



به نام خدا

دانشگاه تهران

دانشکده علوم و فناوری های میان رشته ای

# Machine Learning

تمرین دوم

OMID MORADI	نام و نام خانوادگی
	شماره دانشجویی
	تاریخ ارسال گزارش

## سوال اول

(الف)

هارد مارجین: همه داده‌ها باید به درستی از هم جدا شوند، بدون هیچ خطایی. مناسب برای داده‌های کاملاً جداپذیر.

سافت مارجین: به برخی داده‌ها اجازه می‌دهد تا وارد ناحیه حاشیه شوند یا حتی در سمت نادرست مرز قرار بگیرند. مناسب برای داده‌های دارای نویز یا هم‌پوشانی، چون انعطاف‌پذیری بیشتری دارد و از بیش‌برازش جلوگیری می‌کند.

(ب)

به طور کلی وظیفه  $C$  کنترل جریمه طبقه بندی نادرست و تعادل بین حاشیه بزرگتر و نقاط اشتباه طبقه بندی شده است.

- اگر  $C$  بزرگ باشد:

مدل تمایل بیشتری به کاهش خطاها خواهد داشت، حتی اگر این موضوع به کاهش  $\text{margin}$  منجر شود.

ممکن است مدل بیش‌برازش ( $\text{overfit}$ ) کند.

- اگر  $C$  کوچک باشد:

مدل خطاهای بیشتری را می‌پذیرد و تمرکز بیشتری بر روی افزایش  $\text{margin}$  خواهد داشت.

ممکن است مدل کمتر برازش ( $\text{underfit}$ ) کند.

(پ)

کرنل ابزاری برای تبدیل داده‌ها به فضای ویژگی‌های بالاتر است تا داده‌های غیرخطی را خطی‌پذیر کند. دلایل استفاده:

1. امکان استفاده از SVM برای داده‌های غیرخطی.

2. عدم نیاز به محاسبه صریح تبدیل‌ها، به لطف کرنل تریک. ( $\text{kernel trick}$ )

### پیشنهاد کرنل برای داده‌های مختلف:

- شکل اول: کرنل **RBf (Radial Basis Function)** یا کرنل گوسی مناسب است، زیرا این کرنل به خوبی برای داده‌های با الگوی شعاعی کار می‌کند.
- شکل دوم: کرنل خطی (**Linear Kernel**) کافی است، زیرا داده‌ها ذاتاً خطی جداناپذیر هستند.
- شکل سوم: کرنل چندجمله‌ای (**Polynomial Kernel**) مناسب است.

### سوال دوم

الف)

$$x_{in} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$w_1 = \begin{bmatrix} 0 & 1 & 2 \\ 0 & -1 & 0 \end{bmatrix}$$

$$b = 1$$

$$z_1 = w_1^T x + b = \begin{bmatrix} 0 & 1 & 2 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 1 = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

$$z'_1 = G(z_1) = \begin{bmatrix} \frac{1}{1+e^{-1}} \\ \frac{1}{1+e^{-2}} \\ \frac{1}{1+e^{-3}} \end{bmatrix} = \begin{bmatrix} 0.73 \\ 0.88 \\ 0.95 \end{bmatrix}$$

خروجی:

$$z_{out} = w_2^T z'_1 + b = [1 \ 0 \ 1] \begin{bmatrix} 0.73 \\ 0.88 \\ 0.95 \end{bmatrix} + 1 = 2.68$$

$$z'_{out} = G(z_{out}) = \frac{1}{1+e^{-2.68}} = 0.936$$

Error:

$$MSE = \frac{1}{2} (y_{true} - z'_{out})^2 = \frac{1}{2} (1 - 0.936)^2 = 0.002$$

backpropagation:

$$\delta_{out} = \frac{\partial E}{\partial z_{out}} = (a_{out} - y_{true}) \cdot G'(z_{out}) \quad \leftarrow \text{وزن } w_2$$

$$G'(z_{out}) = a_{out} (1 - a_{out})$$

$$\delta_{out} = (a_{out} - y_{true}) \cdot a_{out} (1 - a_{out})$$

$$= (0.936 - 1) \cdot 0.936 (1 - 0.936) = -0.0038$$

بهتر ساز وزن در لایه مخفی:

$$w_2 = w_2 - \eta \cdot \delta_{out} \cdot a_{hidden} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} - 0.5 (-0.0038) \begin{bmatrix} 0.73 \\ 0.88 \\ 0.95 \end{bmatrix}$$

$$= \begin{bmatrix} 1.0014 \\ 0.0017 \\ 1.0018 \end{bmatrix}$$

بازرسی تغییرات:

$$b_{out} = b_{out} - \eta \cdot \delta_{out} = 1 - 0.5 (-0.0038) = 1.0019$$

در لایه مخفی:

(نصب وزن)

$$\delta_{hidden_1} = \delta_{out} \cdot w_2 \cdot a_{hidden} (1 - a_{hidden})$$

$$= -0.0038 \times 1 \times 0.73 (1 - 0.73) = -0.00075$$

(نودون):

$$\delta_{hidden_2} = -0.0038 \times 0.88 (1 - 0.88) = 0$$

(نودون):

$$\delta_{hidden_1} = -0.0038 \times 1 \times 0.99 (1 - 0.99) = -0.00017$$

$$\delta_{hidden} = \begin{bmatrix} -0.00075 \\ 0 \\ -0.00017 \end{bmatrix}$$

آپوسترون:

$$w_1 = w_1 - \eta \delta_{hidden} \cdot x^T = \begin{bmatrix} 0 & 1 & 2 \\ 0 & -1 & 0 \end{bmatrix} - 0.5 \begin{bmatrix} -0.00075 \\ 0 \\ -0.00017 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0.00037 & 1 & 2 \\ 0 & -1 & 0 \end{bmatrix}$$

بایاس:

$$b_{hidden} = b_{hidden} - \eta \delta_{hidden} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - 0.5 \begin{bmatrix} -0.00075 \\ 0 \\ -0.00017 \end{bmatrix} = \begin{bmatrix} 1.00037 \\ 1 \\ 0.000085 \end{bmatrix}$$

Scanned with CamScanner

(ب)

1. غیرخطی کردن خروجی‌ها
2. ایجاد مرزهای تصمیم‌گیری پیچیده
3. افزایش توانایی شبکه در یادگیری الگوهای پیچیده
4. کنترل و محدود کردن خروجی‌ها
5. مدل‌سازی مفاهیم غیرخطی در داده‌ها

(پ)

### 1. محو شدن گرادیان

یکی از مشکلات سیگموئید، پدیده محو شدن گرادیان است. در این حالت، اگر ورودی بسیار بزرگ یا بسیار کوچک باشد، مشتق سیگموئید تقریباً به صفر نزدیک می‌شود. این موضوع در شبکه‌های عمیق باعث می‌شود گرادیان در طی پس‌انتشار کاهش یابد و لایه‌های ابتدایی نتوانند به خوبی به‌روزرسانی شوند. برای

رفع این مشکل، استفاده از ReLU پیشنهاد می‌شود، زیرا در ناحیه مثبت، مشتق آن برابر با 1 است و از محو شدن گرادیان جلوگیری می‌کند. همچنین نرمال‌سازی ورودی‌ها (مثلاً استاندارد کردن داده‌ها) می‌تواند از ورود مقادیر بیش از حد بزرگ یا کوچک به تابع سیگموئید جلوگیری کند.

## 2. خروجی‌های غیر صفر مرکزی (Non-Zero Centered Outputs)

مشکل دیگر سیگموئید این است که خروجی‌های آن **غیر صفر مرکزی** هستند، یعنی مقادیر خروجی همیشه در بازه  $[0,1]$  قرار دارند. این موضوع باعث می‌شود گرادیان‌ها همیشه یا مثبت یا صفر باشند، که الگوریتم گرادیان نزولی را کند می‌کند. برای حل این مشکل، می‌توان از تابع تانژانت هایپربولیک (Tanh) استفاده کرد که خروجی آن در بازه  $[-1,1]$  قرار دارد و گرادیان‌ها را متقارن‌تر می‌کند. استفاده از روش‌های بهینه‌سازی پیشرفته مانند Adam نیز می‌تواند تأثیر این مشکل را کاهش دهد.

### برتری RELU نسبت به Sigmoid:

ReLU نسبت به سیگموئید مزایای زیادی دارد. این تابع محاسبات بسیار ساده‌تری دارد، زیرا فقط مقدار حداکثر  $\max(0,x)$  را محاسبه می‌کند، در حالی که سیگموئید به محاسبات نمایی نیاز دارد. علاوه بر این، ReLU به دلیل صفر کردن خروجی برای مقادیر منفی، یک خروجی پراکنده تولید می‌کند که باعث ساده‌تر شدن ساختار شبکه می‌شود. همچنین، در شبکه‌های عمیق، ReLU عملکرد بهتری دارد زیرا از محو شدن گرادیان جلوگیری کرده و سرعت یادگیری را افزایش می‌دهد.

