دانشگاه صنعتی شریف
دانشکده‌ی مهندسی کامپیوتر

هوش مصنوعی

بهار ۱۴۰۵

آرش ماری‌اوریاد

مدت زمان: ۶۰ دقیقه

پاسخنامه‌ی کوئیز چهارم (۱۰۰ نمره) Machine Learning تاریخ: ۱۴۰۵/۰۳/۲۴

۱. (۳۰ نمره) با ذکر دلیل، درست یا غلط بودن گزاره‌های زیر را در رابطه با پدیده‌ی Overfitting مشخص نمایید.

(آ) به طور کلی هرچه مدل یادگیری پیچیده‌تر باشد، احتمال Overfitting بیشتر می‌شود.

(ب) به طور کلی هرچه تعداد و تنوع داده‌های آموزش بیشتر باشد، احتمال Overfitting بیشتر می‌شود.

(ج) به طور کلی در روش Laplace Smoothing در الگوریتم Naive Bayes هر چه مقدار هایپرپارامتر K بیشتر باشد، احتمال Overfitting بیشتر می‌شود.

(د) به طور کلی هر چه عمق درخت تصمیم بیشتر باشد، احتمال Overfitting بیشتر می‌شود.

(ه) به طور کلی اضافه کردن عبارت Regularization به تابع هزینه، موجب افزایش احتمال Overfitting می‌شود.

(و) به طور کلی افزایش مقدار ضریب گام (Learning Rate) در الگوریتم Gradient Descent موجب افزایش احتمال Overfitting می‌شود.

حل.

(آ) **درست.** با افزایش پیچیدگی مدل (مانند افزایش تعداد پارامترها، نودها، لایه‌ها یا درجه‌ی چندجمله‌ای در الگوریتم‌های مانند Polynomial Regression)، ظرفیت یادگیری مدل افزایش می‌یابد. در این حالت، مدل به جای یادگیری الگوی کلی، نویزها و جزئیات خاص داده‌های آموزش را نیز حفظ می‌کند که می‌تواند منجر به بیش‌برازش (Overfitting) شود.

(ب) **نادرست.** افزایش تعداد و تنوع داده‌های آموزش باعث می‌شود که مدل با نمونه‌های بیشتری از توزیع واقعی داده‌ها مواجه شود. این امر به مدل کمک می‌کند تا الگوهای عمومی و پایدار را به جای نویزهای تصادفی یا ویژگی‌های خاص یک مجموعه‌ی کوچک یاد بگیرد و در نتیجه احتمال Overfitting کاهش یابد.

(ج) **نادرست.** افزایش هایپرپارامتر K در Laplace Smoothing اثر مشاهدات واقعی را کم‌رنگ‌تر کرده و توزیع احتمالات را به سمت توزیع یکنواخت (Uniform Distribution) سوق می‌دهد. این کار مانند یک تنظیم‌کننده (Regularizer) قوی عمل کرده، و در نتیجه احتمال بیش‌برازش کاهش می‌یابد.

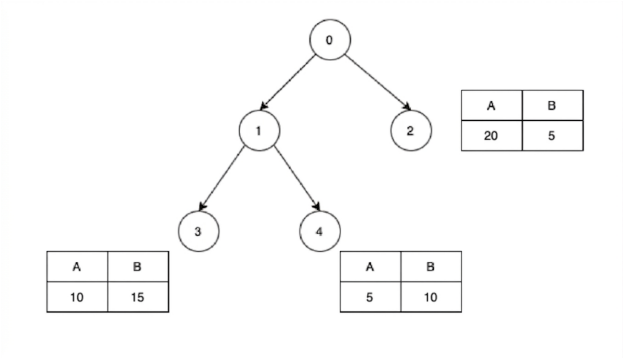
(د) **درست.** هرچه عمق درخت تصمیم بیشتر باشد، مدل می‌تواند شرایط و بخش‌بندی‌های ریزتر و پیچیده‌تری را روی داده‌های آموزش اعمال کند. این امر باعث می‌شود که درخت تا جایی رشد کند که نویزها و داده‌های پرت (Outliers) مجموعه‌ی آموزش را هم کاملا دسته‌بندی کند، که نشان‌دهنده‌ی بیش‌برازش است.

