



به نام خدا  
درس یادگیری عمیق  
تمرین سری اول  
استاد درس : دکتر محمدرضا محمدی  
دستیاران : نفیسه احمدی، علی سبحانی  
دانشگاه علم و صنعت ایران، دانشکده مهندسی کامپیوتر  
نیمسال دوم تحصیلی ۱۴۰۳ - ۱۴۰۴

## مهلت تحویل : ۱۴۰۳/۱۲/۲۲

لطفا به نکات موجود در سند قوانین انجام و تحویل تمرین ها دقت فرمایید.

## سوالات تئوری



۱. فرض کنید  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  یک تابع مشتق پذیر پیوسته به فرم زیر است:

$$f(x) = \exp(-\|Ax - b\|^2) + \sin\left(\sum_{i=1}^n c_i x_i^2\right)$$

که:

•  $A \in \mathbb{R}^{n \times n}$  یک ماتریس با مرتبه کامل است.

•  $b \in \mathbb{R}^n$  یک بردار بایاس است.

•  $c_i \in \mathbb{R}$  ضرایب ثابت نامنفی یا صفر هستند.

ثابت کنید که برای هر مقدار  $\epsilon > 0$  حداقل یک شبکه پرسپترون چند لایه  $F(x)$  با تعداد محدودی نورون وجود دارد که بتواند تابع  $f(x)$  را به طور دلخواه در یک دامنه فشرده  $D \subset \mathbb{R}^n$  تقریب بزند به طوری که: (۱۵ نمره)

$$\|f(x) - F(x)\|_{L_2} < \epsilon$$

**راهنما:** برای حل این مسئله ابتدا تابع  $f(x)$  را به دو بخش نمایی و سینوسی تقسیم کنید. سپس

با استفاده از مبانی تئوری و شبکه‌های عصبی چندلایه، هر بخش را به‌طور جداگانه تقریب بزنید.  
می‌توانید از منابع زیر برای اثبات خود کمک بگیرید:

- Multilayer Feedforward Networks are Universal Approximators
- Approximation by superpositions of a sigmoidal function

$$K \subset \mathbb{R}^d (d \geq 1), K \text{ compact},$$

$$\sigma : \mathbb{R} \rightarrow \mathbb{R}, \lim_{t \rightarrow -\infty} \sigma(t) = 0, \lim_{t \rightarrow +\infty} \sigma(t) = 1.$$

$$\mathcal{G} = \left\{ G(x) = \sum_{j=1}^r \alpha_j \sigma(w_j \cdot x + \theta_j) : r \in \mathbb{N}, \alpha_j, \theta_j \in \mathbb{R}, w_j \in \mathbb{R}^d \right\}.$$

$$\bar{\mathcal{G}} \|\cdot\|_{\infty} = C(K).$$

$$\exists \Lambda \in C(K)^* \setminus \{0\}, \Lambda(G) = \{0\} \Rightarrow$$

$$\exists \mu \neq 0 (\text{finite signed}) : \Lambda(g) = \int_K g d\mu,$$

$$\int_K \sigma(w \cdot x + \theta) d\mu(x) = 0 \forall w, \theta.$$

$$\sigma_k(t) = \sigma(kt), \sigma_k \rightarrow 1[0, \infty); \int_K 1(w \cdot x + \theta \geq 0) d\mu = 0,$$

$$\mu(H_{w,\theta} \cap K) = 0 \forall w, \theta (H_{w,\theta} = \{x : wx + \theta \geq 0\}).$$

$$\Rightarrow \mu = 0 \text{ (half-spaces generate } \mathcal{B}(\mathbb{R}^d) \text{)} \perp$$

$$\Rightarrow \Lambda = 0, \bar{G} = C(K).$$

$$\forall f \in C(K), \forall \varepsilon > 0, \exists r, \alpha_j, w_j, \theta_j :$$

$$\sup \left| f(x) - \sum_{j=1}^r \alpha_j \sigma(w_j x + \theta_j) \right| < \varepsilon.$$

