



دانشکده مهندسی کامپیوتر

**گزارش پایانی درس: مبانی هوش محاسباتی**

**عنوان پژوهش: تمرین سوم بخش اول (سوالات مفهومی در شبکه های عصبی مصنوعی)**

**ارائه دهنده:**

**فرگس سادات موسوی جد**

**شادی شاهی محمدی**

**استاد درس:**

**دکتر کارشناس**

**بهار ۱۴۰۴**

## فهرست:

۱-سوال ۱:	۱
۲-سوال ۲:	۲
۳-سوال ۳:	۳
۴- سوال ۴:	۴
۵- سوال ۵:	۴
۶-سوال ۶:	۵
۷-سوال ۷:	۸
۸-سوال ۸:	۱۳
۹-سوال ۹:	۱۵

## ۱- سوال ۱:

منظور از اطلاعات مشخص، واکنش نورون به یک الگوی خاص یا ویژگی معین در داده‌های ورودی است. به‌طور مثال، در شبکه‌های عصبی پیچشی (CNN)، برخی نورون‌ها به لبه‌ها، بافت‌ها، یا اشیاء خاصی در تصویر واکنش نشان می‌دهند. البته باید توجه داشت که این اطلاعات معمولاً به صورت توزیعی میان چندین نورون تقسیم می‌شود، نه اینکه هر نورون به تنهایی معنای کاملاً مستقلی داشته باشد.

پاسخ، مبتنی بر چند ویژگی شبکه‌های عصبی است:

**وزن‌ها:** نخستین و مهم‌ترین ویژگی، وزن‌های قابل یادگیری است. این وزن‌ها در طی فرآیند آموزش با داده‌ها تطبیق پیدا می‌کنند و مشخص می‌کنند هر نورون به چه ویژگی یا الگویی حساس باشد. به عبارت دیگر، وزن‌ها تعیین می‌کنند که نورون به کدام جنبه‌های ورودی واکنش نشان دهد و به این ترتیب حامل نوعی اطلاعات مشخص می‌شود.

**توابع فعال‌ساز:** ویژگی دوم، استفاده از توابع فعال‌سازی غیرخطی است. این توابع امکان شناسایی و بازنمایی روابط پیچیده و غیرخطی میان داده‌ها را فراهم می‌کنند. به کمک این توابع، نورون‌ها قادرند به الگوهای خاص و معناداری واکنش نشان دهند که در غیر این صورت با مدل‌های خطی ساده ممکن نبود. بنابراین، فعال‌سازی غیرخطی باعث می‌شود که نورون‌ها حامل اطلاعاتی باشند که نه فقط بر اساس شدت ورودی بلکه بر اساس ساختار پیچیده آن شکل می‌گیرد.

**نمایش توزیعی:** ویژگی سوم، مفهوم نمایش توزیعی است؛ به این معنا که اطلاعات به جای اینکه در یک نورون منفرد متمرکز شود، بین مجموعه‌ای از نورون‌ها توزیع می‌شود. این ویژگی باعث می‌شود هر نورون بخشی از یک الگوی کلی را نمایش دهد و تنها در کنار سایر نورون‌ها معنای کامل و مشخصی پیدا کند. به همین دلیل، اطلاعاتی که یک نورون حمل می‌کند، هم مستقل نیست و هم در ترکیب با دیگر نورون‌ها معنا پیدا می‌کند.

در نهایت، ساختار چندلایه و عمق شبکه نیز اهمیت دارد. شبکه‌های عصبی عمیق با داشتن لایه‌های متعدد این امکان را می‌دهند که نورون‌ها در هر لایه به سطوح مختلفی از ویژگی‌ها حساس باشند؛ از ویژگی‌های ساده در لایه‌های ابتدایی تا مفاهیم پیچیده‌تر در لایه‌های عمیق‌تر. این معماری باعث می‌شود هر نورون به نوعی حامل اطلاعات مشخص و معناداری در چارچوب وظیفه کلی شبکه باشد.

## ۲- سوال ۲:

دانش در شبکه‌های عصبی مصنوعی به صورت مجموعه‌ای از وزن‌ها و پارامترهای اتصال بین نورون‌ها شکل می‌گیرد. این شبکه‌ها ساختاری شبیه به مغز انسان دارند که از نورون‌های مصنوعی تشکیل شده‌اند و این نورون‌ها در لایه‌های مختلف سازماندهی شده‌اند. در ابتدا وزن‌ها به صورت تصادفی مقداردهی می‌شوند و شبکه هیچ دانشی درباره مسئله ندارد.

