

گزارش تمرین شماره 2

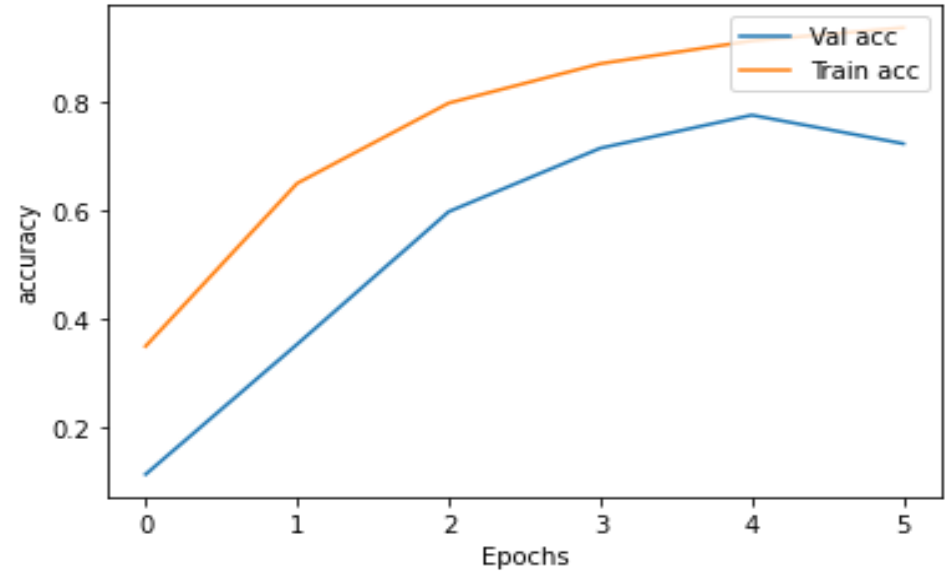
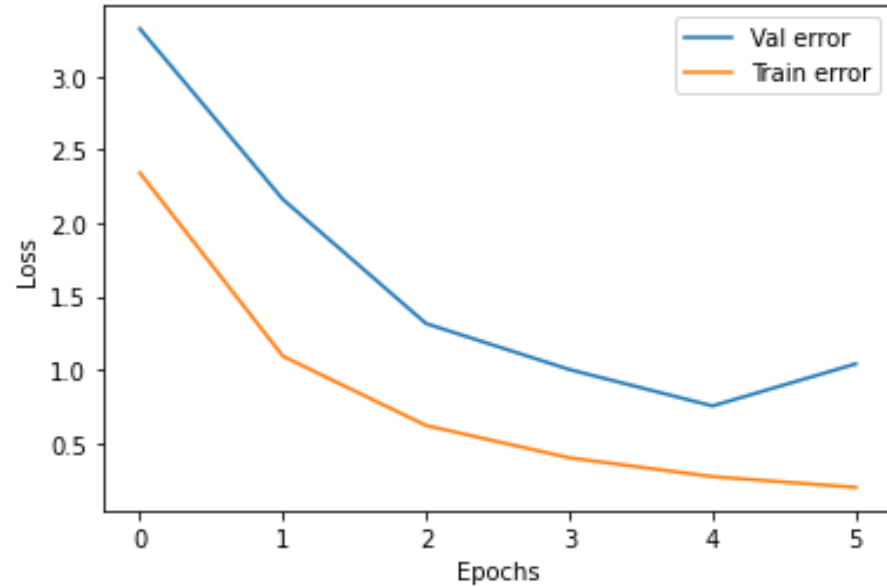
محمد زیاری

« بعضی از اصطلاحات انگلیسی به علت مشکلات
فونت ، فارسی تایپ شده است»

تمرین 1 . تابلو های رانندگی

مدل اولیه

اول از همه نمودار های Accuracy و loss آن را میبینیم.



Weight = 1.hdf

اول از همه من باید بگم که هر کدی رو که برای سوال ها اضافه کردم ، در ابتدا با # مشخص شده است.

سوال اول : متوازن بودن کلاس ها

یک بلاک کد با استفاده از دیتا آگمنشن زدم که هر دفعه کرش میکرد و مجبور به استفاده از دیکشنری شدم. (کد قبلی کامنت شده موجود است.)

