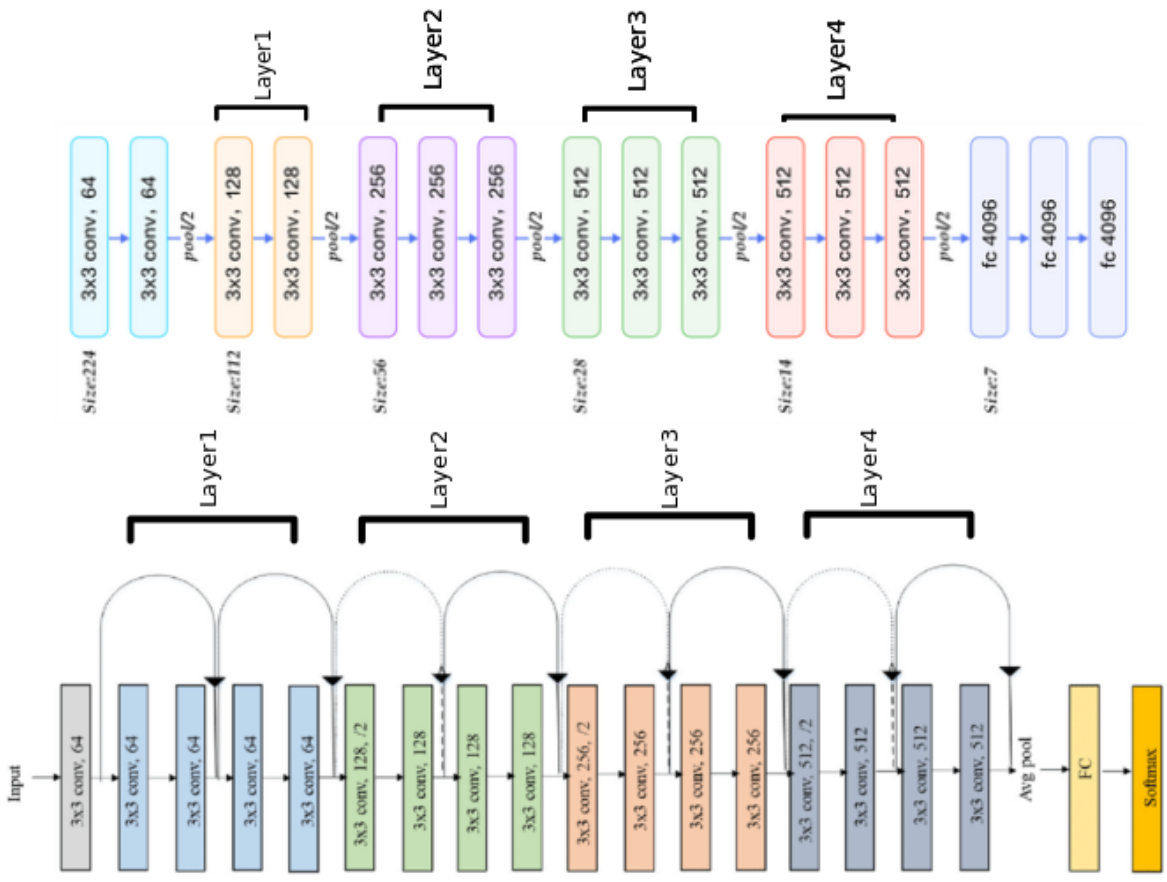
شبکه‌های عصبی
حامد محمدزاده
۹۸۱۲۷۶۲۴۱۸

تمرین دوم پایتون

۱) بخش اول:

۱.۱) توضیح آزمایش‌های بخش اول:

در بخش اول این تمرین خواسته شده از سه مدل از پیش تعلیم دیده VGG۱۶ ، ResNet۱۸ و DenseNet۱۲۱ برای پیش‌بینی سگ و گربه در دیتاست Cats vs Dogs استفاده شود.



شکل ۱: معماری Resnet۱۸ در پایین و معماری VGG۱۶ در بالا قرار دارد. نامگذاری لایه‌ها در این شکل بعدا مورد استفاده قرار می‌گیرد.

