هوش مصنوعی
بهار ۱۴۰۲
استاد: محمدحسین رهبان

گردآورندگان: پارسا حسینی، یاسمن زلفی، امیرحسین محمد رضایی، علی نظری

دانشگاه صنعتی شریف
دانشکده‌ی مهندسی کامپیوتر

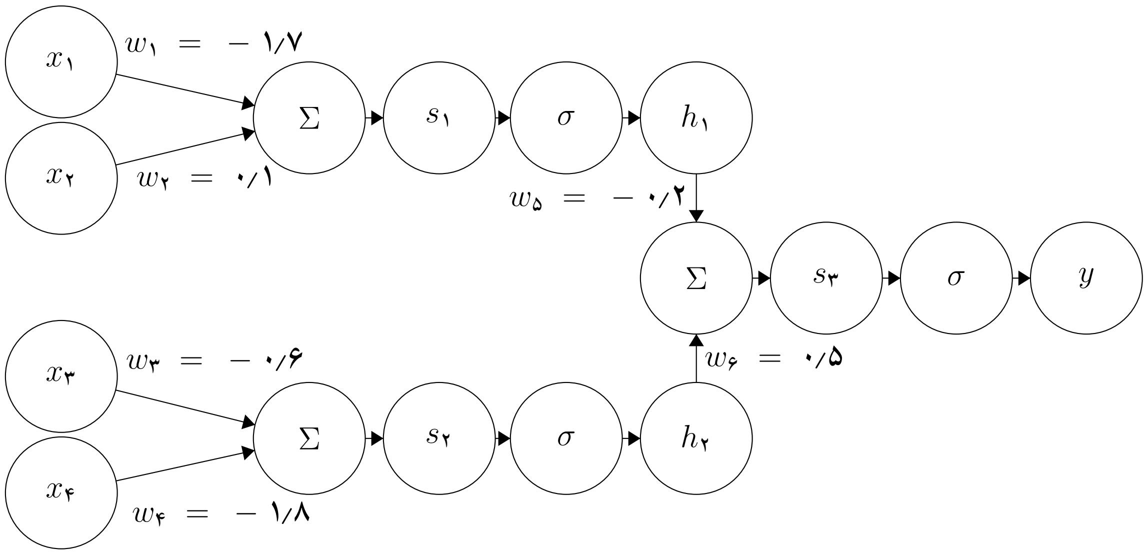
پاسخ تمرین ششم | **شبکه‌های عصبی** | مهلت ارسال: -

سوالات (۱۴۰ نمره)

۱. (۲۰ نمره) شبکه‌ی زیر را در نظر بگیرید که متشکل از برخی متغیرها و برخی تابع‌ها است. منظور از σ همان logistic function است که با کمک تابع زیر محاسبه می‌شود:

$$\sigma(x) = \frac{۱}{۱ + e^{-x}}$$

به عنوان نمونه، مقدار میانی $h_۱$ در شبکه زیر برابر با $h_۱ = \frac{۱}{۱+e^{-x_۱w_۱-x_۲w_۲}}$ است.



برای loss function هم L2 loss را در نظر بگیرید که به شکل $L(y, \hat{y}) = ||y - \hat{y}||^۲$ محاسبه می‌شود. فرض کنیم ورودی $(x_۱, x_۲, x_۳, x_۴) = (-۰/۷, ۱/۲, ۱/۱, -۲)$ است و مقدار واقعی خروجی هم 0.5 باید باشد. با استفاده از backpropagation مقدار $\frac{\delta L}{w_۱}$ را محاسبه کنید.

حل. نخست باید forward path را انجام دهیم.

$$s_۱ = ۱/۳۱$$

$$s_۲ = ۲/۹۴$$

$$h_۱ = ۰/۷۸۷۵$$

$$h_۲ = ۰/۹۴۹۸$$

$$s_۳ = ۰/۳۱۷۳$$

$$\hat{y} = ۰/۵۷۸۷$$

حال مرحله backpropagation را می‌رویم:

$$\frac{\delta E}{\delta w_1} = \frac{\delta E}{\delta \hat{y}} \times \frac{\delta \hat{y}}{\delta s_3} \times \frac{\delta s_3}{\delta h_1} \times \frac{\delta h_1}{\delta s_1} \times \frac{\delta s_1}{\delta w_1}$$

$$= 2||\hat{y} - y|| \times \acute{\sigma}(s_3) \times w_5 \times \acute{\sigma}(s_1) \times x_1$$

$$= 2(0.5787 - 0.5) \times 0.2438 \times -0.2 \times 0.1673 \times -0.7$$

$$= 0.00008988$$

به همین شکل برای بقیه وزن‌ها هم می‌توان اینگونه محاسبات را انجام داد و با این عدد، وزن‌ها را آپدیت کرد.