اما در این کد طبق خروجی که داشتیم بین بیشترین میزان یک کلاس که 2010 بود تا

کمترین آن که 180 است اختلاف فاحشی وجود دارد که آن را با استفاده از **Over sampling**

کردن داده ها پوشش دادیم. طوری که میزان همه ی کلاس ها به 2010 رسید و آن ها

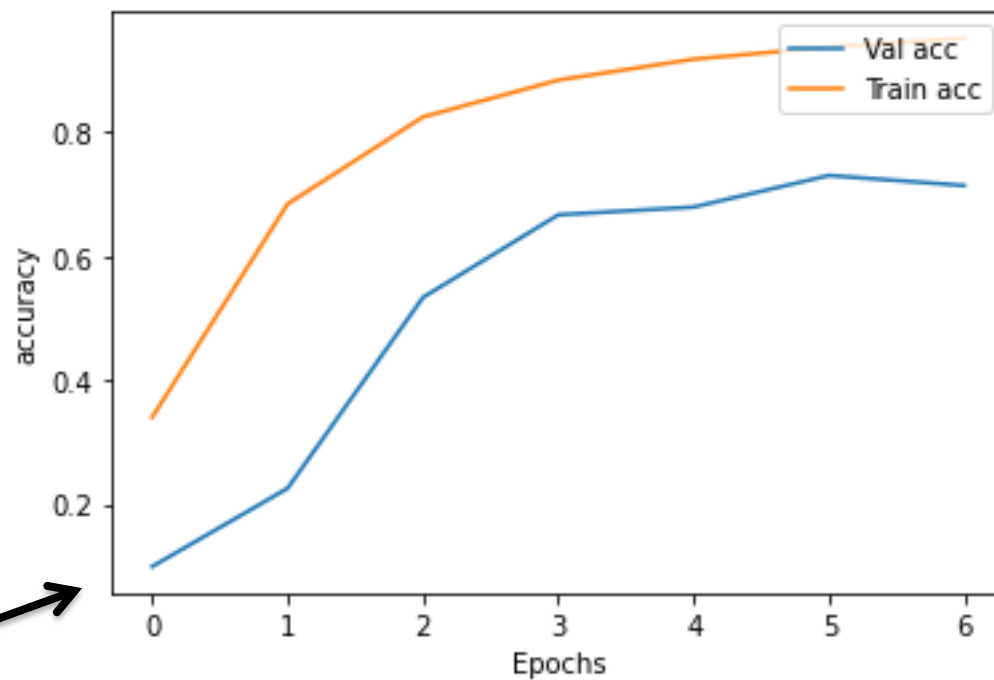
را در آرایه های تمرینمان ریختیم و تست کردیم که دقت آن را در صفحه بعد بررسی

میکنیم.

*یک نکته ی ریزی که هست اینه که دیکشنری که تعریف کردیم ، داده ها رو به ترتیب

کلاس ها تو آرایه میریزه. مدت زیادی صرف حل این مشکل کردم که بعد فهمیدم میشه

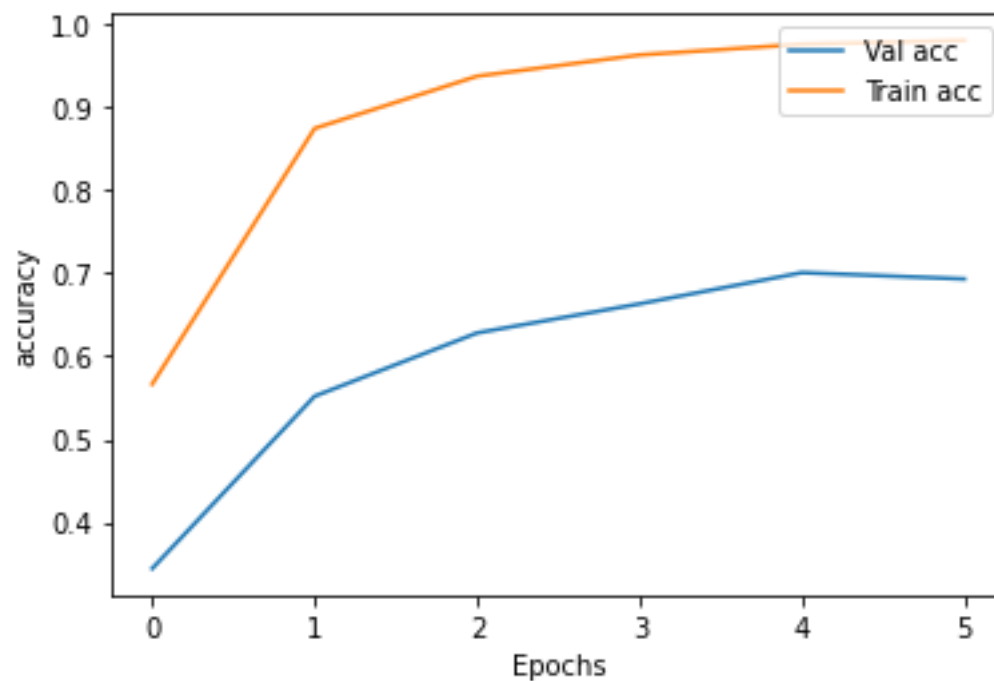
موقع فیت کردن مدل ، مقدار شافل را درست بزاریم تا به همون ترتیب وارد نشوند.