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112×112	7×7 conv, stride 2			
Pooling	56×56	3×3 max pool, stride 2			
Dense Block (1)	56×56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56×56	1×1 conv			
	28×28	2×2 average pool, stride 2			
Dense Block (2)	28×28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28×28	1×1 conv			
	14×14	2×2 average pool, stride 2			
Dense Block (3)	14×14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14×14	1×1 conv			
	7×7	2×2 average pool, stride 2			
Dense Block (4)	7×7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1×1	7×7 global average pool			
		1000D fully-connected, softmax			

شکل ۲: معماری Densenet۱۲۱

برای این کار حالت‌های مختلف که چند درصد از لایه‌های مدل از پیش تعلیم داده شده فریز شوند، امتحان می‌شود. معماری سه مدل در شکل ۱ و ۲ آورده شده‌است، در تمام آزمایشات لایه‌های Fully Connected مدل‌ها حذف و دو لایه Fully Connected با اندازه‌های [2-256] به آخرین لایه پیچشی مدل اضافه می‌شود. بین این دو لایه از تابع فعالیت ReLU و برای Regularization از Dropout استفاده شده. تعداد وزن‌های جدید اضافه شده به هر مدل در جدول شماره ۱ آورده شده است.

نام مدل	وزن‌های متصل به اولین لایه	وزن‌های متصل به دومین لایه	مجموع
VGG16	25088×256	256×2	کل
ResNet18	512×256	256×2	کل
Densenet121	1024×256	256×2	کل

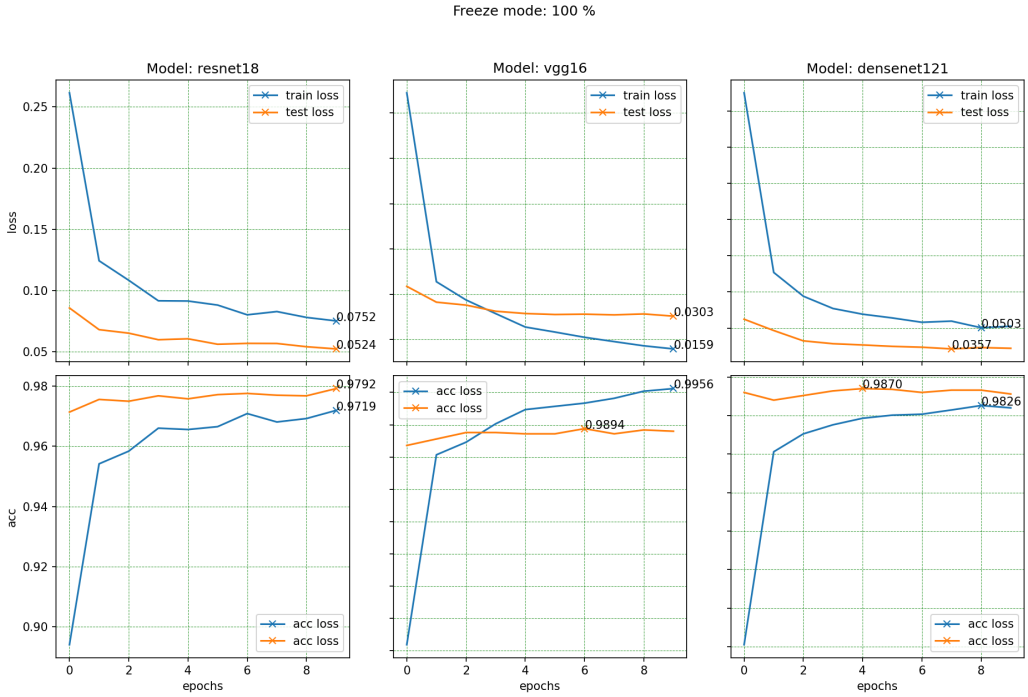
جدول ۱: تعداد پارامترهای جدید اضافه شده به هر شبکه که مربوط به لایه‌های Fully Connected می‌شوند.

اگر تنها یک لایه Fully Connected به هر مدل اضافه شود، در حالتی که ۱۰۰ درصد لایه‌های پیچشی فریز هستند، با یک لایه Fully Connected فقط یک طبقه‌بند خطی خواهیم داشت، برای همین ۲ لایه اضافه می‌شود. در همه آزمایشات، ۲۵ هزار داده کل دیتاست به ۲۰ هزار داده آموزشی و ۵ هزار داده تست تقسیم شده است.همچنین در تمامی آزمایشات وزن‌دهی اولیه، ترتیب داده‌های آموزشی و موارد دیگر که برای Reproducibilty لازم است، یکسان است.

