علیرضا خیاطیان

۱۴۰۰/۹/۷

تمرین پنجم درس شبکه های عصبی

آماده سازی داده ها

در این تمرین قصد داریم با استفاده از شبکه‌های کانولوشنی، تصاویر غذاهای مختلف را دسته‌بندی نماییم. ابتدا مجموعه داده‌های food101 رو از tensorflow بارگذاری می‌کنیم. این مجموعه شامل ۱۰۱ کلاس یا ۱۰۱ نوع غذای مختلف است که برای هر کلاس ۱۰۰۰ داده وجود داره که ۷۵۰ تای آن داده‌های آموزش و ۲۵۰ تای آن داده‌های تست می‌باشند. تصاویر آموزشی شفاف نیستند و دارای نویز می‌باشند. این کار به طور عامدانه صورت گرفته تا از overfitting مدل جلوگیری شود. اما داده‌های تست بدون نویز و clean هستند. علاوه بر داده ها یک پوشه meta نیز در کنار داده‌ها وجود دارد. این پوشه شامل ۳ فایل train.txt، test.txt و class.txt می‌باشد که به ترتیب مشخص می‌کند کدام تصاویر به مجموعه آموزش، مجموعه تست و کدام کلاس تعلق دارد. در اینجا تعدادی از تصاویر را به صورت رندوم انتخاب می‌کنیم و با برچسب آن به نمایش می‌گذاریم.

شکل ۱: برخی از تصاویر مجموعه داده‌ها به همراه برچسب آن‌ها

در ادامه به کمک محتویات پوشه meta داده‌ها را به مجموعه های آموزش و تست برای هر کلاس تقسیم می‌کنیم. ما اکنون داده‌های آموزش و تست را آماده کرده‌ایم، اما برای آزمایش معماری‌های مختلف، کار روی کل داده‌ها با ۱۰۱ کلاس زمان و محاسبات زیادی می‌برد. برای انجام آزمایش مورد نظر، مجموعه داده را به ۱۰ کلاس محدود می‌کنم و train_min و test_mini را ایجاد می‌کنیم. توجه شود که برای انتخاب ۱۰ کلاس از ۱۰۱ کلاس جایگشت‌های مختلفی وجود دارد که هر یک از این جایگشت‌ها بسته به این که ۱۰ کلاس انتخاب شده با هم چقدر همبستگی دارند روی مقدار accuracy تاثیر می‌گذارد. این مقدار می‌تواند حدود ۱۰ درصد باشد. در اینجا ما ۱۰ کلاس را به صورت رندوم انتخاب می‌کنیم تا مدل را نسبت به کلاس‌های خاص بایاس نکنیم.

انتقال یادگیری

استفاده مجدد از مدلی که قبلاً آموخته شده در یک مسئله جدید به عنوان یادگیری انتقال شناخته می شود. در حال حاضر در یادگیری عمیق بسیار محبوب است زیرا می تواند شبکه های عصبی عمیق را با مقدار کمی داده آموزش دهد. برای انتقال یادگیری در شبکه های عصبی ۳ روش وجود دارد. اول آنکه از شبکه پیش اموزش دیده شده به طور کامل استفاده کنیم و ورودی را به آن داده و خروجی را از آن بدون هیچ آموزش مجدد دریافت کنیم. این حالت زمانی امکان پذیر است که خروجی های شبکه از پیش آموزش دیده شده با خروجی مورد نظر ما یکسان باشد. اما هنگامی که ما یک دسته بندی متفاوت از دسته بندی که این شبکه اموزش دیده انجام می‌دهد در نظر داشته باشیم باید از روش دوم استفاده کنیم. در روش دوم قسمت دسته بندی شبکه از پیش اموزش دیده را جدا می‌کنیم و classifier خود را جایگزین آن می‌کنیم در واقع از شبکه از پیش آموزش دیده به عنوان feature extractor استفاده می‌کنیم. در این روش وزن های قسمت استخراج ویژگی را بدون تغییر نگاه می‌داریم و شبکه را آموزش می‌دهیم تا وزن‌های دسته بند جدید آموزش ببینند. اما در روش سوم می‌خواهیم قسمتد استخراج ویژگی ها را نیز آموزش داده تا برای مسئله ما مجدد آموزش ببینند و اصطلاحا fine tune شود. ذکر این نکته لازم است که برای tuning ابتدا باید classifier را با وزن های ثابت برای استخراج کننده ویژگی اموزش دهیم و پس از آموزش آن بار دیگر برای tuning آموزش دهیم. تصمیم برای tuning زمانی گرفته می‌شود که بخواهیم جزییات بیشتری از مسئله جدید پیدا کنیم و در واقع مدل از پیش آموزش دیده شده را برای مسئله خود اختصاصی کنیم.

