به نام خدا

علی قاسم زاده ۴۰۱۱۰۶۳۳۹

① الف) با یک Perceptron توابعی که خطی تفکیک پذیر اند را ساخت.

حالا x و y را ورودی و z را خروجی در نظر می‌گیریم، خواهیم داشت که:

$Z=0$ $Z=\sim x$ $Z=\sim x \wedge y$ $Z=\sim x \vee \sim y$

$Z=1$ $Z=\sim y$ $Z=\sim x \wedge \sim y$ $Z=\sim x \vee y$

$Z=x$ $Z=x \wedge y$ $Z=x \vee y$

$Z=y$ $Z=x \wedge \sim y$ $Z=x \vee \sim y$

← ازین ۱۶ تا فقط ۱۴ تا از آنها قابل ساخت‌اند، به عبارتی $x \oplus y$ و $x \oplus \sim y$ را نمی‌توانیم با perceptron تفکیک کنیم، زیرا خطی تفکیک پذیر نیست.

ب) خیر، مشکل ناپدید شدن گرادیان بخاطر این است که مشتق این تابع در مقادیر خیلی زیاد یا خیلی کم صفر می‌شود که ربطی به صفر محور بودن ندارد ← هردو مشکل ناپدید شدن گرادیان را دارند.

ج) طبق قضیه universal approximation، می‌دانیم که با یک شبکه عصبی که فقط یک لایه پنهان دارد و تابع activation آن غیرخطی است، هر تابع پیوسته در یک مجموعه finite را می‌توان تقریب زد.

د) خیر، ممکن است در صورت کم بودن دیتای train، باعث پیچیده شدن بیش از حد مدل شویم و مشکلاتی مثل overfitting اتفاق بیفتد یا حتی مشکلاتی مثل ناپدید شدن گرادیان پیش آید.

هـ) طبق همان بخش ج و قضیه universal approximation می‌دانیم که با تشکیل یک شبکه wide و تعداد زیادی نورون تقریباً هر تابعی را می‌توان ساخت و تقریب زد. ولی در شبکه‌های عمیق با تعداد نورون کمتری می‌توانیم همان کار را انجام دهیم.

و) ناپدید شدن گرادیان عبارت است از وقتی که گرادیان ما نسبت به پارامترها خیلی کوچک است و باعث می‌شود آپدیت ما خیلی کند انجام شود و عملاً آپدیتی نداشته باشیم. توابعی مثل ReLU، مشتق شان همواره ۱ است و تغییری نمی‌کند و کوچک نمی‌شود و باعث ایجاد پدیده ناپدید شدن گرادیان نمی‌شود.

ز) درست است در SGD چون نسبت به هر سمپل آپدیت می‌کنیم ← فرکانس بالا می‌رود و سریع میل می‌کند

ولی ممکن است ترتیب مینیمم محلی هی تغییر کند، قبل از converge کردن

ح)

$$\frac{\partial y}{\partial x} = 1.2x^3 - 0.3x^2 - 4x - 0.8 \xrightarrow{x_0 = -2.8} \underbrace{= -18.2944}_{g_1}$$

m_0 را صفر در نظر می‌گیریم.

$$\begin{cases} m_{t+1} = \mu m_t + (1-\mu)\, \nabla_x J(x) \\ x_{t+1} = x_t - \gamma\, m_{t+1} \end{cases}$$

$$m_1 = 0.7 \times 0 + 0.3 \times (-18.2944) = -5.48832$$

$$x_1 = -2.8 - (0.05)(-5.48832) = -2.525584$$

$$m_2 = 0.7(-5.48832) + 0.3(g_2) \approx -7.4297$$

$$g_2 = (1.2x^3 - 0.3x^2 - 4x - 0.8) \xrightarrow{x = -2.525584} \approx -11.94278$$

$$x_2 \approx -2.525584 - (0.05)(-7.4297) = -2.15435$$

۲- الف) ابتدا تابعی که ماکزیمم بین دو عدد را خروجی می‌دهد را می‌سازیم، با استفاده از ReLU، طوری که اگر $a_1 > a_2 \rightarrow ReLU(a_1 - a_2) = a_1 - a_2$ وگرنه خروجی صفر است ←

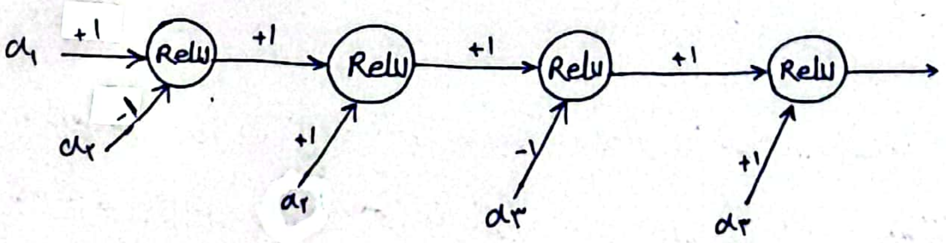
$$ReLU(a_1 - a_2) + a_2 = \max\{a_1, a_2\}$$