۲. (۲۰ نمره) فرض کنید که یک تابع convex مانند f روی بازه بسته [-b,b] داریم. $\acute{f}$ نشان دهنده مشتق تابع f است. نرخ یادگیری را هم با α نشان می‌دهیم. برای رساندن تابع به کمترین مقدار خود، یکی از روش‌ها استفاده از کاهش گرادیان است. به عنوان مثال از $x_0 = 0$ شروع می‌کنیم و از روش کاهش گرادیان استفاده می‌کنیم. اگر مقدار پس از به روز رسانی مقدار زیر b- بود، آن را برابر با b- قرار می‌دهیم و اگر مقدار بعد از به روز رسانی بالای b بود هم مقدار را برابر با b می‌گذاریم. حال به یک الگوریتم optimization مانند کاهش گرادیان، $\epsilon - converges$ گفته می‌شود که اگر در نقطه‌ای مقدار x به فاصله کمتر از ϵ به مقدار واقعی برسد.

(آ) برای $\alpha = 0.1$ و $b = 1$ و $\epsilon = 0.001$ یک تابع convex پیدا کنید که با اجرای کاهش گرادیان روی آن، $\epsilon - converges$ رخ ندهد. به طور خاص جوری جلو بروید که $x_0 = 0$ و $x_1 = b$ و $x_2 = -b$ و $x_3 = b$ و $x_4 = -b$ و ...

(ب) برای $\alpha = 0.1$ و $b = 1$ و $\epsilon = 0.001$ یک تابع convex پیدا کنید که $\epsilon - converges$ داشته باشد ولی بعد از حداقل ۱۰۰۰۰ گام اجرای کاهش گرادیان.

(ج) یک الگوریتم optimization دیگر طراحی کنید که این ویژگی را داشته باشد که همیشه $\epsilon - converges$ کند برای هر تابع convex و در $log_2(2b/\epsilon)$ قدم هم این اتفاق بیفتد.

حل.

(آ) باید تابعی را در نظر بگیریم که در هر گام، علامت مشتق آن عوض شود و در ضمن قدر مطلق مقدار مشتق به گونه ای باشد که مقدار را به بالاتر از b یا پایین تر از $-b$ برساند.
حال می‌توان تابع $f(x) = 100(x - \frac{1}{2})^2$ را در نظر گرفت. حال مقدارهای زیر را داریم:

$$\acute{f}(x) = 200(x - \frac{1}{2})$$

$$\acute{f}(0) = -100$$
$$\acute{f}(1) = 100$$
$$\acute{f}(-1) = -300$$

در گام اول که مقدار x برابر با صفر است. در گام بعد:

$$x_1 = 0 - 0.1(-100) = 10 => x_1 = 1$$

$$x_2 = 1 - 0.1(100) = -9 => x_2 = -1$$

$$x_3 = -1 - 0.1(-300) = 29 => x_3 = 1$$

(ب) می‌توان تابع $f(x) = 0/0001x$ را در نظر گرفت. در بازه $[-1, 1]$ که همان بازه x_t ها است، مقدار کمینه تابع به ازای $x_t = -1$ به دست می‌آید. پس در هر گام، مقدار x_t به اندازه $0/0001 \times 0/1$ کوچک تر از مقدار قبلی خواهد شد. از آنجا که x_0 از صفر می‌شود، پس ظبعد از ۱۰۰۰۰ قدم:

$$x_{10000} = 0 - 100000 \times 0/00001 = -1 = x_{min}$$

خواهد بود.