نقطه شروع tuning

برای tuning کافی است تنها لایه های آخر قسمت استخراج کننده ویژگی را آموزش دهیم و لایه های ابتدایی را ثابت در نظر بگیریم. در واقع لایه های اولیه ویژگی های اولیه همچون خطوط افق و عمودی را در تصویر مشخص می‌کنند که در اغلب مسایل موجود است اما لایه های انتهایی به استخراج جزییات مسئله می‌پردازند که مناسب است با tuning آنها را برای مسئله خود آموزش دهیم. در واقع می‌توانیم ۱۰ تا ۲۰ درصد از لایه های آخر قسمت استخراج کننده ویژگی‌ها را tune کنیم. تا در واقع مدل از پیش آموزش دیده شده را برای مسئله خود اختصاصی کنیم. دقت شود که نرخ یادگیری در tuning بسیار مهم است و باید با نرخ یادگیری کم این موضوع انجام شود که در غیر این صورت خیلی سریع مدل overfit می‌شود.

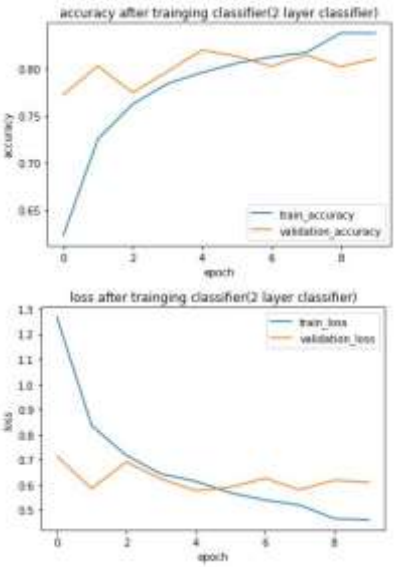
طراحی و آموزش شبکه

برای آموزش شبکه از انتقال یادگیری استفاده می‌کنیم. برای این کار از شبکه از پیش آموزش دیده DenseNet169 که روی دیتاست ImageNet آموزش دیده است استفاده می‌کنیم. به کمک ImageDataGenerator ابعاد تصاویر را ۱۲۸*۱۲۸ می-کنیم. ابتدا قسمت classifier مدل را جدا کرده و با classifier خود جایگزین می‌کنیم. برای آموزش قسمت دسته بند تمام وزن های قسمت استخراج کننده ویژگی مدل را ثابت نگه می‌داریم و آموزش را شروع می‌کنیم. سپس برای fine tuning در مرحله دوم

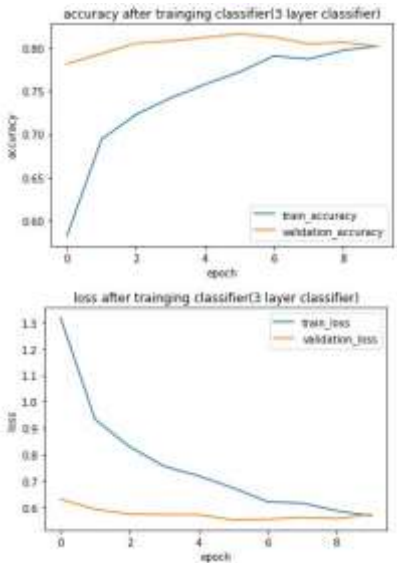
آموزش بخشی از قسمت انتهایی قسمت استخراج کننده را از حالت وزن ثابت درآورده و شبکه را آموزش می‌دهیم. تمامی این سه مرحله را برای تمام آزمایش‌های خواسته شده انجام می‌دهیم.