$$\text{if } a_1 > a_2 \longrightarrow a_1 - a_2 + a_2 = a_1$$

$$\text{else} \longrightarrow 0 + a_2 = a_2$$

$$\max\{a_1, a_2, a_3\} = \max\{\max\{a_1, a_2\}, a_3\}$$

حالا تعمیم می‌دهیم به ۳ تا

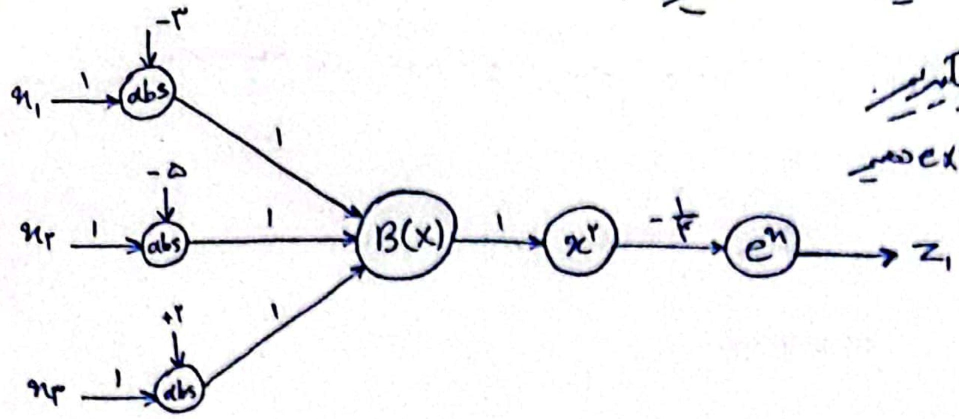


پس این می‌شود شبکه/ نورونی که ماکزیمم سه ورودی را خروجی می‌دهد.

تابع قدرمطلق ← abs

اسم این شبکه را $B(X)$ می‌گذاریم، حالا داریم که:

باید بین این سه مورد ماکزیمم بگیریم
سپس باید $-\frac{1}{4}$ برابر آنرا به توان 2 به exp دهیم



حالا شبکه مربوط به Z_2 را می‌سازیم:

باید داشته باشیم که:

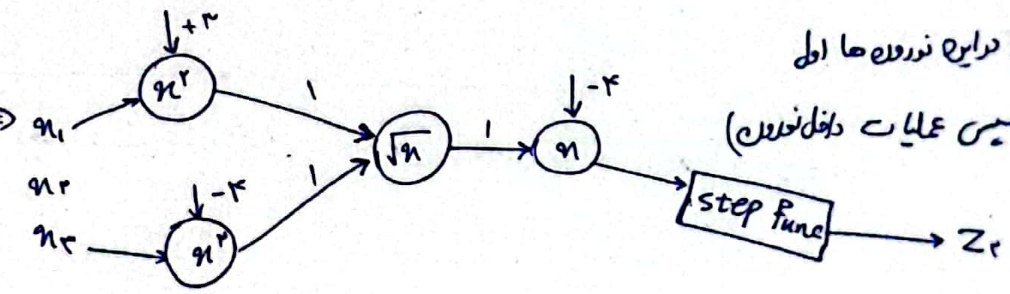
$$\begin{cases} \text{if } \sqrt{(x_1+3)^2+(x_3-4)^2} > 4 \longrightarrow 0 \\ \text{else} \longrightarrow 1 \end{cases}$$

یعنی اگر $\sqrt{(x_1+3)^2+(x_3-4)^2}-4 \le 0$ ← مقدار تابع 1 است و درغیر اینصورت صفر است.

یک step func تعریف می‌کنیم که اگر مقدار ورودی اش بزرگتر مساوی صفر بود ← یک خروجی دهد وگرنه صفر خروجی دهد.

(دقت شود که ورودی‌ها را عدد در این نورون‌ها اول یعنی ورودی اعمال شوند، سپس عملیات داخل نورون)

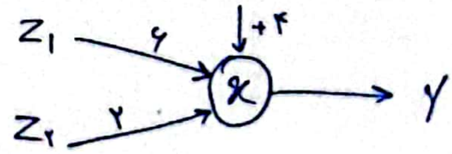
$\Rightarrow$



حالا داریم که:

$Y = 6Z_1 + 2Z_2 + 4$

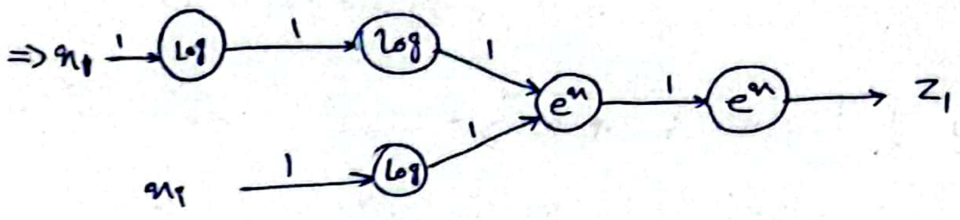
$\Rightarrow$



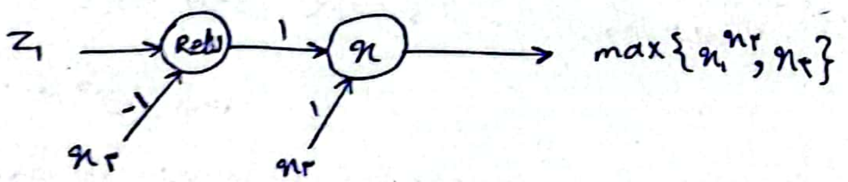
ب) توان را نمی توانیم بنویسیم و آنرا باید به صورت جمع در بیاوریم. پس log ها بگیریم تا توان تبدیل به ضرب شود و دوباره log ها بگیریم، ضرب به جمع تبدیل شود.

$$x_1^{x_2} \xrightarrow{\log} x_2 \log x_1 \xrightarrow{\log} \log x_2 + \log\log x_1$$

$$\Rightarrow x_1^{x_2} = e^{e^{(\log x_2 + \log\log x_1)}}$$



حالا مانند بخش قبل max را با استفاده از ReLU می سازیم،



۳- بعد ورودی عبارت است $D_x \times 1$، بعد z_1، $D_{a_1} \times 1$ است.

پس داریم که:

۱. بعد W_1 عبارت است از $\underline{D_{a_1} \times 1 \times D_x \times 1}$ همچنین بعد b_1 برابر است با بعد z_1 ← $D_{a_1} \times 1$

خروجی هم اسکالر است ← پس بعد b_2 برابر است با 1×1 و بعد W_2 برابر است با $1 \times D_{a_1}$

اگر بخواهیم n تا سمپل داشته باشیم، بعد x می شود: $D_x \times n$

و بعد a_1 و z_1 عبارت می شوند از: $D_{a_1} \times n$، در نتیجه بعد W_1 تغییر نمی کند.

جمله بایاس هر سمپل باید عین قبل خروجی دهیم وقتی داریم روی تعدادی سمپل این ماتریس را ضرب می کنیم و همان نتیجه را می دهد.

ولی بعد ۲ باید تغییر کند و باید در n ضرب شود. (n بار تکرار شود) ← $D_{a_1} \times n$

$$\Longrightarrow \quad W_1 : D_{a_1} \times D_x$$

$$W_2 : 1 \times D_{a_1}$$

$$X : D_x \times n$$

$$\hat{Y} : 1 \times n$$

۲.

$$\delta_1^{(i)} = \frac{\partial J}{\partial \hat{y}^{(i)}} = -\frac{1}{m}\left(\frac{\partial L^{(i)}}{\partial \hat{y}^{(i)}}\right) = -\frac{1}{m}\left(\frac{y^{(i)}}{\hat{y}^{(i)}} - \frac{(1-y^{(i)})}{1-\hat{y}^{(i)}}\right) = \frac{1}{m}\left(\frac{(1-y^{(i)})}{1-\hat{y}^{(i)}} - \frac{y^{(i)}}{\hat{y}^{(i)}}\right)$$

$$\frac{\partial J}{\partial \hat{y}} = \left[\frac{\partial J}{\partial \hat{y}^{(1)}}, \dots, \frac{\partial J}{\partial \hat{y}^{(n)}}\right]^T = \begin{bmatrix} \frac{1-y^{(1)}}{1-\hat{y}^{(1)}} - \frac{y^{(1)}}{\hat{y}^{(1)}} \\ \vdots \\ \frac{1-y^{(n)}}{1-\hat{y}^{(n)}} - \frac{y^{(n)}}{\hat{y}^{(n)}} \end{bmatrix}$$

۳.

$$\delta_2^{(i)} = \frac{\partial \hat{y}^{(i)}}{\partial z_2} = \frac{\partial \sigma(z_2)}{\partial z_2} = \sigma(z_2)\left(1-\sigma(z_2)\right) = \hat{y}^{(i)}\left(1-y^{(i)}\right)$$

۴.

$$\delta_3^{(i)} = \frac{\partial z_2}{\partial a_1} = \frac{\partial(w_2 a_1 + b_2)}{\partial a_1} = W_2$$

۵.

$$\delta_4^{(i)} = \frac{\partial a_1}{\partial z_1} = \frac{\partial \, ReLU(z_1)}{\partial z_1} = \begin{cases} 1 & z_1 > 0 \\ 0 & \text{o.w.} \end{cases}$$

۶.

$$\delta_5^{(i)} = \frac{\partial z_1}{\partial w_1} = \frac{\partial(w_1 x^{(i)} + b_1)}{\partial w_1} = x^{(i)\,T}$$

۷. داریم که:

$$\frac{\partial J}{\partial w_1} = \frac{\partial J}{\partial \hat{y}} \times \frac{\partial \hat{y}}{\partial z_2} \times \frac{\partial z_2}{\partial a_1} \times \frac{\partial a_1}{\partial z_1} \times \frac{\partial z_1}{\partial w_1} = \sim$$