فرایند شکل‌گیری دانش زمانی آغاز می‌شود که شبکه با استفاده از داده‌های آموزشی، خروجی صحیح را برای هر ورودی یاد می‌گیرد. شبکه با مقایسه خروجی خودش با خروجی مطلوب، خطا را محاسبه می‌کند و سپس با استفاده از الگوریتم‌هایی مانند پس‌انتشار خطا و گرادیان کاهشی، وزن‌ها را به تدریج تغییر می‌دهد تا این خطا کم شود. به این ترتیب، شبکه به تدریج ساختار وزن‌هایی را پیدا می‌کند که بهترین پاسخ‌ها را تولید می‌کنند.

در نهایت، وزن‌های بهینه شده که نتیجه این فرایند آموزش هستند، در واقع همان دانش شبکه محسوب می‌شوند. این دانش به شبکه امکان می‌دهد تا الگوهای پنهان در داده‌ها را شناسایی کند و برای ورودی‌های جدید پیش‌بینی یا تصمیم‌گیری مناسبی انجام دهد. به این صورت، شبکه عصبی می‌تواند از داده‌های گذشته خود یاد بگیرد و عملکردش را در مواجهه با داده‌های تازه بهبود دهد.

معادل بودن دو شبکه عصبی به معنای این است که این دو شبکه، برای هر ورودی مشابه، خروجی یکسان یا تقریباً یکسان تولید کنند. اگر  $f_1$  و  $f_2$  خروجی دو شبکه عصبی با وزن‌ها  $W_1$  و  $W_2$  باشند؛ آنگاه این دو شبکه برابرند اگر:

$$\forall x, f_1(x, W_1) = f_2(x, W_2)$$

از نظر عملی، به دلیل پیچیدگی و ویژگی‌های عددی شبکه‌ها، معمولاً این برابری را به صورت تقریب در نظر می‌گیرند. یعنی دو شبکه را معادل می‌دانند اگر اختلاف خروجی آن‌ها برای همه ورودی‌ها کمتر از یک مقدار کوچک مثبت  $\epsilon$  باشد. همچنین، در بسیاری موارد این معادل بودن فقط روی یک مجموعه مشخص از ورودی‌ها مثل داده‌های آموزشی یا آزمون بررسی می‌شود، چون بررسی تمام ورودی‌های ممکن معمولاً عملی نیست.

### ۳- سوال ۳:

**یادگیری:** شبکه‌های عصبی توانایی یادگیری را از طریق تنظیم وزن‌ها و پارامترهای داخلی خود به دست می‌آورند. این وزن‌ها بر اساس داده‌های آموزشی به گونه‌ای تغییر می‌کنند که شبکه بتواند الگوها و روابط موجود در داده‌ها را شناسایی کند.

**به خاطر سپاری:** شبکه، اطلاعات مهم و الگوها را به صورت وزن‌های بهینه‌شده، به صورت پایدار ذخیره می‌کند و این همان چیزی است که به عنوان «به خاطر سپردن» شناخته می‌شود.

**تعمیم:** تعمیم دادن یعنی توانایی شبکه در استفاده از دانشی که از داده‌های آموزشی کسب کرده تا بتواند پاسخ مناسبی برای داده‌های جدید و دیده نشده ارائه دهد. این قابلیت به شبکه اجازه می‌دهد فراتر از نمونه‌های خاص حرکت کند و الگوهای کلی‌تر را شناسایی کند. فرایند آموزش، به کمک الگوریتم‌هایی مثل پس‌انتشار خطا و روش‌های بهینه‌سازی، باعث می‌شود شبکه به سمت مدل‌هایی برود که نه تنها داده‌های آموزش را حفظ کنند، بلکه بتوانند رفتار مناسبی برای داده‌های جدید نیز داشته باشند.

یکی از اصول کلیدی که باعث شکل‌گیری قابلیت‌های یادگیری و تعمیم در شبکه‌های عصبی می‌شود، معماری چندلایه و غیرخطی بودن نورون‌ها است. این ویژگی به شبکه اجازه می‌دهد تا توابع پیچیده و غیرخطی را تقریب بزند و الگوهای مختلف را در داده‌ها کشف کند. به عبارت دیگر، هر لایه شبکه به صورت مرحله‌ای ویژگی‌های سطح پایین را به ویژگی‌های سطح بالاتر تبدیل می‌کند که این روند باعث یادگیری عمیق‌تر و استخراج اطلاعات مهم می‌شود.

همچنین، فرایند آموزش شبکه که مبتنی بر بهینه‌سازی وزن‌هاست، نقش بسیار مهمی دارد. الگوریتم‌هایی مثل پس‌انتشار خطا و گرادیان کاهشی، وزن‌های شبکه را به گونه‌ای تنظیم می‌کنند که خطا کاهش یابد و شبکه بتواند رفتار مناسبی برای داده‌های آموزشی یاد بگیرد. این تغییر تدریجی وزن‌ها در طول آموزش همان فرایند به خاطر سپردن اطلاعات است.

