1

Challenges on Neural Network Optimization:-

Col 865 . Deep Learning

Aug 28, 2017

~~Ba~~.

Key Problem:-

$$\tilde{J}(\theta) = \frac{1}{m}\sum_{i=1}^{m} L\left[f(x^{(i)};\theta);y^{(i)}\right]$$

⇒ proxy for generalization error

Expected loss over the training set

⇓ over training examples

$$\left[\tilde{J}(\theta) = J(\theta) + \underbrace{\tfrac{1}{2}\theta^T\theta}_{\alpha.\,\Omega(\theta)}\right.$$ ⇒ regularizer

~~May not always~~

Not exactly an optimization problem

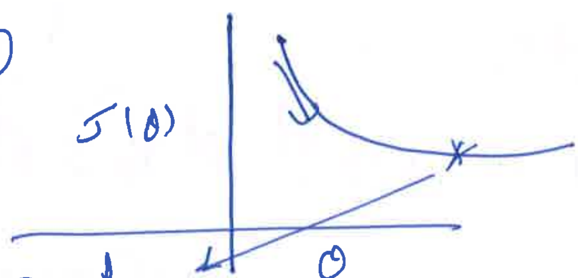
$$J^*(\theta) = E_{(x,y)\sim p^*_{data}}\left[L\left(f(x;\theta);y\right)\right]$$

$$\boxed{\theta^{(t+1)} \leftarrow \theta^t - \eta.\nabla_\theta J(\theta)}$$ ⇒ gradient update rule

How do we minimize a function?

Understand the issues:-

1



Reach $\boxed{g \equiv 0}$

1 Gradient Descent

Find the gradient ~~along~~

$J'(\theta) \equiv g$ &

& move opposite to direction of g.

$\nabla_\theta J(\theta) \equiv g$

Direction of steepest ascent

Key challenge:-

How much to move along $-g$?

Learning Rate: η.

(I) Fixed / Static

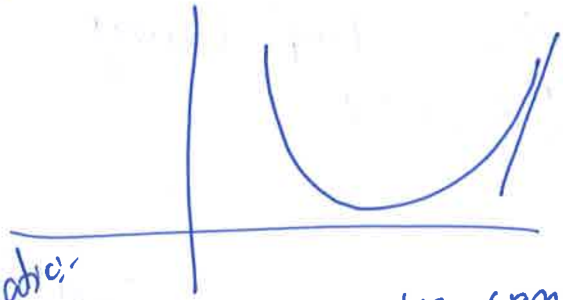
(II) Dynamic

(III) Should we always move along g? Alternate methods ⇒ (1) Adding momentum (2) conjugate gradient

Second order Methods:-

$\mathbb{I}$ when $\theta \in \mathbb{R}$

$$\theta^{(t+1)} = \theta^{(t)} - \frac{f'(\theta^{(t)})}{f''(\theta^{(t)})}$$



$\Rightarrow J'(\theta)$

If $J(\theta)$ is quadratic, optimum in one step.

we want θ^* s.t $J'(\theta^*)=0$

$\Rightarrow$ Newton's method to find zeros of a function.

$$\theta^{(t+1)} = \theta^{(t)} - \underbrace{H^{-1}}\nabla_\theta J(\theta)\big|_{\theta^{(t)}} \longrightarrow \text{computationally expensive}$$

$\boxed{H \equiv \text{Matrix of second order derivatives}}$

Understanding $H \equiv$ curvature Becomes critical to any kind of optimization even gradient based.

---

* Illustration:-

θ using quadratic approximation:-

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

$$J(\theta) = J(\theta^{(0)}) + (\theta-\theta^{(0)})^T \underbrace{\nabla_\theta J(\theta^{(0)})}_{g}$$

$$+ \frac{1}{2}(\theta-\theta^{(0)})^T H (\theta-\theta^{(0)})$$

Now, $\boxed{\theta = \theta^{(0)} - \epsilon\eta g} \rightarrow$ using gradient descent

$$J(\theta^{(0)} - \eta g) = J(\theta^{(0)}) - \eta g^T g + \frac{1}{2}\eta g^T H g$$