ضرب عبارات بالا هست

-۴

$$z_1 = W_1 x + \vec{b}_1^{\,0} = \begin{bmatrix} a & b \\ -a & a \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} a \\ -a \end{bmatrix} \longrightarrow \sigma(z_1) = \begin{bmatrix} \frac{1}{1+e^{-a}} \\ \frac{1}{1+e^{a}} \end{bmatrix}$$

$$z_2 = W_2 z_1 + \vec{b}_2^{\,0} = \begin{bmatrix} a & -b \\ b & -a \end{bmatrix}\begin{bmatrix} \frac{1}{1+e^{-a}} \\ \frac{1}{1+e^{a}} \end{bmatrix} = \begin{bmatrix} \frac{a}{1+e^{-a}} - \frac{b}{1+e^{a}} \\ \frac{b}{1+e^{-a}} - \frac{a}{1+e^{a}} \end{bmatrix} \begin{matrix} \longrightarrow \frac{ae^{a}-b}{1+e^{a}} \\ \longrightarrow \frac{be^{a}-a}{1+e^{a}} \end{matrix}$$

$$\mathrm{ReLU}(z_2) = \begin{bmatrix} \frac{\mathrm{ReLU}(ae^{a}-b)}{1+e^{a}} \\ \frac{\mathrm{ReLU}(be^{a}-a)}{1+e^{a}} \end{bmatrix}$$

$$z_3 = W_3 z_2 + \vec{b}_3^{\,0} = \begin{bmatrix} a & -a \\ -b & b \end{bmatrix}\begin{bmatrix} \frac{\mathrm{ReLU}(ae^{a}-b)}{1+e^{a}} \\ \frac{\mathrm{ReLU}(be^{a}-a)}{1+e^{a}} \end{bmatrix} = \cancel{\frac{1}{1+e^{a}}\left[a\,\mathrm{ReLU}(ae^{a}-b)\right.}$$

$$\begin{bmatrix} \frac{a\,\mathrm{ReLU}(ae^{a}-b) - a\,\mathrm{ReLU}(be^{a}-a)}{1+e^{a}} \\ \frac{-b\,\mathrm{ReLU}(ae^{a}-b) + b\,\mathrm{ReLU}(be^{a}-a)}{1+e^{a}} \end{bmatrix} \begin{matrix} \rightarrow s_1 \\ \rightarrow s_2 \end{matrix}$$

$$\longrightarrow \mathrm{softmax}(z_3) = \begin{bmatrix} \frac{\exp(s_1)}{\exp(s_1)+\exp(s_2)} \\ \frac{\exp(s_2)}{\exp(s_1)+\exp(s_2)} \end{bmatrix}$$

$$z_4 = \begin{bmatrix} a & 0.5a \\ -b & a \end{bmatrix}\begin{bmatrix} \frac{e^{s_1}}{e^{s_1}+e^{s_2}} \\ \frac{e^{s_2}}{e^{s_1}+e^{s_2}} \end{bmatrix} = \begin{bmatrix} \frac{ae^{s_1}+0.5a\,e^{s_2}}{e^{s_1}+e^{s_2}} \\ \frac{-be^{s_1}+ae^{s_2}}{e^{s_1}+e^{s_2}} \end{bmatrix} \begin{matrix} \rightarrow y_1 \\ \rightarrow y_2 \end{matrix}$$

output

$$a_4 = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

ب)

$$E = \frac{1}{2}\left((y_1 - t_1)^2 + (y_2 - t_2)^2 \right) = \frac{1}{2}\left((y_1 - 0)^2 + (y_2 - 1)^2 \right) =$$

$$\frac{1}{2}\left(\frac{ae^{s_1} + 0/5\,a e^{s_2}}{e^{s_1} + e^{s_2}}\right)^2 + \frac{1}{2}\left(\frac{-be^{s_1} + ae^{s_2}}{e^{s_1} + e^{s_2}} - 1\right)^2$$

ج)

$$\frac{\partial E}{\partial a_4} = a_4 - t \qquad \frac{\partial E}{\partial a_3} = \frac{\partial E}{\partial a_4}\,\frac{\partial a_4}{\partial a_3} = W_4^T (a_4 - t)$$

$$\frac{\partial E}{\partial W_4} = \frac{\partial E}{\partial a_4}\,\frac{\partial a_4}{\partial W_4} = (a_4 - t)\, a_3^T$$

$$\frac{\partial E}{\partial b_4} = \frac{\partial E}{\partial a_4}\,\frac{\partial a_4}{\partial b_4} = (a_4 - t) \times 1 = (a_4 - t)$$

$$\frac{\partial E}{\partial z_3} = \frac{\partial E}{\partial a_3}\,\frac{\partial a_3}{\partial z_3} = \left(\mathrm{diag}(a_3) - a_3 a_3^T\right) W_4^T (a_4 - t)$$

$$\frac{\partial E}{\partial W_3} = \frac{\partial E}{\partial z_3}\,\frac{\partial z_3}{\partial W_3} = \left(\mathrm{diag}(a_3) - a_3 a_3^T\right) W_4^T (a_4 - t)\, a_2$$

$$\frac{\partial E}{\partial b_3} = \frac{\partial E}{\partial z_3}\,\overset{\nearrow 1}{\frac{\partial z_3}{\partial b_3}} = \left(\mathrm{diag}(a_3) - a_3 a_3^T\right) W_4^T (a_4 - t)$$

$$\frac{\partial E}{\partial a_2} = \frac{\partial E}{\partial z_3}\,\frac{\partial z_3}{\partial a_2} = \frac{\partial E}{\partial z_3}\,\frac{\partial z_3}{\partial a_2} = W_3^T \left(\mathrm{diag}(a_3) - a_3 a_3^T\right) W_4^T (a_4 - t)$$