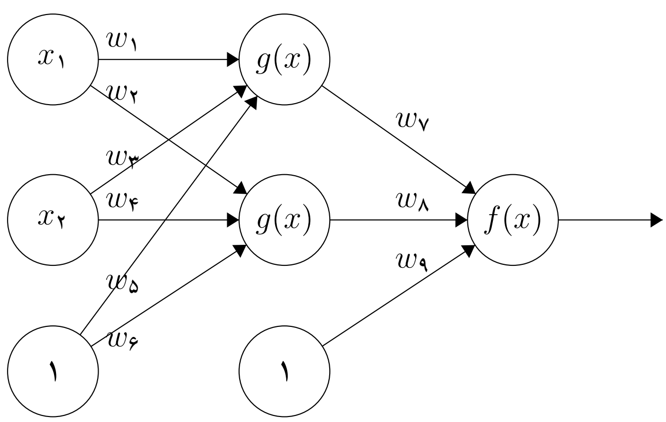
(ج) می‌توان یک binary search را در نظر گرفت که ما از $-b$ شروع می‌کنیم و در b تمام می‌کنیم. پس pseudocode زیر را می‌توان در نظر گرفت:

start = -b, end = b
while start < end:
$x_t = \frac{start+end}{2}$
if $f'(x_t) < 0$: $start = x_t$
else : end = x_t
t += 1
return x_t

پس در هر گام تابع را دو بخش می‌کنیم و به دنبال min می‌گردیم. برای اینکار $f'(x_t)$ را محاسبه می‌کنیم که به ما می‌گوید در کدام جهت مقدار تابع کم می‌شود. اگر مقدار مشتق کوچک تر از صفر باشد، یعنی مقدار min در نیمه راست است. در غیر این صورت به سمت چپ نگاه می‌کنیم. بازه جست و جو اولیه هم $2b$ است. بعد از اینکه بازه سرچ را در t قدم به دو بخش تقسیم می‌کنیم، مقدار min در یک بازه $\frac{2b}{2^t} = \epsilon$ قرار می‌گیرد. پس اگر مقدار t را بخواهیم به دست بیاوریم:

$$t = log_2(\frac{2b}{\epsilon})$$

۳. (۲۰ نمره) شبکه عصبی زیر برای دسته‌بندی binary در نظر گرفته شده است که یک لایه نهان دارد. در واحدهای نهان از تابع فعال‌سازی خطی $g(x) = cx$ و در واحد خروجی از تابع فعال‌سازی sigmoid که به صورت $f(x) = \frac{1}{1+e^{-x}}$ است، استفاده شده است تا تابع $P(y = 1|x, w)$ یاد گرفته شود. ($x = (x_1, x_2)$ ورودی و $w = (w_1, w_2, w_3, ..., w_9)$ وزن‌های شبکه عصبی هستند.)



(آ) خروجی شبکه بالا را بر حسب c، ورودی‌های x_i و وزن‌های w_i بیان کنید. مرز (boundary) مربوط به classifier نهایی چیست؟

(ب) شبکه عصبی معادل شبکه بالا که فاقد لایه نهان است را طراحی کنید و وزن‌های شبکه جدید را بر حسب c و w_i مشخص کنید.

(ج) آیا هر multi layered neural net با توابع فعال‌سازی خطی را می‌توان به صورت یک neural net بدون لایه نهان نشان داد؟ پاسخ خود را توضیح دهید.

حل.

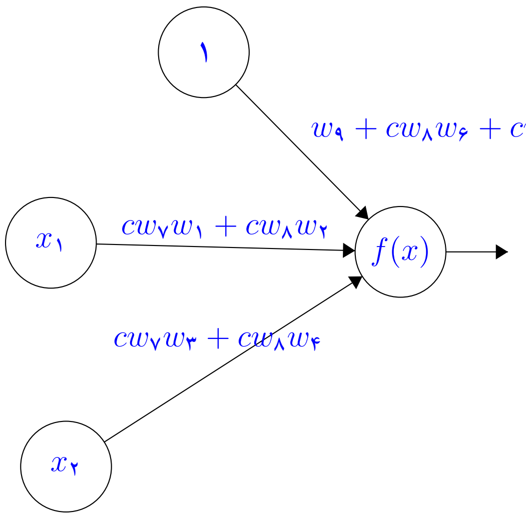
(آ)

$$
\begin{aligned}
& f(w_7 g(w_1 x_1 + w_3 x_2 + w_5) + w_8 g(w_2 x_1 + w_4 x_2 + w_6) + w_9) \\
&= \frac{1}{1 + exp(-(w_7 g(w_1 x_1 + w_3 x_2 + w_5) + w_8 g(w_2 x_1 + w_4 x_2 + w_6) + w_9))} \\
&= \frac{1}{1 + exp(-(w_9 + cw_8 w_6 + cw_7 w_5 + (cw_7 w_1 + cw_8 w_2) x_1 + (cw_7 w_3 + cw_8 w_4) x_2))}
\end{aligned}
$$

مرز مربوط به classifier به صورت زیر می‌باشد:

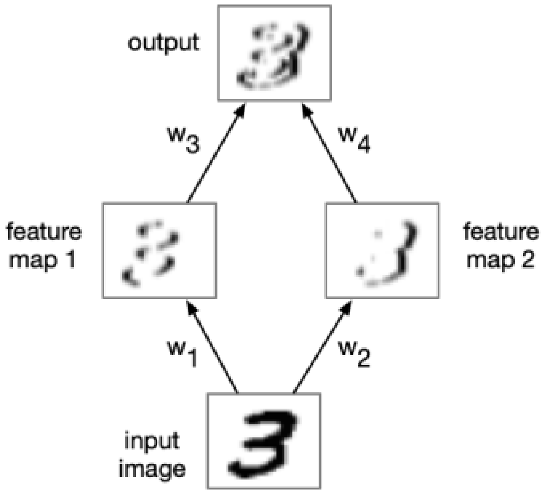
$$
w_9 + cw_8 w_6 + cw_7 w_5 + (cw_7 w_1 + cw_8 w_2) x_1 + (cw_7 w_3 + cw_8 w_4) x_2 = 0
$$

(ب)



(ج) بله. اگر تابع فعال‌سازی همه‌ی واحدهای نهان خطی باشد، خروجی واحدهای نهان به صورت ترکیب خطی از feature های ورودی خواهد بود. از آنجا که خروجی‌های میانی، ورودی لایه خروجی نهایی هستند، همیشه می‌توان یک شبکه عصبی معادل بدون لایه‌های نهان، برای شبکه عصبی چندلایه با توابع فعال‌سازی نهان خطی پیدا کرد.

۴. (۲۰ نمره) در این سوال به دنبال طراحی شبکه convolutional هستیم که مرزهای عمودی تصویر را تشخیص دهد. معماری شبکه در زیر نشان داده شده است.



تابع فعال‌سازی اعمال‌شده به لایه convolution اول ReLU و تابع فعال‌سازی اعمال‌شده به لایه خروجی همانی است.

$$ReLU(x) = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases}$$

در این سوال از رنگ سفید برای نشان دادن صفر و از رنگ‌های تیره‌تر برای نشان دادن مقادیر بزرگ‌تر استفاده شده است.
(فرض کنید که w_1 و w_2 دو کرنل $1 \times 3 \times 3$ هستند که پس از اعمال آن‌ها، دو کانال به شکل نشان داده شده ایجاد می‌شود. سپس با اعمال یک کرنل $2 \times 3 \times 3$ به نام w که از الحاق w_3 و w_4 با ابعاد $1 \times 3 \times 3$ تشکیل شده است، خروجی ایجاد می‌شود.)

(آ) دو کرنل w_1 و w_2 با اندازه $1 \times 3 \times 3$ را طراحی کنید. یکی از آن‌ها مرز سیاه/سفید و دیگری مرز سفید/سیاه را تشخیص می‌دهد.

(ب) حال w_3 و w_4 با اندازه $1 \times 3 \times 3$ را به گونه‌ای معرفی کنید که خروجی مورد نظر را تولید کند.

حل.

(آ) همان‌طور که که در feature map 1 می‌بینیم، w_1 مرزهای عمودی که سمت چپ آن سفید و سمت راست آن سیاه است را تشخیص می‌دهد و w_2 مرزهای عمودی که سمت چپ آن سیاه و سمت راست آن سفید است را تشخیص می‌دهد. بنابراین w_1 و w_2 بایستی به شکل زیر باشند:

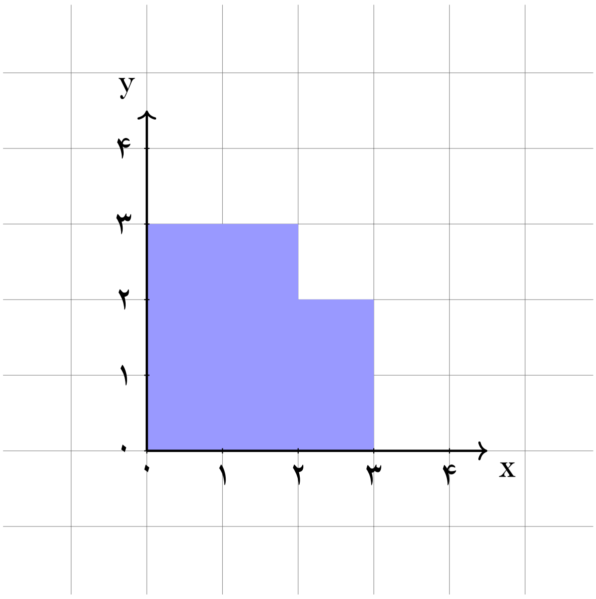
$$w_1 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

$$w_2 = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

(ب) بنابراین می‌توان w_3 و w_4 را به صورت زیر در نظر گرفت که خروجی نهایی حاصل جمع feature map 1 و feature map 2 است.