علاوه بر این، فضای پارامتری بزرگ شبکه‌ها امکان نمایش انعطاف‌پذیر داده‌ها را فراهم می‌کند، اما برای جلوگیری از بیش‌برازش و بهبود تعمیم‌پذیری، از تکنیک‌هایی مثل تنظیم (regularization)، توقف زودهنگام و سایر روش‌ها استفاده می‌شود. این ابزارها کمک می‌کنند تا شبکه به جای حفظ کردن صرف نمونه‌های آموزشی، الگوهای کلی‌تر را یاد بگیرد و بتواند برای داده‌های جدید عملکرد مناسبی داشته باشد.

## ۴- سوال ۴:

الف) تابع درجه دوم :

پارامترها:  $a, b, c$

$$Y = a(WX^T)^2 + b(WX^T) + c$$

ب) تابع کروی :

پارامترها:  $c$  که مرکز کره می باشد.

$$Y = \|WX^T - c\|$$

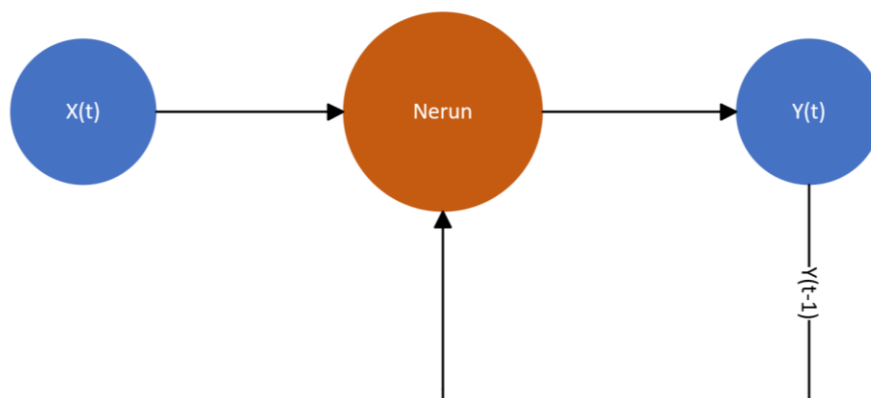
ج) تابع چندجمله ای :

پارامترها:  $a_k$  ضرایب چندجمله ای.

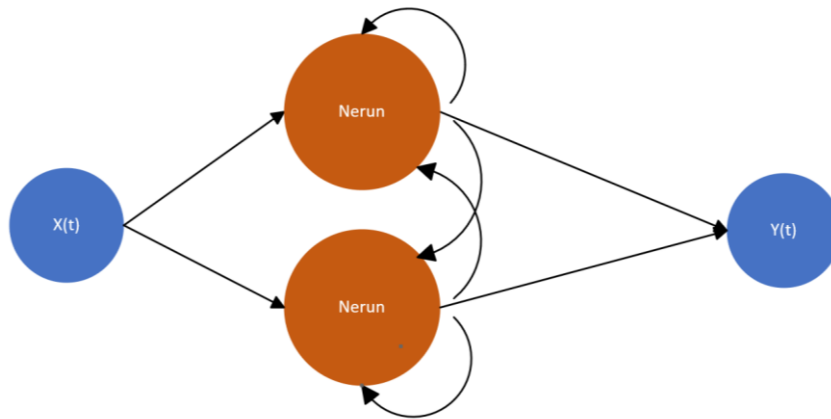
$$Y = \sum_{k=0}^n a_k (WX^T)^k$$

## ۵- سوال ۵:

الف) شبکه عصبی تک نورونی با فیدبک به خود: مقدار خروجی نورون به خروجی قبلی خود ربط دارد.

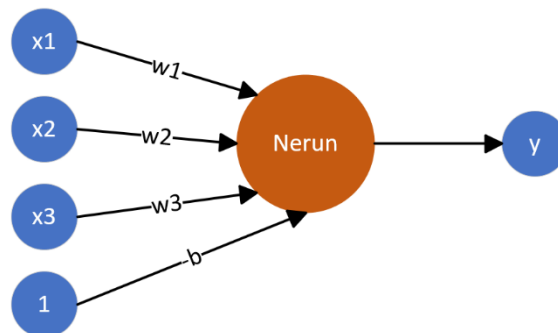


ب) شبکه عصبی تک لایه با فیدبک: در حالت کلی مقدار خروجی هر نورون به خروجی قبلی همه‌ی نورون‌های آن لایه در بستگی دارد. البته در معماری‌های ساده‌تر خروجی یک نورون فقط به خروجی قبلی همان نورون بستگی دارد.



