

گزارش تمرین دوم یادگیری عمیق

زهره ختن لو ۹۸۲۴۳۰۹۲

بخش اول: سوالات

۱. مزایای استفاده از CNN در تسک های پردازش تصویر چیست؟

CNN ها در کاهش تعداد پارامترها بدون از دست دادن کیفیت مدل بسیار موثر هستند. تصاویر دارای ابعاد زیادی هستند (زیرا هر پیکسل به عنوان یک ویژگی در نظر گرفته می شود) که با توانایی های CNN در کاهش ابعاد با تعداد زیادی از پارامترهای یک تصویر مطابقت دارد. این کاهش ابعاد تصویر در لایه های کانولوشن که سعی در استخراج ویژگی های تصویر با استفاده از فیلترها دارد، محقق می شود. همچنین، CNN ها برای شناسایی لبه های اجسام در هر تصویر آموزش داده می شوند.

۲. آنچه در فرآیند Normalization Batch رخ می دهد را به صورت مختصر توضیح دهید و

مزایای استفاده از آن در شبکه های عصبی عمیق را نیز بیان نمایید.
لایه BN ابتدا میانگین μ و واریانس σ^2 مقادیر فعال سازی batch ها را با استفاده از عبارات زیر محاسبه می کند.

$$(1) \mu = \frac{1}{n} \sum_i Z^{(i)} \quad (2) \sigma^2 = \frac{1}{n} \sum_i (Z^{(i)} - \mu)^2$$

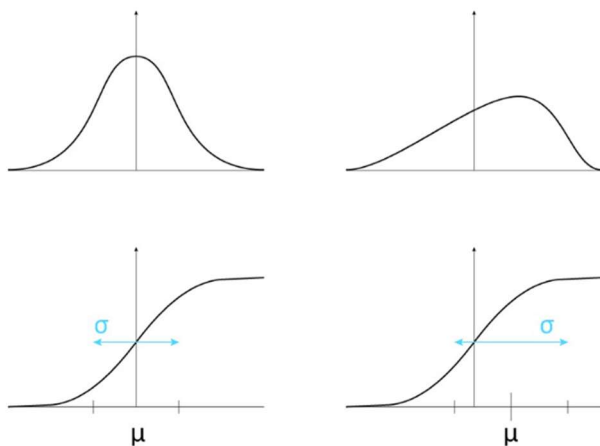
سپس بردار فعال سازی $Z^{(i)}$ را به صورت زیر normalize می کند. به این ترتیب، خروجی هر نورون از توزیع نرمال استاندارد در سراسر batch پیروی می کند. (ϵ ثابتی است که برای اثبات عددی استفاده می شود)

$$(3) Z_{norm}^{(i)} = \frac{Z^{(i)} - \mu}{\sqrt{\sigma^2 + \epsilon}}$$

در نهایت خروجی لایه $\hat{Z}(i)$ را با اعمال یک تبدیل خطی و به کمک دو پارامتر قابل آموزش γ و β به شکل فرمول ۴ محاسبه می‌کند. چنین مرحله‌ای به مدل اجازه می‌دهد تا با تنظیم آن دو پارامتر، توزیع بهینه را برای هر لایه پنهان انتخاب کند:

○ اجازه می‌دهد تا انحراف معیار را تنظیم کنید.

○ امکان تنظیم سوگیری، تغییر منحنی در سمت راست یا چپ را فراهم می‌کند.



در هر iteration، شبکه میانگین μ و انحراف معیار σ مربوط به دسته فعلی را محاسبه می‌کند. سپس با استفاده از میانگین متحرک نمایی (EMA) γ و β را از طریق شیب نزول آموزش می‌دهد.

مزایای استفاده از batch normalization:

این نرمال‌سازی به دلیل اینکه ورودی‌ها را در یک لایه برای هر دسته استاندارد می‌کند باعث تثبیت فرآیند یادگیری و کاهش چشمگیر تعداد دوره‌های آموزشی مورد نیاز برای آموزش شبکه‌های عمیق می‌شود، و در نتیجه باعث افزایش سرعت یادگیری شبکه می‌شود.

۳. توضیح دهید چرا Learning Residual از محو شدگی گرادیان (Gradient Vanishing) جلوگیری میکند.

ResNets به شبکه‌های عصبی اطلاق می‌شود که skip connection ها بخشی از معماری شبکه هستند. این اتصالات به اطلاعات گرادیان اجازه می‌دهد از لایه‌ها عبور کند، با ایجاد این اتصال، اطلاعات خروجی لایه قبلی به خروجی لایه عمیق‌تر اضافه می‌شود. این اجازه می‌دهد تا اطلاعات قسمت‌های قبلی شبکه به بخش‌های عمیق‌تر شبکه منتقل شود و به حفظ انتشار سیگنال حتی در شبکه‌های عمیق‌تر کمک کند.

توجه داشته باشید که ResNet ها مجموعه ای از شبکه های نسبتاً کم عمق هستند و مشکل محو شدگی گرادیان را با حفظ جریان گرادیان در کل عمق شبکه حل نمی کنند - بلکه آنها به سادگی با ساخت مجموعه هایی از بسیاری از شبکه های کوتاه با هم از مشکل جلوگیری می کنند.

