تمرین شبکه‌های عصبی

در این تمرین با استفاده از کتابخانه تنسورفلو شبکه‌های عصبی را پیاده‌سازی می‌کنیم.

تمرین اول (رگرسیون):

۱. یک دیتاست دلخواه مسئله رگرسیون را انتخاب کنید (دیتاست‌های استفاده شده در کلاس و تمرینات قبلی مجاز نیست، برای انتخاب دیتاست می‌توانید از گیت‌هاب یا وبسایت Kaggle استفاده کنید.).

۲. دیتا را به خوبی پردازش کنید. (تمیز کردن دیتا، نرمالیزه کردن با استفاده از تکنیک مناسب، جدا کردن بخش آموزش و تست)

۳. ابتدا یک شبکه عصبی با ساختار ساده پیاده‌سازی کرده و فرایند لرنینگ را انجام دهید. سپس با محاسبه مقدار هزینه بر روی دیتاست train و test وضعیت مدل از نظر underfit، overfit یا just right را تعیین کنید.

۴. در صورتی که مدل‌تان دچار underfit شده است، هر بار مدل خود را پیچیده‌تر کرده و تست‌های لازم را بگیرید. آنقدر فرایند گام‌به‌گام پیچیده‌تر کردن مدل را انجام دهید تا در نهایت به overfit برسید. اکنون با کمی جست‌وجو سعی کنید ترم‌های regularization که قبلا با آنها آشنا شده‌اید را بر روی شبکه خود اعمال کرده و مجدد آن‌را ترین کنید. وضعیت مدل را از نظر underfit، overfit یا just right بررسی کرده و در صورتی که همچنان مدل در شرایط overfit است، مدل را کمی ساده‌تر کنید.

۵. در نهایت مدل مناسب را پیاده‌سازی کرده و نتایج آزمایش عملکرد شبکه را گزارش کنید.

۶. پس از اینکه مدل مناسب را پیدا کردید، اکنون وقت آن است که کمی با توابع فعال‌ساز لایه‌های پنهان بازی کنید. در گام اول مدل زمان ترین یک شبکه را به ازای فعال‌سازهای sigmoid، ReLU و tanh (تانژانت هایپربولیک) مقایسه کنید. در گام دوم از نظر عملکرد آن‌ها را مقایسه کرده و نتایج را توجیه کنید.

تمرین دوم (دسته‌بندی چندکلاسه):

در این تمرین می‌خواهیم بین سه علامت @، $ و & دسته‌بندی انجام دهیم.

گام اول: (جمع‌آوری دیتا)

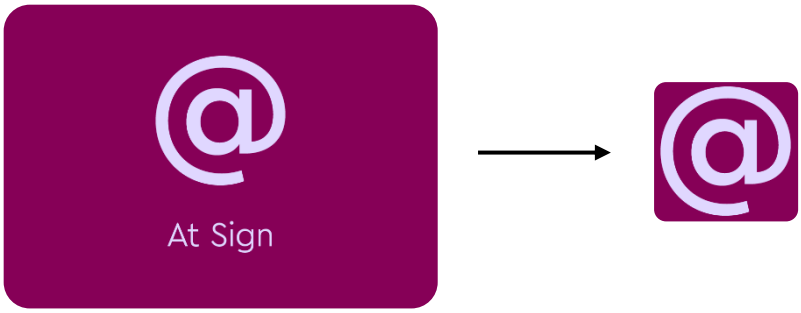
۱. ابتدا باید از هر کدام از علامت‌های فوق که به ترتیب با اسامی at sign، dollar sign و ampersand شناخته می‌شوند دست‌کم ۳۰۰ عکس دانلود کنید.
تذکر: عکس‌های مربوط به هر کدام از علائم فوق را در یک پوشه جداگانه ذخیره کنید.

شاید با خود بگویید چه کار سخت و زمان‌بری! نکته‌ای که وجود دارد عمده کار یک مهندس یادگیری ماشین جمع‌اوری دیتا و تمیز کردن آن و ... است و گرنه که فرایند لرنینگ با چند خط کد توسط تنسورفلو انجام می‌شود!

برای دانلود عکس‌ها در کمترین زمان ممکن، ویدئو زیر را تماشا کنید.

How to download all images in any webpage at one click

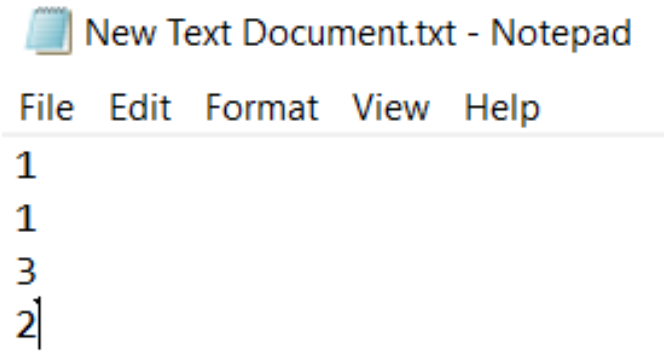
۲. در قسمت بعدی باید عکس‌های دانلود شده را گزینش کنید دست کم ۳۰۰ عکس مناسب را برای هر کدام از علائم فوق انتخاب کنید.
حواستان باشد که تقریبا تمامی عکس‌های گزینش‌شده نیاز به کراپ کردن دارند، برای مثال:



تذکر: به هیچ عنوان در یک عکس نباید همزمان دو تا از این علائم وجود داشته باشد. همچنین سعی کنید عکس‌هایی که خیلی بی‌کیفیت هستند را انتخاب نکنید.