اسمش را ماتریس A میذاریم

$$\frac{\partial E}{\partial z_2} = \frac{\partial E}{\partial a_2}\,\frac{\partial a_2}{\partial z_2} = \underbrace{\left[\quad\right]}\; W_3^T \left(\mathrm{diag}(a_3) - a_3 a_3^T\right) W_4^T (a_4 - t)$$

بسته به اینکه مثبت یا منفی اند عبارات ← مشتق $\mathrm{ReLU}(z_2)$ به z_2 که هر درایه می تواند 0 یا 1 باشد

$$\frac{\partial E}{\partial w_2} = \frac{\partial E}{\partial z_2}\frac{\partial z_2}{\partial w_2} = A w_3^T\left(\mathrm{diag}(a_3) - a_3 a_3^T\right) w_4^T (a_4 - t)$$

$$\frac{\partial E}{\partial b_2} = \frac{\partial E}{\partial z_2}\frac{\partial z_2}{\partial b_2} = A w_3^T\left(\mathrm{diag}(a_3) - a_3 a_3^T\right) w_4^T (a_4 - t)$$

$$\frac{\partial E}{\partial a_1} = \frac{\partial E}{\partial z_2}\frac{\partial z_2}{\partial a_1} = w_2^T A w_3^T\left(\mathrm{diag}(a_3) - a_3 a_3^T\right) w_4^T (a_4 - t)$$

$\odot$ ضرب المنت وایز

$$\frac{\partial E}{\partial z_1} = \frac{\partial E}{\partial a_1}\frac{\partial a_1}{\partial z_1} = \mathrm{diag}\left(a_1(1 - a_1)\right) w_2^T A w_3^T\left(\mathrm{diag}(a_3) - a_3 a_3^T\right) w_4^T (a_4 - t)$$

$$\frac{\partial E}{\partial b_1} = \frac{\partial E}{\partial z_1}\frac{\partial z_1}{\partial b_1} = \mathrm{diag}\left(a_1 \odot (1 - a_1)\right) w_2^T A w_3^T\left(\mathrm{diag}(a_3) - a_3 a_3^T\right) w_4^T (a_4 - t)$$

ت) مقدار وزن‌ها و بایاس‌ها را با استفاده از گرادیان کاهشی و نرخ یادگیری η می‌کاهیم

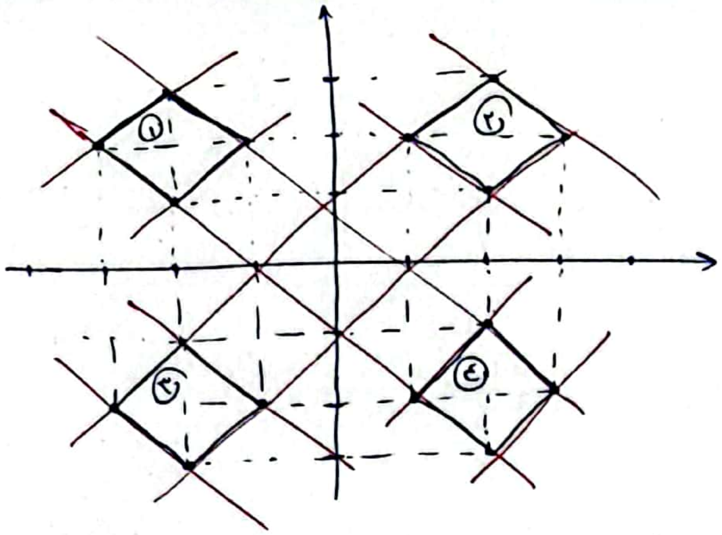
با استفاده از شبه‌کد زیر بدست آوریم:

```
For key in params.key():
    params[key] -= η grads[key]
```

تمام مقادیر وزن‌ها و بایاس‌ها را منهای η برابر گرادیان متناظرشان $(\partial E / \partial p)$ می‌کنیم.

پارامتر

-۵



الف) ۱۶ خط نیاز است که ۴ تا از آنها مشترک هستند.

← ۱۲ خط نیاز داریم.

برای هر نورون هم ۴ تا and نیاز داریم.

