DSA5105 Homework 3 A0236307J

[FCNN] $\longrightarrow$ d-n-1 MLP

$$f(x) = \sum_{i=1}^{n} v_i \sigma(w_i^T x) \qquad \sigma(\cdot) \to \mathrm{Relu}(\cdot)$$

$$f: \mathbb{R}^d \longrightarrow \mathbb{R}$$

(a) Analysis: f is positive homogeneous

$\Leftrightarrow$ $g_i(x) := \sigma(w_i^T x)$ is positive homogeneous for all i

$\Leftrightarrow$ $\sigma(z) = \max\{0, z\}$ positive homogeneous

Pf: firstly, show $\sigma(z)$ is positive homogeneous

1. when $z<0$, then $\sigma(\lambda z) = 0 = \lambda \cdot 0 = \lambda \sigma(z)$
2. when $z \geq 0$, then $\sigma(\lambda z) = \lambda z = \lambda \cdot \sigma(z)$

This shows that $\sigma(z)$ is positive homogeneous

Then
$$f(\lambda x) = \sum_{i=1}^{n} v_i \sigma(w_i^T(\lambda x))$$

$$= \sum_{i=1}^{n} v_i \sigma(\lambda \cdot w_i^T x)$$

$$= \sum_{i=1}^{n} \lambda \cdot v_i \sigma(w_i^T x) \qquad (\sigma(\cdot) \text{ is positive homogeneous})$$

$$= \lambda \cdot f(x)$$

$\boxed{\text{Reason 1}}$

(b) From (a), we know that $f \in \{\text{positive homogeneous function class}\}$

However, we can check the scale of $f^*(\lambda x)$ & $\lambda f^*(x)$

$$\lim_{x \to +\infty} \frac{\lambda f^*(x)}{f^*(\lambda x)} = \frac{\lambda e^x}{e^{\lambda x}} = \begin{cases} +\infty, & \lambda < 1 \\ 1, & \lambda = 1 \\ 0, & \lambda > 1 \end{cases} \Rightarrow \begin{matrix} f^*(\lambda x) \neq \lambda^* f(x) \\ \text{not generally holds for} \\ \text{all } x \in \mathbb{R} \text{ \& } \lambda \in \mathbb{R}^+ \end{matrix}$$

$\boxed{\text{Reason 2}}$

$f^*(0) = e^0 = 1$ while $f(0) = \sum_{i=1}^{n} v_i \sigma(0) = 0 \Rightarrow |f(0) - f^*(0)| \geq 1$

$\Rightarrow$ not possible to approximate $f^*(x)$ through neural network f

To solve this issue, possible methods are:

① add the bias term, i.e., $f(x) = \sum_{i=1}^{n} v_i \sigma(w_i^T x + b_i)$ $b_i \in \mathbb{R}$

② change hidden layer activation function to sigmoid function,

i.e., $\sigma(z) = \dfrac{1}{1+\exp(-z)}$ instead of ReLu

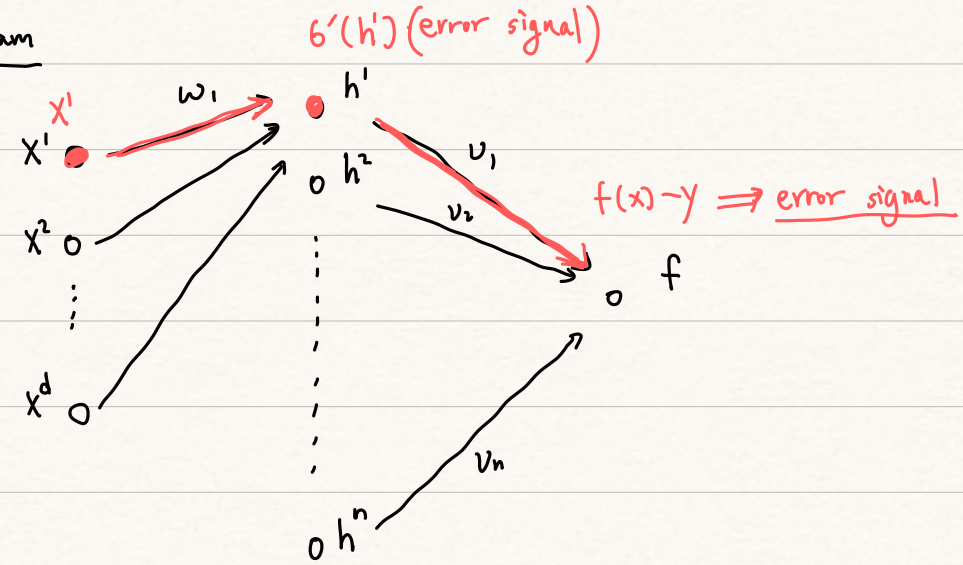
(c) $R_{emp}(\theta) = \dfrac{1}{2N} \sum_{j=1}^{N} (f(x_j) - y_j)^2$ denote $h_j = w_i^T x_j$

$$\frac{\partial R_{emp}}{\partial w_i} = \sum_{j=1}^{N} \frac{\partial R_{emp}}{\partial f(x_j)} \cdot \frac{\partial f(x_j)}{\partial \sigma(h_j)} \cdot \frac{\partial \sigma(h_j)}{\partial h_j} \cdot \frac{\partial h_j}{\partial w_i}$$

$$= \sum_{j=1}^{N} \frac{1}{2N} \cdot 2(f(x_j) - y_j) \cdot v_i \cdot \sigma'(w_i^T x_j) \cdot x_j$$

$$= \frac{v_i}{N} \sum_{j=1}^{N} (f(x_j) - y_j) \cdot \sigma'(w_i^T x_j) \cdot x_j$$