تعداد لایه‌های دسته بند

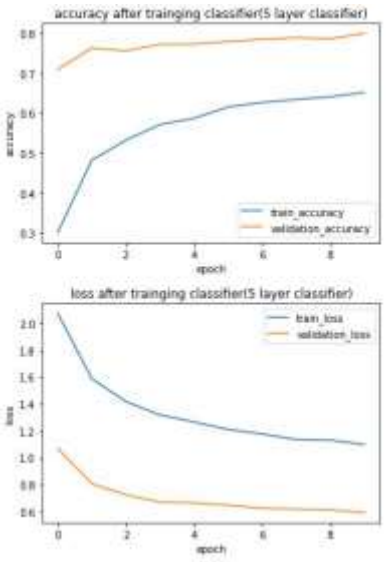
برای این آزمایش ما سه مدل معماری را به کار بردیم در معماری اول دسته بند شامل دو لایه با ۱۰۲۴ و ۱۰ نورون بود. در معماری دوم شامل سه لایه با ۱۰۲۴، ۵۱۲ و ۱۰ نورون بود. همچنین معماری سوم شامل ۵ لایه با ۱۰۲۴، ۵۱۲، ۶۴، ۳۲ و ۱۰ بود. با توجه به اینکه ما ۱۰ کلاس را در نظر گرفته ایم و آخرین لایه دسته بند باید حتما ۱۰ لایه باشد اگر تعداد نورون های دستبند به ترتیب کاهش پیدا کنند مناسب تر از حالتی است که همه لایه ها دارای یک تعداد نورون مثل هم باشند. در شکل زیر نتایج ۳ مدل را پس از آموزش برای هر سه مدل نشان داده شده است در این قسمت تنها دسته بند آموزش می‌بینند و وزن های مدل پایه ثابت در نظر گرفته می‌شوند. در ادامه هر سه مدل را از لایه ۵۰۰ تا آخر مدل مبنا tune می‌کنیم نتایج آن را مشاهده می‌کنید.



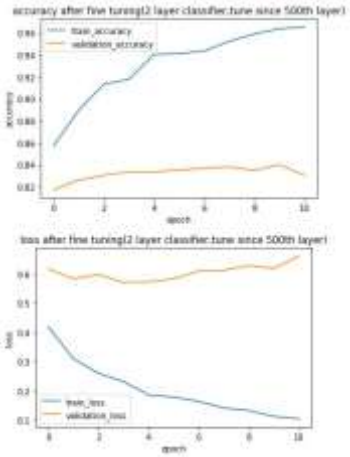
شکل ۱: نمودارهای accuracy و loss برای آموزش دسته بند با ۲ لایه



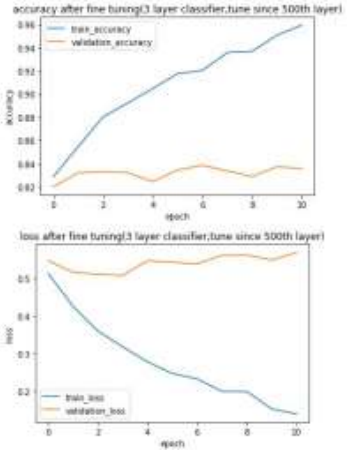
شکل ۲: نمودارهای accuracy و loss برای آموزش دسته بند با ۳ لایه



