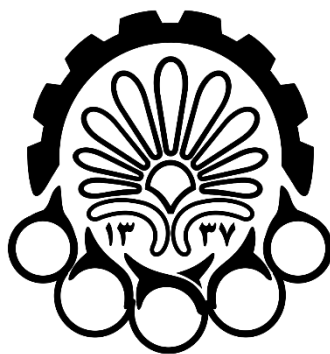


به نام خدا



دانشگاه صنعتی امیرکبیر  
( پلی تکنیک تهران )

تمرین درس شبکه‌های عصبی-سری چهارم

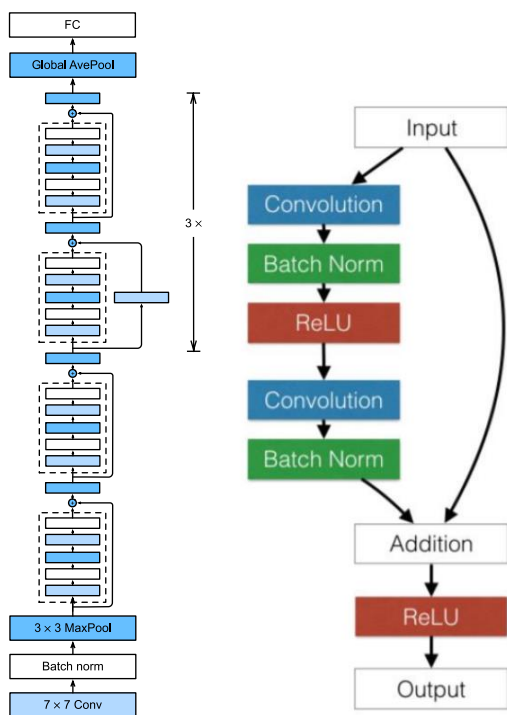
فردین آیار

شماره دانشجویی: ۹۹۱۳۱۰۴۰

استاد: دکتر صفابخش

دانشکده کامپیوتر- زمستان

۱) یک نمونه از بلاک‌های تشکیل دهنده شبکه ResNet در شکل ۱-الف نمایش داده شده است. همانطور که در شکل مشخص است، ویژگی اصلی این بلاک‌ها این است که ورودی هر بلاک با خروجی آن جمع شده و به بلاک بعدی منتقل می‌شود. شبکه ResNet از مجموعه‌ای از این بلاک‌ها و به دنبال آن یک شبکه تماماً متصل تشکیل شده است. این شبکه مشکل گرادینان محوشونده/انفجاری را در شبکه‌های عمیق حل می‌کند و در نتیجه معمولاً تعداد لایه‌های آن بسیار زیاد است. در شکل ۱-ب یک نمونه از این شبکه نمایش داده شده است.



شکل ۱-ب

شکل ۱-الف

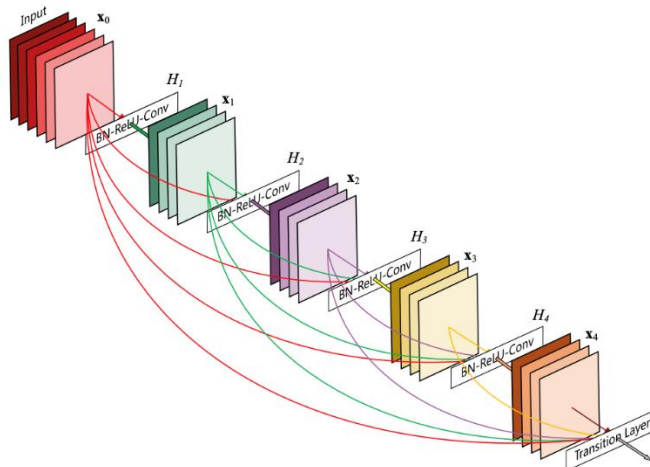
در شکل ۲ یک نمونه از بلاک‌های تشکیل دهنده شبکه DenseNet نمایش داده شده است. هر بلاک از مجموعه‌ای از لایه‌های Conv+ReLU+BN تشکیل شده است. مطابق شکل، کانال‌های ویژگی لایه‌های قبلی به هم متصل شده و به لایه فعلی وارد می‌شوند. این کار علاوه بر حل مشکل گرادینانی محوشونده/انفجاری، باعث می‌شود تعداد فیلترهای موجود در هر لایه کمتر باشد و در نتیجه تعداد پارامترهای این شبکه در مقایسه با ResNet به شدت کمتر است. شبکه DenseNet از مجموعه‌ای از بلاک‌های فوق که بوسیله لایه‌های انتقالی به هم متصل شده‌اند، تشکیل شده است (شکل ۳-۱). همچنین به دنبال این لایه‌ها یک شبکه تماماً متصل برای دسته‌بندی ویژگی‌های استخراج شده قرار می‌گیرد.

