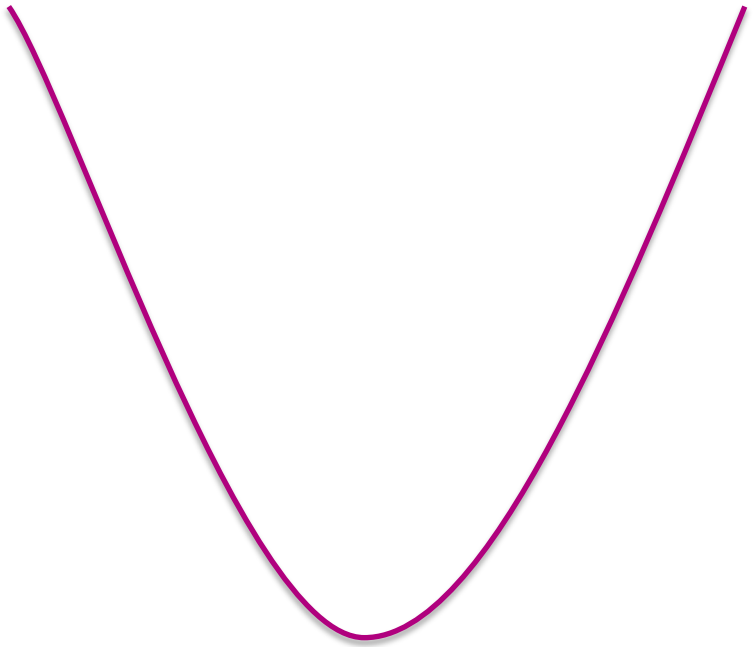
Algorithm:

while not converged

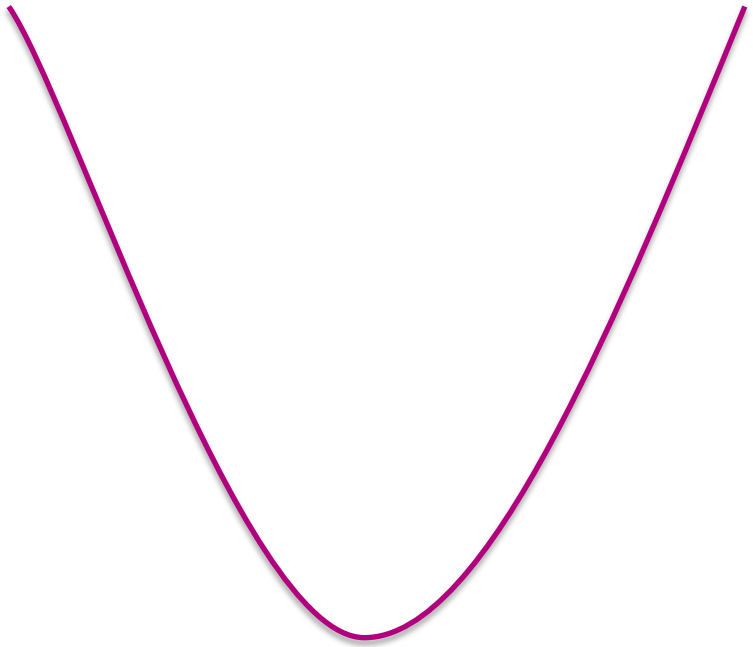
$$w^{(t+1)} \leftarrow w^{(t)} - \eta \left.\frac{dg}{dw}\right|_{w^{(t)}}$$

Choosing the stepsize—

Fixed stepsize



Choosing the stepsize—
Decreasing stepsize



Common choices:

Convergence criteria

For convex functions,

optimum occurs when

In practice, stop when

Algorithm:

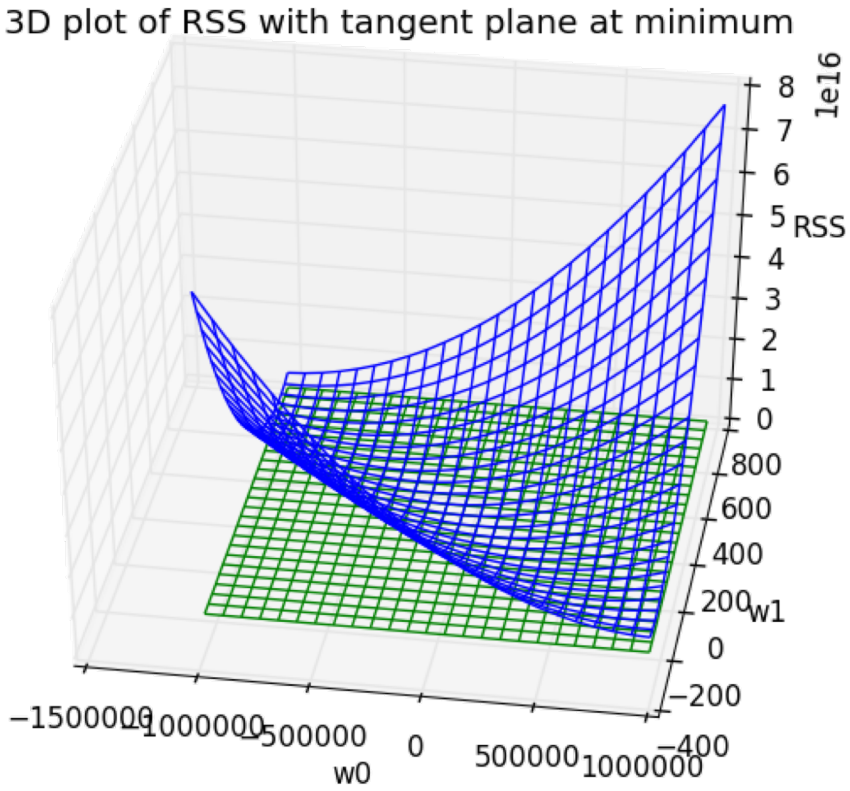
while not converged

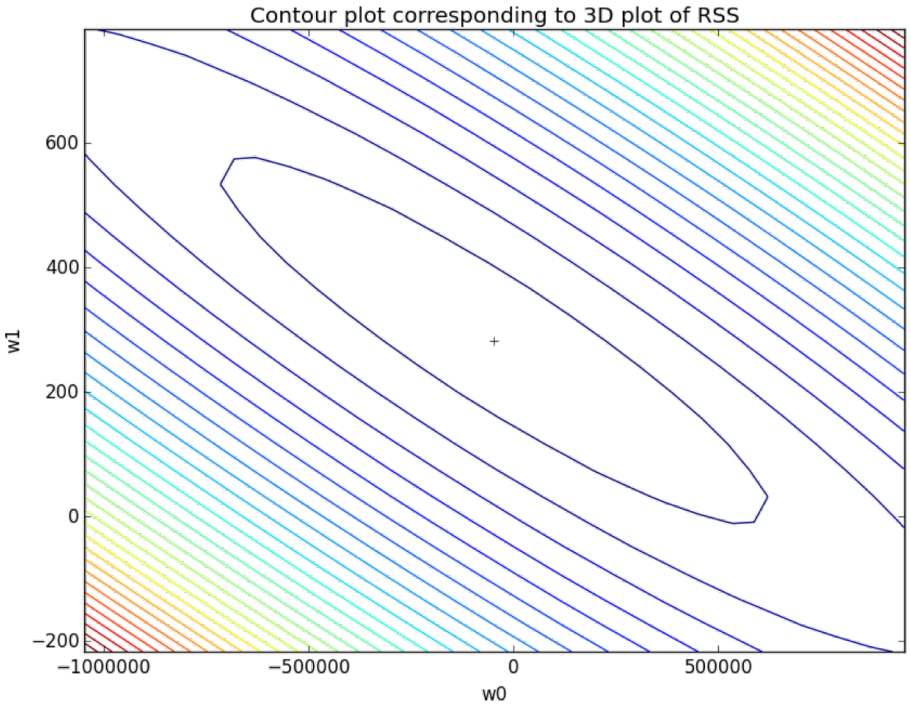
$$w^{(t+1)} \leftarrow w^{(t)} - \eta \left.\frac{dg}{dw}\right|_{w^{(t)}}$$

