



مهلت تحويل: دوشنبه ۲۰ دی ماه ۱۴۰۰

## مقدمه:

### کتابخانه Keras و TensorFlow

شبکه‌های عصبی یکی از قدرتمندترین ساختارهای یادگیری ماشین است که در سال‌های اخیر با توجه به افزایش قدرت محاسباتی پردازنده‌ها کاربردهای بسیاری در حوزه‌های مختلف علمی و صنعتی پیدا کرده است. از مزایای این الگوریتم‌ها آن است که امکان ساخت هر تابع مشتق پذیر دلخواهی را با تنها استفاده از دو لایه‌ی مخفی از نورون‌ها ممکن می‌سازد. بنابراین امکان پیاده‌سازی و یادگیری توابع غیر خطی نیز با کمک آن‌ها ممکن است. با توجه به پیشرفت‌های اخیر این ساختار و کاربرد روز افرون آن در صنعت، فریم ورک‌های بسیار قدرتمندی برای سهولت در ایجاد شبکه‌های عصبی بسیار پیچیده و با کارایی بسیار بالا عرضه شده است. در این پژوهه قصد داریم با کتابخانه TensorFlow و با استفاده از رابط سطح بالای Keras برخی از مسائل مربوط به شبکه‌های عصبی را بررسی نماییم.

## تعریف مسئله

در این مسئله قصد داریم با استفاده از شبکه‌های عصبی بر اساس تصویر یک حیوان، نوع آن حیوان را تشخیص دهیم. توصیه می‌شود به خاطر حجم نسبتاً زیاد مجموعه داده این تمرین را با استفاده از google colab انجام دهید تا نیازی به دانلود مجموعه داده‌ها روی سیستم خود نداشته باشید.

## معرفی مجموعه داده

مجموعه داده حیات وحش Oregon Wildlife یک مجموعه داده تصویر حیوانات وحشی است. شما در این تمرین با ۴ کلاس گوزن، راکون، کلاغ و عقاب از این مجموعه داده به آموزش یک شبکه عصبی می‌پردازید.



محنتویات پوشیده مجموعه داده به صورت زیر است. که در هر پوشیده تعدادی عکس در اختیار شما قرار گرفته است.

dataset

└── train

- ├── elk
- ├── raven
- ├── bald\_eagle
- └── raccoon

└── test

- ├── elk
- ├── raven
- ├── bald\_eagle
- └── raccoon

حتماً توجه داشته باشید که ابتدا مجموعه train را با نسبت منطقی به دو بخش train و validation تقسیم کنید و به آموزش پردازید. در انتها نیز بر روی داده‌های پوشیده تست باید شبکه‌ی خود را امتحان کنید. دقت داشته باشید که بجز بخش آخر از بخش استفاده نمی‌شود و از مجموعه‌های train و validation که تقسیم کرده‌اید استفاده خواهد کرد.

## نکات پیاده‌سازی

در این پروژه باید به کمک رابط Keras یک شبکه Forward Feed ایجاد کنید که حداقل شامل دو لایه مخفی باشد (با در نظر گرفتن لایه‌های ورودی و خروجی (softmax) باید شبکه شما حداقل دارای 4 لایه باشد). و دقت کنید به هیچ عنوان نباید از لایه‌های شبکه‌های Convolution و Recurrent استفاده کنید.

## فاز اول: بررسی و پیش‌پردازش داده

- با استفاده از رابط Keras داده‌های را بخوانید. سپس تصاویر را grayscale کنید.
- تعداد داده‌های خوانده شده و تعداد کلاس‌های موجود را چاپ کنید.
- تعداد تصاویر هر کلاس را محاسبه کنید و برای آن نمودار میله‌ای رسم کنید.
- از هر کلاس موجود فقط یک تصویر همراه با نام کلاس آن را نشان دهید.
- بخش train دیتابست را به دو بخش train و validation و با مقیاس مناسب تقسیم کنید.
- برای برچسب‌ها One Hot Encoding انجام دهید و دلیل این کار را شرح دهید.

## فاز دوم: طراحی شبکه عصبی

یک شبکه‌ی عصبی با شرایط زیر طراحی کنید:

1. بهینه‌ساز شما باید از نوع SGD باشد.
2. مقدار learning rate باید 0.01 باشد.
3. تعداد epoch ها باید 10 باشد.
4. اندازه batch\_size باید 32 باشد.
5. تابع فعال‌سازی تمام لایه‌ها غیر از لایه آخر باید relu باشد.

نکته ۱: دقت کنید batch size در تمامی مراحل بعدی 32 است مگر آنکه به طور مستقیم ذکر شود که آن را تغییر دهید.