## ۶- سوال ۶:

الف) یک پرسپترون ساده تک نورونی بصورت زیر در نظر می‌گیریم



که در آن ماتریس وزن‌ها بصورت  $W = [w1, w2, w3, -b]$  می‌باشد. همچنین برای P1 خروج  $y=0$  و برای P2 خروجی  $y=1$  و تابع فعال‌سازی را هم تابع پله در نظر می‌گیریم:

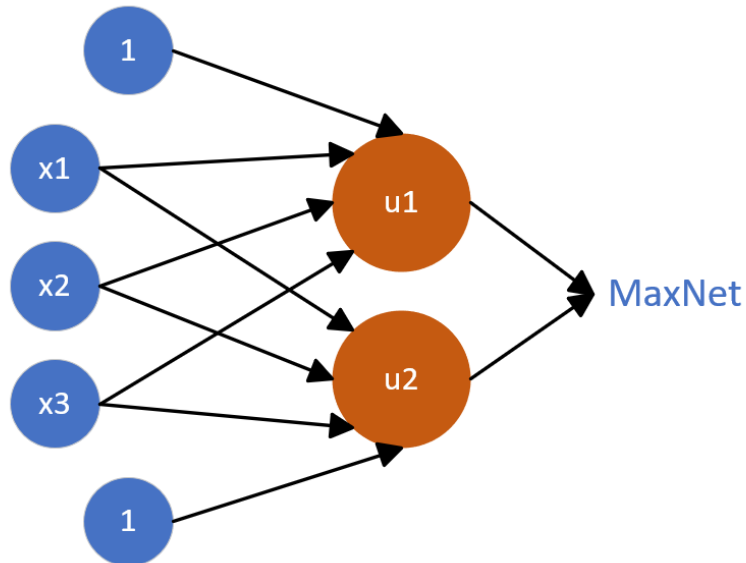
$$y = \begin{cases} 1 & \text{اگر } w1 * x1 + w2 * x2 + w3 * x3 - b > 0 \\ 0 & \text{اگر } w1 * x1 + w2 * x2 + w3 * x3 - b \leq 0 \end{cases}$$

برای P1 باید  $y=0$  شود بنابراین  $-w_1 - w_2 + w_3 - b \leq 0$

برای P2 باید  $y=1$  شود بنابراین  $w_1 - w_2 + w_3 - b > 0$

بنابراین میتوان مقادیر وزن ها و بایاس را به صورت  $W = [1, 0, 0, 0]$

ب) شبکه‌ی همینگ به صورت زیر است:



ابتدا وزن های اولیه و بایاس به شکل زیر محاسبه می شوند:

$$n = 3$$

$$m = 2$$

$$b_1 = b_2 = \frac{3}{2} = 1.5$$

$$W_{ij} = \frac{e_{ij}}{2} \rightarrow W_{ij} = \begin{bmatrix} 0.5 & -0.5 \\ -0.5 & -0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

در ادامه برای هر بردار مقدار خروجی نورون های  $u_1$  و  $u_2$  محاسبه می گردد. نورونی که خروجی بیشتری داشته

باشد، برای آن کلاس در نظر گرفته می شود. برای  $p_1$  داریم:

$$u_1 = 1.5 + [1 \quad -1 \quad 1] \begin{bmatrix} 0.5 \\ -0.5 \\ 0.5 \end{bmatrix} = 3$$

$$u_2 = 1.5 + [1 \quad -1 \quad 1] \begin{bmatrix} -0.5 \\ -0.5 \\ 0.5 \end{bmatrix} = 2$$



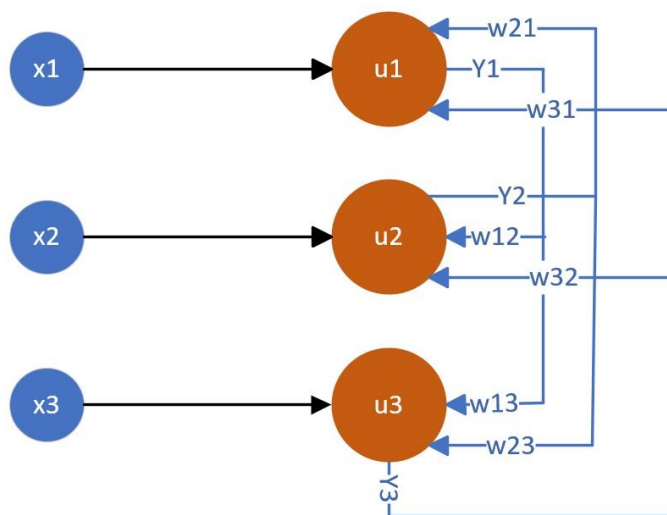
چون مقدار خروجی  $u1$  بزرگتر از  $u2$  است، پس هنگامی که نورون  $u1$  در شبکه فعال تر شود، یعنی مقدار  $p1$  وارد شبکه شده است. برای ورودی  $p2$  نیز محاسبات را برای هر دو نورون مشابه قبل انجام می دهیم:

$$1.5 + [-1 \quad -1 \quad 1] \begin{bmatrix} 0.5 \\ -0.5 \\ 0.5 \end{bmatrix} = 2$$

$$1.5 + [-1 \quad -1 \quad 1] \begin{bmatrix} -0.5 \\ -0.5 \\ 0.5 \end{bmatrix} = 3$$