②

$$J(\theta^{(0)} - \eta g) = J(\theta^{(0)}) - \eta g^T g + \tfrac{1}{2}\eta^2 g^T H g$$

At optimum values of

$J(\theta^{(0)} - \eta g) - J(\theta^{(0)})$ is minimum as a function of $\eta \Rightarrow$

$$g^T g = \eta\, g^T H g$$

$$\Rightarrow \cancel{\eta = \frac{g^T H g}{g^T g}}$$

Note:- OCR then:-

$\eta = \frac{1}{J''(\theta)} \Rightarrow$ Newton's update $\Leftarrow$ $\theta^{(t+1)} = \theta^{(t)} - \frac{J'(\theta)}{J''(\theta)}$

$\Leftarrow \cancel{\eta} g^T g = \eta^2 g^T H g$

$$\boxed{\eta = \frac{g^T g}{g^T H g}} \rightarrow \text{scales } g \text{ in the space of eigenvalues of}$$

$\boxed{g = \alpha_i v_i}$ If g aligns with an eigenvalue $\Rightarrow$ v_i, $g^T H g = \lambda_i \alpha_i^2$

Recall:- $\boxed{H = Q^T \Lambda Q}$ $\Downarrow$ Eigenvalue decomposition

$\boxed{\eta = \frac{1}{\lambda_i}}$ because $\Rightarrow$ problem if $\cancel{\frac{|\lambda_{min}|}{|\lambda_{max}|}}$

$$\boxed{g \equiv \alpha_1 v_1 + \alpha_2 v_2 + \dots \alpha_n v_n}$$

$$\Rightarrow \left[\sum_i \alpha_i v_i\right] H \left[\sum_i \alpha_i v_i\right]$$

$\frac{|\lambda_{max}|}{|\lambda_{min}|}$ is very large, then g may not be the optimal direction

$\Leftarrow$ then conditioning number

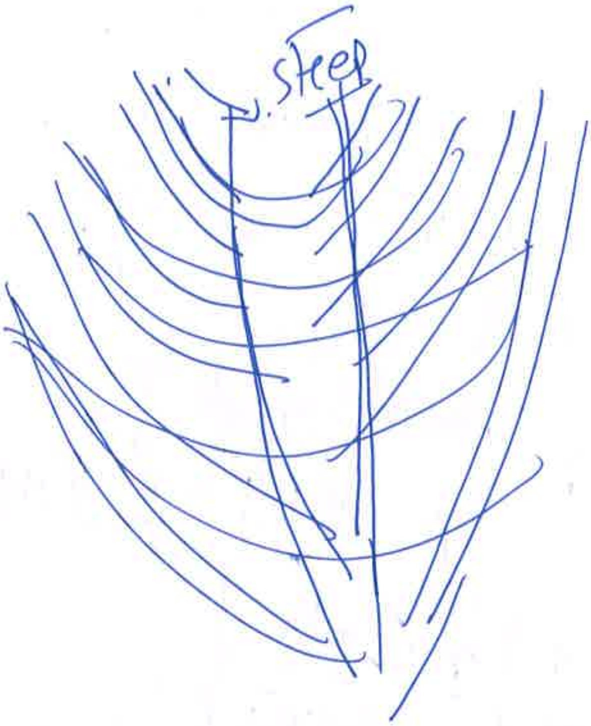
$\boxed{H v_i = \lambda_i v_i}$

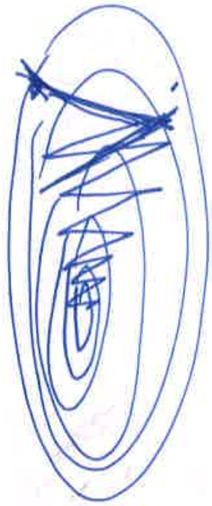
$$g^T H g = \left[\sum_i \alpha_i v_i\right]\left[\sum_i \lambda_i \alpha_i v_i\right]$$

Direction second derivative? $\Leftarrow$ weighted sum of eigenvalues

$$d^T H d = d^T Q \Lambda Q^T d$$
$$= (Q^T d)^T \Lambda (Q^T d)$$

illustration:-





gradient may waste a lot of time in reaching the minima

Solution:-

Second order methods?

if ~~curvatures don~~"

if H does not change as much :- Problem if curvature also varies.