Moving to higher dimensions

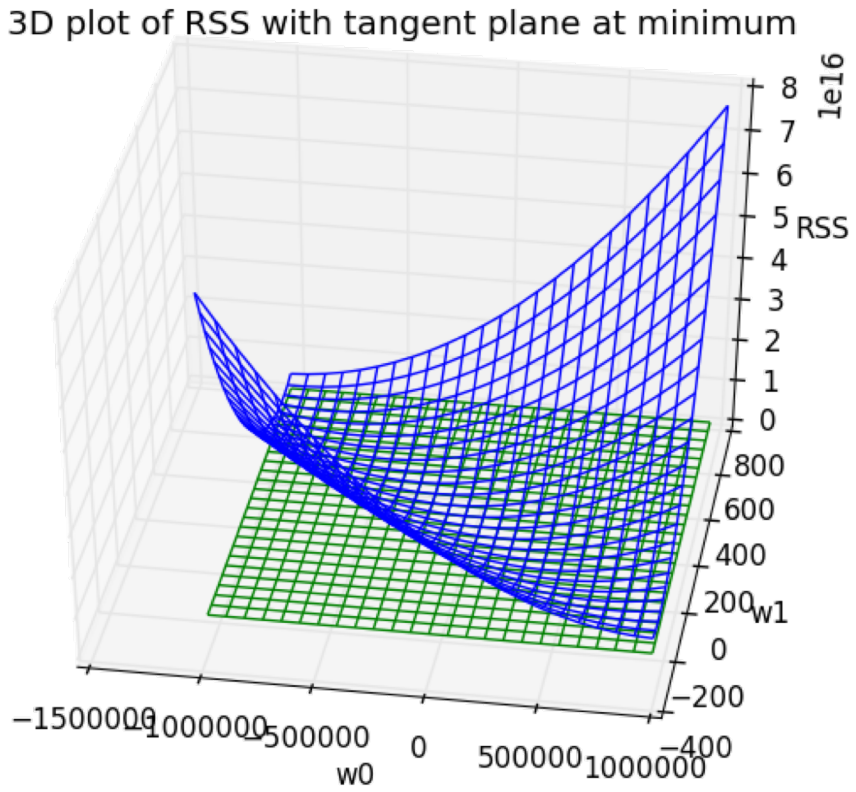
Note: We use the Optional tag to signify that you are not responsible for understanding the following material!

