

تمرین کتبی شماره 5 هوش مصنوعی

بهینه سازی

بهینه سازی

سوال اول

$$h_{1input} = (i_1 \times w_1 + i_2 \times w_3) + b_1 = \overbrace{(0.05 \times 0.1)}^{0.005} + \overbrace{(0.15 \times 0.3)}^{0.045} + 0.25 = 0.3$$

(الف)

$$h_{2input} = (i_1 \times w_2 + i_2 \times w_4) + b_2 = (0.05 \times 0.2 + 0.15 \times 0.4) + 0.35 = 0.42$$

$$\Rightarrow \begin{cases} h_{1output} = \frac{1}{1 + e^{-0.3}} = 0.574 \end{cases}$$

$$h_{2output} = \frac{1}{1 + e^{-0.42}} = 0.603$$

$$o_{1input} = (h_1 \times w_5 + h_2 \times w_7) + b_3 = \overbrace{(0.574 \times 0.5)}^{0.287} + \overbrace{(0.603 \times 0.7)}^{0.422} + 0.49 = 1.159$$

$$o_{2input} = (h_1 \times w_6 + h_2 \times w_8) + b_4 = \overbrace{(0.574 \times 0.6)}^{0.344} + \overbrace{(0.603 \times 0.8)}^{0.482} + 0.55 = 1.376$$

$$\Rightarrow \begin{cases} o_{1output} = \frac{1}{1 + e^{-1.159}} = 0.761 \\ o_{2output} = \frac{1}{1 + e^{-1.376}} = 0.798 \end{cases} \rightarrow \text{جواب}$$

$$\rightarrow \frac{1}{2} \sum (target_i - o_i)^2 \rightarrow 0.436$$

(ب)

$$E_{total} = \frac{1}{2} ((0.1 - 0.761)^2 + (0.9 - 0.798)^2) = \boxed{0.223} \rightarrow \text{total loss}$$

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial o_{output}} \times \frac{\partial o_{output}}{\partial i_{input}} \times \frac{\partial i_{input}}{\partial w_5}$$

$$E_{total} = \frac{1}{2} (t_1 - o_{output})^2 + \frac{1}{2} (t_2 - o_{output})^2 \Rightarrow \frac{\partial E_{total}}{\partial o_{output}} = -(t_1 - o_{output})$$

$$= -(0.1 - 0.761) = 0.661$$

activation function

weight

$$\frac{\partial o_{output}}{\partial i_{input}} = \sigma'(o_{input}) = \sigma(o_{input}) [1 - \sigma(o_{input})] = (0.761)(0.239) = 0.181$$

$$\frac{\partial i_{input}}{\partial w_5} = \frac{\partial (h_1 w_5 + h_2 w_5 + b_3)}{\partial w_5} = h_1 = 0.574$$

$$\Rightarrow \frac{\partial E_{total}}{\partial w_5} = (0.661) \times (0.181) \times (0.574) = 0.068$$

learning rate

$$\Rightarrow w_5 = w_5 - \eta \cdot \frac{\partial E_{total}}{\partial w_5} = 0.5 - (0.5)(0.068) = 0.466$$

