Gradient descent algorithms

We use imperative assignment

$$\vec{\omega} := \vec{\omega} + \Delta\vec{\omega}$$

$\vec{\nabla} E(\vec{\omega})$ refers to the gradient of the error at $\vec{\omega}$

$\vec{\nabla}_i E(\vec{\omega})$ is the i-th component of the gradient, i.e. $\partial E / \partial \omega_i$

$\vec{\nabla}^{(i)} E(\vec{\omega})$ is the gradient computed for a single example $(\vec{x}_i, \vec{y}_i)$

BATCH

$$\vec{\omega} := \vec{\omega} - \alpha \vec{\nabla} E(\vec{\omega})$$

(α: learning rate)

} one update every epoch

On-line

$$\vec{\omega} := \vec{\omega} - \alpha \nabla^{(i)} E(\vec{\omega})$$

} one update for every example

Mini-batch

$$\vec{\omega} := \vec{\omega} - \alpha \nabla^{(i:i+m)} E(\vec{\omega})$$

} one update per mini-batch

Momentum

$$\Delta\vec{\omega} := \gamma \Delta\vec{\omega} - \eta \vec{\nabla} E(\vec{\omega})$$

$$\vec{\omega} := \vec{\omega} + \Delta\vec{\omega}$$

OR

$$\Delta\vec{\omega} := \beta\,\Delta\vec{\omega} + (1-\beta)\left(-\vec{\nabla}E(\vec{\omega})\right)$$

$$\vec{\omega} := \vec{\omega} + \Delta\vec{\omega}$$

If $\beta = 0.9$

$$\Delta\vec{\omega} := \underbrace{0.9\,\Delta\vec{\omega}}_{\text{keep 90\% of last correction}} + \underbrace{0.1\left(-\vec{\nabla}E(\vec{\omega})\right)}_{\text{add 10\% of the negative gradient}}$$

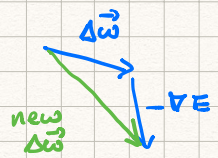
Momentum is only used with batch BP or mini-batches, not with online BP

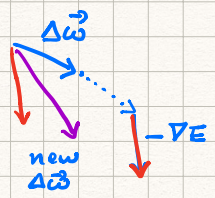
Nesterov accelerated gradient

$$\Delta\vec{\omega} := \gamma\,\Delta\vec{\omega} + \eta\left(-\vec{\nabla}E(\vec{\omega} + \gamma\,\Delta\vec{\omega})\right)$$

$$\vec{\omega} := \vec{\omega} + \Delta\vec{\omega}$$

If $\gamma = 0.9$, $\eta = 0.1$, keep 90% of last correction and add 10% of the negative gradient at an approximation of the gradient at the next point.





Adagrad

Adapt each weight with a different learning rate

sum of squares of past gradients →

$$G_i^t = G_i^{t-1} + (\Delta w_i)^2$$

$$\Delta w_i = \underbrace{-\frac{\alpha}{\sqrt{G_i^t + \varepsilon}}}\,\vec{\nabla}_i E(\vec{\omega})$$

the learning rate is smaller if the RMS of the corrections is too big. (past)

RMSprop

corrections are done element-wise

$$E[\Delta w_i^2] := \underbrace{0.9}_{\gamma} E[\Delta w_i^2] + \underbrace{0.1}_{(1-\gamma)} [\nabla_i E(\vec{\omega})^2]$$

$$\vec{\omega}_i := \vec{\omega}_i - \frac{\alpha}{\sqrt{E[\Delta w_i^2] + \epsilon}} \vec{\nabla}_i E(\vec{\omega})$$

Adam

$$\vec{\mu}_i := \beta_1 \vec{\mu}_i + (1-\beta_1) \vec{\nabla}_i E(\vec{\omega})$$

$$v_i := \beta_2 v_i + (1-\beta_2) \nabla_i E(\vec{\omega})^2$$

$$\vec{\omega} := \vec{\omega} - \frac{\alpha}{\sqrt{v} + \epsilon} \vec{\mu}$$

Two hyperparameters, one for tracking the mean, one for tracking the variance

(similar to the Kalman filter)