نکته ۲: پس از هر مرحله از تمرین (بعد از اتمام تمامی epoch ها) موارد زیر باید چاپ شود:

- نمودار مقدار loss در هر epoch

- نمودار مقدار accuracy در هر epoch

نکته ۳: پس از هر مرحله از تمرین (بعد از اتمام تمامی epoch ها) موارد زیر باید روی داده های تمرین و ولیدیشن چاپ شود:

- معیارهای recall

- معیارهای precision

- معیارهای F1

پس از ایجاد ساختار شبکه عصبی مناسب با توجه به شرایط داده شده تعداد پارامترهای هر لایه را چاپ کرده و اعداد چاپ شده را توجیه کنید (برای چاپ تعداد پارامترها میتوانید از رابط Keras استفاده کنید و نیازی به پیاده سازی توسط خودتان نیست).

### فاز سوم: طبقه بندی داده ها

در این بخش باید شبکه عصبی ایجاد شده را روی داده ها تمرین دهید.

در این بخش باید داده های را که از فایل ها دریافت می کنید را نرمال کنید. برای این کار هر پیکسل داده خوانده شده باید به 255 تقسیم گردد.

در این قسمت باید ساختار شبکه عصبی خود را به گونه ای تغییر دهید که بیشترین مقدار F1 برای نتیجه روی داده ویلدیشن ایجاد شود. برای بقیه بخش ها از همین ساختار بهینه استفاده نمایید.

برای بدست آوردن یک شبکه عصبی مناسب می توانید تابع های فعال ساز دیگری مثل tanh و sigmoid و Leaky Relu را هم بررسی کنید ولی الزاماً ندارد.

نکته: در فاز قبل با بسیاری از مسائل و پارامترهایی که خود توانایی ساخت و تغییر آن را داشتید آشنا شده اید و آن پارامترها در این فاز بررسی نمی گردند و فقط مسائلی که قادر به پیاده سازی آنها نبودید در ادامه به کمک Keras بررسی می شوند.

### قسمت اول) تاثیر optimizer

در این بخش تاثیر optimizer ها را بررسی می کنیم.

- تحقیق کنید که momentum چیست و چرا استفاده از آن مفید است؟

- مدل را با momentum های 0.5 و 0.9 تمرین دهید و نتیجه را توضیح دهید.

- آیا همواره با افزایش momentum نتیجه بهبود میابد؟ توضیح دهید.

- یکی از مشهور ترین optimizer هایی که در فرایند تمرین دادن مدلها بسیار استفاده می شود Adam است. مدل خود را با استفاده از Adam تمرین دهید و نتیجه را با SGD قبل مقایسه کنید. در قسمتهای بعدی از Adam استفاده کنید (از نرخ اولیه پیشفرض برای تمرین به کمک Adam استفاده کنید که مقدار آن 0.001 است).

### قسمت دوم) تاثیر epoch

- در این بخش تاثیر تعداد epoch ها را بررسی می کنیم.
- مدل را به تعداد 20 epoch تمرین دهید. برای سولات بعدی تعداد epoch ها را برابر 20 در نظر بگیرید.
  - چرا معمولا شبکه های عصبی را در چندین epoch تمرین می دهند؟ آیا در همه مسائل نیاز به آن است که شبکه عصبی در چندین epoch تمرین نماید؟
  - آموزش شبکه های عصبی معمولا تا زمانی ادامه پیدا میکند که epoch overfitting شروع شود. حال تعداد epoch هایی که شبکه را با آن آموزش می دهید را تغییر دهید تا زمانی که overfit رخ بدهد. در این حالت نمودار های accuracy و loss را بر روی داده های آموزش و تست رسم کنید و نتایج را تحلیل کنید.
  - سعی کنید با استفاده از معروفی معیاری در زمان تمرین شبکه عصبی یا تعداد epoch بالا تمرین شبکه را وقتی ممکن است overfit کند، متوقف کنید.
  - آیا همواره استفاده از تعداد epoch های بیشتر برای تمرین مفید است؟ اگر جواب شما منفی است، راه حل های مقابله با اتفاق نامطلوبی که روی میدهد را بیان کنید.

### قسمت سوم) تاثیر loss function

- در این بخش تاثیر تعداد تابع Loss های مختلف را بررسی می کنیم.
- شبکه عصبی را با استفاده از تابع Loss مشهور به MSE تمرین دهید و نتیجه را با categorical cross entropy مقایسه نمایید.
  - چرا استفاده از تابعی مثل MSE برای classification مناسب نیست؟ چه زمانی از این تابع استفاده می کنیم؟

### قسمت چهارم) تاثیر regularization

- در این بخش روش های regularization در شبکه های عصبی را بررسی می کنیم.
- با استفاده از رابط keras برای هر لایه regularization L2 را با مقدار 0.0001 اعمال کنید و نتیجه را مشاهده کنید و توضیح دهید.

- با استفاده از رابط keras برای هر لایه `dropout` با مقدار 0.1 قرار دهید و نتیجه را مشاهده نمایید و توضیح دهید.

پس از بدست آوردن بهترین طراحی شبکه خود، شبکه را بر روی داده‌های `test` امتحان کنید و مقدار F1 و اکیورسی را گزارش کنید.

همچنین ۱۰ عکس که به درستی پیش‌بینی کرده اید و ۱۰ عکس که به اشتباه پیش‌بینی کرده اید را نمایش دهید. برای تصاویر که شبکه شما دچار اشتباه شده است، تحلیل کنید علت این اشتباه چه موردی می‌تواند باشد.

