

به نام خدا

پروژه‌ی سوم

Sarmeili.f@gmail

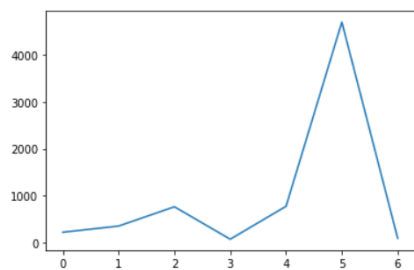
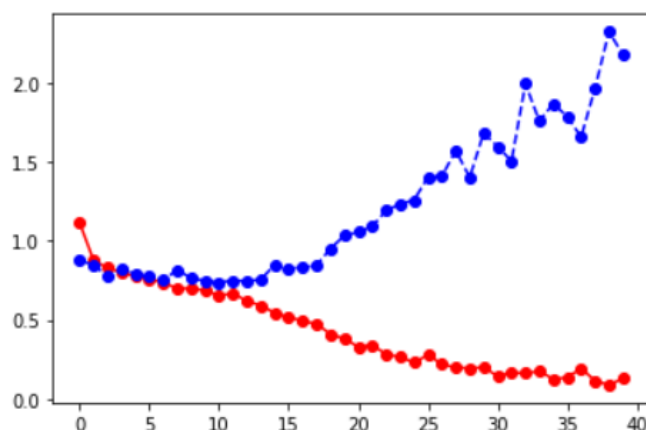
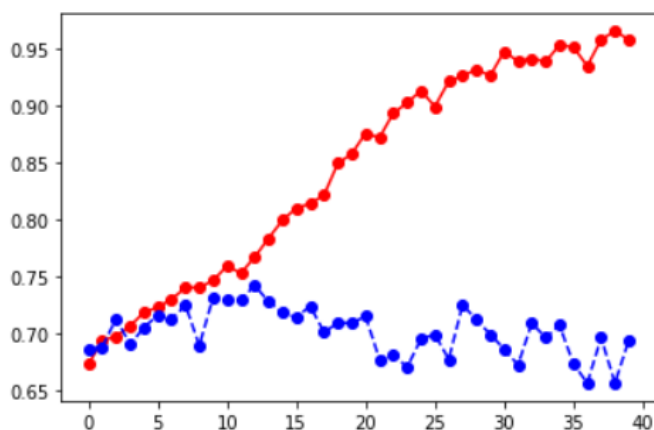
Dermatology MNIST

فراز سرمیلی

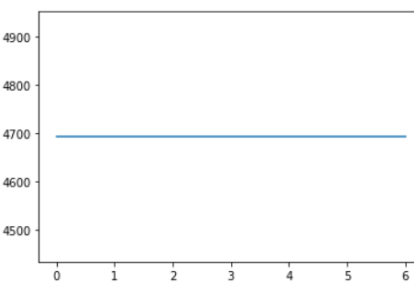
Model: "sequential_11"

Layer (type)	Output Shape	Param #
conv2d_70 (Conv2D)	(None, 28, 28, 16)	448
conv2d_71 (Conv2D)	(None, 28, 28, 16)	2320
max_pooling2d_35 (MaxPooling)	(None, 14, 14, 16)	0
conv2d_72 (Conv2D)	(None, 14, 14, 32)	4640
conv2d_73 (Conv2D)	(None, 14, 14, 32)	9248
max_pooling2d_36 (MaxPooling)	(None, 7, 7, 32)	0
conv2d_74 (Conv2D)	(None, 7, 7, 64)	18496
conv2d_75 (Conv2D)	(None, 7, 7, 64)	36928
max_pooling2d_37 (MaxPooling)	(None, 3, 3, 64)	0
flatten_11 (Flatten)	(None, 576)	0
dense_22 (Dense)	(None, 128)	73856
dense_23 (Dense)	(None, 7)	903
Total params: 146,839		
Trainable params: 146,839		
Non-trainable params: 0		

در ابتدا جهت اینکه معیاری برای سنجش داشته باشیم مدلی را مشابه مدل VGG ساخته و بدون هیچ رگولاریزیشن و هیچگونه پریپراسسینگ به کمک دیتای ترین، آموزش میدهیم و عملکرد آن را بر روی دیتای ولیدیشن میسنجیم. لازم به ذکر است که ورودی مدل VGG اصلی عکس های با ابعاد بزرگ تر از ۲۸ در ۲۸ را به عنوان ورودی قبول می کند و عکس های موجود در این دیتاست اگر به عنوان ورودی به VGG داده شود ارور بعد منفی از مدل دریافت می شود، چرا که ابعاد عکس در لایه های آخر به یک پیکسل و پس از آن به بعد منفی باید تبدیل شود. بنابراین مدلی که اینجا استفاده کرده ام مدلی مشابه VGG می باشد که نسخه اولیه آن را در عکس رو به رو نشان می دهیم. در این مدل به شکل دیفالت از اپتیمایزر Adam با لرنینگ ریت ۰,۰۰۱ که دیفالت خود اپتیمایزر هم هست استفاده شده است. نمودارهای مربوط به ترین مدل در زیر آمده است. همانطور که انتظار می رود مدل به دلیل کم بودن دیتا اورفیت شده است و در نهایت بر روی دیتای تست اکپورسی ۶۹,۵ درصد را به ما داده است.



