

هوش مصنوعی

دانشکده مهندسی کامپیوتر

گردآورندگان: صادق محمدیان، علیرضا ملک حسینی، شایان شعبانی
بهار ۱۴۰۴



خرداد ۱۴۰۴

کویز دوم

زمان آزمون: ۶۰ دقیقه

۱. لطفا پاسخ خود را با خط خوانا بنویسید.
۲. پاسخ هر سوال را در یک صفحه جدا و شماره پرسش را به صورت واضح در بالای هر صفحه بنویسید.
۳. آزمون از ۱۰۵ نمره است. دریافت ۱۰۰ نمره از ۱۰۵ نمره به منزله دریافت نمره کامل خواهد بود. نمره بالای ۱۰۰ سرریز نخواهد کرد.
۴. نوشته‌های شما در قسمت چرک‌نویس به هیچ عنوان تصحیح نخواهد شد.
۵. استفاده از منابع و لوازم الکترونیکی حین پاسخگویی به سوالات آزمون غیرمجاز است.

پرسش‌های آزمون (۱۰۵ نمره)

پرسش ۱ (۳۰ نمره)

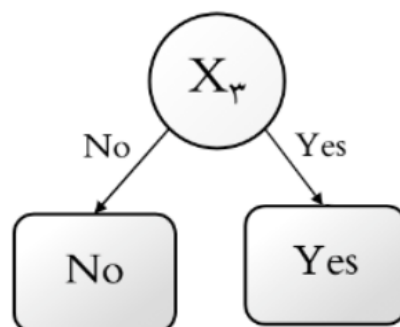
- با توجه به جدول زیر قصد داریم یک درخت تصمیم بسازیم که مقدار Y را براساس X_1, X_2, X_3 و X_3 تعیین کند.

Y	X_3	X_2	X_1
No	No	No	No
Yes	Yes	No	Yes
Yes	Yes	Yes	No
No	No	Yes	Yes

- (الف) (۱۰ نمره) درخت تصمیم را رسم کنید. در هر مرحله ریشه را براساس Gain Information انتخاب کنید. (لزومی به محاسبه عددی IG نیست اما در صورت نیاز از آن استفاده کنید).
- (ب) (۱۰ نمره) آیا درخت بدست آمده (با انتخاب ریشه‌ها از طریق IG) بهینه است؟ اگر بهینه است علت آن را بیان کنید؛ در غیر این صورت درخت بهینه را رسم کنید. (منظور از بهینه، درختی با کوتاه‌ترین ارتفاع ممکن است که نمونه‌های سازگار را جداسازی کند).
- (۱۰ نمره) نشان دهید که هر دسته بند دودویی به فرم $h: \{0, 1\}^d \rightarrow \{0, 1\}$ می‌تواند به صورت یک درخت تصمیم‌گیری به عمق حداکثر $d + 1$ با گره‌هایی به فرم $(x_i = ?)$ برای یک $i \in \{1, \dots, d\}$ پیاده‌سازی شود.

پاسخ

- (آ) درخت در شکل زیر رسم شده است.



(ب) بله

- اثبات (با استقرا روی d):

پایه $d = 0$: وقتی $d = 0$ است، فضای ورودی فقط شامل یک نقطه است. در این حالت، کافی است یک درخت با یک گره برگ بسازیم که برچسب آن $h()$ باشد. عمق این درخت ۱ است که برابر $0 + 1$ می‌باشد.

گام استقرا: فرض کنید ادعا برای $d - 1$ برقرار است. حال تابع

$$h : \{0, 1\}^d \rightarrow \{0, 1\}$$

را در نظر بگیرید. مختصات آخر یعنی x_d را جدا می‌کنیم و دو تابع کمکی تعریف می‌کنیم:

$$h_0(x_1, \dots, x_{d-1}) = h(x_1, \dots, x_{d-1}, 0),$$

$$h_1(x_1, \dots, x_{d-1}) = h(x_1, \dots, x_{d-1}, 1).$$

هر یک از این توابع، روی فضای $\{0, 1\}^{d-1}$ تعریف شده‌اند. طبق فرض استقرا، هر کدام را می‌توان با یک درخت تصمیم به عمق حداکثر d و فقط با پرسش‌هایی از فرم $(x_i = ?)$ برای $i \in \{1, \dots, d-1\}$ پیاده‌سازی کرد.

اکنون یک درخت جدید می‌سازیم که در ریشه، پرسش $(x_d = ?)$ را مطرح کند:

– اگر پاسخ ۰ بود، به زیردرختی برویم که h_0 را پیاده‌سازی می‌کند.

– اگر پاسخ ۱ بود، به زیردرختی برویم که h_1 را پیاده‌سازی می‌کند.

در این حالت، عمق درخت جدید حداکثر $1 + d = d + 1$ خواهد بود و همه پرسش‌ها از فرم $(x_i = ?)$ هستند.

