cs 480/680

2/24/26

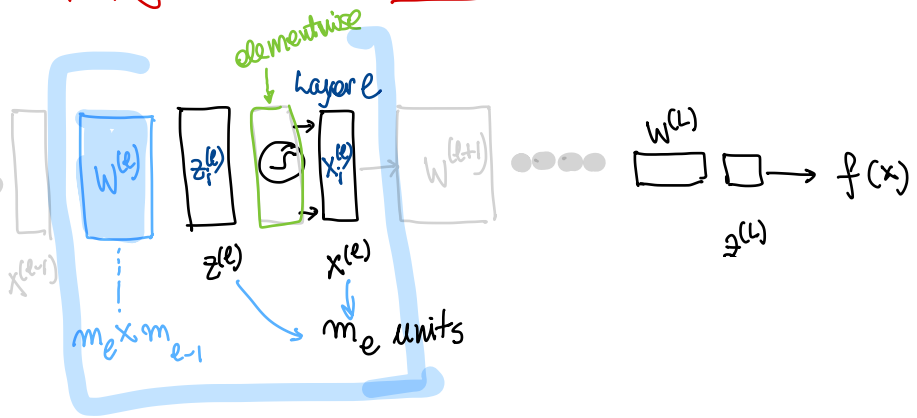
Lecture 12

HW5 W7 due W8
[HW6 W8]
Q2 3/5 or W9
HW7 W10 due W11
Q3 W11
[HW8 W11]

Where we left off before reading week...

Backpropagation

Training a multi layer network



Parameters $W = \{ W^{(1)}, \dots W^{(L)} \}$ $W \in \mathbb{R}^p$

Data $\mathcal{D} = \{ (x^i, y^i), i = 1:n \}$

TRAIN: by G.Descent on loss $\mathcal{L}$

1. Init W with small random values $W_{ij}^{(\ell)} \sim N(0, \sigma^2 \frac{1}{m_\ell})$ σ^2 small
2. for $t = 1, 2, \dots$

$$W^{t+1} \leftarrow W^t - \eta \frac{\partial \mathcal{L}(W^t)}{\partial W} \quad (g = \text{gradient}) \quad \Leftrightarrow \quad w_j^{t+1} \leftarrow w_j^t - \eta \frac{\partial \mathcal{L}}{\partial w_j}$$

step size

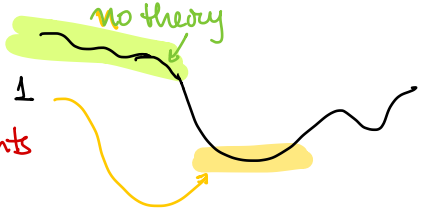
y^i, x^i ← training ex. i

$x^{(\ell)}$ ← layer ℓ

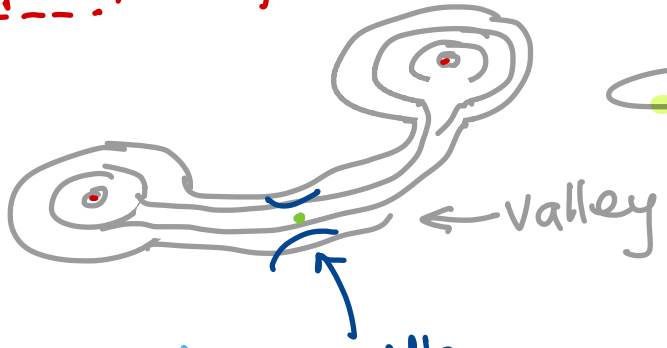
$x_j^{(\ell)}$ ← unit j

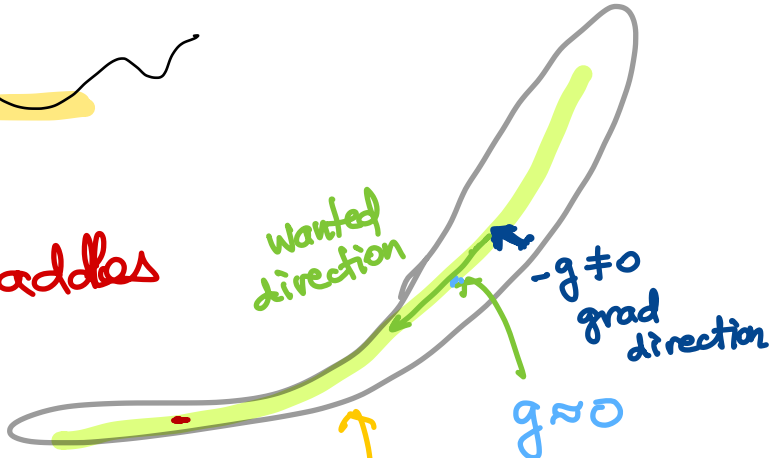
$x_2^{(0)i}$ = input value layer 0, 2nd attribute i-th example

I comp / iter ✓ np
T #iterations ← GD is order 1
S saturation / vanishing gradients
L local minima
O verfitting

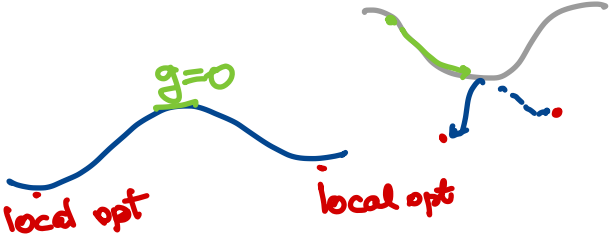


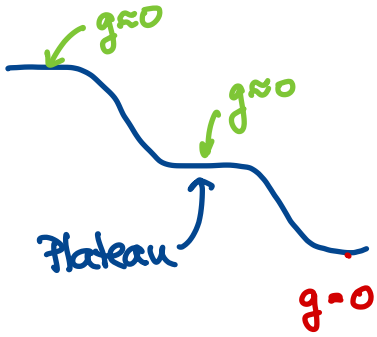
Local Optima, Valleys, Plateaus, Saddles





g^t = gradient at t





Stochastic Grad Descent

$$w_j^{t+1} \leftarrow w_j^t - \eta \frac{1}{n} \sum_{i=1}^{n} \frac{\partial \mathcal{L}(y^i, f(x^i, w^t))}{\partial w_j}$$