Further:-

$$\eta = \frac{g^Tg}{g^THg}$$

g^Tg → large gradient → ~~greater~~ greater step

g^THg → High curvature → shorter step

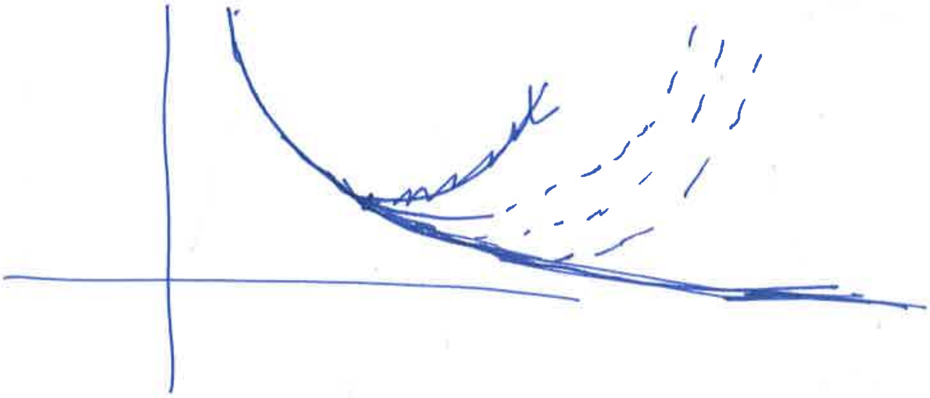
:- Quadratic approximation

other issues:-

if g^THg ~~decrea~~ decreases very fast ⇒ under estimate the step size.

learning very slow

③



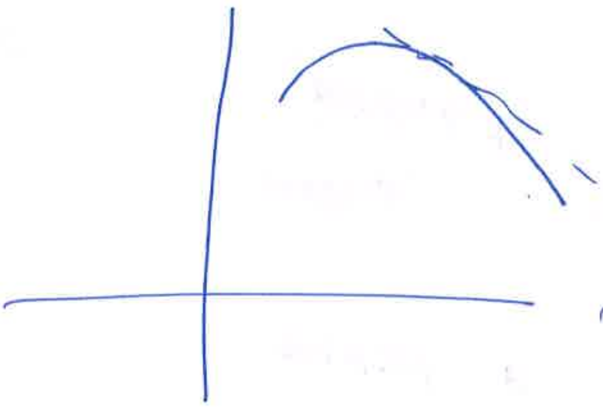
If $g^T H g$ increases very fast:- over at most the step size:- Overshoot the minima.

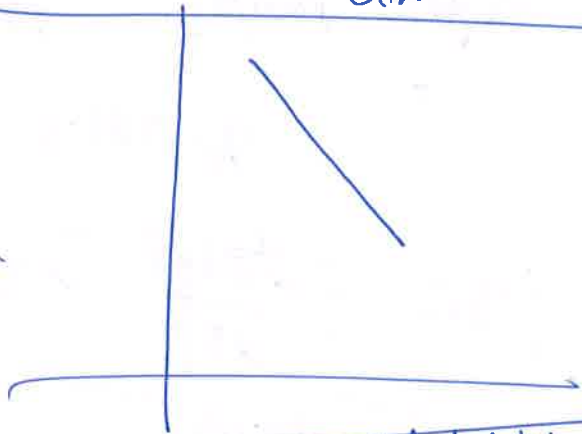
Newton's method:- exact solution if function is quadratic.

