سوال ۱

آ) فایل عکس با نام A1_1 ضمیمه شده است.

ب) و **پ)** فایل عکس با نام A1_2&3 ضمیمه شده است.

ت) زمانیکه از مکس پولینگ استفاده می‌کنیم (عموما با اندازه ۲ در ۲)، اندازه تصویر را کاهش می‌دهیم تا بار محاسباتی برای سیستم کمتر شود. در این حالت در کرنل‌های ۲ در ۲ (یعنی هر ۴ پیکسل) بیشترین مقدار پیسکل را قرار می‌دهیم که نماینده و نمایش دهنده مقدار پیکسل‌ها در این محدوده می‌باشد؛ همین امر باعث می‌شود که اگر چرخی به اندازه ۴۵ درجه صورت گرفته باشد، دیگر اهمیت پیدا نکند چون ما بالاخره موقعیت پیکسلی که محتوای تصویر را در خود دارد پیدا می‌کنیم و آن را در تصویر کاهش یافته قرار می‌دهیم. زمانیکه که ما یک لایه از ویژگی‌ها داریم عددی که بیشتر است احتمال این را می‌دهد که ویژگی موردنظر ما را در خود نگهداری کند؛ کرنل مکس پولینگ این ویژگی را از بین دیگر ویژگی‌های بیرون می‌کشد و در این حین اگر ما چرخشی داشته باشیم، در حقیقت جای این ویژگی را عوض کرده‌ایم، که بسته به اندازه کرنل مکس پولینگ، مجددا آن را پیدا خواهیم کرد.

ث) در یادگیری انتقالی (Transfer Learning)، یک مدل، برای کاری که به عنوان نقطه شروع مدل ، در انجام کار دیگری استفاده مجدد می‌شود، توسعه میابد. نوعی از یادگیری انتقال استفاده از ضرایب و ترم‌های بایاس بدست آمده پس از فرآیند یادگیری از سیستم یادگیری اولیه، به عنوان ضرایب اولیه سیستم یادگیری ثانویه است. برای حل یک مسئله، ما باید یک مدل از قبل آموزش دیده برای حل یک مسئله مشابه را در اختیار داشته باشیم. ما به جای اینکه برای مسئله‌ای مشابه یک مساله از پیش حل شده، یک مدل جدید بسازیم، از مدلی به عنوان نقطه شروع استفاده می‌کنیم که در موارد دیگر آموزش دیده است. پس باید دقت کرد که اگر صورت مسئله‌ای که در دست ماست، بسیار متفاوت از مدل از قبل آموزش دیده باشد، پیش‌بینی ما از این مسئله بسیار نادرست خواهد بود. ما از یادگیری انتقالی، برای تعمیم دادن به داده خارج از مجموعه داده‌های خود استفاده می‌کنیم. این کار فقط با داشتن یک مدل از قبل آموزش دیده اتفاق می‌افتد. بدین صورت، ما از یک مدل تنظیم دقیق (Fine-tuning) که از اصلاح یک مدل از قبل آموزش دیده بدست آمده، استفاده می‌کنیم. از آنجایی که فرض می‌کنیم که شبکه از قبل، به خوبی آموزش دیده است، بنابراین نمی‌خواهیم وزن‌ها را خیلی زود و بیش از حد تغییر دهیم. در زمان اصلاح، معمولاً از یک نرخ یادگیری کوچکتر از مقداری که در آموزش ابتدایی مدل استفاده شده است، استفاده می‌کنیم. ما از یادگیری انتقالی، برای صرفه‌جویی در زمان و یا برای رسیدن به عملکرد بهتر، به عنوان یک بهینه‌سازی استفاده می‌کنیم. سه روش برای تنظیم دقیق مدل وجود دارد که عبارتند از:

- استخراج ویژگی: برای یک مکانیزم استخراج ویژگی، می‌توانیم از یک مدل از قبل آموزش دیده استفاده کنیم که لایه خروجی آن را حذف کرده‌ایم. علاوه بر این، ما باید از کل شبکه به عنوان یک استخراج کننده ویژگی معین برای مجموعه داده‌های جدید استفاده کنیم.
- آموزش برخی از لایه‌ها در حالی که سایر لایه‌ها ثابت نگه داشته شده‌اند: یک روش دیگر برای استفاده از یک مدل از قبل آموزش دیده وجود دارد که آموزش جزئی مدل (بخش

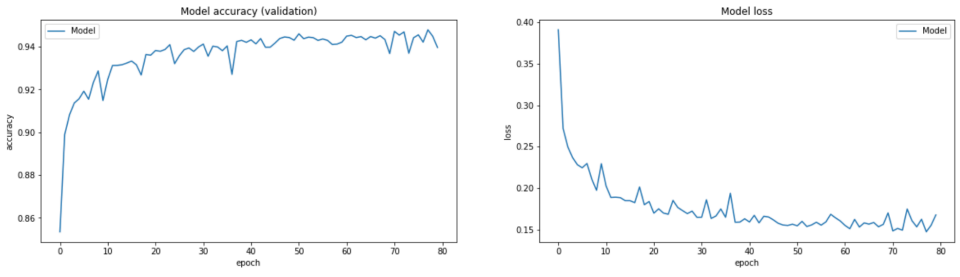
بخش) نامیده می‌شود. در این روش ما باید وزن لایه‌های اولیه مدل را ثابت نگه داریم در حالی که فقط باید لایه‌های بالاتر را بازآموزی کنیم. در اینجا می‌توانیم، امتحان کنیم و ببینیم که چند لایه باید ثابت نگه داشته شود و چند لایه آموزش داده شود.

- استفاده از معماری مدل از قبل آموزش دیده: با توجه به یک مجموعه داده، در زمان مقداردهی اولیه و آموزش مدل، از معماری آن استفاده می‌کنیم.

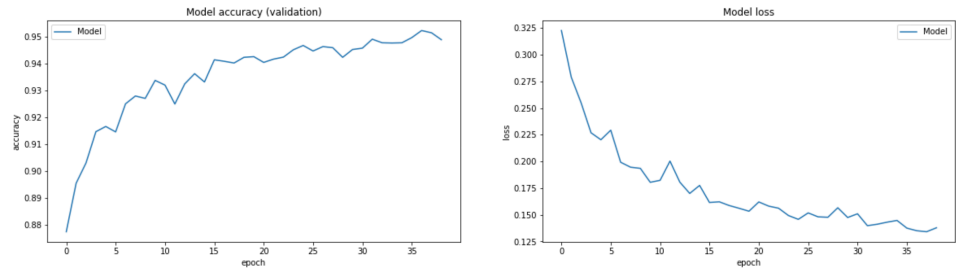
Transfer Learning و Fine-tuning به جای یکدیگر نیز استفاده و به عنوان فرآیند آموزش یک شبکه عصبی بر روی داده‌های جدید تعریف می‌شوند، اما مقداردهی اولیه آن با وزن‌های از پیش آموزش دیده به دست آمده از آموزش آن بر روی یک مجموعه داده متفاوت، که عمدتاً بسیار بزرگتر و تا حدودی مربوط به داده‌ها و وظایفی است که شبکه قبلاً روی آن آموزش دیده بود. در یادگیری انتقالی، معمولاً چند لایه آخر شبکه با لایه‌های جدید جایگزین می‌شوند و با وزن‌های تصادفی مقداردهی اولیه می‌شوند، لایه‌های بدون تغییر می‌توانند منجمد شوند، یعنی غیر قابل آموزش باشند یا قابل آموزش نگه داشته شوند.

