مینی‌پروژهٔ شمارهٔ دو

در انجام این مینی‌پروژه حتماً به نکات زیر توجه کنید:

- موعد تحویل این مینی‌پروژه، ساعت ۱۸:۰۰ روز جمعه ۲۸ اردیبهشت‌ماه ۱۴۰۳ است.
- اطلاعات خود را در این گوگل‌شیت مطابق نمونهٔ سطر دوم تکمیل کنید.
- برای این مینی‌پروژه ملزم به ارائهٔ گزارش متنی شامل توضیحات کامل هر قسمت هستید. هم گزارش و هم کدهای خود را در گیت‌هاب و سامانهٔ دانشگاه بارگذاری کنید.
- برای گزارش لازم است که پاسخ هر سوال و زیربخش‌هایش به‌ترتیب و به‌صورت مشخص نوشته شده باشند. بخش زیادی از نمره به توضیحات دقیق و تحلیل‌های کافی شما روی نتایج بستگی خواهد داشت.
- لازم است که در صفحهٔ اول گزارش خود لینک پوشهٔ گیت‌هاب و گوگل‌کولب مربوط به مینی‌پروژهٔ خود را در حالتی که دسترسی Public دارد به اشتراک گذاشته باشید. دفترچه‌کد گوگل‌کولب باید به‌صورت منظم و با بخش‌بندی مشخص تنظیم شده باشد، و خروجی سلول‌های اجراشده قابل مشاهده باشد. در گیت‌هاب هم برای هر مینی‌پروژه یک پوشهٔ مجزا ایجاد کنید.
- هرجا از دفترچه‌کد گوگل‌کولب شما نیاز به فراخوانی فایلی خارج از محیط داشت، مطابق آموزش‌های ارائه‌شده ملزم هستید از دستور gdown استفاده کنید و مسیرهای فایل‌ها را طوری تنظیم کنید که صرفاً با اجرای سلول‌های کد، امکان فراخوانی و خواندن فایل‌ها توسط هر کاربری وجود داشته باشد.
- در تمامی مراحل تعریف داده و مدل و هرجای دیگری که مطابق آموزش‌های ویدیویی و به لحاظ منطقی نیاز است، Random State را برابر با دو رقم آخر شمارهٔ دانشجویی خود در نظر بگیرید.
- استفاده از ابزارهای هوشمند (مانند ChatGPT) در کمک‌گرفتن برای بهبود کدها مجاز است؛ اما لازم است تمام جزئیات مواردی که در خروجی‌های مختلف گزارش خود عنوان می‌کنید را به خوبی خوانده، درک و تحلیل کرده باشید. استفاده از این ابزارهای هوشمند در نوشتن گزارش و تحلیل‌ها ممنوع است.
- در جاهایی که با توجه به دو رقم آخر شمارهٔ دانشجویی خود محدود به انتخاب عدد، متغیر و یا داده‌ای خاص شده‌اید، برای تست‌های اضافه‌تر و نمایش بهبود در نتایج خود، مجاز هستید از مقادیر دیگر هم استفاده کنید. ۱۵ تا ۲۰ درصد از نمرهٔ هر سوال به بهترین نتایج کسب‌شده اختصاص خواهد یافت.
- رعایت نکات بالا به حرفه‌ای‌ترشدن شما کمک خواهد کرد و اهمیتی معادل مطالب درسی فراگرفته‌شده دارد؛ بنابراین، درصورت عدم رعایت هریک از این نکات، گزارش شما تصحیح نخواهد شد.

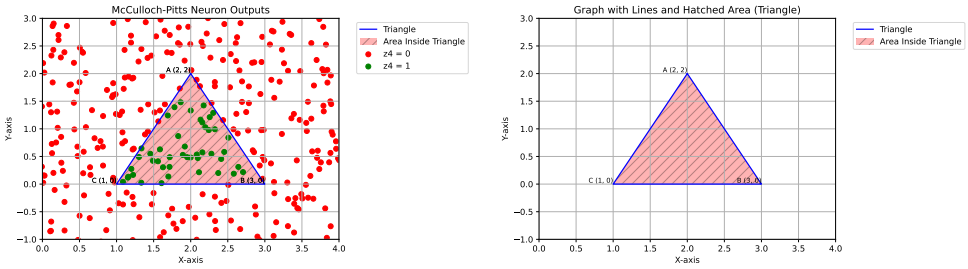
۱ سوال اول

۱. فرض کنید در یک مسألهٔ طبقه‌بندی دوکلاسه، دو لایهٔ انتهایی شبکهٔ شما فعال‌ساز ReLU و سیگموید است. چه اتفاقی می‌افتد؟

