



دانشگاه صنعتی امیرکبیر  
(پای تکنیک تهران)

## پاسخ تمرین چهارم شبکه های عصبی

استاد درس جناب آقای دکتر صفابخش

تدریس یار جناب آقای مهندس اسدی

نیمسال دوم سال تحصیلی ۱۴۰۰-۱۴۰۱

محسن عبادپور

شماره دانشجویی: ۴۰۰۱۳۱۰۸۰

ایمیل: [m.ebadpour@aut.ac.ir](mailto:m.ebadpour@aut.ac.ir)

### فهرست پاسخ ها

|    |   |
|----|---|
| ۲  | .....Inception V3 و LeNet-5                   |
| ۸  | .....مسئله دوم؛ توضیحات و پیاده سازی          |
| ۹  | .....مسئله دوم؛ سناریو A                      |
| ۱۰ | .....مسئله دوم؛ سناریو B                      |
| ۱۱ | .....مسئله دوم؛ سناریو C                      |
| ۱۲ | .....مسئله دوم؛ سناریو D                      |
| ۱۳ | .....مسئله دوم؛ سناریو E                      |
| ۱۴ | .....مسئله دوم؛ مطالعه‌ی منظم ساز ها          |
| ۲۰ | .....مسئله دوم؛ مطالعه‌ی تعداد کرنل ها        |
| ۲۲ | .....مسئله دوم؛ مطالعه‌ی اندازه کرنل ها       |
| ۲۴ | .....مسئله سوم؛ فرآیند انتقال یادگیری         |
| ۲۵ | .....مسئله چهارم؛ انتقال یادگیری Inception V3 |

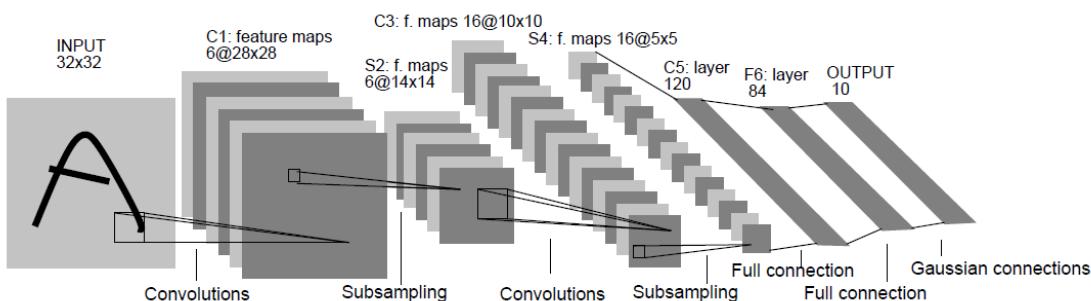
## مسئله اول

### مسئله اول: ساختار شبکه های Inception V3 و LeNet-5

برای پاسخ به این پرسش، ابتدا ساختار معماری LeNet-5 را بررسی و شرح داده و سپس اقدام به مطالعه‌ی شبکه Inception V3 می‌کنیم تا بتوانیم مزایای موجود در آن را بهتر نشان دهیم.

شبکه LeNet-5 اولین بار در سال ۱۹۹۸ و توسط چهار محقق شناخته شده‌ی حوزه شبکه‌های عصبی معرفی شد که سرپرستی این پژوهش بر عهده Yann LeCun بود که نیز به افتخار آن اسم شبکه شامل دو حرف اول نام خانوادگی وی قرار گرفته است.[\(لینک مقاله اصلی\)](#)

در آن پژوهش هدف توسعه یک شبکه یادگیری با آموزش گرادیان محور بود تا بتوان کاراکتر‌های تصویری را شناسایی نمود. تصویر این شبکه در عکس زیر قابل ملاحظه است.[\(مرجع عکس همان مقاله چاپ شده می‌باشد\)](#)



لایه ورودی شبکه LeNet-5 یک تصویر با ابعاد  $32 \times 32$  پیکسل و در یک کanal می‌باشد لذا در شبکه معرفی شده تصاویر RGB هدف نبوده است. لایه‌ی بعدی این شبکه حاصل شش Convolutional دو بعدی با فیلتر های  $5 \times 5$  قابل آموزش روی لایه ورودی می‌باشد که با stride برابر با یک اعمال می‌شود لذا در پایان لایه‌ی دوم و ورودی لایه سوم، داده‌هایمان به شکل  $28 \times 28$  پیکسل با ۶ کanal (خروجی هر فیلتر یک کanal) حاصل می‌شود.

در مرحله بعدی یک عملیات sub-sampling با انجام Average Pooling با سایز  $2 \times 2$  با انجام میگیرد تا ابعاد ویژگی‌ها کاهش یابد؛ پس خروجی این مرحله و ورودی مرحله بعدی نیز با شکل همان ۶ کanal ولی ابعاد  $14 \times 14$  حاصل می‌شود. در این مرحله مجدداً ۱۶ فیلتر با سایز  $5 \times 5$  برای اعمال Convolutional دو بعدی با همان stride یک آموخته شده و حاصل ۱۶ کanal و با ابعاد  $10 \times 10$  پدید می‌آید. همانند دو مرحله قبل یک عملیات sub-sampling با انجام Average Pooling با سایز  $2 \times 2$  با انجام میگیرد تا ابعاد ویژگی‌ها مجدد کاهش یافته و به ۱۶ کanal با سایز  $5 \times 5$  برسد.

هم اکنون برای بار سوم  $120 \times 5 \times 5$  فیلتر برای Convolutional دو بعدی با همان stride یک آموزش داده می شود تا خروجی  $1 \times 1$  با  $120$  کanal حاصل شود. هم اکنون این  $120$  کanal بصورت flatten کنار هم قرار گرفته و بصورت تماماً متصل(fully-connected) با  $84$  نورون لایه‌ی بعدی لینک می شوند. این لایه که مسئولیت Classifier را بر عهده دارد، یاد می‌گیرد تا بتواند بر اساس ویژگی‌های استخراج شده داده‌های ورودی را دسته بندی کند. لایه‌ی نهایی نیز لایه‌ی خروجی و به تعداد برقسپ‌های مجموعه داده دارای نورون می باشد( $=10$ ): اعداد دست نویس) تا بتواند باتابع فعالیت softmax احتمال متعلق بودن به یک از کلاس‌ها را خروجی دهد.

همانطور که ملاحظه می شود این شبکه بسیار ساده بوده و با لایه‌های انگشت شمار تشکیل شده است اما با این حال توانسته است در زمان خود به نتایج ارزشمندی در استخراج و دسته بندی تصاویر دست یابد. با وجود این که شبکه LeNet-5 دارای معماری ساده و قابل فهم بوده و پیاده سازی چالشی در پی نداشته و حتی نتایج مطلوبی نیز دارد اما دارای نقاط ضعفی نیز می‌باشد. سه مورد از نقاط ضعف این شبکه عبارت است از:

۱. کanal ورودی: این شبکه صرفاً دارای یک کanal ورودی می باشد لذا برای تصاویر رنگی بایستی معادل gray-scale آن مد نظر گیرد. این مورد باعث می شود در مواردی که ویژگی‌های استخراجی سطح خاکستری نزدیک به هم به خوبی عمل ننماید. مثلا در پرسش بعدی برای دسته بندی بیماری برگ گیاهان خواسته شده است از شبکه مذکور استفاده نماییم ولی می‌بینیم چون رنگ زرد بیماری گیاهان با رنگ اصلی سبز گیاهان در فضای خاکستری بسیار نزدیک به هم است، نتایج دسته بندی چندان مطلوب حاصل نمی شود.

۲. ابعاد ورودی: همانطور که دیدیم ابعاد ورودی این شبکه برابر با  $32 \times 32$  بوده و هدف در پژوهش انجام شده در آن مقاله استخراج ویژگی از تصاویر low-resolution اعداد دست نویس است؛ ورودی با این سایز باعث می شود دسته بندی تصاویر high-resolution با چالش مواجه بوده و در صورت کاهش سایز، جزئیات را از دست بدهیم.

۳. ویژگی‌های پیچیده: ملاحظه شد که تعداد لایه‌های Convolution محدود و کم بوده و سایز فیلترها نیز تقریباً بزرگ است؛ این مورد باعث می شود ویژگی‌های پیچیده، کوچک و جزئی بصورت مطلوب استخراج و شناسایی نشود(مانند چشم در چهره) لذا این شبکه برای استخراج ویژگی‌های بزرگ می تواند بهتر عمل کند.

هم اکنون به بررسی شبکه InceptionV3 می پردازیم؛ برای مطالعه‌ی این شبکه از چند محتوای چند لینک استفاده کرده ام که این [لینک](#) دید و درک بهتری از شبکه مذکور و موارد مربوط به آن را به بندۀ داده است.

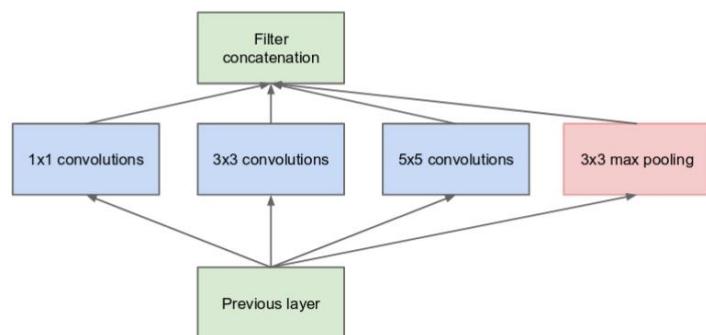
ویژگی‌های مورد نظر در تصاویر یک مجموعه داده برای دسته بندی می تواند در ابعاد مختلفی ظاهر شود. به عنوان مثال اگر تصویر زیر را در نظر بگیریم:



فرض کنیم می خواهیم ویژگی هایی از تصاویر فوق را استخراج کنیم که بتوانیم به وسیله‌ی آن وجود یک سگ در تصویر را شناسایی کنیم؛ همانطور که می بینیم در هر کدام از تصاویر سگ های حاصل در تعداد پیکسل های بسیار متفاوتی حاضر شده است و این باعث می شود ما با چالش انتخاب سایز فیلتر برای Convolution مواجه باشیم. اگر سایز فیلتر را بزرگ در نظر بگیریم آنگاه از تصاویری مانند سمت چپ که پیکسل های زیادی از سگ وجود دارد، ویژگی های خوبی برای دسته بندی استخراج می شود(ویژگی های سراسری و بزرگ)؛ اگر سایز فیلتر را کوچک در نظر بگیریم آنگاه از تصاویری مانند سمت راست که پیکسل های کمی از سگ وجود دارد، ویژگی های خوبی برای دسته بندی استخراج می شود.(ویژگی های جزئی و کوچک)

از طرفی اگر بخواهیم با عمیق تر کردن یک شبکه عصبی کانولوشنی به چالش فوق پاسخ دهیم، چالش بیش برآذش و انتشار خطا در زمان یادگیری رخ می دهد؛ همچنین از طرفی بزرگ تر کردن یک شبکه عصبی کانولوشنی نیز باعث افزایش هزینه محاسباتی و پارامتر های یادگیری می شود.

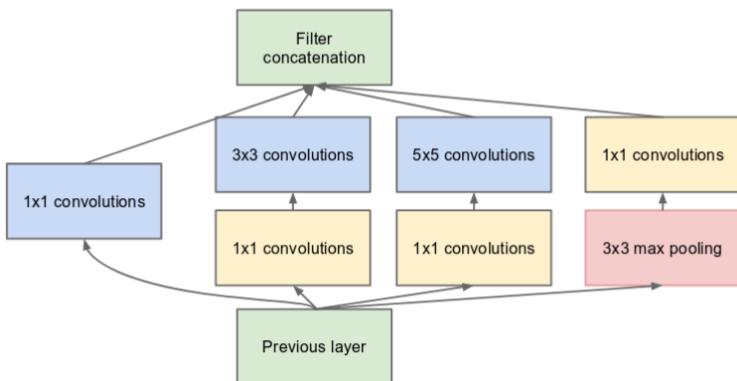
برای پاسخ به هر سه چالش فوق شبکه Inception ورژن یک معرفی و پیشنهاد شده است؛ ایده‌ی این شبکه در اعمال همزمان چندین فیلتر با سایز های مختلف در یک سطح می باشد. در شبکه‌ی مذکور یک مازول به نام Inception معرفی می شود که دلیل انتخاب نام شبکه نیز بخارط استفاده از این مازول ها در ساختار خود می باشد. مازول اولیه Inception در تصویر زیر آمده است:



همانطور که گفته شد این مازول بروی ورودی خود ۳ فیلتر با ابعاد مختلف برای استخراج همزمان ویژگی های کوچک و بزرگ اعمال می کند. ابعاد فیلتر های اعمالی به ترتیب برابر با  $1*1$  و  $3*3$  و  $5*5$  می باشد. به عنوان استخراج ویژگی یک sub-sampling نیز همراه

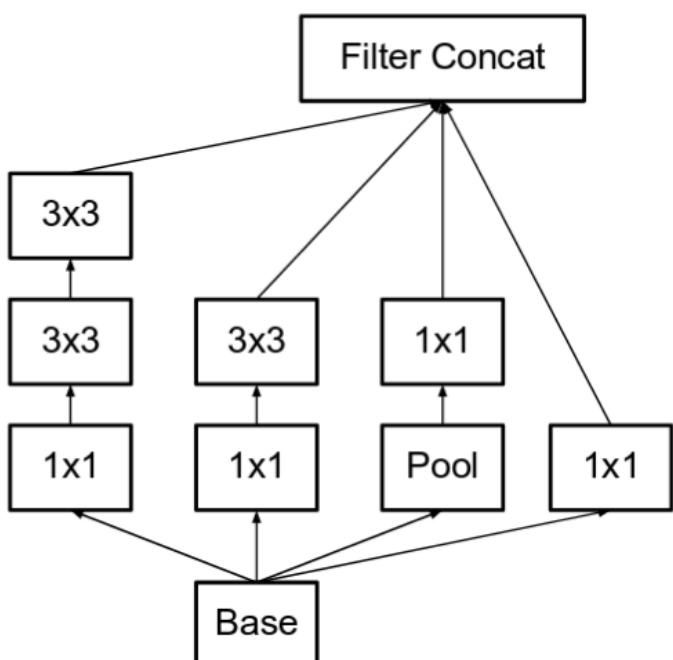
دیگر فیلتر ها انجام می شود تا بعد ویژگی های کاهش یافته نیز در استخراج ویژگی تاثیر داده شود. در نهایت خروجی این فیلتر ها باهم ادغام می شود.