۲.۱) 100% فریز لایه‌های پیچشی

همانطور که گفته شد در این آزمایشات تمامی لایه‌های Fully Connected حذف شده و دو لایه Fully Connected جایگزین آن‌ها شده اند. تمامی لایه‌های از پیش تعلیم داده شده فریز شده‌اند و هر مدل با بهینه‌ساز SGD با نرخ یادگیری 0.001 و مقدار momentum 0.9 به اندازه ۱۰ مرحله آموزش داده شده است.

در شکل ۳ نمودار تابع زیان و دقت برای هر سه مدل در حالت فریز 100% آورده شده است.



شکل ۳: نمودار تابع زیان و دقت برای هر سه مدل در حالتی که تمام لایه‌های پیچشی فریز شده‌اند. نقاطی که در هر نمودار با ضربدر مشخص شده‌اند مربوط به کمترین/بیشترین مقدار تابع زیان/دقت برای داده‌های آموزشی و تست هستند.

طبق شکل ۳، مشاهده می‌شود که مدل VGG۱۶ ، overfit شده است و بعد از مرحله ۴ آموزش مقدار تابع زیان برای داده‌های تست کاهش زیادی نداشته اما تابع زیان داده‌های آموزشی به کاهش خود ادامه داده. اما در دو مدل دیگر overfitting دیده نمی‌شود، دلیل این موضوع به تعداد وزن‌های اضافه شده به شبکه برمی‌گردد. طبق جدول شماره ۱ ، از آنجایی که مدل VGG۱۶ بردارهای ۲۵۰۸۸ برای دو لایه Fully Connected لایه آخر تولید می‌کند، تعداد وزن‌های دو لایه Fully Connected هم به شدت نسبت به دو مدل دیگر بیشتر است و با این تعداد وزن زیاد و این تعداد داده کم آموزشی، overfitting رخ می‌دهد. در دو مدل دیگر که بردارهای ۱۰۲۴ و ۵۱۲ بعدی تولید می‌کنند، چون تعداد پارامترها زیاد نیست overfitting رخ نمی‌دهد.

با این تفاسیر و با توجه به شکل ۳ مدل DenseNet۱۲۱ بهترین عملکرد را در حالت فریز % 100 دارد.

۳.۱) فریز کردن بخشی از لایه‌های پیچشی

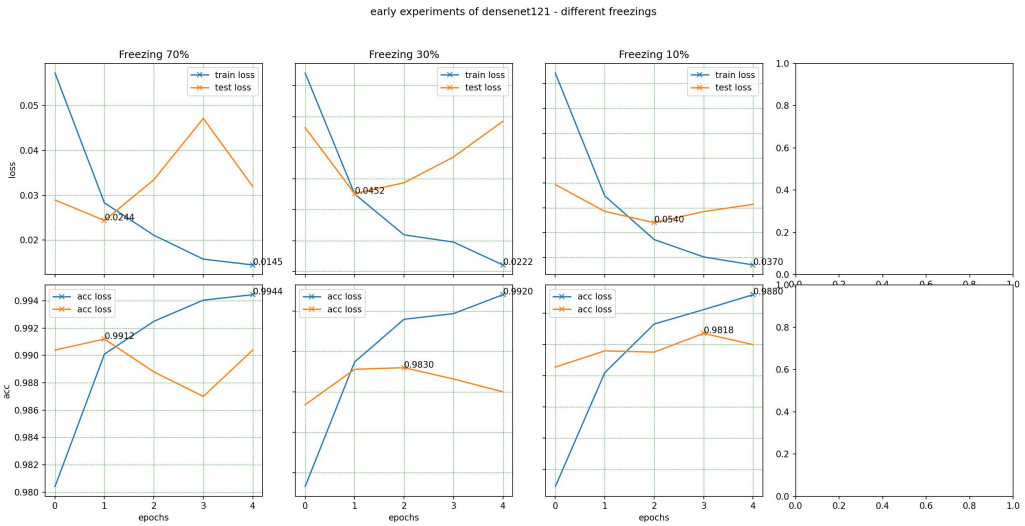
حالت‌های مختلفی از فریز کردن لایه‌های پیچشی آزمایش شده‌است. ابتدا تعریف می‌کنیم برای هر حالت فریز کردن ,70%) (10%, 30%, 50% چه لایه‌هایی از هر ۳ مدل فریز می‌شوند. این اطلاعات در جدول شماره ۲ آورده شده است.