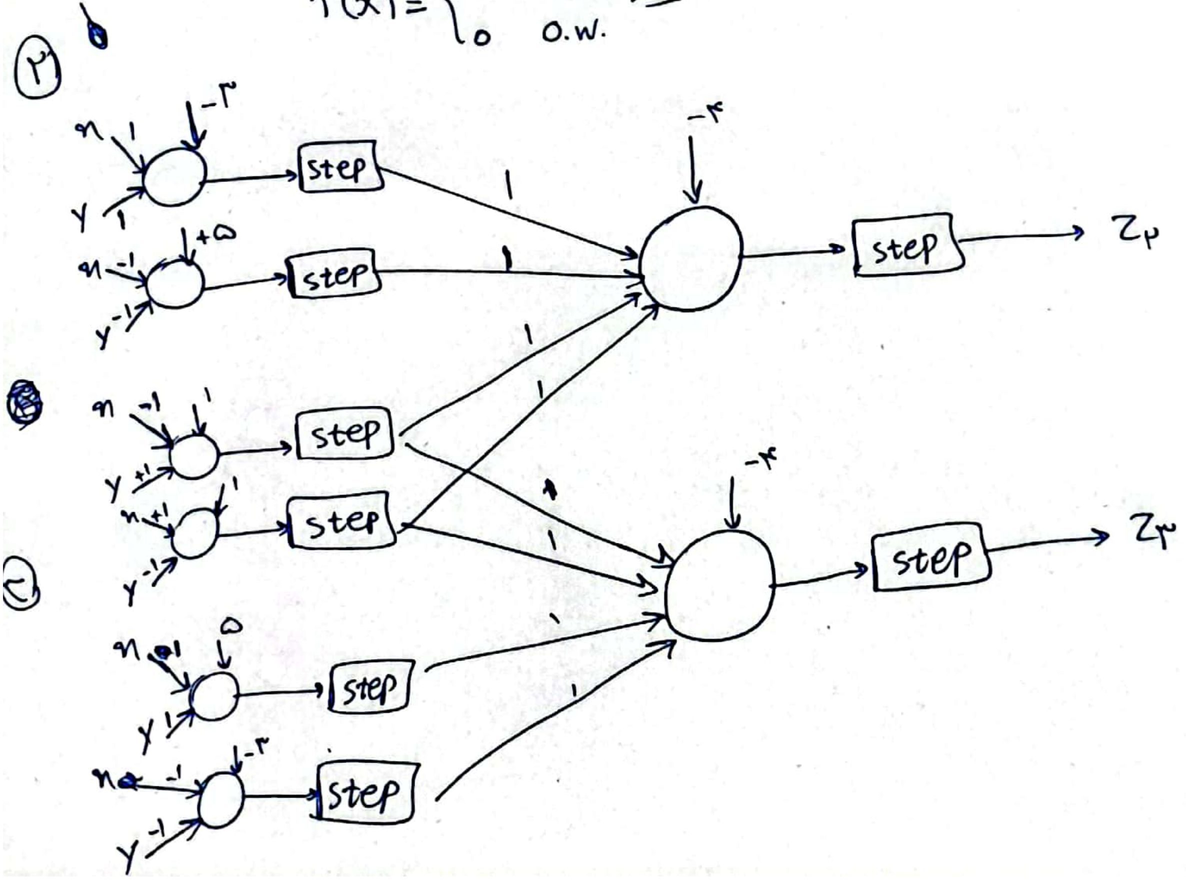
و در نهایت هم تمام لوزی ها را باید or کنیم.