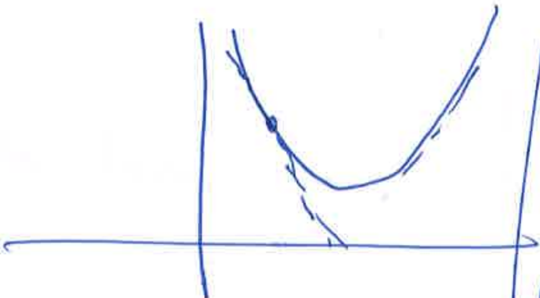
~~Other issues:-~~

Curvature Related:-

:- Multiple nonlinearities - sharp transitions in curvature

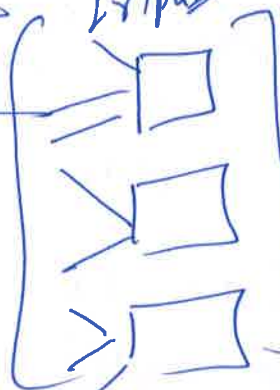






Model identifiability problem:- Unique combination of weights which result in global minima

Inputs



units exchange units

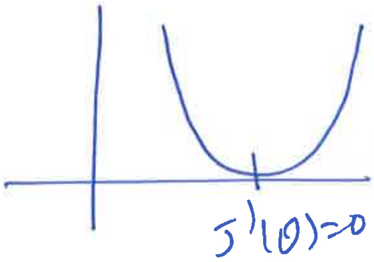
Randomly initialize around zero. Break the symmetry.

Similarly:-

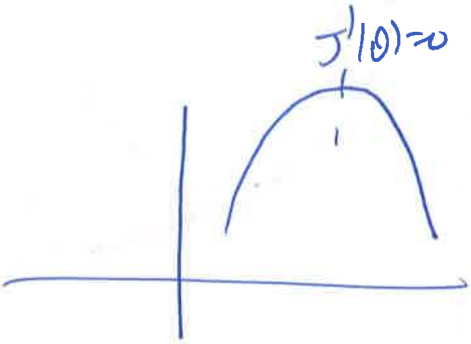
$g(z) = |z|$ (Absolute value rectifier)

Multiply incoming weights by 2 & outgoing weights by $1/2$.

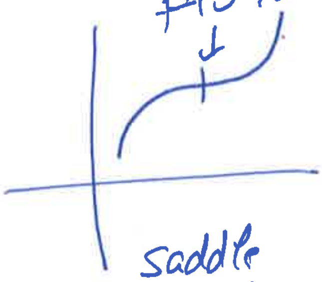
Another Issue:-



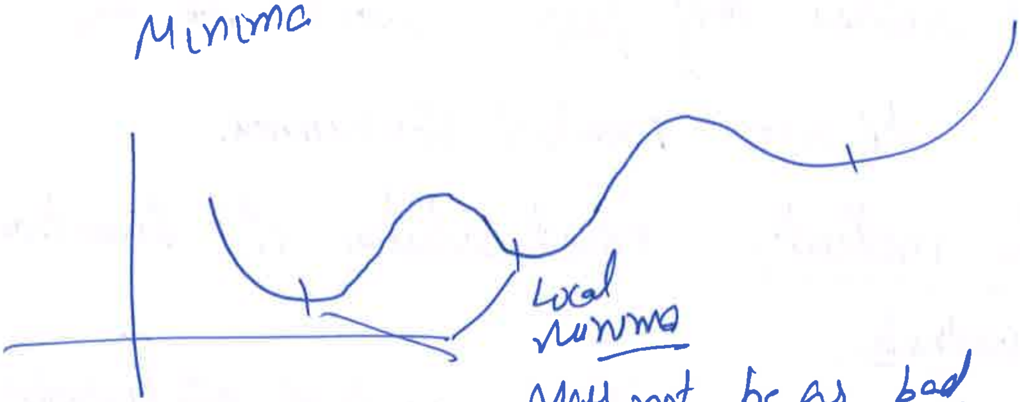
Minima



Maxima



saddle point



Local minima

May not be as bad

Saddle points:-

In multiple dimensions:-

$$\nabla_\theta J(\theta)=0$$

$$\cancel{\exists i \quad \frac{\partial^2 J(\theta)}{\partial \theta_i^2} > 0 \rightarrow \text{Minima}} \quad \cancel{\forall \text{ some}}$$

$$\cancel{\exists j \quad \frac{\partial^2 J(\theta)}{\partial \theta_j^2} \leq 0 \rightarrow \text{Maxima}}$$

$\boxed{d^T H d \geq 0 \quad \forall d}$:- Local Minima

$\lambda_i \geq 0 \quad \forall i$

$d^T H d \leq 0 \quad \forall d$:-

$\lambda_i \leq 0 \quad \forall i$ local maxima

otherwise saddle point $\Rightarrow$ Much more likely to occur than minima/maxima

4

Algorithms:-

Stochastic gradient Descent:-

- Batch Gradient Descent
- Mini-batch gradient Descent.