در مازول فوق برای جلوگیری از افزایش هزینه محاسباتی و پردازش و کاهش بعد ویژگی، یک Convolution با سایز فیلتر  $1 \times 1$  قبل از انجام سایر فیلتر ها اعمال می شود؛ لذا مازول Inception بصورت زیر تغییر داده می شود:



همچنین استفاده متعدد از این مازول ها در معماری یک شبکه عصبی کانولوشنی باعث عمیق تر شدن و در نتیجه مشکل vanishing gradient می شود که در ویدیوهای ابتدایی یوتیوب تدریس یار محترم توضیح داده شده است. برای حل این مشکل نیز راهکار پیشنهادی استفاده از چند دسته بند کمکی در کنار دسته بند نهایی در شبکه می باشد بدین صورت که خروجی یک یا چند مازول Inception مستقیماً دسته بندی شود؛ همچنین loss کل شبکه نیز از ترکیب وزن دار loss ها شبکه بدست آید.

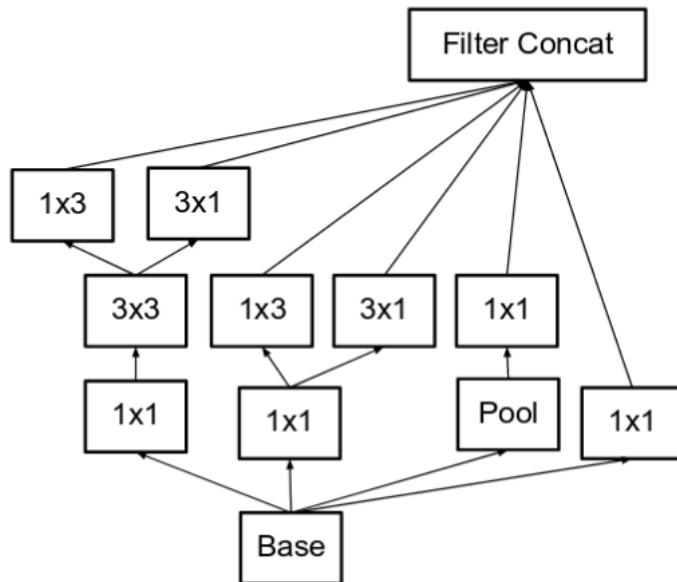
توضیحات فوق مربوط به Inception ورژن یک بود. ورژن دوم و سوم این شبکه همزمان باهم و در یک مقاله معرفی شد که درباره بهبود



مازول Inception Factorize کردن فیلتر های اعمالی در جهت کاهش زمان پردازشی و هزینه های محاسباتی می باشد؛ در آن مقاله اثبات می شود که می توان یک کانولوشن با سایز فیلتر  $5 \times 5$  را توسط دو فیلتر با سایز  $3 \times 3$  جایگذین نمود چرا که هزینه‌ی آن  $2.78$  برابر بیشتر است. لذا در مازول Inception فیلتر های  $5 \times 5$  با دو فیلتر  $3 \times 3$  مانند تصویر مقابل جایگذین شده اند.

دومین بهبود پیشنهادی در جایگزین کردن فیلتر های  $3 \times 3$  با دو فیلتر  $1 \times 3$  و  $3 \times 1$  می باشد؛ در آن پژوهش این مورد بصورت کلی تر نشان داده می شود که می تواند هر فیلتر  $n \times n$  را توسط دو فیلتر  $1 \times n$  و  $n \times 1$  توصیف کرد؛ لذا ماثول inception مجدد بصورت زیر تغییر

فرم می دهد:



محقق این پژوهش، بهبود های فوق را مربوط به ورژن دوم Inception دانسته و برای ورژن سوم، آپدیت های زیر را معرفی و در تکمیل ورژن دو پیشنهاد کرده اند:

- ✓ استفاده از RMSProp به عنوان بهینه ساز آموزش
- ✓ استفاده از Convolution ها با سایز فیلتر های  $7 \times 7$  برای استخراج ویژگی های کلی تر( بصورت Factorized تا مشکلصفحه قبل مجدد به وجود نیاید)
- ✓ استفاده از BatchNorm در دسته بند های کمکی به عنوان منظم ساز( زیرا در مقاله نشان دادند که این دسته بند های کمکی مگر در حد اشباع نهایی آموزش، کمکی در سایز زمان های آموزش انجام نمی دهد)

| type                 | patch size/stride or remarks | input size                 |
|----------------------|------------------------------|----------------------------|
| conv                 | $3 \times 3 / 2$             | $299 \times 299 \times 3$  |
| conv                 | $3 \times 3 / 1$             | $149 \times 149 \times 32$ |
| conv padded          | $3 \times 3 / 1$             | $147 \times 147 \times 32$ |
| pool                 | $3 \times 3 / 2$             | $147 \times 147 \times 64$ |
| conv                 | $3 \times 3 / 1$             | $73 \times 73 \times 64$   |
| conv                 | $3 \times 3 / 2$             | $71 \times 71 \times 80$   |
| conv                 | $3 \times 3 / 1$             | $35 \times 35 \times 192$  |
| $3 \times$ Inception | As in figure 5               | $35 \times 35 \times 288$  |
| $5 \times$ Inception | As in figure 6               | $17 \times 17 \times 768$  |
| $2 \times$ Inception | As in figure 7               | $8 \times 8 \times 1280$   |
| pool                 | $8 \times 8$                 | $8 \times 8 \times 2048$   |
| linear               | logits                       | $1 \times 1 \times 2048$   |
| softmax              | classifier                   | $1 \times 1 \times 1000$   |

✓ استفاده از Label Smoothing برای جلوگیری از بیش برازش  
جدول زیر از وبسایت رسمی [پایتورچ](#) می باشد که برای توصیف مشخصات لایه های Inception3 آورده شده است؛ همانطور که قابل ملاحظه است ورودی معرفی شده برای شبکه مذکور تصاویری با سه کanal و با ابعاد  $299 \times 299 \times 3$  می باشد:

## جمع بندی

بنابر توضیحات ارائه و مطالعه انجام شده نتیجه میگیریم شبکه Inception V3 بسیار قدرتمند تر از شبکه LeNet-5 بوده و مزایای متعددی برای استفاده از آن وجود دارد زیرا:

- ابعاد ورودی بزرگتر و high-resolution تری دارد لذا ویژگی های جزئی نادیده گرفته نشده و representation بهتری از تصویر اصلی را ورودی می گیرد.
- سه کanal ورودی دارد لذا می توان تصاویر رنگی را به آن ورودی داد. (مشکل همپوشانی رنگ های مختلف در سطح-gray (scale
- در آن از انواع سایز فیلتر(۱\*۱، ۳\*۳، ۵\*۵ و ۷\*۷) در Convolution های متعدد برای استخراج ویژگی مورد استفاده قرار می گیرد که این باعث می شود ویژگی های کوچک و جزئی توامان با ویژگی های سراسری و کلی استخراج شود.
- در آن لایه های عمیق بیشتری وجود دارد لذا ویژگی های پیچیده تری می تواند استخراج شود(البته طبق توضیحات صفحه قبل، برای جلوگیری از هزینه بالای محاسبات تکنیک های معرفی شده مورد استفاده قرار میگیرد)

## مسئله دوم

### مسئله دوم: توضیحات و پیاده سازی

برای پیاده سازی LeNet-5 از کتابخانه Keras استفاده شده است؛ برای این منظور در قسمت مشخص شده در سورس کد پیاده سازی با عنوان (Functions to make primary and custom LeNet) (Functions to make primary and custom LeNet) مدل LeNet-5 یا نسخه مدنظر از آن را ساخته و بر میگرداند؛ تابعی که نسخه ساده‌ی اصلی آن را می‌سازد در تصویر زیر آمده است:

```
def LeNetMaker(input_size=(32,32,1)):
    model = Sequential()
    model.add(keras.layers.InputLayer(input_size))
    model.add(keras.layers.Rescaling(1./255))
    model.add(keras.layers.Conv2D(6,kernel_size=(5,5),strides=(1,1),activation="tanh", padding='valid'))
    model.add(keras.layers.AveragePooling2D((2,2),strides=2))

    model.add(keras.layers.Conv2D(16,kernel_size=(5,5),strides=(1,1),activation="tanh", padding='valid'))
    model.add(keras.layers.AveragePooling2D((2,2),strides=2))

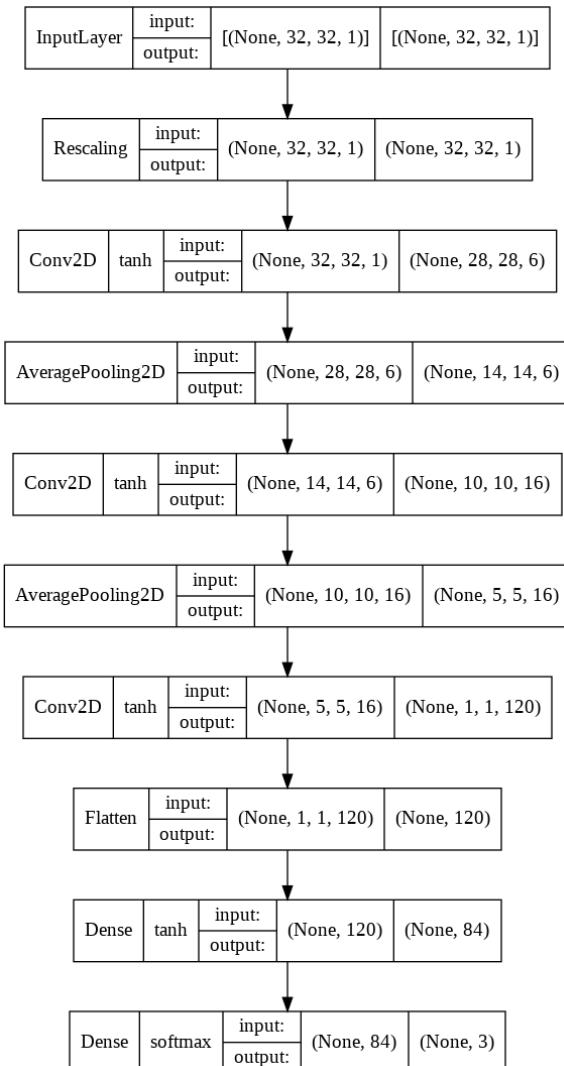
    model.add(keras.layers.Conv2D(120,kernel_size=(5,5),strides=(1,1),activation="tanh", padding='valid'))

    model.add(keras.layers.Flatten())

    model.add(keras.layers.Dense(84,activation="tanh"))

    model.add(keras.layers.Dense(3,activation="softmax"))

    return model
```

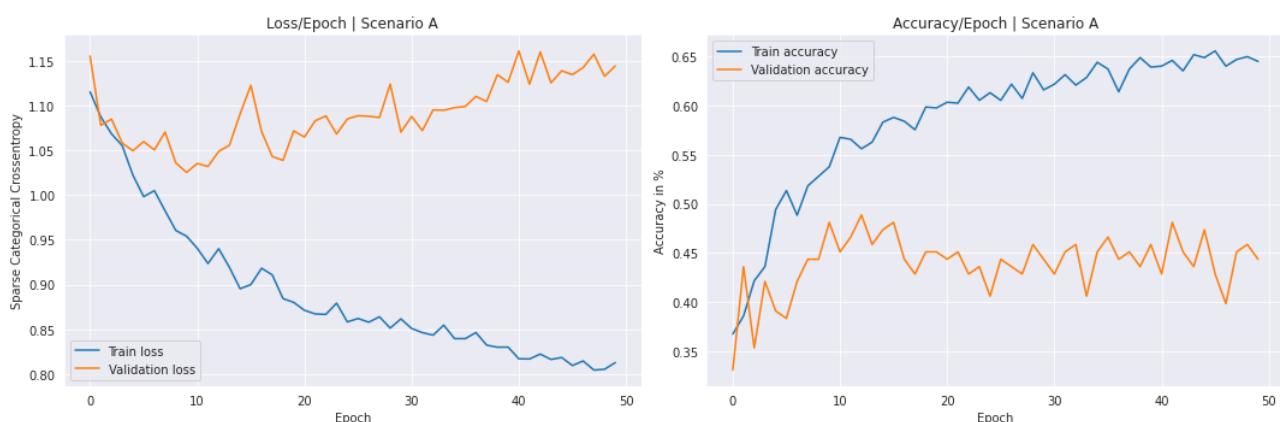
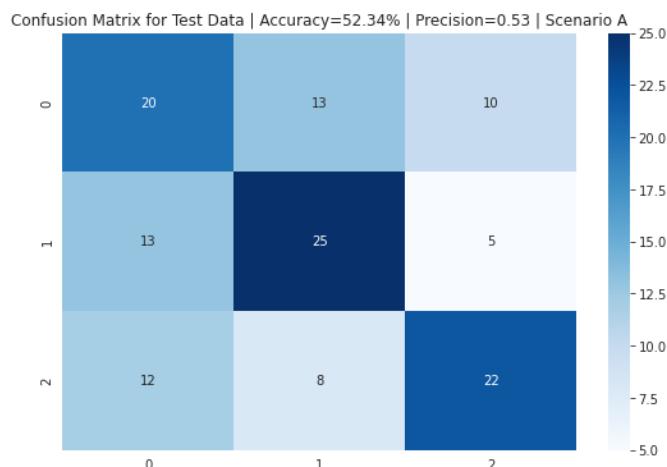


طبق مقاله‌ی اصلی، پیاده سازی فوق انجام شده و از تابع `tanh` برای تابع فعالساز لایه‌های کانولوشنی استفاده شده است؛ تصویر مقابل معماری پیاده سازی شده را نمایش می‌دهد. تنها تغییر انجام شده نسبت به مقاله، تعداد نورون‌های لایه‌ی آخر می‌باشد که بایستی به تعداد کلاس‌های مورد نظر باشد. در اینجا بر حسب مجموعه داده‌ی مفروض، سه نورون با ۱۰ نورون جایگذین شده است. بر حسب تجرب بدهست آمده از تمارین قبلی، در این آموزش‌ها از adam برای optimizer و از sparse categorical cross entropy برای خطا استفاده شده است. همچنین آموزش‌ها تا ۵۰ تکرار ادامه یافته‌اند و دسته‌های آموزش، اعتبار سنجی و آزمون نیز بر اساس تفکیک خود کتابخانه Tensorflow در نظر گرفته شده است.

شبکه LeNet-5 پیاده سازی شده را در چندین صورت خالص و با اعمال تغییرات در شبکه آموزش داده و به دسته بند رسیده ایم؛ هر آزمایش را در قالب یک سناریو شرح و نتایج آن را گزارش می‌کنیم.