Contour plots



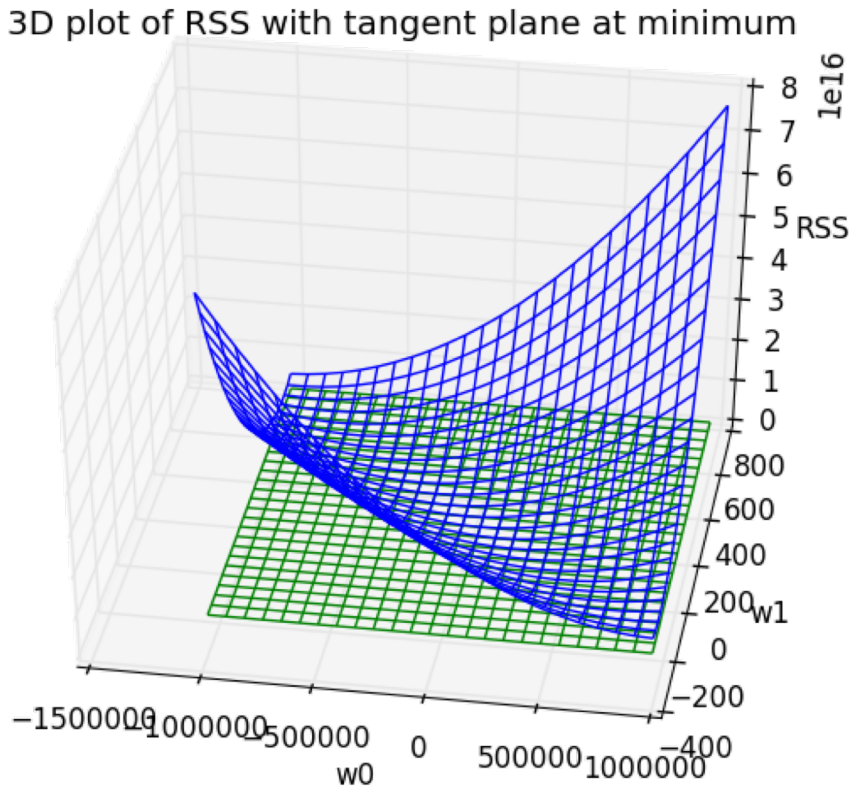


Moving to multiple dimensions: Gradients



$\nabla g(w) =$

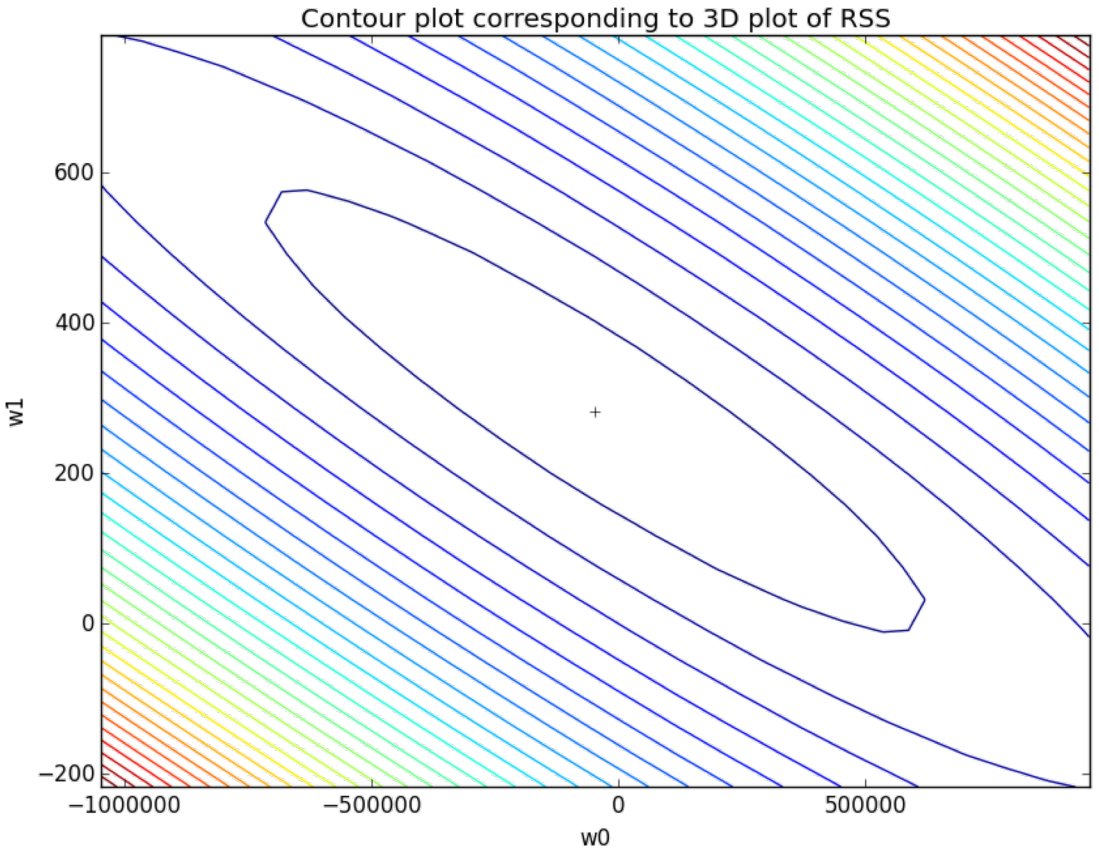
Gradient example



$$g(w) = 5w_0 + 10w_0w_1 + 2w_1^2$$

$$\nabla g(w) =$$

Gradient descent



Algorithm:

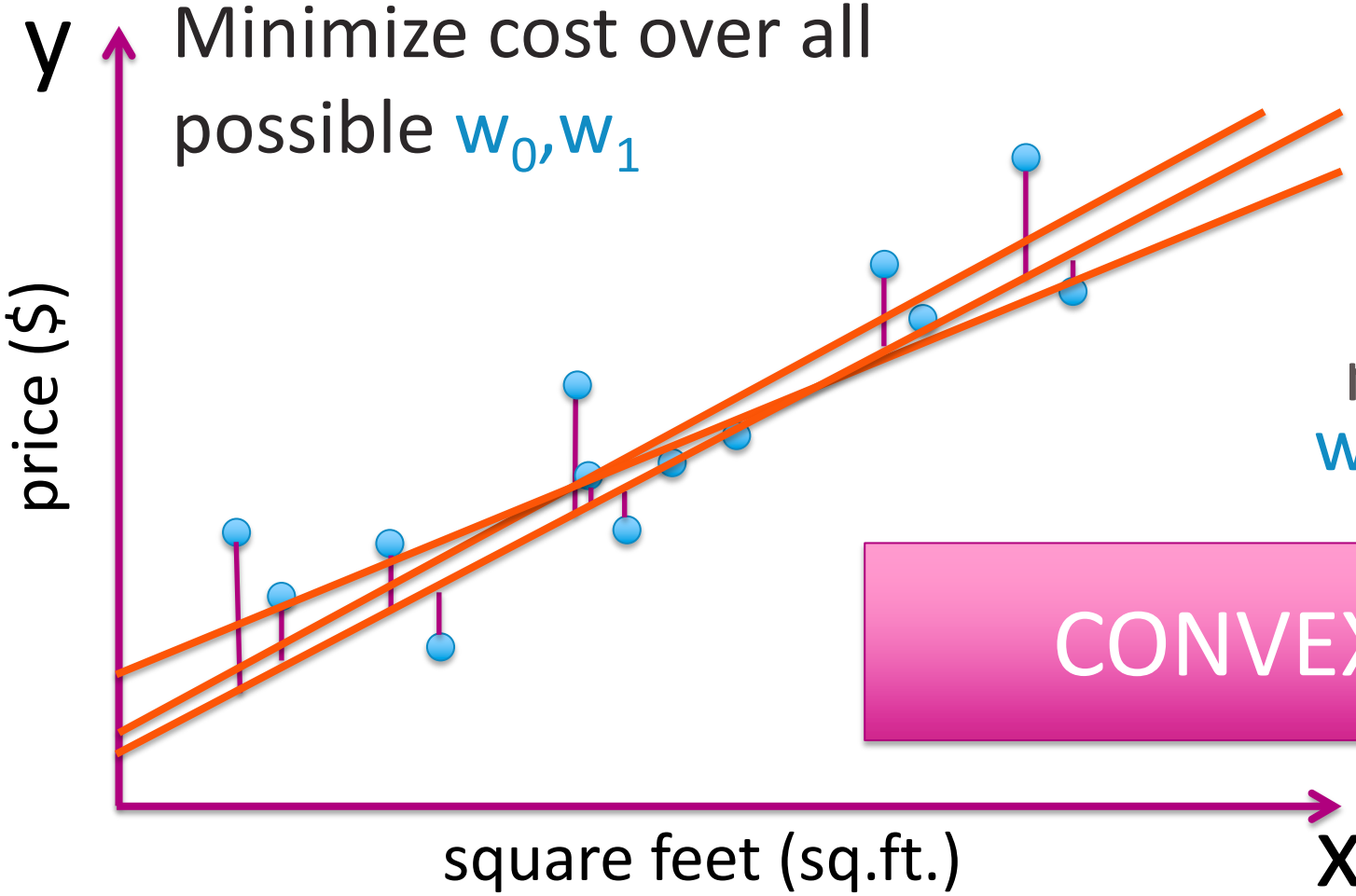
while not converged

$$w^{(t+1)} \leftarrow w^{(t)} - \eta \nabla g(w^{(t)})$$

Gradient Descent for Linear Regression

Note: We use the Cap to point out that the following section contains advanced topics, passed the level we expect from the class.

Find “best” line



Minimize cost over all possible w_0, w_1

$$\min_{w_0,w_1} \sum_{i=1}^{N} (y_i - [w_0 + w_1 x_i])^2$$

CONVEX

Compute the gradient

$$\text{RSS}(w_0, w_1) = \sum_{i=1}^{N} (y_i - [w_0 + w_1 x_i])^2$$

Aside:

$$\begin{aligned}
