



دانشگاه صنعتی اصفهان
دانشکده مهندسی برق و کامپیوتر

تمرین برنامه‌نویسی سری اول
مبانی هوش محاسباتی
زمستان ۱۴۰۳

استاد درس: دکتر سمانه حسینی
دستیار آموزشی: امین کافی
موعد تحویل تکلیف: ۲۱ اسفند ۱۴۰۳

۱ پیش‌بینی بقای تایتانیک - ۶۰ نمره

در این تمرین، یک شبکه عصبی ساده را از ابتدا، بدون استفاده از فریمورک‌های یادگیری عمیق مانند TensorFlow یا PyTorch، پیاده‌سازی خواهید کرد. هدف، آموزش مدلی است که پیش‌بینی کند آیا یک مسافر از فاجعه‌ی تایتانیک جان سالم به در برده است یا خیر. معماری شبکه باید حداقل یک لایه‌ی پنهان با حداقل ۳ نورون داشته باشد.

۱.۱ مشخصات مجموعه‌داده

فایل‌های train.csv و test.csv در اختیار شما قرار داده شده‌اند. این مجموعه‌داده شامل ویژگی‌های زیر است:

□ **survival**: آیا مسافر از پس این فاجعه زنده مانده است؟ (۰ = خیر، ۱ = بله)

□ **pclass**: کلاس بلیط (۱ = درجه یک، ۲ = درجه دو، ۳ = درجه سه)؛ معیار وضعیت اقتصادی

□ **sex**: جنسیت

□ **age**: سن (مقادیر اعشاری کمتر از ۱ برای نوزادان است)

□ **sibsp**: تعداد خواهر و برادر یا همسر همراه

□ **parch**: تعداد والدین یا فرزندان همراه

□ **ticket**: شماره‌ی بلیط

□ **fare**: کرایه‌ی بلیط

□ **cabin**: شماره‌ی کابین

□ **embarked**: بندر سوار شدن

۲.۱ مراحل انجام

۱. بارگذاری داده‌ها: داده‌ها را با استفاده از کتابخانه‌هایی مانند pandas بارگذاری کنید.

۲. پیش‌پردازش داده‌ها (۱۵ نمره): درمورد ^۱EDA تحقیق کنید. سپس مراحل زیر را انجام دهید:

□ مقدارهای گم‌شده را حدس بزنید یا سطر متناظر را حذف کنید (سه بهترین روش امتیاز اضافه دریافت می‌کنند).

□ ویژگی‌های دسته‌ای را به مقدار عددی تبدیل کنید.

□ ویژگی‌های عددی را نرمال‌سازی کنید.

۳. جداسازی داده‌ها: داده‌های آموزشی را به:

□ X_{train} : ماتریس ویژگی‌ها

□ Y_{train} : برچسب‌های بقا (۰ یا ۱)

تقسیم کنید. همچنین X_{test} را برای داده‌های تست آماده کنید.

۴. پیاده‌سازی مدل شبکه‌ی عصبی (۳۰ نمره):

□ مقداردهی اولیه‌ی وزن‌ها و بایاس‌ها با مقادیر تصادفی کوچک

□ پیاده‌سازی تابع Forward Propagation برای محاسبه‌ی خروجی شبکه

□ پیاده‌سازی تابع Backpropagation برای محاسبه‌ی گرادیان‌ها

□ پیاده‌سازی به‌روزرسانی پارامترها با روش Gradient Descent

۵. تابع پیش‌بینی (۵ نمره): تابعی بنویسید که یک بردار ویژگی را دریافت کرده و پیش‌بینی کند که آیا مسافر زنده می‌ماند یا خیر.

۶. آموزش و ارزیابی مدل (۱۰ نمره): مدل‌تان را روی داده‌های آموزشی آموزش دهید و:

□ نمودار تغییرات تابع هزینه در هر epoch را رسم کنید.

□ پیش‌بینی‌های داده‌های تست را در فایل CSV ذخیره کنید.

۷. ارسال نتایج به **Kaggle**: خروجی مدل خود را در Kaggle Titanic Competition ثبت کنید و دقت مدل خود را گزارش کنید. امتیاز اضافه برای سه نفر برتر در دقت پیش‌بینی اختصاص داده می‌شود.

۳.۱ نکات مهم

□ تمامی کدها باید در یک فایل Jupyter Notebook با دسته‌بندی مناسب و توضیح کد قرار داده شود. از ذخیره خروجی هر سلول کد اطمینان حاصل کنید و فایل با خروجی هر سلول کد را آپلود کنید.

□ استفاده از vectorization اجباری است. استفاده از حلقه‌های تکرار در حالتی که vectorization ممکن است، موجب کسر امتیاز خواهد شد.

□ **EDA** اختیاری است اما سه راهکار برتر برای مدیریت داده‌های گمشده امتیاز اضافه دریافت می‌کنند.

□ استفاده از هر نوع معماری شبکه مجاز است، اما باید حداقل یک لایه مخفی با حداقل ۳ نورون داشته باشد.