(ه) **نادرست.** هدف اصلی اضافه کردن عبارت Regularization (مانند عبارات L_1 یا L_2) به تابع هزینه، جریمه کردن وزن‌های بزرگ و کنترل پیچیدگی مدل است. این روش متسقیما برای جلوگیری از Overfitting طراحی شده است.

(و) **نادرست.** ضریب گام (Learning Rate) سرعت همگرایی بهینه‌سازی را کنترل می‌کند. افزایش بیش از حد آن معمولاً باعث می‌شود که الگوریتم از روی کمینه‌ی تابع هزینه جهش کند، دچار نوسان شدید شود یا اصلا همگرا نشود (Divergence). این رفتار مستقیما روی Overfitting یا Underfitting نقشی ندارد.

۲. (۴۰ نمره)

(آ) آنتروپی و میزان اطلاعات به دست آمده (Information Gain) بر اساس گره شماره صفر را بدست آورید. دقت نمایید که مسئله دسته‌بندی دو کلاسه می‌باشد که در آن A و B لیبل‌ها می‌باشند و اعداد نوشته شده ذیل هر لیبل، تعداد سمپل‌های موجود در زیرشاخه با آن لیبل است.



(ب) معیار دقت را برای classification انجام شده بر روی این درخت تصمیم به دست آورید.

حل. ابتدا باید تعداد نمونه‌های هر کلاس (A و B) را در تمامی گره‌های درخت مشخص کنیم. با توجه به جداول داده شده برای برگ‌ها (گره‌های ۲، ۳ و ۴) داریم:

- گره ۳ (برگ): $A = ۱۰$ و $B = ۱۵$ (مجموع: ۲۵ نمونه)
- گره ۴ (برگ): $A = ۵$ و $B = ۱۰$ (مجموع: ۱۵ نمونه)
- گره ۲ (برگ): $A = ۲۰$ و $B = ۵$ (مجموع: ۲۵ نمونه)
- گره ۱ (گره میانی): مجموع فرزندانش (گره‌های ۳ و ۴) است. بنابراین $A = ۱۰ + ۵ = ۱۵$ و $B = ۱۵ + ۱۰ = ۲۵$ (مجموع: ۴۰ نمونه)
- گره ۰ (ریشه): مجموع فرزندانش (گره‌های ۱ و ۲) است. بنابراین $A = ۱۵ + ۲۰ = ۳۵$ و $B = ۲۵ + ۵ = ۳۰$ (مجموع: ۶۵ نمونه)

(آ) ابتدا آنتروپی گره ریشه (گره ۰) را محاسبه می‌کنیم. فرمول آنتروپی به شکل زیر است:

$$H(S) = -p_A \log_۲(p_A) - p_B \log_۲(p_B)$$

$$H(\text{Node}۰) = -\left(\frac{۳۵}{۶۵}\right)\log_۲\left(\frac{۳۵}{۶۵}\right) - \left(\frac{۳۰}{۶۵}\right)\log_۲\left(\frac{۳۰}{۶۵}\right) \approx ۰/۹۹۶$$

حال برای محاسبه‌ی Information Gain بر اساس Split انجام شده در گره ۰، باید آنتروپی گره‌های فرزند (گره‌های ۱ و ۲) را محاسبه کنیم:

$$H(\text{Node}۱) = -\left(\frac{۱۵}{۴۰}\right)\log_۲\left(\frac{۱۵}{۴۰}\right) - \left(\frac{۲۵}{۴۰}\right)\log_۲\left(\frac{۲۵}{۴۰}\right) \approx ۰/۹۵۴$$

$$H(\text{Node}۲) = -\left(\frac{۲۰}{۲۵}\right)\log_۲\left(\frac{۲۰}{۲۵}\right) - \left(\frac{۵}{۲۵}\right)\log_۲\left(\frac{۵}{۲۵}\right) \approx ۰/۷۲۲$$