سوال ۲

ب) در حالتیکه optimizer ما adam است مدل سریعتر همگرا می‌شود و سرعت افزایش دقت، هم در فاز آموزش و هم در معتبرسازی، بالا می‌رود.اما چون الگوریتم محاسبه‌ی آن از RMSprop پیچیده‌تر می‌باشد، تکرارهای ما با سرعت بالاتری طی می‌شوند. الگوریتم RMSprop تا پایان یافتن ۸۰ تکرار، حتی به دقت ۹۵ درصد در داده های تست نرسیده است، در حالیکه با استفاده از Adam در تکرار ۳۸اُم به این دقت دست پیدا می‌کنیم. در الگوریتم RMSprop گام‌های ما در نقطه‌ی ابتدایی تا نقطه بهینه، نوسانات بالایی را طی می‌کنند. در اشکال زیر دقت و میزان اتلاف هر دو الگوریتم روی داده‌های دیده‌نشده (یا به اصلاح داده های صحت یا تست) نشان داده می‌شود؛ همانطور که گفته شد نوسانات در RMSprop بالاتر می‌باشد و همچنین با مقایسه‌ی نمودار lossها متوجه می‌شویم که Adam نمودار هموارتری دارد و روی این مجموعه داده بهتر عمل می‌کند.

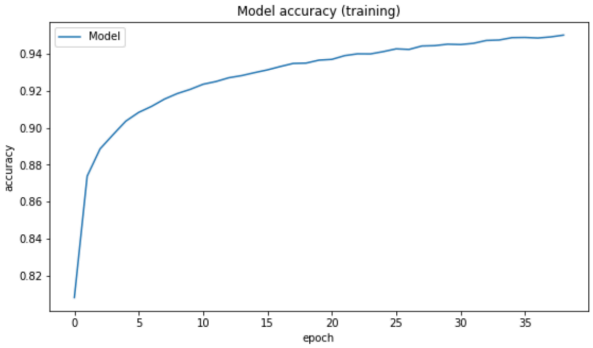


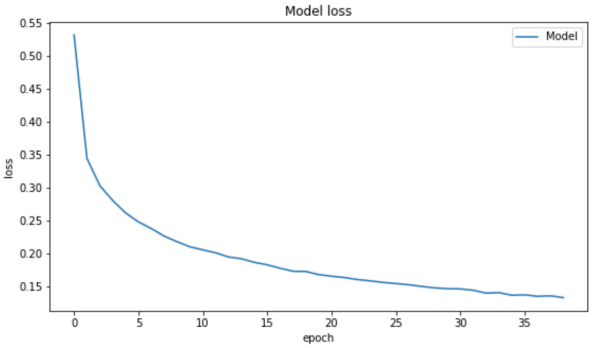
RMSprop



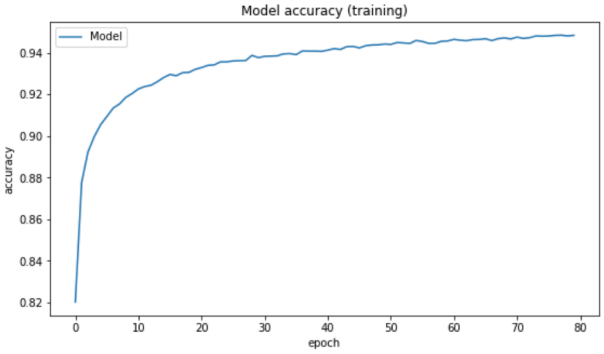
Adam

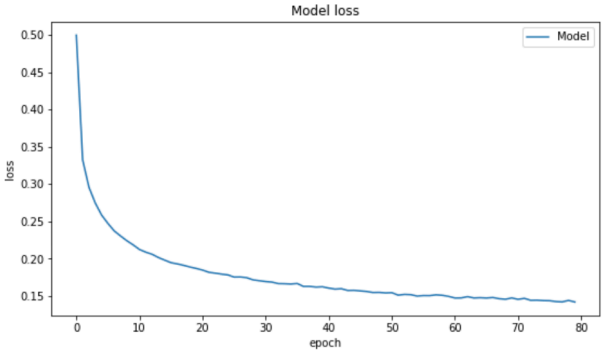
پ)





