یادگیری ماشین

پاییز ۱۴۰۳

استاد: علی شریفی زارچی

مسئول تمرین: علی شاه حیدر

دانشگاه صنعتی شریف

دانشکده‌ی مهندسی کامپیوتر

تمرین سوم | **فصل سوم** | مهلت ارسال نهایی: ۹ آذر

- مهلت ارسال پاسخ تا ساعت ۲۳:۵۹ روزهای مشخص‌شده است.
- در طول ترم، برای هر تمرین می‌توانید تا ۵ روز تأخیر مجاز داشته باشید و در مجموع حداکثر ۱۵ روز تأخیر مجاز خواهید داشت. توجه داشته باشید که تأخیر در تمرین‌های عملی و تئوری به صورت جداگانه محاسبه می‌شود و مجموع تأخیر هر دو نباید بیشتر از ۱۵ روز شود. پس از اتمام زمان مجاز، دو روز اضافی برای آپلود غیرمجاز در نظر گرفته شده است که در این بازه به ازای هر ساعت تأخیر، ۲ درصد از نمره تمرین کسر خواهد شد.
- اگر بخش عملی یا تئوری تمرین را قبل از مهلت ارسال امتیازی آپلود کنید، ۲۰ درصد نمره اضافی به آن بخش تعلق خواهد گرفت و پس از آن، ویدئویی تحت عنوان راهنمایی برای حل تمرین منتشر خواهد شد.
- حتماً تمرین‌ها را بر اساس موارد ذکرشده در صورت سوالات حل کنید. در صورت وجود هرگونه ابهام، آن را در صفحه تمرین در سایت کوئرا مطرح کنید و به پاسخ‌هایی که از سوی دستیار آموزشی مربوطه ارائه می‌شود، توجه کنید.
- در صورت هم‌فکری و یا استفاده از هر منابع خارج درسی، نام هم‌فکران و آدرس منابع مورد استفاده برای حل سوال مورد نظر را ذکر کنید.
- فایل پاسخ‌های سوالات نظری را در قالب یک فایل pdf به فرمت $HW۳\_T\_[STD\_ID].pdf$ آماده کنید و برای سوالات عملی، هریک را در یک فایل zip جداگانه قرار دهید و فایل zip اول را به فرمت $HW۳\_P۱\_[STD\_ID].zip$ و فایل zip دوم را به فرمت $HW۳\_P۲\_[STD\_ID].zip$ و فایل سوم را نیز به فرمت $HW۳\_P۳\_[STD\_ID].zip$ نامگذاری کرده و هرکدام را به صورت جداگانه آپلود کنید.
- گردآورندگان تمرین: کیارش جولایی، علیرضا صابونچی، امیرعلی لقمانی، آسمانه نافع، دنیا نوابی، فاطمه شیری

سوالات نظری (۱۰۰ نمره)

حل.

سوال ۱

الف)

پاسخ برابر ۱۴ است. برای ورودی‌ها ۴ حالت {(۰،۰)،(۱،۰)،(۰،۱)،(۱،۱)} داریم. برای هر یک از این ۴ حالت، یک perceptron داریم که فقط یکی از این ۴ تا را با برچسب ۱ خروجی می‌دهد و بقیه را برچسب ۰ می‌دهد. مشابهاً هم برای هر کدام فقط یک perceptron هست که آن یکی را برچسب ۰ می‌دهد و بقیه را برچسب ۱ می‌دهد. لذا تا اینجا ۸ تا داریم. سپس یک perceptron وجود دارد که ورودی دوم را نادیده می‌گیرد و دو نقطه‌ای را که ورودی اولشان برابر ۱ است برچسب ۱ می‌دهد و دو نقطه دیگر را برچسب ۰ می‌زند. همچنین یک perceptron وجود دارد که برعکسش را انجام می‌دهد. این هم دو تای دیگر اضافه می‌کند. در نهایت یک جفت کاملاً عین هم از perceptron ها وجود دارد که ورودی اول را نادیده می‌گیرد و مانند حالت قبل عمل می‌کند، فقط این بار از ورودی دوم استفاده می‌کند. همچنین یک حالت بدیهی وجود دارد که همه داده‌ها برچسب ۱ می‌گیرند یا همه برچسب ۰ می‌گیرند که نماینده مرزهای خطی هستند که همه نقاط روی یک سمت آن قرار دارند.

ب)

غلط است. با اینکه تابع tanh برعکس sigmoid میانگین صفر است، اما باز هم مشکل ناپدیدشدن گرادیان را دارد.

ج)

صحیح است. می‌دانیم که یک رابطه منطقی را می‌توان به‌صورت جمع p_iهایی نوشت که هر یک به‌صورت ضرب a_jها و نَه آن‌ها نوشته می‌شوند. نورون‌های ورودی را a_jها تعریف می‌کنیم که در صورتی که خودشان به کار رفته باشند، با وزن یک و در صورتی که not نشان به کار رفته باشد با وزن -1 مشخص می‌کنیم و اگر آن ورودی به کار نرفته باشد، وزن آن را صفر قرار می‌دهیم. بایاس هر جمله هم به اندازه یکی کمتر از تعداد نورون‌ها است که در ساخت جمله ضربی کاربرد دارند البته به‌صورت منفی. در لایه اول ضرایب یعنی p_iها را می‌سازیم. در لایه دوم هم از هر p_i با وزن یک به نورون خروجی متصل می‌کنیم و بایاس نمی‌گذاریم تا در صورتی که حداقل یکی از p_iها یک شد، مقدار خروجی مثبت شود.

د)
غلط است. افزایش عمق و عرض ظرفیت مدل را افزایش می‌دهد، اما بدون تنظیم منظم‌سازی مناسب یا داده‌های آموزشی کافی، ممکن است منجر به بیش‌برازش یا چالش‌های بهینه‌سازی شود.