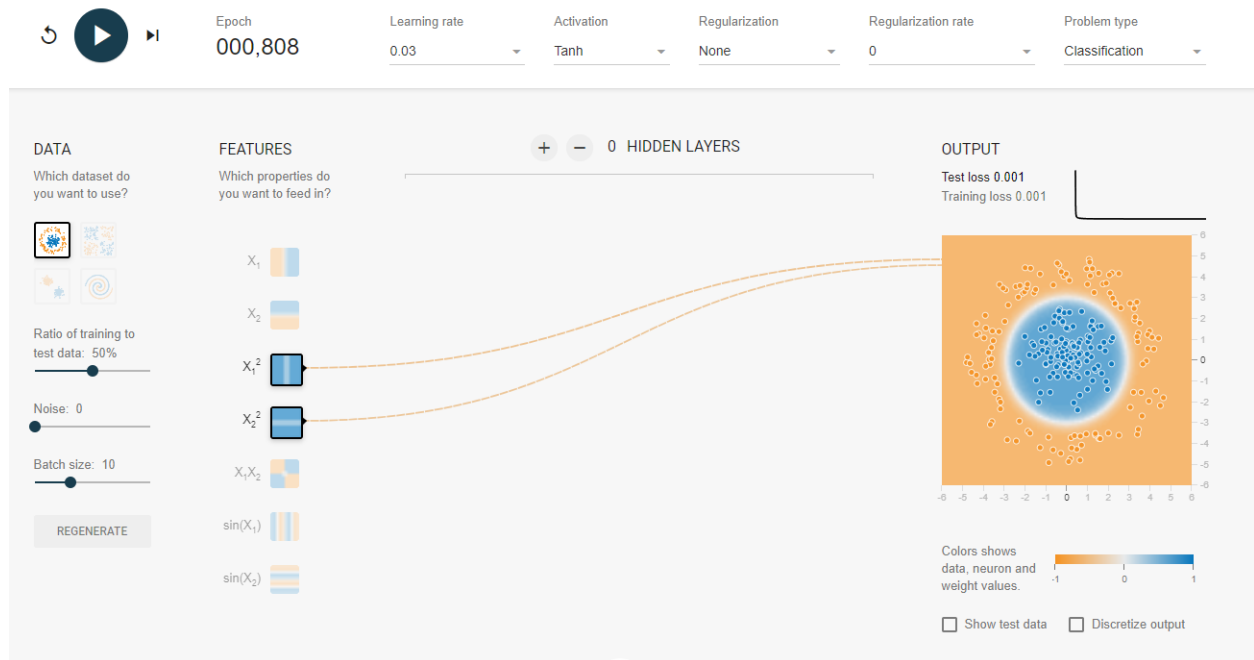
CS

Scanned with CamScanner

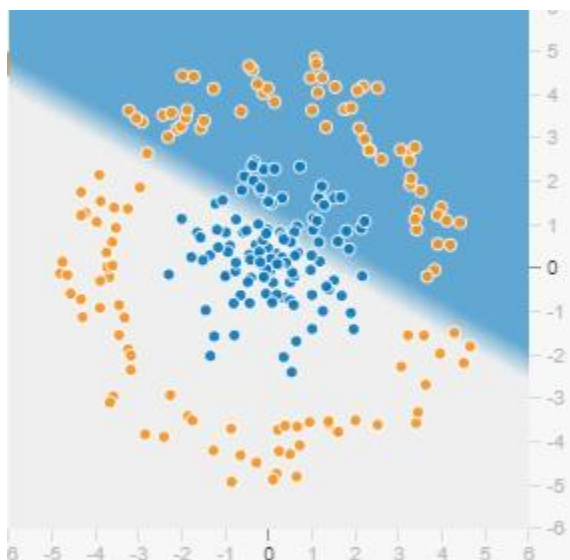
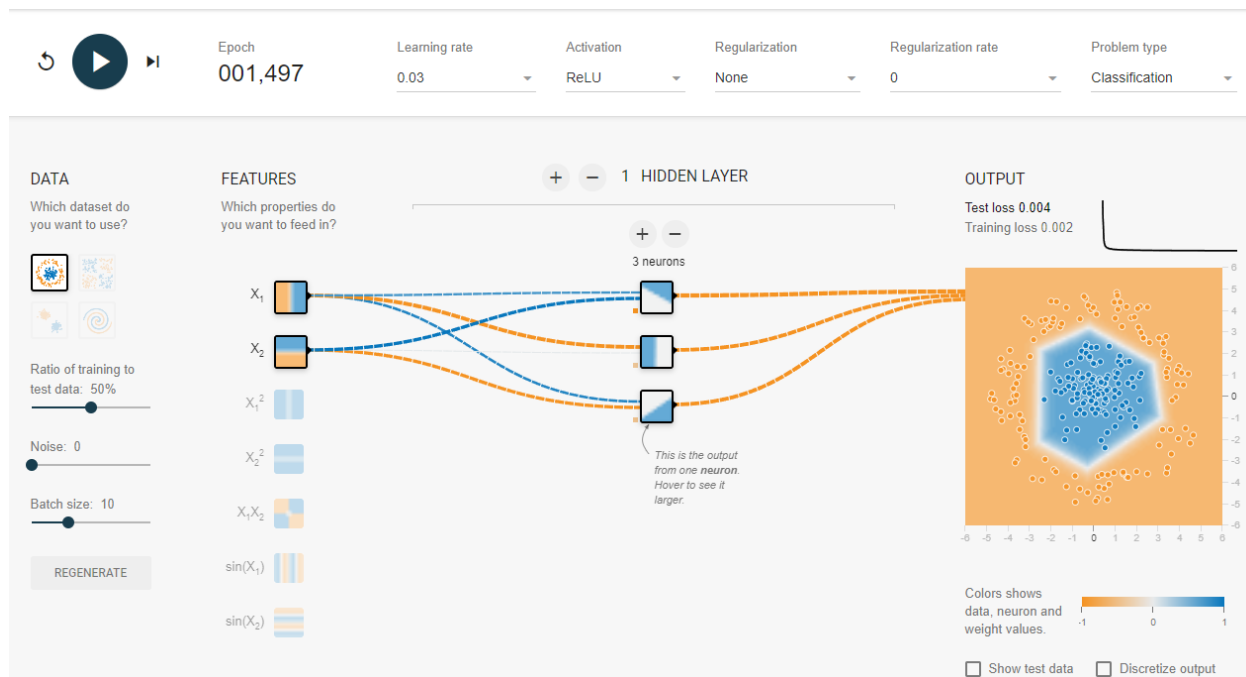
# شبکه های عصبی

