Machine Learning Assignment 6

B08611035

January 2021

1 Neural Network

(1) **b.** 36
$\delta_j^{(l)}$ for $l \in [1, 2]$ and $j \in [1, 2, \cdots, d^{(l)}]$.

$$\delta_j^{(l)} = \sum_k \left(\delta_k^{(l+1)} w_{jk}^{(l+1)}\right) \cdot \left(\tanh'(s_j^{(l)}\right)$$

For the layer (1) and (2) are

$$\delta_j^{(1)} = \sum_k wjk^{(2)} \delta_k^{(2)} \tanh'(s_j^{(j)})$$

$$\delta_j^{(2)} = \sum_k wj1^{(3)} \delta_1^{(3)} \tanh'(s_j^{(j)})$$

Thus,

$$5 \times 6 + 6 \times 1 = 30 + 6 = 36$$

(2) **d.** 1219

$$\sum_l (d^{(l)} + 1) = 50 \Rightarrow d^{(1)} + d^{(2)} + \cdots + d^{(L-1)} = 50 - (L-1) = 51 - L$$

and the total weights is $20d^{(1)} + (d^{(1)} + d)d^{(2)} + \cdots + (d^{(L-1)} + 1) \cdot 3$. Thus, when $L = 2$, $d^{(1)} = 49$:

$$\Rightarrow 20d^{(1)} + \left(d^{(1)} + 1\right) \cdot 3 = 20 \cdot 49 + 50 \cdot 3 = 1130$$

(3) **d.**

$$v = \Big[[[y = 1]] \quad \cdots \quad [[y = k]]\Big] = [v_1, v_2, \cdots, v_k]$$

$$x^{(L)} = \Big[\frac{\exp(s_1^{(L)}}{\sum \exp(e_k^{(L)}}, \frac{\exp(s_2^{(L)}}{\sum \exp(e_k^{(L)}}, \cdots, \frac{\exp(s_k^{(L)}}{\sum \exp(e_k^{(L)}}\Big] \equiv q = [q_1, q_2, \cdots, q_k]$$

Thus, the error measurement is

$$err(x,y) = -\sum_k v_k \ln(q_k)$$

and when $y = k$,

$$\delta_k^{(L)} = \frac{\partial err}{\partial s_k^{(L)}} = \frac{\partial}{\partial s_k^{(L)}}\Big(- v_k \ln(q_k)\Big)$$

$$= \frac{\partial}{\partial s_k^{(L)}}\left(- [[y=k]] \cdot \ln\Big(\frac{\exp(s_k^{(L)})}{\sum \exp_k^{(L)}}\Big)\right)$$

$$= -\frac{\sum \exp(s_k^{(L)})}{\exp(s_k^{(L)})} \cdot \frac{\exp(s_k^{(L)}) \sum \exp(s_k^{(L)}) - \exp(s_k^{(L)}) \exp(s_k^{(L)})}{(\sum \exp(s_k^{(L)}))^2} = -1 + q_k = -v_k + q_k$$

(4) **a.** 0

Initial $(w_{01}^{(1)})_0 = 0$, and weight update rule is

$$w_{01}^{(1)} \leftarrow w_{01}^{(1)} - \delta_1^{(1)}$$

Thus,

$$\Big(w_{01}^{(1)}\Big)_1 = \Big(w_{01}^{(1)}\Big)_0 - \delta_1^{(1)} = -\delta_1^{(1)}$$

$$\Big(w_{01}^{(1)}\Big)_2 = \Big(w_{01}^{(1)}\Big)_1 - \delta_1^{(1)} = -2\delta_1^{(1)}$$

$$\Big(w_{01}^{(1)}\Big)_3 = \Big(w_{01}^{(1)}\Big)_2 - \delta_1^{(1)} = -3\delta_1^{(1)}$$

And for the $\delta_1^{(1)}$,

$$\delta_1^{(1)} = \sum_k (\delta_k^{(2)})(w_{1k}^{(2)}) \tanh'(s_1^{(1)}) = \sum_k (\delta_k^{(2)})(w_{1k}^{(2)})(x_1^{(1)})'$$

$$= \sum_k (\delta_k^{(2)})(w_{1k}^{(2)}) \cdot 1 = 0$$

Thus,

$$\Big(w_{01}^{(1)}\Big)_3 = -3\delta_1^{(1)} = -3 \cdot 0 = 0$$

2 Matrix Factorization

(5) **e.**

$\tilde{d} = 1$, $w_m : \tilde{d} \times 1$ dimension, therefore w_m is a 1×1 dimension matrix (scalar). Let fix v_n, and minimize E_{in}:

$$\min_{w,v} E_{in}\Big([w_m],[v_n]\Big) \propto \sum_m \Big(\sum_{(x_n,r_{nm}) \in \mathcal{D}_m} (r_{nm} - w_m^T v_n)^2 \Big)$$

$$\frac{\partial}{\partial w_m} \left(\sum_m \Big(\sum_{(x_n,r_{nm}) \in \mathcal{D}_m} (r_{nm} - w_m^T v_n)^2 \Big) \right) = \frac{\partial}{\partial w_m} \Big(\sum_n (r_{nm} - w_m^T v_n)^2 \Big)$$

$$= -2 \sum_n (r_{nm} - w_m^T v_n) v_n = 0 \Rightarrow \sum_n (rnm - w_m^T v_n) v_n = 0$$