دو تفاوت اصلی شبکه DenseNet با شبکه ResNet عبارت‌اند از:

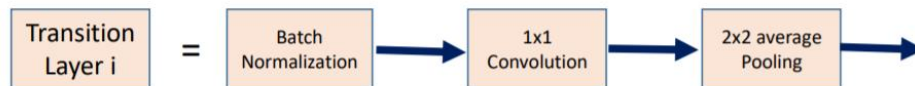
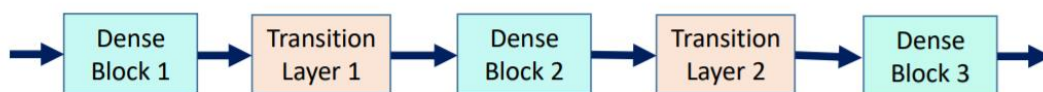
۱) در شبکه DenseNet برخلاف شبکه ResNet که هر بلاک تنها از دو بلاک قبل از خود ورودی می‌گیرد، لایه‌های موجود در یک بلاک از تمام لایه‌های پیشین موجود در همان بلاک ورودی می‌گیرد. به عبارت دیگر تعداد و عمق اتصالات در شبکه DenseNet بیشتر است.

۲) کانال‌های ویژگی‌های ورودی از لایه‌های مختلف، در ResNet با هم جمع می‌شوند در حالی که در DenseNet این کانال‌ها به هم متصل<sup>۱</sup> می‌شوند.

<sup>۱</sup> concatenate



شکل ۲



شکل ۳

۲) این کار در کد مربوط به هر شبکه به صورت جداگانه انجام شده است. عکس‌ها به ابعاد  $128 \times 128$  تبدیل شده‌اند عکس‌های مربوط به محیط بیرون با برچسب ۱ و محیط داخلی با برچسب ۰ مشخص شده‌اند. همچنین پیکسل‌ها به منظور نرمال سازی بر ۲۵۵ تقسیم شده‌اند.

۳) ابتدا کل شبکه‌های مذکور به جز لایه‌های تماماً متصل آن‌ها را بارگیری کرده (شبکه پایه) و سپس یک شبکه تماماً متصل جدید (شبکه دسته‌بندی) به آن‌ها اضافه می‌کنیم. تعداد نرون‌های خروجی شبکه جدید باید برابر با تعداد کلاس‌های مسئله باشد. پارامترهای شبکه پایه را ثابت کرده و شبکه دسته‌بندی را آموزش می‌دهیم تا پارامترهای آن تعیین شود. سپس باید شبکه‌های ایجاد شده را برای انطباق کامل به فضای مسئله جدید، تنظیم دقیق کنیم. بدین منظور پارامترهای بخشی از شبکه پایه را از حالت ثابت خارج کرده و شبکه را با یک بار با ضریب یادگیری کوچک آموزش می‌دهیم.

(۴)

## شبکه ResNet:

کد مربوط به این بخش پیوست شده است. همچنین نسخه colab آن در این [لینک](#) موجود است. از شبکه ResNet101 به عنوان شبکه پایه استفاده می‌کنیم. خروجی این شبکه را به یک شبکه تماماً متصل با مشخصات زیر متصل می‌کنیم.