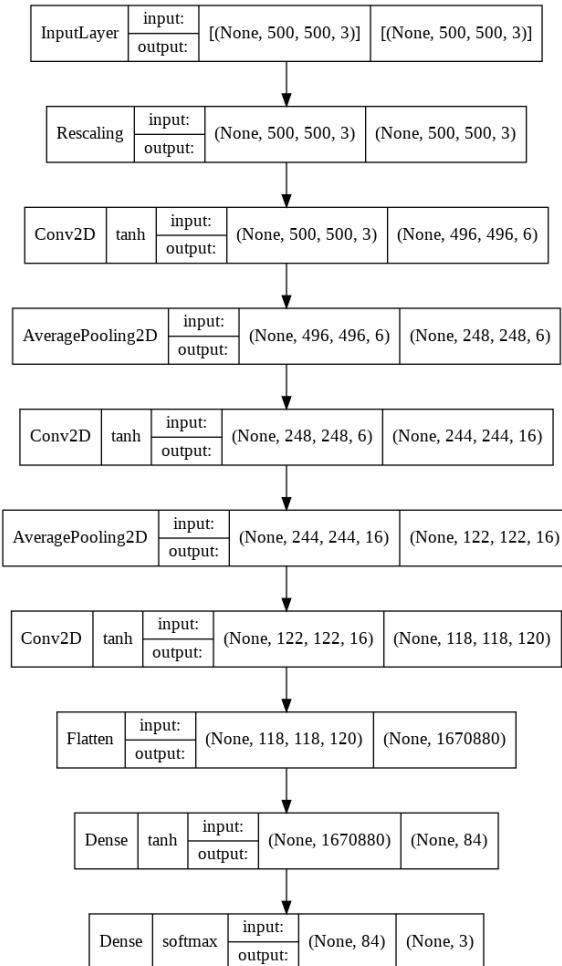
## مسئله دوم: سناریو A

در این آزمایش دقیقا شبکه LeNet-5 روی مجموعه داده ارزیابی می شود. تصاویر ورودی ما که  $32 \times 32 \times 1$  هستند را به  $500 \times 500$  تبدیل کرده و به شبکه ورودی میدهیم و سپس آموزش را انجام می دهیم. نتایج آموزش بصورت زیر حاصل می شود:

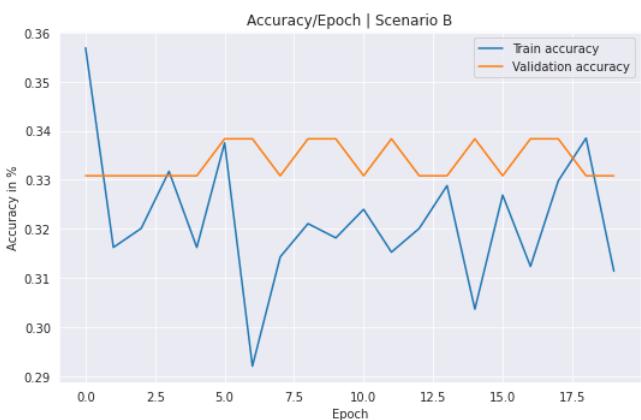
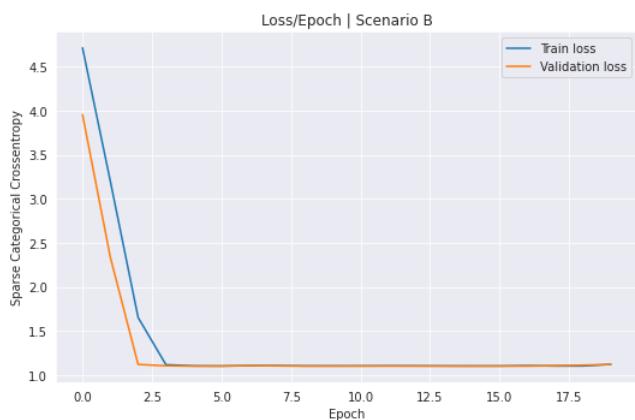
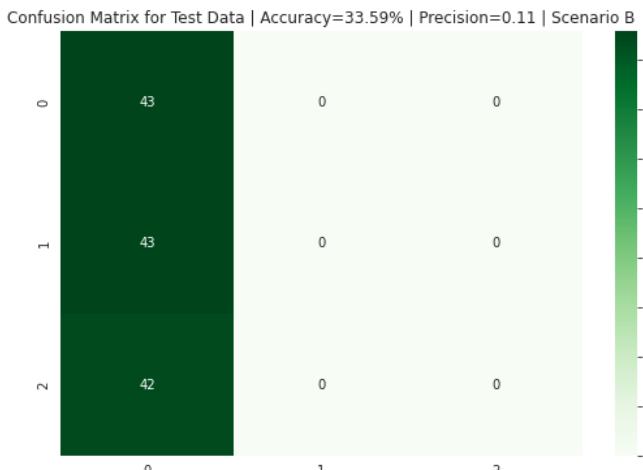


مالحظه می شود که LeNet-5 علاوه بر این که نتوانسته است به عملکرد خوبی در دسته بندی برسد، در همان دقت پایین نیز دچار بیش برآژش شده است چرا که نمودار های loss مربوط به آموزش و اعتبار سنجی از هم فاصله گرفته اند. دلیل این امر را می توان نقطه ضعف های مورد اشاره در پرسش یک برای LeNet-5 دانست زیرا ویژگی های اصلی تصاویر ما که high-resolution و در فضای RGB بوده و اساس دسته بندی بودند در فضای gray-scale و در سایز کوچک از بین رفته است.

## مسئله دوم: سناریو B

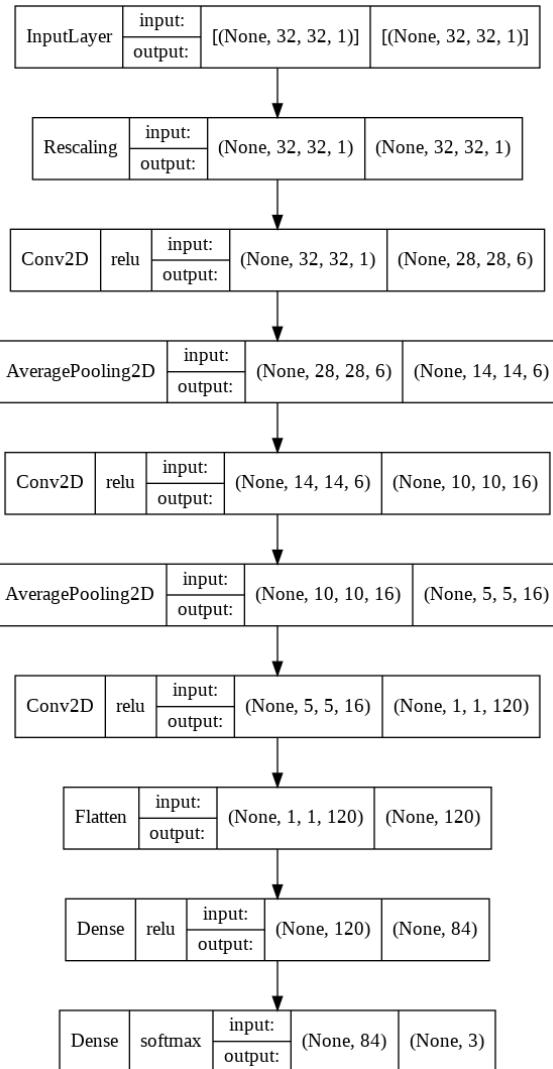


در این آزمایش شبکه LeNet-5 را با این تفاوت که ورودی ما سایز اصلی تصاویر مجموعه داده هستند آموزش می دهیم که معماری آن به همراه نتایج دسته بندی و آموزش در زیر آمده است:

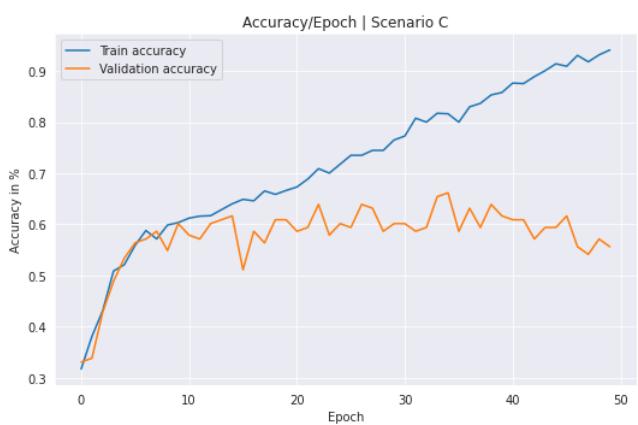
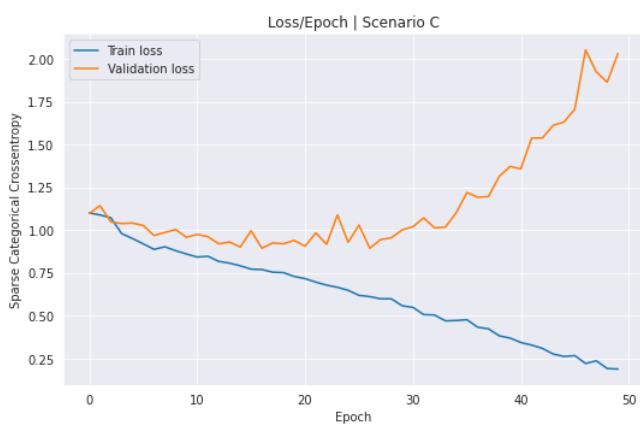
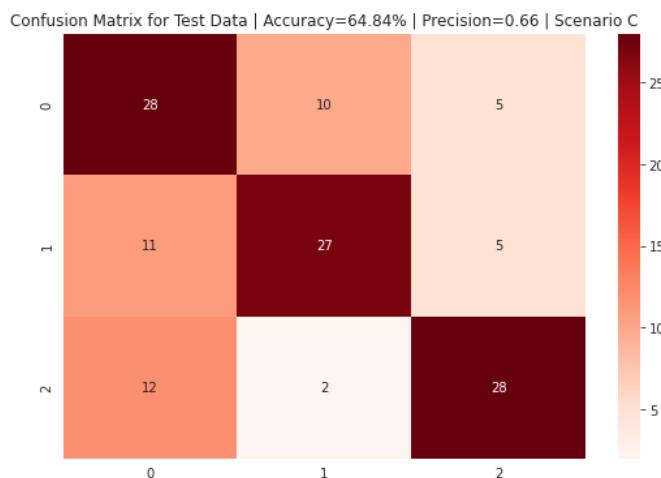


همانطور که مشاهده می شود، شبکه به هیچ عنوان نتوانسته است ویژگی استخراج کرده و دسته بندی بیاموزد؛ دلیل آن را می توان فضای ورودی بسیار بزرگ و گسستردہ در مقابل استخراج ویژگی ساده LeNet-5 با تعداد محدود لایه Convolutional توصیف کرد.

## مسئله دوم: سناریو C

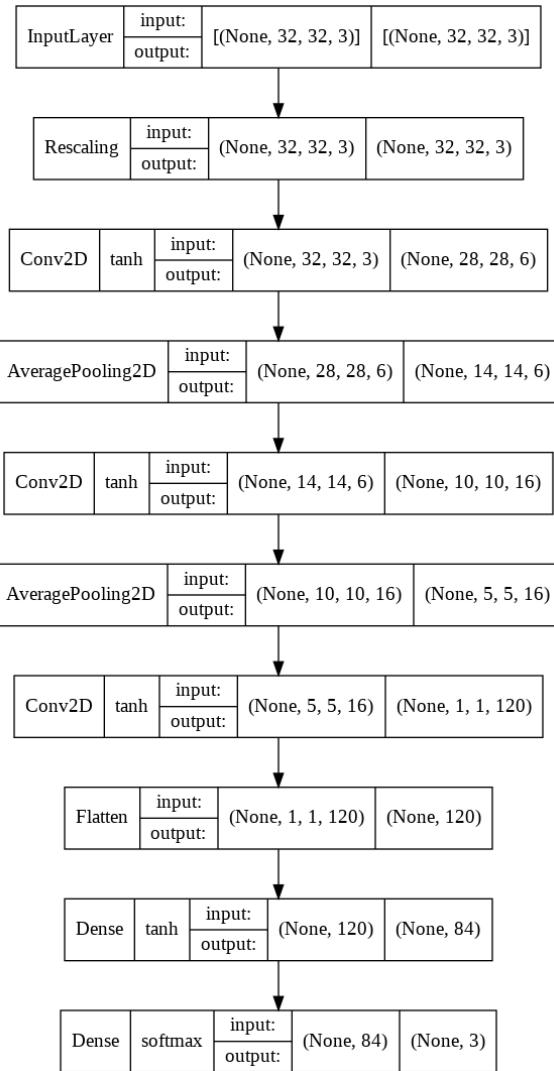


در این آزمایش شبکه LeNet-5 را با این تفاوت که از ReLU برای تابع فعالیت لایه های استفاده شده است آموزش می دهیم که معماری آن به همراه نتایج دسته بندی و آموزش در زیر آمده است:

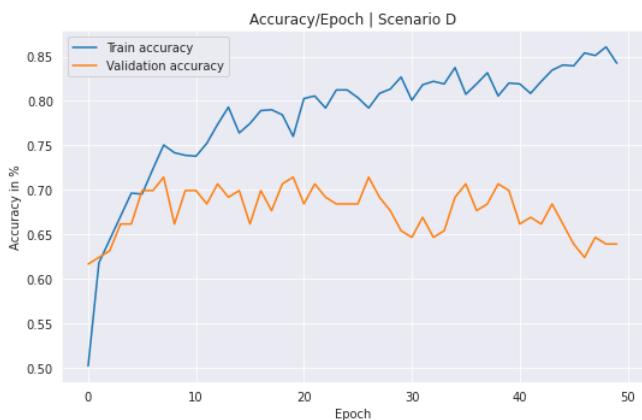
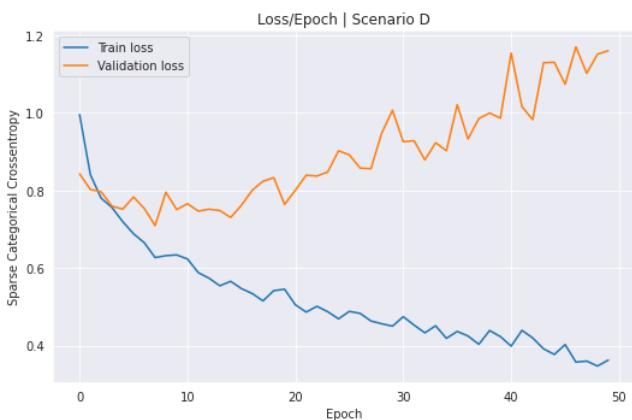
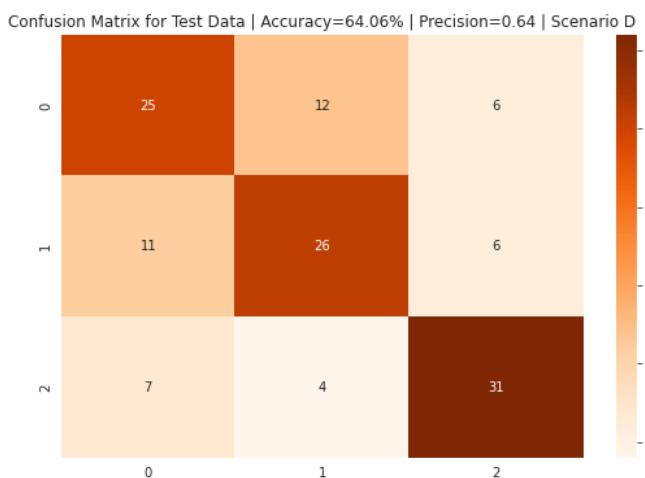


همانند نسخه اصلی، مشکل بیش برآش همچنان مطرح می باشد؛ اما ملاحظه می شود که دقت دسته بندی بالاتر رفته است زیرا می توان تبدیل تابع فعالیت از tanh به relu را یک بهبود مثبت در شبکه LeNet-5 برای مجموعه داده ای آموزشی مفروض نتیجه گرفت.