بنابراین، ReLU به دلیل سادگی، کارایی بالا و توانایی حل مشکلاتی که سیگموئید ایجاد می‌کند، انتخاب بهتری برای شبکه‌های عمیق است.

(ت)

**در مسیر پیشرو (Forward Pass)**، تابع فعال‌ساز ReLU به صورت  $f(x)=\max(0,x)$  عمل می‌کند. اگر ورودی ( $x$ ) مثبت باشد، خروجی همان مقدار ورودی است، و اگر ورودی منفی یا صفر باشد، خروجی صفر می‌شود. این رفتار باعث می‌شود که مقادیر مثبت بدون تغییر به لایه بعدی منتقل شوند و مقادیر منفی به صفر تبدیل شوند. بنابراین، برخی از نورون‌ها خاموش شده و در محاسبات بعدی تأثیر ندارند.

**در مسیر پس‌رو (Backward Pass)**، مشتق تابع ReLU نسبت به ورودی محاسبه می‌شود.

برای ورودی‌های مثبت ( $x>0$ )، مشتق برابر با 1 است، که به این معنی است که گرادیان به طور کامل به لایه‌های قبلی منتقل می‌شود و یادگیری ادامه پیدا می‌کند. اما برای ورودی‌های غیر مثبت ( $x\leq 0$ )، مشتق

برابر صفر است، و گرادیانی به وزن‌های مرتبط با این نورون‌ها منتقل نمی‌شود. این باعث می‌شود که این نورون‌ها به اصطلاح "خاموش" شوند و دیگر یاد نگیرند.

این رفتار مزایایی دارد، مانند جلوگیری از محو شدن گرادیان و تسریع یادگیری، زیرا مشتق ReLU برای مقادیر مثبت ثابت است. اما مشکل اصلی این است که نورون‌هایی که ورودی غیرمثبت دریافت می‌کنند ممکن است برای همیشه خاموش باقی بمانند (مشکل نورون‌های مرده). برای رفع این مشکل می‌توان از نسخه‌های اصلاح‌شده ReLU مانند Leaky ReLU استفاده کرد، که برای مقادیر منفی یک گرادیان کوچک (مثلاً  $0.01 \times$ ) در نظر می‌گیرد. این کار باعث می‌شود نورون‌های خاموش همچنان امکان یادگیری داشته باشند. گزینه دیگر ReLU6 است که خروجی ReLU را در بازه  $[0, 6]$  محدود می‌کند و کنترل بیشتری روی گرادیان فراهم می‌کند.

به طور کلی، ReLU در مسیر پیشرو و پس‌رو برای مقادیر مثبت بسیار مؤثر عمل می‌کند و در شبکه‌های عمیق به دلیل جلوگیری از محو شدن گرادیان محبوب است.

## سوال سوم

(ب)

### بررسی مقادیر گمشده: (Missing Values)

- **دلیل:** مقادیر گمشده می‌توانند باعث خطا در مدل‌سازی شوند. روش‌های مختلفی مانند حذف ردیف‌ها، پر کردن مقادیر با میانگین، میانه، یا مدل‌سازی آماری وجود دارند.

### حذف یا مدیریت مقادیر پرت: (Outliers)

- **دلیل:** مقادیر پرت می‌توانند تأثیر زیادی بر الگوریتم‌های حساس به مقیاس داشته باشند. حذف یا تعدیل این مقادیر باعث می‌شود مدل پایدارتر شود.

### مقیاس‌بندی داده‌ها: (Scaling)

- **دلیل:** در الگوریتم‌هایی که از فاصله برای محاسبات استفاده می‌کنند (مانند رگرسیون یا الگوریتم‌های مبتنی بر درخت)، مقیاس متغیرها اهمیت زیادی دارد.

### رمزگذاری متغیرهای دسته‌ای: (Encoding)

- **دلیل:** داده‌های دسته‌ای باید به فرم عددی تبدیل شوند تا الگوریتم‌های یادگیری ماشین بتوانند با آن‌ها کار کنند.

