Lec 10 SGD

1. GD: $\underline{w}_{k+1} = \underline{w}_k - \varepsilon_k \nabla E_{in}(\underline{w}_k)$

compute gradient of all examples

SGD: At each iteration k, select uniformly randomly the nth example from $\{1, 2, \cdots N\}$ and set

$$\underline{w}_{k+1} = \underline{w}_k - \varepsilon_k \nabla e_n(\underline{w}_k)$$

compute gradient of a single example

↳ Note ①:

Why we can use SGD if "randomly" chosen?

$\mathbb{E}[\nabla e_n(\underline{w}_k)]$ // Expectation

$= Pr(n=1) \cdot \nabla e_1(\underline{w}_k) + Pr(n=2) \cdot \nabla e_2(\underline{w}_k) + \cdots + Pr(n=N) \nabla e_N(\underline{w}_k)$

$= \frac{1}{N} \sum_{n=1}^{N} \nabla e_n(\underline{w}_k)$ // $\frac{1}{N}$ is the $Pr(n=i)$ since uniform distribution

$= \nabla E_{in}(\underline{w}_k)$

∴ Even randomly select, the mean value is in the direction that we want to go.

i.e. $\nabla e_n(\underline{w}_k) = \text{mean value} + \text{noise}$

the right (GD) direction

$= \nabla E_{in}(\underline{w}_k) + \text{noise}$

⇒ A noisy estimate of the true gradient

↳ Note ②:

- Full GD ("batched" GD) has complexity $O(Nd)$ per iteration to compute $\nabla E_{in}(\underline{w})$. N is large
- But SGD has $O(d)$ ⇒ Can do many more iterations than GD
- often there is high reduncy in big dataset.

  ⇒ No need to consider all data points in one step

2. Minibatch SGD:

↳ At each iteration, select M examples uniformly randomly:

$\underline{x}(1), \underline{x}(2) \cdots x(M)$

Then update: $\underline{w}_{k+1} = w_k - \frac{\varepsilon_k}{M} \sum_{n=1}^{M} e_n(\underline{w}_k)$

↳ Purpose:

- get a better estimate for $\nabla E_{in}(\underline{w})$
- If choose M = #. of cores of CPU

  ⇒ Parallization across multiple processor.

3. Logistic Regression example:

$$\nabla e_n(\underline{w}) = \frac{-y_n X_n}{1 + e^{y_n \underline{w}^T \underline{x}_n}}$$

$$\therefore \text{SGD}: \quad \underline{w}_{k+1} = \underline{w}_k + \varepsilon_k \frac{y_n X_n}{1 + e^{y_n \underline{w}^T \underline{x}_n}}$$

// close connection to the PLA

↳ Suppose $(\underline{x}_n, y_n)$ is misclassified.

i.e. $y_n \underline{w}_k^T x_n \leq 0$ (deduced previously)

$\therefore e^{y_n \underline{w}_k^T x_n}$ is small

Then the SGD update rule becomes:

$$\underline{w}_{k+1} = \underline{w}_k + \eta_k y_n \underline{x}_n$$

↳ Suppose $(\underline{x}_n, y_n)$ is correctly classified:

// Recall that the PLA will not do anything

$\therefore y_n \underline{w}^T \underline{x}_n > 0$

$\therefore e^{y_n \underline{w}^T \underline{x}_n}$ is large

$\therefore$ The SGD update rule becomes:

$$\underline{w}_{k+1} \approx \underline{w}_k$$

$\therefore$ PLA is an exterme case of SGD + logistic regression