## مسئله دوم: سناریو D



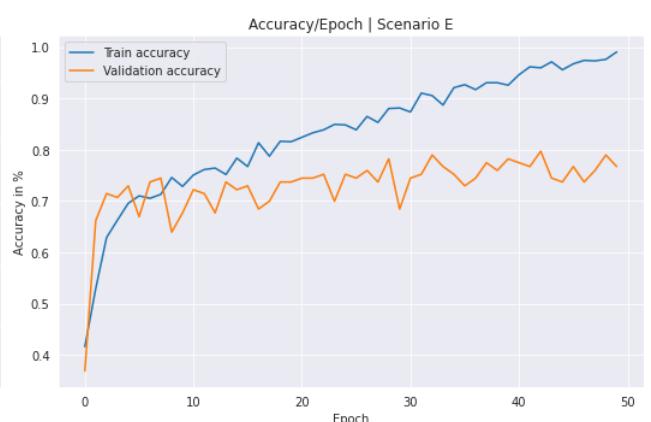
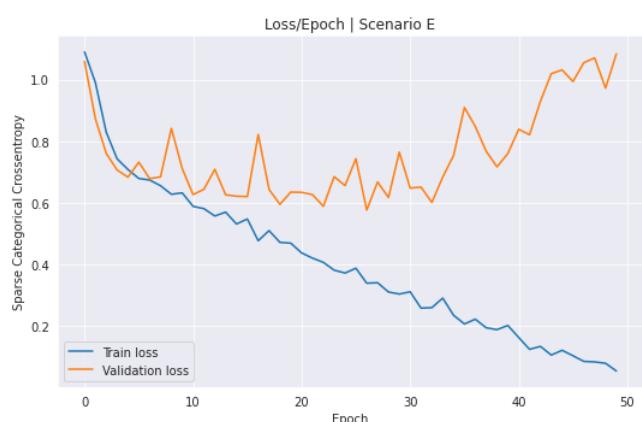
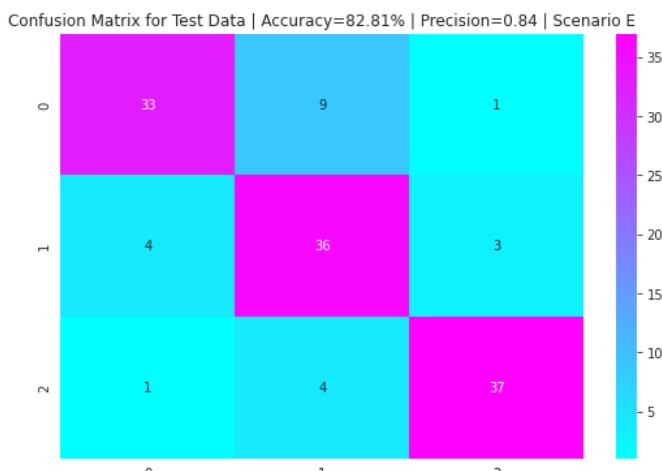
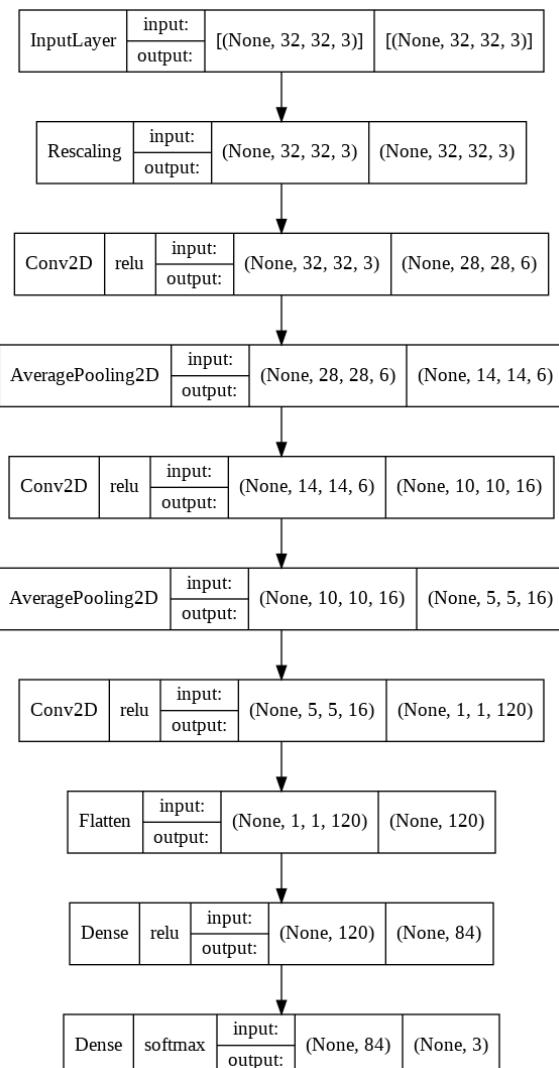
در این آزمایش شبکه LeNet-5 را با این تفاوت که از ۳ کanal RGB به جای یک کanal خاکستری استفاده شده است آموزش می دهیم که معماری آن به همراه نتایج دسته بندی و آموزش در زیر آمده است:



همانند نسخه اصلی، مشکل بیش برآش همچنان مطرح می باشد؛ اما ملاحظه می شود که دقت دسته بندی بالاتر رفته است زیرا می توان تبدیل تک کanal خاکستری به RGB را در زمان ورودی یک بهبود مثبت در شبکه LeNet-5 بروی مجموعه داده‌ی آموزشی مفروض نتیجه گرفت؛ چرا که قبل تراشاره شد چون رنگ زرد بیماری گیاه با رنگ سبز برگ در فضای خاکستری به هم نزدیک است، باعث کاهش کیفیت ویژگی‌های استخراجی و در نتیجه عملکرد دسته بندی می شود.

## مسئله دوم: سناریو E

در این آزمایش شبکه LeNet-5 را با این تفاوت که از سه کanal RGB به جای یک کanal خاکستری در حالی که از تابع فعالیت نیز به ReLU تغییر یافته است آموزش می دهیم که معماری آن به همراه نتایج دسته بندی و اموزش در زیر آمده است (ترکیب دو سناریوی قبلی):



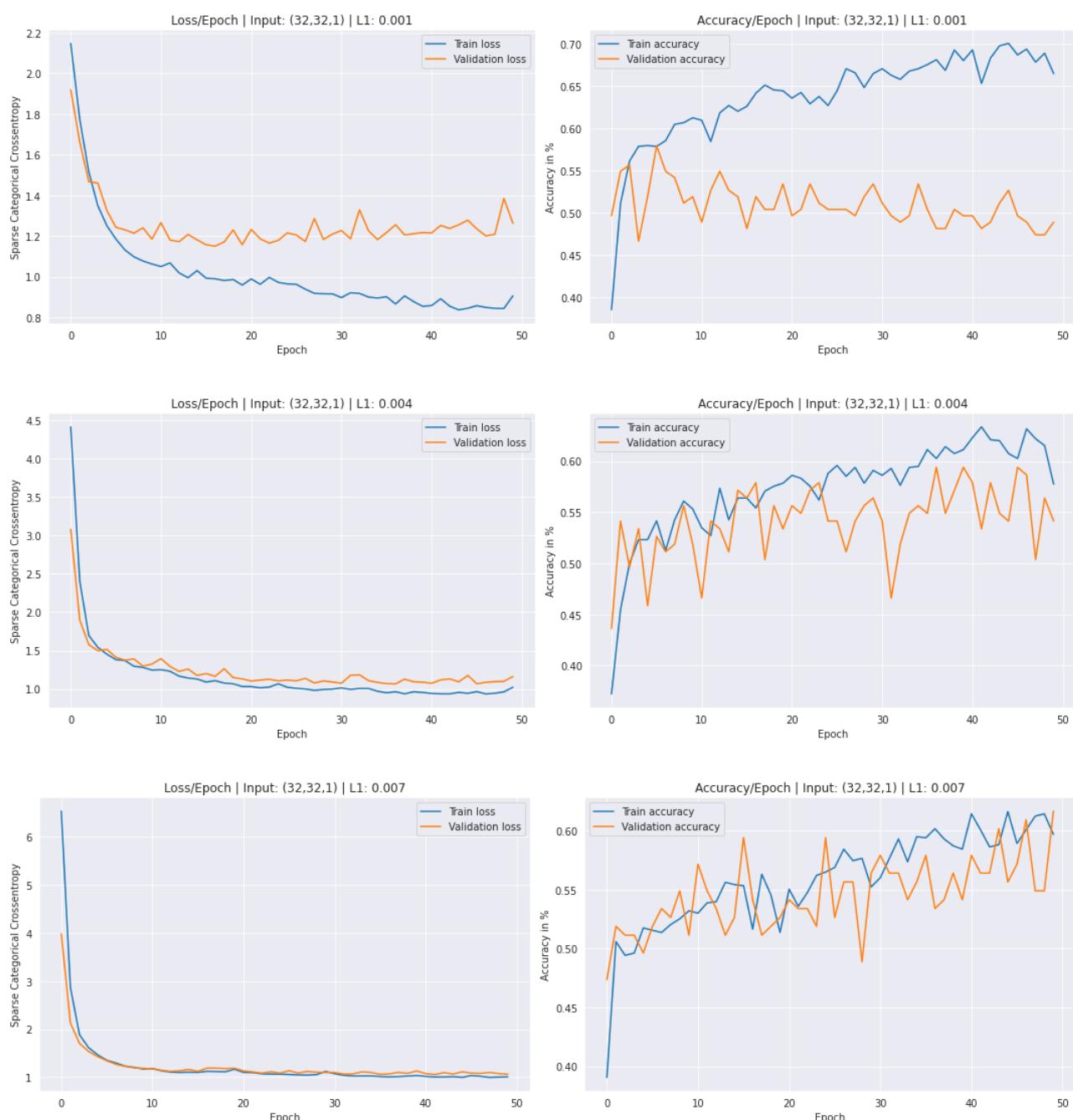
همانند نسخه اصلی، مشکل بیش برآش همچنان مطرح می باشد؛ اما ملاحظه می شود که دقیق دسته بندی بسیار بیشتر از همه سناریوهای قبلی بالاتر رفته است زیرا می توان تبدیل توامان تک کanal خاکستری به RGB و tanh بهReLU را یک بهبود عالی در شبکه LeNet-5 بر روی مجموعه داده‌ی آموزشی مفروض نتیجه گرفت که نسبت به حالت پایه تقریبا ۳۲ درصد افزایش دقیق پیش‌بینی برای داده‌های آزمون وجود داشته است. صرف نظر از آن که در هر دو بیش برآش رخ می دهد

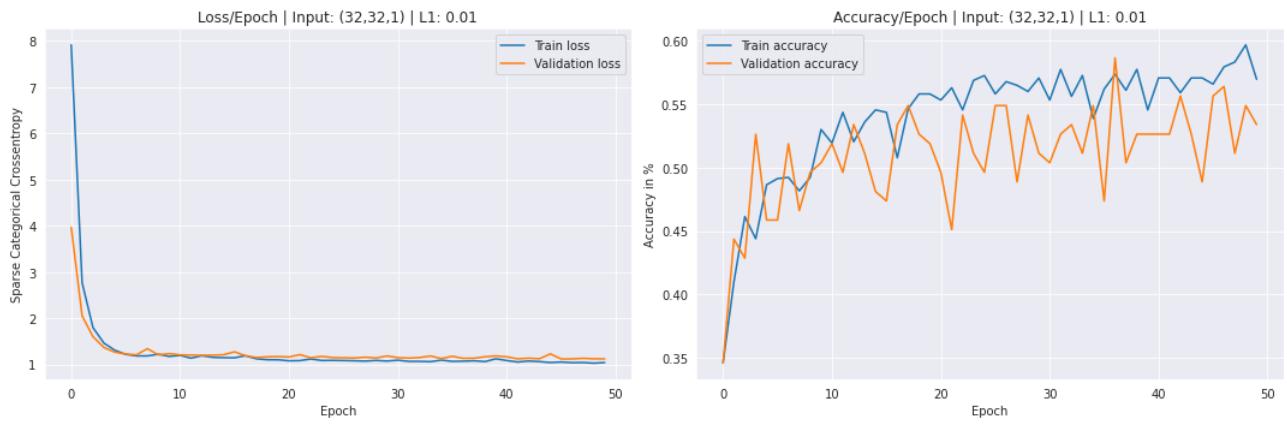
## مسئله دوم: مطالعه‌ی منظم‌ساز‌ها

در این بخش به ازای هر یک از سه منظم‌ساز خواسته شده، چند آزمایش با پارامترهای مختلف برای هر کدام انجام می‌دهیم تا بتوانیم تاثیر مربوطه را در مدل گزارش و ارزیابی کنیم. در این آزمایش‌ها از نسخه اصلی و بدون هیچ تغییر LeNet-5 استفاده می‌کنیم.