ه)
غلط است. اگرچه یک شبکه سطحی و پهن می‌تواند طبق قضیه تقریب عمومی توابع را تقریب بزند، ممکن است نیاز به تعداد بسیار بیشتری نورون نسبت به یک شبکه عمیق برای نمایش کارآمد توابع پیچیده داشته باشد.

و)
مشکل ناپدیدشدن گرادیان زمانی رخ می‌دهد که گرادیان‌ها در لایه‌های ابتدایی شبکه‌های عمیق بسیار کوچک می‌شوند و یادگیری شبکه متوقف می‌شود. توابع فعال‌سازی مانند ReLU با صفر کردن مقادیر منفی ورودی، این مشکل را کاهش می‌دهند و از کاهش بیش از حد گرادیان جلوگیری می‌کنند.

ز)
غلط است. در حالی که Stochastic Gradient Descent (SGD) در مقایسه با Mini-Batch Gradient Descent به‌روزرسانی‌ها را بیشتر (پس از هر نمونه) انجام می‌دهد، این همگرایی سریع‌تر را تضمین نمی‌کند. نویز بودن به‌روزرسانی‌های فردی در SGD می‌تواند منجر به نوسانات بیشتر شود که ممکن است همگرایی را کاهش دهد یا باعث شود مدل یک راه‌حل بهینه را از دست بدهد. Mini-batch Gradient Descent، با اندازه دسته‌ای مناسب، با کاهش نویز تعادل ایجاد می‌کند و در عین حال امکان به‌روزرسانی‌های مکرر را فراهم می‌کند، که اغلب منجر به همگرایی پایدارتر و سریع‌تر در مقایسه با SGD خالص می‌شود. نرخ همگرایی همچنین به عواملی مانند نرخ یادگیری، اندازه دسته‌ای و مسئله بهینه‌سازی خاص بستگی دارد.

ی)
معادلات لازم به‌صورت زیر هستند:

$$g_i = \nabla_{x_i} y(x_{i-1}) = 1/2x_{i-1}^3 - 0/3x_{i-1}^2 - 4x_{i-1} - 0/8 \quad (1)$$
$$m_i = \mu m_{i-1} + g_i \quad (2)$$
$$x_i = x_{i-1} - \gamma m_i \quad (3)$$

مقادیر m_i از ممان و اطلاعات تکرارهای قبلی استفاده می‌کنند. برای تکرار اول، $m_0 = 0$. بنابراین، تکرار اول همان جواب مسئله بدون ممان را خواهد داشت.

مرحله ۱:

$$\begin{aligned}
g_1 &= 1/2 \times (-2/8000)^3 - 0/3 \times (-2/8000)^2 - 4 \times (-2/8000) - 0/8 = -18/2944,\\
m_1 &= 0/7 \times m_0 + g_1 = 0/7 \times 0 + (-18/2944) = -18/2944,\\
x_1 &= x_0 - 0/05 \times m_1 = -2/8000 - 0/05 \times (-18/2944) = -1/8853,\\
y(x_1) &= 0/3 \times (-1/8853)^4 - 0/1 \times (-1/8853)^3 - 2 \times (-1/8853)^2 - 0/8 \times (-1/8853)\\
&= -1/1404
\end{aligned}$$

$$
\begin{aligned}
g_2 &= 1.2 \times (-1.8853)^3 - 0.3 \times (-1.8853)^2 - 4 \times (-1.8853) - 0.8 = -2.3661,\\
m_2 &= 0.7 \times m_1 + g_2 = 0.7 \times (-18.2944) + (-2.3661) = -15.1722,\\
x_2 &= x_1 - 0.05 \times m_2 = -1.8853 - 0.05 \times (-15.1722) = -1.1267,\\
y(x_2) &= 0.3 \times (-1.1267)^4 - 0.1 \times (-1.1267)^3 - 2 \times (-1.1267)^2 - 0.8 \times (-1.1267)\\
&= -1.0110
\end{aligned}
$$

مرحله ۲:

حل.

سوال ۲

الف)

$$z_1 : f_{net} = max\left\{|x_1 + w_{z_1,1}|, |x_2 + w_{z_1,2}|, |x_3 + w_{z_1,3}|\right\} \Rightarrow \; w_{z_1,1} = -3 \; w_{z_1,2} = -5 \; w_{z_1,3} = 2$$

$$f_{act} = \exp\left(\frac{-net^2}{\theta}\right) \Rightarrow \theta = 4$$

$$f_{out} = f_{act}$$

$$z_2 : f_{net} = \sqrt{(x_1 + w_{z_2,1})^2 + (x_3 + w_{z_2,3})^2} \Rightarrow \; w_{z_2,1} = 3 \; w_{z_2,3} = -4$$

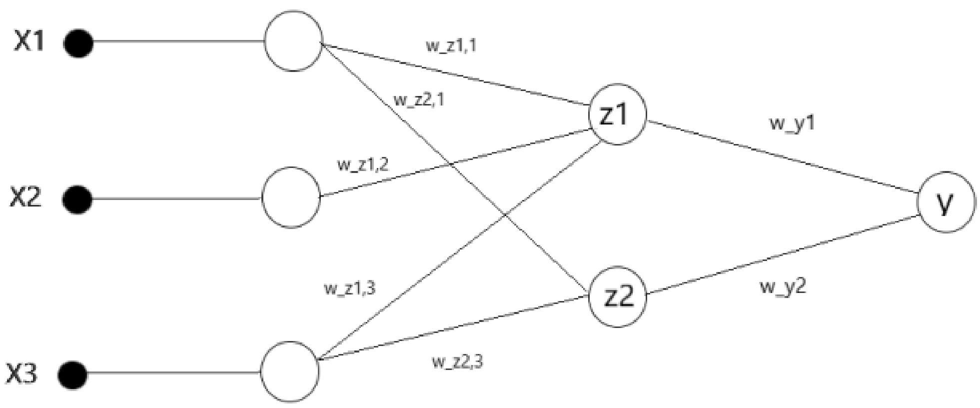
$$f_{act} = \begin{cases} 0, & \text{net} > \theta \\ 1, & \text{otherwise} \end{cases} \quad \theta = 4$$

$$f_{out} = f_{act}$$

$$y : f_{net} = w_{y1} z_1 + w_{y2} z_2 + \sigma_1 \Rightarrow w_{y1} = 6 \; w_{y2} = 2 \; \sigma_1 = 4$$

$$f_{act} = f_{net}$$

$$f_{out} = act$$



ب) برای طراحی این شبکه مسئله را به دو بخش تقسیم می‌کنیم.