تا میانگین و
بدون تاثیر
روی ولیدیشن

واضح است نسبت به قبلی کارایی بهتری داشته و حتی در ایپاک ششم به دقت
0.96 رسیده و دلیل اینکه به توقف زود هنگام خورده **Validation accuracy**

است که هنوز داده های آن متوازن نیست و آن ها را اصلاح نکردیم. $Weight = 2.hdf$
روی داده های ولیدیشن هم تست میکنیم و نتیجه گیری میکنیم.



اول از همه این دفعه تعداد عکس ها 2010 تا شده با این حال نتوانستیم با این روش ولیدیشن رو بهبود ببخشیم ولی یه کمکی به مدل کردیم که دقتش رو به نزدیک 0.98 برسونه به خاطر اینکه به **Early Stopping** برخورد نکردیم. تعبیر من از این داستان اینه که من برای ولیدیشن از دیتا جنریتور استفاده نکردیم و شاید مدل ما هنوز به اون درجه نرسیده که توانایی تشخیص خودش رو بالا ببره و به جای حفظ کردن ، یادگیری انجام بده که در مدل های بعدی بهت میشود.

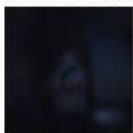
سوال دوم و سوم: تاثیر سیاه سفید کردن و شفافیت

تا حالا ما داده ها رو رنگی تست میکردیم . اینبار از عکس های سیاه و سفید استفاده کردیم تا تاثیرش رو بسنجیم.

اما راجب شفافیت یک بلاک از کد از هر کلاس یک عکس نمایش داده شده که مشخص است شرایط نوری برابری ندارند. برای اصلاح این روند هم از **equalization**

استفاده میکنیم که در کد علامت زده شده و در صفحه بعد خطا و دقت آن را بررسی میکنیم.