\frac{d}{dw} \sum_{i=1}^{N} g_i(w) &= \frac{d}{dw} \left(g_1(w) + g_2(w) + \dots g_N(w)\right) \\
&= \frac{d}{dw} g_1(w) + \frac{d}{dw} g_2(w) + \dots \frac{d}{dw} g_N(w) \\
&= \sum_{i=1}^{N} \frac{d}{dw} g_i(w)
\end{aligned}$$

Compute the gradient

$$\text{RSS}(w_0, w_1) = \sum_{i=1}^{N} (y_i - [w_0 + w_1 x_i])^2$$

Taking the derivative w.r.t. w_0

Compute the gradient

$$\text{RSS}(w_0, w_1) = \sum_{i=1}^{N} (y_i - [w_0 + w_1 x_i])^2$$

Taking the derivative w.r.t. w_1

Compute the gradient

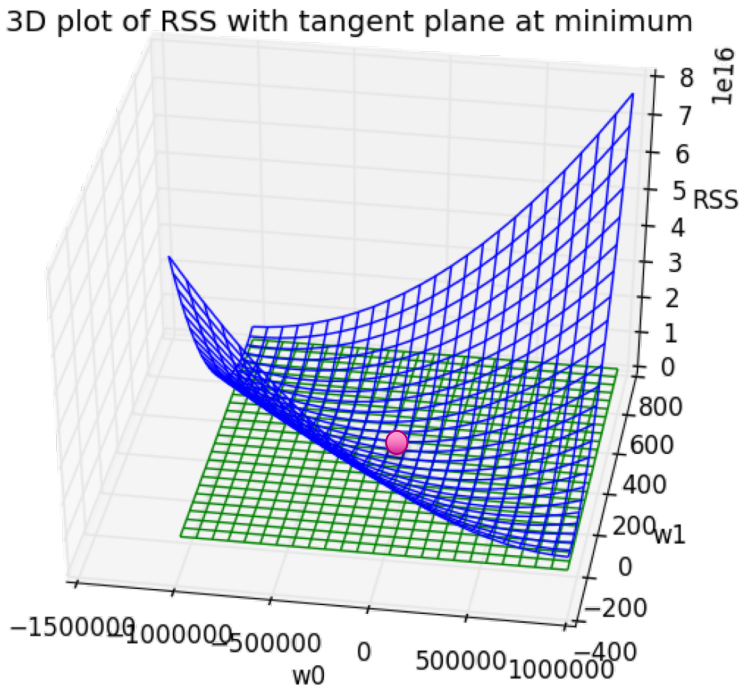
$$\text{RSS}(w_0,w_1) = \sum_{i=1}^{N}(y_i-[w_0+w_1x_i])^2$$