$$x \in K$$

$$f : K \rightarrow \mathbb{R}^m, f = (f_1, \dots, f_m), \varepsilon > 0 :$$

$$\exists F_k \in \mathcal{G}, \sup_x |f_k(x) - F_k(x)| < \varepsilon / \sqrt{m},$$

$$F = (F_1, \dots, F_m), \sup_{x \in K} \|f(x) - F(x)\|_2 < \varepsilon$$

در این برهان، نخست مجموعه تمام توابع شبکه عصبی تک لایه با فعال ساز سیگموئید را تعریف می کنیم؛ هر تابع این مجموعه ترکیبی خطی محدودی از سیگموئیدهای  $\sigma(w.x + \theta)$  است. گام بعدی نشان می دهد که اگر تابع خطی پیوسته ای روی فضای توابع پیوسته  $C(K)$  وجود داشته باشد که بر تمام اعضای این مجموعه صفر شود، آن تابع باید انتگرال نسبت به یک اندازه  $Signed\mu$  باشد که شرط  $\sigma(w.x + \theta)d\mu = 0$  را برای همه  $(w, \theta)$  برآورده می کند. با بزرگ کردن شیب سیگموئید،  $\sigma(kt)$  به تابع پله واحد همگرا شده و نتیجه می دهد اندازه  $H(w, \theta)$  روی هر نیم فضای  $H(w, \theta)$  صفر

است. چون نیم‌فضاها  $\sigma$  جبر بورل را تولید می‌کنند، تنها اندازه ممکن  $\mu$  صفر است؛ از این تناقض می‌فهمیم هیچ تابع خطی غیرصفر نمی‌تواند کل مجموعه شبکه‌ها را نقض کند. طبق قضیه هان باناک، این امر به چگالی مجموعه شبکه‌ها در  $C(K)$  منتهی می‌شود؛ بنابراین برای هر تابع پیوسته و هر  $\epsilon > 0$ ، شبکه‌ای با تعداد متناهی نورون وجود دارد که خطای یکنواخت آن از  $\epsilon$  کمتر است. در پایان، برای خروجی‌های برداری  $R^m$ ، هر مؤلفه جداگانه تقریب می‌شود و با کنار هم گذاشتن لایه‌های پنهان، شبکه‌ای با خطای  $\epsilon$  در نرم اقلیدسی بر کل دامنه فشرده حاصل می‌شود.



۲. اثبات کنید که افزودن یک ترم  $L_2$  به تابع هزینه:

$$R_\lambda(F) = R(F) + \lambda \|W\|_2^2$$

واریانس مدل را کاهش داده و به بهبود تعمیم‌دهی کمک می‌کند. تأثیر پارامتر  $\lambda$  بر کران تعمیم (generalization bound) را استخراج کنید (۱۰ نمره).

$$\hat{R}_n(F) = \frac{1}{n} \sum_{i=1}^n \ell(F(x_i), y_i), \quad \mathcal{R}_\lambda(F) = \hat{R}_n(F) + \lambda \|W\|_2^2,$$

$$F^* = \arg \min_{F \in \mathcal{H}} \mathcal{R}_\lambda(F), \quad \ell \in [0, 1],$$

$$\lambda \|W^*\|_2^2 \leq 1 \Rightarrow \|W^*\|_2 \leq \frac{1}{\sqrt{\lambda}},$$

$$\hat{\mathcal{R}}_n(\mathcal{H}_B) \leq \frac{L_\sigma R B}{\sqrt{n}} \left( B = \frac{1}{\sqrt{\lambda}} \right) \Rightarrow \hat{\mathcal{R}}_n(\mathcal{H}_B) \leq \frac{L_\sigma R}{\sqrt{n\lambda}},$$

$$\Pr \left( R(F^*) \leq \hat{R}_n(F^*) + \frac{2L_\sigma R}{\sqrt{n\lambda}} + 3\sqrt{\frac{\log(2/\delta)}{2n}} \right) \geq 1 - \delta,$$

$$R(F^*) \leq \hat{R}_n(F^*) + \frac{2L_\sigma R}{\sqrt{n\lambda}} + 3\sqrt{\frac{\log(2/\delta)}{2n}}.$$