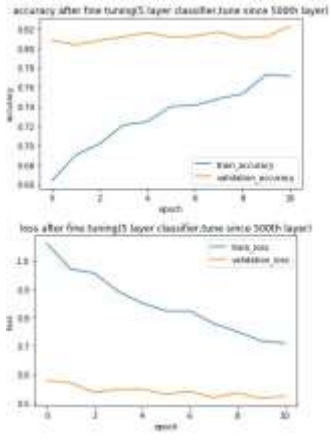
شکل ۳: نمودارهای accuracy و loss برای آموزش دسته بند با ۵ لایه



شکل ۴: نمودارهای accuracy و loss برای tuning ۲ لایه از لایه ۵۰۰



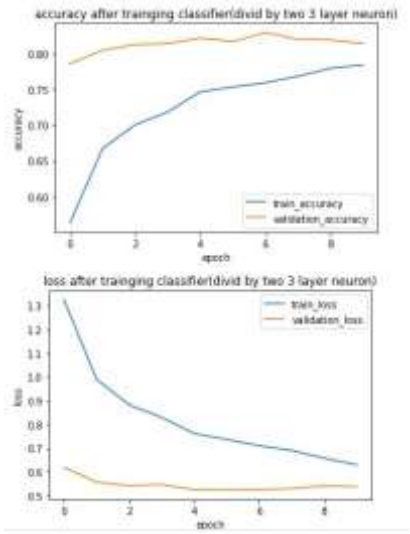
شکل ۵: نمودارهای accuracy و loss برای tuning ۳ لایه از لایه ۵۰۰



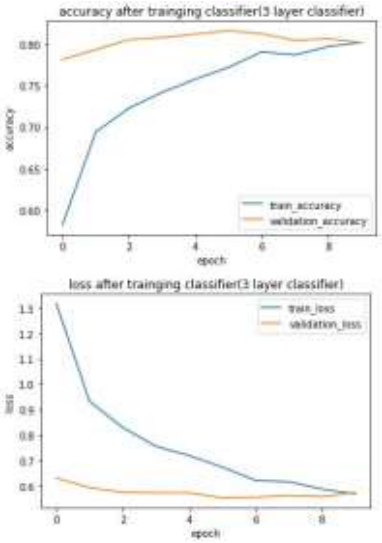
شکل ۶: نمودارهای accuracy و loss برای tuning ۵ لایه از لایه ۵۰۰

شبکه با دسته بند سه لایه هم از نظر مقدار حداکثر مقدار val_accuracy و هم از نظر generalization عمکرد بهتری دارد در واقع در بخش آموزش دسته بند فاصله دو نمودار دقت داده‌های ارزیابی و آموزش کم تر بوده است.

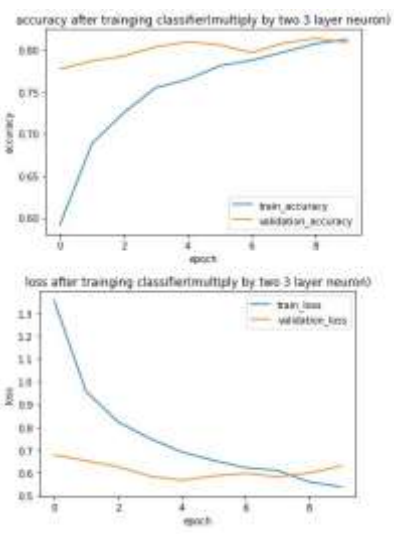
تعداد نورون‌هابرای بررسی تعداد نورون‌های مختلف اینگونه عمل می‌کنیم که تعداد نورون های هر لایه در مدل متنخب سه لایه در قسمت قبل را یک بار تقسیم بر دو می‌کنیم و یک بار هم ضرب در دو می‌کنیم و سه مدل حاصل را مقایسه می‌کنیم. بنابراین مدل اول دارای لایه ها با



