①

COL865
Deep Learning

Optimization:-

Algorithms with adaptive learning Rate:-

(Popular Algorithms):-

① Adagrad:-

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} L(f(x^{(i)};\theta), y^{(i)})$$

$$g = \frac{1}{m}\nabla_\theta J(\theta)$$

$$r = r + g \odot g$$ ⇒ accumulation of second moment (squared gradient)

$$\Delta\theta = -\frac{\eta}{\delta + \sqrt{r}} \odot g$$

δ : small parameter for stabilization, 10^{-7}

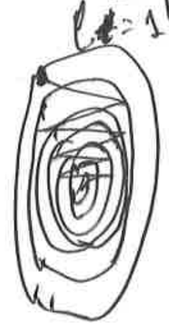
scale η by $\left[\frac{1}{\delta+\sqrt{r}}\right]$

⇒ pointwise scales g_j as:-

$$\boxed{\Delta\theta_j = -\frac{\eta\, g_j^{(t)}}{\delta + \sqrt{r_j^{(t)}}}}$$

$$r_j^{(t)} = \sum_{l=1}^{t} \left[g_j^{(l)}\right]^2$$

$$\boxed{\theta^{(t+1)} = \theta^{(t)} + \Delta\theta}$$



Parameters with large partial derivative (accumulated) have a large decrease in learning rate.

Parameters with small partial derivative (accumulated) have a slower decrease in learning rate.

→ ⊖ Amortized (happens over time).

Issue:- Premature & excessive decrease in the learning rate. Stuck in local minima.

2. RMSProp:

$r=0$ in the beginning

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} L(f(x^{(i)};\theta), y^{(i)})$$

$$g^{(t)} = \nabla_\theta J(\theta)\big|_{\theta^{(t)}}$$

$$r^{(t)} \leftarrow \rho r^{(t-1)} + (1-\rho)\, g^{(t)} \odot g^{(t)}$$

as opposed to $r^{(t)} = r^{(t-1)} + g^{(t)} \odot g^{(t)}$

past effect decays exponentially.

slower decrease in learning rate

$$\Delta\theta = -\frac{\eta}{\delta+\sqrt{r}} \odot g$$

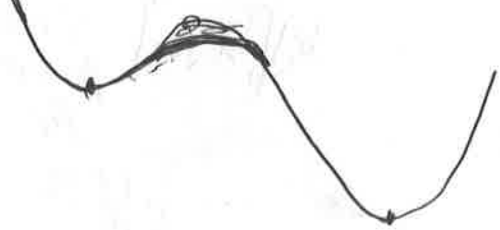
$$\theta^{(t+1)} = \theta^{(t)} + \Delta\theta$$

can combine with (Nesterov) momentum

Additional momentum term

Adadgrad vs RMSProp

Better chances of escaping local minima



$$\boxed{v^{(t+1)} = \alpha v^{(t)} - \frac{\eta}{\delta+\sqrt{r^{(t)}}} \odot g^{(t)}}$$

$$\boxed{\theta^{(t+1)} = \theta^{(t)} + v^{(t)}}$$

3.

② Adam:- (Adaptive Moments) | Kingma & Ba, 2014)

① Takes into account both the first order moment & second order moments

⇓ momentum

⇓ ~~weight decay~~ learning rate decay

$\boxed{\eta = .001}$ why? how?

$P_1 = 0.9$

$P_2 = 0.999$

momentum

⇒

First moment:- (accumulated) contrast with

More principled

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} L(f(x^{(i)};\theta), y^{(i)})$$

$\boxed{\begin{matrix} s^{(0)} = 0 \\ r^{(0)} = 0 \end{matrix}}$ ⇒ Initialized

$$g = \nabla_\theta J(\theta)\Big|_{\theta^{(t)}}$$

~~$s^{(t+1)} = P_1 s^{(t)} + (1-P_1)$~~

$$\left\{ \begin{aligned} s^{(t)} &= P_1 s^{(t-1)} + (1-P_1)\, g \\ r^{(t)} &= P_2\, r^{(t-1)} + (1-P_2)\, g \odot g \end{aligned} \right.$$] momentum

Imp:- Bias Correction factor

$$\hat{s} = \frac{s}{1-P_1^{\,t}}$$

$$\hat{r} = \frac{r}{1-P_2^{\,t}}$$

$$\Delta\theta = \frac{-\eta \hat{s}}{\delta + \sqrt{\hat{r}}} \;\left[\frac{-\eta\, s}{\underbrace{\delta + \sqrt{r}}} \right]$$

$-\eta s$ ⇒ gradient (accumulated)