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

$= E[$

$J_{st}(\theta) = \frac{1}{r}\sum_{i=1}^{r} L[f(x^{(i)};\theta), y^{(i)}]$

r ⇓ mini-batch size

minibatch > 1

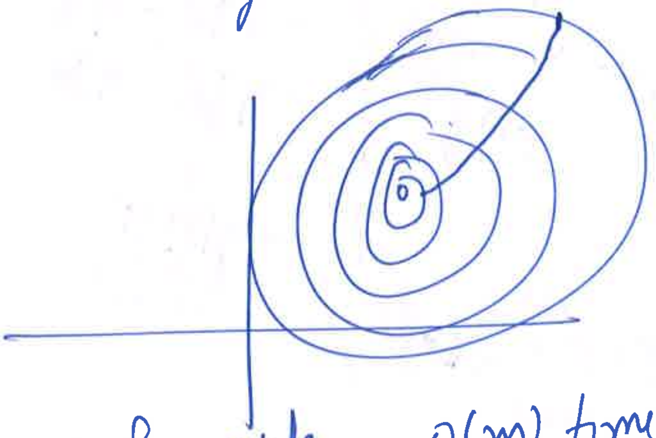
~~Advantages:~~

$\nabla_\theta J(\theta)$ $O(m)$ time

$O(r)$ time

$\nabla_\theta J_{st}(\theta)$ ⇓ Approximation

Approximate loss using a smaller number of examples.



Each update: $O(m)$ time

only movement in the rough direction is important

Each update: $O(r)$ time

Guaranteed to converge to local optima with sufficiently small η.

Considerations in choosing γ?

↳ γ :- small gradient approximate

↳ γ :- large Increased computation time
⇒ can be parallelized

→ γ :- small :- approximates generalization error (until you do a single pass)

Section 8.3: -
- → variations to ~~#~~ gradient based methods
- → Second order methods.

Momentum:-

$\nabla_\theta J(\theta)$:- Gradient

Update Rule:

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

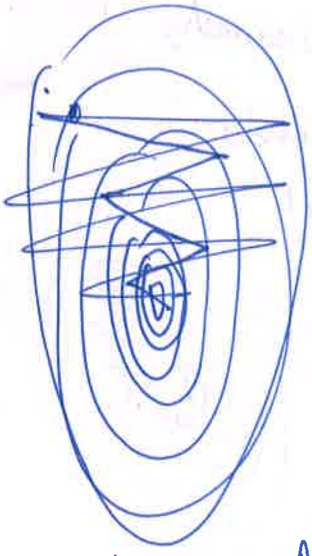
$$\boxed{v = \alpha v - \eta \nabla_\theta J(\theta)}$$

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

→ decay (exponentially decays wrt # iterations).

⇒ Accumulation of previous gradients

[Use Along with SGD]



$$v^{(t+1)} = \alpha v^{(t)} - \eta g$$

$$= \alpha(\alpha v^{(t-1)} - \eta g) - \eta g$$

If $\nabla_\theta J(\theta)$ are aligned

$= g$

then

$-\eta g$:-

$-\eta g\alpha - \eta g$

$-\eta g\alpha^2 - \eta g\alpha - \eta g$

$-\eta\left(\frac{\|g\|}{1-\alpha}\right)$

Nesterov Momentum :-

$$\boxed{\begin{aligned} v^{(t+1)} &= \alpha v^{(t)} - \eta \nabla_\theta J(\theta^{(t)} + \alpha v^{(t)}) \\ \theta^{(t+1)} &= \theta^{(t)} + \alpha v^{(t+1)} \end{aligned}}$$