### فاز پنجم: ترسیم داده‌های با بعد کاهش‌یافته (امتیازی)

دو کاربرد عملی جالب `encoder`، عبارت است از `data denoising` و کاهش ابعاد برای ترسیم داده‌ها است.

در این بخش قصد داریم به کمک یک شبکه‌ی عصبی Feed Forward، بعد داده‌های ورودی را از 10000 به 2 کاهش دهیم و داده‌های با بعد کاهش‌یافته را در فضای دو بعدی ترسیم کنیم. برای این کار، شبکه‌ای با معماری زیر پیاده‌سازی کنید:

- لایه آخر شبکه (لایه خروجی) 5 نورون داشته باشد (به تعداد کلاسها).
- لایه یکی مانده به آخر شبکه 2 نورون داشته باشد (جهت ترسیم خروجی آن در فضای دو بعدی).
- لایه نخست شبکه 10000 نورون داشته باشد (به تعداد فضای ویژگی داده‌های ورودی).
- تعداد نورون‌های سایر لایه‌های شبکه از سمت ورودی به سمت خروجی کاهشی باشد.

بخش نخست شبکه فوق (شامل تمامی لایه‌ها بجز لایه آخر) را اصطلاحاً (Encoder) و وظیفه‌ی آن کاهش بعد ویژگی داده‌های ورودی است. در نهایت، داده‌های کاهش‌یافته در لایه آخر طبقه‌بندی می‌شوند.

قصد داریم تا داده‌های کاش‌یافته‌ی خروجی لایه‌ی یکی مانده به آخر شبکه فوق را در فضای دو بعدی ترسیم نماییم. برای این کار، مراحل زیر را دنبال کنید:

- لازم است تا در ابتدا خروجی لایه‌ی یکی مانده به آخر کلاس شبکه دخیره شود.
- در گام دوم شبکه‌ای با معماری مطرح شده طراحی کنید و آموزش دهید و با پارامترهای موردنظرتان آموزش دهید تا زمانی که به دقت مناسبی برسید.
- نهایتاً خروجی لایه‌ی یکی مانده به آخر شبکه‌ی آموزش داده شده را به ازای داده‌های آموزش و تست ترسیم کنید. نقاط مربوط به هر کلاس را با یک رنگ بخصوص نمایش دهید.
- در نهایت، با توجه به نمودارهای بدست آمده، نتیجه را تفسیر کنید و عملکرد شبکه را شرح دهید. با توجه به تصویر و دقت بدست آمده از شبکه، به نظر شما شبکه‌تان در تفکیک کدام دسته از داده‌ها از هم دچار مشکل می‌شود و کدام داده‌ها را به راحتی از هم تفکیک می‌کند؟

## منابع

برای یادگیری شروع کار با Keras می‌توانید از این [لینک](#) کمک بگیرید.

برای آموزش شیوه استفاده از Google Colab می‌توانید از این [لینک](#) استفاده نمایید.

در این [لینک](#) درباره Optimizer Adam مطالعه کنید.

در این [لینک](#) درباره روشهای مختلف regularization مطالعه کنید.

در این [لینک](#) درباره ترسیم داده‌های با بعد کاهش یافته مطالعه کنید.

## نکات پایانی

- دقت کنید که هدف پروژه تحلیل نتایج است؛ بنابراین از ابزارهای تحلیل داده بطور مثال نمودارها استفاده کنید و توضیحات مربوط به هر بخش از پروژه را بطور خلاصه و در عین حال مفید در گزارش خود ذکر کنید.
- نتایج و گزارش خود را در یک فایل فشرده با عنوان AI-CA5-<#SID>.zip. تحویل دهید. محتویات پوشه باید شامل فایل jupyter-notebook، خروجی html و فایل‌های مورد نیاز برای اجرای آن باشد. توضیح و نمایش خروجی‌های خواسته شده بخشنامه این تمرین را تشکیل می‌دهد. از نمایش درست خروجی‌های مورد نیاز در فایل html مطمئن شوید.
- همچین پیشنهاد می‌شود با توجه به حجم بالای مجموعه داده و همچنین حتماً از Google Colab استفاده نمایید.
- توجه داشته باشید که علاوه بر ارسال فایل‌های پروژه، این پروژه به صورت حضوری نیز تحویل گرفته خواهد شد. بنابراین تمام بخش‌های پروژه باید قابلیت اجرای مجدد در زمان تحویل حضوری را داشته باشند. همچنین در صورت عدم حضور در تحویل حضوری نمره ای دریافت نخواهد کرد.
- در صورتی که سوالی در مورد پروژه داشتید بهتر است در فروم درس مطرح کنید تا بقیه از آن استفاده کنند؛ در غیر این صورت توسط ایمیل با طراحان در ارتباط باشید.
- هدف از تمرین، یادگیری شماست. لطفاً تمرین را خودتان انجام دهید.

موفق باشید.