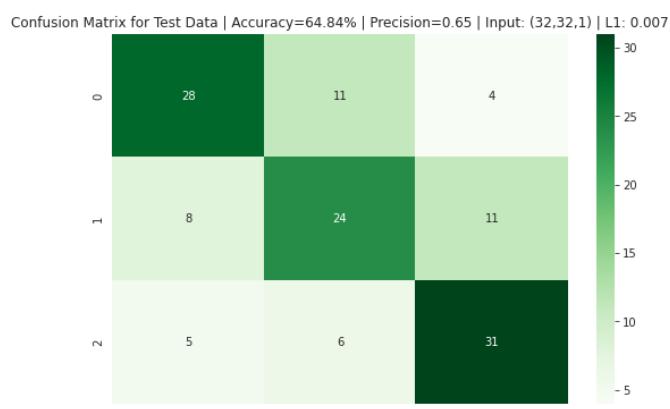
### مطالعه : L1

به ازای چهار پارامتر ۰.۰۰۱, ۰.۰۰۰۷, ۰.۰۰۰۴, ۰.۰۰۰۱ آموزش شبکه را با منظم‌ساز L1 انجام می‌دهیم که نمودارهای هر یک در ادامه آورده شده است:





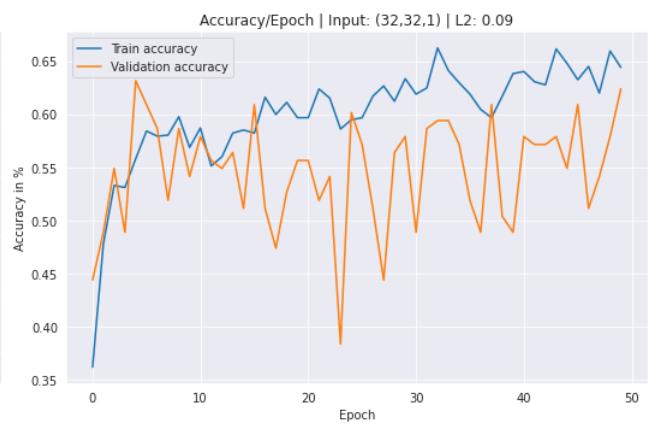
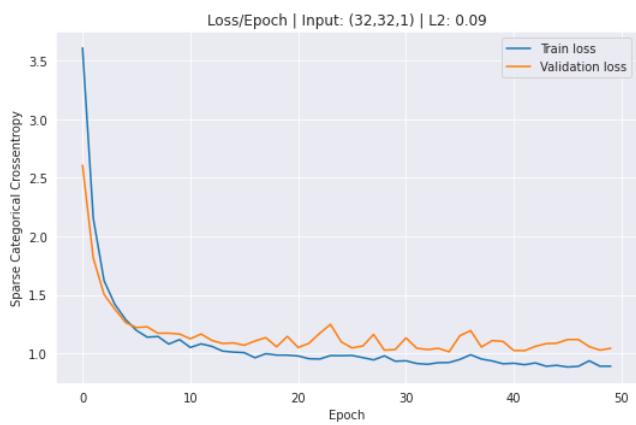
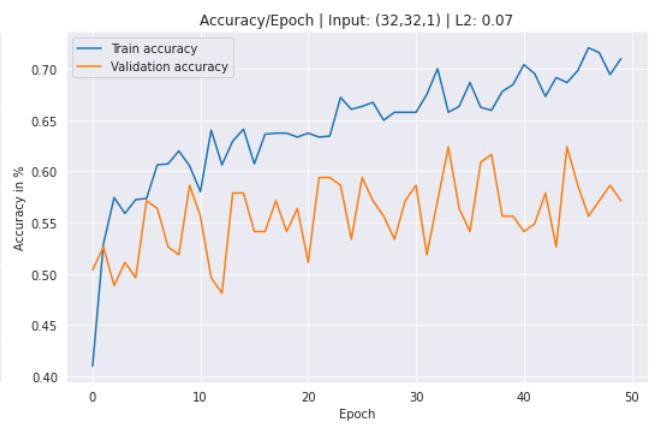
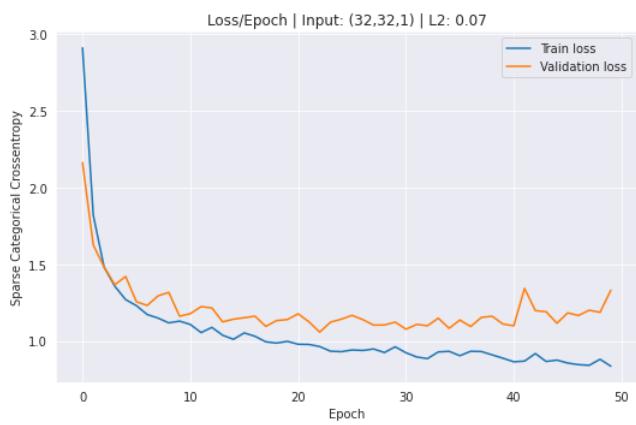
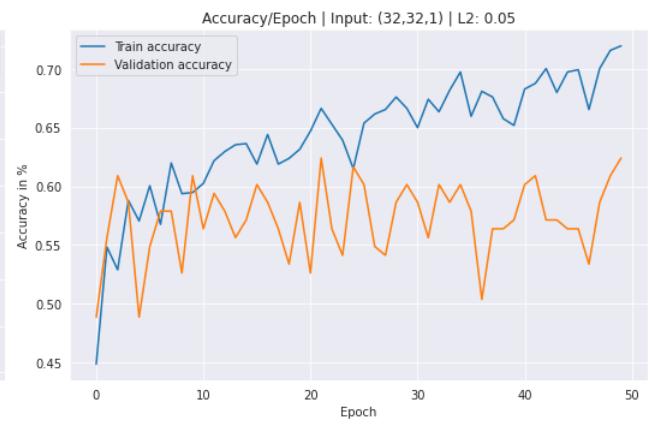
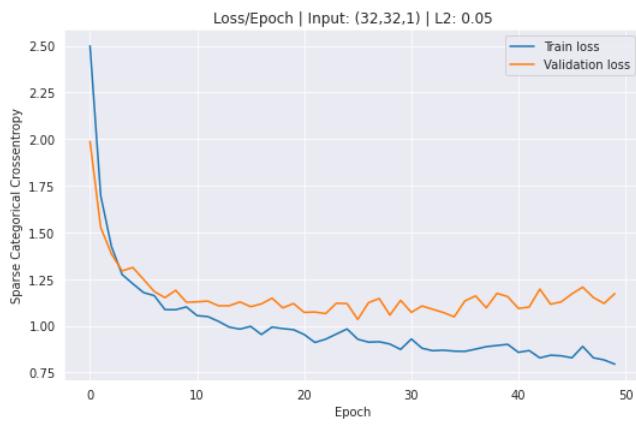
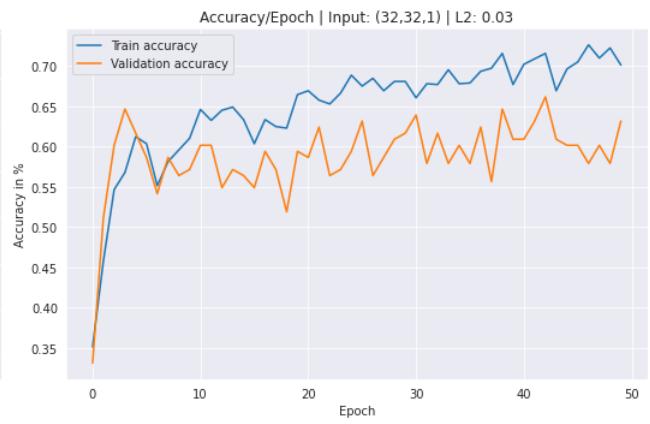
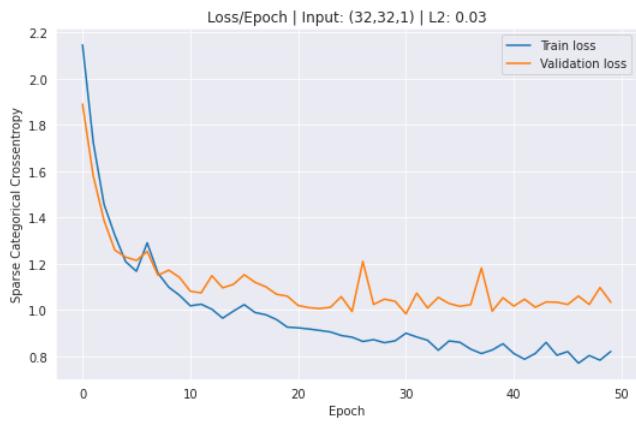
مشاهده می کنیم که منظم ساز L1 به ازای انتخاب پارامتر های مناسب می تواند مشکل بیش برآذش را در آموزش رفع نماید اما همچنان افزایش چندانی در عملکرد دسته بند را شاهد نیستیم. در این آزمایش ها بهترین نتیجه که دریافت شده است مربوط به پارامتر ۰۰۰۷ می باشد که گزارش دسته بندی آن در زیر آمده است:



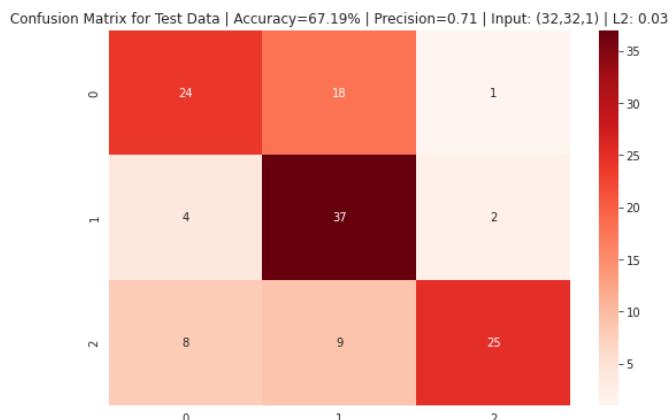
## مطالعه : L2

به ازای پنج پارامتر ۰۹، ۰۷، ۰۵، ۰۳، ۰۱ منظم ساز L2 آموزش شبکه را با دهیم که نمودار های هر یک در ادامه آورده شده است:



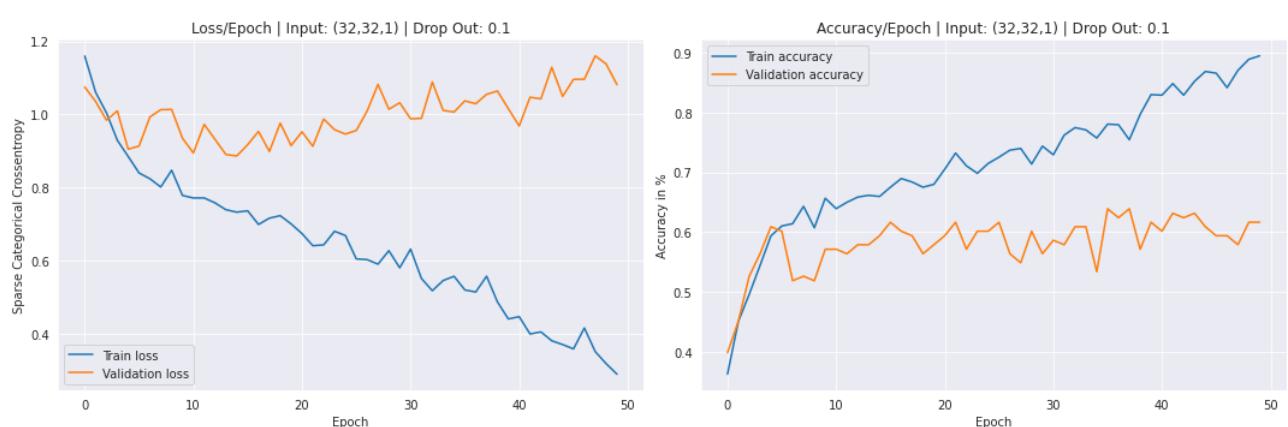


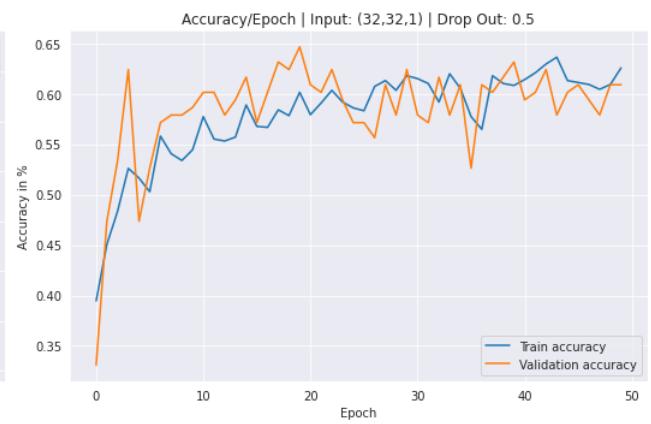
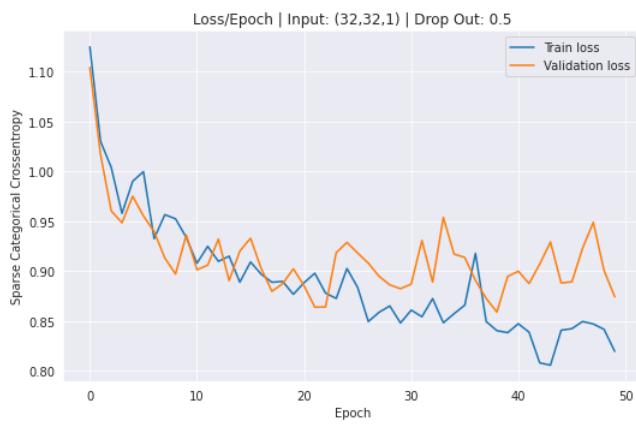
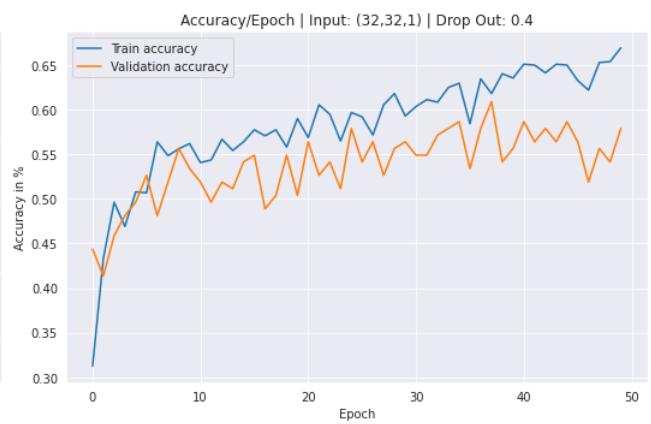
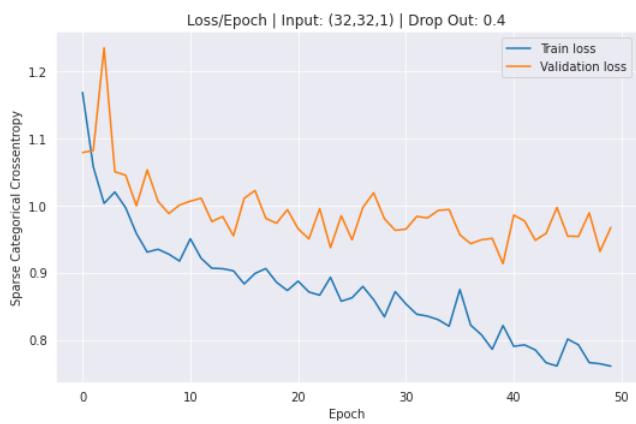
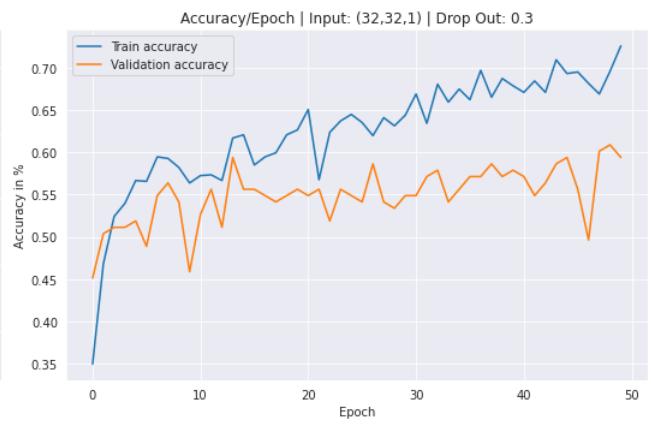
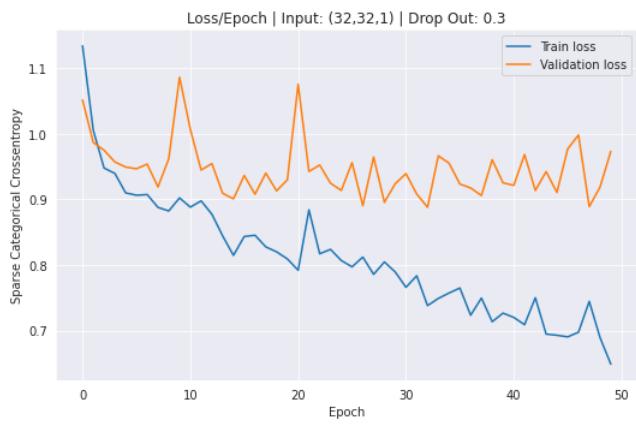
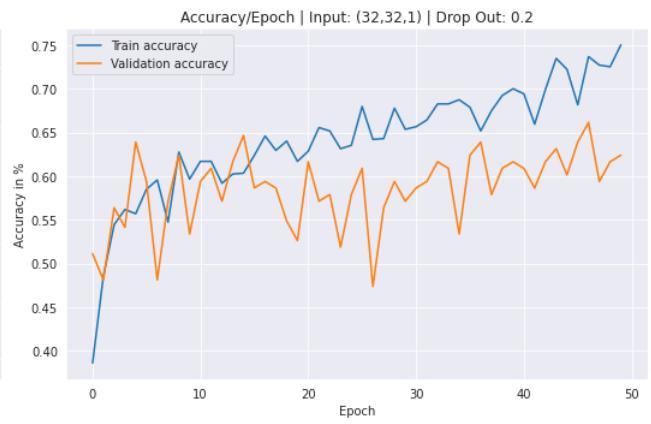
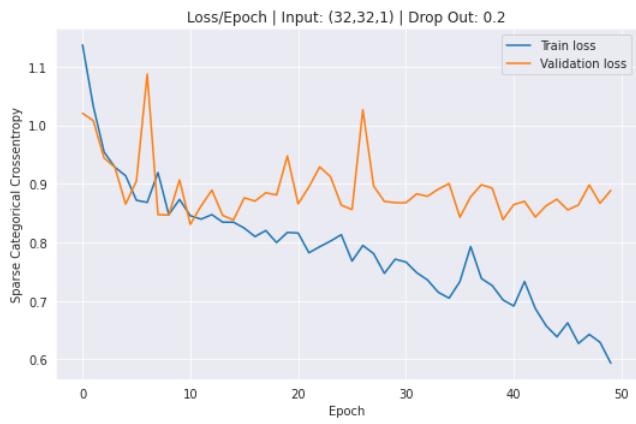
مشاهده می کنیم که منظم ساز L2 به ازای انتخاب پارامتر های مناسب می تواند مشکل بیش برآذش را در آموزش بصورت smooth رفع نماید و بر خلاف L1 می تواند عملکرد دسته بند را نیز بهبود خوبی دهد. در این آزمایش ها بهترین نتیجه که دریافت شده است مربوط به پارامتر  $0.003$  می باشد که گزارش دسته بندی آن در زیر آمده است و می بینیم که نسبت به حالت اصلی علاوه بر کنترل بیش برآذش، نزدیک  $15\%$  دقت پیش بینی داده های آزمون در دسته بند افزایش یافته است:

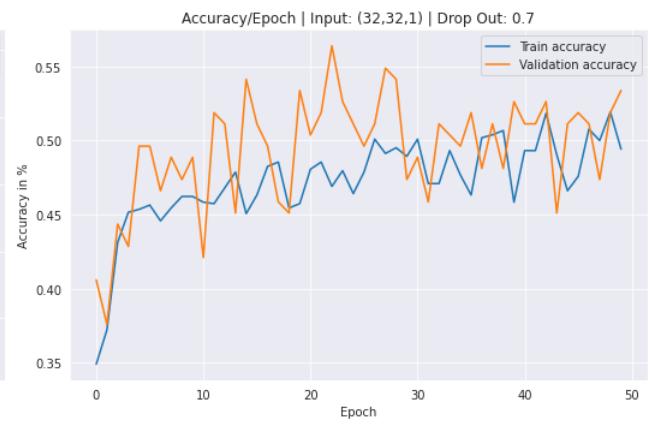
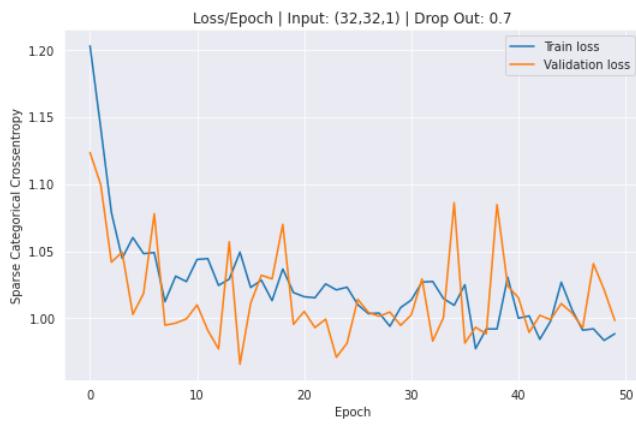
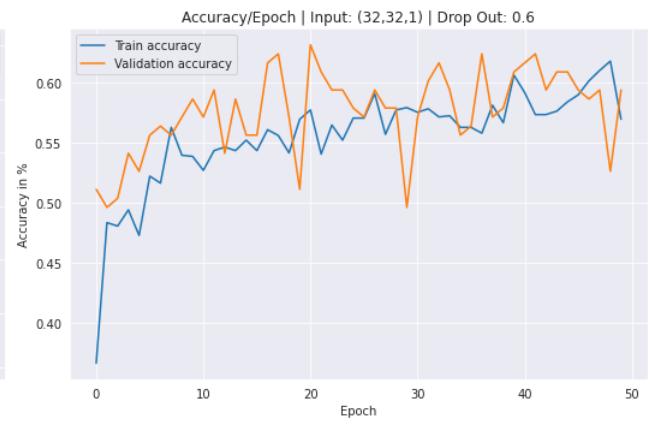


## مطالعه : Drop out

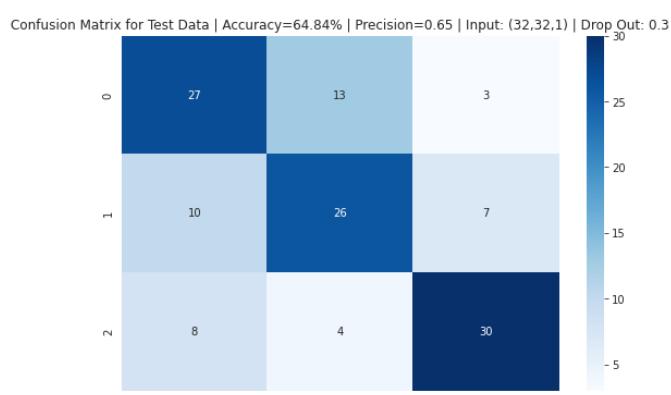
به ازای اعمال هفت drop outing برابر با میزان  $0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7$  آموزش شبکه انجام می دهیم که نمودار های هر یک در ادامه آورده شده است:







ملاحظه می کنیم که اعمال drop out در شبکه باعث کنترل تقریبی بیش برآذش شده اما همچنان باعث شده که آموزش نوسانی انجام گیرد لذا زمان بیشتری در این drop outting برای همگرایی مورد نیاز است. بهترین عملکرد مشاهده شده مربوط به drop out ۳۰٪ می باشد که گزارش عملکرد دسته بندی اش درزیر آمده است؛ مشاهده می شود که دقت بیش بینی داده های آزمون نسبت به حالت اصلی، تقریباً ۱۲٪ افزایش یافته است.



## مسئله دوم: مطالعهٔ تعداد کرنل‌ها

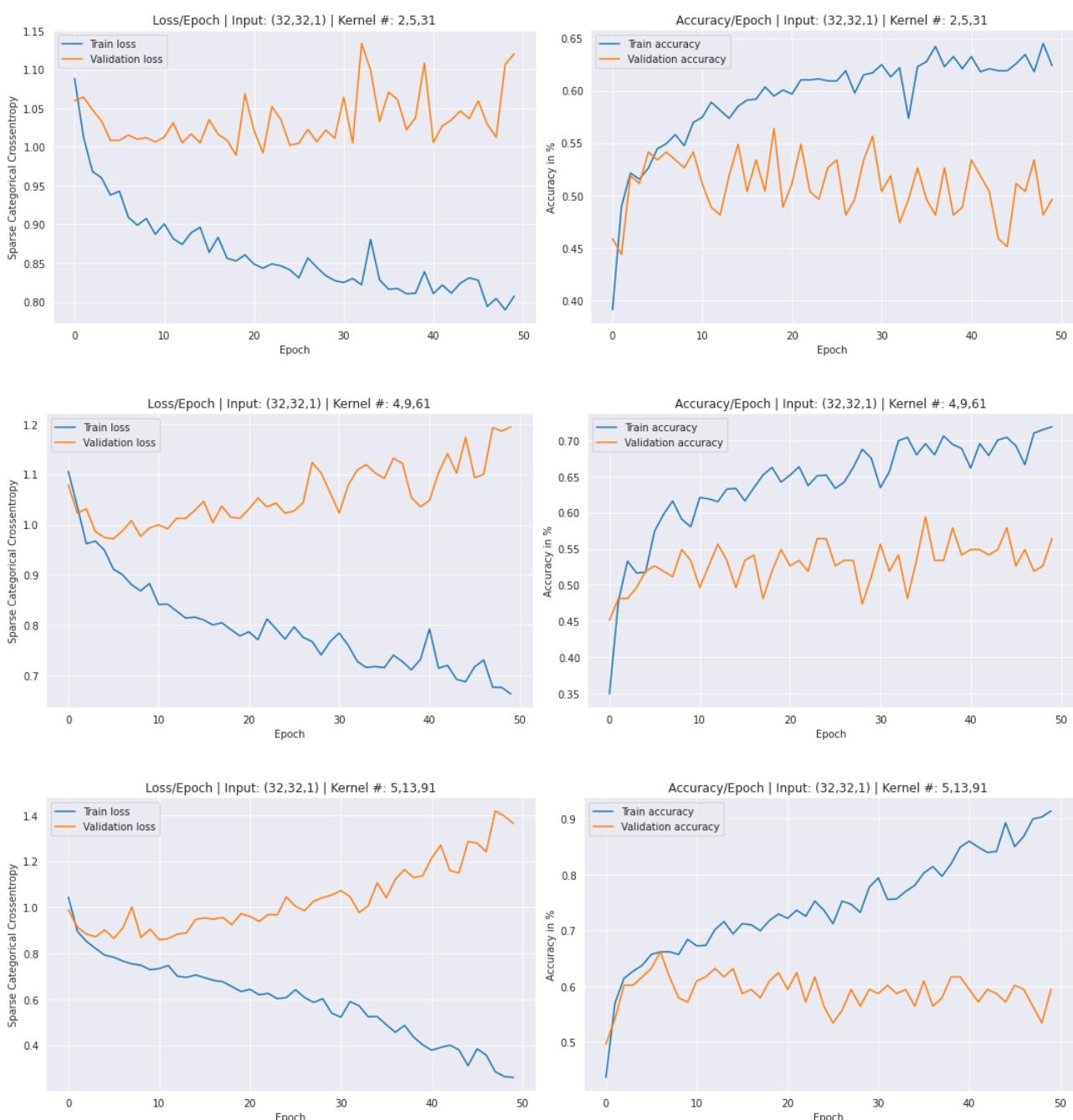
در این بخش به ازای ۵ ضریب مختلف تعداد کرنل‌های هر لایه را تغییر داده و آموزش را انجام می‌دهیم؛ این ضریب‌ها عبارت‌اند:

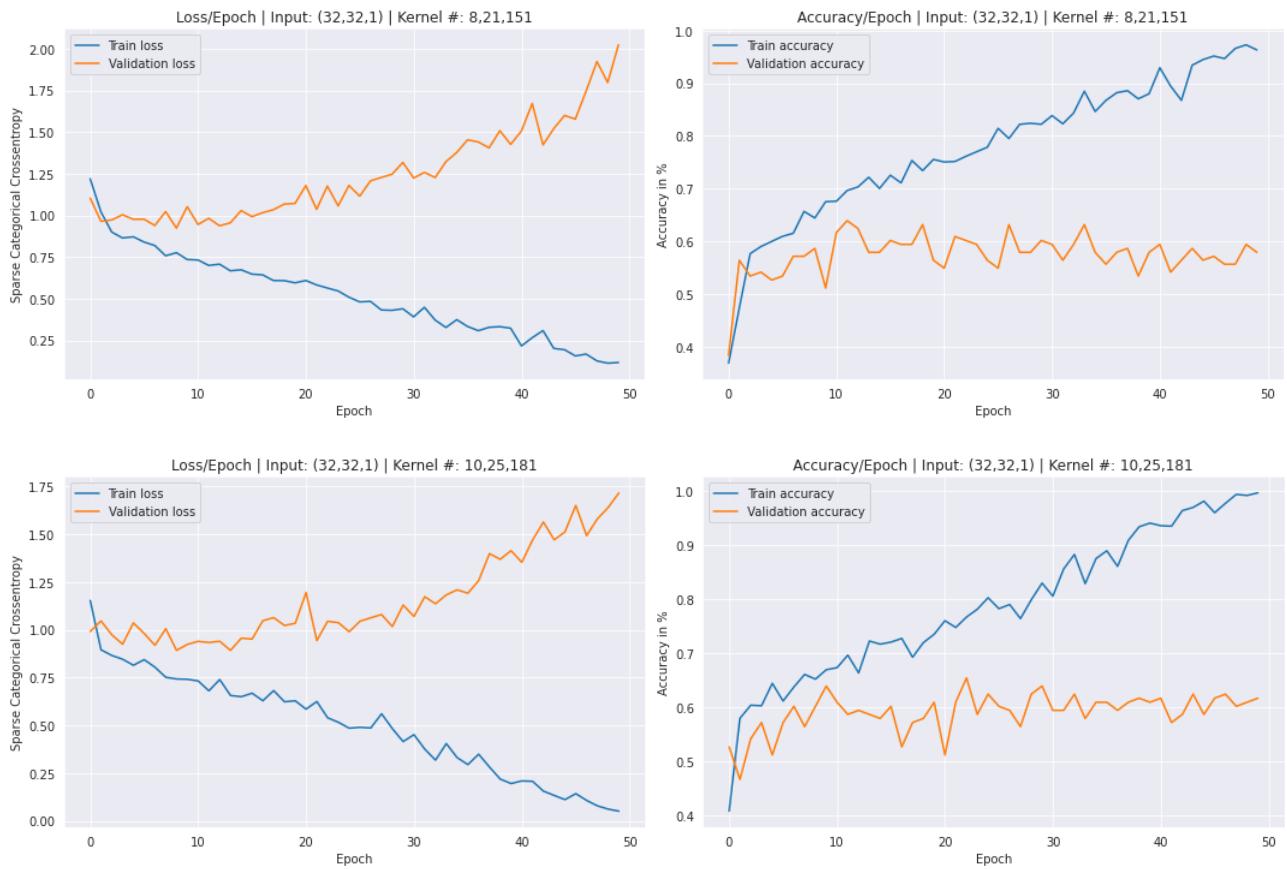
➤ کاهش‌های ۲۵٪ و ۵۰٪ تعداد کرنل‌ها

➤ افزایش‌های ۲۵٪ و ۵۰٪ تعداد کرنل‌ها

این تغییرات شامل لایه‌های کانولوشنی بوده و لایه‌ی دسته‌بند دچار تغییر نشده است؛ در قسمت title هر کدام از نمودار‌های آموزشی،