۲. یک جایگزین برای ReLU در معادله ۱ آورده شده است. ضمن محاسبهٔ گرادیان آن، حداقل یک مزیت آن نسبت به ReLU را توضیح دهید.

$$\mathrm{ELU}(x) = \begin{cases} x & x >= 0 \\ \alpha\left(e^{x} - 1\right) & x < 0 \end{cases} \qquad (۱)$$

۳. به کمک یک نورون ساده یا پرسپترون یا نورون McCulloch-Pitts[۱] شبکه‌ای طراحی کنید که بتواند ناحیهٔ هاشورزدهٔ داخل مثلثی که در نمودار شکل ۱(آ) نشان داده شده را از سایر نواحی تفکیک کند. پس از انجام مرحلهٔ طراحی شبکه (که می‌تواند به‌صورت دستی انجام شود)، برنامه‌ای که در این دفترچه‌کد و در کلاس برای نورون McCulloch-Pitts آموخته‌اید را به‌گونه‌ای توسعه دهید که ۲۰۰۰ نقطهٔ رندوم تولید کند و آن‌ها را به‌عنوان ورودی به شبکهٔ طراحی‌شده توسط شما دهد و نقاطی که خروجی «۱» تولید می‌کنند را با رنگ سبز و نقاطی که خروجی «۰» تولید می‌کنند را با رنگ قرمز نشان دهد. خروجی تولیدشده توسط برنامهٔ شما باید به‌صورتی که در شکل ۱(ب) نشان داده شده است باشد (به محدودهٔ عددی محورهای x و y هم دقت کنید). اثر اضافه‌کردن دو تابع فعال‌ساز مختلف به فرآیند تصمیم‌گیری را هم بررسی کنید.



(آ) نمودار هاشورزدهٔ مورد سوال

(ب) خروجی مطلوب برنامه

شکل ۱: نمودارهای مربوط به بخش «۳» سوال اول و خروجی برنامه.

۲ سوال دوم

۱. دیتاست CWRU Bearing که در «مینی‌پروژهٔ شمارهٔ یک» با آن آشنا شدید را به خاطر آورید. علاوه بر دو کلاسی که در آن مینی‌پروژه در نظر گرفتید، با مراجعه به صفحهٔ داده‌های عیب در حالت 12k، دو کلاس دیگر نیز از طریق فایل‌های B007_X[۲] و OR007@6_X اضافه کنید. با انجام این کار یک کلاس دادهٔ سالم و سه کلاس از داده‌های دارای سه عیب متفاوت خواهید داشت. در مورد این که هر فایل مربوط به چه نوع عیبی است به‌صورت کوتاه توضیح دهید.

سپس در ادامه، تمام کارهایی که در بخش «۲» سوال دوم «مینی‌پروژهٔ یک» برای استخراج ویژگی و آماده‌سازی دیتا انجام داده بودید را روی دیتاست جدید خود پیاده‌سازی کنید. در قسمت تقسیم‌بندی داده‌ها، یک بخش برای «اعتبارسنجی» به بخش‌های «آموزش» و «آزمون» اضافه کنید و توضیح دهید که کاربرد این بخش چیست.

۲. یک مدل Multi-Layer Perceptron (MLP) ساده با ۲ لایهٔ پنهان یا بیش‌تر بسازید. بخشی از داده‌های آموزش را برای اعتبارسنجی کنار بگذارید و با انتخاب بهینه‌ساز و تابع اتلاف مناسب، مدل را آموزش دهید. نمودارهای اتلاف و Accuracy مربوط به آموزش و اعتبارسنجی را رسم و نتیجه را تحلیل کنید. نتیجهٔ تست مدل روی داده‌های آزمون را با استفاده ماتریس درهم‌ریختگی و classification_report نشان داده و نتایج به‌صورت دقیق تحلیل کنید.