$$w_3 = w_4 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

۵. (۲۰ نمره) شبکه عصبی طراحی کنید که با ورودی گرفتن دو عدد حقیقی x و y مشخص کند که آیا نقطه با مختصات داده شده در ناحیه رنگی قرار دارد یا ندارد. وزن لایه‌های شبکه و توابع فعال‌سازی را که در شبکه عصبی استفاده کرده‌اید را مشخص کنید.



حل. میتوانیم یک بار بررسی کنیم که آیا نقطه داده شده در مستطیل افقی قرار دارد یا نه و یکبار دیگر هم اینکار برای مستطیل عمودی انجام دهیم و نتیجه این دو را or بگیریم.

شرط قرار گیری نقطه داده شده در مستطیل افقی:

$$rect1 = (x \geq 0) \wedge and(x \leq 3) \wedge (y \geq 0) \wedge (y \leq 2)$$

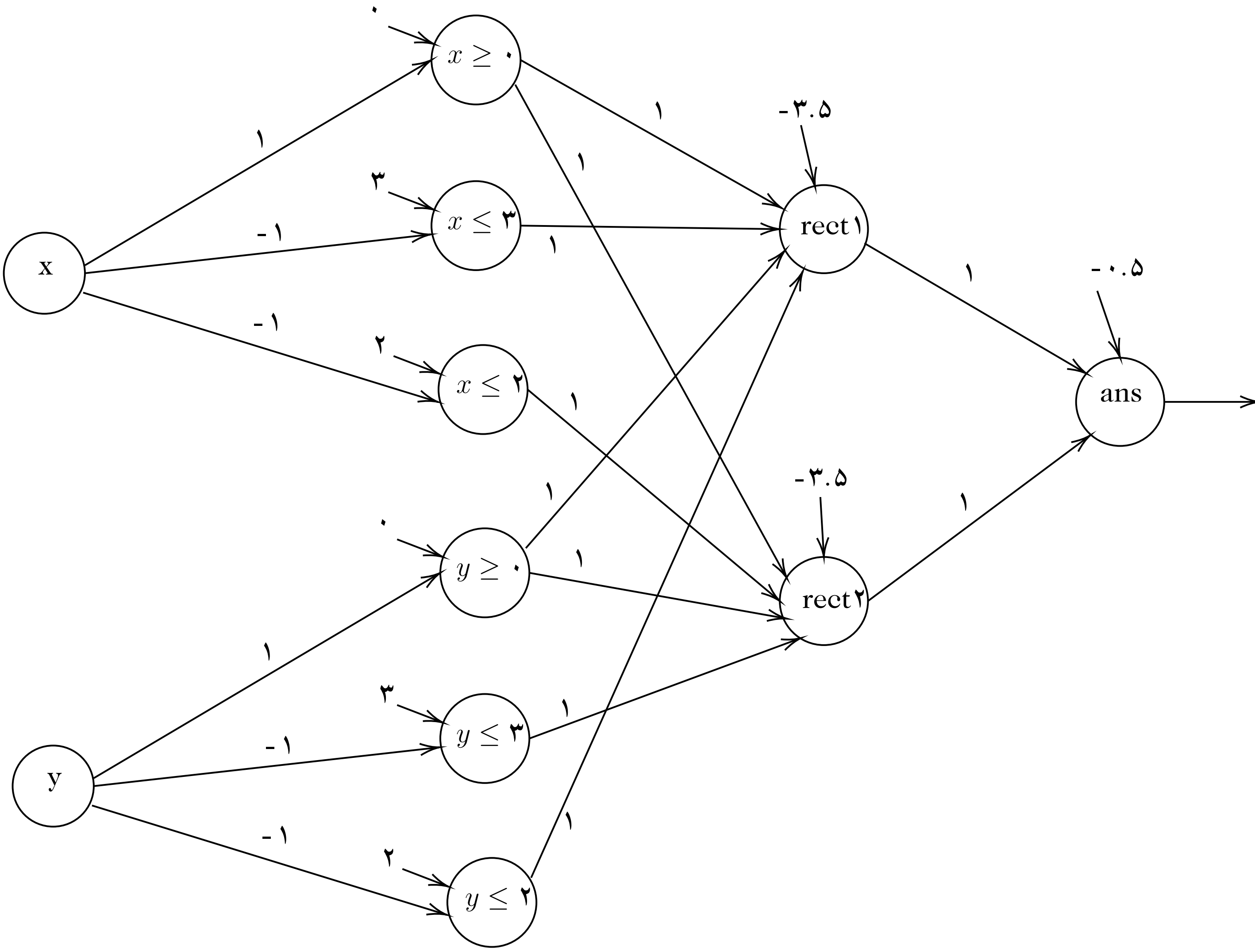
شرط قرارگیری نقطه داده شده در مستطیل عمودی:

$$rect2 = (x \geq 0) \wedge and(x \leq 2) \wedge (y \geq 0) \wedge (y \leq 3)$$

برای تابع فعال‌سازی در اینجا از تابع heaviside استفاده می‌کنیم:

$$heaviside(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

با استفاده از نورون‌های لایه دوم، شرط‌های بر مختصات‌ها بررسی می‌شود و در لایه بعدی هرکدام از شرط‌ها برای تعیین اینکه نقطه در مستطیل افقی یا عمودی قرار دارد بررسی می‌شود، در نهایت در نورون لایه آخر این دو شرط or می‌شوند و جواب خروجی داده می‌شود.



۶. (۲۰ نمره) به سوالات زیر پاسخ دهید.

(آ) مشکل vanishing gradient و exploding gradient در شبکه‌های عصبی را شرح دهید و برای مقابله با هرکدام راهکاری پیشنهاد بدهید.

(ب) مزایا و معایب اضافه کردن لایه‌های بیشتر به یک شبکه عصبی چه می‌تواند باشد؟

(ج) کدام یک از توابع زیر را میتوان به‌عنوان تابع فعال‌ساز در یک شبکه عصبی استفاده کرد؟

$$f(x) = -min(۲, x)$$

$$f(x) = \begin{cases} max(x, ۰/۱x), & x \geq ۰ \\ min(x, ۰/۱x), & x < ۰ \end{cases}$$

$$f(x) = \begin{cases} min(x, ۰/۱x), & x \geq ۰ \\ min(x, ۰/۱x), & x < ۰ \end{cases}$$

حل.

(آ) مشکل vanishing gradient هنگامی رخ می‌دهد که اندازه گرادیان در حین انجام الگوریتم back-propagation هرچه به سمت لایه‌های عقب‌تر پیش می‌رود کمتر می‌شود و به صفر میل می‌کند. همین مورد باعث خواهد شد که تغییر وزن لایه‌های قبلی در هنگام train کردن یک شبکه عصبی سخت‌تر بشود و مقدار وزن‌ها به سختی تغییر کنند.
راه‌حل‌ها:

i. استفاده از توابع فعال‌سازی که مشتق آنها هیچوقت صفر نشود (مانند Leaky ReLU)

ii. کاهش تعداد لایه‌های شبکه عصبی

iii. استفاده از روش‌های مناسب برای وزن‌دهی شبکه عصبی

مشکل exploding gradient برخلاف مورد قبلی است که در اینجا اندازه گرادیان‌های وزن‌های شبکه عصبی در حین انجام الگوریتم backpropagation هرچه به سمت لایه‌های عقب‌تر می‌روند، بزرگ و بزرگتر می‌شوند و این مورد میتواند باعث ایجاد تغییرات بسیار زیاد برای وزن‌های شبکه و در نتیجه همگرا نشدن وزن‌های شبکه به مدل مطلوب بشود.
راه‌حل‌ها:

i. استفاده از gradient clipping :ایده کلی این روش این است که اندازه گرادیان‌های ایجاد شده در الگوریتم backpropagation را محدود کنیم و اجازه ندهیم که از مقدار معینی بزرگتر بشوند.

ii. در اینجا نیز استفاده از روش‌های مناسب وزن‌دهی اولیه شبکه نیز می‌تواند مفید باشد.

(ب) مزایا:

i. جلوگیری از underfitting شبکه عصبی

ii. با افزایش تعداد لایه‌ها، شبکه می‌تواند ویژگی‌ها (feature) بیشتری از داده‌ها یاد بگیرد و در نتیجه کارکرد و دقت بیشتری داشته باشد.