### حذف یا ترکیب ویژگی‌های غیرمفید: (Feature Selection/Engineering)

- **دلیل:** برخی ویژگی‌ها ممکن است همبستگی بالایی با یکدیگر داشته باشند یا اطلاعات کمی برای مدل‌سازی ارائه دهند.

### تبدیل داده‌های نامتقارن یا غیرخطی:

- **دلیل:** داده‌های نامتقارن می‌توانند نتایج مدل را تحت تأثیر قرار دهند. استفاده از تبدیلات لگاریتمی یا جذر می‌تواند توزیع داده‌ها را متعادل کند.

(پ)

(1)

- **Grid Search:** جستجوی سازمان‌یافته روی تمام ترکیب‌های ممکن از مقادیر پارامترهای مشخص شده. دقیق است اما زمان‌بر.
- **Random Search:** نمونه‌برداری تصادفی از ترکیبات پارامترها در فضای جستجو. سریع‌تر است و برای فضاهای بزرگ‌تر مؤثرتر.

(2)

- **Linear Kernel:** برای داده‌های خطی؛ پارامتر مهم: نیاز به پارامتر اضافی خاصی ندارد.

- **RBF Kernel:** مناسب برای داده‌های غیرخطی؛ پارامترها:

○ **Gamma:** تأثیرگذاری نقاط دور و نزدیک.

○ **C:** جریمه خطای دسته‌بندی.

- **Polynomial Kernel:** برای مسائل با روابط چندجمله‌ای؛ پارامترها:

○ **degree:** درجه چندجمله‌ای.

○ **C:** میزان تحمل خطا.

(3)



- **One-vs-Rest (OvR)**: هر کلاس را در مقابل سایر کلاس‌ها مقایسه می‌کند. مزیت: ساده‌تر و سریع‌تر برای تعداد کلاس زیاد.

- **One-vs-One (OvO)**: تمام جفت کلاس‌ها را مقایسه می‌کند. مزیت: معمولاً برای داده‌های کوچک‌تر بهتر عمل می‌کند.

**در این مسئله:** اگر دسته‌بندی چندکلاسی باشد، بسته به تعداد کلاس‌ها یکی از این روش‌ها لازم است. OvR برای تعداد کلاس‌های زیاد مناسب‌تر است. در نتیجه برای این مثال OvR مناسب نیست.

(ث)

مدل SVM با استفاده از روش GridSearch برای بهینه‌سازی پارامترها به نتایج قابل توجهی دست یافت. بهترین پارامترهای به‌دست‌آمده شامل C برابر با 1، degree برابر با 2، gamma مقدار "scale" و کرنل "rbf" هستند. امتیاز اعتبارسنجی متقاطع (Cross-Validation) مدل 77.89 درصد گزارش شده است. دقت مدل روی داده‌های تست برابر با 82.23 درصد است که عملکرد مطلوب آن را نشان می‌دهد.

ماتریس درهم‌ریختگی نشان می‌دهد که مدل توانسته است 56.58 درصد از نمونه‌های منفی واقعی را به‌درستی منفی پیش‌بینی کند و 25.66 درصد از نمونه‌های مثبت واقعی را نیز به‌درستی به‌عنوان مثبت شناسایی کند. خطاهای False Positive و False Negative به ترتیب برابر با 8.55 درصد و 9.21 درصد هستند. این مقادیر نشان می‌دهند که مدل در ایجاد تعادل بین پیش‌بینی‌های مثبت و منفی عملکرد مناسبی داشته است.

به‌طور کلی، مدل عملکرد خوبی روی داده‌ها ارائه داده است و برای بهبود بیشتر می‌توان روش‌های پیش‌پردازش داده را تقویت کرد یا از داده‌های آموزشی بیشتری استفاده کرد.

## سوال چهارم

(ب)

### پاک‌سازی داده‌ها (Data Cleaning)

- **حذف یا جایگزینی مقادیر گمشده:** داده‌های ناقص می‌توانند عملکرد مدل را تحت تأثیر قرار دهند. روش‌های معمول شامل حذف رکوردهای ناقص یا جایگزینی مقادیر گمشده با میانگین، میانه یا مقادیر پیش‌بینی‌شده هستند.

- **حذف داده‌های تکراری:** وجود داده‌های تکراری می‌تواند منجر به انحراف نتایج شود؛ بنابراین، شناسایی و حذف این داده‌ها ضروری است.
- **اصلاح خطاهای ورودی:** برخی داده‌ها ممکن است دارای خطاهای تایپی یا مقادیر نامعتبر باشند که باید شناسایی و تصحیح شوند.