۳. فرآیند سوال قبل را با یک بهینه‌ساز و تابع اتلاف جدید انجام داده و نتایج را مقایسه و تحلیل کنید. بررسی کنید که آیا تغییر تابع اتلاف می‌تواند در نتیجه اثرگذار باشد؟

[۱] تشخیص اینکه با کدام روش می‌توانید این کار را انجام دهید با شماست.

[۲] X، باقی‌ماندهٔ تقسیم دو رقم آخر شمارهٔ دانشجویی شما بر ۴ است.

۴. در مورد K-Fold Cross-validation و Stratified K-Fold Cross-validation و مزایای هریک توضیح دهید. سپس با ذکر دلیل، یکی از این روش‌ها را انتخاب کرده و بخش «۲» سوال سوم را با آن پیاده‌سازی کنید و نتایج خود را تحلیل کنید.

۳ سوال سوم

یکی از مجموعه‌داده‌های مربوط به طبقه‌بندی پوشش جنگلی یا دارو را در نظر بگیرید.

۱. با استفاده از بخشی از داده‌ها، مجموعه‌داده را به دو بخش آموزش و آزمون تقسیم کنید (حداقل ۱۵ درصد از داده‌ها را برای آزمون نگه دارید). توضیح دهید که از چه روشی برای انتخاب بخشی از داده‌ها استفاده کرده‌اید. آیا روش بهتری برای این کار می‌شناسید؟
در ادامه، برنامه‌ای بنویسید که درخت تصمیمی برای طبقه‌بندی کلاس‌های این مجموعه‌داده طراحی کند. خروجی درخت تصمیم خود را با برنامه‌نویسی و یا به‌صورت دستی تحلیل کنید.

۲. با استفاده از ماتریس درهم‌ریختگی و حداقل سه شاخصهٔ ارزیابی مربوط به وظیفهٔ طبقه‌بندی، عملکرد درخت آموزش‌داده‌شدهٔ خود را روی بخش آزمون داده‌ها ارزیابی کنید و نتایج را به‌صورت دقیق گزارش کنید.
تأثیر مقادیر کوچک و بزرگ حداقل دو فراپارامتر را بررسی کنید. تغییر فراپارامترهای مربوط به هرس‌کردن چه تأثیری روی نتایج دارد و مزیت آن چیست؟

۳. توضیح دهید که روش‌هایی مانند جنگل تصادفی و AdaBoost چگونه می‌توانند به بهبود نتایج کمک کنند. سپس، با انتخاب یکی از این روش‌ها و استفاده از فراپارامترهای مناسب، سعی کنید نتایج پیاده‌سازی در مراحل قبلی را ارتقاء دهید.
راهنمایی: می‌توانید از پیوندهای زیر کمک بگیرید:

- sklearn.ensemble.RandomForestClassifier
- sklearn.ensemble.AdaBoostClassifier

اگر به دقت کلیِ آزمونِ زیر ۸۰ درصد رسیده‌اید یا تحلیل درخت تصمیم به‌صورت دستی برایتان مشکل شده است لازم است با ذکر توضیحات، پیاده‌سازی‌هایی علاوه بر پیاده‌سازی‌های قبلی و با فراپارامترهای جدید جهت حل این مشکلات انجام دهید. هم‌چنین می‌توانید حداقل چهار فراپارامتر برای درخت تصمیم خود در نظر بگیرید و این فراپارامترها را با روش‌هایی مانند GridSearch بهینه کنید.

۴ سوال چهارم

دیتاست بیماری قلبی را در نظر بگیرید. داده‌ها را به دو بخش آموزش و آزمون تقسیم کرده و ضمن انجام پیش‌پردازش‌هایی که روی آن لازم می‌دانید و با فرض گاوسی‌بودن داده‌ها، از الگوریتم طبقه‌بندی Bayes استفاده کنید و نتایج را در قالب ماتریس درهم‌ریختگی و classification_report تحلیل کنید. تفاوت میان دو حالت Macro و Micro را در کتابخانهٔ سایکیت‌لرن شرح دهید.

درنهایت، پنج داده را به‌صورت تصادفی از مجموعهٔ آزمون انتخاب کنید و خروجی واقعی را با خروجی پیش‌بینی‌شده مقایسه کنید.

منابع

[1] https://github.com/MJAHMADEE/MachineLearning2024W