Dangerous curve to the right



Double curve



Road narrows on the right



Road work

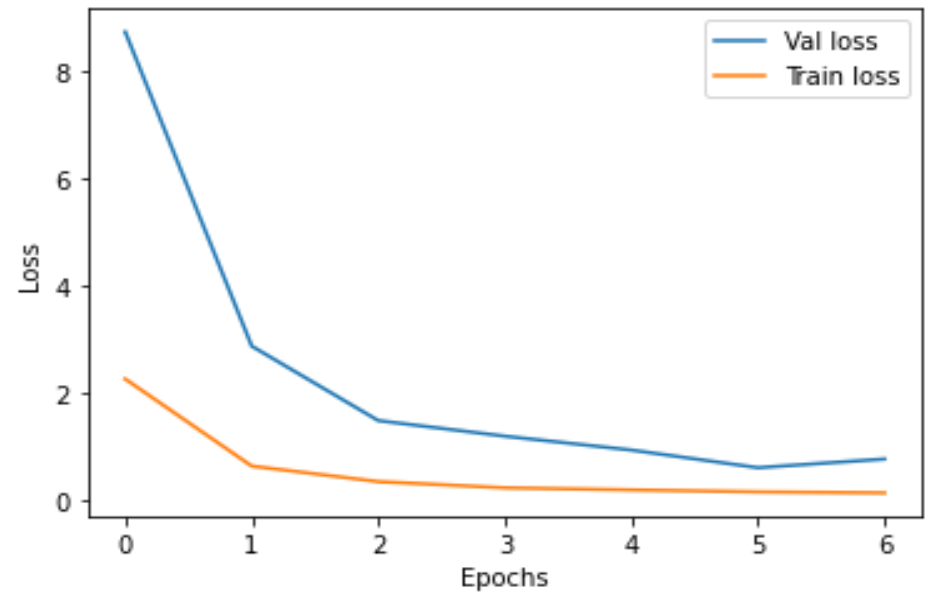
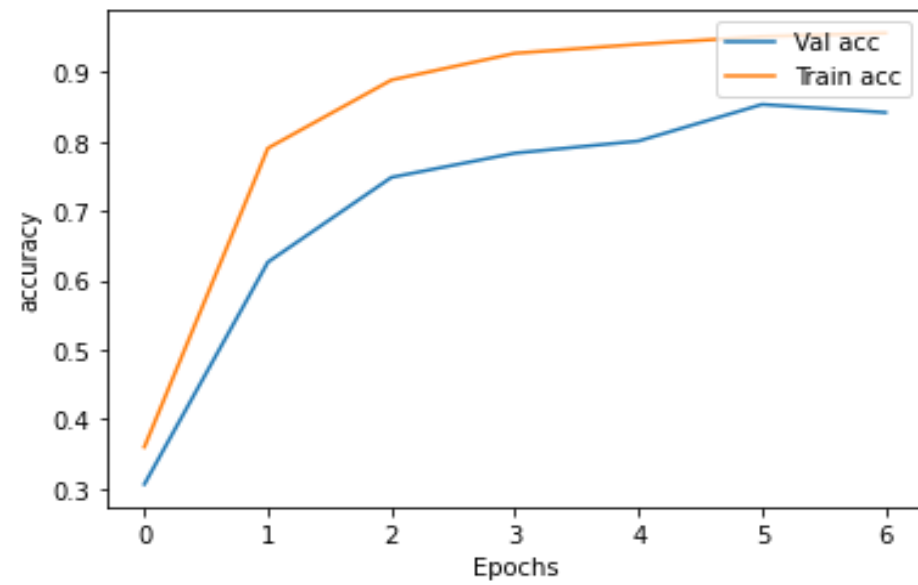


Children crossing



Bicycles crossing





تأثیر بسیار بسزایی در افزایش دقت (به خصوص برای