حالت‌های مختلف فریز				
نام مدل	۷۰٪ فریز	۵۰٪ فریز	۳۰٪ فریز	۱۰٪ فریز
VGG16	Layer1, 2, 3, 4	Layer1, 2, 3	Layer 1, 2	Layer1
ResNet18	conv1+Layer1, 2, 3	conv1+Layer1, 2	conv1+Layer1	conv1
Densenet121	DenseBlock 1, 2, 3	-	DenseBlock 1, 2	DenseBlock 1, 2

جدول ۲: جدول مشخصات مربوط به لایه‌های فریز شده در هر کدام از آزمایشات، که در آن نشان داده شده برای هر مدل، در هر حالت فریز چه لایه‌هایی فریز شده‌اند. نام لایه‌ها به شکل ۱ و ۲ بر می‌گردد. دقت کنید اگر لایه‌های پیچشی در مدل وجود دارند و نام‌گذاری نشده‌اند، در همه حالات فریز هستند.

در آزمایشات اولیه‌ای که انجام شد، لایه‌های Fully Connected جدید که به شبکه اضافه شدند و وزن‌های اولیه تصادفی داشتند، با نرخ یادگیری یکسان به همراه بقیه لایه‌های پیچشی که فریز نیستند آموزش دیدند.

نتایج مربوط به این آزمایشات اولیه برای مدل Densenet۱۲۱ در شکل ۴ آورده شده‌است. (از آزمایشات مربوط به مدل‌های دیگر هم می‌توان به نتایج یکسانی رسید)



شکل ۴: آزمایشات اولیه برای مدل Densenet۱۲۱

طبق نتایج شکل ۴، به وضوح دیده می‌شود که overfitting رخ داده است. دلیل این موضوع این است که لایه‌های Fully Connected با وزن تصادفی چیزی یاد ندارند، برای همین در اوایل یادگیری مقدار خطا بسیار زیاد است و مشتق تابع زیان هم

زیاد می‌شود، این مشتق زیاد به لایه‌های پیچشی که فریز نیستند برمی‌گردد و وزن‌هایی که آن‌ها یاد گرفتند را به هم می‌زند، در صورتی که اشتباه اصلی به لایه Fully Connected بر می‌گردد، اما اپدیت وزن‌های شدیدی در لایه‌های پیچشی اتفاق می‌افتد. همچنین وقتی پارامترهای لایه‌های پیچشی که فریز نشده اند، به مجموعه کل پارامترهای قابل یادگیری شبکه اضافه می‌شوند، مقدار پارامترهای شبکه بالا رفته، درنتیجه با این مقدار داده کم (۲۰ هزار داده آموزشی) و این شبکه بزرگ، وزن‌ها به داده آموزشی overfit می‌شوند.

برای حل این مشکل، ۲ راه حل پیاده‌سازی شد که در ادامه توضیح داده شده‌اند.

- در ابتدای آموزش، تمام لایه‌های پیچشی فریز بمانند، فقط لایه‌های Fully Connected آموزش ببیند، بعد از چند مرحله که لایه آخر آموزش دیده‌شد، لایه‌های پیچشی که قرار است تعلیم ببینند هم از حالت فریز خارج شوند و همه لایه‌ها با هم آموزش داده شوند. به اصطلاح چندین مرحله warmup برای لایه‌های Fully Connected داشته باشیم. این کار در کد با مقدار دهی پارامتر warmup_steps انجام می‌شود.
- نرخ یادگیری برای لایه‌های پیچشی باید کم باشد، چون وزن‌های از پیش تعلیم شده آن‌ها نباید نابود شوند، برای همین برای هر حالت فریز یک نرخ یادگیری تعیین شده است که در جدول شماره ۳ آورده شده است. همچنین این آزمایشات با بهینه‌ساز SGD با momentum = 0.9 انجام شده‌اند. لازم به ذکر است مراحل آموزشی warmup که فقط وزن‌های لایه Fully Connected را تغییر می‌دهند با نرخ یادگیری جداگانه به مقدار 0.001 انجام می‌شود.