بخش اول (طراحی تابع $x_1^{x_2}$) :

در این مرحله از توالی نورون‌هایی با تابع فعالسازی لگاریتمی استفاده می‌کنیم. سپس با یک نورون با تابع فعالسازی نمایی برای ایجاد خروجی مدنظر استفاده می‌کنیم.

بخش دوم (طراحی تابع ماکسیمم):

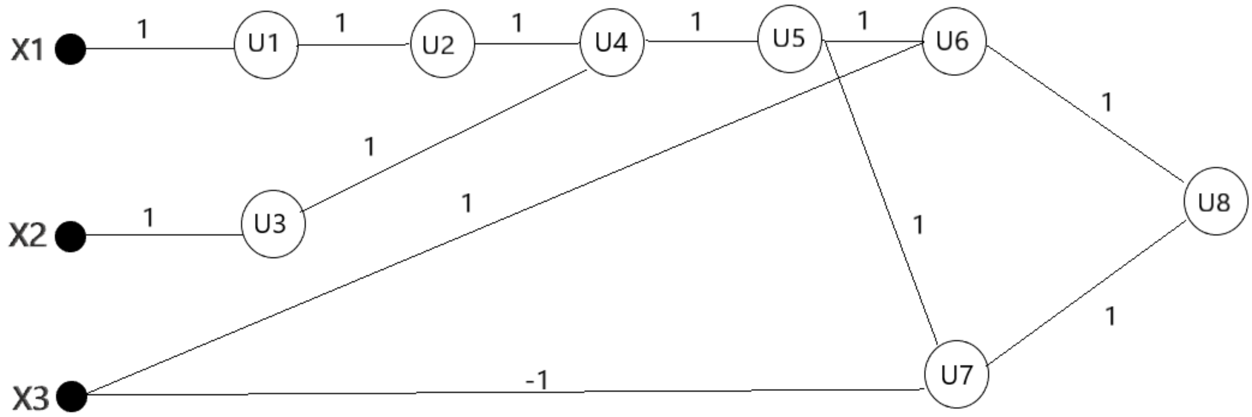
برای طراحی این بخش از عبارت زیر استفاده می‌کنیم:

$$max(x_1, x_2) = \frac{x_1+x_2+|x_1-x_2|}{2}$$

به کمک ترکیب کردن بخش‌های فوق به شبکه نهایی می‌رسیم.

$$U_{1,2,3}\begin{cases} f_{net}(x) = x \\ f_{act}(x) = ln(x) \\ f_{out}(x) = x \end{cases} U_4\begin{cases} f_{net}(x,y) = x + y \\ f_{act}(x) = e^x \\ f_{out}(x) = x \end{cases} U_5\begin{cases} f_{net}(x) = x \\ f_{act}(x) = e^x \\ f_{out}(x) = x \end{cases}$$

$$U_6\begin{cases} f_{net}(x,y) = x + y \\ f_{act}(x) = x \\ f_{out}(x) = x \end{cases} U_7\begin{cases} f_{net}(x,y) = x - y \\ f_{act}(x) = |x| \\ f_{out}(x) = x \end{cases} U_8\begin{cases} f_{net}(x,y) = x + y \\ f_{act}(x) = \frac{x}{2} \\ f_{out}(x) = x \end{cases}$$



حل.

سوال ۳

۱. $W_1 \in R^{D_{a_1}\times D_x}, b_1 \in R^{D_{a_1}\times 1}, W_2 \in R^{1\times D_{a_1}}, b_2 \in R^{1\times 1}$. شکل وزن‌ها/بایاس‌ها بعد از برداری‌سازی یکسان خواهد بود. $X \in R^{D_x\times m}, Y \in R^{m\times 1}$ بعد از برداری‌سازی.

۲. $\delta J/\delta \hat{y} = -\frac{1}{m}\sum_i \delta_1^{(i)}$. $\delta_1^{(i)} = y^{(i)}\hat{y}^{(i)} - (1-y^{(i)})/(1-\hat{y}^{(i)})$

۳. $\delta_2^{(i)} = \sigma(z_2)(1-\sigma(z_2))$

۴. $\delta_3^{(i)} = W_2$

۵. $\delta_4^{(i)} = 0$ اگر $z_1 < 0, 1$ اگر $z_1 \geq 0$

۶. $\delta_5^{(i)} = x^{(i)T}$

۷. $\delta_6 = -\frac{1}{m}\sum_i \delta_1^{(i)} * \delta_2^{(i)} * (\delta_3^{(i)} \circ \delta_4^{(i)}) * \delta_5^{(i)}$

حل.