در این اثبات، ابتدا ریسک تجربی منظم‌شده را تعریف می‌کنیم. این ریسک شامل دو بخش است: میانگین خطاهای مدل روی داده‌های آموزش و یک جمله‌ی جریمه که به صورت  $\lambda \|W\|_2^2$  تعریف می‌شود. این جمله باعث می‌شود که مدل‌هایی با وزن‌های بزرگ‌تر کمتر ترجیح داده شوند و در نتیجه پیچیدگی مدل کاهش یابد. سپس نشان داده می‌شود که اگر این تابع هزینه را کمینه کنیم، نورم وزن‌های به دست آمده حداکثر برابر با  $1/\lambda$  خواهد بود. به عبارت دیگر، بزرگ‌تر شدن  $\lambda$  باعث می‌شود مدل ساده‌تری انتخاب شود. در ادامه، با استفاده از ویژگی‌هایی مثل محدود بودن داده‌ها و پیوستگی تابع فعال‌سازی (مثلاً سیگموئید)، می‌توان پیچیدگی مدل را به کمک پیچیدگی «رادماخر»

اندازه‌گیری کرد. نتیجه این است که هرچه  $\lambda$  بیشتر باشد، این پیچیدگی کمتر می‌شود. در نهایت، با استفاده از نابرابری‌های آماری (مثل نابرابری بوسکه)، کرانی برای ریسک واقعی مدل به دست می‌آید که شامل سه جمله است: ریسک تجربی، یک جمله‌ی مربوط به پیچیدگی مدل (که به  $\lambda$  وابسته است)، و یک جمله‌ی تصادفی مربوط به تعداد داده‌ها و سطح اطمینان. این کران نشان می‌دهد که اضافه کردن جمله‌ی منظم‌ساز باعث کاهش واریانس مدل و بهبود تعمیم آن روی داده‌های جدید می‌شود، هرچند ممکن است کمی دقت مدل روی داده‌های آموزش را کاهش دهد. بنابراین باید  $\lambda$  را با دقت و بر اساس داده‌های اعتبارسنجی تنظیم کرد تا بین بایاس و واریانس تعادل برقرار شود.



۳. تصور کنید چند سال از فارغ‌التحصیلی شما گذشته و حالا در یک شرکت مشغول به کار هستید. به این نتیجه رسیده‌اید که به‌جای صعود در مسیر شغلی دیگران، کسب‌وکار شخصی خود را راه‌اندازی کنید. شما یک وب‌سایت آموزشی مفید و الهام‌بخش ساخته‌اید که حالا بازدیدکنندگان زیادی دارد و می‌خواهید از طریق تبلیغات آنلاین درآمد کسب کنید (۱۵ نمره).

برای کسب حداکثر درآمد از تبلیغات، به‌جای نمایش تصادفی تبلیغات، از یک سیستم حراجی استفاده می‌کنید که بهترین تبلیغات را برای هر موقعیت انتخاب می‌کند. اطلاعات تبلیغات در جدولی ثبت می‌شود که شامل موارد زیر است:

- **adv\_id**: شناسه تبلیغ‌دهنده
- **cam\_id**: شناسه کمپین تبلیغاتی
- **bid**: مبلغ پیشنهادی برای هر کلیک یا اقدام
- **type**: نوع درآمد (کلیک یا اقدام خاص)
- **pos\_id**: موقعیت تبلیغ در سایت
- **ad\_id**: شناسه تبلیغ
- **views**: تعداد نمایش تبلیغ
- **clicks**: تعداد کلیک‌ها روی تبلیغ
- **actions**: تعداد اقدامات انجام‌شده پس از کلیک
- **week\_id**: زمان جمع‌آوری داده‌ها (بر اساس هفته)