### یکسان‌سازی داده‌ها (Data Standardization)

- **تبدیل مقیاس‌ها:** ویژگی‌های مختلف ممکن است دارای مقیاس‌های متفاوتی باشند. برای جلوگیری از تأثیر نامتناسب ویژگی‌ها بر مدل، از روش‌هایی مانند استانداردسازی (Standardization) یا نرمال‌سازی (Normalization) استفاده می‌شود.
- **رمزگذاری داده‌های دسته‌ای (Categorical Data):** داده‌های دسته‌ای باید به فرم عددی تبدیل شوند. روش‌های معمول شامل رمزگذاری برچسبی (Label Encoding) و رمزگذاری یک-هاتی (One-Hot Encoding) هستند.

### کاهش ابعاد (Dimensionality Reduction)

- **انتخاب ویژگی (Feature Selection):** انتخاب ویژگی‌های مهم و حذف ویژگی‌های غیرضروری می‌تواند به بهبود عملکرد مدل کمک کند.
- **استخراج ویژگی (Feature Extraction):** تبدیل ویژگی‌های موجود به ویژگی‌های جدید و مفید با استفاده از تکنیک‌هایی مانند تحلیل مؤلفه‌های اصلی (PCA).

### نمونه‌سازی داده‌ها (Data Sampling)

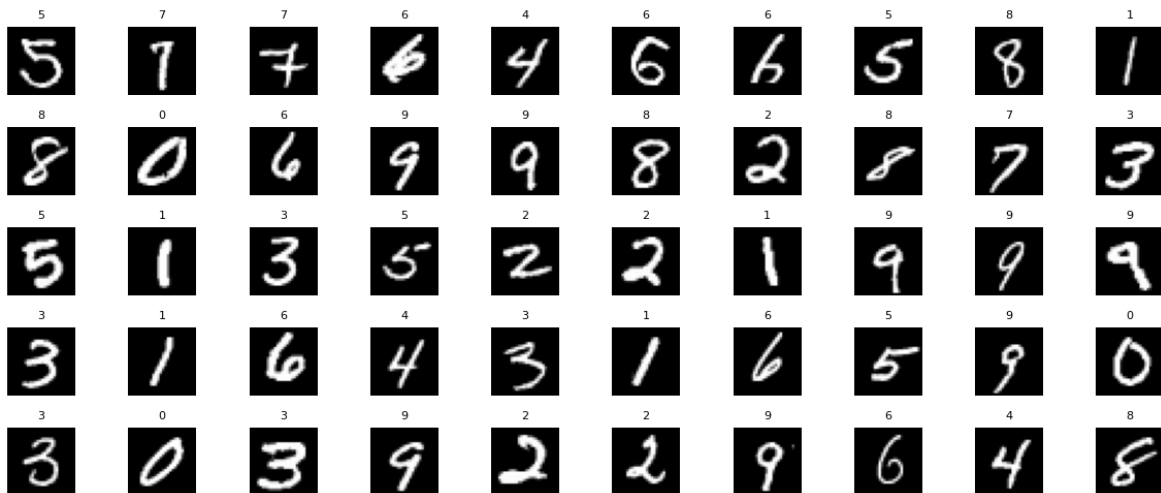
- **تقسیم داده‌ها:** تقسیم داده‌ها به مجموعه‌های آموزشی و آزمایشی برای ارزیابی عملکرد مدل ضروری است.
- **تعادل داده‌ها:** در صورت عدم توازن در کلاس‌ها، از روش‌هایی مانند نمونه‌برداری بیش از حد (Oversampling) یا نمونه‌برداری کمتر از حد (Undersampling) استفاده می‌شود.

### حذف نویز (Noise Removal)

- **فیلتر کردن داده‌ها:** حذف داده‌های پرت یا نویزی که می‌توانند بر عملکرد مدل تأثیر منفی بگذارند.

## تبدیل داده‌ها (Data Transformation)

- مقیاس‌بندی ویژگی‌ها: تبدیل ویژگی‌ها به مقیاس‌های مشابه برای بهبود عملکرد مدل.
- ایجاد ویژگی‌های جدید: با ترکیب یا تغییر ویژگی‌های موجود، ویژگی‌های جدیدی ایجاد می‌شود که می‌تواند به مدل کمک کند.



