Otimização avançada de função custo na regressão logística pode ser feita com os seguintes algoritmos:

- Gradient Descent
- Conjugate gradient
- BFGS
- L-BFGS

more complex
• line surge algorithm

• no need to manually pick α
• often faster than gradient descent
→ Those methods would take very few days to understand the general lines.

↳ When implementing these algorithms look for good libraries, no matter what language are you using.

Try some, and compare the results to pick a good one.

options = optimset('GradObj', 'on', 'MaxIter', '100');

in Octave, use: fminunc(@costFunction, initialTheta, options);

gradient steps vector

function [jVal, gradient] = costFunction(theta)

cost function → jVal = $(\theta_1 - 5)^2 + (\theta_2 - 5)^2$;

gradient = zeros(2,1);
gradient(1) = 2 * theta(1) - 5;
gradient(2) = 2 * theta(2) - 5;

Usar esses códigos quando os problemas de regressão logística forem muito grdes, do invés de Gradient Descent.

Isso porque esses outros algoritmos convergem muito mais rapidamente do que a Gradient Descent.

(permitem exploration e exploitation, além de estarem otimizados numericamente)