(d) Diagram



From (c), we know that:

$$\frac{\partial R_{emp}}{\partial w_i} = \frac{v_i}{N} \sum_{j=1}^{N} (f(x_j) - y_j)\, \sigma'(w_i^T x_j) \cdot x_j$$

Recall: $\sigma(z) = \begin{cases} 0, & z < 0 \\ z, & z \geq 0 \end{cases}$

Therefore, ① for $z > 0$, $\sigma'(z) = 1$

② for $z < 0$, $\sigma'(z) = 0$

③ for $z = 0$, subgradient of σ $(\partial \sigma(0))$ is

$$\partial \sigma(0) = [0, 1]$$

Conclude from Property ① ② ③, denote that

$J_i = \{ j \in [N] : w_i^T x_j > 0 \}$ $\Rightarrow$ activation set for i-th neuron

we have $$\frac{\partial R_{emp}}{\partial w_i} = \sum_{j \in J_i} \underbrace{\frac{v_i}{N} (f(x_j) - y_j)}_{\text{weight}} \cdot x_j$$

This can be interpreted as: average (weighted) of the datapoints x_j for which the i-th neuron is activated

(e) $\boxed{\text{Gradient Descent Framework}}$

$$w_i^{(k+1)} = w_i^{(k)} - \alpha \cdot \nabla_{w_i} R_{emp}(f) \Big|_{w_i = w_i^{(k)}}$$

If for i-th neuron, for all $j \in [N]$, it holds $w_i^{(k)T} x_j \leq 0$

then $J_i = \{ j \in [N] : w_i^{(k)T} x_j > 0 \} = \phi$.

Thus, $$\frac{\partial R_{emp}}{\partial w_i} \Big|_{w_i = w_i^{(k)}} = \sum_{j \in J_i} \frac{v_i}{N} (f(x_j) - y_j) \cdot x_j = 0$$

Thus, $w_i^{(k+1)} = w_i^{(k)}$ after gradient descent algo for one step

(f) (i) $x_j \in \mathbb{R}^d \sim N_p(0, I_p)$ $\quad j = 1, 2, \dots, N$

$\Leftrightarrow x_{j,k} \in \mathbb{R} \sim N(0, 1)$ $\quad k = 1, 2, \dots, d$

① $\mathbb{E}[w_i^T x_j] = w_i^T \mathbb{E}[x_j] = w_i^T \cdot \underline{0} = 0$

② $$\begin{aligned} Var[w_i^T x_j] &= w_i^T Var[x_j] \cdot w_i \\ &= w_i^T \mathbb{E}[(x_j - \mathbb{E}[x_j]) \cdot (x_j - \mathbb{E}[x_j])^T] \cdot w_i \\ &= w_i^T \mathbb{E}[x_j \cdot x_j^T] w_i \\ &= w_i^T \cdot I_p w_i \\ &= \|w_i\|_2^2 \end{aligned}$$

Also, the linear transformation of <u>Multi-variate Gaussian</u> (x_j) is still Gaussian distributed $\Rightarrow$ ③ <u>$w_i^T x_j$ satisfy Gaussian distribution</u>

① + ② + ③ $\Rightarrow$ $w_i^T x_j \sim N(0, \|w_i\|_2^2)$

(ii) from assumption, $x_j \overset{i.i.d}{\sim} N_p(0, I_p)$ $\quad j = 1, 2, \dots, N$

$\Rightarrow$ $w_i^T x_j$ is independent with each other for $j = 1, 2, \dots, N$.

$\Rightarrow$ Event $A_j = \{w_i^T x_j \leq 0\}$ are independant with each other for $j = 1, 2, \dots, N$

Therefore, $\mathbb{P}(A) = \mathbb{P}(\bigcap_{j=1}^{N} A_j)$

$$= \prod_{j=1}^{N} \mathbb{P}(A_j) \quad (\text{independence of } \{A_j\}_{j=1}^{N})$$

$$= (\mathbb{P}(w_i^T x_j \leq 0))^N \quad (\text{i.i.d of } \{x_j\}_{j=1}^{N})$$

$$= (\tfrac{1}{2})^N \quad (w_i^T x_j \sim \mathcal{N}(0, \|w_i\|_2^2))$$

Interpretation: when given enough data points and under appropriate assumption (distribution of x_j), the probability of $w_i^T x_j \leq 0$ for all $j \in [N]$ is very close to 0 (converge to 0 as $N \to +\infty$).

Therefore, we do not need to worry about i-th neuron is de-activated for all data points x_j in real life!

#