سوال ۴

آ) Pass Forward

لایه ۱ (ورودی به مخفی ۱)

محاسبه ورودی وزن‌دار $z^{(1)}$:

$$z^{(1)} = W^{(1)}x + b^{(1)} = \begin{bmatrix} a & -a \\ b & a \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix}$$

محاسبه خروجی فعال‌سازی $a^{(1)}$ با استفاده از تابع Sigmoid:

$$a^{(1)} = \sigma(z^{(1)}) = \begin{bmatrix} \sigma(a) \\ \sigma(b) \end{bmatrix} = \begin{bmatrix} \frac{1}{1+e^{-a}} \\ \frac{1}{1+e^{-b}} \end{bmatrix}$$

لایه ۲ (مخفی ۱ به مخفی ۲)

محاسبه ورودی وزن‌دار $z^{(2)}$:

$$z^{(2)} = W^{(2)} \cdot a^{(1)} + b^{(2)} = \begin{bmatrix} a & b \\ -b & -a \end{bmatrix} \cdot \begin{bmatrix} \sigma(a) \\ \sigma(b) \end{bmatrix} = \begin{bmatrix} a \cdot \sigma(a) + b \cdot \sigma(b) \\ -b \cdot \sigma(a) - a \cdot \sigma(b) \end{bmatrix}$$

محاسبه عناصر $z^{(2)}$:

$$z_1^{(2)} = a \cdot \sigma(a) + b \cdot \sigma(b)$$
$$z_2^{(2)} = -b \cdot \sigma(a) - a \cdot \sigma(b)$$

محاسبه خروجی فعال‌سازی $a^{(2)}$ با استفاده از تابع ReLU:

$$a^{(2)} = \mathrm{ReLU}(z^{(2)}) = \begin{bmatrix} \max(0, z_1^{(2)}) \\ \max(0, z_2^{(2)}) \end{bmatrix} = \begin{bmatrix} \max(0, a \cdot \sigma(a) + b \cdot \sigma(b)) \\ \max(0, -b \cdot \sigma(a) - a \cdot \sigma(b)) \end{bmatrix}$$

لایه ۳ (مخفی ۲ به مخفی ۳)

محاسبه ورودی وزن‌دار $z^{(3)}$:

$$z^{(3)} = W^{(3)}a^{(2)} + b^{(3)} = \begin{bmatrix} a & -b \\ -a & b \end{bmatrix} a^{(2)} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

محاسبه عناصر $z^{(3)}$:

$$z_1^{(3)} = a \cdot a_1^{(2)} - b \cdot a_2^{(2)}$$
$$z_2^{(3)} = -a \cdot a_1^{(2)} + b \cdot a_2^{(2)}$$

محاسبه خروجی فعال‌سازی $a^{(3)}$ با استفاده از تابع Softmax:

$$e^{z_1^{(3)}}, \quad e^{z_2^{(3)}}$$

$$a_i^{(3)} = \frac{e^{z_i^{(3)}}}{e^{z_1^{(3)}} + e^{z_2^{(3)}}}$$

لایه ۴ (مخفی ۳ به خروجی)

محاسبه ورودی وزن‌دار $z^{(4)}$:

$$z^{(4)} = W^{(4)}a^{(3)} + b^{(4)} = \begin{bmatrix} a & -b \\ 0.5a & a \end{bmatrix} a^{(3)} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

محاسبه عناصر $z^{(4)}$:

$$z_1^{(4)} = a \cdot a_1^{(3)} - b \cdot a_2^{(3)}$$
$$z_2^{(4)} = 0.5a \cdot a_1^{(3)} + a \cdot a_2^{(3)}$$

از آنجا که تابع فعال‌سازی لایه خروجی Identity است، داریم:

$$a^{(4)} = z^{(4)}$$

ب) محاسبه خطا

با استفاده از تابع خطای میانگین مربعات (MSE):

$$E = \frac{1}{2}\left((a_1^{(4)} - t_1)^2 + (a_2^{(4)} - t_2)^2\right)$$

ج) Backward Pass

لایه ۴ (خروجی)

محاسبه $\delta^{(4)}$:

تابع خطا به صورت زیر است:

$$E = \frac{1}{2}\sum_{k=1}^{2}(a_k^{(4)} - t_k)^2$$

گرادیان خطا نسبت به خروجی شبکه:

$$\frac{\partial E}{\partial a_k^{(4)}} = a_k^{(4)} - t_k$$

$$\frac{\partial a_k^{(4)}}{\partial z_k^{(4)}} = 1$$

$$\delta_k^{(4)} = \frac{\partial E}{\partial z_k^{(4)}} = \frac{\partial E}{\partial a_k^{(4)}} \cdot \frac{\partial a_k^{(4)}}{\partial z_k^{(4)}} = (a_k^{(4)} - t_k) \times 1 = a_k^{(4)} - t_k$$

–

لایه ۳ (مخفی ۳)

محاسبه $\delta^{(3)}$:

$$z^{(4)} = W^{(4)} a^{(3)} + b^{(4)}$$

$$\frac{\partial z_i^{(4)}}{\partial a_j^{(3)}} = W_{ij}^{(4)}$$

$$\frac{\partial E}{\partial a_j^{(3)}} = \sum_{i=1}^{2} \frac{\partial E}{\partial z_i^{(4)}} \cdot \frac{\partial z_i^{(4)}}{\partial a_j^{(3)}} = \sum_{i=1}^{2} \delta_i^{(4)} W_{ij}^{(4)} = \sum_k (a_k^{(4)} - t_k) \cdot W_{ki}^{(4)}$$

$$\frac{\partial a_l^{(3)}}{\partial z_k^{(3)}} = a_l^{(3)} (\delta_{lk} - a_k^{(3)})$$

که در آن δ_{lk} دلتای کرونکر است که اگر $l = k$ باشد برابر ۱ و در غیر این صورت برابر ۰ است.

$$\frac{\partial a_i^{(3)}}{\partial z_j^{(3)}} = \begin{cases} a_i^{(3)} (1 - a_i^{(3)}) & \text{if} i = j \\ -a_i^{(3)} \cdot a_j^{(3)} & \text{if} i \neq j \end{cases}$$

بنابراین، گرادیان خطا نسبت به $z_k^{(3)}$:

$$\delta_k^{(3)} = \frac{\partial E}{\partial z_k^{(3)}} = \sum_{j=1}^{2} \frac{\partial E}{\partial a_j^{(3)}} \cdot \frac{\partial a_j^{(3)}}{\partial z_k^{(3)}}$$

$$\delta_k^{(3)} = \sum_{j=1}^{2} \left(\frac{\partial E}{\partial a_j^{(3)}} \cdot \begin{cases} a_j^{(3)} (1 - a_j^{(3)}) & \text{if} j = k \\ -a_j^{(3)} a_k^{(3)} & \text{if} j \neq k \end{cases} \right)$$