$\delta + \sqrt{r}$ ⇒ ~~we~~

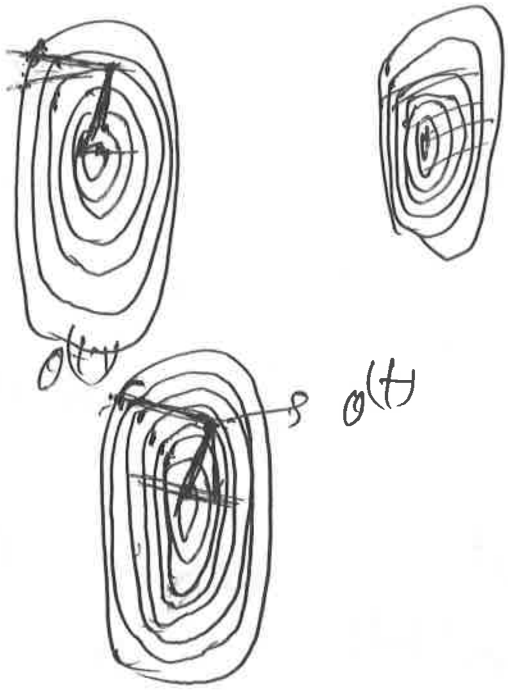
$$\theta^{(t+1)} = \theta^{(t)} + \Delta\theta$$

Step size depends on gradient (+ momentum term)

[~~bad~~ decay in learning rate over time]

Conjugate Gradient Method:-

Each successive iteration might undo the effect of previous iteration.



$d_{t-1}^{(t-1)}$:- Direction of movement at step $(t-1)$

$$\boxed{d_{\cancel{t-1}}^{\cancel{(t)}} \cdot \nabla_\theta J(\theta)\big|_{\theta^{(t)}} = 0}$$

Do a line search in this direction

Find minima.

Should we move $d^{(t)}_{\cancel{t}} = g^{(t)}$?

~~If~~ But $d^{(t)}$ may undo the effect of previous iteration

$$d^{(t)} = \nabla_\theta J(\theta)\big|_{\theta^{(t)}} + \beta_t d^{(t-1)}$$

⇓ Some left over from previous search direction

We say that $d^{(t-1)}$ & $d^{(t)}$ are conjugate given ~~G~~ ~~So~~ ~~if~~ the semi-definite matrix H if $\boxed{d^{(t-1)T} H d^{(t)} = 0}$ → useful for solving a set of linear equations

③ Progressively move towards the solution

For us:- if H is constant (function is quadratic)

$~~d^{(t)}~~$ $d^{(1)}, d^{(2)}, \ldots d^{(k)}$ be obtained such that

$$d^{(t-1)T} H d^{(t)} = 0$$

Does not undo the effect of previous iterations

then:- ~~gradient along $d^{(t-1)}$~~

if gradient along $d^{(t-1)} = 0$ at $\theta^{(t-1)}$

then gradient along ~~$d^{(t)}$ is also~~ $d^{(t-1)}$ is also 0 at $\theta^{(t)}$.

$d^{(t)}$ can be found analytically

Further:- conjugacy is preserved across iterations:-

Use information from the Hessian

$$\boxed{d^{(t')} H d^{(t)} = 0 \quad \forall t' < t}$$

if $d^{(t-1)} H d^{(t)} = 0 \quad \forall t$

Further:- $d^{(1)}, d^{(2)}, \ldots d^{(n)}$ will span the entire space.

$$d^{(t)} = \frac{\nabla_\theta J(\theta^{(t)})^T \nabla_\theta J(\theta^{(t)})}{\nabla_\theta J(\theta^{(t-1)})^T \nabla_\theta J(\theta^{(t-1)})}$$

$\Rightarrow$ After n steps, we will reach the ~~minima~~ following conjugate directions

$\boxed{\text{Scaled conjugate gradient method}}$

Non-linear conjugate gradient method : Reset $\beta^{(t)} = 0$ ~~always~~ occasionally.

BFGS:-

Boyden
Fletcher
Goldfarb
Shanno

Newton's method:-

$$\theta^{(t+1)} = \theta^{(t)} - H^{-1}g$$

$\Downarrow$

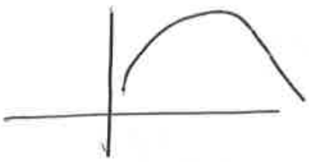
Approximate by another matrix $M^{(t)}$ which successively becomes better.

LBFGS

Do not store the matrix (approximation) explicitly in memory.

Batch normalization:- Do it later

Newton's method:- escaping saddle points:-



Scale down the grad movement in directions in which gradient moves upwards.

$$\theta^{(t+1)} = \theta^{(t)} - \left[\left[H\left(f(\theta^{(t)})\right) + \underbrace{\alpha I}_{\text{Attempt to make eigenvalues +ve.}}\right]^{-1}\right] \overbrace{\nabla_\theta f(\theta^{(t)})}^{g}$$

④ Parameter Initialization Strategies :- close to zero! - Randomly (range)??

↳ ① Initializing W's →
② Initializing biases b's
③

$\boxed{\text{Read section 8.4}}$

$x^{(t)} \cancel{= W^{T}}$ → Activation function

$\boxed{x^{(t)} = h\left[W^{T} x^{(t-1)} + b^{(t)}\right]}$