با استفاده از این داده‌ها و فرمول‌های بهینه‌سازی که در ادامه ارائه شده است، می‌توانید انتخاب تبلیغات را در جایگاه‌های مختلف بهینه کنید تا درآمد شما حداکثر شود. به این روابط دقت کنید

week id	actions	clicks	views	ad id	pos id	type	bid	cam id	adv id
۱۷۳۵۱۱۵۵۶۳	۰	۱۵	۱۵۴۳۶	۸۹۷۵	۱۰	Click	۱۰۰۰	۶۵۷۵	۱۲۳۴
۱۷۳۵۱۱۵۵۶۳	۰	۱۰	۱۸۴۶۶	۶۷۳۵	۱۳	Click	۱۰۰۰	۶۵۷۵	۱۲۳۴
۱۷۳۵۱۲۴۵۶۹	۲	۲۰	۱۰۳۲۱	۷۱۸۵	۷۸	Action	۹۰۰۰	۹۸۷۶	۴۳۲۱
۱۷۳۵۱۱۴۵۶۳	۰	۲۵	۲۱۰۰۰	۱۰۲۴	۵	Click	۵۰۰	۴۵۳۲	۵۶۷۸
۱۷۳۵۱۱۸۵۶۳	۰	۳۰	۱۸۰۰۰	۲۳۴۱	۲۰	Click	۲۰۰۰	۳۴۵۶	۲۳۴۵
۱۷۳۵۱۳۴۵۶۳	۵	۱۸	۱۲۰۰۰	۶۵۲۳	۲۵	Action	۸۰۰۰	۷۶۵۴	۷۸۹۰

و سعی کنید درک کنید که چرا این فرمول ها می توانند به انتخاب تبلیغات با بیشترین درآمد مورد انتظار کمک کنند.

### فرمول بهینه سازی تبلیغات کلیکی

For each position( $p$ ), select:  $\arg \max_{ad \in A(p)} (bid_{ad} \times ctr(ad, p))$

### فرمول بهینه سازی تبلیغات اکشنی

For each position( $p$ ), select:  $\arg \max_{ad \in A(p)} (bid_{ad} \times ctr(ad, p) \times cvr_{ad})$

- $A(p)$ : مجموعه تبلیغاتی که امکان نمایش در جایگاه  $p$  دارند.
- $bid_{ad}$ : مبلغی که تبلیغ دهنده برای هر کلیک پرداخت می کند.
- $ctr(ad, p)$ : احتمال کلیک کاربر بر روی تبلیغ  $ad$  در جایگاه  $p$  (نرخ کلیک).
- $cvr_{ad}$ : احتمال انجام اکشن توسط کاربر پس از کلیک روی تبلیغ.

فرض کنید مدل های آموزش داده شده در مواجهه با ورودی های ناشناخته (مانند تبلیغات، کمپین ها یا تبلیغ دهندگان جدید)، به صورت میانگین گیری سیستماتیک عمل می کنند. به طور مشخص، اگر یک تبلیغ جدید باشد اما کمپین مرتبط با آن قبلاً در سیستم دیده شده باشد،  $CVR$  و  $CTR$  آن تبلیغ به صورت میانگین وزنی از مقادیر مربوط به تبلیغات قبلی آن کمپین محاسبه می شود.

**الف)** یکی از مهم ترین مراحل در فرآیند آموزش هر مدل یادگیری ماشین، تقسیم بندی داده ها به مجموعه های Train، Dev، Train-Dev و Test است. این کار به ما کمک می کند تا عملکرد مدل را بر روی توزیع داده های مختلف بررسی کنیم. در این مسئله خاص چطور این کار را باید انجام داد و در انتخاب این مجموعه ها به چه نکاتی باید توجه داشت؟

**الف)** چگونه داده ها را به مجموعه های Train، Test و Dev تقسیم کنیم.

(آ) تقسیم تصادفی در مقابل تقسیم زمانی (Time-Based Split):

- اگر رفتار کاربران، تبلیغ کنندگان یا مبلغ های پیشنهادی (bids) در طول زمان به شکل قابل ملاحظه ای تغییر کند (اتفاقی که در دنیای واقعی تبلیغات بسیار معمول است)، استفاده از یک تقسیم مبتنی بر زمان رویکرد واقعی تری خواهد بود.

□ مثال: استفاده از داده های ماه ژانویه تا مارس برای آموزش (Train)، آوریل برای توسعه (Dev) و مه برای آزمون (Test).