به دو بخش تقسیم می‌شود:

$$\delta_k^{(3)} \ (\text{for } j = k) = \frac{\partial E}{\partial a_k^{(3)}} \cdot a_k^{(3)} (1 - a_k^{(3)})$$

$$\delta_k^{(3)} \ (\text{for } j \neq k) = \sum_{j \neq k} \left(\frac{\partial E}{\partial a_j^{(3)}} \cdot (-a_j^{(3)} a_k^{(3)}) \right)$$

با ترکیب دو بخش:

$$\delta_k^{(3)} = \sum_{l=1}^{2} \frac{\partial E}{\partial a_l^{(3)}} \cdot \frac{\partial a_l^{(3)}}{\partial z_k^{(3)}} = \sum_{l=1}^{2} \left(\frac{\partial E}{\partial a_l^{(3)}} \cdot a_l^{(3)} (\delta_{lk} - a_k^{(3)}) \right)$$

$$\delta_k^{(3)} = \frac{\partial E}{\partial a_k^{(3)}} \cdot a_k^{(3)} (1 - a_k^{(3)}) - a_k^{(3)} \sum_{j \neq k} \left(\frac{\partial E}{\partial a_j^{(3)}} a_j^{(3)} \right)$$

(تا بالای اینجا اگر نوشته شود نمره کامل گرفته می‌شود.)

در ادامه می‌توان نوشت:

$$\sum_{j\neq k}\left(\frac{\partial E}{\partial a_j^{(۳)}}a_j^{(۳)}\right)=\left(\sum_{j=۱}^{۲}\frac{\partial E}{\partial a_j^{(۳)}}a_j^{(۳)}\right)-\frac{\partial E}{\partial a_k^{(۳)}}a_k^{(۳)}$$

$S=\sum_{j=۱}^{۲}\frac{\partial E}{\partial a_j^{(۳)}}a_j^{(۳)}$

$$\delta_k^{(۳)}=\frac{\partial E}{\partial a_k^{(۳)}}a_k^{(۳)}(۱-a_k^{(۳)})-a_k^{(۳)}\left(S-\frac{\partial E}{\partial a_k^{(۳)}}a_k^{(۳)}\right)$$

$$\delta_k^{(۳)}=\frac{\partial E}{\partial a_k^{(۳)}}a_k^{(۳)}(۱-a_k^{(۳)})-a_k^{(۳)}S+\frac{\partial E}{\partial a_k^{(۳)}}a_k^{(۳)۲}$$

$$\delta_k^{(۳)}=\frac{\partial E}{\partial a_k^{(۳)}}a_k^{(۳)}-a_k^{(۳)}S$$

$$\delta_k^{(۳)}=a_k^{(۳)}\left(\frac{\partial E}{\partial a_k^{(۳)}}-S\right)$$

$$\delta^{(۳)}=a^{(۳)}\odot\left(\frac{\partial E}{\partial a^{(۳)}}-S۱\right)$$

لایه ۲ (مخفی ۲)

محاسبه $\delta^{(۲)}$:

$$\frac{\partial E}{\partial a_j^{(۲)}}=\sum_{k=۱}^{۲}\delta_k^{(۳)}\cdot\frac{\partial z_k^{(۳)}}{\partial a_j^{(۲)}}$$

$$z^{(۳)}=W^{(۳)}a^{(۲)}+b^{(۳)}$$

$$\frac{\partial z_k^{(۳)}}{\partial a_j^{(۲)}}=W_{kj}^{(۳)}$$

$$\frac{\partial E}{\partial a_j^{(۲)}}=\sum_{k=۱}^{۲}\delta_k^{(۳)}W_{kj}^{(۳)}$$

$$\frac{\partial a_j^{(۲)}}{\partial z_j^{(۲)}}=\begin{cases}۱ & \mathrm{z}_j^{(۲)}>۰\\ ۰ & \mathrm{o.w.}\end{cases}$$

$$\delta_j^{(۲)}=\frac{\partial E}{\partial a_j^{(۲)}}\cdot\frac{\partial a_j^{(۲)}}{\partial z_j^{(۲)}}=\begin{cases}\frac{\partial E}{\partial a_j^{(۲)}} & \mathrm{z}_j^{(۲)}>۰\\ ۰ & \mathrm{o.w.}\end{cases}$$

لایه ۱ (مخفی ۱)

محاسبه $\delta^{(1)}$:

$$\frac{\partial E}{\partial a_j^{(1)}} = \sum_{k=1}^{2} \delta_k^{(2)} \cdot \frac{\partial z_k^{(2)}}{\partial a_j^{(1)}}$$

$$z^{(2)} = W^{(2)} a^{(1)} + b^{(2)}$$

$$\frac{\partial z_k^{(2)}}{\partial a_j^{(1)}} = W_{kj}^{(2)}$$

$$\frac{\partial E}{\partial a_j^{(1)}} = \sum_{k=1}^{2} \delta_k^{(2)} W_{kj}^{(2)}$$

$$\frac{\partial a_j^{(1)}}{\partial z_j^{(1)}} = a_j^{(1)}(1 - a_j^{(1)})$$

$$\delta_j^{(1)} = \frac{\partial E}{\partial a_j^{(1)}} \cdot \frac{\partial a_j^{(1)}}{\partial z_j^{(1)}} = \left(\sum_{k=1}^{2} \delta_k^{(2)} W_{kj}^{(2)} \right) \cdot a_j^{(1)}(1 - a_j^{(1)})$$

