

بسم الله الرحمن الرحيم

دانشگاه علم و صنعت ایران



زمستان ۱۳۹۹

تحويل: جمعه ۱۲ دی




تمرین سری چهاردهم

مبانی بینایی کامپیوتر

توضیحات

- در ابتدای هر سوال یک نشانگر وجود دارد که مشخص می‌کند آن سوال تشریحی است یا برنامه‌نویسی؛ اگر نشانگر  یعنی سوال تشریحی است و اگر نشانگر  باشد یعنی سوال یک سوال برنامه‌نویسی است.
- برای حل سوالات برنامه‌نویسی فایل HW14.ipynb را تا انتها دنبال کنید.
- خواهشمندیم اگر مطلبی را از اینترنت استفاده کردید حتما منبع آن را مشخص کنید.
- لطفا فایل HW14.ipynb نهایی خود را همراه با فایل pdf مربوط به پاسخ سوالات تشریحی فشرده کرده و ارسال کنید.
- **لطفا در فایل گزارش خود به صورت دقیق، زمانی را که برای حل این تمرین صرف کردید، گزارش کنید.**

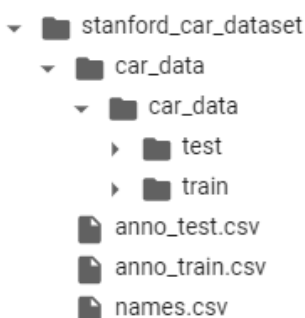
سوالات

۱.  الف) علت به وجود آمدن مشکل underfitting در شبکه چیست و چه زمانی رخ می‌دهد؟ (۵ نمره)
ب) مشکل بیش برازش (overfitting) چیست؟ چه راه‌هایی برای رفع آن پیشنهاد می‌کنید؟ (۵ نمره)
۲.  استفاده از شبکه‌های pre-train یکی از روش‌های موثر برای مقداردهی اولیه پارامترهای یک شبکه است. در مورد نحوه استفاده از این نوع یادگیری و مزایای آن، توضیح دهید. (۱۰ نمره)
۳.  اگر یک تصویر ۳ کاناله با ابعاد ۲۸ در ۲۸ داشته باشیم، تعداد پارامترها را برای یک لایه کانوولوشنی با تعداد ۱۲۸ فیلتر ۵×۵ را در حالتی که از کانوولوشن ساده و کانوولوشن با عمق جداپذیر (Depthwise)

Separable Convolution) استفاده می‌شود (به ازای `depth_multiplier` برابر با ۲)، به دست آورده و با هم مقایسه کنید. (۱۰ نمره)

۴. 📖 (۷۰ نمره) در این سوال می‌خواهیم یک مسئله دسته‌بندی را با استفاده از تکنیک‌های داده‌افزایی و انتقال یادگیری انجام دهیم.

در این سوال از مجموعه داده [cars](#) استفاده می‌کنیم که شامل ۱۶۱۸۵ تصویر است و ۱۹۶ کلاس دارد. تعداد تصاویر آموزشی در این مجموعه داده برابر با ۸۱۴۴ و تعداد تصاویر آزمون برابر با ۸۰۴۱ است. برای دانلود این مجموعه داده کافی است مراحل که در فایل HW14.ipynb ذکر شده است انجام دهید تا مجموعه داده در Google Drive شما ذخیره گردد. پس از انجام این مراحل پوشه‌ای به نام `stanford_car_dataset` در درایو شما ایجاد شده است که محتویات آن باید مطابق شکل زیر باشد:



داخل هر کدام از پوشه‌های `train` و `test`، ۱۹۶ پوشه دیگر وجود دارد که داخل هر کدام از آن‌ها تصاویر مربوط به یکی از ۱۹۶ کلاس است و نام این پوشه‌ها نشان‌دهنده‌ی آن کلاس است. برای مثال در پوشه `BMW 1 Series Convertible 2012` تصاویر مربوط به کلاس `BMW 1 Series Convertible 2012` وجود دارد. در ادامه توضیح خواهیم داد که چگونه این ساختار ذخیره‌سازی به ما کمک خواهد کرد تا مجموعه داده را به مدل یادگیری عمیق ورودی بدهیم.