لایه	
۱	Flatten()
۲	Dense(256, activation="relu")
۳	Dropout(0.2) (headModel)
۴	Dense(256, activation="relu")
۵	Dropout(0.2) (headModel)
۶	Dense(2, activation="softmax")

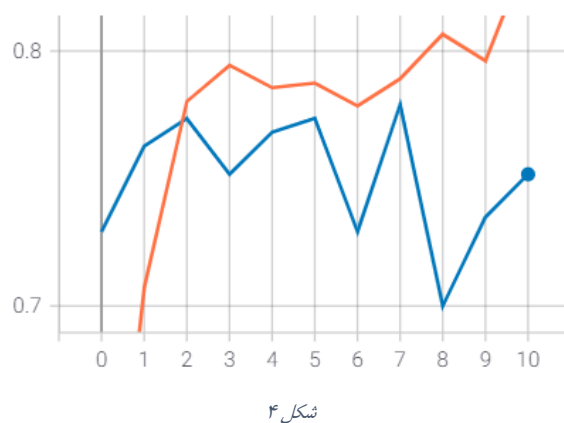
با روشی که در سوال ۳ توضیح داده شد، شبکه را در دو مرحله آموزش می‌دهیم. در مرحله اول فقط شبکه تماماً متصل و در مرحله دوم شبکه تماماً متصل و ۵ لایه آخر ResNet آموزش داده می‌شود. نتایج هر مرحله در جدول زیر ارائه شده است.

مرحله	ضریب یادگیری	تعداد تکرار <sup>۲</sup>	دقت آموزش	دقت اعتبارسنجی	دقت آزمون
۱	۰.۰۰۱	۱۱	۰.۷۶۶۶	۰.۷۷۲۲	---
۲	۰.۰۰۰۱	۲۰	۰.۸۴۶۰	۰.۷۶۱۱	۰.۷۸۷۵

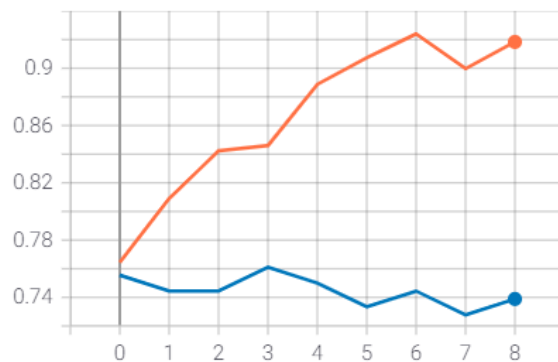
مطابق جدول فوق، اگرچه در مرحله دوم دقت مدل روی مجموعه آزمون افزایش یافته است، اما به علت بیش برآزش دقت اعتبارسنجی بهبود خاصی نداشته است. ماتریس درهم‌ریختگی داده‌های تست در جدول زیر ارائه شده است.

۳۱	۱۰
۷	۳۲

در انتها نمودار دقت آموزش و اعتبارسنجی برحسب تکرار برای مرحله اول و دوم آموزش در شکل‌های ۴ و ۵ ارائه شده است. وجود بیش برآزش در مدل، به ویژه در شکل ۵ کاملاً مشهود است.



<sup>۲</sup> شرط توقف، عدم بهبود هزینه داده‌های اعتبارسنجی در پنج تکرار متوالی است. همچنین محدودیت حداکثر ۲۰ تکرار برای تعداد تکرار تعیین شده است.



شکل ۵

### شبکه DenseNet:

کد مربوط به این بخش پیوست شده است. همچنین نسخه colab آن در این [لینک](#) موجود است. از شبکه DenseNet121 به عنوان شبکه پایه استفاده می‌کنیم. خروجی این شبکه را به یک شبکه تماماً متصل با مشخصات زیر متصل می‌کنیم. (مشابه آنچه در شبکه ResNet استفاده شد)

لایه	
۱	Flatten()
۲	Dense(256, activation="relu")
۳	Dropout(0.2) (headModel)
۴	Dense(256, activation="relu")
۵	Dropout(0.2) (headModel)
۶	Dense(2, activation="softmax")

با روشی که در سوال ۳ توضیح داده شد، شبکه را در دو مرحله آموزش می‌دهیم. در مرحله اول فقط شبکه تماماً متصل و در مرحله دوم شبکه تماماً متصل و ۱۰ لایه آخر DenseNet آموزش داده می‌شود. نتایج هر مرحله در جدول زیر ارائه شده است.

