

به نام خدا



دانشگاه تهران

پردیس دانشکده‌های فنی

دانشکده مهندسی برق و کامپیوتر



## هوش مصنوعی

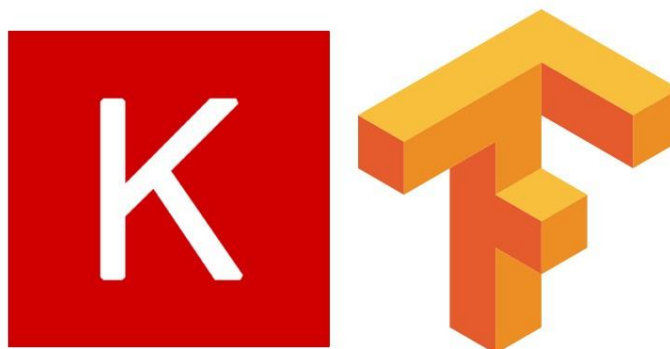
پروژه پنجم (فاز دوم)

بررسی برخی از مسائل شبکه‌های عصبی به کمک Tensorflow

مهلت ارسال تا ۲۰ دی ماه

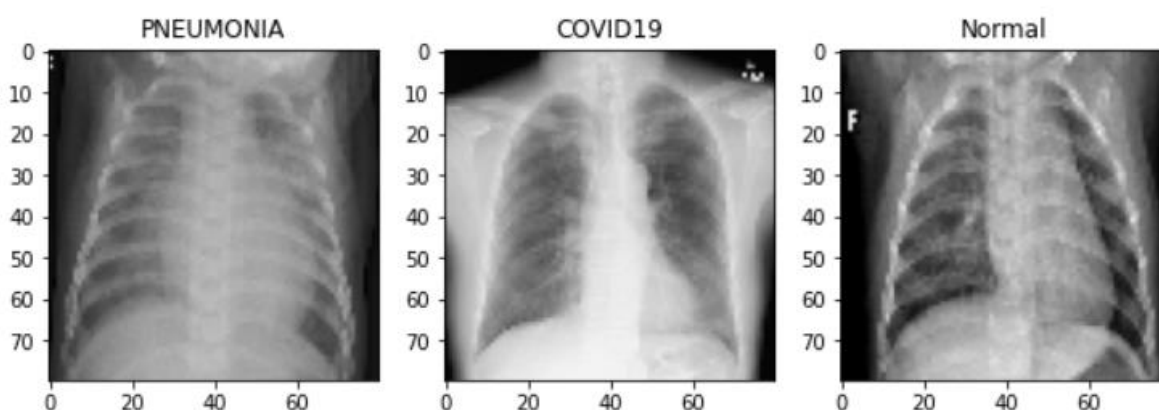
طراح پروژه: علیرضا سالمی

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 لایه مخفی از نرون‌ها ممکن می‌سازند. بنابراین امکان پیاده‌سازی و یادگیری توابع غیر خطی نیز با کمک آن‌ها ممکن است. با توجه به پیشرفت‌های اخیر این ساختار و کاربرد روز افزون آن در صنعت، فریمورک‌های بسیار قدرتمندی برای سهولت در ایجاد شبکه‌های عصبی بسیار پیچیده و با کارایی بسیار بالا عرضه شده است. در این پروژه قصد داریم با کتابخانه TensorFlow و با استفاده از رابط سطح بالای Keras برخی از مسائل مربوط به شبکه‌های عصبی را بررسی نماییم.



## تعریف مسأله

در این مسئله قصد داریم با استفاده از شبکه‌های عصبی وجود بیماری Covid19 یا ذات‌الریه را در تصاویر سیتی‌اسکن مربوط به بیمارانی که به مرکز درمانی مراجعه کرده‌اند تشخیص دهیم. برای این مسئله از مجموعه داده‌ای که در [این](#) لینک آمده است. این مجموعه داده شامل ۲ بخش تمرین و تست می‌شود که برای سادگی کار به طور پیشفرض در دو پوشه جداگانه قرار داده شده‌اند. در هر بخش تعدادی تصویر از ریه بیمارانی ذات‌الریه و کرونایی و همچنین انسان‌های سالم قرار داده شده است. توصیه می‌شود به خاطر حجم بالای مجموع داده این تمرین را حتما در سایت google colab انجام دهید!



