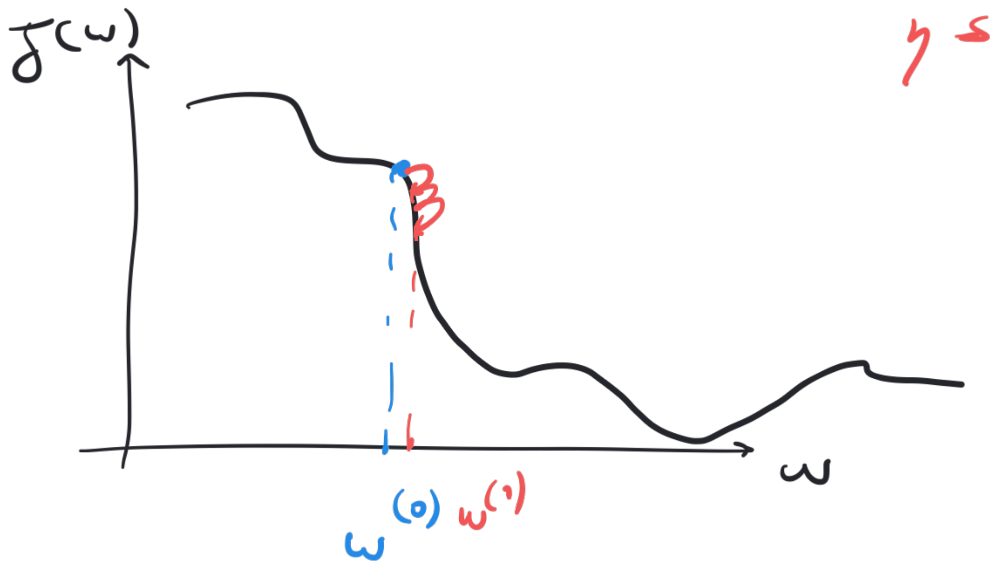



η small

weight update rule : $w^{(t+1)} \leftarrow w^{(t)} \overbrace{- \eta \cdot \nabla J(w^{(t)})}^{\Delta w^{(t)}}$

$\eta \equiv$ learning rate (global variable)

Delta correction rule : $\Delta w^{(t)} = - \eta \cdot \nabla J(w^{(t)})$

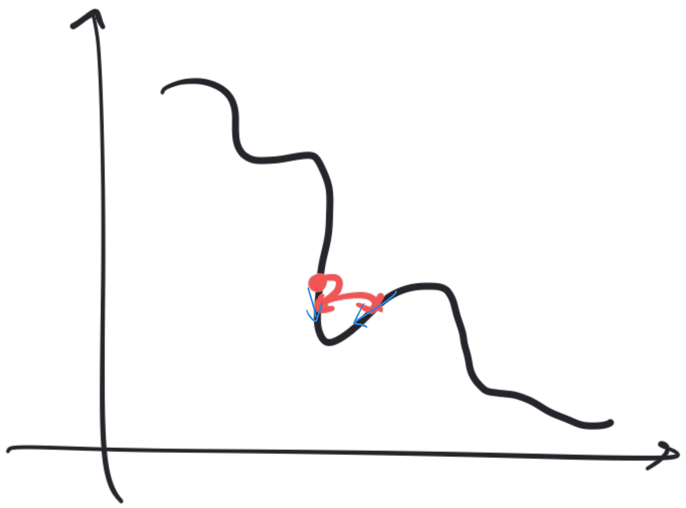
$\therefore \; w^{(t+1)} \leftarrow w^{(t)} + \Delta w^{(t)}$

## Momentum

Accelerate learning by adding information about previous gradient

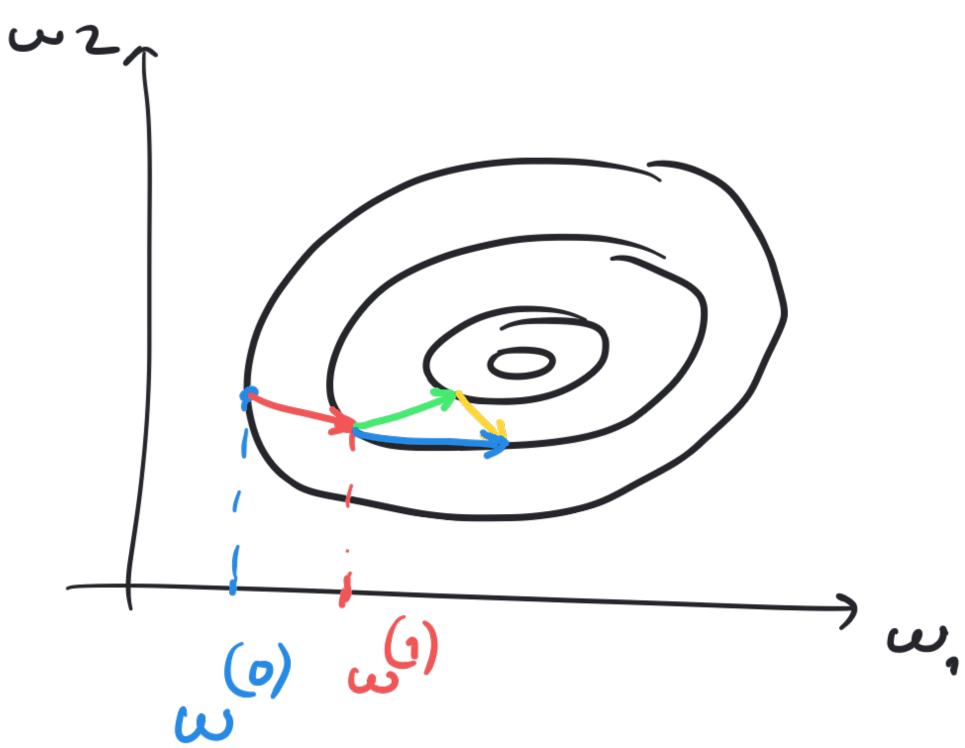
$$\Delta w^{(t)} = -\eta \cdot \nabla J(w^{(t)}) + \alpha \Delta w^{(t-1)}$$