۱. در مرحله پریپراسسینگ اولین کاری که در پردازش تصویر به ذهن می رسد نرمالایز کردن به کمک تقسیم بر ۲۵۵ کردن است که به نوعی مقدار پیکسل ها را بین صفر و یک مقیاس می کند. پس از نرمالایز کردن مدل به این شیوه وضعیت مدل کمی بهتر شد و دقت به ۷۲ درصد نزدیک شد که خب قابل انتظار هم بود. با توجه به اورفیت شدن مدل شاید یکی از کارهایی که بتواند به مدل کمک کند افزایش تعداد دیتا به وسیله دیتا آگمنتیشن باشد که به همین دلیل یک بار هم به کمک دیتا آگمنتیشن مدل را ترین می کنیم. تا حدودی دیتا آگمنتیشن توانست جلوی اورفیت را بگیرد البته دقت مدل کمی افت پیدا کرد و دقت ۶۷ درصد رسید.

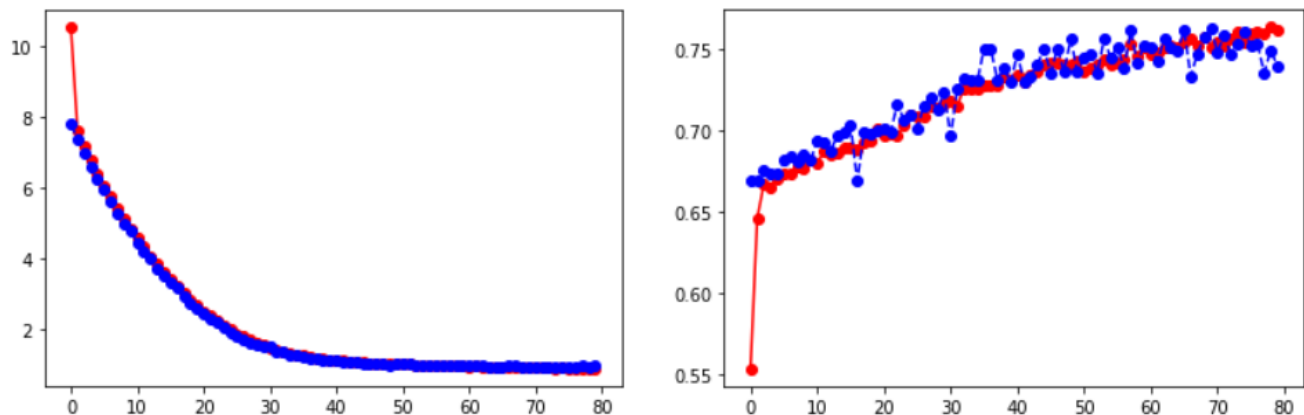


نکته دیگری که خیلی حائز اهمیت می باشد این است که وقتی نمودار فراوانی لیبیل ها را می کشیم متوجه می شویم که دیتای ما آنبالانس می باشد. برای حل مشکل در این دیتاست و تمرین راهکاری که به ذهنم رسید استفاده از ریسمپلینگ می باشد. در زیر نمودار فراوانی لیبیل ها قبل و بعد از اور ریسمپلینگ را می بینید. بعد از انجام اور ریسمپلینگ بر روی دیتاها دیتا آگمنتیشن انجام دادم و به این شکل تعداد داده ها نیز کمی بیشتر شد. مقدار دیتاها تا حدودی جلوی اورفیت شدن مدل را گرفتند ولی خب دقت باز هم پایین تر آمد و به ۶۳ درصد رسید. از آنجا که ترین کردن مدل به کمک این روش خیلی طول می کشد پس برای ترین نهایی مدل از آن استفاده می کنیم. یک بار هم مدل را بدون دیتا آگمنتیشن و فقط به کمک رندم اور سمپلینگ ترین کردم و خب مدل خیلی اورفیت

شد که قابل انتظار هست با توجه به ایجاد عکس های تکراری در هر کلاس. به همین دلیل یک بار هم به سراغ رندم آندر سمپلینگ رفتیم و به علت کم بودن دیتاها در این روش از جنریتور و آگمنتیشن استفاده کردیم. که باز به علت کم بودن دیتا مدل اصلا ترین نشد. با توجه به نتایج گرفته شده ابتدا سعی میکنیم مدل را به کمک تنها نرمالیزیشن تیون کنیم و در نهایت باز به آگمنتیشن یا مسئله آنبالانس بودن دیتا ست برمیگردیم. (در ارتباط با ریسایز کردن تصویر هم در سوال آینده توضیح داده شده است).