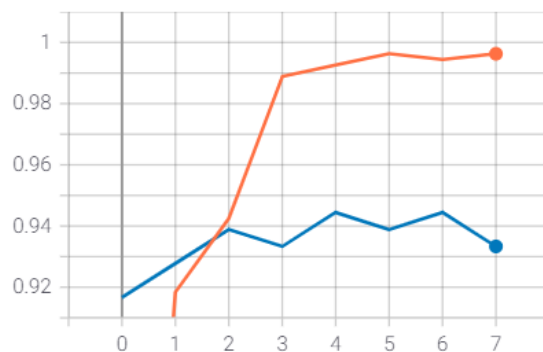
مرحله	دقت آزمون	دقت اعتبارسنجی	دقت آموزش	تعداد تکرار <sup>۳</sup>	ضریب یادگیری
۱	---	۰.۹۳۸۹	۰.۹۴۲۵	۸	۰.۰۰۱
۲	۱.۰	۰.۹۴۴۴	۰.۹۷۵۹	۱۰	۰.۰۰۰۱

مطابق جدول فوق نتایج دو مرحله تفاوت چندانی با هم ندارند و مدل حتی بدون تنظیم دقیق به دقت بسیار بالایی رسیده است. ماتریس درهم‌ریختگی داده‌های تست در جدول زیر ارائه شده است.

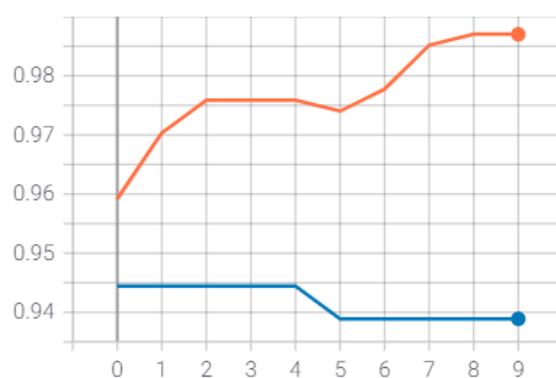
۴۱	۰
۰	۳۹

در انتها نمودار دقت آموزش و اعتبارسنجی برحسب تکرار برای مرحله اول و دوم آموزش در شکل شکل‌های ۷ ارائه شده است. در این مدل نیز به نظر می‌رسد در مرحله دوم اندکی بیش‌برازش وجود دارد.

<sup>۳</sup> شرط توقف، عدم بهبود هزینه داده‌های اعتبارسنجی در پنج تکرار متوالی است. همچنین محدودیت حداکثر ۲۰ تکرار برای تعداد تکرار تعیین شده است.



شکل ۶



شکل ۷

#### مقایسه و نتیجه گیری:

با توجه به نتایج، در این مسئله، مدل DenseNet هم از نظر سرعت و هم از نظر دقت نسبت به ResNet برتری دارد. نکته مهم دیگر درباره DenseNet این است که بدون انجام مرحله تنظیم دقیق، دقت آن به مقدار بالایی رسیده است؛ در نتیجه به نظر می‌رسد ویژگی‌های استخراج شده آن عمومیت بیشتری دارد. به طور کلی در هر دوشبکه قبل از تنظیم دقیق به دقت بالایی رسیده‌ایم. علت این اتفاق احتمالا این است که این دو شبکه برای وظیفه‌های پیچیده‌تری آموزش دیده‌اند و بنابراین برای وظیفه آسان‌تر این سوال، ویژگی‌های استخراج شده کافی می‌باشند.