تمرین سری سوم یادگیری ماشین

امیدمحمد عذری 402212269

سوال ①

الف) میزان تخطی بیشتر بودن پیش‌بینی از y_i ← ξ_i

$$\langle w, x_i \rangle - y_i - \epsilon \leqslant \xi_i$$

$$y_i - \langle w, x_i \rangle - \epsilon \leqslant \xi_i^* \quad \text{where} \quad \xi_i, \xi_i^* \geqslant 0$$

پس می‌توانیم تابع هزینه کلی را بصورت زیر بنویسیم:

$$\min_{w \in \mathbb{R}^m,\ \xi_i, \xi_i^* \in \mathbb{R}^n} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} (\xi_i + \xi_i^*)$$

ب) تابع لاگرانژین بصورت زیر نوشته می‌شود:

$$L(w, \xi_i, \xi_i^*, \alpha, \alpha^*, \beta, \beta^*) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{m} (\xi_i + \xi_i^*)$$
$$- \sum_{i=1}^{n} (\beta_i \xi_i + \beta_i^* \xi_i^*) - \sum_{i=1}^{n} \alpha_i (\epsilon + \xi_i - y_i + \langle w, x_i \rangle)$$
$$- \sum_{i=1}^{n} \alpha_i^* (\epsilon + \xi_i^* + y_i - \langle w, x_i \rangle)$$

$\alpha_i, \alpha_i^*, \beta_i, \beta_i^* \geqslant 0$ چون باید شرط مثبت بودن را ارضا کند.

Scanned with CamScanner

$$\min_{w, \xi_i, \xi_i^*} \ \max_{\alpha, \alpha^*, \beta, \beta^*} L(w, \xi, \xi^*, \alpha, \alpha^*, \beta, \beta^*)$$

$$= \max_{\alpha, \alpha^*, \beta, \beta^*} \ \min_{w, \xi, \xi^*} L(w, \xi, \xi^*, \alpha, \alpha^*, \beta, \beta^*)$$

حال نسبت به w و ξ_i ، ξ_i^* مشتق می‌گیریم :

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) x_i = 0$$

$$\frac{\partial L}{\partial \xi_i} = C - \beta_i - \alpha_i = 0 \quad \Rightarrow \quad \beta_i = C - \alpha_i \geqslant 0$$

$$\frac{\partial L}{\partial \xi_i^*} = C - \beta_i^* - \alpha_i^* = 0 \quad \Rightarrow \quad \beta_i^* = C - \alpha_i^* \geqslant 0$$

$$L = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} (\xi_i + \xi_i^*) - \sum_{i=1}^{n} \beta_i \xi_i + \beta_i^* \xi_i^*$$

$$- \sum_{i=1}^{n} \alpha_i \left(\epsilon + \xi_i - y_i + \langle w, x_i \rangle \right) - \sum_{i=1}^{n} \alpha_i^* \left(\epsilon + \xi_i^* + y_i - \langle w, x_i \rangle \right)$$

$$= \frac{1}{2} \left\| \sum_{i=1}^{n} (\alpha_i^* - \alpha_i) x_i \right\|^2 + \sum_{i=1}^{n} \xi_i (C - \beta_i - \alpha_i) + \sum_{i=1}^{n} \xi_i^* (C - \beta_i^* - \alpha_i^*)$$

$$- \epsilon \sum_{i=1}^{n} (\alpha_i + \alpha_i^*) + \sum_{i=1}^{n} y_i (\alpha_i - \alpha_i^*) + \sum_{i=1}^{n} (\alpha_i^* - \alpha_i) \langle w, x_i \rangle$$

$$\left\langle \sum_{j=1}^{n} (\alpha_j - \alpha_j^*) x_j \ x_i \right\rangle$$

$$= -\frac{1}{2}\sum_{i,j=1}^{n} (\alpha_i-\alpha_i^*)(\alpha_i-\alpha_i^*)\langle x_i , x_j\rangle \; -$$

$$\epsilon \sum_{i=1}^{n} (\alpha_i+\alpha_i^*) + \sum_{i=1}^{n} y_i(\alpha_i-\alpha_i^*)$$

صورت دوگان :

$$\max_{\alpha,\alpha^*} \; -\frac{1}{2}\sum_{i,j=1}^{n} (\alpha_i-\alpha_i^*)(\alpha_i-\alpha_i^*)\langle x_i , x_j\rangle$$

$$-\epsilon \sum_{i=1}^{n} (\alpha_i+\alpha_i^*) + \sum_{i=1}^{n} y_i(\alpha_i-\alpha_i^*)$$

$$\alpha, \alpha^* \in [0, C]$$

پ) مسئله دارای هدف درجه دوم است و محدودیت هایش خطی است سپس می توان
با یک برنامه Quadratic Programming آن را حل کرد.