Because each $v_n = 2$, thus

$$\sum_n (r_{nm} - 2w_m^T) = 0 \Rightarrow \sum_n r_{nm} - 2Nw_m^T = 0$$

$$\Rightarrow w_m = \frac{1}{2} \cdot \sum_{n=1}^{N} r_{nm}$$

(6) **b.**

$$err(n, m, r_{nm}) = (r_{nm} - w_m^T v_n - a_m - b_n)^2$$

$$\nabla a_m = -2(r_{nm} - w_m^T v_n - a_m - b_n)$$

Per example gradient is proportional to $(\propto)$ -2(residual). Thus stochastic gradient descent update for a_m with learning rate equal to $1/2$ will be:

$$a_m \leftarrow a_m + \frac{1}{2} \cdot 2(r_{nm} - w_m^T v_n - a_m - b_n)$$

$$= (1 - \eta)a_m + \eta(r_{nm} - w_m^T v_n - b_n)$$

3 Aggregation

(7) **d.**

$$E_{out}(G) = \frac{1}{N}\sum_i [[\text{sign}\Big(g_1(x_i) + g_2(x_i) + g_3(x_i)\Big) \neq y_i]]$$

$$E_{out}(g_j) = \frac{1}{N}\sum_i [[g_j(x_i) \neq y_i]]$$

Thus, $E_{out}(G) = 20/100$, treat as we have total examples $N = 100$ and there are 20 examples wrong. Like this, we have:

$$E_{out}(g_1) = 0.16, E_{out}(g_2) = 0.08, E_{out}(g_3) = 0.24$$

(8) **c.** 0.32

$$E_{out}(G) = \binom{5}{3} \cdot 0.4^3 \cdot 0.6^2 + \binom{5}{4} \cdot 0.4^4 \cdot 0.6 + \binom{5}{5} \cdot 0.4^5 = 0.31744 \approx 0.32$$

(9) **b.**

Bootstrapping to sample $0.5N$ examples out of N:

$$P(\text{example is not sample}) = \frac{N-1}{N}$$

$$\Rightarrow P(0.5N \text{ sample with replacement}) = \Big(\frac{N-1}{N}\Big)^{0.5N} = \Big(1 - \frac{1}{N}\Big)^{0.5N}$$

With large N,

$$\lim_{N\to\infty} P(\cdots) = \lim_{N\to\infty} \Big(1 - \frac{1}{N}\Big)^{0.5\cdot N} = e^{-0.5} = 0.6065 \approx 60.7\%$$

(10) **e.**

The decision stump hypothesis as

$$g_{s,i,\theta}(\mathbf{x}) = s \cdot \text{sign}(x_i - \theta)$$

Thus, the kernel of decision stump could be:

$$K_{ds}(\mathbf{x}, \mathbf{x}') = (\phi_{ds}(\mathbf{x}))^T(\phi_{ds}(\mathbf{x}'))$$

$$= \sum_j \sum_i 2\text{sign}\Big(x_j - (2L + 1 + 1i)\Big) \cdot \text{sign}\Big(x_j - (2L + 1 + 1i)\Big)$$

Observe that:

$$\text{sign}(x_i - \theta) \cdot \text{sign}(x_i - \theta) = \begin{cases} -1 & \min(x_i, x_i') < \theta < \max(x_i, x_i') \\ 1 & \text{otherwise} \end{cases}$$

The number of -1 depends on how many θ_i between x_j, x_j'. And the total number of -1 between x_j and x_j' is:

$$\frac{|x_j - x_j'|}{2}$$

And the number of 1 is maximum number of 1 minus number of -1 occurs.Thus, for each j, we have

$$2 \sum_i \text{sign}\Big(x_j - (2L + 1 + 1i)\Big) \cdot \text{sign}\Big(x_j - (2L + 1 + 1i)\Big)$$

$$= 2\Big(R - L - \frac{|x_j - x_j'|}{2} - \frac{|x_j - x_j'|}{2}\Big) = 2(R - L - |x_j - x_j'|)$$

Thus,

$$\forall j, 2 \sum_i \text{sign}\Big(x_j - (2L + 1 + 1i)\Big) \cdot \text{sign}\Big(x_j - (2L + 1 + 1i)\Big) = \sum_j 2(R - L - |x_j - x_j'|)$$

$$= 2d(R - L) - 2||\mathbf{x} - \mathbf{x}'||$$

4 Adaptive Boosting

(11) **a.** 19

$$u^{(1)} = \begin{bmatrix} u_1^{(1)} & u_2^{(1)} & \cdots & u_n^{(1)} \end{bmatrix} = \begin{bmatrix} \frac{1}{N} & \frac{1}{N} & \cdots & \frac{1}{N} \end{bmatrix}$$