ولیدیشن) داشت و به نظر هر دو مورد کمک زیادی به مدل ما کرده اند.

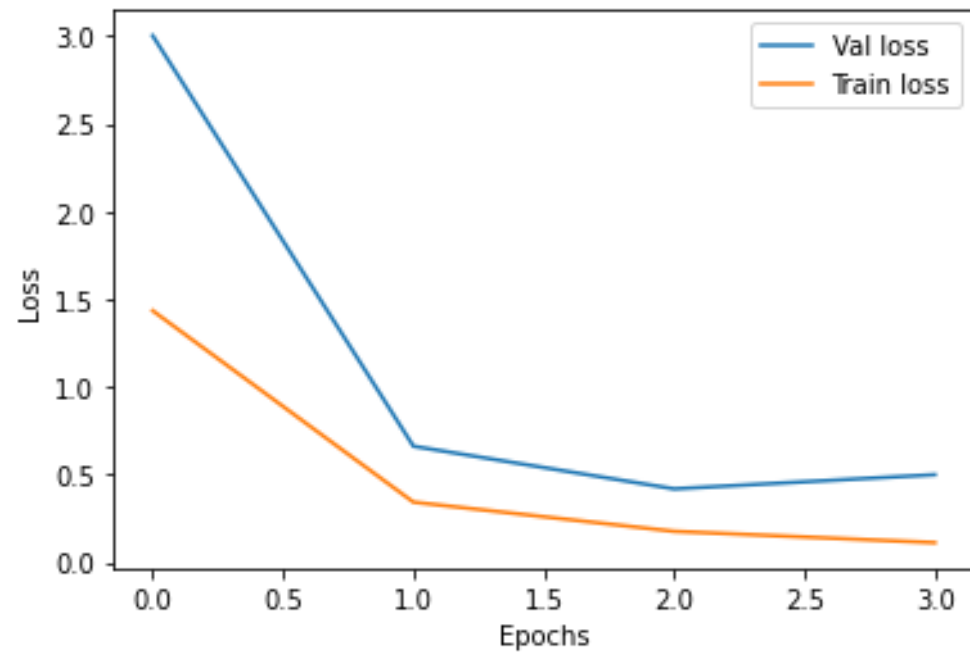
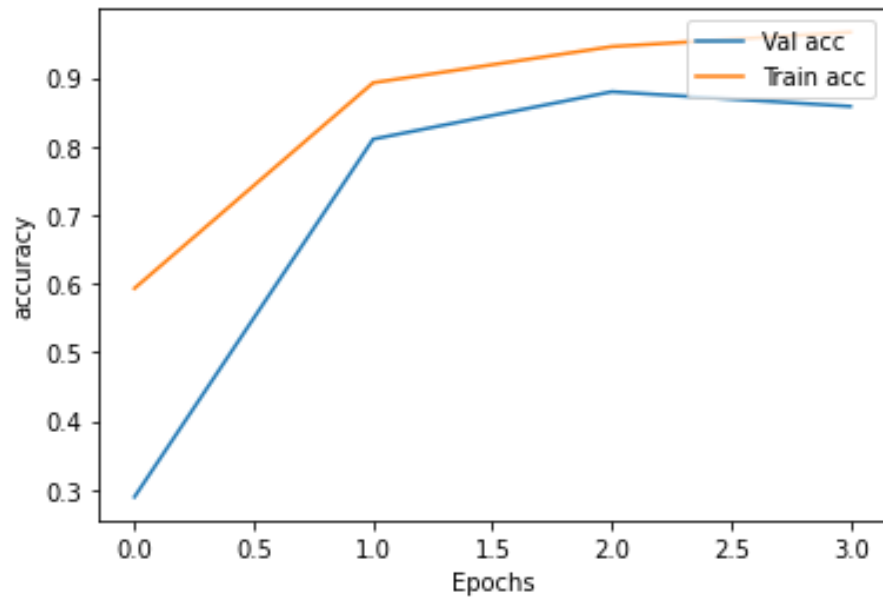
همانطور که گفتم برای شفافیت تنها یک خط در همان روند سیاه سفید کردن اعمال شده.

```
eq = cv2.equalizeHist(np.uint8(pic))
```