این بار خروجی  $u2$  بزرگتر از  $u1$  است پس برای ورودی  $p2$ ، نورون  $u2$  فعال می شود.

ج) شبکه ی هاپفیلد به صورت زیر است:



وزن ها در شبکه به این صورت محاسبه می شود:

$$w_{ij} = \sum_{\mu=1}^P s_i^{(\mu)} s_j^{(\mu)} \rightarrow W = s^{(1)} s^{(1)T} + s^{(2)} s^{(2)T}$$

$$s^{(1)} s^{(1)T} = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} [-1 \quad -1 \quad 1] = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 1 & -1 \\ -1 & -1 & 1 \end{bmatrix}$$

$$s^{(2)} s^{(2)T} = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} [1 \quad -1 \quad 1] = \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix}$$

$$W = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 1 & -1 \\ -1 & -1 & 1 \end{bmatrix} + \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix} - \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -2 \\ 0 & -2 & 0 \end{bmatrix}$$

اکنون برای یک بردار دیگر مثلاً  $[1 \ 1 \ -1]$  به عنوان ورودی به شبکه می‌دهیم. اگر خروجی یکی از دو بردار پیش‌فرض بودن یعنی با آن دسته تعلق دارد در غیر این صورت این مرحله را آنقدر تکرار می‌کنیم تا در نهایت به یکی از آن دو بردار پیش‌فرض برسیم.

$$h_i = \sum_j w_{ij} x_j \quad S_i = \text{sign}(h_i)$$

$$\text{sign}(x) = \begin{cases} 1 & x > 0 \\ -1 & x \leq 0 \end{cases}$$

$$h_1 = 0 \rightarrow \text{sign}(0) = -1$$

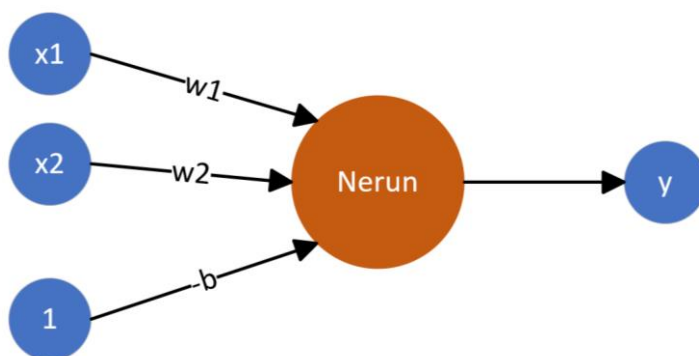
$$h_2 = -2 \times 1 = -2 \rightarrow \text{sign}(-2) = -1$$

$$h_3 = -2 \times -1 = 2 \rightarrow \text{sign}(2) = 1$$

مشابه  $p2$  شده است.

## ۷- سوال ۷:

الف) پرسپترون تک لایه به صورت زیر تعریف شده:



در پرسپترون عملیات زیر بر روی ورودی انجام می‌شود که در آن  $b$  بایاس و  $W = \begin{bmatrix} w1 \\ w2 \end{bmatrix}$  ماتریس وزن‌ها می‌باشد:

$$y = f(W^T P + b)$$

تابع فعال‌ساز نیز به صورت تابع پله تعریف شده است:

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

برای هر  $(P_i, t_i)$  مراحل زیر را انجام می دهیم:

۱- محاسبه مقدار پیش‌بینی شده توسط مدل:  $a_i = f(W^T P_i + b)$

۲- محاسبه خطا  $e_i = t_i - a_i$  (  $t_i$  مقدار واقعی و  $a_i$  مقدار محاسبه شده است)

۳- به‌روز نمودن مقادیر وزن‌ها و بایاس:

$$W = W + \eta e_i P_i$$

$$b = b + \eta e_i$$

این مراحل را تا زمانی که خطایی نداشته باشیم یا تعداد مراحل به حد آستانه برسد ادامه می‌دهیم.