معایب:

i. شبکه عصبی به احتمال بیشتری دچار overfitting خواهد شد.

ii. افزایش تعداد لایه‌ها باعث طولانی‌تر شدن روند training خواهد شد.

(ج) تابع اول و سوم مناسب هستند چراکه این دو تابع غیرخطی هستند اما تابع دوم خطی است. ویژگی اصلی که یک تابع فعال‌ساز باید داشته باشد غیرخطی بودن است که این ویژگی باعث خواهد شد که شبکه عصبی بتواند الگوهای پیچیده‌تری یاد بگیرد و یا از هم تفکیک کند.

۷. (۲۰ نمره) به سوالات زیر پاسخ دهید.

(آ) یک شبکه عصبی دولایه برای رگرسیون با p متغیر ورودی $x_۱, \dots x_p$ ، یک لایه مخفی با اندازه U، تابع فعال‌سازی $h : \mathbb{R} \to \mathbb{R}$ و خروجی $\widehat{y} \in \mathbb{R}$ رسم کنید. تعداد پارامترهای مدل چندتاست؟

(ب) مدل قسمت قبل را می‌توان به صورت زیر نوشت:

$$q_i = h\left(b_i^{(۱)} + \sum_{j=۱}^{p} W_{ij}^{(۱)} x_j\right), \quad i = ۱, \dots, U \qquad (۱)$$

$$\widehat{y} = b^{(۲)} + \sum_{l=۱}^{U} W_l^{(۲)} q_l \qquad (۲)$$

همانطور که می‌دانید، برای پیاده‌سازی این مدل باید معادلات را به فرمت برداری/ماتریسی داشته باشیم. معادلات بالا را به کمک متغیرهای زیر برداری کنید و در نهایت به فرمت $\widehat{y} = f(\mathbf{x})$ بنویسید. دقت کنید که جواب نهایی نباید هیچ سیگما یا حلقه‌ای داشته باشد.

$$\mathbf{x} = \begin{bmatrix} x_۱ \\ \vdots \\ x_p \end{bmatrix}, \mathbf{q} = \begin{bmatrix} q_۱ \\ \vdots \\ q_U \end{bmatrix}, \mathbf{b}^{(۱)} = \begin{bmatrix} b_۱^{(۱)} \\ \vdots \\ b_U^{(۱)} \end{bmatrix}, \mathbf{W}^{(۱)} = \begin{bmatrix} W_{۱۱}^{(۱)} & \dots & W_{۱p}^{(۱)} \\ \vdots & & \vdots \\ W_{U۱}^{(۱)} & \dots & W_{Up}^{(۱)} \end{bmatrix}, \tag{۳}$$

$$\mathbf{b}^{(۲)} = \left[b^{(۲)}\right], \mathbf{W}^{(۲)} = \left[W_۱^{(۲)}, \dots, W_U^{(۲)}\right]$$

(ج) حالا فرض کنید $\{\mathrm{x}_i\}_{i=۱}^n$ داده‌های ما باشند. برای هر کدام از این داده‌ها رابطه قسمت قبل برقرار است، یعنی:

$$\widehat{\mathbf{y}}_i = f(\mathbf{x}_\mathbf{i}), \quad i = ۱, \dots, n \tag{۴}$$

این n معادله را هم به فرمت ماتریسی بنویسید، یعنی به فرمت $\widehat{\mathbf{Y}} = f(\mathbf{X})$. از متغیرهای زیر کمک بگیرید.

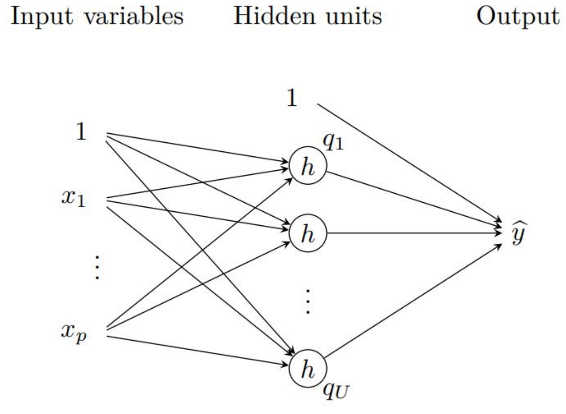
$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_۱^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix}, \widehat{\mathbf{Y}} = \begin{bmatrix} \widehat{\mathbf{y}}_۱^T \\ \vdots \\ \widehat{\mathbf{y}}_n^T \end{bmatrix}, \mathbf{Q} = \begin{bmatrix} \mathbf{q}_۱^T \\ \vdots \\ \mathbf{q}_n^T \end{bmatrix} \tag{۵}$$