$j = 1:p$

g_i contribution of (x^i, y^i)

for any t: $g = \frac{1}{n} \sum_i g_i$

Idea sample $i \sim \text{unif}[1:n]$ $\times n'$ times $\Rightarrow$ $\mathcal{B} = \{(x^i, y^i),\ i = 1:n'\} \subset \mathcal{D}$ "batch"

$$g_{\mathcal{B}} = \frac{1}{n'} \sum_{i \in \mathcal{B}} g_i \approx g$$

$\sigma^2 = \text{Var}\{g_i\}$

$g = \text{mean}\{g_i\}$

random variable: $g_{\mathcal{B}} = g + \varepsilon$ ("noise")

$\text{mean}(\varepsilon) = 0$

$\text{Var}(\varepsilon) = \frac{\sigma^2}{n'}$

$\text{std}(\varepsilon) = \frac{\sigma}{\sqrt{n'}}$

in theory $n' = 1$

in practice $\longrightarrow n' > 1$

Benefits 1) Faster computation / iteration (I) $n'p$ vs np

2) —||— overall $T n' p$ vs Tnp

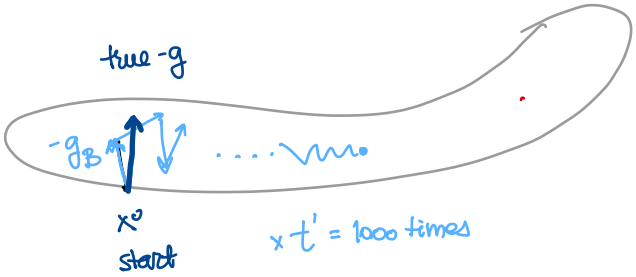
3) Randomness: ~~avoid~~ escape saddles, shallow local minima,

Why SGD "faster" than GD ?

$n = 10^6$

$n' = 1000$

$t' = \frac{n}{n'} = 1000 = \#\mathcal{B}'s$ in $\mathcal{D}$



Theoretical convergence

near W^* optimum
- increase $n' \nearrow n$
- average $g^t_{\mathcal{B}'}$ over last $\tilde{T}$ iterations
- $\eta \searrow 0$ (like $\sim \frac{1}{t}$)

Momentum — can we mimic 2nd order methods cheaply?

GD : 1 st order approx

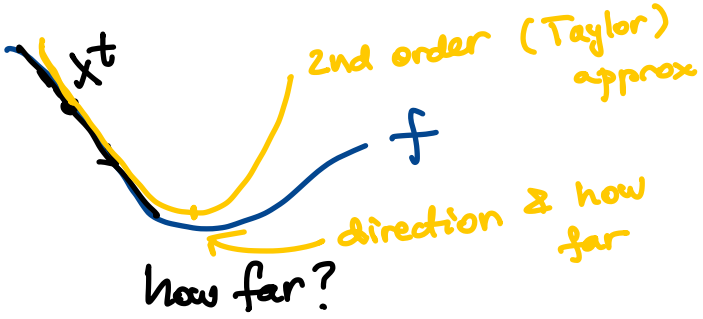
$g \in \mathbb{R}^p$

Newton : 2nd order

$$\left[\frac{\partial^2 \mathcal{L}}{\partial w_j w_{j'}} \right]_{j,j'=1:p} \in \mathbb{R}^{p \times p}$$

Hessian

TOO EXPENSIVE



Heavy Ball / Momentum $\gamma \in (0,1)$

$$w_j^{t+1} \leftarrow w_j^t - \eta g_j^t + (1-\gamma)(w_j^t - w_j^{t-1})$$

$g_j^t = \frac{\partial \mathcal{L}^t}{\partial w_j}$

previous step

$$w_j^{t+1} \leftarrow w_j^t - \gamma\eta g_j^t + (1-\gamma)(w_j^t - w_j^{t-1})$$

$$w_j^t - w_j^t = \gamma\eta g_j^{t-1} + (1-\gamma)(w_j^{t-1} - w_j^{t-2})$$

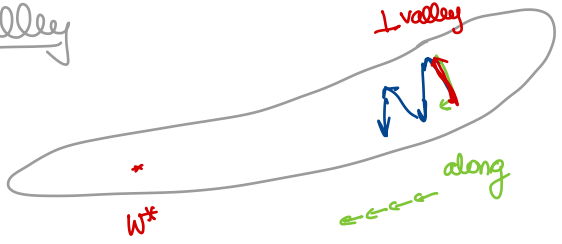
$$w_j^{t-1} - w_j^{t-2} = \gamma\eta g_j^{t-2} + (1-\gamma)(\dots)$$

$$\dots$$

Ex

$$w_j^{t+1} \leftarrow w_j^t - \gamma\eta \{ g_j^t + (1-\gamma) g_j^{t-1} + (1-\gamma)^2 g_j^{t-2} + \dots \}$$

Valley



$\nearrow\nearrow\nearrow \approx 0$

Plateau

g^t with momentum

$g^t \approx 0$

small local op