مقدار درصد فریز شدن	نرخ یادگیری
70%	0.0005
50%	0.0001
30%	0.00005
10%	0.00001

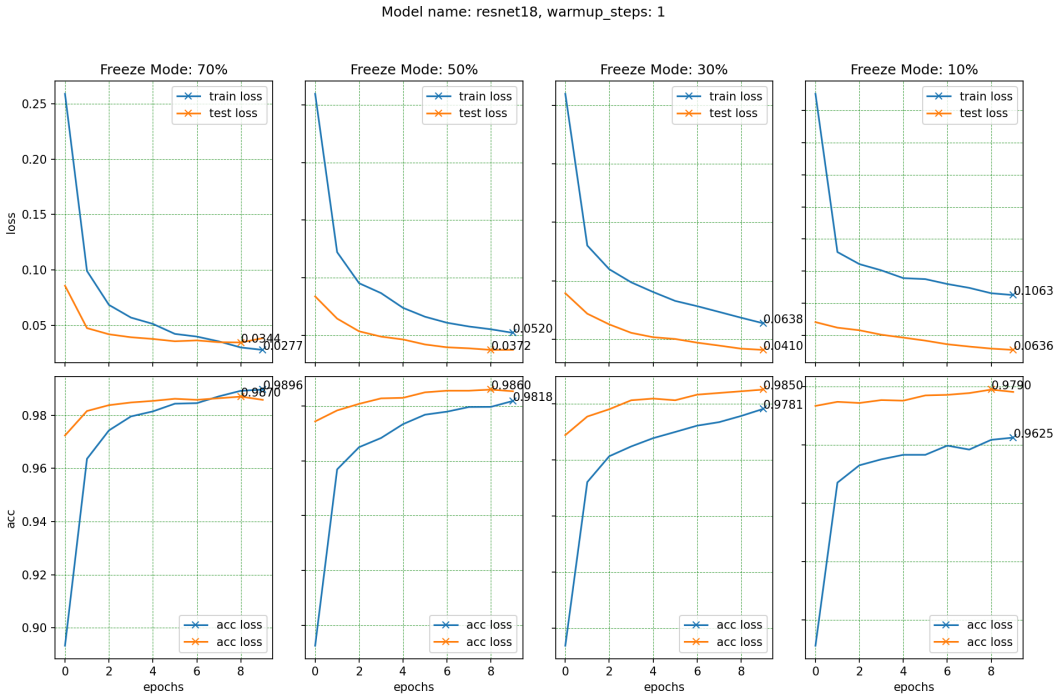
جدول ۳: مشخصات بهینه‌سازها برای حالت‌های مختلف فریز شدن

در ادامه به بررسی نتایج آزمایشات هر مدل با حالات مختلف فریز کردن می‌پردازیم که در این آزمایشات از راه‌حل‌های مطرح شده در بالا استفاده شده است.

۱.۳.۱) ResNet۱۸

در این آزمایشات مدل Resnet۱۸ برای هر حالت فریز شدن ، ۱۰ مرحله با batch size ۶۴ و نرخ یادگیری مطابق جدول شماره ۳ آموزش داده شده. همچنین تعداد مراحل warmup برای لایه Fully Connected مقدار ۱ در نظر گرفته شده.

شکل ۵ نمودار تابع زیان و مقدار خطا را برای هر مدل نشان می‌دهد. نکته قابل توجه، مقایسه شکل ۵ با شکل ۴ است، در شکل ۴ به خاطر دلایل گفته شده، overfitting رخ داده اما در شکل ۵ اینگونه نیست.