$g_1 = -1$, and

$$\epsilon_1 = \frac{\sum_n u_n^{(1)}[[y_n \neq g_1(\mathbf{x}_n)]]}{\sum_n u_n^{(t)}} = \sum_n u_n^{(1)}[[y_n \neq -1]] = \frac{1}{N} \cdot 0.5N = 0.5$$

Thus, we can obtain that:

$$f_t = \sqrt{\frac{1 - \epsilon_t}{\epsilon_t}} = \sqrt{\frac{95}{5}} = \sqrt{19}$$

Incorrect examples are y_i is 1 but g_1 is -1, which $[[1 = y_n \neq g_1(\mathbf{x}_n) = -1]]$.Thus,

$$u_+^{(2)} = u_+^{(1)} \cdot f_t$$

On the other hand, correct examples are that $y_i = -1$ and $g_1 = -1$ as well, which $[[-1 = y_n = g_1(\mathbf{x}_n) = -1]]$. And

$$u_-^{(2)} = u_-^{(1)} / f_t$$

$$\frac{u_+^{(2)}}{u_-^{(2)}} = \frac{u_+^{(1)} \cdot f_t}{u_-^{(1)}/f_t} = f_t^2 = (\sqrt{19})^2 = 19$$

(12) **d.**

$$\frac{U_{t+1}}{U_t} = \frac{\sum_n u_n^{(t+1)}}{\sum_n t_n^{(t)}} = \frac{\sum_n u_n^{(t+1)}[[y_n \neq g_t(\mathbf{x}_n)]] + \sum_n u_n^{(t+1)}[[y_n = g_t(\mathbf{x}_n)]]}{\sum_n u_n^{(t)}[[y_n \neq g_t(\mathbf{x}_n)]] + \sum_n u_n^{(t)}[[y_n = g_t(\mathbf{x}_n)]]}$$

$$= \sqrt{\frac{1-\epsilon_t}{\epsilon_t}} \cdot \frac{\sum_n u_n^{(t)}[[y_n \neq g_t(\mathbf{x}_n)]]}{\sum_n u_n^{(t)}[[y_n \neq g_t(\mathbf{x}_n)]] + \sum_n u_n^{(t)}[[y_n = g_t(\mathbf{x}_n)]]}$$

$$+\sqrt{\frac{\epsilon_t}{1-\epsilon_t}} \cdot \frac{\sum_n u_n^{(t)}[[y_n \neq g_t(\mathbf{x}_n)]]}{\sum_n u_n^{(t)}[[y_n = g_t(\mathbf{x}_n)]] + \sum_n u_n^{(t)}[[y_n = g_t(\mathbf{x}_n)]]}$$

$$= \sqrt{\frac{1-\epsilon_t}{\epsilon_t}}\epsilon_t + \sqrt{\frac{\epsilon_t}{1-\epsilon_t}}(1-\epsilon_t) = 2\sqrt{\epsilon_t(1-\epsilon_t)} \leq 2\sqrt{\epsilon(1-\epsilon)} \leq \exp\Big(-2(\frac{1}{2}-\epsilon)^2\Big)$$

Thus, we can infer that

$$U_{T+1} \leq \exp\Big(-2T(\frac{1}{2}-\epsilon)^2\Big) \cdot U_1 = \exp\Big(-2T(\frac{1}{2}-\epsilon)^2\Big) \cdot \sum_n u_n^{(1)}$$

and note that $\sum u_n^{(1)} = \frac{1}{N} \cdot N = 1$, thus

$$E_{in}(G_T) \leq U_{T+1} \leq \exp\Big(-2T(\frac{1}{2}-\epsilon)^2\Big)$$

5 Decision Tree

(13) **d.**

Let $a, b \in \mathbb{R}$, and $a < b$, then we have

$$a = \frac{a+b}{2} - \frac{b-a}{2}$$

and if $a, b \in \mathbb{R}$, but $b < a$, then similarly have

$$b = \frac{a+b}{2} - \frac{a-b}{2}$$

Thus, from the two results above, we have that $\forall x, y \in \mathbb{R}$

$$\frac{x+y}{2} - \frac{|x-y|}{2} = \min(x, y) \Rightarrow |x+y| - |x-y| = 2\min(x, y)$$

$$|\mu_+ + \mu_-| - |\mu_+ - \mu_-| = 2\min(\mu_+, \mu_-) \Rightarrow 1 - |\mu_+ - \mu_-| = 2\min(\mu_+, \mu_-)$$

6 Programming*

(14) **c.** 0.18

(15) **d.** 0.23

(16) **a.** 0.01

(17) **d.** 0.16

(18) **b.** 0.07