**تمرین چهارم هوش مصنوعی و یادگیری ماشین (شبکه‌های عصبی)**

اردیبهشت ۱۴۰۲

تاریخ تحویل: ۵ / خرداد / ۱۴۰۲

---

هدف این تمرین آشنایی با شبکه های عصبی Multi-Layer Perceptron (MLP) است.

**بخش اول: مفاهیم پایه**

به پرسش‌های زیر پاسخ دهید:

- در مورد مشکل Imbalanced Datasets توضیح دهید. برای حل این مشکل از چه روش‌هایی استفاده می‌شود؟
- دو تابع هزینه Binary Cross Entropy و Categorical Cross Entropy را توضیح دهید. تفاوت این دو با هم چیست؟ آیا می‌توان از این توابع هزینه در یک مسئله Regression استفاده کرد؟
- آیا Accuracy به‌تنهایی معیار قابل اعتمادی از عملکرد یک مدل است؟ چرا؟
- نرمال‌سازی و استانداردسازی داده‌ها را تعریف کنید. اگر از این روش‌ها استفاده نکنیم چه مشکلی در روند آموزش ایجاد می‌شود؟

## بخش دوم: پیش‌بینی قیمت مسکن

از شبکه های عصبی می‌توان در قیمت‌گذاری و پیش‌بینی قیمت املاک استفاده کرد. دادگان houses_1.csv دارای ۲۱ ستون است که ۲۰ ستون به مشخصات خانه و تاریخ ساخت و قیمت‌گذاری و . . . اختصاص دارد و یک ستون (ستون سوم) قیمت خانه را بدست می‌دهد. می‌خواهیم در محیط Google Colab یک شبکه عصبی MLP پیاده‌سازی و تربیت کنیم که با دقت مناسبی قیمت خانه‌ها را پیش‌بینی کند. برای پیاده‌سازی کد خود از قسمت راهنمای حل تمرین کمک بگیرید.

## فراخوانی داده‌ها

در ابتدا با استفاده از ابزار مناسب دادگان را بخوانید و تعداد سطر و ستون آن را گزارش کنید. برای این دادگان ماتریس همبستگی (Correlation Matrix) را تشکیل دهید و توضیح دهید که این ماتریس چه اطلاعاتی به ما می‌دهد. کدام ویژگی همبستگی بیشتری با قیمت خانه‌ها دارد؟ نمودار توزیع قیمت را رسم کنید.

## پیش پردازش داده‌ها

در این مرحله داده‌ها را برای آموزش شبکه آماده می‌کنیم.

- ستون Date را به دو ستون ماه و سال تبدیل کنید و آن را از دادگان حذف کنید.
- ستون قیمت را به‌عنوان ستون خروجی (برچسب یا Y) درنظر بگیرید و بیشترین و کمترین قیمت را گزارش کنید. بقیه ستون‌ها را به‌عنوان داده‌های ورودی یا X درنظر بگیرید.
- داده‌ها را به‌نسبت ۸۰-۲۰ به داده‌های train و test تقسیم کنید.
- با کمک MinMaxScaler داده‌ها را scale کنید. توجه داشته باشید که در فرآیند scale کردن نباید از داده‌های آزمون استفاده کنید، چون باعث نشت اطلاعات (Data Leakage) می‌شود (این پدیده را توضیح دهید.)

## پیاده‌سازی مدل

### معماری شبکه

برای حل این مسئله از یک مدل MLP با ۲ لایه پنهان یا بیشتر استفاده می‌کنیم. تعداد لایه‌ها و تعداد نورون‌های هر لایه را به دلخواه خود انتخاب کنید اما توجه داشته باشد که مدل نباید بیش از حد سنگین یا سبک باشد.

## آموزش شبکه

شبکه را با ویژگی‌های زیر آموزش دهید:

- تابع هزینه: تابع هزینهٔ مناسب را با توجه به نوع مسئله انتخاب کنید.
- تابع فعال‌ساز لایه‌های پنهان: ReLU
- بهینه‌ساز: SGD
- سایز batch: ۶۴
- نرخ یادگیری: یکبار برابر ۰,۰۰۱ و بار دیگر برابر ۰,۱
- تعداد ایپاک[1]: یکبار برابر ۲۰ و بار دیگر برابر ۴۰۰۰

* برای پیاده‌سازی می‌توانید از Keras استفاده کنید. هم‌چنین برای رسم نمودارهای خواسته‌شده می‌توانید از کتاب‌خانه‌های matplotlib و seaborn استفاده کنید.

