DSA5202 LEC4 $\longrightarrow$ Neural Network

1. Shallow NN

→ formulation:

$$f(x) = \sum_{j=1}^{M} v_j\, \sigma(w_j^T x + b_j)$$

$$= v^T \sigma(Wx + b)$$

$x \in \mathbb{R}^d \quad v \in \mathbb{R}^M$

$$\begin{cases} W \in \mathbb{R}^{M \times d} \\ b \in \mathbb{R}^M \end{cases}$$

→ $\boxed{\text{universal approximation theorem}}$ → on compact set

2. Deep NN

→ formulation:

$$\begin{cases} f(x) = v^T f_T(x) \\ f_{t+1}(x) = \sigma(W_t f_t(x) + b_t) & t = 0, \dots, T-1 \end{cases}$$

$\boxed{f_0(x) = x}$

$\boxed{T = 1}$ → degrade to shallow NN

3. Training NN

$$\hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \ell(f_\theta(x_i), y_i) \rightarrow \boxed{\text{ERM}}$$

→ $\boxed{\text{using GD / SGD / momentum GD}}$

Recap: our deepest goal: $\boxed{\min_{\theta \in \Theta} \mathbb{E}_{(x,y) \sim \mathcal{D}} [\ell(f_\theta(x), y)]}$

4. Expolration on $\begin{cases} \text{width } M \\ \text{depth } T \end{cases}$ of DNN

$\rightarrow$ Define $H_{M,T} = \{ \text{NN with width } M \text{ \& depth } T \}$

$$= \{ f : f(x) = V^T f_T(x) \}.$$

where : $f_{t+1}(x) = \sigma(W_t f_t(x) + b_t)$

$$\begin{cases} W_t \in \mathbb{R}^{M \times M}, & t = 1, 2, \ldots, T-1 \\ W_0 \in \mathbb{R}^{M \times d} \end{cases}$$

Richness of Hypothesis Space

generally not true

$\Rightarrow$ we want to show $\begin{cases} ① \text{ Width} \Rightarrow H_{M,T} \subseteq H_{M+1,T} \\ ② \text{ Depth} \Rightarrow H_{M,T} \subseteq H_{M,T+1} \end{cases}$

① $\forall f \in H_{M,T}$, $f(x) = V^T f_T(x)$ & $f_{t+1}(x) = \sigma(W_t f_t(x) + b_t)$

construct $\tilde{f}$ as follows :

$$\begin{cases} a) & \tilde{f}_1(x) = \sigma\left(\begin{bmatrix} W_0 \\ 0 \end{bmatrix} x + \begin{bmatrix} b_0 \\ 0 \end{bmatrix} \right) \\ b) & \tilde{f}_{t+1}(x) = \sigma\left(\begin{bmatrix} W_0 & 1 \\ 0 & 0 \end{bmatrix} \tilde{f}_t(x) + \begin{bmatrix} b_t \\ 0 \end{bmatrix} \right) \\ c) & \tilde{f}(x) = \begin{pmatrix} V \\ 0 \end{pmatrix}^T \tilde{f}_T(x) \end{cases}$$

It is obvious that $f(x) \equiv \tilde{f}(x) \in \mathcal{H}_{M+1,T}$

[trick] : utilize the dummy dimension $(M+1)$

② $\forall f \in H_{M,T}$, $f(x) = V^T f_T(x)$ & $f_{t+1}(x) = \sigma(W_t f_t(x) + b_t)$

suppose $\sigma(\cdot)$ is ReLu

Construct $\hat{f}$ as follows :

$$\begin{cases} a) & \hat{f}_T(x) = f_T(x) \end{cases}$$

b) $\hat{f}_{T+1}(x) = \sigma(\widetilde{W}_T f_T(x) + \tilde{b}_T)$

where $\begin{cases} \widetilde{W}_T = I_M \in \mathbb{R}^{M \times M} \\ \tilde{b}_T = \text{sufficiently large} \end{cases}$

then $\hat{f}(x) = V^T \hat{f}_{T+1}(x)$

$= V^T f_T(x) + V^T \tilde{b}_T$

$\Rightarrow$ $H_{M,T} \subseteq H_{M,T+1} + \text{costant}$ → only holds for ReLu

5. $\boxed{\text{Gradient Calculation for DNN}}$ → via Back propagation

→ objective : $\min_\theta \frac{1}{N} \sum_{i=1}^{N} \ell(f_\theta(x_i), y_i)$

$$\boxed{\theta = \{ \{W_t\}_{t=0}^{T-1}, \{b_t\}_{t=0}^{T-1}, V \}}$$

→ we only consider gradient derivation for one data (x, y)

omit bias

$\Downarrow$

$$\begin{cases} f(x) = V^T f_T(x) := x_{T+1} = g_T(x_T, W_T) \\ f_{t+1}(x) = \sigma(W_t f_t(x) + b_t) := x_{t+1} = g_t(x_t, W_t) \end{cases}$$

prediction of x

label of x

Our interest is: $\nabla_{W_t} \ell(x_{T+1}, y)$

$= \nabla_{W_t} x_{t+1} \cdot \nabla_{x_{t+1}} \ell(x_{T+1}, y)$

$= \nabla_{W_t} g_t(W_t, x_t) \cdot \nabla_{x_{t+1}} \ell(x_{T+1}, y)$

(Here, we denote $p_t = \nabla_{x_t} \ell(x_{T+1}, y)$)

$$= \underbrace{\nabla_{W_t} g_t(W_t, x_t)}_{\text{easy to compute}} \cdot \underbrace{p_{t+1}}_{\text{manage}}$$

$$p_t = \nabla_{x_t} \ell(x_{T+1}, y)$$

$$= \nabla_{x_t} x_{t+1} \cdot \nabla_{x_{t+1}} \cdot \ell(x_{T+1}, y)$$

$$= \underbrace{\nabla_{x_t} g_t(W_t, x_t)}_{\text{easy to compute}} \cdot p_{t+1}$$

$\boxed{\text{Boundary condition}}$

$\Downarrow$

$$\boxed{p_{T+1} = \nabla_{x_{T+1}} \ell(x_{T+1}, y)}$$

(starting from this)

BP Algorithm Summary:

Algorithm 1: Backpropagation for FC-DNN

$x_0 = x \in \mathcal{R}^d$ **for** $t = 0, 1, \ldots, T$ **do**
  $x_{t+1} = g_t(x_t, W_t) = \sigma(W_t^\top x_t)$;
end

forward pass (function value)

Set $p_{T+1} = \nabla_{x_{T+1}} \ell(x_{T+1}, y)$;
for $t = T, T-1, \ldots, 1$ **do**
  $\nabla_{W_t} \ell(x_{T+1}, y) = p_{t+1}^\top \nabla_{W_t} g_t(x_t, W_t)$;
  $p_t = [\nabla_{x_t} g_t(x_t, W_t)]^\top p_{t+1}$;
end

backward propagation (gradient)

return $\{\nabla_{W_t} \ell(x_{T+1}, y) : t = 0, \ldots, T\}$