(پ)

مشکلات عدم استفاده از مجموعه اعتبارسنجی:

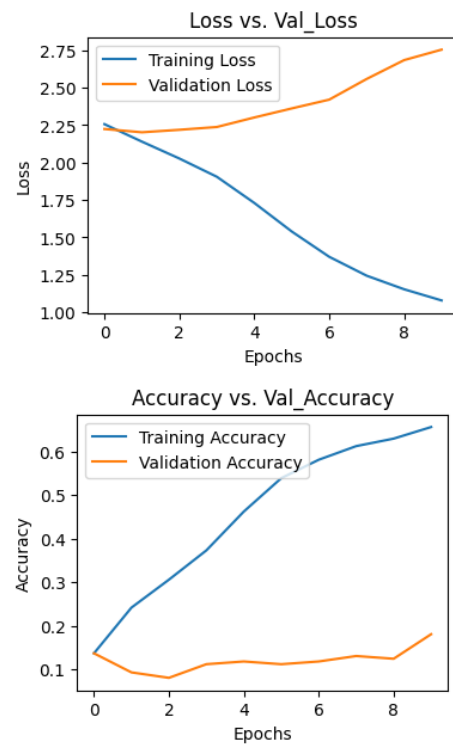
### 1. عدم تنظیم مناسب هایپرپارامترها:

- بدون مجموعه اعتبارسنجی، نمی‌توان پارامترهایی مانند تعداد اپوک‌ها یا نرخ یادگیری را بهینه کرد.
- ممکن است مدل روی داده‌های آموزش بیش‌برازش کند و روی داده‌های جدید عملکرد ضعیفی داشته باشد.

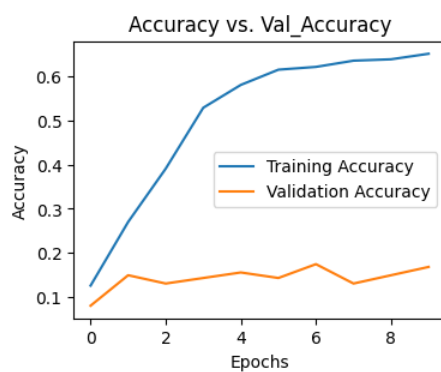
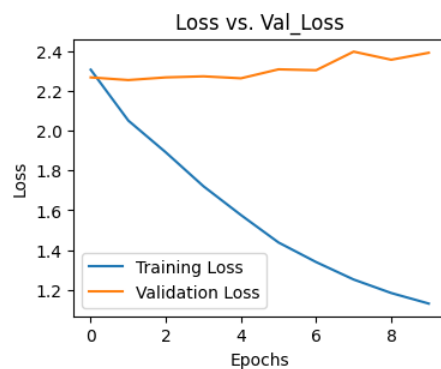
### 2. استفاده از داده آزمون برای تنظیم مدل:

- اگر داده آزمون به جای اعتبارسنجی برای تنظیم پارامترها استفاده شود، مدل ممکن است به داده آزمون نیز بیش‌برازش کند.
- این کار باعث می‌شود ارزیابی مدل بر روی داده‌های واقعی غیرقابل اعتماد باشد.

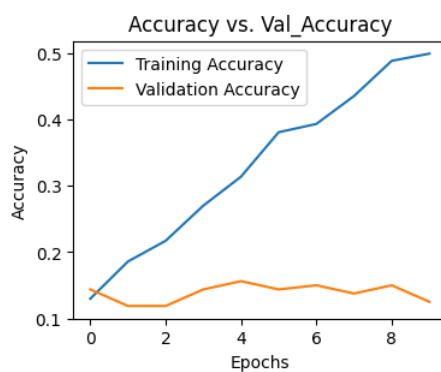
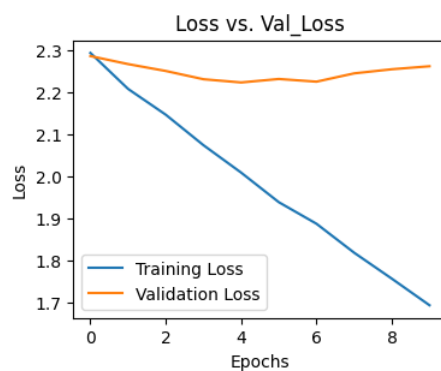
(ت)



شکل 1-دو لایه (64 و 128)



شکل 2- یک لایه (128 نورون)



شکل 3- یک لایه (16 نورون)

پیکربندی    Training Loss    Validation Loss    Training Accuracy    Validation Accuracy