معمولاً برای آموزش مدل‌های یادگیری عمیق، از مجموعه داده‌های نسبتاً بزرگی استفاده می‌شود که حتی پیشرفته‌ترین سخت‌افزارها، حافظه کافی برای پردازش داده‌ها به صورت یکجا و یکپارچه را ندارند. به همین دلیل است که ما باید راه‌های دیگری برای انجام کارآمد آن پیدا کنیم. در ادامه قصد داریم به شما نشان دهیم که چگونه مجموعه داده را در چندین هسته و در زمان اجرا تولید کرده و بلافاصله آن را به مدل یادگیری عمیق خود بدهیم. بدین منظور کلاسی به نام [ImageDataGenerator](#) در `keras` پیاده سازی شده است که همزمان با تولید `batch` ای از تصاویر در زمان اجرا، می‌تواند اعمال مختلف داده‌افزایی مانند `flipping`، `rotation` و ... را انجام دهد. باتوجه به ساختار ذخیره‌سازی مجموعه داده `cars` که قبلاً

ذکر شد؛ در این مسئله می‌توانیم به راحتی از تابع `flow_from_directory` برای تولید تصاویر استفاده کنیم. در غیر این صورت باید خودمان با توجه به نیاز مسئله کلاس `generator` را پیاده‌سازی می‌کردیم (اگر علاقه‌مند به نوشتن `custom generator` هستید؛ از این [لینک](#) استفاده نمایید). برای آشنایی بیشتر با `ImageDataGenerator` و نحوه استفاده از آن برای داده‌افزایی، از مثال موجود در [لینک](#) استفاده نمایید.

الف) در این قسمت شما باید برای هر کدام از داده‌های آموزشی و داده‌های تست، به ترتیب دو `generator` با استفاده از `ImageDataGenerator` و `flow_from_directory` با نام‌های `train_datagen` و `test_datagen` بسازید. سپس از `resnet50` (موجود در کراس) با [وزن‌های تصادفی](#) استفاده کنید و یک مدل برای این مسئله طراحی کنید. پس از آموزش مدل، آن را بر روی داده‌های تست ارزیابی کنید.

ب) در این قسمت، از مدلی که در قسمت الف طراحی کرده‌اید؛ استفاده کنید و این بار مدل را با داده‌افزایی آموزش دهید. بنابراین برای داده‌های آموزشی یک `generator` با استفاده از `ImageDataGenerator` و `flow_from_directory` با نام `train_datagen_aug` بسازید که برای داده‌های آموزشی نیز داده‌افزایی انجام دهد. پس از آموزش مدل، آن را بر روی داده‌های تست ارزیابی کنید. نتیجه حاصل از این قسمت را با قسمت الف مقایسه نمایید.

پ) در این قسمت از شبکه از قبل آموزش‌دیده `ResNet50` بر روی مجموعه داده `Imagenet` استفاده کنید ([لینک](#)) و پس از آن یک دسته‌بند قرار دهید. موقع آموزش یک بار تمامی وزن‌های شبکه `resnet50` را `freeze` کنید و مابقی وزن‌ها را آموزش دهید. در بار دیگر بگذارید تا وزن‌های شبکه `resnet50` نیز آموزش ببینند (fine tuning). برای آموزش این مدل در هر دو حالت از `train_datagen_aug` (`generator` با داده‌افزایی) استفاده نمایید. پس از آموزش مدل، آن را بر روی داده‌های تست ارزیابی کنید. نتیجه حاصل از این قسمت را با قسمت الف و ب مقایسه نمایید.

چند نکته برای طراحی مدل یادگیری عمیق:

- برای آموزش مدل از تابع [fit](#) استفاده نمایید و برخلاف قبل، به جای ورودی دادن داده‌های آموزشی و برچسب آن، کافی است `generator` داده‌های آموزشی را به تابع `fit` ورودی بدهید. نیازی به استفاده از `fit_generator` در نسخه جدید `tensorflow` نیست.

- برای ارزیابی مدل نیز کافی است از تابع [evaluate](#) استفاده نمائید و generator داده‌های تست را به آن ورودی دهید.
- موفق باشید.