## خواسته‌ها

- برای چهار مدل train شده نمودارهای loss و validation loss را بر حسب شماره ایپاک رسم کنید و دقت آموزش و سنجش هر مدل را گزارش کنید.
- آیا تعداد ۲۰ ایپاک برای آموزش کافی بود؟ ۴۰۰۰ چطور؟ با استفاده از نمودار، مقدار حدودی تعداد ایپاک کافی برای هر حالت را گزارش کنید.
- بررسی کنید که تغییر اندازه نرخ یادگیری چه تاثیری در فرآیند آموزش داشته است.
- مدلی را که بیشترین دقت را داشته تعیین کنید و این‌بار این مدل را با تابع فعال‌ساز tanh برای لایه‌های پنهان آموزش دهید و دقت آموزش و سنجش آن را گزارش کنید.
- همین مدل را یکبار بار با batchsize برابر با ۱ و بار دیگر برابر با ۲۵۶ آموزش دهید. نمودارهای هزینه و دقت داده‌های آموزش و اعتبارسنجی را رسم کنید.

[1] epoch

- با استفاده از شبکه‌ای که بیشترین دقت را در بین شبکه‌های آموزش یافته داشته، خروجی داده‌های آزمون را پیش‌بینی کنید و ماتریس سردرگمی[2] را تشکیل دهید. برای این کار از کتاب‌خانه sklearn استفاده کنید.

**بخش سوم: دسته‌بندی دستگاه‌ها**

** این بخش در نرم‌افزار Matlab انجام می‌شود.

شبکه‌های عصبی ابزار مناسبی برای پایش سلامت و عیب‌یابی ماشین‌آلات صنعتی به شمار می‌آیند. در این بخش قصد داریم با استفاده از دادگان dataset_2.csv مدلی را پیاده کنیم که سالم بودن یا خراب بودن نوع خاصی از ماشین‌ابزار را تشخیص دهد. دادگان مربوطه در ۹ ستون ارائه شده است که ویژگی‌ها (ورودی‌ها) شامل دمای هوا، دمای فرآیند، سرعت زاویه‌ای، گشتاور و ساییدگی ابزار در ستون‌های چهارم تا هشتم، و خروجی (خراب بودن یا نبودن دستگاه یا علت خرابی آن) در ستون آخر آمده است.

## پیش پردازش داده‌ها

- ستون‌های چهارم تا هشتم را به عنوان داده یا X در نظر بگیرید.
- ستون آخر را به عنوان خروجی (برچسب یا Y) در نظر بگیرید. تمامی نمونه‌هایی را که با برچسب عدم خرابی مشخص شده‌اند با عدد صفر و باقی برچسب‌ها را، بدون توجه به نوع خرابی آنها با عدد ۱ جایگزین کنید تا برچسب‌ها حالت دو کلاسه داشته باشند.
- داده‌ها را به دو قسمت آموزش و آزمون تقسیم کنید. (به نسبت ۰٫۸ و ۰٫۲ )
- داده‌ها را استانداردسازی کنید.

## پیاده‌سازی و آموزش شبکه

یک شبکه با ویژگی‌های زیر پیاده‌سازی کنید:

- تعداد لایه‌های پنهان: ۱ لایه
- تابع فعال‌ساز لایه پنهان: ReLU

[2] Confusion Matrix

- تابع هزینه را با توجه به نوع مسئله انتخاب کنید و دلیل انتخاب خود را توضیح دهید.
- بهینه‌ساز: Levernberg-Marquardt
- نرخ یادگیری: ۰٫۱
- معیار توقف: max_fail=20 . توضیح دهید که این معیار نشان‌دهنده چیست.

## خواسته ها

- تعداد نورون‌های لایه پنهان را برابر با ۱، ۳۰ و ۵۰۰ قرار دهید و معیار RMSE را برای هر شبکه برای داده‌های آموزش و آزمون گزارش دهید. اندازه شبکه چه تاثیری بر روی یادگیری آن دارد؟
- : با همان پارامترهای ذکر شده در ابتدای مسئله، تعداد نورون‌ها را روی عدد ۲۰ ثابت نگه دارید و نمودار Performance را رسم کنید. تاثیر تغییر max_fail را روی آموزش شبکه بررسی کنید.

---

## راهنمایی ها

### پیاده‌سازی به کمک پایتون

- هنگام استفاده از google colab، برای خواندن داده‌ها می‌توانید ابتدا داده‌های خود را در مسیری دلخواه روی google drive قرار دهید و سپس کد زیر را در محیط colab بنویسید:

```
from google.colab import drive
drive.mount("/content/drive")
```

پس از آن که به کولب اجازه‌ی دسترسی به محتویات درایو را دادید، می‌توانید از مسیر زیر به فایل خود دسترسی داشته باشید:

```
file_dir = "/content/drive/MyDrive/[file path in drive]"
```

- میتوانید برای آموزش سریع تر شبکه، آن را روی GPU ران کنید.( در محیط colab ،Runtime را انتخاب

کرده و با کلیک روی گزینه‌ی Change runtime type نوع آن را روی GPU قرار دهید. اگر از کراس استفاده می‌کنید این کار برای اجرای آموزش شبکه روی GPU کافی است.) دقت کنید که کولب به مدت محدودی اجازه‌ی استفاده از GPU را به شما می‌دهد و پس از آن نیاز است مدتی صبر کنید تا مجدداً بتوانید از GPU روی کولب استفاده کنید. البته بار محاسباتی این تمرین آن قدر زیاد نیست که عدم استفاده از GPU برای شما مشکل‌ساز باشد.