اکنون با استفاده از فرمول Information Gain داریم:

$$IG = H(\text{Node}۰) - \left[\frac{|\text{Node}۱|}{|\text{Node}۰|}H(\text{Node}۱) + \frac{|\text{Node}۲|}{|\text{Node}۰|}H(\text{Node}۲)\right]$$

$$IG = ۰/۹۹۶ - \left[\left(\frac{۴۰}{۶۵}\right)(۰/۹۵۴) + \left(\frac{۲۵}{۶۵}\right)(۰/۷۲۲)\right] \approx ۰/۱۳۲$$

(ب) برای محاسبه دقت درخت تصمیم، باید پیش‌بینی هر برگ را بر اساس کلاس اکثریت (Majority Vote) در آن برگ مشخص کنیم:

- برگ شماره ۳: تعداد B (۱۵) بیشتر از A (۱۰) است: پیش‌بینی: B (تعداد تشخیص صحیح: ۱۵)
- برگ شماره ۴: تعداد B (۱۰) بیشتر از A (۵) است: پیش‌بینی: B (تعداد تشخیص صحیح: ۱۰)
- برگ شماره‌ی ۲: تعداد A (۲۰) بیشتر از B (۵) است: پیش‌بینی: A (تعداد تشخیص صحیح: ۲۰)

تعداد کل نمونه‌ها برابر با ۶۵ است. مجموع تشخیص‌های صحیح در کل درخت برابر است با $۱۵ + ۲۰ + ۱۰ = ۴۵$. بنابراین دقت (Accuracy) برابر است با:

$$\text{Accuracy} = \frac{\text{Predictions Correct}}{\text{Samples Total}} = \frac{۴۵}{۶۵}$$

۳. (۳۰ نمره) می‌دانیم در الگوریتم Logistic Regression از تابع $g(w_۰, w_۱, \ldots, w_d) = w_۰x_۰ + w_۱x_۱ + \cdots + w_dx_d$ برای مدل‌سازی خطی مسئله و از تابع $\sigma(z) = \frac{۱}{۱+e^{-z}}$ به عنوان Activation Function استفاده می‌شود که در آن d تعداد فیچرهای مسئله می‌باشد.

(آ) برای حل یک مسئله دسته‌بندی باینری به کمک الگوریتم Logistic Regression معمولا از تابع هزینه‌ی Binary Cross Entropy استفاده می‌گردد. این تابع هزینه را به صورت کامل بنویسید و تحلیل کنید که چرا تابع هزینه‌ی مناسبی است.

(ب) گرادیان این تابع هزینه را نسبت به پارامتر w_i محاسبه نمایید.

حل.

(آ) تابع هزینه BCE (یا Log-Loss) برای N نمونه‌ی داده‌ی آموزشی به شکل زیر تعریف می‌شود:

$$J(w) = -\frac{۱}{N}\sum_{k=۱}^{N}\left[y^{(k)}\log(\hat{y}^{(k)}) + (۱ - y^{(k)})\log(۱ - \hat{y}^{(k)})\right]$$

که در آن $y^{(k)} \in \{۰, ۱\}$ برچسب واقعی نمونه‌ی kـام و $\hat{y}^{(k)} = \sigma(w^Tx^{(k)})$ پیش‌بینی مدل (احتمال تعلق به کلاس یک) برای آن نمونه است.
دلایل مناسب بودن این تابع هزینه:

- تحدب (Convexity): استفاده از تابع MSE برای رگرسیون لجستیک (به دلیل وجود تابع غیرخطی سیگموید) منجر به یک سطح خطای غیرمحدب با چندین کمینه‌ی محلی (Local Minima) می‌شود. در مقابل، تابع BCE با تابع سیگموید یک ترکیب محدب (Convex) می‌سازد که از نظر ریاضی تضمین می‌کند الگوریتم Gradient Descent همواره به سمت کمینه‌ی سراسری (Global Minimum) همگرا شود.