پرسش ۲ (۳۵ نمره) یک مدل خطی به فرم زیر را در نظر بگیرید:

$$y(x_n, \mathbf{w}) = w_0 + \sum_{i=1}^D w_i x_{ni}$$

خطای آن را نیز به صورت زیر در نظر می‌گیریم:

$$E_D(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^N \{y(x_n, \mathbf{w}) - t_n\}^2$$

حال فرض کنید که یک نویز گوسی $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ به هر ورودی x_i اضافه شده است. (ϵ_i) ها به صورت i.i.d تولید شده‌اند. اگر خطای مدل وقتی از $x_i + \epsilon$ استفاده می‌کنیم، $\tilde{E}_D(\mathbf{w})$ باشد، رابطه‌ی زیر را اثبات کنید.

$$E[\tilde{E}_D(\mathbf{w})] = E_D(\mathbf{w}) + \frac{N}{N} \sum_{i=1}^D w_i^2 \sigma^2$$

پاسخ

$$\begin{aligned} e_n &= w_0 + \sum_{i=1}^D w_i (x_{ni} + \epsilon_{ni}) \\ &= y_n + \sum_{i=1}^D w_i \epsilon_{ni} \end{aligned}$$

برای $E_\epsilon(E_D)$ می‌توانیم بنویسیم:

$$\begin{aligned} E_D &= \frac{1}{N} \sum_{n=1}^N (y_{en} - t_n)^2 \\ &= \frac{1}{N} \sum_{n=1}^N \left(y_n^2 + 2y_n \sum_{i=1}^D w_i \epsilon_{ni} + \left(\sum_{i=1}^D w_i \epsilon_{ni} \right)^2 - 2t_n y_n - 2t_n \sum_{i=1}^D w_i \epsilon_{ni} + t_n^2 \right) \end{aligned}$$

اگر از رابطه‌ی بالا امید ریاضی بگیریم، جمله‌ی دوم و پنجم داخل سیگما با توجه به صفر بودن میانگین ϵ_i صفر می‌شوند.

اگر جمله‌ی سوم را باز کنیم، خواهیم داشت:

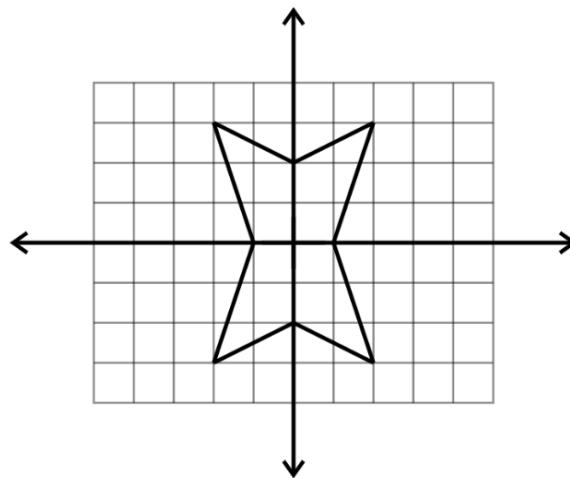
$$\begin{aligned}
E \left[\left(\sum_{i=1}^D w_i \epsilon_{ni} \right)^2 \right] &= E \left[\sum_{i=1}^D \sum_{\substack{j=1 \\ j \neq i}}^D w_i \epsilon_{ni} w_j \epsilon_{nj} + \sum_{i=1}^D w_i^2 \epsilon_{ni}^2 \right] \\
&= \sum_{i=1}^D \sum_{\substack{j=1 \\ j \neq i}}^D w_i w_j E[\epsilon_{ni} \epsilon_{nj}] + \sum_{i=1}^D w_i^2 E[\epsilon_{ni}^2] \\
&= 0 + \sum_{i=1}^D w_i^2 \sigma^2
\end{aligned}$$

پس:

$$\begin{aligned}
E[E_D(w)] &= \frac{1}{N} \sum_{n=1}^N \left(y_n^2 + 0 + \sum_{i=1}^D w_i^2 \sigma^2 - 2 t_n y_n - 0 + t_n^2 \right) \\
&= \frac{1}{N} \sum_{n=1}^N (y_n^2 - 2 t_n y_n + t_n^2) + \frac{N}{N} \sum_{i=1}^D w_i^2 \sigma^2 \\
&= E_D(w) + \frac{N}{N} \sum_{i=1}^D w_i^2 \sigma^2
\end{aligned}$$

پرسش ۳ (۴۰ نمره)

الف) (۲۰ نمره) شبکه ای با دو ورودی x و y طراحی کنید به طوری که برای ورودی که در ناحیه داخلی شکل زیر قرار می گیرد خروجی یک بدهد و در غیر این صفر خروجی شبکه باشد. وزن ها و بایاس ها و توابع فعالسازی را به طور کامل مشخص کنید.