اما راجب هزینه ، خوبی آن است که کانالش 1 است و در واقع کار خواندن عکس را راحت تر میکند.

برای مدل من هر ایپاک در حالت قبل 60 ثانیه و بعد سیاه سفید کردن 30 ثانیه طول کشید ، حالا مدل من خیلی پیچیده و طولانی نیست اما حتما در شبکه های بزرگ تر هزینه کمتر آن در فرآیند آموزش مدل به ما کمک خواهد کرد.

یک مدل خوب با همین روش ها

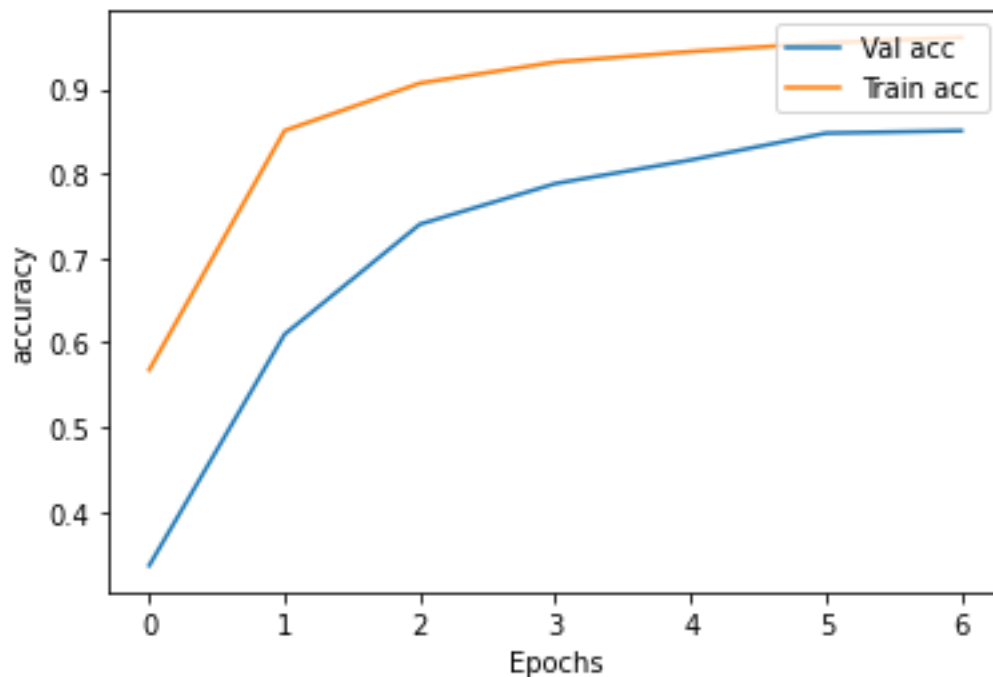


eq

سوال چهارم : افزایش تعداد داده ها

خب ما با استفاده از همان سوال 2 تعداد عکس ها را افزایش دادیم که همه رو به یک اندازه برسانیم ، هم برای ولیدیشن هم برای داده های تمرین.

اما در این قسمت میخواهیم با استفاده از دیتا آگمنتیشن به صورت تصادفی مقداری عکس (همراه با دیتا جنریتور یعنی زوم و فلیپ و ...) هم به داده های تمرین و هم به ولیدیشن اضافه کنیم که کد آن به #4 مشخص شده است و نمودار ها و دقت را در صفحه بعدی مشاهده میکنیم.



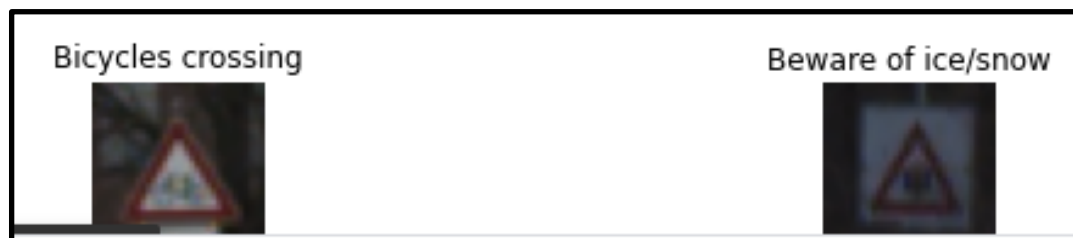
همانطور که گفتم تغییر زیادی رو نسبت به قبلی شاهد نبودیم چون خودمون در قسمت اول داده ها رو بیشتر کرده بودیم ، ولی از همین مدل برای انجام کارهای قسمت پنجم و انجام تست استفاده میکنیم.

سوال پنجم : تست و اشتباهاتش

داده ها رو تست کردیم و جواب نهایی رو تست هامون 90.33% بود.

بیشترین داده هایی که با هم اشتباه گرفته میشوند هم نمایش دادیم

درایه های روی قطر اصلی درستی جواب ها را نمایش میدهند و هرچه غیر آن اشتباه بودن مدلمان در حدس تابلو است.



این دو هم اشتباه گرفته میشوند (45 بار) که با توجه به شکلی که دارند قابل توجیه است.

اما بیش از همه تابلوی **Dangerous curve to the left** اشتباه گرفته شده ، به طور کلی من این نتیجه گیری رو داشتم این تابلو رو شبکه خوب تشخیص نداده ، به طوری که با 2 تابلو که مثلث هستند اشتباه گرفته شده است ، شاید شبکه مان بیش از حد بر روی فرم دور تابلو ها مانور میدهد.

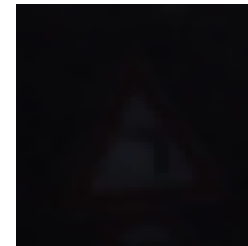
در مورد تابلوی دوچرخه هم این مورد صادق است که ، 2 بار در بیشترین تعداد اشتباهات از 3 بار قرار گرفته است.

Bicycles crossing	0	45	683	3	16	0	0
Bumpy road	0	1	0	429	0	9	0
Children crossing	1	18	0	2	616	2	0
Dangerous curve to the left	1	26	135	89	6	346	0

Bicycles crossing



Dangerous curve to the left



سوال ششم : نشان دادن لایه ها و فیلتر ها

در آخر این کار انجام شده است آن هم روی 42 لایه Cnn که داشتیم

ولی در آخر عکس ها انقدر ریز هستند که قابل تشخیص نبودند .