تعداد کرنل‌های از لایه از چپ به راست درج شده است:



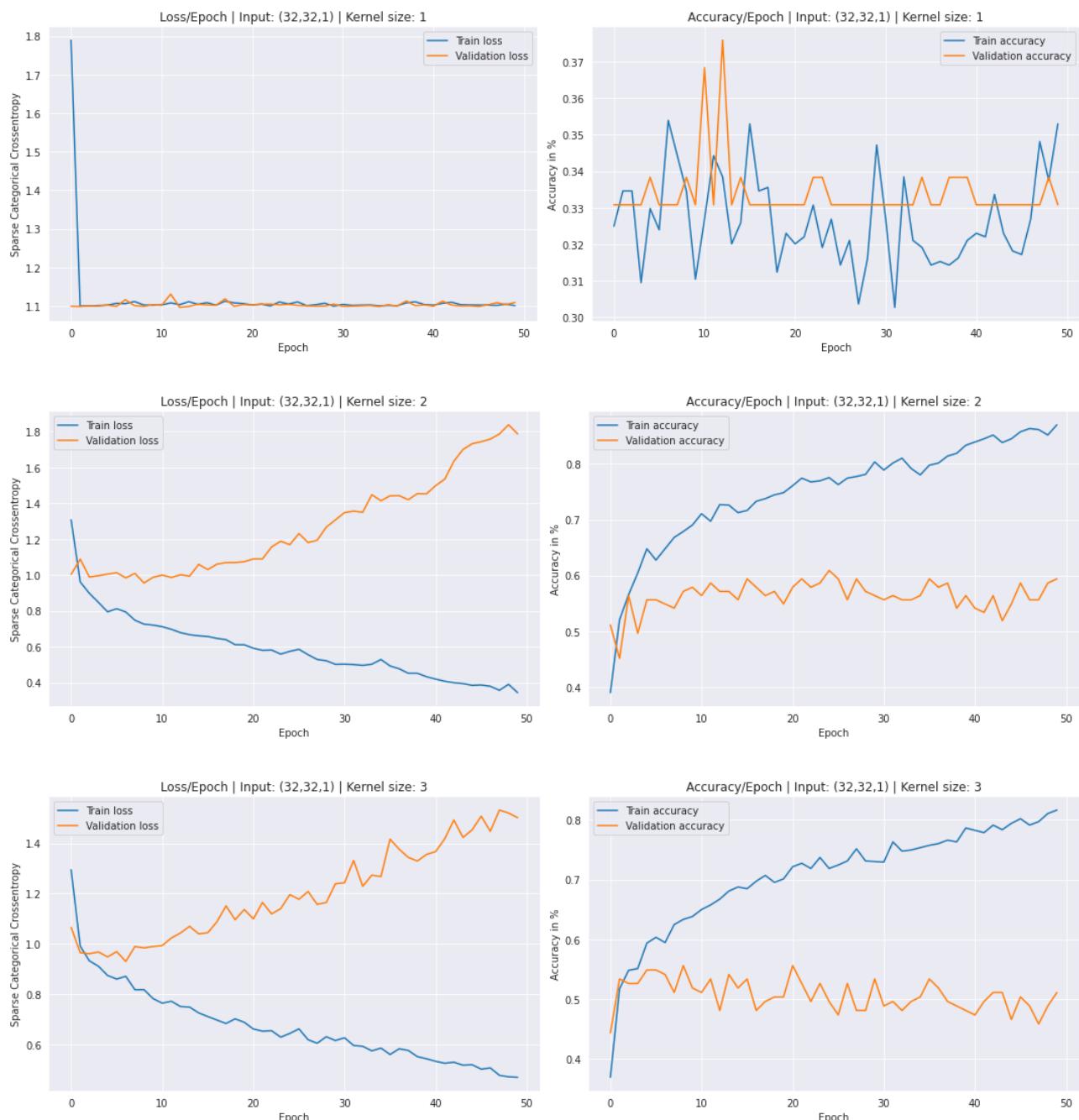


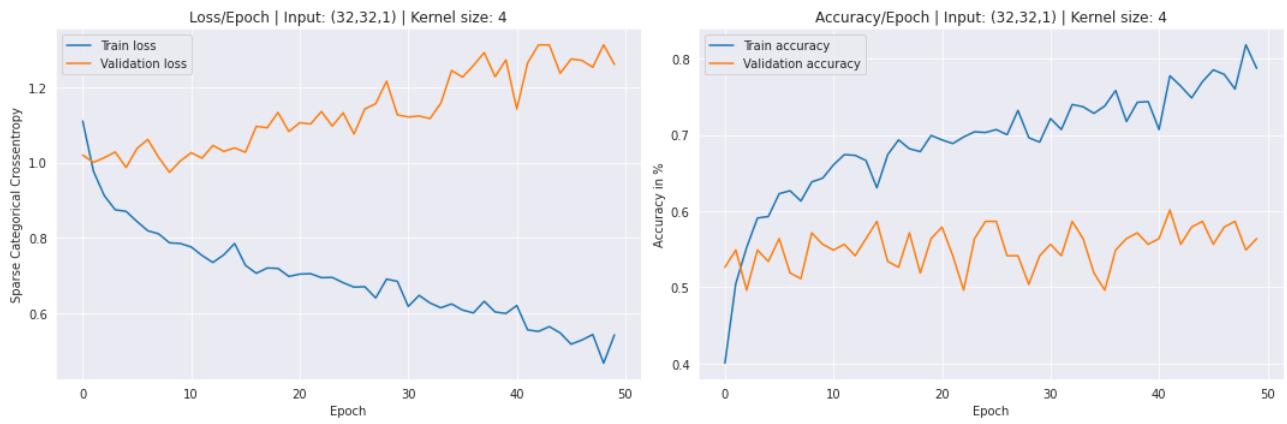
در آزمایش های فوق، اگر نمودار loss را در نظر بگیریم، مشاهده می کنیم که با افزایش تعداد کرنل های لایه های کانولوشنی، بیش برآش باشد. بیشتری انجام می پذیرد؛ همچنین اگر نمودار accuracy را مورد بررسی قرار دهیم، می بینیم که شدت بیش برآش بسیار زیاد می باشد. این نشان می دهد که ما نیازمند ویژگی های پیچیده تر هستیم نه ویژگی های بیشتر؛ زیرا دسته بند می تواند کاملاً ویژگی های استخراج شده را بیاموزد و بیش برآش کند، پس تعداد ویژگی ها برای این دسته بند کافی بوده و کاستی در پیچیدگی و کیفیت ویژگی های استخراج شده است و این را نمی توان با تعداد کرنل ها جبران نمود.

## مسئله دوم: مطالعه اندازه کرنل ها

در این بخش به ازای ۴ سایز مختلف شامل  $1*1$ ,  $2*2$ ,  $3*3$ ,  $4*4$  کرنل های هر لایه کانولوشنی را تغییر داده و آموزش را انجام می‌دهیم:

در قسمت title هر کدام از نمودار های آموزشی، سایز کرنل های کانولوشنی درج شده است:





بر اساس نتایج به دست آمده مشاهده می کنیم که در صورت استفاده از کرنل های با سایز  $1 \times 1$  دسته بند بر اساس ویژگی های استخراج شده هیچ آموزشی نمیتواند بینند؛ در خصوص سایر اندازه های نیز و با مد نظر قرار دادن فاصله خطوط نمودار های loss و accuracy می توانیم نتیجه بگیریم که با افزایش سایز کرنل ها، شدت بیش برازش اندکی کاهش می یابد.

دلیل آن که ابعاد بزرگتر از سایز  $5 \times 5$  مورد مطالعه قرار نگرفته است که این مورد است که برای لایه کانولوشنی آخر حداکثر سایز ممکن  $5 \times 5$  است که همان مقاله آن را استفاده کرده است و ما اگر بخواهیم سایز بزرگتر را اعمال کنیم بایستی از zero padding استفاده کنیم که در این صورت ابعاد لایه های دچار تغییر شده و شبکه از حالت پایه‌ی LeNet-5 خارج می شود.

### مسئله سوم: فرآیند انتقال یادگیری

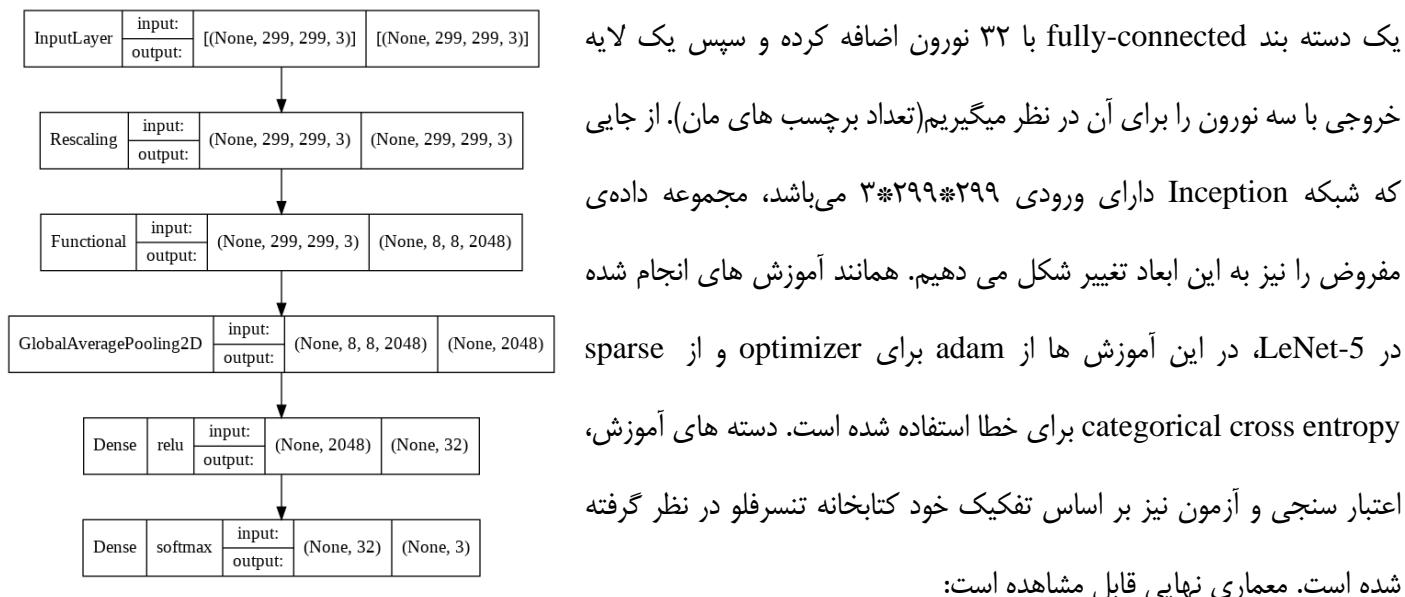
می دانیم که آموزش شبکه های عصبی قدرتمند و دقیق پر هزینه بوده و مستلزم آموزش مجموعه داده های بزرگ و منابع سخت افزاری قدرتمند می باشد؛ از این رو انتقال یادگیری اهمیت پیدا می کند. به طور خلاصه می توان گفت که انتقال یادگیری عبارت است از استفاده از وزن های یک شبکه عصبی آموزش دیده شده در فرآیند آموزش یک شبکه دیگر که ممکن است فضای ویژگی ورودی و حتی نوع ویژگی های استخراجی و دامنه برچسب متفاوت از هم داشته باشند.