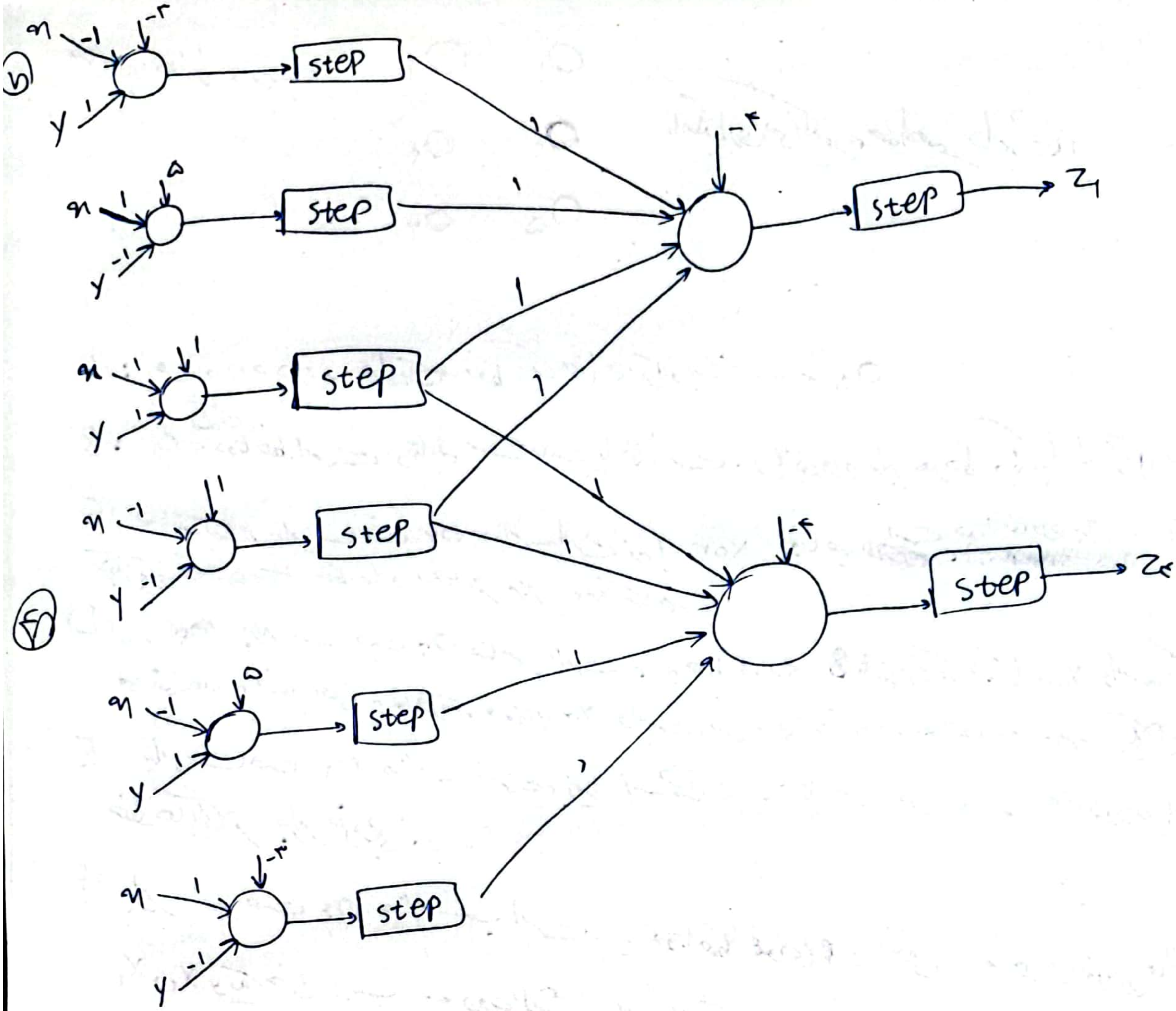
① $\begin{cases} y+x \le 1 \\ y+x \ge -1 \\ y-x \le 5 \\ y-x \ge 3 \end{cases}$ ② $\begin{cases} y+x \le 5 \\ y+x \ge 3 \\ y-x \le 1 \\ y-x \ge -1 \end{cases}$ ③ $\begin{cases} y+x \le -3 \\ y+x \ge -5 \\ y-x \le 1 \\ y-x \ge -1 \end{cases}$ ④ $\begin{cases} y+x \le 1 \\ y+x \ge -1 \\ y-x \le -3 \\ y-x \ge -5 \end{cases}$

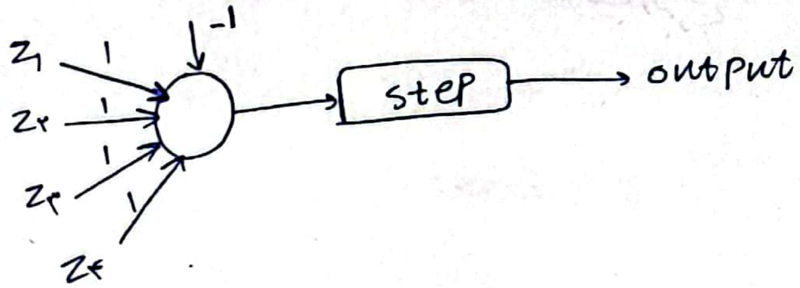
① و ④ با هم اشتراک دارند ② و ③ هم با هم اشتراک دارند، حالا جدا جدا می‌کشیم برای هر بخش، تابع step func هم در سوال قبل تعریف کرده‌ایم،

$$f(x) = \begin{cases} 1 & x \ge 0 \\ 0 & \text{o.w.} \end{cases}$$

ب)







or کردن خروجی ها

← در کل به ۱۷ نورون نیاز داریم.

ب) این نورون به ۸ خط نیاز دارد و یک and و ما در مدل قبلی بیش از این تعداد نورون داریم بنابراین با تغییر وزن ها می توان این مدل را نیز شبیه سازی کرد

خروجی ها را به ترتیب O_1 O_2

نامگذاری می کنیم، خواهیم داشت: O_3 O_4

O_5 O_6

A: یه نورون داره و فقط یک خط می تواند بسازد ⟵ O_5

B: O_1، دوتا خط است و قابل ساخت با B است. (۲ نورون برای هر خط و یکی برای اشتراک)

C: برای ساخت O_6 مناسب است زیرا xor است. دوتا نورون برای ساخت خطها و ۳تا هم برای xor کردن.

D: برای ساخت O_3 مناسب است، یه خط عمودی ۲ نورون که فقط به X_1 وابسته است می خواهد، یه خط افقی یه نورون که فقط به X_2 وابسته است می خواهد. ⟵ D مناسب است.