- اگر الگوهای زمانی قوی در داده ها وجود نداشته باشد یا نگرانی از نشت داده در طول زمان نداشته باشیم، تقسیم تصادفی میتواند کفایت کند. با این حال، تقسیم تصادفی ممکن است تغییرات زمانی متداول در تبلیغات را پنهان کند.

(ب) حفظ توزیع نماینده (Representative Distribution):

- باید مطمئن شد که هر بخش، توزیع مشابهی از تبلیغ دهندگان، کمپینها، جایگاههای نمایش (positions) و سطوح عملکرد CTR، CVR و غیره داشته باشد.
- اینکار مانع میشود که مثلاً یک تبلیغ دهنده یا موقعیت خاص فقط در دیتاست آموزش باشد و هرگز در دیتاست Dev یا Test مشاهده نشود.

(ج) جلوگیری از نشت داده (Data Leakage):

- اگر ویژگیها در سطح کاربر یا کمپین تعریف شده اند، باید اطمینان حاصل کنیم که داده های یک کاربر یا کمپین به گونهای بین Train و Dev/Test تقسیم نشود که به طور مصنوعی عملکرد مدل را بهتر نشان دهد.
- در سامانه های مبتنی بر مزایده، کمپین ها ممکن است در طول زمان تکامل یابند. برای مثال میتوان داده ها را در سطح کمپین گروه بندی کرد و سپس تقسیم بر اساس کمپین انجام داد تا عملکرد روی داده های آینده واقعا داده های «دیده نشده» را منعکس کند.

(ب) عوامل کلیدی در تقسیم داده

(آ) الگوهای زمانی: رفتار تبلیغات (استراتژی مزایده، مشارکت کاربران و غیره) معمولاً در طول زمان تغییر میکند.

(ب) فصلی بودن: برخی هفته ها یا ماهها، ترافیک یا مشارکت کاربران بالاتر/پایین تر است (مثلاً رویدادهایی مانند جمعه سیاه یا تعطیلات).

(ج) تغییر توزیع (Distribution Shift): ورود تبلیغ دهندگان جدید، موقعیتهای جدید، یا تغییر در جمعیت کاربران باعث می شود توزیع داده ها متفاوت شود.

(د) کمیابی برچسب های مثبت: در کمپین های «اقدام (Action)» که CVR پایین است، باید مراقب بود در تقسیم داده حداقل تعداد قابل قبولی از نمونه های مثبت (تبدیل ها) در مجموعه های Train و Dev قرار گیرد.

ب-1) در هر یک از سناریوهای زیر، به طور مختصر مشکل را معرفی کرده و راه حلی برای آن ارائه دهید:

(آ) از قطعیت داده های ورودی اطمینان داریم ولی خطای آموزش مدل (Training Error) بالا است.

- مشکل احتمالی : مدل دچار Underfitting شده است و نمی تواند پیچیدگی داده ها را فرا بگیرد.

- ممکن است مدل بیش از حد ساده باشد، ویژگی های کافی در دسترس نباشد، یا ابرپارامترها (Hyperparameters) نامناسب تنظیم شده باشند.

- راهکارهای پیشنهادی:

- افزایش پیچیدگی مدل: استفاده از معماری های قویتر یا افزودن ویژگی های مهمتر.

- کاهش منظم سازی: (Regularization) اگر مدل بیش از حد منظم شده باشد، ممکن است بیش از حد محدود شود.

- تنظیم ابرپارامترها: تنظیم نرخ یادگیری، تعداد لایه ها (در شبکه های عمیق)، یا عمق درختها (در مدل های درخت تصمیم).

ب) خطای آموزش مدل پایین است ولی خطای آن روی مجموعه (Train-Dev) همچنان بالا است.

- مشکل احتمالی : مدل در یادگیری داده های آموزشی موفق عمل کرده اما در مجموعه Train-Dev عملکرد خوبی ندارد؛ نشانه ای از Overfitting به مجموعه Train است.

- راهکارهای پیشنهادی:

- افزایش داده یا استفاده از Data Augmentation برای کاهش بیش برازش.