پایین	متوسط	نوسانی/افزایشی	بالا	یک لایه (16 نرون)
اندکی بهتر از حالت اول	بالاتر از حالت اول	نوسانی	کمتر از حالت اول	یک لایه (128 نرون)
بهترین	بهترین	کمترین نوسان	کمترین	دو لایه (64 و 128)

(د)

### بیش‌برازش: (Overfitting)

زمانی رخ می‌دهد که مدل به‌طور دقیق داده‌های آموزشی را یاد می‌گیرد اما در تعمیم‌دهی به داده‌های جدید و نادیده (مانند داده‌های آزمون) عملکرد ضعیفی دارد. این معمولاً زمانی رخ می‌دهد که مدل بیش از حد پیچیده باشد یا داده‌های آموزشی ناکافی باشند.

**نشانه‌ها:**

- دقت زیاد روی مجموعه آموزش، اما دقت پایین روی مجموعه اعتبارسنجی یا آزمون.
- افزایش مقدار Validation Loss با کاهش مقدار Training Loss.

#### • راهکارها برای پیشگیری و حل:

#### 1. رگولاراسیون: (Regularization)

- اضافه کردن جریمه‌هایی به تابع هزینه، مانند L1 یا L2.

#### 2. Dropout:

- به صورت تصادفی غیرفعال کردن درصدی از نرون‌ها در طول آموزش برای کاهش پیچیدگی مدل.

### کم‌برازش: (Underfitting)

زمانی رخ می‌دهد که مدل حتی روی داده‌های آموزشی نیز عملکرد ضعیفی دارد و نتوانسته الگوهای موجود در داده‌ها را به درستی یاد بگیرد. این معمولاً زمانی رخ می‌دهد که مدل بیش از حد ساده باشد یا زمان کافی برای آموزش نداشته باشد.

#### **نشانه‌ها:**

- دقت پایین روی مجموعه آموزش.
- عدم کاهش مقدار Training Loss حتی پس از تعداد زیادی اپوک.

#### • راهکارها برای پیشگیری و حل:

1. افزایش پیچیدگی مدل:

▪ اضافه کردن تعداد نورون‌ها یا لایه‌ها.

2. کاهش نرخ یادگیری: (Learning Rate)

▪ کاهش نرخ یادگیری برای بهبود جزئیات یادگیری مدل.

تحلیل نتایج مراحل قبل با این مفاهیم:

بیش برآزش برای دو حالت با 128 نورون و دولایه

کم برآزش برای حالت با 16 نورون

### سوال پنجم

(الف)

برای یافتن مرز تصمیم‌گیری باید نقاطی که نتایج یکسان را می‌دهند را پیدا کنیم:

class 1 = class 2 :

$$x - y + 1 = -x - y + 2 \rightarrow x = \frac{1}{2}$$

class 1 = class 3 :

$$x - y + 1 = -3 + y - x \rightarrow x - y = -2$$

class 1 = class 4 :

$$x - y + 1 = x + y \rightarrow y = \frac{1}{2}$$

class 2 = class 3 :