Adam





RMSprop

سوال ۳

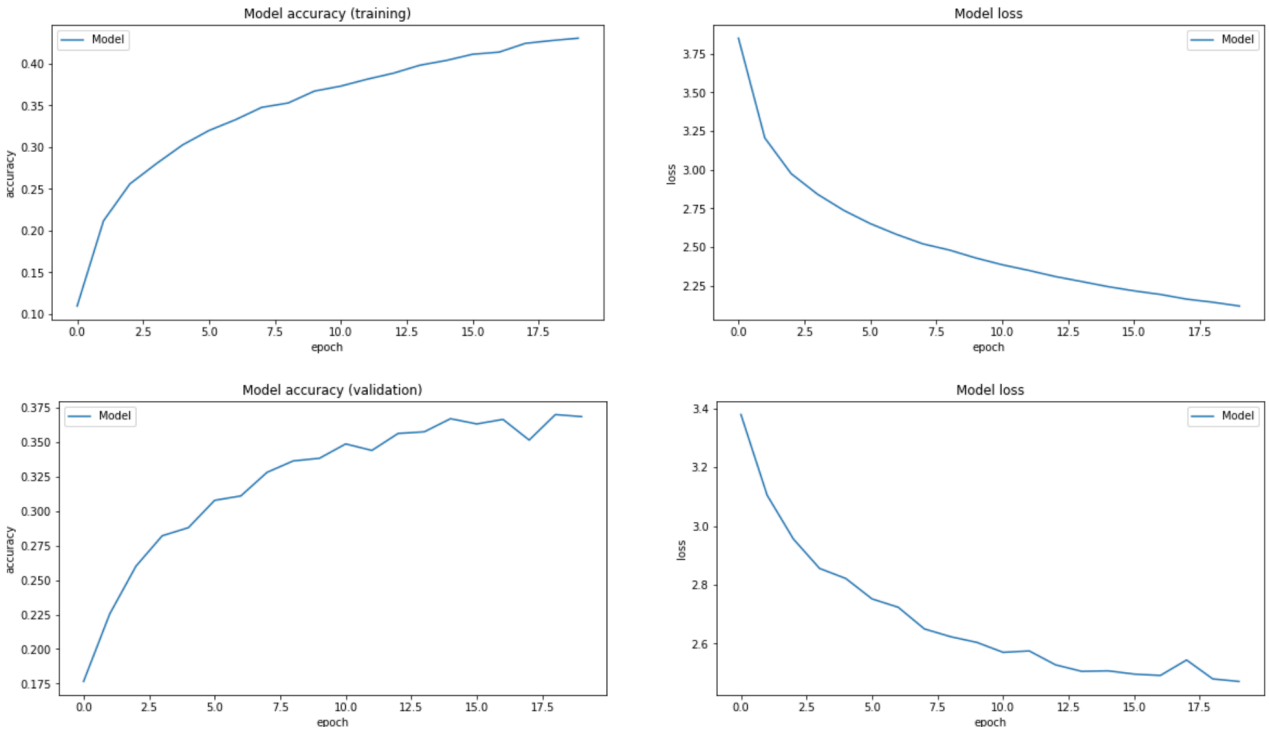
آ) معماری شبکه و تعداد پارامترها در شکل زیر مشاهده می‌شود:

```
Model: "sequential_9"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_32 (Conv2D)          (None, 30, 30, 32)        896

 max_pooling2d_3 (MaxPooling  (None, 15, 15, 32)       0
 2D)

 conv2d_33 (Conv2D)          (None, 13, 13, 64)        18496

 max_pooling2d_4 (MaxPooling  (None, 6, 6, 64)         0
 2D)

 conv2d_34 (Conv2D)          (None, 4, 4, 128)         73856

 max_pooling2d_5 (MaxPooling  (None, 2, 2, 128)        0
 2D)

 flatten_8 (Flatten)         (None, 512)               0

 dense_30 (Dense)            (None, 512)               262656

 dense_31 (Dense)            (None, 256)               131328

 dense_32 (Dense)            (None, 128)               32896

 dense_33 (Dense)            (None, 100)               12900

=================================================================
Total params: 533,028
Trainable params: 533,028
Non-trainable params: 0
_________________________________________________________________
```