ت) در پاسخ optimal داشتیم :

$$\alpha_i\left(\epsilon + \xi_i - y_i + \langle w, x_i\rangle\right) = 0$$

$$\alpha_i^*\left(\epsilon + \xi_i^* + y_i - \langle w, x_i\rangle\right) = 0$$

$$\beta_i \xi_i = 0 \qquad \xrightarrow{\alpha_i, \alpha_i^* > 0} \qquad \epsilon + \xi_i - y_i + \langle w, x_i\rangle = 0$$

$$\beta_i^* \xi_i^* = 0$$

اگر $\xi_i = 0$ باشد نشان می دهد که x_i روی مرزهای SVM قرار دارد پس x_i یک بردار
margin support است. اگر $\xi_i > 0$ باشد نشان می دهد x_i خارج از منطقه حاشیه ϵ
است پس x_i یک بردار non-margin support است.

ادامه ت) برای α_i^* و ξ_i^* نیز این موضوع برقرار است.

ث)

$$f(x) = \langle w, x \rangle$$

$$w = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) x_i$$

$$\Longrightarrow f(x) = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) \langle x_i, x \rangle$$

می‌توان عبارت بالا را به فرم کرنل نوشت:

$$f(x) = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) K(x_i, x)$$

ج) نقش ϵ و C برعکس یکدیگر است.

هرچه ϵ کوچکتر باشد SVM سعی می‌کند روی خطاهای کمتری fit شود.

مدل تولیدی SVM پیچیده‌تر و واریانس آن زیادتر و بایاس آن کمتر می‌شود.

هرچه ϵ بزرگتر باشد، مدل تولید ساده‌تر، واریانس آن پایین‌تر اما بایاس آن بیشتر خواهد بود.

C معیاری است که چقدر برای اشتباهات جریمه در نظر می‌گیریم.

هرچه قدر C را بزرگتر در نظر بگیریم، مدل می‌خواهد جریمه بیشتری برای خطا در نظر بگیرد پس روی داده‌ها over fit می‌شود و بایاس کم اما واریانس زیاد دارد.

اما هرچه قدر C کوچک باشد، مدل ساده‌تر با بایاس زیاد و واریانس کم خواهیم داشت.

سوال ②

برای بدست آوردن حداقل تعداد نمونه m از فرمول زیر استفاده می‌کنیم:

$$m \geqslant \frac{1}{\epsilon}\left(\ln|H| + \ln\frac{1}{\delta}\right)$$

$$m \geqslant \frac{1}{0.05}\left(\ln 1000 + \ln\frac{1}{0.05}\right)$$

$$m \geqslant 198.06 \Rightarrow \boxed{m \geqslant 199}$$

چون باید صحیح باشد.

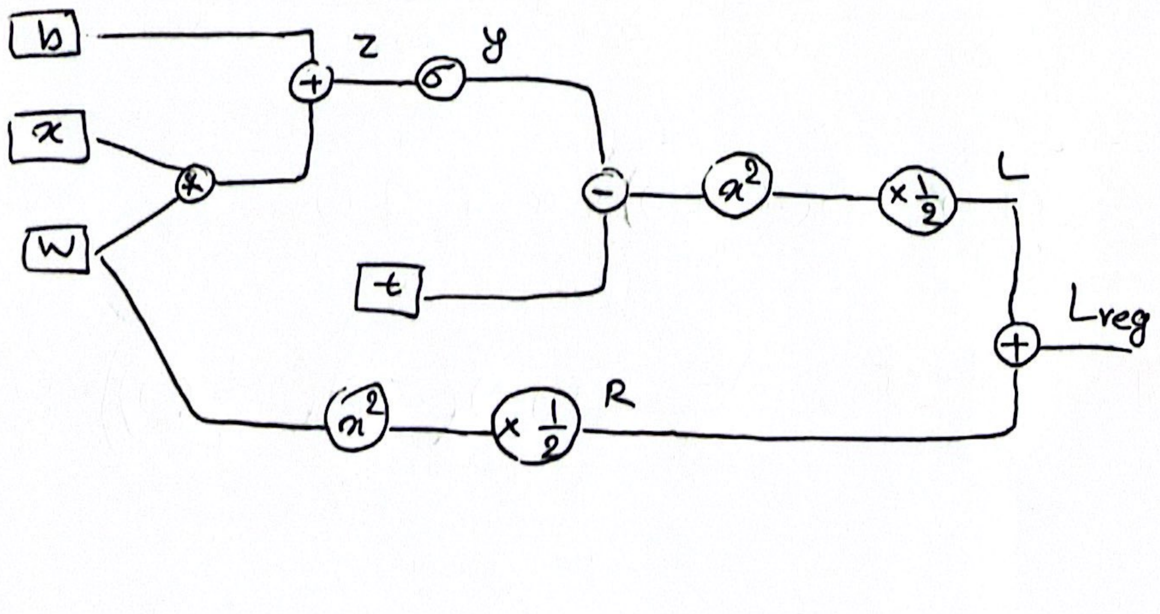
سوال ③

الف)

$$z = wx + b$$

$$y = \sigma(z)$$

$$L_{reg} = \underbrace{\frac{1}{2}(y-t)^2}_{L} + \lambda \underbrace{\frac{1}{2}w^2}_{R}$$



