Loss functions

Linear regression and MSE:

$$L(w) = \frac{1}{\ell}\|Xw - y\|^2$$

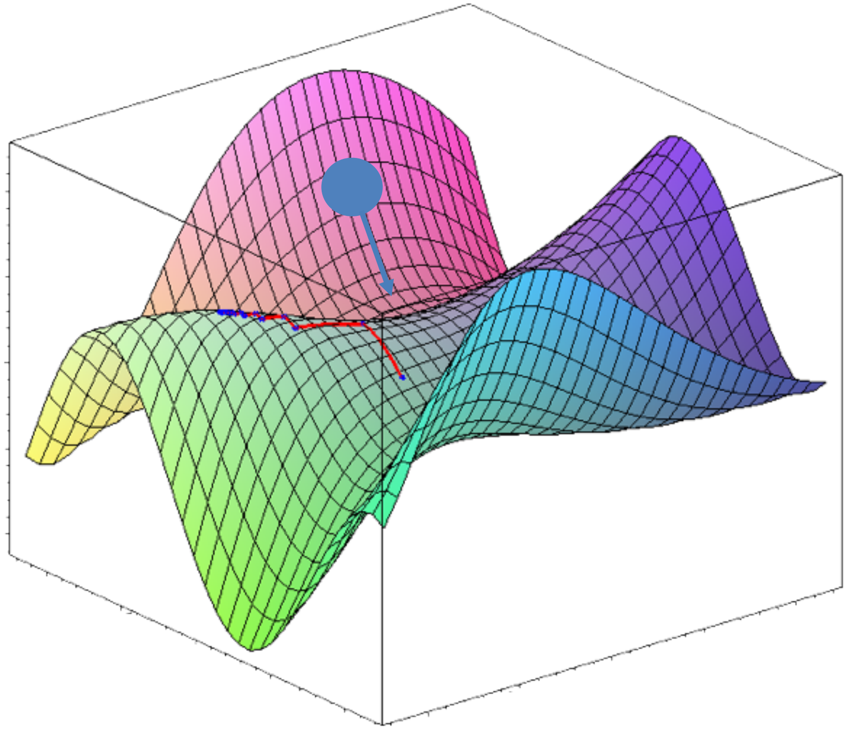
Linear classification and cross-entropy:

$$L(w) = -\sum_{i=1}^{\ell}\sum_{k=1}^{K}[y_i = k]\log\frac{e^{w_k^T x_i}}{\sum_{j=1}^{K} e^{w_j^T x_i}}$$

Gradient descent

Optimization problem: $L(w) \to \min_{w}$

Suppose we have some approximation w^0 — how to refine it?



Gradient descent

Optimization problem: $L(w) \to \min_{w}$

w^0 — initialization

$\nabla L(w^0) = \left(\frac{\partial L(w^0)}{\partial w_1}, \ldots, \frac{\partial L(w^0)}{\partial w_n}\right)$ — gradient vector

- Points in the direction of the steepest slope at w^0
- The function has fastest decrease rate in the direction of negative gradient

Gradient descent

Optimization problem: $L(w) \to \min_{w}$

w^0 — initialization

$\nabla L(w^0) = \left(\frac{\partial L(w^0)}{\partial w_1}, \dots, \frac{\partial L(w^0)}{\partial w_n}\right)$ — gradient vector