## نکات پیاده‌سازی

- در این پروژه باید به کمک رابط Keras یک شبکه Feed Forward ایجاد کنید که حداقل شامل دو لایه مخفی باشد (یعنی با احتساب لایه ورودی و لایه خروجی یا همان softmax باید شبکه شما حداقل دارای ۴ لایه باشد).
- برای یادگیری شروع کار با Keras می‌توانید از [این](#) لینک کمک بگیرید.
- استفاده از jupyter notebook در این پروژه الزامی است. همچنین پیشنهاد می‌کنیم با توجه به حجم بالای مجموعه داده و همچنین اجبار در استفاده از GPU حتماً از Google Colab استفاده نمایید. برای آموزش شیوه استفاده از Google Colab می‌توانید از [این](#) لینک استفاده نمایید.
- در این پروژه باید حتماً از GPU یا در صورتی که می‌توانید از TPU استفاده نمایید. دقت کنید که فقط از طریق Google Colab در دسترس شما است. برای اجرای مدل‌های شما روی TPU ابتدا لازم است یک

کد برای وصل شدن به TPU اجرا نمایید که راهنمای آن در [این](#) لینک آمده است. برای استفاده از GPU نیازی به استفاده از کدی برای مقدار دهی اولیه نیست و فقط وجود Tensorflow-GPU کافی است.

## مواردی که باید پیاده‌سازی کنید

۱. با استفاده از رابط Keras داده‌های مربوط به تمرین و تست را بخوانید و تصاویر را grayscale کرده و به اندازه  $80 \times 80$  تغییر دهید و همچنین تعداد داده‌های خوانده شده و تعداد دسته‌های موجود را نیز چاپ کنید. دقت کنید batch size در تمامی مراحل بعدی ۳۲ است مگر آنکه به طور مستقیم ذکر شود که آن را تغییر دهید (دقت کنید در اینجا نباید مقدار پیکسل‌های تصویر را نرمال کنید).

۲. موارد زیر را پیاده‌سازی کنید.

۱. از هر دسته موجود فقط یک تصویر و نام دسته را همانند تصویر بخش تعریف مسئله نمایش دهید.

۲. تعداد تصاویر هر دسته را محاسبه کنید و برای آن نمودار میله‌ای رسم کنید.

- از این قسمت به بعد موارد زیر را برای شبکه خود در نظر بگیرید:
  - از تابع categorical cross entropy به عنوان تابع loss استفاده کنید.
  - بهینه‌ساز شما باید از نوع SGD و نرخ یادگیری 0.01 باشد.
  - سایز batchها باید برابر با ۳۲ باشد.
  - تعداد دوره‌های آموزش مدل باید ۱۰ دور باشد.
  - تابع فعال‌سازی تمام لایه‌ها غیر از لایه آخر باید relu باشد.
- پس از هر مرحله از تمرین (بعد از اتمام تمام epochها) موارد زیر باید چاپ شود:
  - معیارهای recall و precision و F1 روی داده تمرین و داده تست
  - نمودار مقدار loss در هر epoch

۳. ساختاری برای شبکه عصبی با استفاده از Keras مطابق با شرایط توضیح داده شده در نکات پیاده‌سازی تعریف کنید و 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 پیاده‌سازی توسط خودتان نیست).

۴. در این بخش باید شبکه عصبی ایجاد شده را روی داده‌ها تمرین دهید.

- I. برای تمرین از تابع فعال‌سازی relu استفاده نمایید و نتیجه به دست آمده را توجیه کنید.
- II. از برای تمرین از تابع فعال‌سازی tanh استفاده نمایید و نتیجه به دست آمده را توجیه کنید.
- III. علت مشاهدات متفاوت درباره دو تابع استفاده شده و نکته مشترک تمرین هر دو مدل چیست؟  
راهکار شما برای حل این مشکل چیست؟

۵. در این بخش باید داده‌ای را که از فایل‌ها دریافت می‌کنید را نرمال کنید. برای این کار هر پیکسل از داده خوانده شده باید به 255 تقسیم گردد. در این قسمت باید ساختار شبکه عصبی خود را به گونه‌ای تغییر دهید که بیشترین مقدار F1 برای نتیجه روی داده تست ایجاد شود. برای بقیه بخش‌ها از همین ساختار بهینه استفاده نمایید.

در فاز قبل با بسیاری از مسائل و پارامترهایی که خود توانایی ساخت و تغییر آن را داشتید آشنا شده‌اید و آن پارامترها در این فاز بررسی نمی‌گردند و فقط مسائلی که قادر به پیاده‌سازی آن‌ها نبودید در ادامه به کمک Keras بررسی می‌شوند.