۲. همانطور که میدانیم خود مدل VGG به عنوان ورودی عکس هایی با ابعاد ۲۲۴ در ۲۲۴ را دریافت می کند به همین دلیل حدس زدیم شاید ریسایز کردن تصاویر و استفاده از خود مدل VGG شاید موثر تر واقع شود. وقتی عکس های ریسایز شده را بر روی مدل دیفالتی که در بالا اشاره شد ترین کردیم مدل همچنان اورفیت و یک مقدار اکپورسی پایین تر آمد و به عدد ۶۶ درصد رسید. با توجه به طولانی بودن ترین یک بار هم بر روی خود مدل اورجینال VGG16 مدل را ترین میکنم و از رگولارایز های دراپ اوت و L2 هم استفاده می کنم. در اینجا به علت بزرگ بودن ابعاد عکس و زیاد شدن پارامترهای مدل اورجینال ترین کردن مدل خیلی طول میکشد ولی با توجه به ۱۰ ۱۵ اپیک اول خیلی تفاوتی در ایجاد اکپورسی یا مسئله اورفیت شدن دیده نشد. برای همین از همان مدل قبلی استفاده کرده و سعی می کنیم مدل را در همان سایز دیفالت ترین کنیم. به نظر میرسد علتی که خیلی تغییر ابعاد کمک نکرد این بود که عکس های دیتاست از اول تغییر ابعاد داده شده بودند به ۲۸ در ۲۸ و یک سری دیتا های غیر ضروری شاید از آن حذف شده است و عکس ها به نوعی پری پراسس داشته اند و تغییر ابعاد عکس عملا دیتای جدیدی به مدل اضافه نکرده است و پیچیده تر کردن مدل تاثیری بر روی عملکرد آن نداشته است. به کمک دراپ اوت و رگولایزر ۰.۰۱ مسئله اورفیت به طور کلی حل شد ولی نمودار دقت آن کمی دچار نویز شد و به همین علت به نظر میرسد که بهتر باشید لرنینگ ریت را پایین تر بیاوریم. همانطور که در زیر مبینید با تغییر نرخ یاد گیری به ۰.۰۰۰۱ و اضافه کردن L2 به هر لایه و همچنین اضافه کردن دراپ اوت به هر بلاک مسئله اورفیتینگ و نویز زیاد بهتر شد. و دقت به ۷۵ درصد رسید بدون اورفیت شدن که در سایت هایی که بنچمارک ها را گذاشتند هم بالاترین دقتی که از این دیتا ست گرفتند همین مقدار بوده است. (<https://medmnist.com/>)

در این سوال یک بار هم با تغییر تعداد لایه های بلوک دوم و سوم یعنی به جای استفاده از دو لایه کانولوشنی ۳۲ و ۶۴ از هر کدوم سه لایه استفاده کردیم خب اندکی مدل را اورفیت کرد ولی در دقت تغییر چندانی ایجاد نشد.



۳. در مدل یک بار به جای همه توابع فعالسازی Relu از tanh استفاده کردیم و تغییر چندانی در عملکرد و دقت مدل مشاهده نشد ولی یک مقدار ترین شدن مدل بیشتر طول کشید. یک بار هم به کمک اکتیویشن فانکشن sigmoid سعی کردیم مدل را ترین کنیم که مدل ترین نمیشد اصلا. اکتیویشن فانکشن دیگه ای که امتحان کردیم selu بود که دقت و عملکرد بهتری را بر روی دیتاست داشت.