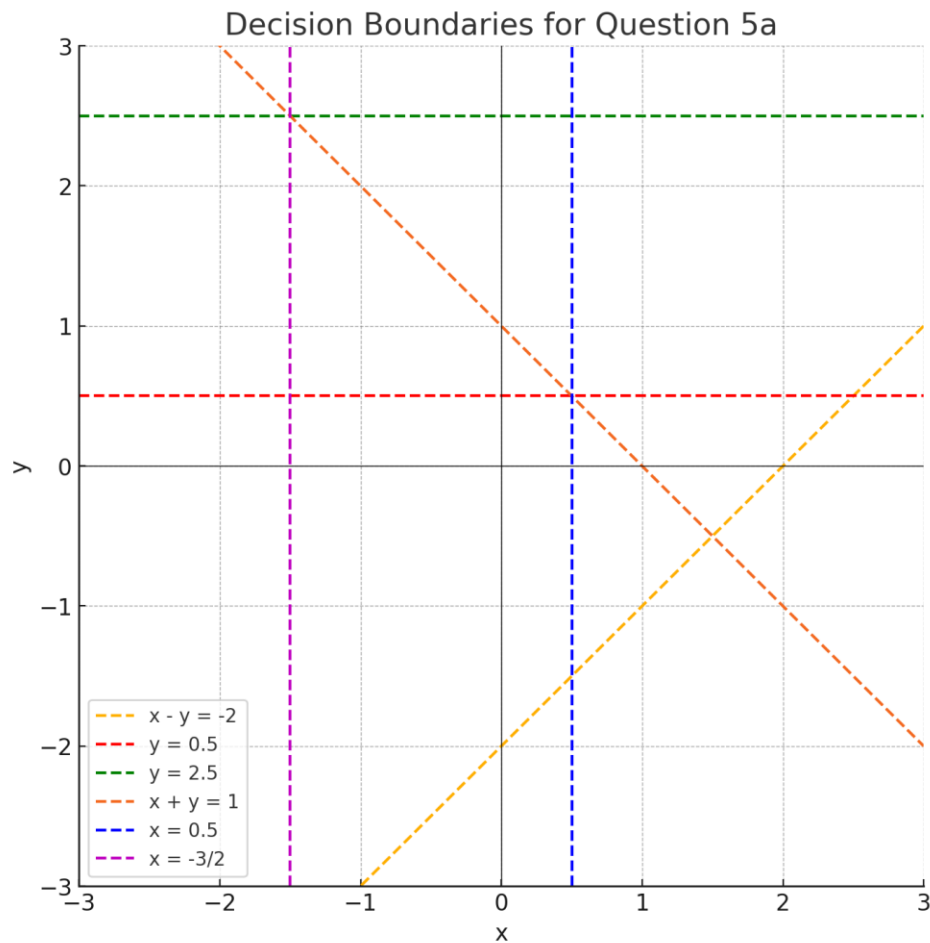
$$-x - y + 2 = -x + y - 3 \rightarrow y = \frac{5}{2}$$

class 2 = class 4 :

$$-x - y + 2 = x + y \rightarrow 1 + y = 1$$

class 3 = class 4 :

$$-x + y - 3 = x + y \rightarrow x = \frac{3}{2}$$



(ب)

- اگر داده‌ها به صورت خطی جداسدنی باشند، یک پرسپترون با دو گره ورودی کفایت می‌کند.
- اگر داده‌ها به صورت غیرخطی جداسدنی باشند، نیاز به افزودن ویژگی‌هایی است تا داده‌ها در فضای ویژگی جدید جداسدنی شوند.

(پ)

(1)

برای حل این بخش، باید شبکه‌ای طراحی شود که ورودی‌های  $X_1, X_2, \dots, X_n$  را دریافت کرده و تابع اکثریت را محاسبه کند. تابع اکثریت به این صورت است که اگر تعداد ورودی‌هایی که مقدار 1 دارند از نصف تعداد کل ورودی‌ها ( $n/2$ ) بیشتر باشد، خروجی برابر با 1 است. در غیر این صورت، خروجی برابر با 0 خواهد بود.



این شبکه یک لایه ورودی دارد که شامل  $n$  نورون است، و وظیفه آن دریافت ورودی‌های  $X_i$  است. وزن هر ورودی برابر با 1 در نظر گرفته می‌شود تا مقدار هر ورودی مستقیماً به جمع نهایی اضافه شود. برای تصمیم‌گیری، در لایه خروجی یک بایاس ( $b$ ) با مقدار  $-n/2$  تعریف می‌شود. خروجی شبکه از معادله

$$f\left(\sum_{i=1}^n wX_i\right) + b$$

به دست می‌آید، که  $f$  یک تابع فعال (مانند سیگموئید یا Heaviside) است.

(2)

تمامی مفادیر با توجه به فرمول  $\sum w_i x_i + b$  مجاسبه شده.

$\alpha$  گره

مقدار	وزن
0	$w1\alpha$
1	$w2\alpha$
0	$w\alpha$

$\beta$  گره

مقدار	وزن
1	$w1\beta$
0	$w2\beta$
-4	$w\beta$

$\gamma$  گره

مقدار	وزن
0	$w1\gamma$
1	$w2\gamma$
-4	$w\gamma$

$\delta$  گره

مقدار	وزن
1	$w\alpha\delta$
1/2	$w\beta\delta$
-1/2	$w\delta$

### A گره

وزن	مقدار
$w\delta A$	1
$w\gamma A$	1
$wA$	-1

### B گره

وزن	مقدار
$w\alpha B$	1
$w\beta B$	1
$w\gamma B$	-1
$wB$	-1.5

### C گره

وزن	مقدار
$w\alpha C$	-1
$w\beta C$	1
$wC$	-1

منابع