ب) برای سنجیدن میزان خطای یا همان فاصله‌ی ما از نقطه بهینه، از تابع خطا استفاده می‌کنیم. برای مسائلی که چند کلاسه هستند (توزیع داده‌ها طبقه‌بندی شده است) از تابع خطای Categorical Cross-entropy استفاده می‌کنیم. این تابع ترکیبی از softmax activation و cross-entropy loss می‌باشد. softmax activation احتمال وقوع هریک از کلاس‌ها را به ما برمی‌گرداند (عددی بین ۰ و ۱ برای هرکلاس) و پس از آن برای محاسبه cross-entropy از منفی لگاریتم در مبنای e برای هریک از خروجی‌ها استفاده می‌کنیم؛ در بخش آموزش مسئله چون برچسب خروجی را می‌دانیم، تنها از منفی لگاریتم احتمال وقوع همان دسته به عنوان میزان خطا استفاده می‌کنیم. خروجی مسئله‌ی ما در اینجا ۱۰۰ کلاسه می‌باشد، برای اینکه احتمال وقوع هرکدام از کلاس‌ها در نظر گرفته شود ازین تابع خطا استفاده شده است. استفاده ازین تابع باعث می‌شود که میزان خطا، برای ورودی‌هایی که مقدار پیش‌بینی شده‌ی بسیار پرت است، نسبت به ورودی‌هایی که نسبتاً درست پیش‌بینی کرده‌اند، بسیار متفاوت‌تر باشد (شیب بیشتری به پیش‌بینی‌های پرت نسبت داده می‌شود)، همین امر باعث می‌شود که اگر بیش‌بینی بسیار بدی داشتیم، قدم بزرگتری به سمت پیش‌بینی بهتر برداریم.

سوال ۴

ب) با آموزش دادن لایه‌ی کانولوشن پنجم به نتیجه‌ی زیر می‌رسیم:

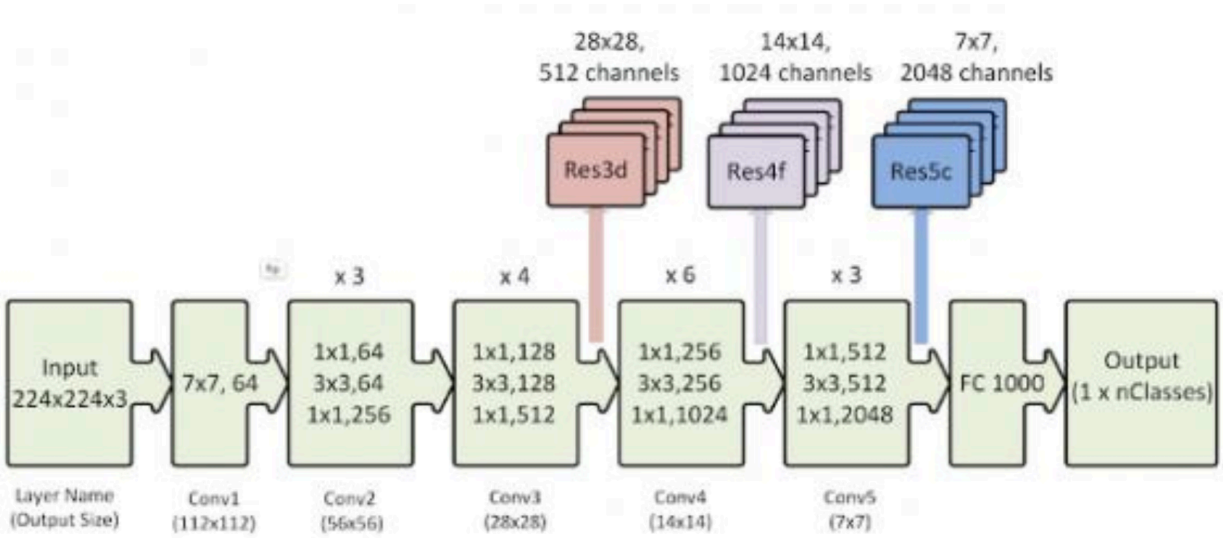


پ) با مقایسه کردن نمودارهای دقت متوجه می‌شویم که در فاز تست، هنگامی که از آموزش لایه کانولوشنی پنجم استفاده کرده‌ایم، به دقت بالاتری رسیده‌ایم. این نشان می‌دهد که تعمیم‌پذیری مدل بالا رفته است و حال روی دیتاست ما عملکرد بهتری را نشان می‌دهد. کار