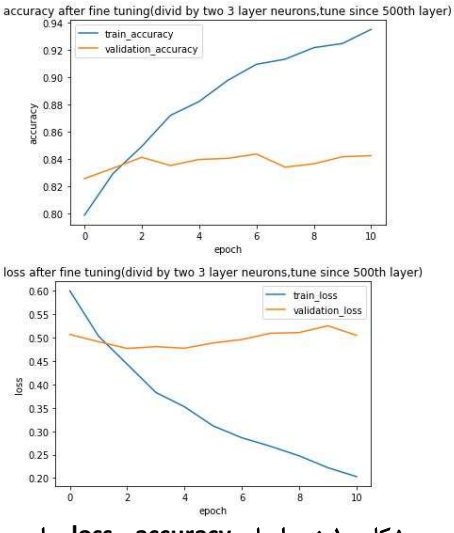
شکل ۷: نمودارهای accuracy و loss برای آموزش دسته بند با ۵۱۲، ۲۵۶ و ۱۰ نورون



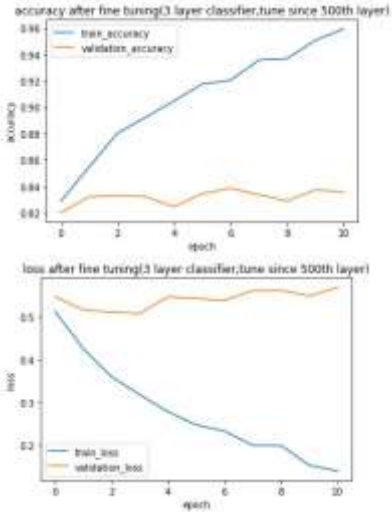
شکل ۸: نمودارهای accuracy و loss برای آموزش دسته بند با ۱۰۲۴، ۵۱۲ و ۱۰ نورون



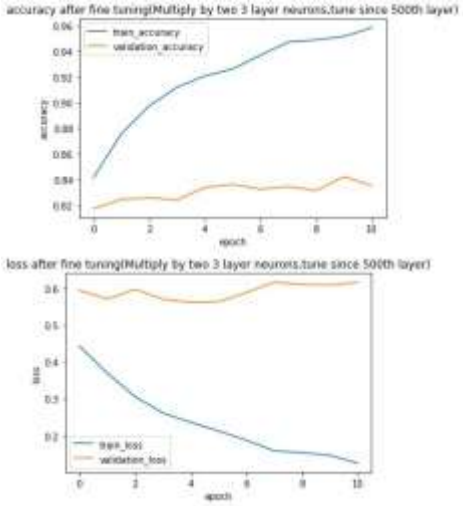
شکل ۹: نمودارهای accuracy و loss برای آموزش دسته بند با ۲۰۴۸، ۱۰۲۴ و ۱۰ نورون