Putting it together:

$$\nabla \text{RSS}(w_0,w_1) = \begin{bmatrix} -2 \sum_{i=1}^{N}[y_i - (w_0+w_1x_i)] \\ -2 \sum_{i=1}^{N}[y_i - (w_0+w_1x_i)]x_i \end{bmatrix}$$

Approach 1: Set gradient = 0

$$\nabla \text{RSS}(w_0, w_1) = \begin{bmatrix} -2 \sum_{i=1}^{N} [y_i - (w_0 + w_1 x_i)] \\ -2 \sum_{i=1}^{N} [y_i - (w_0 + w_1 x_i)] x_i \end{bmatrix}$$



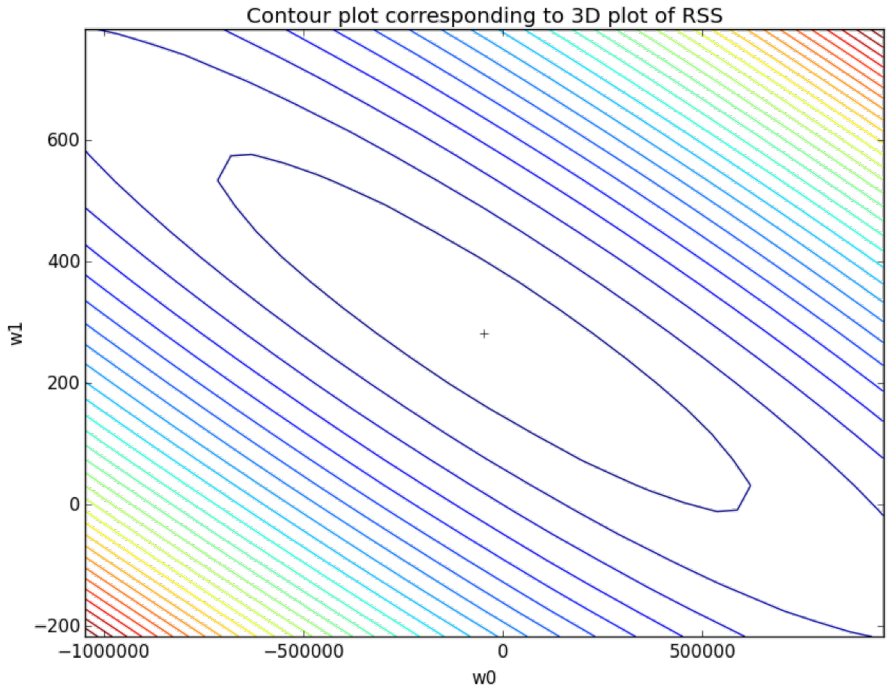
Approach 2: Gradient descent

Interpreting the gradient:

$$\nabla \text{RSS}(w_0, w_1) = \begin{bmatrix} -2 \sum_{i=1}^{N} [y_i - (w_0 + w_1 x_i)] \\ -2 \sum_{i=1}^{N} [y_i - (w_0 + w_1 x_i)] x_i \end{bmatrix} = \begin{bmatrix} -2 \sum_{i=1}^{N} [y_i - \hat{y}_i(w_0, w_1)] \\ -2 \sum_{i=1}^{N} [y_i - \hat{y}_i(w_0, w_1)] x_i \end{bmatrix}$$

Approach 2: Gradient descent

$$\nabla \text{RSS}(w_0, w_1) = \begin{bmatrix} -2 \sum_{i=1}^{N} [y_i - \hat{y}_i(w_0, w_1)] \\ -2 \sum_{i=1}^{N} [y_i - \hat{y}_i(w_0, w_1)] x_i \end{bmatrix}$$



Comparing the approaches

- For most ML problems, cannot solve gradient = 0
- Even if solving gradient = 0 is feasible, gradient descent can be more efficient
- Gradient descent relies on choosing stepsize and convergence criteria