- به‌طور کلی، هر زمان در کد نویسی دچار خطایی شدید که نحوه‌ی رفع آن را نمی‌دانستید ابتدا پیام آن خطا را کپی و در گوگل جست‌وجو کنید، چرا که به احتمال زیاد افرادی پیش از شما دچار مشکل مشابهی شده‌اند و پاسخ آن را برای استفادهٔ دیگران ارایه کرده‌اند.
- وبسایت کراس راهنمای بسیار کامل و قابل فهمی برای نحوه‌ی به کارگیری این کتابخانه دارد، حتما از آن استفاده کنید.
- در اینجا می‌توانید پیاده‌سازی یک شبکه‌ی عصبی ساده‌ی چندکلاسه را ببینید.

---

## پیاده‌سازی به کمک متلب

- دقت کنید که داده و کدی که اجرا می‌کنید حتماً در یک پوشه قرار داشته باشند.
- از راهنمای fitnet متلب و نحوه‌ی آموزش آن استفاده کنید. با فهمیدن نحوه‌ی استفاده از دستور آموزش شبکه، خواهید دانست که چه‌طور به یک training record برای شبکه‌ی خود دسترسی داشته باشید. با پرینت کردن این training record و زیرمجموعه‌های درون آن به اطلاعات جالبی دسترسی خواهید یافت که کار شما را برای پیدا کردن نحوه‌ی مقداردهی به پارامترهای مختلف راحت‌تر می‌کند.
- یک مثال پیاده‌سازی شبکه در متلب. توجه کنید که تمام بخش‌های این مثال مشابه خواسته‌های این مسئله نیست؛ بنابراین طبق خواسته عمل کنید و از این مثال تنها برای متوجه شدن روش کلی کار ایده بگیرید. به طور مثال، در این فیلم داده‌ها نرمال شده‌اند درحالی که از شما خواسته شده تا داده‌ها را استاندارد کنید. یا برای محاسبه‌ی معیار RMSE در این مثال ترم exp دخیل شده که در تمرین شما نیازی به این موضوع نیست و می‌توانید این معیار را به روش عادی محاسبه کنید.
- توجه داشته باشید که متلب از نام relu برای این تابع فعال‌ساز استفاده نمیکند و شما باید در راهنمای متلب به‌دنبال معادل آن بگردید. برای این کار doc transferFcn را در محیط Command Window متلب تایپ کنید و با توجه به تعاریف ارائه‌شده توسط متلب، تابع فعال‌ساز مناسب را انتخاب کنید. نحوه‌ی اختصاص دادن یک تابع فعال‌ساز به یک لایه در مثالی در راهنمای مربوطه ذکر شده است.

## چند تذکر

- یادگیری مفاهیمی که در تمرین مطرح شده و تدریس نشده اند با مطالعه شخصی ضروری است. برای این کار می توانید از لینک هایی که در تمرین معرفی شده است کمک بگیرید.
- تحویل گزارش برای این تمرین ضروری است و به تمرین بدون گزارش نمرهای تعلق نمی گیرد. حجم گزارش معیاری برای ارزیابی نخواهد بود اما توضیحات کدهای زده شده در گزارش این تمرین بسیار مفید خواهد بود.
- در فرایند ارزیابی گزارش، کدهای شما لزوما اجرا نخواهند شد. بنابراین همهٔ نتایج و تحلیل های خود را به طور کامل ارائه کنید
- به منابعی که از آنها استفاده کرده اید ارجاع دهید.
- شباهت بیش از حد گزارش و کدها باعث صفر شدن نمرهٔ تمرین خواهد شد. همچنین گزارش هایی که در آنها از کدهای آماده استفاده شده باشد پذیرفته نخواهند شد.
- گزارش شما باید به صورت تایپ شده و با فرمت pdf ارائه شود و کدهایی که به همراه گزارش تحویل می دهید باید قابل اجرا باشند. تمامی فایل های لازم را در یک فایل zip یا rar قرار داده و ارسال کنید.
- اگر پاسخ پرسش های خود را در منابع معرفی‌شده نیافتید می توانید از دستیاران آموزشی کمک بگیرید:


سورنا سعیدی :

Email: Ssuorena@gmail.com
Telegram: @Ssuorena

محمدسعید ظفری :

Email: Mohsaeedzaf@gmail.com
Telegram: @The0M

نویدرضا قنبری :

Email: nrghanbari97@gmail.com
Telegram: @NovidR