می دانیم که شبکه های عصبی کانولوشنی(CNN) از دو بخش دسته بند و استخراج کننده ویژگی تشکیل شده است؛ در ابتدای CNN ابتدا لایه های کانولوشنی و لایه های مرتبط با آن نظیر کاهش بعد(Poolings)، استخراج ویژگی از تصویر را بر عهده داشته و پس از آن ویژگی های استخراج شده در اختیار یک دسته بند fully-connected قرار می گیرد تا آموزش رخ دهد. به عبارتی آموزش شامل دو بخش آموزش دسته بندی و آموزش استخراج ویژگی می شود. استخراج ویژگی از تصاویر(تصاویر RGB معمول) تا حد زیادی شبیه هم می باشد چرا که ذات استخراج ویژگی از تصاویر تقریباً یکی است.(لبه یابی، اتصال لبه، تشخیص الگو توسط لبه ها و...) طبیعتاً هر چقدر ویژگی های استخراجی قوی تر و با کیفیت تر بوده و حاوی اطلاعات پیچیده و سطح بالا باشد، دسته بندی و در نتیجه عملکرد پیش بینی بهتری برای داده های unseen حاصل خواهد شد. در انتقال یادگیری CNNs معمولاً مقصود آن است که قسمت استخراج ویژگی یک شبکه آموزش دیده را قدرتمند، به شبکه جدید منتقل شود و آموزش شبکه جدید سریع تر و با دقت بیشتر رخ دهد. برای این کار شبکه آموزش دیده را بازگزاری کرده و قسمت classifier آن را حذف کرده و قسمت استخراج ویژگی را فریز می کنند؛ به عبارتی تنظیم شان می کنند تا در زمان آموزش شبکه جدید تغییر و بروز نشوند(چون خطای دسته بند زیاد است و ممکن است انتشار این خطای کل وزن های لایه های کانولوشنی را خراب کند). سپس به انتهای شبکه یک classifier جدید با مشخصات مورد نظر اضافه می کنند و آموزش را انجام می دهند. در این صورت فقط دسته بند آموزش دیده و قسمت استخراج کننده ویژگی دچار تغییر نمی شود. پس آموزش دسته بند تقریباً سریع می باشد(زیرا قسمت استخراج ویژگی انتقالی قدرتمند بوده و ویژگی های حاصل از کیفیت خوبی برخوردار هستند) پس از آن که دسته بند آموزش دید، کل(یا قسمتی) از بخش استخراج کننده ویژگی را از حالت فریز خارج کرده و آنها را نیز قابل آموزش و تغییر می سازند(چون خطای دسته بند کاهش یافته است)؛ سپس آموزش را مجدداً از سر گرفته و دوباره ادامه می دهند. در این صورت بخش استخراج ویژگی نیز برای مجموعه داده ای جدید آموزش دیده و برای آن بهینه می شود. به طور خلاصه تر می توان گفت در انتقال یادگیری ابتدا با انتقال و ثابت نگه داشتن بخش استخراج ویژگی، قسمت دسته بند را آموزش می دهند، پس از آن با آزاد کردن بخش استخراج ویژگی، آن را نیز کاملاً با مجموعه داده ای جدید بهینه و شخصی سازی می کنند.

## مسئله چهارم

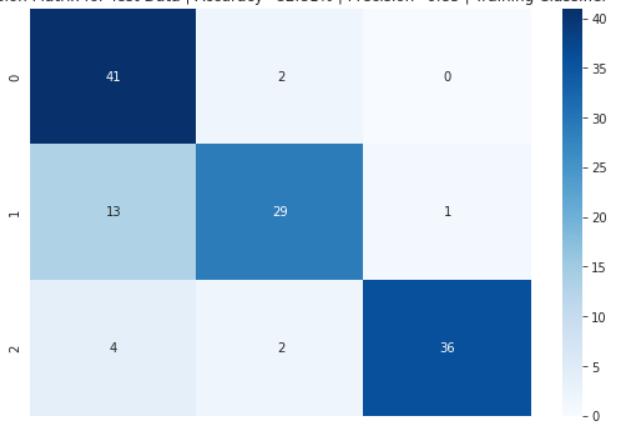
### مسئله چهارم: انتقال یادگیری V3

در قسمت توضیحات ارائه شده برای پرسش اول کاملا با شبکه و مژول Inception آشنا شده و در پاسخ گویی به پرسش سه نیز فرآیند انتقال یادگیری را به طور کامل آموختیم. همانند توضیحات ارائه شده مدل آموزش دیده شده Inception با مجموعه داده‌ی ImageNet را بدون دسته‌بندش بارگزاری می‌کنیم؛ سپس به انتهای آن یک میانگین گیر سراسری دو بعدی (تبديل هر نقشه ویژگی به یک ویژگی) و



برای بررسی و یافتن تعداد لایه‌های فریز شده مناسب در انتقال یادگیری برای مجموعه داده‌ی فوق، شش آزمایش انجام می‌دهیم؛ در این شش آزمایش، در هر کدام ابتدا دسته بند اضافه شده به شبکه را با ۱۶ تکرار آموزش می‌دهیم و سپس در هر کدام از آزمایش‌ها، به ترتیب یک الی شش مژول Inception را از حالت فریز خارج کرده و مجدد ۸ تکرار آموزش را ادامه می‌دهیم (اساس ساختار شبکه بر اساس مژول‌های تشکیل دهنده Inception می‌باشد، لذا بررسی تعداد لایه‌های فریز شده یا خارج کردن آن از فریز مبتنی بر هر مژول معنی‌دار تر می‌باشد)؛ نتایج ارزیابی عملکرد این شش آموزش به ترتیب در ادامه آمده است؛ در قسمت title هر آموزش تعداد مژول‌های

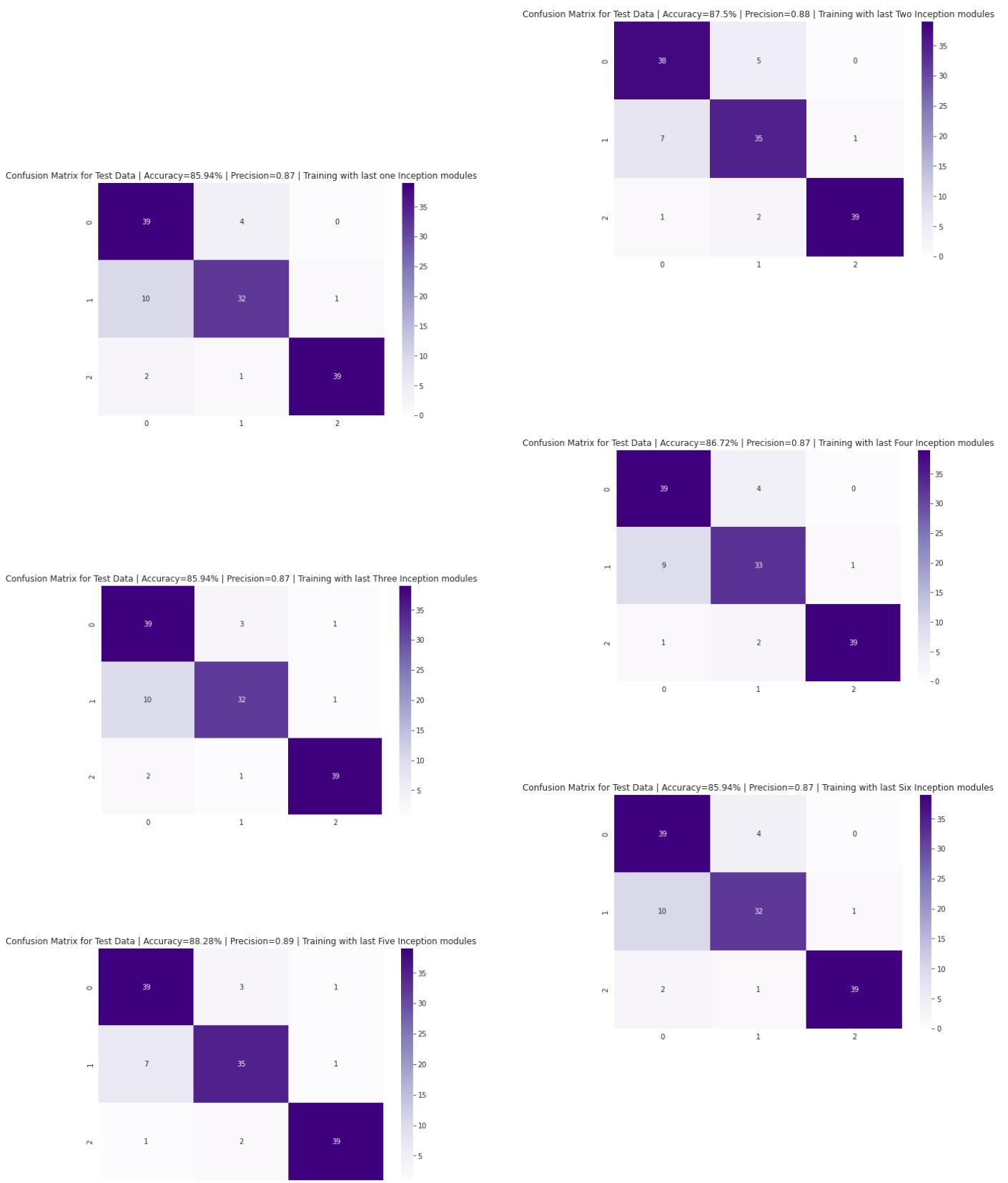
Confusion Matrix for Test Data | Accuracy=82.81% | Precision=0.85 | Training Classifier



خارج شده از حالت فریز درج شده است:

ارزیابی رو به رو مربوط به آموزش دسته بند بدون از فریز خارج کردن

هیچ لایه‌ای می‌باشد:

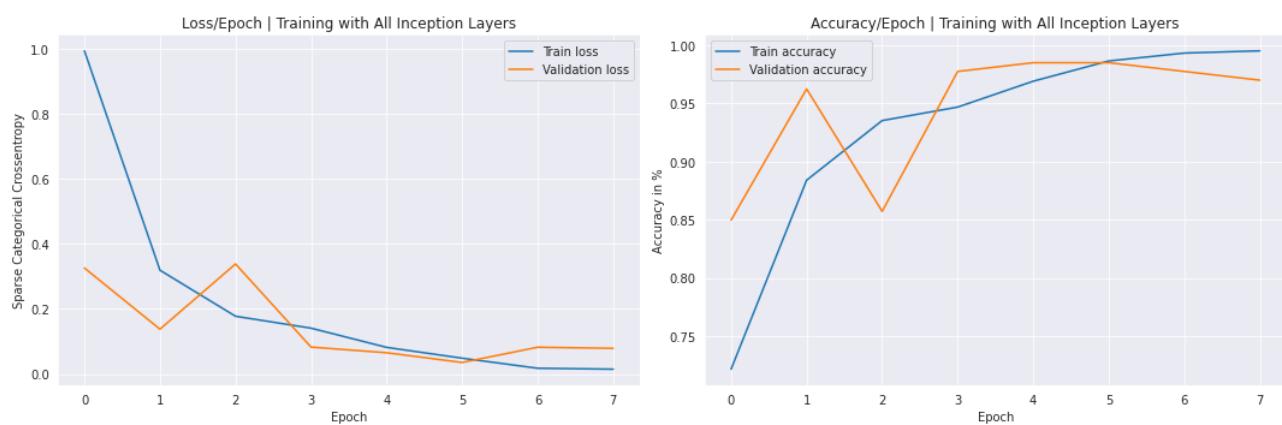
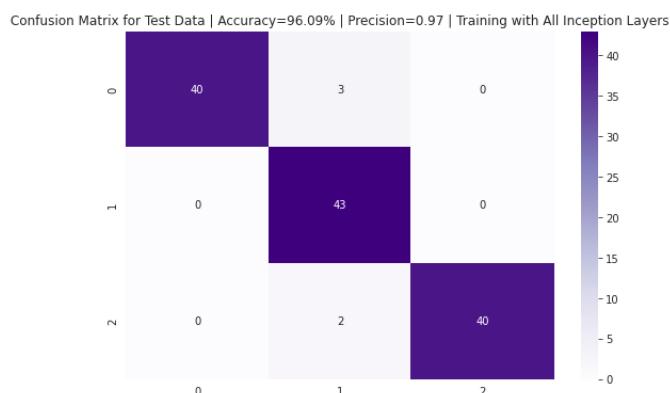


بر اساس نتایج حاصل از عملکرد دسته بندی در شش آزمایش بالا، مشاهده می کنیم که در صورتی که ۵ مارژول Inception انتهاي را پس از آموزش دسته بند از شبکه از حالت فریز خارج کرده و مجدد آموزش را ادامه دهیم، به بهترین نتیجه ممکن دست خواهیم یافت.

نکته بعدی قابل بیان آن است که تک تک آزمایش های انجام شده نتیجه بهتری نسبت به آموزش صرفا دسته بند دارد.

البته قابل ملاحظه است که تفاوت نتایج عملکرد نهایی دسته بند در آزمایش های فوق بسیار محسوس و چشم گیر نبوده و در رنج ۲-۴٪ می باشد. با این وجود و بر اساس سیر تغییرات عملکرد دسته بند نهایی، می توانیم نتیجه بگیریم مرز پنج ماثول نهایی برای از فریز خارج کردن و از سر گیری آموزش مرز مناسبی بوده و بهترین نتیجه ممکن را تا این مرحله حاصل می کند چرا که هم ماثول کمتر و هم ماثول بیشتر نتیجه پایین تری را دارد.

در یک آزمایش دیگر(هفتم) پس از آموزش دسته بند، همه لایه ها را از فریز خارج کرده و آموزش را از سر می گیریم که نتیجه آن بصورت زیر حاصل می شود:



این آزمایش که بهترین نتیجه را حاصل کرده است نشان می دهد در صورت از فریز خارج کردن کل لایه ها در زمان انتقال یادگیری(بخش دوم)، می توانیم به دقت ۸ درصد بالاتر و معادل با ۹۶٪ در پیش بینی داده های آزمون دست یابیم. نمودار های آموزش نیز نشان می دهد بیش برآذش یا کم برآذش نیز رخ نداده است. دلیل آن را می توان اینگونه گفت که فرآیند استخراج ویژگی یک فرآیند زنجیره ای بوده و در صورتی که قسمتی از آن فریز شده باقی بماند، این زنجیره ارتباط معنایی هر چه تمام خود را اندکی از دست می دهد لذا در صورت امکان

در فاز دوم اگر کل لایه ها دوباره آموزش داده شود(از فریز خارج شود) نتیجه بهتری حاصل می شود. البته این نتیجه گیری برای این مساله و شرایط در نظر گرفته شده معنی دار بوده و برای یک مساله دیگر میتواند صحیح نباشد.

## جمع بندی و مقایسه با LeNet-5

همانطور که از نتایج بدست آمده مشهود و عیان است، مدل Inception ورژن سوم به جهت دقت دسته بندی با اختلاف زیاد برتر می باشد که دلیل آن را می توان به برتر بودن خود شبکه نیز وابسته دانست(توضیحات پرسش یک) چرا علاوه بر سایز ورودی که جزئیات بیشتری را به شبکه می دهد، دارای سه کanal رنگی می باشد که تفاوت های رنگی را نیز بین شان تمایز می گذارد. موردم بعدی اینکه دارای فیلتر ها با سایز های مختلف می باشد لذا جزئیات و ویژگی های حاصل در ابعاد مختلف استخراج و در اختیار دسته بند قرار می گیرد.

به جهت سرعت همگرایی نیز این شبکه با اختلاف بهتر عمل کرده است؛ معیار مقایسه ما تعداد تکرار می باشد که این شبکه و با انتقال یادگیری در مجموع با ۲۴ تکرار توانسته به دقت بسیار بالا برسد که دلیل آن را میتوان استفاده از انتقال یادگیری دانست چرا که وزن های اولیه معرفی دار که قبل از آموزش دیده اند با وزن های تصادفی اولیه بسیار متفاوت است و سرعت آموزش شبکه را جلو می اندازد.

به جهت تعمیم پذیری نیز این شبکه عالی عمل کرده است چرا که پیش‌بینی داده های Unseen اش نسبت به شبکه LeNet-5 بهتر و با دقت تر بوده است.(در حین آموزش و برای داده های اعتبار سنجی نیز این مورد صادق است)