شکل ۱۰: نمودارهای accuracy و loss برای tune دسته بند با ۵۱۲، ۲۵۶ و ۱۰ نورون



شکل ۱۱: نمودارهای accuracy و loss برای tune دسته بند با ۱۰۲۴، ۵۱۲ و ۱۰ نورون

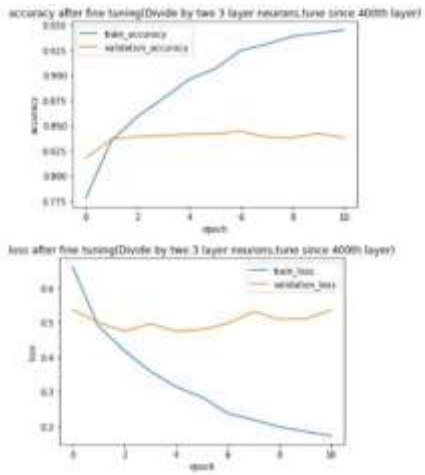


شکل ۱۲: نمودارهای accuracy و loss برای tune دسته بند با ۲۰۴۸، ۱۰۲۴ و ۱۰ نورون

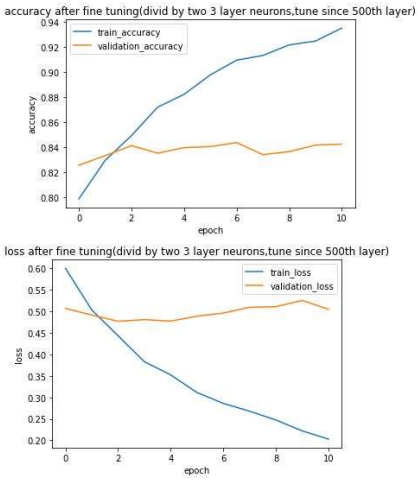
۱۰۲۴، ۵۱۲ و ۱۰ نورون می‌باشد. مدل دوم دارای لایه ها با ۵۱۲، ۲۵۶ و ۱۰ نورون می‌باشد و مدل سوم دارای لایه ها با ۲۰۴۸، ۱۰۲۴ و ۱۰ نورون می‌باشد. در شکل بالا همانند نمودارهای قسمت قبل نتایج را مشاهده می‌کنید. تغییر تعداد نورون‌ها به به این صورت تاثیر چندانی در نتایج نهایی ندارد و تنها قدری حالت ۵۱۲، ۲۵۶ و ۱۰ نورون نتایج بهتری دارد

نقطه شروع tuning

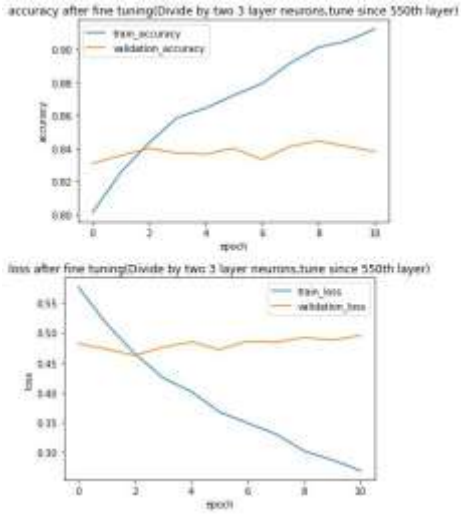
در این قسمت می‌خواهیم اثر نقطه شروع tuning را بررسی کنیم. با سه مقدار مختلف این کار را انجام می‌دهیم یکبار از لایه ۴۰۰ به بعد یک بار از لایه ۵۰۰ به بعد و یک بار هم ازلایه ۵۵۰ به بعد tuning را روی مدل منتخب در بخش قبل روی مدل مبنا انجام می‌دهیم توجه شدم مدل مبنا دارای ۵۹۵ لایه می‌باشد. در زیر نتایج این آزمایش به روال بخش‌های قبل آمده است.



شکل ۱۳: نمودارهای accuracy و loss برای tune دسته بند با ۵۱۲، ۲۵۶ و ۱۰ نورون از لایه ۴۰۰ مدل مبنا



شکل ۱۴: نمودارهای accuracy و loss برای tune دسته بند با ۵۱۲، ۲۵۶ و ۱۰ نورون از لایه ۵۰۰ مدل مبنا



شکل ۱۵: نمودارهای accuracy و loss برای tune دسته بند با ۵۱۲، ۲۵۶ و ۱۰ نورون از لایه ۵۵۰ مدل مبنا