۶. در این بخش با optimizerها بیشتر آشنا می‌شویم.

- I. تحقیق کنید که momentum چیست و چرا استفاده از آن مفید است؟
- II. مدل را با momentum های 0.5 و 0.9 و 0.99 تمرین دهید و نتیجه را توضیح دهید.
- III. آیا همواره استفاده از momentum بسیار زیاد مفید است؟ توضیح دهید.
- IV. یکی از مشهورترین optimizerهایی که در فرایند تمرین دادن مدل‌ها بسیار استفاده می‌شود Adam است. در باره این optimizer در [این](#) لینک مطالعه کنید. سپس 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).

۷. در این بخش قصد داریم تاثیر تعداد epochها روی مدل را بررسی نماییم.

- I. مدل را به تعداد ۲۰ epoch تمرین دهید. برای سولات بعدی تعداد epochها را برابر ۲۰ در نظر بگیرید.

II. چرا معمولا شبکه‌های عصبی را در چندین epoch تمرین می‌دهند؟ آیا در همه مسائل نیاز به آن است که شبکه عصبی در چندین epoch تمرین نماید؟

III. آیا همواره استفاده از تعداد epoch‌های بیشتر برای تمرین مفید است؟ اگر جواب شما منفی است، راه حل‌های مقابله با اتفاق نامطلوبی که روی می‌دهد را بیان کنید.

۸. در این بخش قصد داریم تابع Loss‌های مختلف را بررسی کنیم.

I. شبکه عصبی را با استفاده از تابع Loss مشهور به MSE تمرین دهید و نتیجه را با categorical cross entropy مقایسه نمایید.

II. چرا استفاده از تابعی مثل MSE برای classification مناسب نیست؟ چه زمانی از این تابع استفاده می‌کنیم؟

۹. در این بخش با روش‌های regularization در شبکه‌های عصبی آشنا می‌شویم.

I. با استفاده از [این](#) در باره روش‌های مختلف regularization مطالعه کنید.

II. 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.

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

- موعده تحویل غیرحضورى تا پایان روز ۲۰ دى ماه مى‌باشد.
- تمامى نتایج باید در یک فایل فشرده با عنوان AI-CA5-<#STID>.zip تحویل داده شود. این فایل باید شامل موارد زیر باشد:
  - یک پوشه به نام Code شامل کدهای تمام قسمت‌هایی از تمرین که پیاده‌سازی نموده‌اید.
  - گزارش پروژه با فرمت PDF و شامل شرح تمامى کارهای انجام شده، نتایج به دست آمده و تحلیل‌ها و بررسی‌های خواسته شده در صورت پروژه.
  - در صورتی که از Jupyter Notebook استفاده مى‌کنید نیازی به ارسال جداگانه کدها و گزارش نیست و هردو را مى‌توانید در یک فایل Notebook ارائه دهید. حتما خروجی html فایل Notebook خود را نیز همراه فایل Notebook ارسال کنید.
- توجه داشته باشید که علاوه بر ارسال فایل‌های پروژه، این پروژه به صورت حضورى نیز تحویل گرفته خواهد شد. بنابراین تمام بخش‌های پروژه باید قابلیت اجرای مجدد در زمان تحویل حضورى را داشته باشند. همچنین در صورت عدم حضور در تحویل حضورى نمره‌ای دریافت نخواهید کرد.
- هیچگونه شباهتى در انجام این پروژه بین افراد مختلف پذیرفته نمى‌شود. در صورت کشف هرگونه تقلب برای همه افراد متقلب نمره ۱۰۰- در نظر گرفته مى‌شود.
- استفاده از مراجع با ارجاع به آنها بلامانع است. اما در صورتی که گزارش شما ترجمه عینى از آنها باشد، یا از گزارش افراد دیگر استفاده کرده باشید کار شما تقلب محسوب مى‌شود.
- در صورتی که سوالی در مورد پروژه داشتید بهتر است در فروم درس مطرح کنید تا بقیه از آن استفاده کننده، در غیر این صورت به طراحان پروژه ایمیل بزنید یا حضورى از یکى از آنها بپرسید.

[alirezasailemi7@gamil.com](mailto:alirezasailemi7@gamil.com)

[ae.561999@gmail.com](mailto:ae.561999@gmail.com)

[ahatefi@ut.ac.ir](mailto:ahatefi@ut.ac.ir)

موفق باشید!