ب) (۱۰ نمره) می دانیم که 2^{2^n} تابع boolean متمایز بر روی n ورودی وجود دارد. در نتیجه برای ۲ ورودی ۱۶ تابع boolean متمایز وجود دارد. چند تا از این ۱۶ تابع را می توان توسط یک perceptron نشان داد؟

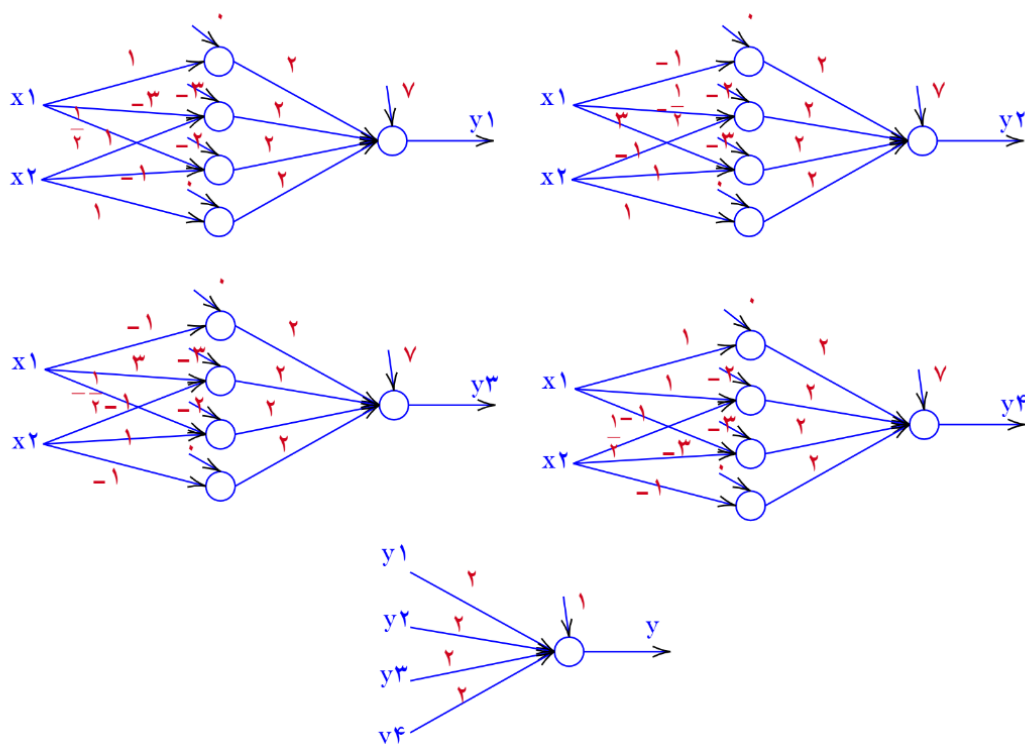
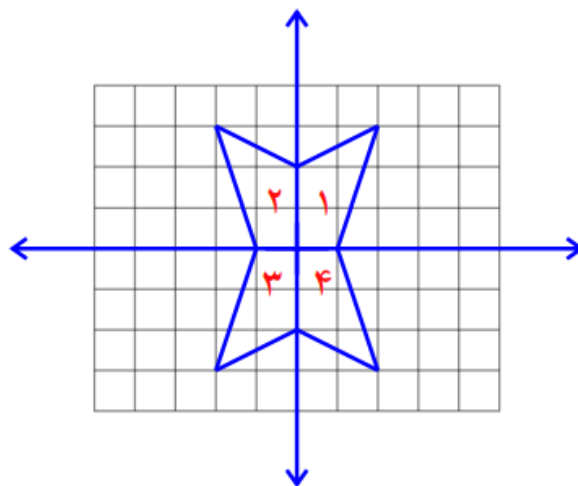
ج) (۱۰ نمره) تابع $f: \{0, 1\}^n \rightarrow \{0, 1\}$ را مطابق عبارت زیر در نظر بگیرید:

$$f(x) = \begin{cases} 1 & \text{if } x \text{ has an odd number of ones,} \\ 0 & \text{if } x \text{ has an even number of ones.} \end{cases}$$

شبکه عصبی با $O(n)$ راس (نورون) بسازید که این تابع را درستی نمایش دهد. شما صرفاً مجاز به استفاده از ReLU به عنوان تابع فعال ساز هستید. مقادیر وزن ها و بایاس شبکه عصبی خود را مشخص کنید.

پاسخ

(آ) شکل را به چهار ناحیه تقسیم می کنیم. برای هر ناحیه i شبکه ی عصبی با خروجی y_i را طوری تشکیل می دهیم که آن ناحیه را پوشش دهد و در نهایت همه ی خروجی ها را OR می کنیم. توابع فعالسازی پله هستند.



(ب) پاسخ برابر ۱۴ است. برای ورودی‌ها ۴ حالت $\{(000), (100), (001), (010), (011), (111)\}$ داریم. برای هر یک از این ۴ حالت، یک perceptron داریم که فقط یکی از این ۴ تا را 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. 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 perceptron‌ها 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، که نماینده مرزهای خطی هستند که همه نقاط روی یک سمت آن قرار دارند.

(ج) این تابع معادل XOR کردن ورودی هاست و XOR را می‌توان با $O(1)$ نورون ساخت. با کنار هم قرار دادن حداکثر n واحد از این شبکه، می‌توانیم شبکه مورد نظر را بسازیم.

