

تمرین پنجم هوش مصنوعی و یادگیری ماشین (یادگیری عمیق)

خرداد ۱۴۰۲

هدف این تمرین آشنایی با شبکه‌های عصبی عمیق و مقایسه کارایی آن‌ها با شبکه‌های MLP است.

بخش اول: مفاهیم پایه

به سوالات زیر پاسخ دهید:

الف) برای هر یک از کاربردهای دسته‌بندی (classification) و پیش‌بینی (regression) دو نمونه تابع زیان (loss function) معرفی کرده و عملکرد آن‌ها را بصورت کوتاه توضیح دهید.

ب) عملکرد لایه Batch Normalization را در شبکه‌های عمیق توضیح دهید و اهمیت این لایه را در شبکه‌های عصبی عمیق را توضیح دهید.

پ) در مورد عملکرد لایه‌های Layer Normalization و Group Normalization و تفاوت آن‌ها با لایه Batch Normalization توضیح دهید. (برای آشنایی با عملکرد این لایه‌ها می‌توانید از این [لینک](#) کمک بگیرید.)

بخش دوم: تشخیص (دسته‌بندی) تابلوهای راهنمایی و رانندگی

یکی از کاربردهای مهم شبکه‌های عصبی، کمک به سامانه رانندگی خودکار (auto pilot) در خودروهای خودران^۱ از طریق تشخیص تابلوهای راهنمایی و رانندگی است که سامانه را قادر می‌سازد تا در رویارویی با تابلوهای مختلف، واکنش مناسبی از خود نشان دهد. در این بخش می‌خواهیم با طراحی دو نوع شبکه عصبی و تربیت آن‌ها به کمک دادگان ترافیکی موجود، تابلوهای راهنمایی و رانندگی را تشخیص دهیم.

ابتدا دادگان مورد نظر [German Traffic Sign Recognition Benchmark (GTSRB)] را از این [لینک](#) دانلود کنید. سپس یک بار با استفاده از یک شبکه MLP و بار دیگر با استفاده از یک شبکه پیچشی (CNN) و با بهره‌گیری از کتابخانه‌های مختلف این داده‌ها را در ۴۳ گروه دسته‌بندی کنید

¹ Autonomous vehicles

پیش پردازش داده ها:

در این دادگان تصاویر سه کاناله RGB با ابعاد مختلف از تابلوهای راهنمایی و رانندگی موجود است. ابتدا با استفاده از دستور `resize` ابعاد همه تصاویر را 30×30 قرار دهید. و مقدار هر پیکسل را به ۲۵۵ (بیشترین مقدار ممکن برای هر پیکسل) تقسیم کنید. سپس داده‌های موجود در پوشه `train` را بصورت تصادفی به نسبت ۰/۹ برای آموزش (`training`) و ۰/۱ برای ارزیابی (`validation`) جدا کنید و داده‌های موجود در پوشه `test` را نیز برای آزمون عملکرد شبکه‌های خود در نظر بگیرید.

پیاده سازی شبکه ها

الف - شبکه MLP

معماری شبکه

در طراحی شبکه MLP خود می‌توانید از هر تعداد لایه پنهان و هر تعداد نرون در لایه‌ها استفاده کنید، به شرط آنکه شبکه بدست آمده بیش از حد سنگین و تربیت آن بیش از حد زمان‌بر و یا دقت نهایی آن از حد مطلوب کمتر نشود. همچنین در صورتی که سرعت آموزش شبکه کم بود (تنها در این بخش) می‌توانید به جای استفاده از همه داده‌های آموزش، از زیرمجموعه‌ای از این داده‌ها استفاده کنید. دقت کنید که در این صورت باید این زیرمجموعه را به صورت تصادفی انتخاب کنید و این زیرمجموعه باید شامل داده‌های تمامی دسته‌ها باشد. در گزارش خود توضیح دهید که در آموزش این شبکه از چه تابع زیانی استفاده کرده‌اید و دلیل این تصمیم را نیز مشخصا بیان کنید. همچنین تابع فعال‌سازی لایه آخر را نیز به دلخواه (و البته با ذکر دلیل) مشخص نمایید.