ابتدا مقادیر اولیه وزن‌ها و بایاس را به صورت زیر تعریف می‌کنیم:

نرخ یادگیری  $\eta = 1$  و  $b = 0$  و  $W = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

**دور اول:**

برای نمونه اول داریم:

$$a_1 = f(W^T P_1 + b) = f\left(\begin{bmatrix} 0 & 0 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix} + 0\right) = f(0) = 1$$

$$e_1 = t_1 - a_1 = 1 - 1 = 0$$

چون خطا نداریم احتیاج به بروز کردن وزن‌ها و بایاس نمی‌باشد.

به همین ترتیب برای بقیه نمونه‌ها داریم:

$$a_2 = f(W^T P_2 + b) = f\left(\begin{bmatrix} 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} + 0\right) = f(0) = 1$$

$$e_2 = t_2 - a_2 = 1 - 1 = 0$$

$$a_3 = f(W^T P_3 + b) = f\left(\begin{bmatrix} 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} + 0\right) = f(0) = 1$$

$$e_3 = t_3 - a_3 = 1 - 1 = 0$$

$$a_4 = f(W^T P_4 + b) = f\left(\begin{bmatrix} 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 0\right) = f(0) = 1$$

$$e_4 = t_4 - a_4 = 0 - 1 = -1$$

در این نمونه خطا داریم و باید وزن‌ها و بایاس بروز شود.

$$W = W + \eta e_4 P_4 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + 1 \times (-1) \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

$$b = b + \eta e_4 = 0 + 1 \times (-1) = -1$$

$$a_5 = f(W^T P_5 + b) = f\left([-1 \ 0] \begin{bmatrix} 0 \\ 1 \end{bmatrix} - 1\right) = f(-1) = 0$$

$$e_5 = t_5 - a_5 = 0 - 0 = 0$$

$$a_6 = f(W^T P_6 + b) = f\left([-1 \ 0] \begin{bmatrix} 1 \\ 1 \end{bmatrix} - 1\right) = f(-2) = 0$$

$$e_6 = t_6 - a_6 = 0 - 0 = 0$$

**دور دوم:** حال با این وزن‌ها و بایاس جدید الگوریتم را دوباره روی نمونه‌ها اجرا می‌کنیم:

$$a_1 = f(W^T P_1 + b) = f\left([-1 \ 0] \begin{bmatrix} -1 \\ 1 \end{bmatrix} - 1\right) = f(1 - 1) = 1$$

$$e_1 = t_1 - a_1 = 1 - 1 = 0$$

$$a_2 = f(W^T P_2 + b) = f\left([-1 \ 0] \begin{bmatrix} 0 \\ 0 \end{bmatrix} - 1\right) = f(-1) = 0$$

$$e_2 = t_2 - a_2 = 1 - 0 = 1$$

در این نمونه خطا داریم و باید وزن‌ها و بایاس بروز شود:

$$W = W + \eta e_2 P_2 = \begin{bmatrix} -1 \\ 0 \end{bmatrix} + 1 \times (1) \times \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

$$b = b + \eta e_2 = -1 + 1 \times (1) = 0$$

$$a_3 = f(W^T P_3 + b) = f\left([-1 \ 0] \begin{bmatrix} 1 \\ -1 \end{bmatrix} + 0\right) = f(-1) = 0$$

$$e_3 = t_3 - a_3 = 1 - 0 = 1$$

در این نمونه خطا داریم و باید وزن‌ها و بایاس بروز شود:

$$W = W + \eta e_3 P_3 = \begin{bmatrix} -1 \\ 0 \end{bmatrix} + 1 \times (1) \times \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

$$b = b + \eta e_3 = 0 + 1 \times (1) = 1$$

$$a_4 = f(W^T P_4 + b) = f\left([0 \ -1] \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 1\right) = f(1) = 1$$

$$e_4 = t_4 - a_4 = 0 - 1 = -1$$

در این نمونه خطا داریم و باید وزن‌ها و بایاس بروز شود:

$$W = W + \eta e_4 P_4 = \begin{bmatrix} 0 \\ -1 \end{bmatrix} + 1 \times (-1) \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

$$b = b + \eta e_4 = 1 + 1 \times (-1) = 0$$

$$a_5 = f(W^T P_5 + b) = f\left(\begin{bmatrix} -1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} + 0\right) = f(-1) = 0$$

$$e_5 = t_5 - a_5 = 0 - 0 = 0$$

$$a_6 = f(W^T P_6 + b) = f\left(\begin{bmatrix} -1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + 0\right) = f(-2) = 0$$

$$e_6 = t_6 - a_6 = 0 - 0 = 0$$

دور سوم:

$$a_1 = f(W^T P_1 + b) = f\left(\begin{bmatrix} -1 & -1 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix} + 0\right) = f(1 - 1) = 1$$

$$e_1 = t_1 - a_1 = 1 - 1 = 0$$

$$a_2 = f(W^T P_2 + b) = f\left(\begin{bmatrix} -1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} + 0\right) = f(0) = 1$$