محاسبه گرادیان‌ها نسبت به وزن‌ها و بایاس‌ها

لایه ۴:

$$\frac{\partial E}{\partial W_{ij}^{(4)}} = \delta_i^{(4)} \cdot a_j^{(3)}$$

$$\frac{\partial E}{\partial b_i^{(4)}} = \delta_i^{(4)}$$

لایه ۳:

$$\frac{\partial E}{\partial W_{ij}^{(3)}} = \delta_i^{(3)} \cdot a_j^{(2)}$$

$$\frac{\partial E}{\partial b_i^{(3)}} = \delta_i^{(3)}$$

لایه ۲:

$$\frac{\partial E}{\partial W_{ij}^{(2)}} = \delta_i^{(2)} \cdot a_j^{(1)}$$

$$\frac{\partial E}{\partial b_i^{(2)}} = \delta_i^{(2)}$$

لایه ۱:

$$\frac{\partial E}{\partial W_{ij}^{(1)}} = \delta_i^{(1)} \cdot x_j$$

$$\frac{\partial E}{\partial b_i^{(1)}} = \delta_i^{(1)}$$

د) به‌روزرسانی وزن‌ها و بایاس‌ها

با استفاده از نرخ یادگیری η، وزن‌ها و بایاس‌ها را به‌روزرسانی می‌کنیم:

$$W_{ij}^{(l)} \leftarrow W_{ij}^{(l)} - \eta \frac{\partial E}{\partial W_{ij}^{(l)}}$$

$$b_i^{(l)} \leftarrow b_i^{(l)} - \eta \frac{\partial E}{\partial b_i^{(l)}}$$

حل.

سوال ۵

الف) هدف از این طراحی، ایجاد یک شبکه با استفاده از واحدهای خطی آستانه‌ای (TLUs) است که بتواند تعیین کند آیا یک نقطه در نواحی رنگی مشخص‌شده در تصویر قرار دارد یا خیر. این نواحی به شکل لوزی و متقارن نسبت به مبدا هستند. در تصویر چهار ناحیه لوزی شکل وجود دارد که مراکز آنها در مختصات $(۲,۲)$، $(-۲,۲)$، $(۲,-۲)$ و $(-۲,-۲)$ قرار دارند. این لوزی‌ها می‌توانند به صورت مجموعه‌ای از نابرابری‌های خطی توصیف شوند که مرزهای آنها را تعریف می‌کنند. برای هر لوزی، می‌توان مرزهای آن را با استفاده از معادله زیر نمایش داد:

$$|x - x_c| + |y - y_c| \leq d$$

که در آن:

(x_c, y_c) مرکز لوزی است، d فاصله مرکز تا رأس لوزی (نصف طول قطر) است.

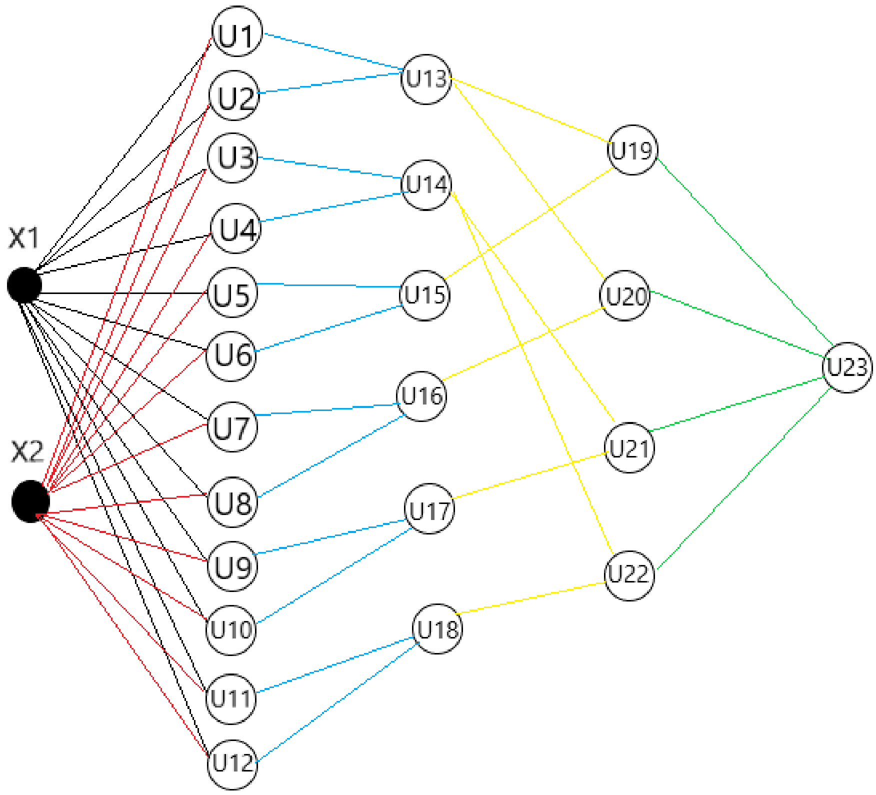
برای هر لوزی، چهار نابرابری خطی برای تعیین مرزها وجود دارد. هر واحد TLU می‌تواند یک نابرابری خطی را نشان دهد. به عنوان مثال، برای یک لوزی، نابرابری‌های زیر را می‌توان استفاده کرد:

$$x + y \leq b_۱$$
$$x - y \leq b_۲$$
$$-x + y \leq b_۳$$
$$-x - y \leq b_۴$$

در مجموع، برای هر لوزی چهار TLU نیاز است که هر کدام یک مرز را نمایش می‌دهند. اگر یک نقطه تمام این نابرابری‌ها را برآورده کند، یعنی در داخل لوزی قرار دارد. برای تعیین اینکه آیا نقطه‌ای در هر یک

از لوزی‌ها قرار دارد، خروجی‌های مربوط به هر لوزی باید با یک تابع OR ترکیب شوند. اگر خروجی یکی از لوزی‌ها ۱ باشد، نتیجه نهایی نیز ۱ خواهد بود، یعنی نقطه در یکی از نواحی رنگی قرار دارد. ساختار نهایی شبکه به این صورت است:

- **لایه ورودی:** شامل دو ورودی x و y که مختصات نقطه هستند.
- **لایه مخفی اول:** شامل ۱۲ واحد TLU (۴ واحد برای هر لوزی) که شرایط مرزی لوزی‌ها را بررسی می‌کنند. (توجه کنید تعدادی از خطوط بین نواحی مشترک هستند.)
- **لایه مخفی دوم:** شامل ۶ واحد TLU که قرار گرفتن ورودی بین جفت‌های خطوط موازی شکل را بررسی می‌کند. (۶ جفت خط موازی داریم.)
- **لایه مخفی سوم:** شامل ۴ واحد TLU که قرار گرفتن ورودی درون هر یک از نواحی را بررسی می‌کند.
- **لایه خروجی:** ترکیب نهایی نتایج لایه مخفی برای تعیین اینکه آیا نقطه در داخل یکی از لوزی‌ها قرار دارد یا خیر.



$$U_1\begin{cases}W_{U1,X1}=-1\\W_{U1,X2}=-1\\\theta_1=-1\end{cases}\quad U_2\begin{cases}W_{U2,X1}=-1\\W_{U2,X2}=-1\\\theta_2=1\end{cases}\quad U_3\begin{cases}W_{U3,X1}=-1\\W_{U3,X2}=1\\\theta_3=-1\end{cases}\quad U_4\begin{cases}W_{U4,X1}=-1\\W_{U4,X2}=1\\\theta_4=1\end{cases}$$

$$U_5\begin{cases}W_{U5,X1}=-1\\W_{U5,X2}=1\\\theta_5=3\end{cases}\quad U_6\begin{cases}W_{U6,X1}=1\\W_{U6,X2}=-1\\\theta_6=-5\end{cases}\quad U_7\begin{cases}W_{U7,X1}=1\\W_{U7,X2}=-1\\\theta_7=3\end{cases}\quad U_8\begin{cases}W_{U8,X1}=-1\\W_{U8,X2}=1\\\theta_8=-5\end{cases}$$

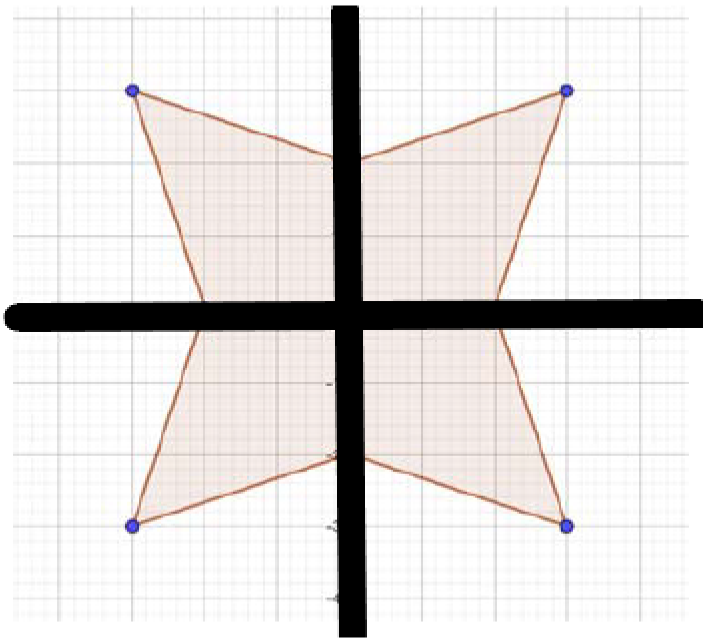
$$U_9 \begin{cases} W_{U_9,X_1} = 1 \\ W_{U_9,X_2} = 1 \\ \theta_9 = 3 \end{cases} U_{10} \begin{cases} W_{U_{10},X_1} = -1 \\ W_{U_{10},X_2} = -1 \\ \theta_2 = -5 \end{cases} U_{11} \begin{cases} W_{U_{11},X_1} = -1 \\ W_{U_{11},X_2} = -1 \\ \theta_{11} = 3 \end{cases} U_{12} \begin{cases} W_{U_{12},X_1} = 1 \\ W_{U_{12},X_2} = 1 \\ \theta_{12} = -5 \end{cases}$$

$$U_{13} \begin{cases} W_{U_{13},U_1} = 2 \\ W_{U_{13},U_2} = 2 \\ \theta_{13} = 3 \end{cases} U_{14} \begin{cases} W_{U_{14},U_3} = 2 \\ W_{U_{14},U_4} = 2 \\ \theta_{14} = 3 \end{cases} U_{15} \begin{cases} W_{U_{15},U_5} = 2 \\ W_{U_{15},U_6} = 2 \\ \theta_{15} = 3 \end{cases} U_{16} \begin{cases} W_{U_{16},U_7} = 2 \\ W_{U_{16},U_8} = 2 \\ \theta_{16} = 3 \end{cases}$$

$$U_{17} \begin{cases} W_{U_{17},U_9} = 2 \\ W_{U_{17},U_{10}} = 2 \\ \theta_{17} = 3 \end{cases} U_{18} \begin{cases} W_{U_{18},U_{11}} = 2 \\ W_{U_{18},U_{12}} = 2 \\ \theta_{18} = 3 \end{cases} U_{19} \begin{cases} W_{U_{19},U_{13}} = 2 \\ W_{U_{19},U_{15}} = 2 \\ \theta_{19} = 3 \end{cases} U_{20} \begin{cases} W_{U_{20},U_{13}} = 2 \\ W_{U_{20},U_{16}} = 2 \\ \theta_{20} = 3 \end{cases}$$