E: برای ساخت O_2 مناسب است، زیرا اشتراک ۳ تا خط است ⟵ ۳ تا نورون برای خط ها یکی هم برای خروجی.

F: برای ساخت O_4 مناسب است. دوتا خط عمودی و افقی ⟵ ورودی نورون ها X_1 و X_2 و یک خط اریب ⟵ ورودی اش X_1 و X_2 و یک نورون هم برای خروجی ⟵ F برای ساخت O_4 مناسب است.

Scanned with CamScanner

۶- می دانیم که داریم،

$$Z^{(L+1)} = W^{(L)} h^{(L)} + b^{(L)}$$

$$h^{(L+1)} = \mathrm{ReLU}(Z^{(L+1)})$$

حالا اگه $z_i^{[L]} \geq 0$ ⟵ خروجی ReLU برابر خود عبارت است. عبارت $1x$

$z_i^{[L]} < 0$ ⟵ خروجی ReLU برابر صفر است. عبارت $0x$

پس بردار $c^{[L]}$ را تعریف می کنیم برای هر لایه و آنرا در $z^{[L]}$ ضرب المنت وایز می کنیم،

$$c_i^{[L]} = 1 \quad \text{iff} \quad z_i^{[L]} \geq 0$$

$$c_i^{[L]} = 0 \quad \text{iff} \quad z_i^{[L]} < 0 \longrightarrow h^{[L]} = c^{[L]} \odot z^{[L]}$$

حالا این را در معادله بالاتر جایگذاری می کنیم، خواهیم داشت،

$$Z^{[L+1]} = W^{[L]} h^{[L]} + b^{[L]} = W^{[L]}(c^{[L]} \odot z^{[L]}) + b^{[L]}$$

می توانیم آن درایه ای که $c^{[L]}$، آنرا در $z^{[L]}$ صفر می کند را ستون متناظرش در $W^{[L]}$ را صفر کنیم و به عبارتی $\hat{W}^{[L]}$ بجای $W^{[L]}$ استفاده کنیم به صورت $\hat{W}^{[L]} Z^{[L]}$ (هر سطر $W^{[L]}$ را ضرب المنت وایز در c می کنیم تا $\hat{W}$ حاصل شود.)

$$\longrightarrow \hat{W}^{[L]} Z^{[L]} + b^{[L]}$$

$$\rightsquigarrow Z^{[L+1]} = \hat{W}^{[L]}(\hat{W}^{[L-1]}(\cdots) + b^{[L-1]}) + b^{[L]} =$$

$$\left(\prod_{k=1}^{L} \hat{W}^{[k]}\right) Z^{[1]} + \sum_{i=1}^{L}\left(\prod_{k=i+1}^{L} \hat{W}^{[k]}\right) b^{[i]}$$

از طرفی $Z^{[1]} = W^{[0]} x + b^{[0]}$

پس داریم که،

$$\underbrace{\left(\prod_{k=0}^{L} \hat{W}^{[k]}\right)}_{\tilde{W}^{[L]}} x + \sum_{i=0}^{L} \underbrace{\left(\prod_{k=i+1}^{L} \hat{W}^{[k]}\right) b^{[i]}}_{\tilde{b}^{[i]}}$$

$\tilde{W}^{[L]}$ می‌نامیم. $\tilde{b}^{[i]}$ می‌نامیم

~~حالا~~

حالا $\tilde{b}^{[i]}$ و $\tilde{W}^{[L]}$ متغیرهای ثابتی هستند و به صورت یکتا توسط activation function ها تعیین می‌شوند، پس داریم که،

$$\left(2c^{[L+1]}-1\right)\circ\left(\tilde{W}^{[L]}x+\tilde{b}^{[L]}\right)\geq 0$$

ب) اگر $m > k$:

$$r(k,m) \leq \sum_{i=0}^{k}\binom{k}{i} = 2^k \leq k^k \leq k^m$$

اگر $m \leq k$،

$$r(k,m) \leq \sum_{i=0}^{m}\binom{k}{i}\left(\frac{k}{m}\right)^{m-i} \leq \sum_{i=0}^{k}\binom{k}{i}\left(\frac{k}{m}\right)^{m-i} = \left(\frac{k}{m}\right)^m \sum_{i=0}^{k}\binom{k}{i}\left(\frac{k}{m}\right)^{-i}$$

$1-m \leq e^{-m}$

$$= \left(\frac{k}{m}\right)^m \sum_{i=0}^{k}\binom{k}{i}\left(\frac{m}{k}\right)^{i} = \left(\frac{k}{m}\right)^m\left(1+\frac{m}{k}\right)^k \leq \left(\frac{e}{m}\right)^m k^m \leq O(k^m)$$

از عبارت اول هر دو شتیم هر لایه یک پلی توپ تشکیل می‌دهد.

حالا هر لایه نهایتاً $O(k^m)$ تا خط اضافه می‌کند در نتیجه،

$$① \to O(k^m) \qquad ② \to O(k^{2m}) \;\ldots$$

در نهایت برای n تا $O(k^{nm})$ است.