- افزایش منظم سازی: مثلاً اعمال Dropout، L2 در شبکه های عصبی، یا توقف

زود هنگام Early Stopping



□□□ انتخاب ویژگی (Feature Selection): حذف یا محدود کردن ویژگی هایی که باعث می شوند مدل جزئیات نویزی مجموعه Train را حفظ کند.

(ج) خطای مدل در مجموعه های Train و Train-Dev پایین است ولی روی مجموعه Dev خطا زیاد است.

- مشکل احتمالی: مدل روی داده هایی که مشابه داده آموزشی هستند (Train-Dev) به خوبی تعمیم می یابد ولی روی مجموعه Dev ضعیف عمل میکند. این اغلب نشان از ناهمخوانی توزیع (Distribution Mismatch) دارد؛ داده Dev ممکن است شامل تبلیغ دهندگان جدید، موقعیت های متفاوت یا دوره زمانی دیگری باشد.
- راهکارهای پیشنهادی:

□ بررسی تفاوت توزیع: اطمینان پیدا کنید داده Dev از همان توزیع Train باشد یا دست کم نماینده شرایط دنیای واقعی باشد.

□□ انطباق دامنه (Domain Adaptation): اگر داده Dev واقعا توزیع متفاوتی دارد (مثلاً کاربران یا تبلیغ دهندگان جدید)، مدل را مجدداً آموزش یا ریزتنظیم (Fine-tune) کنید تا با آن توزیع سازگار شود.

(د) خطای Dev پایین است ولی روی مجموعه Test خطا همچنان زیاد است.

- مشکل احتمالی: مدل به مجموعه Dev بیش برآزش کرده یا مجموعه Dev به قدر کافی نماینده مجموعه Test نبوده است. گاهی اوقات، با تنظیم بیش از حد ابرپارامترها روی Dev، عملکرد مدل در تست واقعی افت می کند.
- راهکارهای پیشنهادی:

□ استفاده از یک مجموعه Test جدید: یا یک مجموعه Hold-out جداگانه.

□□ بازبینی فرایند تقسیم: اطمینان حاصل کنید مجموعه Dev به اندازه کافی بزرگ و متنوع است.

□□□ منظم سازی یا استفاده از Cross-Validation: برای جلوگیری از تنظیم بیش از حد روی Dev.

ب-۲) آیا در اولین سناریوی مطرح شده در قسمت قبل، افزایش سایز داده‌های آموزش راه حل خوبی خواهد بود؟

- اگر مشکل Underfitting به این دلیل باشد که مدل ظرفیت کافی برای یادگیری روابط راندارد، صرف افزودن داده ممکن است مشکل را حل نکند. باید ظرفیت مدل یا مهندسی ویژگی را بهبود داد.
- با این حال، اضافه کردن داده معمولاً مضر نیست؛ فقط شاید اولویت اول برای رفع Underfitting نباشد.

ب-۳) فرض کنید مدلهایی که برای پیش‌بینی نرخ کلیک (CTR) و نرخ تبدیل (CVR) تبلیغات آموزش داده‌اید، در داده‌های آموزش به دقت بالایی دست یافته‌اند. با این حال، زمانی که این مدل‌ها بر روی داده‌های Dev اعمال می‌شوند و شما قصد دارید درآمد را بهینه کنید، اختلاف قابل توجهی بین پیش‌بینی‌های مدل و درآمد واقعی مشاهده می‌کنید. علت این خطا را شناسایی کرده و تحلیل کنید که چه عواملی ممکن است باعث این اختلاف شوند.

توجه: جدول ارائه شده صرفاً برای آشنایی با ساختار داده‌ها و فضای مسئله است و مقادیر آن فاقد اهمیت هستند.

(آ) دقت بالای CTR/CVR اما تفاوت با درآمد واقعی:

□. ناهمخوانی تابع هدف (Objective Function):

- صرفاً دقت بالای CTR یا CVR لزوماً به حداکثر درآمد منجر نمی‌شود. درآمد به موارد ذیل بستگی دارد:

$$bid \times CTR \text{ (for clicks) or } bid \times CTR \times CVR \text{ (for actions)}$$

- مدلی که تنها روی پیش‌بینی CTR/CVR تمرکز کرده و مقدار bid یا پراکندگی این پیش‌بینی‌ها را در نظر نگیرد، ممکن است به حداکثر درآمد نرسد.