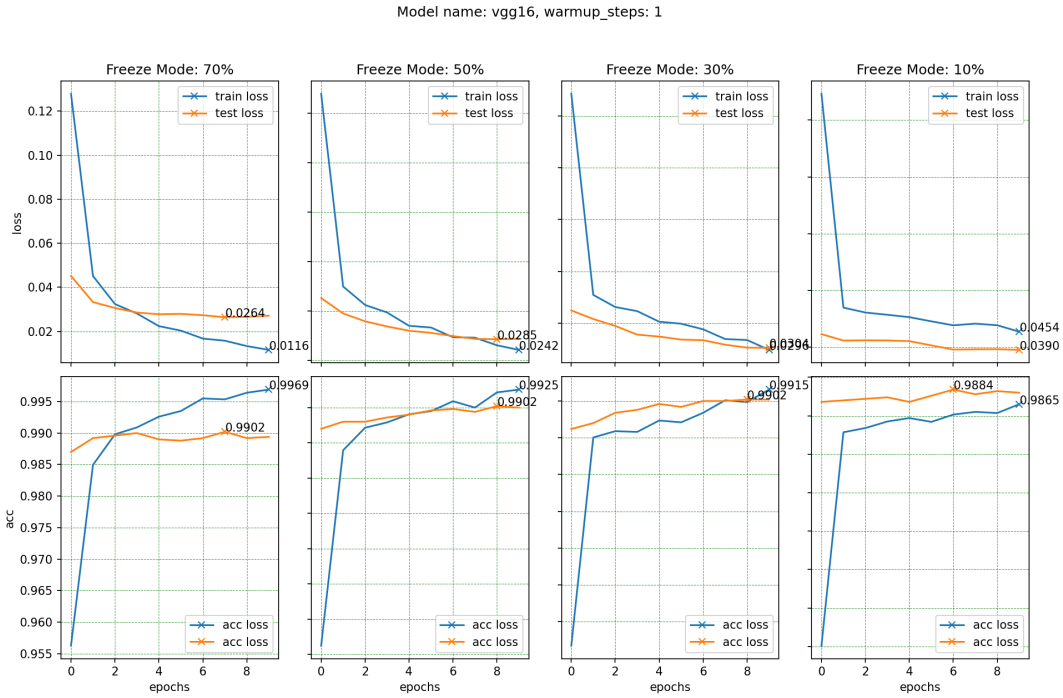


شکل ۵: نمودار تابع زیان و مقدار خطا برای مدل ResNet در حالت‌های مختلف فریز کردن. نقاطی که در هر نمودار با ضربدر مشخص شده‌اند مربوط به کمترین/بیشترین مقدار تابع زیان/دقت برای داده‌های آموزشی و تست هستند.

طبق شکل ۵، بهترین مدل از Resnet۱۸ ، حالت فریز ۷۰ درصدی است. (مدل مرحله ۸ از یادگیری به عنوان بهترین مدل ResNet در نظر گرفته می‌شود) همچنین طبق مقایسه شکل ۵ با شکل ۳، نتایج فریز کردن بخشی از لایه‌های پیچشی (با اعمال warmup و نرخ یادگیری متناسب) از حالت فریز کامل لایه‌های پیچشی بهتر است.

VGG۱۶ ۲.۳.(۱

شرایط آزمایشات مدل VGG۱۶ مشابه مدل Resnet۱۸ است که در قسمت ۱.۳.۱ توضیح داده شد. نمودار تابع زیان و مقدار دقت برای این آزمایشات در شکل ۶ آمده است.

