Lecture #5: Optimization

GD: $\theta^{n+1} = \theta^{n} - \eta \nabla_{\theta} L(\theta^{n})$

Newton: $\theta^{n+1} = \theta^{n} - \eta H_{n}^{-1} \nabla_{\theta} L(\theta^{n})$

Remarks:

1.) Choosing η is often an "art" (GD).

2.) $\theta \in \mathbb{R}^{d}$, it may not be smart to use the same learning rate for all of them.

e.g. fitting a univariate Gaussian, $\theta := \{\mu, \sigma^2\}$

3.) Exact Hessians are often very expensive to compute / store / invert. $\longrightarrow$ Quasi-Newton

$\longrightarrow$ 4.) Scalability to big data

Stochastic gradient descent:

In many ML applications the loss function factorize accross data-points:

$$L(\theta) = \frac{1}{n} \sum_{i=1}^{n} \underbrace{L_i(\theta)} \quad \text{(see eg. linear regression, NNs)}$$

GD: $\theta^{n+1} = \theta^{n} - \eta \nabla_{\theta} L(\theta^{n})$

$$= \theta^n - \eta \underbrace{\frac{1}{n}\sum_{i=1}^{n} \nabla_\theta L_i(\theta^n)}_{\text{exact gradient}}$$

SGD : $\theta^{n+1} = \theta^n - \eta \nabla_\theta L_i(\theta^n) \leftarrow$ Robins - Monroe

Mini-batch SGD : $\underline{\theta^{n+1} = \theta^n - \eta \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta L_i(\theta^n)}$, $\underline{m} \ll n$

<u>Epoch</u> : # iterations needed to perform on full sweep through our data-set (SGD)

$\frac{n}{m}$

Gradient descent variants :

GD with momentum :

$$\left.\begin{aligned} u^{n+1} &= \gamma u^n + \eta \nabla_\theta L(\theta^n) \\ \theta^{n+1} &= \theta^n - u^{n+1} \end{aligned}\right\}$$

$\eta : 10^{-3}$

$\gamma = 0 \longrightarrow$ GD

$\gamma = 0.9$

Nesterov Accelerated Gradient :

$$u^{n+1} = \gamma u^n + \eta \nabla_\theta L(\theta^n - \gamma u^n)$$

$$\theta^{n+1} = \theta^n - u^{n+1}$$

Adaptive Learning rate methods :

RMS Prop : $\underbrace{\mathbb{E}[g^2]_n}_{\text{variance of the gradients}}$:= average of the square gradi at iteration n.

$g_n := \nabla_\theta L(\theta^n)$

$$\rightarrow \underset{d\times 1}{\mathbb{E}[g^2]_{n+1}} = \gamma \mathbb{E}[g^2]_n + (1-\gamma) g_n^2 \quad , \gamma \sim 0.9$$

$$\theta^{n+1} = \theta^n - \frac{\eta}{\sqrt{E[g^2]_{n+1} + \varepsilon}} \cdot g_n$$ (standard SGD with adaptive learning rate

Adam: (adaptive moment estimation)

$$m^{n+1} = \beta_1 m^n + (1-\beta_1) g^n$$: Estimate of the 1^{st} m... of the gradien...

$$v^{n+1} = \beta_2 v^n + (1-\beta_2) g_n^2$$: – || – of the 2^{nd} mo...

$$\hat{m}^n = \frac{m^n}{1-\beta_1^n}, \quad \hat{v}^n = \frac{v^n}{1-\beta_2^n}$$

$$\underline{\theta}^{n+1} = \theta^n - \frac{\eta}{\sqrt{\hat{v}^{n+1} + \varepsilon}} \hat{m}^{n+1}$$

Parameters: θ

State of optimizer: e.g. $\{\theta^n, u^n\}$ momentum, $\{\theta^n, \underline{\hat{m}}^n, \underline{\hat{v}}^n\}$ Adam