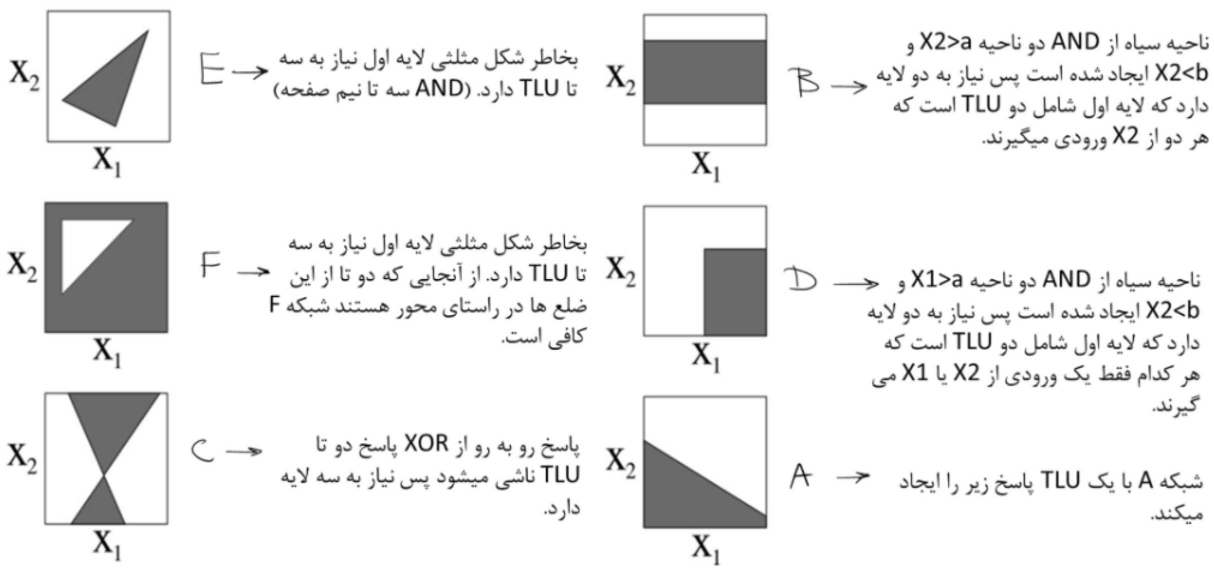
$$U_{21} \begin{cases} W_{U_{21},U_{14}} = 2 \\ W_{U_{21},U_{17}} = 2 \\ \theta_{21} = 3 \end{cases} U_{22} \begin{cases} W_{U_{22},U_{14}} = 2 \\ W_{U_{22},U_{18}} = 2 \\ \theta_{22} = 3 \end{cases} U_{23} \begin{cases} W_{U_{23},U_{19}} = 2 \\ W_{U_{23},U_{20}} = 2 \\ W_{U_{23},U_{21}} = 2 \\ W_{U_{23},U_{22}} = 2 \\ \theta_{23} = 1 \end{cases}$$

ب) شکل این بخش را می‌توان به چهارناحیه زیر تقسیم کرد.



هربخش از تقاطع چهار خط بدست می‌آید. از آنجاییکه تعدادی از خطوط مشترک هستند، در واقع ده خط متفاوت داریم. بنابراین می‌توانیم با تغییر ضرایب و حد آستانه‌ها این طبقه‌بندی را انجام دهیم.

ج)



حل.

سوال ۶

الف) با استفاده از استقرا این قضیه را اثبات می‌کنیم. نورون‌های لایه اول $v_i^{(1)}$ را در نظر بگیرید. درایه یا سطر iام را برای پارامترهای نورون یعنی $W_i^{(\cdot)}$ و $b_i^{(\cdot)}$ در نظر بگیرید. حال اگر تصویر ورودی نسبت به لایه اول شبکه را تشکیل دهیم، هر نورون یک ابرصفحه به فرم $W_i^{(\cdot)}x + b_i^{(\cdot)} = ۰$ در فضا تشکیل می‌دهد. در نهایت در فضا تعدادی ابرصفحه خواهیم داشت که ناحیه‌های مختلف را تشکیل می‌دهند. هر polytope تشکیل‌شده مخصوص یک activation pattern خواهد بود.

حال فرض کنید لایه‌های ۱ تا $d - ۱$ فضا را به polytope هاconvex تقسیم کرده‌اند. نورون $v_i^{(d)}$ و ناحیه R_i را در نظر بگیرید. در ناحیه R_i یک activation pattern ثابت داریم. اگر دقت کنید متوجه می‌شوید که ورودی‌های نورون در واقع ترکیب خطی‌ای از تعدادی از ورودی‌ها به فرم $\sum_j \lambda_j x_j + b$ می‌باشد. چرا که تابع ReLU یا خود ورودی را که ترکیب خطی‌ای از ورودی‌هاست خروجی می‌دهد و یا صفر خروجی می‌دهد. با قرار دادن این رابطه برابر ۰ به یک ابرصفحه دیگر می‌رسیم که البته تنها در ناحیه R_i صحت دارد. حال یا این صفحه در صورت برخورد با ناحیه آن را به دو ناحیه تقسیم می‌کند و یا تغییری در pattern خروجی این نورون نمی‌دهد. بنابراین همچنان فضا به convex polytopeها تقسیم می‌شود و قضیه در نتیجه آن اثبات می‌شود.

ب) مشابه بخش قبل با استقرا ثابت می‌کنیم. طبق بخش الف، لایه اول شبکه فضا را به $r(k, m)$ ناحیه تقسیم می‌کند که طبق راهنمایی تعداد این ناحیه‌ها کمتر از $\mathcal{O}(k^m)$ است. حال فرض کنید تعداد نواحی ایجادشده توسط لایه $n - ۱$ام کمتر از $\mathcal{O}(k^{m(n-۱)})$ است. حال از آنجایی که هر ناحیه دارای pattern activation ثابت است طبق بخش الف هر کدام از این ناحیه‌ها می‌توانند به حداکثر $r(k, m)$ ناحیه دیگر تقسیم شوند. یعنی تعداد نواحی ما حداکثر $\mathcal{O}(k^{m(n-۱)}) \times r(k, m)$ است که خود این مقدار کمتر از $\mathcal{O}(k^{mn})$ می‌باشد.