البته تحلیل کردن این عکس ها هم کار راحتی نیست . تابلوی ابتدایی که داشتیم

سرعت مجاز 20 بود . در ابتدا به نظر هر پیکسلی در عکس عوض میشود

ولی بیش از همه دور عکس. (دلیلش هم مشخصا خود لایه conv است که

متمرکز کار میکند)

ولی پس از اینکه مدتی گذشت دیگه همه چیز به پیکسل هایی تقسیم شد که قابل

پیگیری نبود و مشخص نیست شبکه مان بر کدام قسمت عکس بیشتر توجه

دارد.

عکسی که داشتیم در هر لایه و فیلتر یک سری تغییرات رویش اعمال شده که همانطور که گفتم تحلیلشان کار دشواری به نظر میرسد .

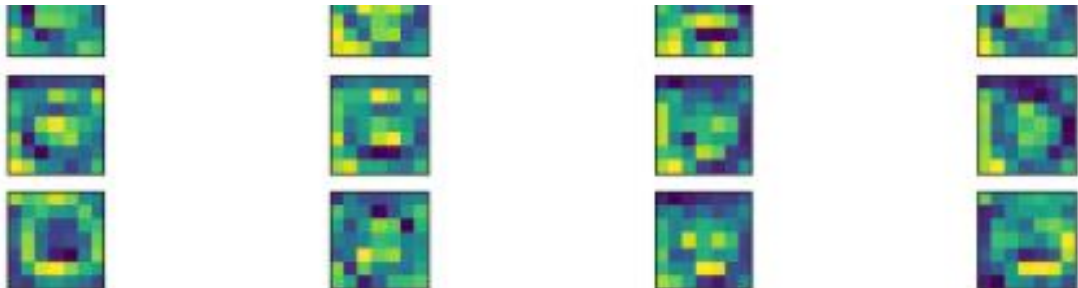
در آخر هم یک سری عکس سیاه به نمایش در می آیند که واقعا مشخص نیست شبکه

به چه شکل آنرا نگاه میکند.

یکبار دیگر اینکار را روی همان داده انجام دادم و فقط در 15 لایه اول نمایش دادم.

عکس ها را هم سیاه سفید نکردم ، باز هم به صورت پیکسلی در آمده در آخر کار که

باهم میبینیم.



تمرین 2 . قیمت خانه ها

تمرین دوم چون فقط یک خواسته از ما داشته یک مقدار توضیحات کمتری نسبت به اولی دارد.

اول از همه باید بگم داده های csv را به چه شکل در نظر گرفتم.

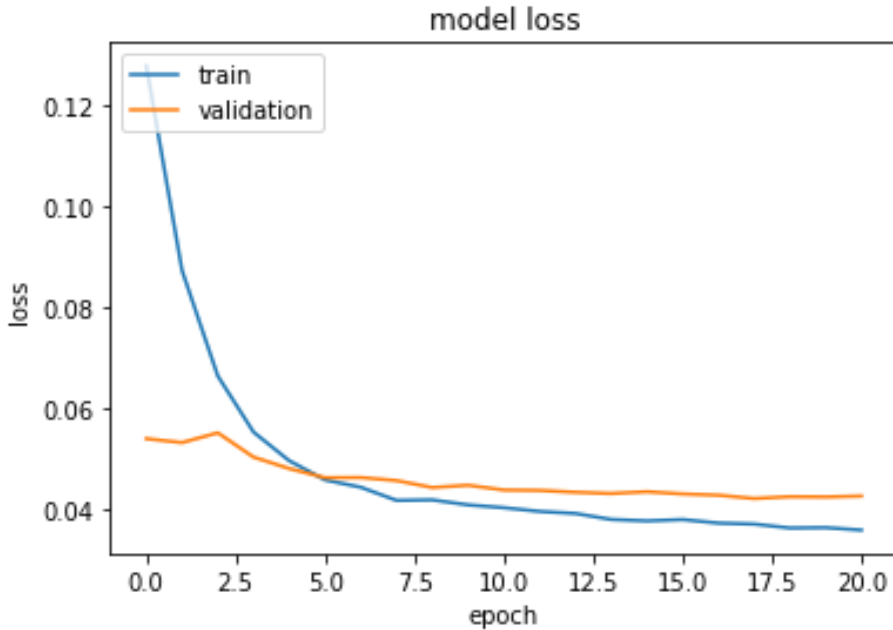
برای تعداد اتاق و مساحت و برجسب ستون شهر همان داده ها را اسکیل کردم. برای شهر اول خواستم از One hot coding استفاده کنم ولی برجسب ستون شهر ، تفاوت شهر ها را متمایز میکرد و دیگه از اون استفاده نکردم و کامنتش کردم.