## سوال اول

(الف)



(ب)



معادله خط روبرو با توجه به وزن های ورودی از  $X_1$  و  $X_2$  به

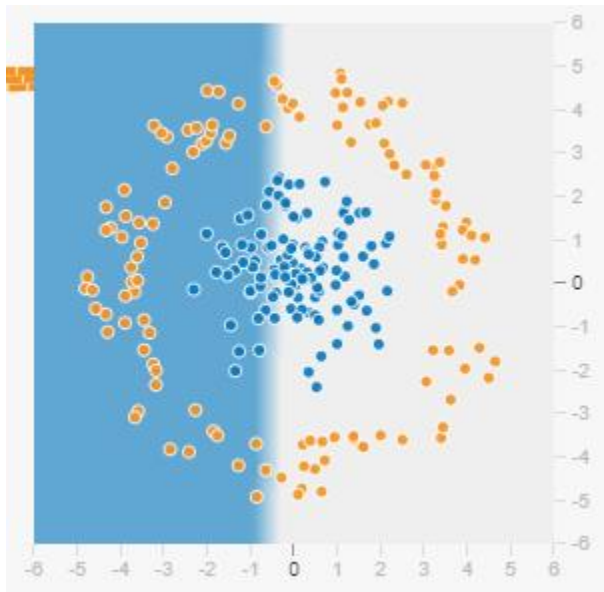
نورون مورد نظر و بایاس بدست آمده به صورت زیر است:

$$0.74X_1 + 1.3X_2 - 1 = 0$$

معادله خط روبرو با توجه به وزن های ورودی از  $X_1$  و  $X_2$  به

نورون مورد نظر و بایاس بدست آمده به صورت زیر است:

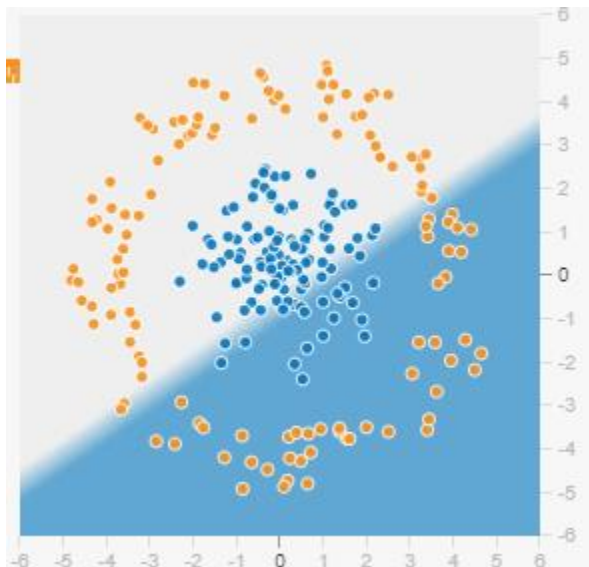
$$-1.6X_1 + 0.02X_2 - 0.41 = 0$$



معادله خط روبرو با توجه به وزن های ورودی از  $X_1$  و  $X_2$  به

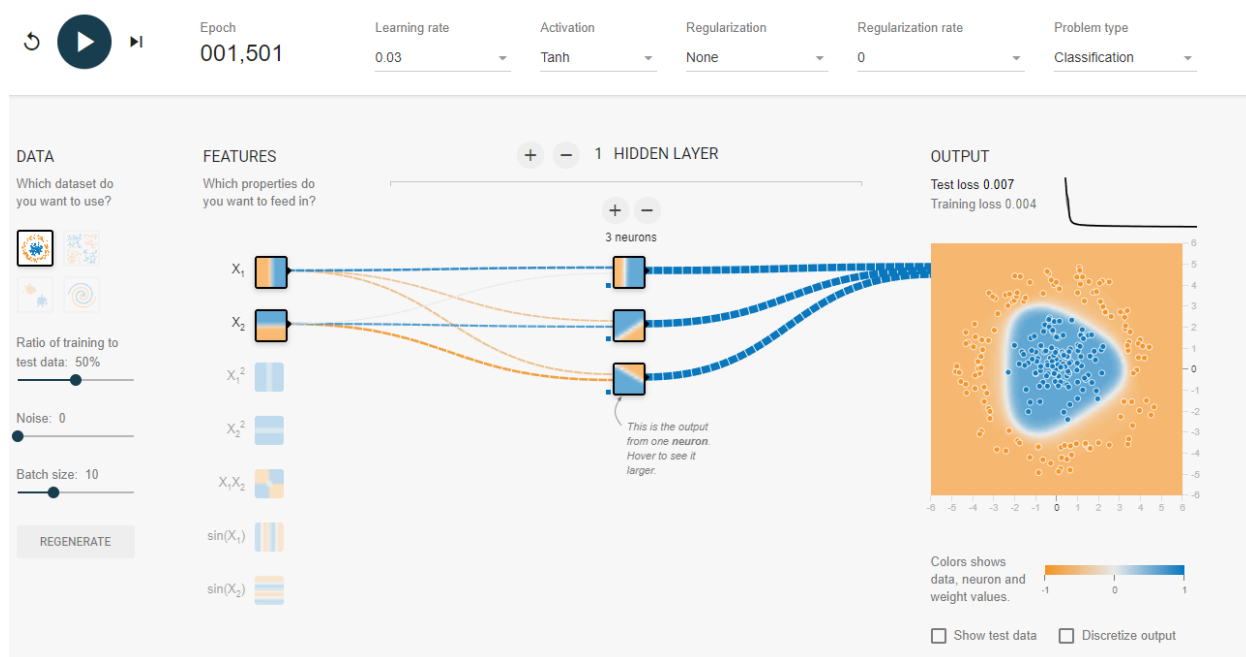
نورون مورد نظر و بایاس بدست آمده به صورت زیر است:

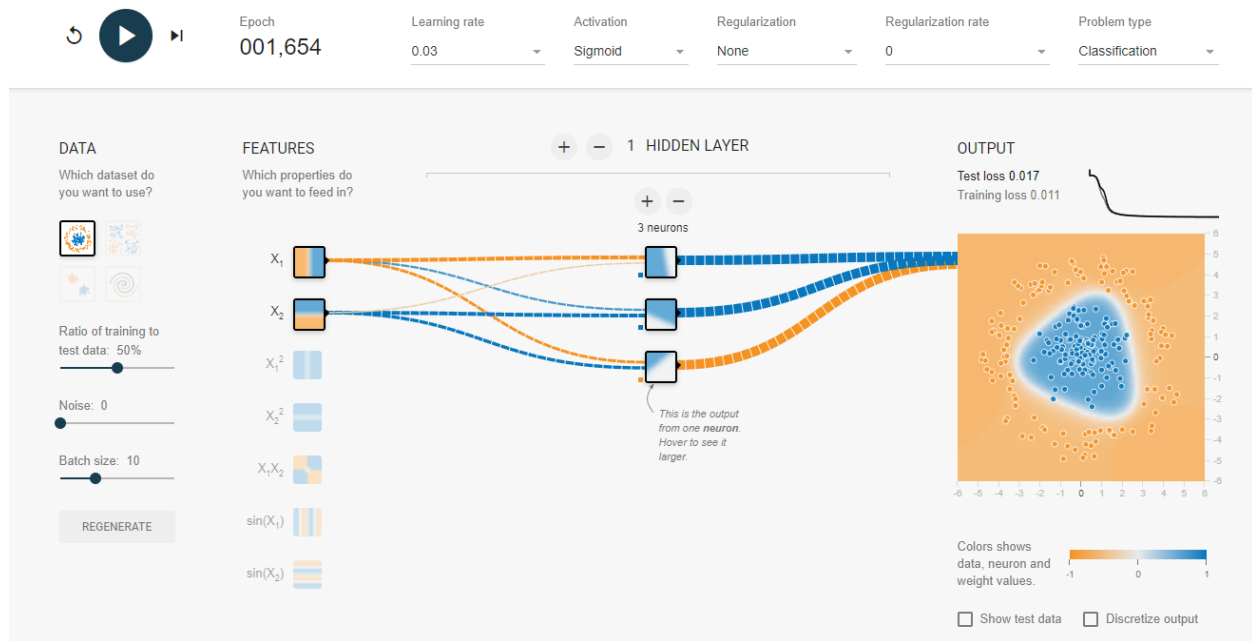
$$0.88X_1 - 1.3X_2 - 0.46 = 0$$



وزن های بدست آمده برای خروجی شبکه به ترتیب از بالا به پایین برابر 1.8- و 1.5- و 1.5- بدست آمده است(اولین شکل این قسمت که تصویر آن را گذاشته ام). از آنجایی که تابع فعال سازی ما Relu است، پس خروجی های لایه پنهان همواره اعدادی بزرگتر مساوی 0 هستند. همچنین از آنجایی که تمام وزن های منتهی به پاسخ منفی هستند، برای رسیدن به کلاس های مثبت، باید علامت بایاس حتما مثبت باشد تا امکان رسیدن به کلاس های مثبت هم فراهم شود. همچنین رفتار خطی Relu بعد از نقطه 0 باعث می شود که شبکه بتواند خطوط مرزی جدا کننده را بسازد.

(پ)





تفاوت تابع Relu با دو تابع دیگر مربوط به خطی بودن آن است. رفتار خطی ReLU بعد از صفر به آن اجازه می دهد تا پس از فعال سازی در هر نورون فقط وابستگی های خطی را مدل سازی کند(به دلیل وجود تنها یک لایه پنهان). این منجر به ایجاد مرزهایی با خطوط مستقیم و خطی می شود . اما tanh و sigmoid که ذاتا غیرخطی هستند ، می توانند مرزهای منحنی را حتی با یک لایه ایجاد کنند، که این باعث می شود که مرز های جدا کننده بتوانند به صورت منحنی وار باشند.



## شبکه های عصبی پیچشی

### شبکه های عصبی پیچشی

#### مسئله اول

(1) لایه های پیچشی از لحاظ تعداد بسیار بیشتر از لایه های کاملاً متصل هستند. در هر لایه پیچشی، یک فیلتور یکسان (در واقع وزن یکسان) به تمام قسمت های تصویر نسبت داده می شود و از آن عبور می کنند. این باعث کاهش شدید تعداد پارامترهای ورودی مادر لایه های کاملاً متصل می شود و نورون یک وزن وجود دارد. این یعنی تعداد بسیار زیادی پارامتر برای عکس ها با اندازه بزرگتر ایجاد می شود که بهینه کردن هر کدام از آنها از لحاظ زمانی و حافظه بسیار سنگین است و ممکن است باعث  $overfitting$  هم بشود.

(به دلیل تعداد زیاد پارامترها)

(2) لایه های پیچشی دارای خاصیت مستقل بودن از مکان عکس (تأثیر خوبی) هستند. یعنی آنچه به دلیل اینکه یک فیلتور از کل تصویر عبور می کند، شبکه می تواند الگوهای مورد نظر را مستقل از مکان قرارگیری آنها تشخیص دهد. یعنی یک ویژگی که در مکانی از تصویر توسط شبکه یاد گرفته شده می تواند در هر جای دیگر از تصویر هم توسط این شبکه شناسایی شود. اما لایه های کاملاً متصل این ویژگی را ندارند زیرا هر نورون وابسته به نورون های اطراف و فیلتورهای اطراف درون عکس است و این خاصیت



سوال دوم

برای کانولوشن، باید بردار فیلتر مورد نظر را معلوم کنیم و بامکت دادن این فیلتر روی  
رستخالی داده شده، ضرب داخلی را در هر مرحله انجام داده و مجهول ها را بدست آوریم:

$$f = [x, y, z] \rightarrow f^{-1} = \begin{bmatrix} z \\ y \\ x \end{bmatrix}$$

$$\begin{matrix} \textcircled{1} & \textcircled{2} & \textcircled{3} \\ [1, 4, 0] & [0, -2, 3] \end{matrix}$$

$$\begin{cases} \textcircled{1}: z + 4y + 0x = -2 \\ \textcircled{2}: 4z + 0y - 2x = -2 \\ \textcircled{3}: 0z - 2y + 3x = 11 \end{cases} \rightarrow \begin{cases} z + 4y = -2 \\ z + 4y = -2 \\ 2z - x = -1 \end{cases} \rightarrow \begin{cases} 8y + x = -3 \\ -2y + 3x = 11 \xrightarrow{\times 4} -8y + 12x = 44 \end{cases}$$

$$\rightarrow \begin{cases} x = \frac{41}{13} \approx 3.153 \\ y = -0.769 \\ z = 1.076 \end{cases}$$

$$\Rightarrow f = [3.153, -0.769, 1.076]$$