$$e_2 = t_2 - a_2 = 1 - 1 = 0$$

$$a_3 = f(W^T P_3 + b) = f\left(\begin{bmatrix} -1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} + 0\right) = f(0) = 1$$

$$e_3 = t_3 - a_3 = 1 - 1 = 0$$

$$a_4 = f(W^T P_4 + b) = f\left(\begin{bmatrix} -1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 0\right) = f(-1) = 0$$

$$e_4 = t_4 - a_4 = 1 - 1 = 0$$

$$a_5 = f(W^T P_5 + b) = f\left(\begin{bmatrix} -1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} + 0\right) = f(-1) = 0$$

$$e_5 = t_5 - a_5 = 0 - 0 = 0$$

$$a_6 = f(W^T P_6 + b) = f\left(\begin{bmatrix} -1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + 0\right) = f(-2) = 0$$

$$e_6 = t_6 - a_6 = 0 - 0 = 0$$

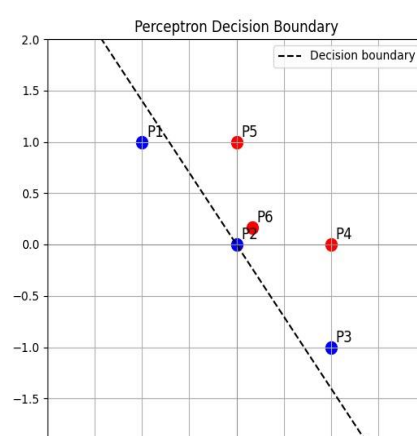
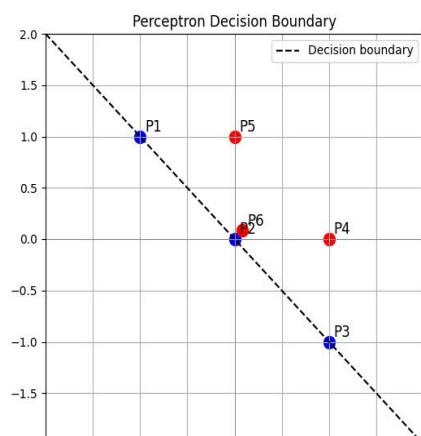
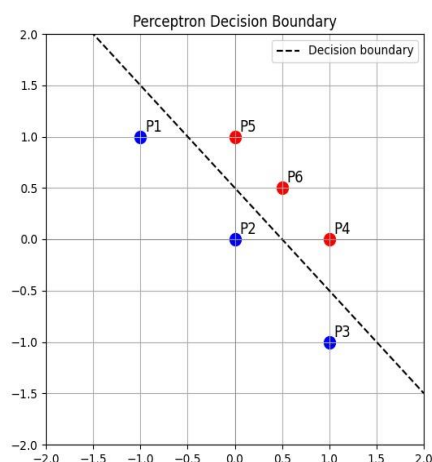
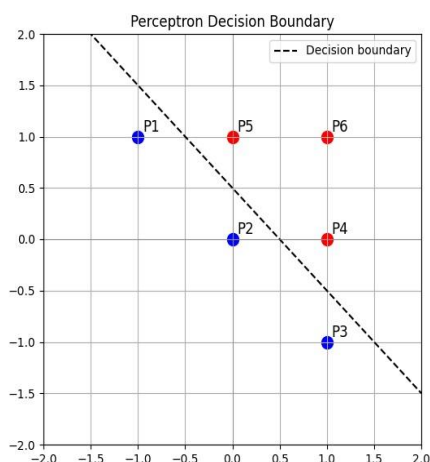
چون در دور سوم خطا نداشتیم و بروزرسانی انجام نشد، الگوریتم پایان می‌یابد. و وزن‌ها  $W = \begin{bmatrix} -1 & -1 \end{bmatrix}$

و بایاس  $b = 0$  می‌شود. معادله خط نیز برابر می‌شود با:  $-x_1 - x_2 = 0$

ب) برای  $\epsilon=2$  به همین ترتیب مقدار وزن‌ها برابر است با  $W=[-1 \ -1]$  و بایاس  $b=0.5$  می‌شود.

برای  $\epsilon=6$  به همین ترتیب مقدار وزن‌ها برابر است با  $W=[-7 \ -5]$  و بایاس  $b=0$  می‌شود.

برای  $\epsilon=12$  به همین ترتیب مقدار وزن‌ها برابر است با  $W=[-5.5 \ -5.5]$  و بایاس  $b=0$  می‌شود.