برای خیابان هم اگر نمودار رو ببینید فراونی بیشترین خیابان کمتر از 5 تا بوده و مشخصا این ویژگی نمیتواند مدل خوب و دقیقی بسازد (حتی وقتی پلاک را از آن جدا کردم باز هم مقدار خوبی نبود و به طور کل تصمیم گرفتم این ویژگی رو دراپ کنم) برای حمام هم ، حمام نصف را به عنوان داده ای جدید به ستون هایم اضافه کردم. عکس ها را هم بدون تغییری خواندم. برویم سراغ مدل ها.

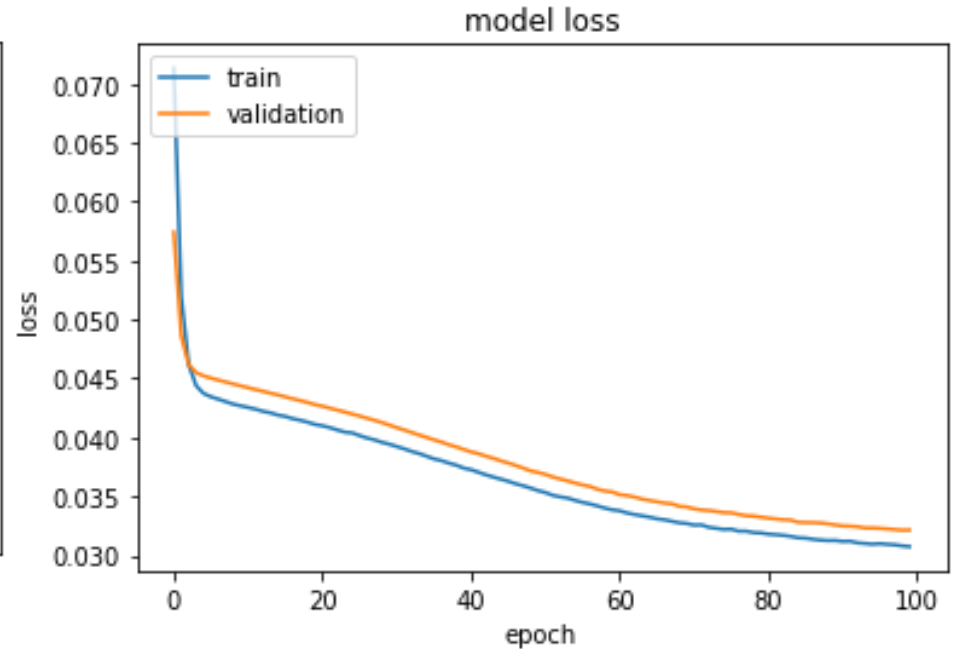
عکس ها را هم جدا از فایل جدولمان خواندیم و در یک Cnn آن را تست خواهیم کرد ،
همانطور که داده های CSV را در NN تست میکنیم و سپس دو مدل را به هم میچسبانیم و
میبینیم که چه دقت و خطایی دارد.

یکبار هم این کار را بدون استفاده از عکس ها انجام میدیم و نتیجه آن را گزارش میدهیم.

CSV and Images



Just CSV



میبینیم که روند لرن کردن و کاهش خطا را در 2 حالت گفته شده میبینیم ، در صفحه بعد میانگین خطای بدست آمده از داده های تست را در 2 حالت بررسی میکنیم.

کمترین میزان قیمت خانه ، 195000 و بیشترین 2000000 است که این مقادیر برای میانگین خطا شاید خیلی خوب نباشد.

در هر دو حالت تقریباً میانگین خطای برابری داشتیم و نمیتوانیم از برتری یکی بر دیگری صحبت کنیم ، ولی به طور کل وقتی عکس ها رو هم دادیم ، شبکه بهتری داشتیم و فکر کنم اگر شبکه عمیقتری داشتیم یا یک سری کار روی عکس ها انجام میدادیم با استفاده از دیتاجنریتور ، مدل بهتری هم میتوانستیم داشته باشیم ، این که هر بار لایه را بیشتر کردم جواب بهتری داد نشان دهنده ی همین است. (با 2 مدل مختلف ران شده که در کد هست)

```
print (mid*2000000)
```

```
[273299.64095612]
```