۴. خب همانطور که میدانیم در شروع اندازه کرنل ها ۳ در ۳ بود و الان با تغییر کرنل به اندازه ۵ در ۵ تغییر خاصی در دقت ایجاد نشد و شاید یک مقدار مدل به سمت اورفیتینگ حرکت کرده باشد. یک بار هم با کرنل ۷ در ۷ این کار رو انجام دادیم دقت مدل کمی پایین تر امد ولی مدل کاملاً اورفیت شد. در مدل های گذشته ابعاد کرنل مکس پولینگ ۲ در ۲ بود با استراید ۲ ولی این بار با کرنل ۳ در ۳ مدل را طراحی کردیم که دقت مدل مقدار خیلی کمی بهتر شد ولی به طور کلی تغییر خاصی حس نشد. به علت کوچک بودن ابعاد عکس خیلی با توجه به معماری مدل شاید نتوان با پارامتر استراید بازی کرد و تغییر داد برای همین با کمی ساده تر کردن مدل و گذاشتن استراید دو برای کرنل لایه های کانولوشنی یک بار مدل را تست کردیم. دقت به دلیل ساده شدن مدل کمتر شد و مدل به سمت اورفیت حرکت کرد. یک بار هم روی لایه پولینگ استراید یک در یک گذاشتیم به جای دو در دو و مدل کاملاً به سمت اورفیتینگ رفت و دقت مدل هم به مراتب پایین امد. اگر به جای ماکس پولینگ هم از اورج پولینگ استفاده کنیم دقت مدل کمی پایین می آید و به همین دلیل استفاده از ماکس پولینگ انتخاب بهتری می باشد.

در پایان باز هم اشاره میکنم که دیتاست به علت آنبالانس بود شاید دقتی که گزارش میکند حقیقی نباشد به همین دلیل یک بار هم دیتاست را به کمک رندم اور سمپلینگ با مدلی که به دست آوردیم ترین کردیم تا عملکرد را بهتر بررسی کنیم. مدل دچار اورفیت شد ولی دقت ۶۸ درصد را بر روی تست به ما داد که فکر میکنم این دقت به واقعیت نزدیک تر باشد که البته مسئله اورفیتینگ هم به خاطر اور سمپلینگ کاملاً طبیعی می باشد.

کانفیوژن ماتریکس:

```
array([[ 0., 18.,  8.,  0.,  1.,  6.,  0.],
       [ 0., 28.,  7.,  0.,  0., 17.,  0.],
       [ 0., 16., 45.,  0.,  0., 49.,  0.],
       [ 0.,  7.,  1.,  0.,  0.,  4.,  0.],
       [ 0.,  2., 13.,  0.,  8., 88.,  0.],
       [ 0.,  7., 20.,  0.,  5., 637.,  2.],
       [ 0.,  4.,  0.,  0.,  1.,  5.,  4.]])
```

کانفیوژن ماتریکسی که در اینجا آورده شده بر مبنای دیتاهای ولیدیشن می باشد همانطور که پیش تر هم اشاره شد این دیتاست دیتاست آنبالانسی می باشد و همانطور که مشهود است کلاسی که دیتای بیشتری را شامل میشد اکویورسی بهتری را به نسبت کلاس های دیگه نمایش میدهد و به مدل به طور کلی در پیش بینی دو کلاس به خاطر حجم دیتاهای خیلی کم ناتوان است.

برای حل این مسئله هم همانطور که اشاره کردم میتوانیم از روش رندم اور سمپلینگ استفاده کنیم تا دیتاهای اندکی بالانس شوند. در رو به رو کانفیوژن ماتریکسی از مدلی که بر روی دیتاست اورسمپل ترین شده است را ملاحظه می کنید. همانطور که مشهود است شاید دقت کلی مدل پایین باشد ولی همانطور که قابل محاسبه هست و در زیر نیز آمده عملکرد مدل بر روی تک تک مدل ها به شکل کلی و با متریک های دیگر به مراتب از مدل

قبلی بهتر شده است. لازم به ذکر است که بر روی دیتای اور سمپل شده عمل دیتا آگمنتیشن نیز انجام شده تا اندکی از اورفیتینگ جلوگیری شود.

پیش از اورسمپلینگ:

```
on label 0, acc=96.71, precision=nan, recall=0.00
on label 1, acc=91.43, precision=32.29, recall=59.62
on label 2, acc=88.83, precision=48.61, recall=31.82
on label 3, acc=98.80, precision=nan, recall=0.00
on label 4, acc=88.93, precision=50.00, recall=2.70
on label 5, acc=79.86, precision=78.63, recall=95.98
on label 6, acc=99.00, precision=70.00, recall=50.00
```

پس از اورسمپلینگ:

```
on label 0, acc=83.69, precision=29.79, recall=10.43
on label 1, acc=82.67, precision=41.92, recall=55.29
on label 2, acc=79.63, precision=35.10, recall=50.22
on label 3, acc=91.89, precision=87.18, recall=50.67
on label 4, acc=84.25, precision=46.05, recall=59.91
on label 5, acc=84.03, precision=46.13, recall=70.19
on label 6, acc=92.46, precision=97.60, recall=48.44
```