## ۸- سوال ۸:

(الف)

P1	P2	P3	P4	P5	P6

ب) با یک نرون نمی‌توان ۶ کلاس را جدا نمود ولی یک کلاس را می‌توان جدا کرد که با توجه به شکل کلاس P5 با بقیه متفاوت است بنابراین P5 را از بقیه جدا می‌کنیم:

$$\{P1, t1 = -1\}, \{P2, t2 = -1\}, \{P3, t3 = -1\}, \{P4, t4 = -1\}, \{P5, t5 = 1\}, \{P6, t6 = -1\}$$

در پرسپترون مقادیر را بصورت کلی در نظر می‌گیریم در حالت دو بعدی:

$$y = \text{sign}(W^T P + b)$$

که در آن وزن‌ها به صورت  $W = [w1 \ w2 \ w3 \ w4 \ w5 \ w6 \ w7 \ w8 \ w9]$  و بایاس  $b$  می‌باشد.

تابع فعال‌ساز تابع علامت است:

$$\text{sign}(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases}$$

برای هر  $\{P_i, t_i\}$  مشابه قبل مراحل زیر را انجام می‌دهیم

$$1- \text{ محاسبه مقدار پیش بینی توسط مدل } a_i = \text{sign}(W^T P_i + b)$$

$$2- \text{ محاسبه خطا } e_i = t_i - a_i \text{ ( } t_i \text{ مقدار واقعی و } a_i \text{ مقدار محاسبه شده است)}$$

$$3- \text{ به‌روز نمودن مقادیر وزن‌ها و بایاس}$$

$$W = W + \alpha e_i P_i$$

$$b = b + \alpha e_i$$

این مراحل را تا زمانی که خطایی نداشته باشیم یا تعداد مراحل به حد آستانه برسد ادامه می‌دهیم.

مقادیر اولیه به صورت زیر است:

$$W = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0] \quad b = 0 \quad \alpha = 1 \quad \text{و بایاس:} \quad \text{نرخ یادگیری:}$$

-برای نمونه اول داریم

$$a_1 = \text{sign}(W^T P_1 + b) = \text{sign} \left( [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0] \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \end{bmatrix} + 0 \right) = \text{sign}(0) = 0$$

$$e_1 = t_1 - a_1 = -1 - 0 = -1$$

خطا داریم پس وزن‌ها و بایاس اصلاح می‌شود

$$W = W + \alpha e_i P_i$$

$$b = b + \alpha e_i$$

$$W = W + \alpha e_i P_i = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0] + 1 \times (-1) \times [1 \ 1 \ 1 \ -1 \ -1 \ -1 \ -1 \ -1 \ -1] \\ = [-1 \ -1 \ -1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]$$

$$b = 0 + 1 \times (-1) = -1$$

برای نمونه دوم داریم:

$$a_2 = \text{sign}(W^T P_2 + b) = \text{sign} \left( [-1 \ -1 \ -1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1] \begin{bmatrix} -1 \\ -1 \\ -1 \\ 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1 \end{bmatrix} + 0 \right) = \text{sign}(3) = 1$$



$$e_2 = t_2 - a_2 = -1 - 1 = -2$$

خطا داریم پس وزن ها و بایاس اصلاح می شود

$$W = W + \alpha e_i P_i$$

$$b = b + \alpha e_i$$

$$W = W + \alpha e_i P_i = [-1 - 1 - 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1] + 1 \times (-1) \times [1 \ 1 \ 1 - 1 - 1 - 1 - 1 - 1 - 1] \\ = [-2 - 2 - 2 \ 2 \ 2 \ 2 \ 2 \ 2 \ 2]$$

$$b = -1 + 1 \times (-2) = -3$$

این مراحل را برای تمام نمونه ها انجام می دهیم و وزن ها و بایاس را به روز می کنیم. وزن های نهایی بدست آمده و بایاس نهایی برای تشخیص P5 از بقیه به شرح زیر است:

$$W = [1 \ 5 \ 1 \ 3 \ -1 \ 3 \ -1 \ 3 \ -1]$$

$$b = -3$$

## ۹- سوال ۹:

هر نورون یک مرز خطی به فضای دوبعدی اضافه می کند. بنابراین اگر  $n$  خط داشته باشیم، تعداد ناحیه - های بوجود آمده در صورت داشتن  $n$  خط مرزی، به صورت زیر محاسبه می شود:

$$R(n) = \frac{n(n+1)}{2} + 1$$

البته این در صورتی است که خطها با هم موازی نباشند و هیچ سه خطی یکدیگر را در یک نقطه قطع نکنند. بنابراین اگر فضا را بخواهیم به  $m$  ناحیه تقسیم کنیم، حداقل به این تعداد نورون نیاز داریم:

$$m = \frac{n(n+1)}{2} + 1 \rightarrow n(n+1) = 2m \rightarrow n = \frac{-1 + \sqrt{8m-7}}{2}$$