□□. توزیع مبلغ‌های پیشنهادی (Bids):

- ممکن است برخی تبلیغ دهندگان مبلغهای پیشنهادی بالایی داشته باشند اما عملکرد نادر یا الگوهای خاصی داشته باشند. اگر مدل به ندرت داده های چنین کمپینهایی را ببیند، پیشبینی درآمد آنها میتواند خطا داشته باشد.

□□□ سوگیری انتخاب (Selection Bias):

- اگر مدل صرفاً روی تبلیغاتی آموزشی ببیند که در گذشته اغلب نمایش داده شده اند، پیشبینی در مورد تبلیغات جدید یا سناریوهای دیده نشده ممکن است دقیق نباشد.
- داده های قدیمی ممکن است فقط انواع خاصی از تبلیغات (با CTR بالاتر) را شامل شود و باعث شود مدل به آن نواحی از فضا بیشتر توجه کند.

□□. نشت داده یا برچسب گذاری نادرست:

- اگر ثبت کلیک ها یا اکشن ها کامل نباشد یا با تأخیر انجام شود، مدل الگوهای نادرستی یاد می گیرد.

- اگر تبدیل (Conversion) به روش غلط نسبت داده شود (مثلاً اتریبیوشن کلیک آخر ممکن است بعضی تبدیلهای را به درستی نشمارد)، برچسب CVR دقیقاً بیانگر رفتار واقعی کاربر نخواهد بود.

(ب) مشکلات ساختاری یا داده ای احتمالی:

- . بهینه نکردن معیار مناسب: ممکن است فرایند آموزش صرفاً روی تابع زیان استاندارد (مثلاً باینری کراس انتروپی) برای CTR متمرکز باشد، بدون در نظر گرفتن مبنای درآمد یا تابع رتبه بندی (Ranking) بر اساس  $bid \times CTR$

- . عدم تفکیک کافی جزئیات: درآمد واقعی وابسته به فرکانس نمایش، محدودیت بودجه تبلیغ دهندگان و محدودیتهای دیگر است. اگر در مدل این موارد لحاظ نشود، اختلاف بین پیشبینی درآمد و درآمد واقعی زیاد میشود.

- . حلقه های بازخورد متفاوت: پس از اینکه مدلی تبلیغات خاصی را زیاد نمایش داد (چون پیشبینی می کرد خوب هستند)، ممکن است به دلیل اشباع کاربران (User Fatigue) یا دیگر پویایی های پلتفرم، عملکرد واقعی افت کند و درآمد واقعی کمتر از پیشبینی شود.

(ج) در یک سیستم بهینه سازی تبلیغات مبتنی بر داده های واقعی، یکی از چالش های اساسی، مواجهه با تغییرات ناگهانی در رفتار کاربران (Concept Drift) و نامتوازن بودن داده ها است. فرض کنید در دوره های زمانی خاصی (مانند مناسبت های خاص یا تغییر الگوریتم جستجو در موتورهای جستجو)،

نرخ کلیک (CTR) و نرخ تبدیل (CVR) به طور ناگهانی دچار تغییرات چشمگیر می‌شوند.

- چگونه می‌توان پایداری مدل را در مواجهه با Concept Drift تضمین کرد؟ برای حفظ پایداری مدل در شرایطی که رفتار کاربر دچار تغییرات ناگهانی می‌شود، میتوان از روشهای زیر بهره برد:
  - به روزرسانی مداوم مدل: مدلها را به صورت دوره ای یا به صورت پیوسته با استفاده از یادگیری آنلاین مجددا آموزش داد تا همواره به داده های جدید تطبیق یابند.
  - Sliding Window Approach: به جای استفاده از کل داده ها، فقط از آخرین بازه زمانی مثلا ۴ هفته اخیر برای آموزش مدل استفاده شود. این کار به مدل کمک می کند تا سریعتر به تغییرات واکنش نشان دهد.
  - استفاده از مدل های ensemble: ترکیب چندین مدل که هر کدام بر روی بخشهای متفاوت یا دوره های زمانی مختلف آموزش دیده اند، می تواند به کاهش حساسیت به تغییرات ناگهانی کمک کند.
  - Adaptive Boosting: الگوریتم های تقویتی تطبیقی مانند Adaptive Random Forest قادر به تشخیص و سازگاری با تغییرات سریع در داده ها هستند.
  - Early Drift Detection Method روش هایی مانند (DDM) Drift Detection Methods (EDDM) می توانند تغییر در توزیع داده ها را تشخیص داده و هشدار لازم برای بازآموزی مدل را صادر کنند.