۳. پس از اینکه عکس‌ها را به خوبی فیلتر کردید. اکنون بایستی در سه پوشه جداگانه از هر کدام از علائم دست‌کم ۳۰۰ عکس کراپ‌شده داشته باشیم. اکنون نوبت آن رسیده که یک پوشه جدید ساخته و فرایندی را پیاده‌سازی کنیم که با ترتیب تصادفی عکس‌ها را در این پوشه قرار دهد. برای این‌کار می‌توانید از کتابخانه os در پایتون استفاده کنید. برای اطلاعات تکمیلی لینک زیر را بررسی کنید.
Randomly shuffling photos in a folder : r/learnpython
۴. پس از انجام مرحله ۳ باید یک پوشه داشته باشیم که تمام عکس‌های جمع‌آوری شده به صورت تصادفی در آن نام‌گذاری شده باشند. در این مرحله باید یک فایل txt. باز کرده و به ترتیب اسامی که عکس‌ها در این پوشه ذخیره شده‌اند، برچسب مناسب را بنویسید. برچسب علامت @ برابر ۱، برچسب علامت $ برابر ۲ و برچسب علامت & برابر ۳ در نظر گرفته شود.

برای مثال اگر دو عکس اول موجود در پوشه علامت @، عکس سوم علامت & و عکس چهارم علامت $ بود، فایل txt. باید به صورت زیر باشد:



بدین ترتیب در صورتی که در مجموع ۹۰۰ عکس مناسب جمع‌آوری کرده‌اید، فایل txt. شما باید ۹۰۰ خط داشته باشد.

۵. در این مرحله نوبت آن است که عکس‌ها و برچسب‌های نظیر را بخوانیم. برای این‌کار از کتابخانه‌های OpenCV و os استفاده می‌کنیم. نحوه کار باید به نحوی باشد که ابتدا هر کدام از تصاویر موجود در پوشه نهایی توسط OpenCV خوانده شده، سپس به صورت grayscale در آمده و به سایز 50×50 در بیاید. با کمی جست‌وجو در اینرنت دستورات فوق را می‌توانید پیدا کنید. در نهایت هر عکس grayscale تغییر سایز یافته که یک ماتریس 50×50 است درون یک آرایه نامپای ذخیره می‌شود. برای مثال اگر تعداد کل عکس‌های جمع‌آوری شده شما ۹۰۰ است، در در نهایت آرایه نامپای باید ابعادی به صورت (900,50,50) داشته باشد.

در مرحله بعد باید با استفاده از کتابخانه os فایل txt. که برچسب‌ها را داخل آن نوشتیم خوانده شده و درون یک آرایه نامپای تک بعدی ذخیره شود. در صورتی که ۹۰۰ عکس داریم، شکل این آرایه به صورت (900,) خواهد بود.

۶. در این مرحله باید آرایه نامپای تصویر که شامل ۹۰۰ تصویر ۵۰ در ۵۰ است را به دو قسمت ترین و تست جدا کنیم. توجه داشته باشید که این امر باید کاملا تصادفی اتفاق بیفتد. همچنین حواستان باشد که آرایه نامپای برچسب‌ها را هم باید متناسب با آرایه تصویر به دو قسمت ترین و تست جدا کنید. یک روش برای انجام این کار آن است که به صورت تصادفی تعدادی index ساخته و آن‌ها را از آرایه تصویر و برچسب جدا کنیم و به عنوان دیتاست تست در نظر بگیریم. سپس این index ها را از آرایه اصلی تصویر و برچسب پاک کرده تا قسمت ترین بدست آید. نحوه پیاده‌سازی این موضوع کاملا دلخواه است!

۷. پس از اینکه جداسازی ترین و تست انجام شد، نرمالیزه کردن آن‌ها فراموش نشود!

۸. اکنون دیتای ما برای انجام فرانید لرنینگ آماده است.

گام دوم:

یک شبکه عصبی مناسب برای این مسئله ساخته و عملکرد آن را بسنجید. در صورتی که underfit یا overfit داد به درستی ساختار شبکه عصبی را تغییر دهید. در نهایت بهترین نتایج را گزارش کنید.

تذکر: تحلیل نتایج قسمت‌های مختلف تمرین اول و دوم باید به صورت تایپی با توضیحات کامل آورده شود. به تمرینات بدون گزارش نمره‌ای تعلق نخواهد گرفت.