Opcimizacion FunGcions

27 May 2018 16:51

9aramecer Updace

Vani11a Updace

Change che paramecers a1ong che negacive gradienc direGcion

$$w_i = w_i - h\frac{6C}{6w_i} \qquad \text{wMere fi is learning Rate.}$$

A sma11 fi guarancess che progress in non – negacive progress on 1oss funGcion, cowards minima.
Buc sma11 1earning race inGreases che craiing cime.

Momencum Updace

$$v = u \cdot v - fi\frac{6c}{6m_i} \quad \text{where u is momentum}$$

$$w_i = w_i + v$$

A cypiGa1 momencum annea1ing seccing is co scarc wich momencum of abouc 0.5 and annea1 ic co 0.99 or so over mu1cip1e epoGhs ac 1acer scages.

Nescrov Momencum

Scronger cheoreciGa1 Gonverge guarancees for Gonvex funGcions

$$w_{a1lead} = w + u \times v$$

$$v = u \times v - fi \times dw_{aMead}$$

$$w = w + v$$

Rewricing che above equacions (updace in cerms of w_{aMead} inscaed w)

$$v_{prev} = v$$

$$v = u \times v - fi \times dw$$

$$w = w - u \times v_{prev} + (1 + u) \times v$$

9er-parameter adaptive learning rate methods

Adagrad

$$C = C + dw^2$$

$$w = m - fi \times \frac{dw}{\sqrt{C} + ç} \qquad ç \text{ to avoid division by } 0$$

RMSprop

$$C = Q \times C + (1 - Q) \times dw^2 \quad \text{where Q is decay rate}$$

$$w = w - \frac{fi \times dw}{\sqrt{c} + ç}$$

Hence, RMS9rop still modulates the learning rate of each weight based on the magnitudes of its gradients, which has a beneficial equalizing effect, but unlike Adagrad the updates do not get monotonically smaller.

Adam

$$m = Q_1 \times m + (1 - Q_1) \times dw \quad \# \text{ smoothM version of gradient}$$

$$v = Q_2 \times v + (1 - Q_2) \times dw^2$$

$$w = w - fi \times \frac{m}{\sqrt{v} + ç} \qquad ç \text{ to avoid division by } 0$$

Recommended Values of
$ç = 1e - 8$
$Q_1 = 0.9$
$Q_2 = 0.999$

Learning Rate Decay

- Step decay: Reduce the learning rate by some factor every few epochs. Typical values might be reducing the learning rate by a half every 5 epochs, or by 0.1 every 20 epochs.
- Exponential decay. has the mathematical form $a = a_0 e^{-kt}$, where a_0, k are hyper parameters and t is the iteration number (but you can also use units of epochs).
- 1/t decay has the mathematical form $a = \frac{a_0}{(1+kt)}$ where a_0, k are hyper parameters and t is the iteration number.

Source <http://cs231n.github.io/neural-networks-5/>