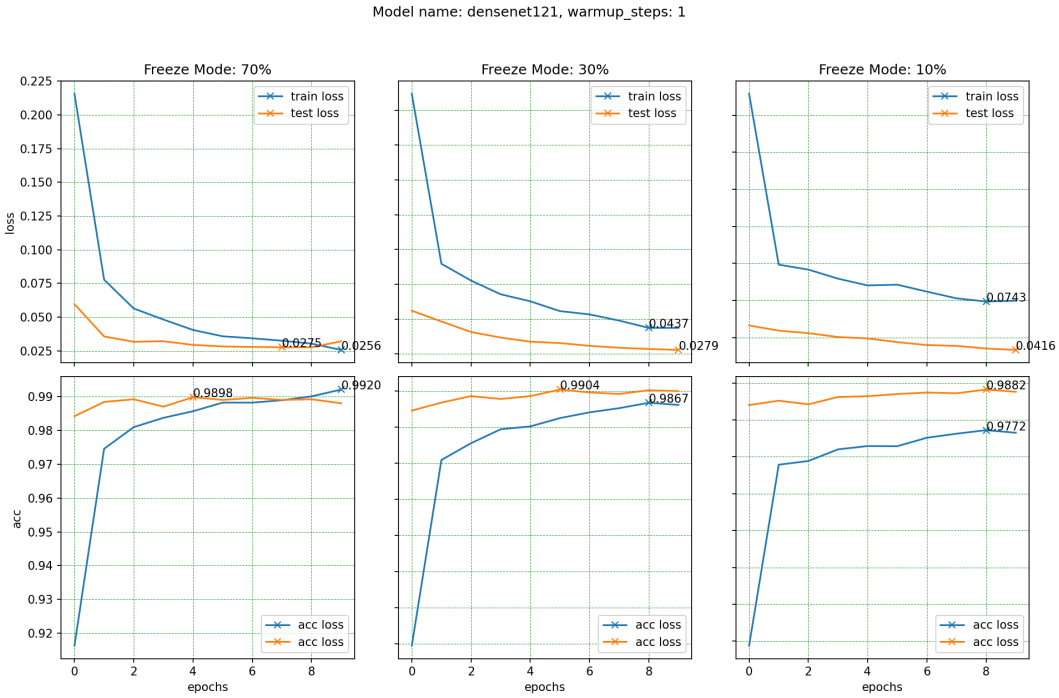


شکل ۶: نمودار تابع زیان و مقدار خطا برای مدل VGG۱۶ در حالت‌های مختلف فریز کردن. نقاطی که در هر نمودار با ضربدر مشخص شده‌اند مربوط به کمترین/بیشترین مقدار تابع زیان/دقت برای داده‌های آموزشی و تست هستند.

طبق شکل ۶، حالت فریز ۷۰ درصدی overfit شده است و فریز ۳۰ درصدی به عنوان بهترین مدل VGG۱۶ در نظر گرفته می‌شود.

۳.۳.۱) DenseNet۱۲۱

شرایط آزمایشات مدل DenseNet۱۲۱ مشابه مدل Resnet۱۸ است که در قسمت ۱.۳.۱ توضیح داده شد. همچنین لازم به ذکر است برای مدل DensNet۱۲۱ حالت فریز کردن ۵۰ درصد لایه‌ها وجود ندارد، زیرا لایه‌هایی که همگی یک DenseBlock را تشکیل می‌دهند، رابطه‌ای فراتر از لایه‌های پشت سر هم متصل شده عادی در مدلی مثل VGG۱۶ دارند، لذا تصمیم گرفته شده که لایه‌های پیچشی در هر DenseBlock همگی فریز یا فعال باشند. با این تصمیم، حالتی بوجود نمی‌آید که ٪۵۰ درصد از تعداد کل لایه‌های پیچشی مدل فریز باشند. (اینکه عدم رعایت این نحوه فریز کردن تاثیر منفی خواهد داشت یا نه، امتحان نشده است) نمودار تابع زیان و مقدار دقت برای این آزمایشات در شکل ۷ آمده است.



شکل ۷: نمودار تابع زیان و مقدار خطا برای مدل DenseNet۱۲۱ در حالت‌های مختلف فریز کردن. نقاطی که در هر نمودار با ضربدر مشخص شده‌اند مربوط به کمترین/بیشترین مقدار تابع زیان/دقت برای داده‌های آموزشی و تست هستند.

طبق شکل ۷، حالت فریز ۳۰ درصدی به عنوان بهترین مدل Densent۱۲۱ در نظر گرفته می‌شود.

۲) بخش دوم:

۱.(۲ توضیح آزمایشات بخش دوم:

در بخش دوم آزمایشات خواسته شده تا از مدل‌های بخش اول به عنوان استخراج کننده ویژگی استفاده شود و بردارهای ویژگی با طبقه بند Random Forrest آموزش ببینند. برای این کار بهترین شبکه آموزش دیده شده در قسمت اول تمرین، برای هر ۳ مدل Resnet۱۸ ، VGG۱۶ و DenseNet۱۲۱ استفاده شده است. در جدول شماره ۴ بهترین شبکه برای هر مدل آورده شده است.

Resnet۱۸ ۲.(۲

طبق جدول ۴ مدلی که با حالت فریز ۷۰ درصد آموزش دیده شده، برای استخراج ویژگی استفاده می‌شود و بردارهایی با ابعاد ۵۱۲ تولید می‌کند. این بردارها دقیقا بردارهایی است که به اولین لایه Fully Connected داده می‌شود. طبقه بند Random Forrest با پارامترهای مختلف بر ۲۰ هزار بردار ویژگی که این شبکه تولید کرده، آموزش داده شده. در جدول شماره ۵ مقدار دقت هر Random Forrest برای ۵ هزار داده تست آورده شده است.