(د) فرض کنید تابع فعالسازی همانی باشد یعنی $h(x) = x$. در این صورت نشان دهید مدل بالا معادل با یک مدل رگرسیون خطی می‌شود. به طور خاص، باید نشان دهید تمام پارامترهای شبکه عصبی به فرمت یک مدل رگرسیون خطی می‌شود:

$$\widehat{y} = \theta_. + \sum_{j=۱}^{p} \theta_j x_j \tag{۶}$$

حل.

(آ) به شکل ۱ دقت کنید. هر یال نشاندهنده ضرب ورودی و پارامترها است. با توجه به شکل تعداد پارامترهای مدل می‌شود: $(p + ۱)U + U + ۱$



شکل ۱

(ب) با ترکیب روابط داده شده به دو رابطه (۷) و (۸) می‌رسیم.

$$\begin{bmatrix} q_1 \\ \vdots \\ q_U \end{bmatrix} = \begin{bmatrix} h\left(b_1^1 + \sum_{j=1}^{p} W_{1j}^{(1)} x_j\right) \\ \vdots \\ h\left(b_U^1 + \sum_{j=1}^{p} W_{Uj}^{(1)} x_j\right) \end{bmatrix} \tag{7}$$

$$\widehat{y} = b^{(2)} + \sum_{i=1}^{U} W_i^{(2)} q_i \tag{8}$$

با توجه به این‌ها جواب نهایی می‌شود:

$$\mathbf{q} = h\left(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}\right) \tag{9}$$

$$\widehat{y} = \mathbf{W}^{(2)}\mathbf{q} + \mathbf{b}^{(2)} \tag{10}$$

دقت کنید که تابع فعالسازی به صورت element wise اعمال می‌شود.

(ج) در ابتدا از جواب قسمت قبلی transpose می‌گیریم:

$$\mathbf{q}_i^T = h\left(\mathbf{x}_i^T \mathbf{W}^{(1)T} + \mathbf{b}^{(1)T}\right) \tag{11}$$

$$\widehat{y}_i^T = \mathbf{q}_i^T \mathbf{W}^{(2)T} + \mathbf{b}^{(2)T} \tag{12}$$

بنابراین جواب نهایی می‌شود:

$$\mathbf{Q} = h\left(\mathbf{X}\mathbf{W}^{(1)T} + \mathbf{b}^{(1)T}\right) \tag{13}$$

$$\widehat{\mathbf{Y}} = \mathbf{Q}\mathbf{W}^{(2)T} + \mathbf{b}^{(2)T} \tag{14}$$

این فرمت نهایی است که در کد استفاده می‌شود. برای راحتی می‌توانید ماتریس‌ها را از اول transpose کنید که مجبور نشوید در هر لایه این کار را انجام دهید.

(د) این قسمت را می‌توان به ۲ روش حل کرد. روش اول کافی است در روابط قسمت ب $h(x) = x$ را جایگذاری و ساده کنید تا به فرمت رگرسیون بنویسید. پیشنهاد می‌شود برای تمرین بیشتر خودتان این روش را هم بنوسید. در ادامه به کمک معادلات ماتریسی که به دست آوردیم این قسمت را حل می‌کنیم. با ترکیب روابط (۹) و (۱۰) داریم:

$$\widehat{y} = \mathbf{W}^{(2)}\mathbf{W}^{(1)}\mathbf{x} + \mathbf{W}^{(2)}\mathbf{b}^{(1)} + \mathbf{b}^{(2)} \tag{15}$$

همچنین دقت کنید که مدل رگرسیون خطی را می‌توان به صورت زیر نوشت:

$$\widehat{y} = \theta_0 + \sum_{j=1}^{p} \theta_j x_j = \theta_{-0}^T \mathbf{x} + \theta_0 \tag{16}$$

که در آن $\theta_{-0}^T = [\theta_1, \ldots, \theta_p]^T$. با مقایسه روابط (۱۵) و (۱۶) می‌فهمیم که کافی است داشته باشیم:

$$\theta_{-0}^T = \mathbf{W}^{(1)T}\mathbf{W}^{(2)T}, \quad \theta_0 = \mathbf{W}^{(2)}\mathbf{b}^{(1)} + \mathbf{b}^{(2)} \tag{17}$$