- با توجه به اینکه تابع log یک تابع اکیدا صعودی در بازه‌ی $(۰, ۱)$ است، تابع هزینه‌ی BCE هر چه یک سمپل به درستی دسته‌بندی شده باشد، هزینه‌ی کمتری اعمال می‌کند و بالعکس در صورتی که سمپل اشتباه دسته‌بندی شود، هزینه‌ی زیادی (میل به بی‌نهایت) اعمال می‌گردد. مثلا یک سمپل با لیبل واقعی $y = ۱$ را در نظر بگیرید. در این صورت عبارت دوم در تابع هزینه برابر با صفر می‌شود. از طرفی در عبارت اول نیز هر چه میزان $\hat{y}$ به عدد ۱ نزدیک‌تر باشد، عبارت $-\log \hat{y}$ به صفر نزدیک‌تر می‌شود. به صورت بر عکس هر چه مقدار $\hat{y}$ به عدد ۰ نزدیک‌تر باشد (یعنی سمپل به اشتباه دسته‌بندی شده باشد)، عبارت $-\log \hat{y}$ به بی‌نهایت میل می‌کند، که هزینه‌ی بسیار زیادی می‌باشد.

(ب) برای محاسبه‌ی گرادیان $\frac{\partial J}{\partial w_i}$ از قاعده‌ی زنجیره‌ای (Chain Rule) استفاده می‌کنیم. برای سادگی، ابتدا گرادیان را برای یک نمونه‌ی منفرد استخراج می‌کنیم:

$$\frac{\partial J}{\partial w_i} = \frac{\partial J}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial z}{\partial w_i}$$

که در آن $z = w_۰x_۰ + \cdots + w_dx_d$ و $\hat{y} = \sigma(z)$.

حال مشتق هر سه بخش را جداگانه محاسبه می‌کنیم:

- مشتق تابع هزینه نسبت به خروجی $\hat{y}$:

$$\frac{\partial J}{\partial \hat{y}} = \frac{\partial}{\partial \hat{y}}\left[-y\log(\hat{y}) - (۱-y)\log(۱-\hat{y})\right] = -\left(\frac{y}{\hat{y}} - \frac{۱-y}{۱-\hat{y}}\right) = \frac{\hat{y}-y}{\hat{y}(۱-\hat{y})}$$

- مشتق تابع فعال‌ساز سیگموید نسبت به ورودی‌اش z:

$$\frac{\partial \hat{y}}{\partial z} = \sigma'(z) = \sigma(z)(۱-\sigma(z)) = \hat{y}(۱-\hat{y})$$

- مشتق ترکیب خطی نسبت به وزن w_i:

$$\frac{\partial z}{\partial w_i} = \frac{\partial}{\partial w_i}(w_۰x_۰ + w_۱x_۱ + \cdots + w_ix_i + \cdots + w_dx_d) = x_i$$

اکنون با جایگذاری و ضرب این سه عبارت، مخرج کسر با عبارت مشتق سیگموید ساده می‌شود:

$$\frac{\partial J}{\partial w_i} = \left[\frac{\hat{y}-y}{\hat{y}(۱-\hat{y})}\right] \cdot [\hat{y}(۱-\hat{y})] \cdot [x_i] = (\hat{y}-y)x_i$$

در نهایت، با بسط دادن این معادله برای کل مجموعه‌ی آموزشی شامل N نمونه، میانگین گرادیان‌ها به صورت زیر به دست می‌آید:

$$\frac{\partial J}{\partial w_i} = \frac{۱}{N}\sum_{k=۱}^{N}(\hat{y}^{(k)} - y^{(k)})x_i^{(k)}$$