(ادامه الف)

$$\frac{\partial L_{reg}}{\partial L} = 1 \quad , \quad \frac{\partial L_{reg}}{\partial R} = \lambda$$

$$\frac{\partial L}{\partial y} = (y-t)$$

$$\frac{\partial L_{reg}}{\partial y} = \frac{\partial L_{reg}}{\partial L} \times \frac{\partial L}{\partial y} = (y-t)$$

$$\frac{\partial L_{reg}}{\partial z} = \frac{\partial L_{reg}}{\partial y} \times \underset{y=\sigma(z)}{\frac{\partial y}{\partial z}} = (y-t)\Big(\sigma(z)\big(1-\sigma(z)\big)\Big)$$

$$\frac{\partial L_{reg}}{\partial b} = \frac{\partial L_{reg}}{\partial z} \times \overset{1}{\frac{\partial z}{\partial b}} = (y-t)\Big(\sigma(z) - \big(1-\sigma(z)\big)\Big)$$

$$\frac{\partial L_{reg}}{\partial w} = \overset{1}{\frac{\partial L_{reg}}{\partial L}} \times \frac{\partial L}{\partial z} \times \frac{\partial z}{\partial w} + \overset{\lambda}{\frac{\partial L_{reg}}{\partial R}} \times \overset{w}{\frac{\partial R}{\partial w}}$$

$$\frac{\partial L_{reg}}{\partial w} = (y-t)\Big(\sigma(z)\big(1-\sigma(z)\big) \times x + \lambda w$$

$$\frac{\partial L_{reg}}{\partial x} = \frac{\partial L_{reg}}{\partial z} \times \overset{w}{\frac{\partial z}{\partial x}} = (y-t)\Big(\sigma(z)\big(1-\sigma(z)\big)\Big) \cdot w$$

ب) وزن‌ها بصورت رندوم در ابتدا انتخاب می‌شوند چون شروع از هر وزنی مارا می‌تواند به نقطه‌ی بهینه دیگری برساند و برخی از این نقاط بهتر از نقاط دیگر باشند. امّا اگر همواره از یک نقطه مشخص فرایند آموزش را انجام دهیم، همواره به یک نقطه بهینه local/global minima می‌رسیم و این دلخواه نخواهد بود.

وزن‌ها بصورت مقادیر کوچک نزدیک صفر انتخاب می‌شوند. چون در این نقاط گرادیان توابع activation غیر صفر است و کمک می‌کند تا با الگوریتم گرادیان کاهشی به نقاط بهینه دست یابیم. امّا اگر وزن‌ها مقادیر بزرگی باشند مثلا در تابع فعال ساز sigmaid دارای گرادیان صفر خواهیم بود و باعث جابه‌جایی درستی در الگوریتم gradient descent نخواهد شد و در یک‌جا ممکن است متوقف شویم که بهینه محلی است و نتوانیم از آن فرار کنیم.

ج) $\alpha = 0.1 \rightarrow$ learning_rate

initial value : $\begin{cases} x = 1 \\ b = 1 \\ w = 2 \\ t = 1 \\ \lambda = 0.01 \end{cases}$

<u>forward propagation</u>

$$z = wx + b = 2 + 1 = 3$$

$$y = \frac{1}{1+e^{-3}} = \frac{1}{1.0497} = 0.952$$

$$L_{reg} = \frac{1}{2}(1-0.952)^2 + 0.01 \times \frac{1}{2}\, 2^2$$

backward propagation

$$w' = w - \alpha \frac{\partial L_{reg}}{\partial w} = 2 - 0.1 \times \left[(0.952-1)\left(0.952\,(1-0.952)\right) \cdot 1 + 0.01 \times 2 \right]$$

$$\Rightarrow \quad w' = 2 - 0.1 \times 0.0178 \simeq 1.99822$$

$$b' = b - \alpha \frac{\partial L_{reg}}{\partial b} = 1 - 0.1 \left[(0.952-1)\left(0.952\,(1-0.952)\right) \right]$$

$$\Rightarrow \quad b' = 1 - 0.1 \times (-0.0021) \simeq 1.00021$$