□ تغییرات هزینه در طول آموزش باید نمایش داده شود (نمودار هزینه بر حسب epoch).

۲ پیش‌بینی نمره دانش‌آموزان - ۵۵ نمره

در این تمرین، شما باید از یک شبکه عصبی برای پیش‌بینی معدل دانش‌آموزان استفاده کنید. این شبکه عصبی باید با استفاده از فریمورک‌هایی مانند TensorFlow، PyTorch یا هر فریمورک دیگری که انتخاب می‌کنید پیاده‌سازی شود.

۱.۲ مشخصات مجموعه داده

فایل‌های train.csv و test.csv در اختیار شما قرار داده شده‌اند. این مجموعه داده شامل ویژگی‌های زیر است:

StudentID: شناسه‌ی یکتا برای هر دانش‌آموز (این ویژگی به عنوان ورودی در نظر گرفته نمی‌شود)

Age: سن دانش‌آموز

Gender: جنسیت دانش‌آموز (۰ برای مرد، ۱ برای زن)

Ethnicity: قومیت دانش‌آموز

ParentalEducation: سطح تحصیلات والدین دانش‌آموز

StudyTimeWeekly: ساعات مطالعه هفتگی (بین ۰ تا ۲۰ ساعت)

Absences: تعداد غیبت‌های دانش‌آموز در سال تحصیلی (بین ۰ تا ۳۰)

Tutoring: آیا دانش‌آموز از تدریس خصوصی بهره می‌برد؟ (۰ برای خیر، ۱ برای بله)

ParentalSupport: سطح حمایت والدین

Extracurricular: مشارکت در فعالیت‌های فوق برنامه (۰ برای خیر، ۱ برای بله)

Sports: مشارکت در برنامه‌های ورزشی (۰ برای خیر، ۱ برای بله)

Music: مشارکت در برنامه‌های موسیقی (۰ برای خیر، ۱ برای بله)

Volunteering: مشارکت در برنامه‌های داوطلبانه (۰ برای خیر، ۱ برای بله)

GPA: معدل دانش‌آموز (متغیر هدف، بین ۰ تا ۴)

۲.۲ مراحل انجام

۱. بارگذاری داده‌ها: داده‌ها را با استفاده از کتابخانه‌هایی مانند pandas بارگذاری کنید و ویژگی‌ها و هدف (معدل) را از هم تفکیک کنید.

۲. پیش‌پردازش داده‌ها (۱۵ نمره): داده‌ها را به گونه‌ای پردازش کنید که برای آموزش شبکه عصبی مناسب باشد. این مراحل شامل موارد زیر است:

□ مقدارهای گمشده را حدس بزنید یا سطر متناظر را حذف کنید.

□ ویژگی‌های دسته‌ای را به مقدار عددی تبدیل کنید.

□ ویژگی‌های عددی را نرمال‌سازی کنید.

۳. تعریف مدل (۱۵ نمره): یک مدل شبکه عصبی برای پیش‌بینی معدل دانش‌آموز تعریف کنید. این مدل باید شامل:

□ لایه ورودی که با تعداد ویژگی‌ها مطابقت دارد

□ یک یا چند لایه پنهان

□ لایه خروجی با یک نورون (چون معدل یک مقدار پیوسته است)

۴. آموزش مدل (۱۵ نمره): مدل خود را آموزش دهید و از موارد زیر استفاده کنید:

□ استفاده از تابع هزینه Mean Squared Error (MSE) برای رگرسیون

□ استفاده از یک بهینه‌ساز مانند Adam

□ نظارت بر هزینه‌های آموزش

مدل خود را برای تعداد معقولی از epoch ها آموزش دهید و تغییرات هزینه را در طول این دوره‌ها پیگیری کنید.

۵. رسم نمودار هزینه و اعتبارسنجی برحسب epoch (۵ نمره): بعد از آموزش، نمودارهای تغییرات هزینه و اعتبارسنجی را در مقابل تعداد epoch ها رسم کنید تا عملکرد مدل را مشاهده کنید.

۶. آزمون مدل (۵ نمره): مدل خود را روی داده‌های تست ارزیابی کرده و معیار R^2 را محاسبه کنید که به صورت زیر تعریف می‌شود:

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

که در آن y_i مقادیر واقعی، \hat{y}_i مقادیر پیش‌بینی شده و \bar{y} میانگین مقادیر واقعی است.

۳.۲ نکات مهم

□ تمامی کدها باید در یک فایل Jupyter Notebook با دسته‌بندی مناسب و توضیح کد قرار داده شود. از ذخیره خروجی هر سلول کد اطمینان حاصل کنید و فایل با خروجی هر سلول کد را آپلود کنید.

□ شما مجاز به آزمایش معماری‌های مختلف مدل، بهینه‌سازها و استراتژی‌های آموزشی هستید. حتماً از روش‌های تدریس شده در کلاس استفاده کنید تا بهترین عملکرد ممکن را به‌دست آورید.

□ تغییرات هزینه روی داده‌های آموزش و معیار R^2 روی داده‌های تست در طول آموزش باید نمایش داده شود.