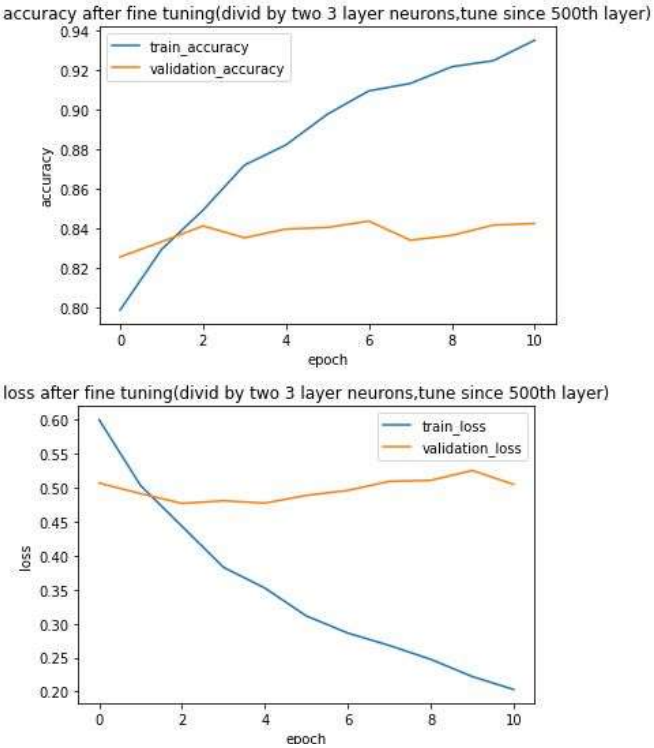
نقطه شروع ۵۰۰ مناسب تر از دو حالت دیگر می‌باشد البته تفاوت زیادی بین این سه نیست اما نقطه شروع ۵۰۰ قدری به لحاظ val_accuracy بالا تر است.

نتیجه گیری

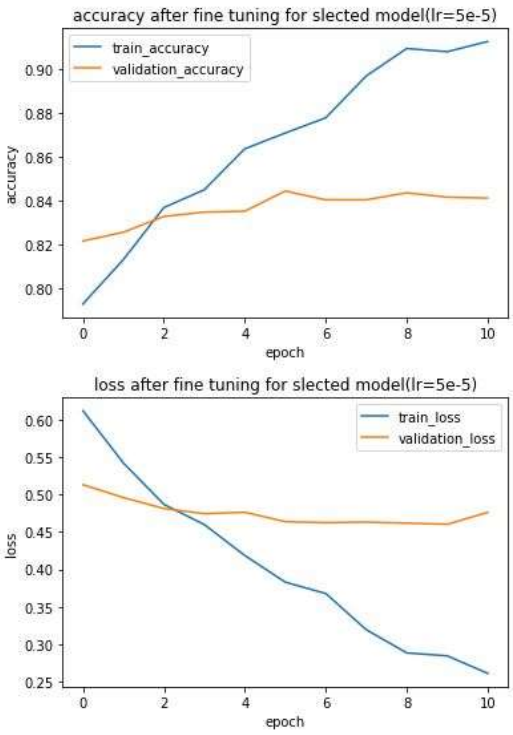
استفاده از انتقال یادگیری به مراتب از تنایج بهتری از آموزش از صفر یک شبکه عصبی دارد. در اینجا میزان دقت ارزیابی یه ۸۴ درصد در یک زمان کوتاه و با مجموعه آموزشی کم رسید اما در تمرین قبل در بهترین حالت به دقت ارزیابی در حدود ۶۷ درصد رسیدیم. در واقع انتقال یادگیری هم از نظر نتایج و هم از زمان و حجم پردازش بسیار مناسب تر از آموزش صفر شبکه است.

نرخ یادگیری

برای بررسی اثر نرخ یادگیری این آزمایش را روی مدل منتخب انجام می‌دهیم. در سراسر این گزارش نرخ یادگیری در آموزش دسته بند 0.001 و برای نرخ یادگیری در tuning مقدار 0.0001 را در نظر گرفتیم حال با تقسیم این دو مقدار بار دیگر دو مرحله آموزش را با دو مقدار 0.0005 و 0.00005 انجام می‌دهیم.



شکل ۱۶: نمودارهای accuracy و loss برای tune مدل منتخب با نرخ یادگیری 0.0001



شکل ۱۷: نمودارهای accuracy و loss برای tune مدل منتخب با نرخ یادگیری 0.00005

کاهش نرخ یادگیری سبب کاهش دقت آموزش شده است اما این کاهش سبب بهبود محسوس مقدار دقت ارزیابی نشده است

در این آموزش مقدار دقت روی داده های تست نیز برای مدل منتخب برابر ۸۴,۱۶ درصد می‌باشد.