$w^1 = w^0 - \eta_1 \nabla L(w^0)$ — gradient step

Gradient descent

Optimization problem: $L(w) \to \min_{w}$

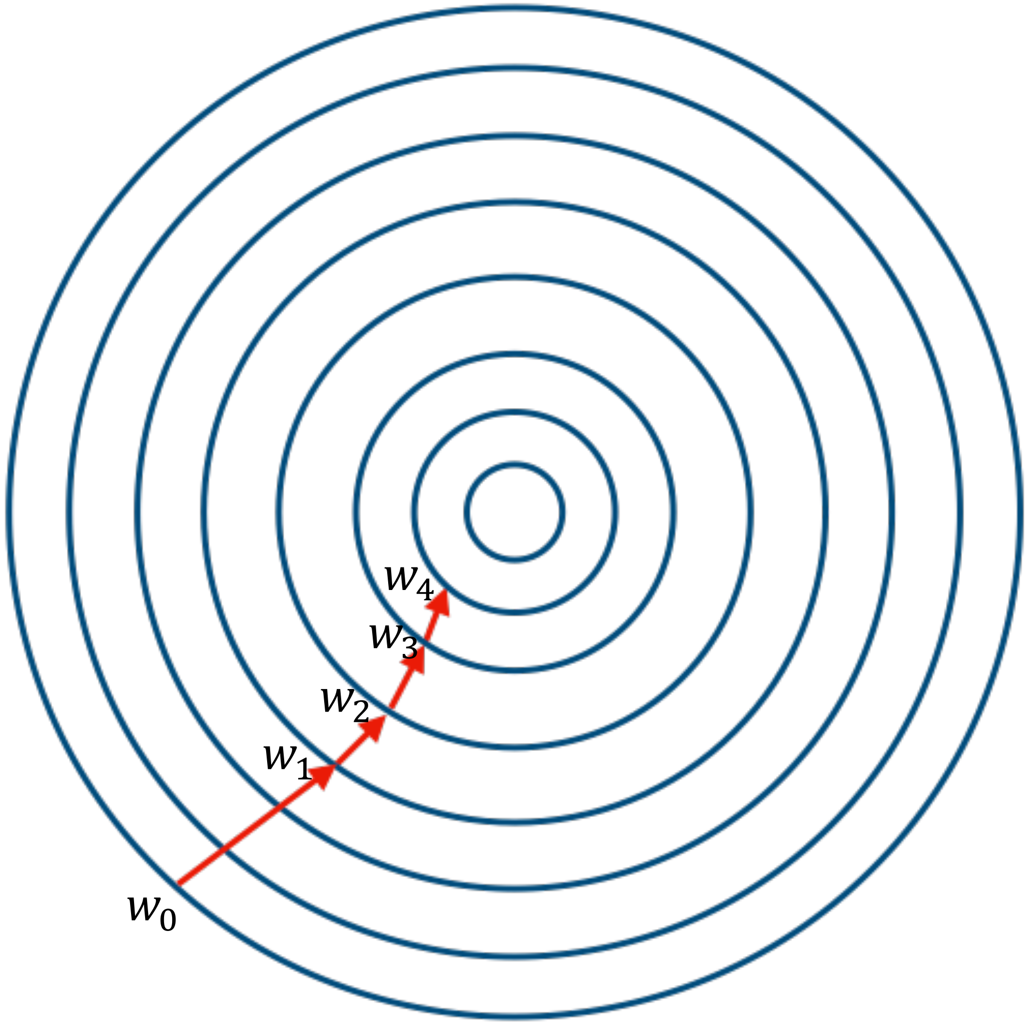
w^0 — initialization

while True:

$$w^t = w^{t-1} - \eta_t \nabla L(w^{t-1})$$

if $\|w^t - w^{t-1}\| < \epsilon$ then break

Gradient descent



Gradient descent

Lots of heuristics:

- How to initialize w^0
- How to select step size η_t
- When to stop
- How to approximate gradient $\nabla L(w^{t-1})$

Gradient descent for MSE

Linear regression and MSE:

$$L(w) = \frac{1}{\ell} \|Xw - y\|^2$$

Derivatives:

$$\nabla L_w(w) = \frac{2}{\ell} X^T (Xw - y)$$

Gradient descent vs analytical solution

Analytical solution for MSE: $w = (X^T X)^{-1} X^T y$

Gradient descent:

- Easy to implement
- Very general, can be applied to any differentiable loss function
- Requires less memory and computations (for stochastic methods)

Summary

- Gradient descent provides a general learning framework
- Can be used both for classification and regression tasks
- Advanced methods — in next lessons