$(\alpha = 0.9)$



→ Momentum may avoid local optima

→ it deaccelerates as the gradient sign changes.




Contours of objective function

$$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

$$w^{(1)} = w^{(0)} - \eta \cdot \nabla J(w^{(0)})$$

$\rightarrow$ $-\eta \nabla J(w^{(1)})$

$\rightarrow$ $\alpha \cdot \Delta w^{(1)}$

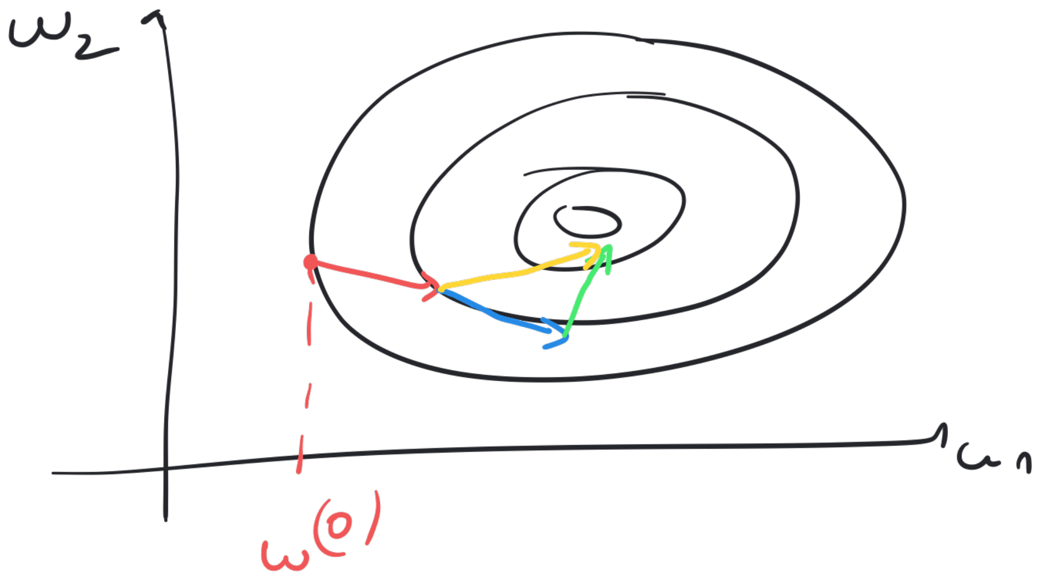
$\rightarrow$ $-\eta \nabla J(w^{(1)}) + \alpha \Delta \cdot w^{(1)} + w^{(1)} = w^{(2)}$

# Nesterov's Momentum

We first add the correction rule at $(t-1)$ and then compute the gradient at that location.

$$\Delta w^{(t)} = m^{(t)} - \eta \cdot \nabla J(m^{(t)})$$

$$m^{(t)} = w^{(t)} + \alpha \cdot \Delta w^{(t-1)}$$




$\rightarrow \; -\eta \cdot \nabla J(w^{(0)}) = \Delta . w^{(0)}$

$\rightarrow \; \alpha . \Delta w^{(0)}$

# Adaptive Learning Rate

$$\Delta w^{(t)} = -\eta^{(t)} \cdot \nabla J(w^{(t)})$$

① SCHEDULER — periodically schedule the learning rate decrease.

$\delta_{ij}^{(t)} \equiv$ gain for connection $w_{ij}$ at iteration $t$

$$\Delta w_{ij}^{(t)} = -\eta \cdot \delta_{ij}^{(t)} \cdot \nabla J(w_{ij}^{(t)})$$

---

If $\nabla J(w_{ij}^{(t)}) \cdot \nabla J(w_{ij}^{(t-1)}) > 0$ :

moving towards same minima

ADDITIVE INCREASE

$$\delta_{ij}^{(t+1)} = \delta_{ij}^{(t)} + 0.01$$

Else :

moving in different directions

$$\delta_{ij}^{(t+1)} = \delta_{ij}^{(t)} \times 0.99$$

**RMS Prop** : only <u>decreases</u> the gain using a multiplicative decrease strategy.

## Adam (2015)

1. Adds momentum term (to increase speed) $\beta_1 = 0.9$
2. perform adaptive learning rate. $\beta_2 = 0.99$ (multiplicative decrease

**Nadam** : it uses Nesterov's momentum.