نام مدل	بهترین شبکه
ResNet18	freeze 70%
VGG16	freeze 30%
DenseNet121	freeze 30%

جدول ۴: بهترین شبکه برای هر مدل در قسمت اول تمرین

	Number of Estimators				
	۵	۱۰	۲۰	۵۰	
max depth	۱۰	0.9783	0.9813	0.9841	0.9851
	۵۰	0.9779	0.9827	0.9839	0.9875*
	۱۰۰	0.9779	0.9827	0.9839	0.9875
	۲۰۰	0.9779	0.9827	0.9839	0.9875

جدول ۵: مقدار دقت RandomForrest با پارامترهای مختلف بر روی داده تست ، که با بردارهای ویژگی مدل ResNet۱۸ با ۷۰ درصد فریز آموزش دیده.

مشاهده می‌شود که بهترین دقت 0.9875 است که از دقتی که مدل به تنهایی (با استفاده از لایه‌های Fully Connected) می‌گیرد، کمتر است.

VGG۱۶ ۳.(۲

طبق جدول ۴ مدلی که با حالت فریز ۳۰ درصدی آموزش دیده شده، برای استخراج ویژگی استفاده می‌شود و بردارهایی با ابعاد ۲۵۰۸۸ تولید می‌کند. این بردارها دقیقا بردارهایی است که به اولین لایه Fully Connected داده می‌شود. طبقه بند Random Forrest با پارامترهای مختلف بر ۲۰ هزار بردار ویژگی که این شبکه تولید کرده، آموزش داده شده. در جدول شماره ۶ مقدار

	Number of Estimators				
	۵	۱۰	۲۰	۵۰	
max depth	۱۰	0.9585	0.9707	0.9779	0.9807
	۵۰	0.9587	0.9729	0.9807	0.9851
	۱۰۰	0.9579	0.9737	0.9815	0.9861*
	۲۰۰	0.9579	0.9737	0.9815	0.9861

جدول ۶: مقدار دقت RandomForrest با پارامترهای مختلف بر روی داده تست ، که با بردارهای ویژگی مدل VGG۱۶ با ۳۰ درصد فریز آموزش دیده.

دقت هر Random Forrest برای ۵ هزار داده تست آورده شده است. مشاهده می‌شود که بهترین دقت 0.9861 است که از

دقتی که مدل به تنهایی (با استفاده از لایه‌های Fully Connected) می‌گیرد، کمتر است.

DenseNet۱۲۱ ۴.(۲

طبق جدول ۴ مدلی که با حالت فریز ۳۰ درصد آموزش دیده شده، برای استخراج ویژگی استفاده می‌شود و بردارهایی با ابعاد ۱۰۲۴ تولید می‌کند. این بردارها دقیقا بردارهایی است که به اولین لایه Fully Connected داده می‌شود. طبقه بند Random Forrest با پارامترهای مختلف بر ۲۰ هزار بردار ویژگی که این شبکه تولید کرده، آموزش داده شده. در جدول شماره ۷ مقدار

		Number of Estimators			
		۵	۱۰	۲۰	۵۰
max depth	۱۰	0.9751	0.9817	0.9837	0.9853
	۵۰	0.9749	0.9809	0.9847	0.9867*
	۱۰۰	0.9749	0.9809	0.9847	0.9867
	۲۰۰	0.9749	0.9809	0.9847	0.9867

جدول ۷: مقدار دقت RandomForrest با پارامترهای مختلف بر روی داده تست ، که با بردارهای ویژگی مدل DenseNet۱۲۱ با ۳۰ درصد فریز آموزش دیده.

دقت هر Random Forrest برای ۵ هزار داده تست آورده شده است. مشاهده می‌شود که بهترین دقت 0.9867 است که از دقتی که مدل به تنهایی (با استفاده از لایه‌های Fully Connected) می‌گیرد، کمتر است.

۳) پیاده‌سازی

کد مربوط به انجام آزمایشات که در محیط Google Colab اجرا شده در فایل HW2_exps.py موجود است. کد مربوط به بخش اول آزمایشات در هدرهای Experiments1 Base Class و Experiments1 قرار دارد. همچنین کد مربوط به بخش دوم آزمایشات هم در هدرهای :Experiments2 Base class و RandomForrest train قرار دارد.

کد و داده‌های ذخیره شده از مراحل آموزش برای تولید نمودارها در پوشه plotting و training_info موجود است.