آموزش شبکه

برای آموزش شبکه MLP می‌توانید از هر تابع زیان و هر الگوریتم بهینه‌سازی استفاده کنید. نرخ یادگیری را (در این بخش) با آزمون و خطا و تعیین کنید و توضیح دهید نرخ یادگیری نامناسب چه تاثیری بر روند یادگیری خواهد داشت.

ب - شبکه پیچشی

معماری شبکه

در طراحی شبکه پیچشی خود می‌توانید از هر معماری دلخواه استفاده کنید، به شرط آنکه شبکه بدست آمده بیش از حد سنگین و تربیت آن بیش از حد زمان‌بر و یا دقت نهایی آن از حد مطلوب کمتر نشود. در گزارش خود توضیح دهید که

در آموزش این شبکه از چه تابع زیبایی استفاده کرده‌اید و دلیل این تصمیم را نیز مشخصا بیان کنید. همچنین تابع فعال‌سازی لایه آخر را نیز به دلخواه (و البته با ذکر دلیل) مشخص نمایید.

آموزش شبکه

الف) شبکه پیچشی را در حالت‌های زیر آموزش دهید. در هر حالت می‌توانید تغییراتی را که باعث بهبود عملکرد شبکه شده است به حالت بعدی انتقال دهید (انجام این کار اجباری نیست).

- استفاده از pooling های مختلف در ساختار شبکه (دو حالت)
- استفاده یا عدم استفاده از dropout در آموزش شبکه (دو حالت)
- استفاده از بهینه سازهای Adam و Gradient Descent (دو حالت)
- استفاده از توابع فعال‌سازی sigmoid ، ReLU و tanh (سه حالت)

ب) ابتدا در مورد روش Data Augmentation و اهمیت آن در یادگیری عمیق توضیح دهید. سپس ساختاری را که در بخش قبل به بیشترین دقت دست یافته این بار به کمک Data Augmentation آموزش دهید (حداقل از سه نوع نگاشت تصویری استفاده کنید). توضیح دهید که با توجه به ماهیت مسئله، چه نگاشتهایی مجاز هستند (برخی از نگاشتها باعث می‌شوند عکس یک تابلو در دسته نادرست طبقه بندی شود). برای پیاده سازی Data Augmentation می‌توانید از تابع ImageDataGenerator در کتابخانه `keras.preprocessing.image` استفاده کنید. برای پیاده سازی این بخش می‌توانید از این [لینک](#) کمک بگیرید.

پ) نتایج بخش الف و ب را با نتایج بدست آمده از شبکه ی MLP به اختصار مقایسه کنید.

از هر ساختار انتخابی مدل خود یک پلات در گزارش خود بیاورید و هر بار که مدل را تغییر می‌دهید پلات بروز شده را نیز در گزارش خود بیاورید. برای پلات کردن مدل می‌توانید از دستور زیر استفاده کنید.

```
keras.utils.plot_model(MODEL, to_file=path)
```

بس از هر بار آموزش، نمودار خطا و دقت را برحسب ایپاک برای داده‌های آموزش و ارزیابی رسم کنید. همچنین خطا و دقت شبکه بدست آمده را نیز بر روی داده‌های آزمون و ارزیابی گزارش کرده و ماتریس سردرگمی^۲ را نیز هر بار برای داده‌های آزمون و ارزیابی رسم کنید.

² Confusion matrix

- 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 لزوماً اجرا نخواهند شد. بنابراین همه 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 نمره 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 معرفی شده نیافتید می توانید از دستیاران آموزشی کمک بگیرید:

: سورنا سعیدی

Email: Ssuorena@gmail.com

Telegram: @Ssuorena

: محمدسعید ظفری

Email: Mohsaheedzaf@gmail.com

Telegram: @The0M

: نویدرضا قنبری

Email: navidrezaghanbari@gmail.com

Telegram: @NovidR