لایه‌ی کانولوشن پیدا کردن شباهت است و زمانیکه تصمیم می‌گیریم که وزن‌های مربوط به این لایه نیز آموزش پیدا کنند، باعث می‌شود که به واریانس داده‌های ورودی توجه بیشتری شود و الگوهای مربوط به دیتاست را بیاموزد.

سوال ۵

پ) این شبکه از ۵۰ لایه کانولوشنی ایجاد شده است که به شرح آن‌ها می‌پردازیم. برای افزایش سرعت محاسبات، لایه‌ها کانولوشنی به صورت بلاک بلاک اجرا می‌شوند و روند هر بلاک به این صورت است که چند لایه‌ی کانولوشنی با یکدیگر کانکت شده و خروجی به بلاک بعدی داده می‌شود. در ابتدا اندازه‌های ورودی به لایه‌ی ورودی داده می‌شود (در اینجا ۲۲۴ در ۲۲۴ در ۳) و بعد zero padding برای آن‌ها صورت می‌پذیرد، بعد از آن یک کانولوشن ۶۴تایی و سپس لایه‌ی نرمال‌سازی دسته‌ای را با همین اندازه فیلتر داریم، بعد از آن، مقادیر خروجی را از تابع فعال‌ساز رِلو عبور می‌دهیم و در آخر لایه‌ی مکس‌پولینک را با zero padding اضافه می‌کنیم. حال ازینجا به بعد بلاک‌های کانولوشنی ما آغاز می‌شوند؛ در بلاک اول سه لایه‌ی کانولوشنی وجود دارد (کانولوشن اول: 1x1x64، کانولوشن دوم: 3x3x64، کانولوشن سوم: 1x1x256) که بعد از آن‌ها نرمال‌سازی دسته‌ای، به علاوه تابع فعال‌ساز قرار گرفته است، در آخر این لایه‌های کانولوشنی با یکدیگر کانکت شده و به عنوان ورودی به بلاک بعدی داده می‌شوند. سه عدد از این بلاک در شبکه ما وجود دارد. برای بلاک بعدی باز سه کانولوشن با اندازه و تعداد فیلتر متفاوت از بلاک قبلی در نظر گرفته شده که تعدادشان چهار عدد می‌باشد. تعداد بلاک با سه کانولوشن جدید برای بخش بعد شش عدد و بخش بعدتر سه عدد می‌باشد. در آخر یک لایه فولی‌کانکتت ۱۰۰۰ نورونه وجود دارد که بتواند عملیات طبقه‌بندی به ۱۰۰۰ کلاس را برای ما انجام دهد.



ج) برای استخراج ویژگی‌ها از لایه‌ی آخر با نام predictions استفاده کرده‌ایم، چون این لایه آخرین لایه‌ی ماست که فولی‌کانکتد نیز می‌باشد و عملیات دسته‌بندی را انجام می‌دهد.

لایه‌های قبلی اطلاعات ارزشمندی را از ۱۰۰۰ دسته‌ای که در imageNet وجود دارد را نگهداری می‌کنند، با استفاده از این اطلاعات یک شبکه بازگشت طراحی شد که شامل یک لایه ۵۱۲تایی بلوک LSTM و ۳ لایه فولی کانکتد بود، که این شبکه توانسته عملیات دسته‌بندی ویدیو را برای ما انجام دهد. این شبکه بازگشتی می‌تواند مفهوم زمان و توالی را بهتر از شبکه‌های دیگر درک کند.

فاطمه طاهر ۴۰۰۱۰۰۰۶۹۷