- چه روش هایی برای مدیریت داده های نامتوازن در این مسئله مناسب هستند؟

در بیشتر سیستم های تبلیغاتی، نرخ کلیک و اقدام CTR و CVR به صورت ذاتی بسیار نامتوازن هستند (اغلب تبلیغات کلیک یا اقدام نمی گیرند). روشه ای زیر می توانند به بهبود عملکرد مدل در این شرایط کمک کنند:

□ تکنیک های نمونه برداری:

✱ افزایش نمونه های اقلیت (Oversampling): استفاده از روش هایی مانند SMOTE

(Synthetic Minority Over-sampling Technique) برای تولید نمونه های مصنوعی.

✱ کاهش نمونه های اکثریت (Undersampling): حذف بخشی از نمونه های کلاس اکثریت

به منظور تعادل داده ها.

□ یادگیری باتابع هزینه حساس (Cost-sensitive Learning): تنظیم الگوریتم به گونه ای که

اشتباهات در پیشبینی کلاس های اقلیت هزینه بالاتری داشته باشند.

□ استفاده از تکنیک های ensemble: الگوریتم های مانند Random Forest و Boosting می توانند با تنظیم وزن ها به بهبود عملکرد مدل در داده های نامتوازن کمک کنند.

- تحقیق کنید که چگونه می توان با استفاده از الگوریتم های آنلاین یادگیری (Online Learning)، عملکرد سیستم را بهبود بخشید و به تغییرات سریع بازار واکنش نشان داد.


□ به روزرسانی مدل به صورت افزایشی: الگوریتم هایی نظیر Online Gradient Descent یا FTRL-Proximal به مدل اجازه می دهند تا به طور لحظه ای با داده های ورودی جدید سازگار شود.


□ مدل های خودتنظیم: استفاده از مدل هایی که به طور خودکار پارامترهای خود را بر اساس داده های ورودی به روز می کنند (Adaptive Learning Rate) کمک می کند تا مدل همواره عملکرد بهینه داشته باشد.

□ توسعه سیستم های یادگیری پیوسته: طراحی سیستم های یادگیری که بتوانند در محیط های پویایی مانند تبلیغات آنلاین، تغییرات را به سرعت ثبت و به روزرسانی کنند.

□ مدیریت حافظه و پنجره های زمانی: استفاده از تکنیک های windowing که در آن داده های قدیمی حذف شده و تنها داده های جدید یا اخیر در مدل نگه داشته می شود، تا مدل بتواند به تغییرات جدید واکنش نشان دهد.

## سوالات عملی

۴.  هدف این تمرین، پیش پردازش داده و پیاده سازی دستی ماژول های شبکه عصبی است. شما باید بخش های `#TODO` را تکمیل کنید. لذا از هرگونه تغییر یا دستکاری ساختار اصلی کد اجتناب فرمایید. کلیه کدها باید از قبل اجرا شده باشند؛ در غیر این صورت نمره مربوطه تعلق نخواهد گرفت (۲۵ نمره).

۵.  هدف این تمرین، ایجاد ماژول های شبکه عصبی و ترکیب آنها جهت ساخت یک شبکه کامل است. شما باید بخش های `#TODO` و سوالاتی که در نوت بوک بیان شده

را تکمیل کنید. لذا از هرگونه تغییر یا دستکاری ساختار اصلی کد اجتناب فرمایید. کلیه کدها باید از قبل اجرا شده باشند؛ در غیر این صورت نمره مربوطه تعلق نخواهد گرفت (۳۵ نمره).