بخش دوم: پیاده سازی

در این بخش بعد از پیاده سازی شبکه با بعد ثابت ۳ برای کرنل و مقادیر مختلف برای number of filter تست کردیم و نتایج در نوت بوک ها به شکل زیر حاصل شده است.

NF = 16

```
Epoch [30/30], Loss: 1.6004, Time (s): 457  
Epoch [30/30], Val Loss: 1.5906, Val Acc: 43.2%, Time(s): 459
```

NF = 32

```
Epoch [30/30], Loss: 1.5026, Time (s): 644  
Epoch [30/30], Val Loss: 1.5095, Val Acc: 45.2%, Time(s): 646
```

NF = 48

```
Epoch [30/30], Loss: 1.4513, Time (s): 836  
Epoch [30/30], Val Loss: 1.5207, Val Acc: 44.0%, Time(s): 838
```

NF = 64

```
Epoch [30/30], Loss: 1.4169, Time (s): 1068  
Epoch [30/30], Val Loss: 1.5019, Val Acc: 45.4%, Time(s): 1071
```

NF = 80

```
Epoch [30/30], Loss: 1.3794, Time (s): 1284  
Epoch [30/30], Val Loss: 1.5415, Val Acc: 44.3%, Time(s): 1288
```

NF = 96

```
Epoch [30/30], Loss: 1.3514, Time (s): 1661  
Epoch [30/30], Val Loss: 1.4845, Val Acc: 46.0%, Time(s): 1665
```

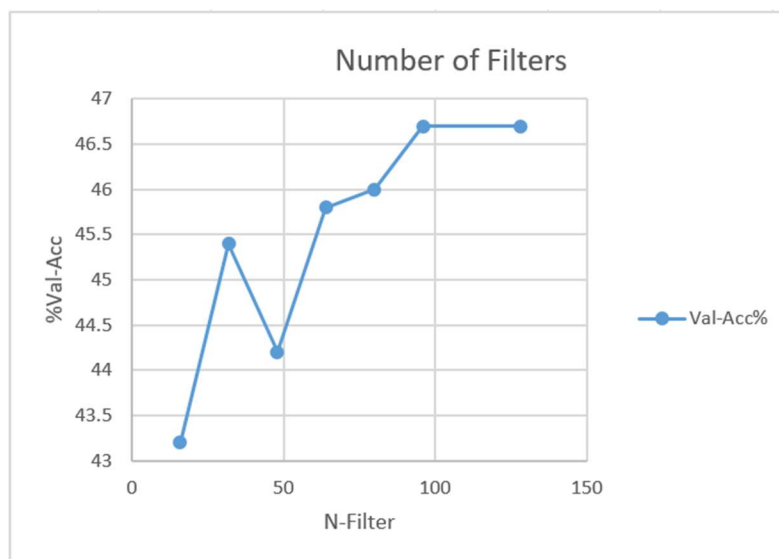
NF = 128

Epoch [30/30], Loss: 1.3359, Time (s): 2373

Epoch [30/30], Val Loss: 1.5095, Val Acc: 45.3%, Time(s): 2379

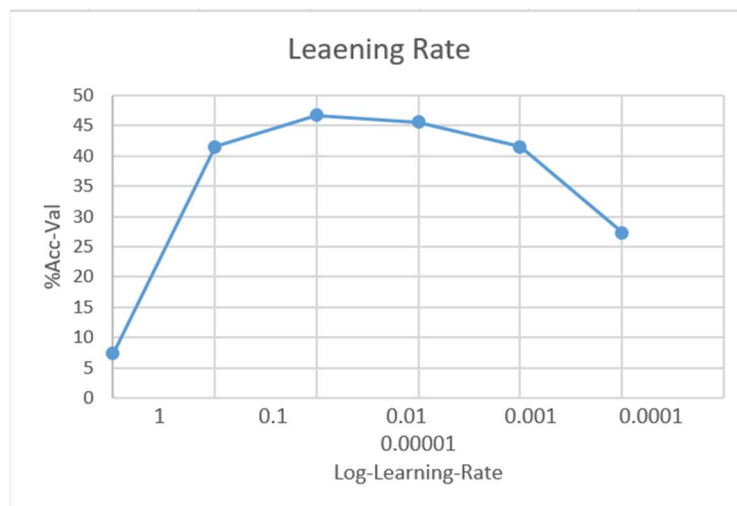
با توجه به نتایج نموداری به شرح زیر بدست می آید که همانطور که مشخص است با کرنل ثابت، با افزایش NF مقدار acc نیز افزایش می یابد تا زمانی که به مقدار ثابتی می رسد. بنابراین مقدار ۹۶ برای NF مناسب به نظر می رسد.

| N-Filter | Val-Acc% |
|----------|----------|
| 16 | 43.2 |
| 32 | 45.4 |
| 48 | 44.2 |
| 64 | 45.8 |
| 80 | 46 |
| 96 | 46.7 |
| 128 | 46.7 |



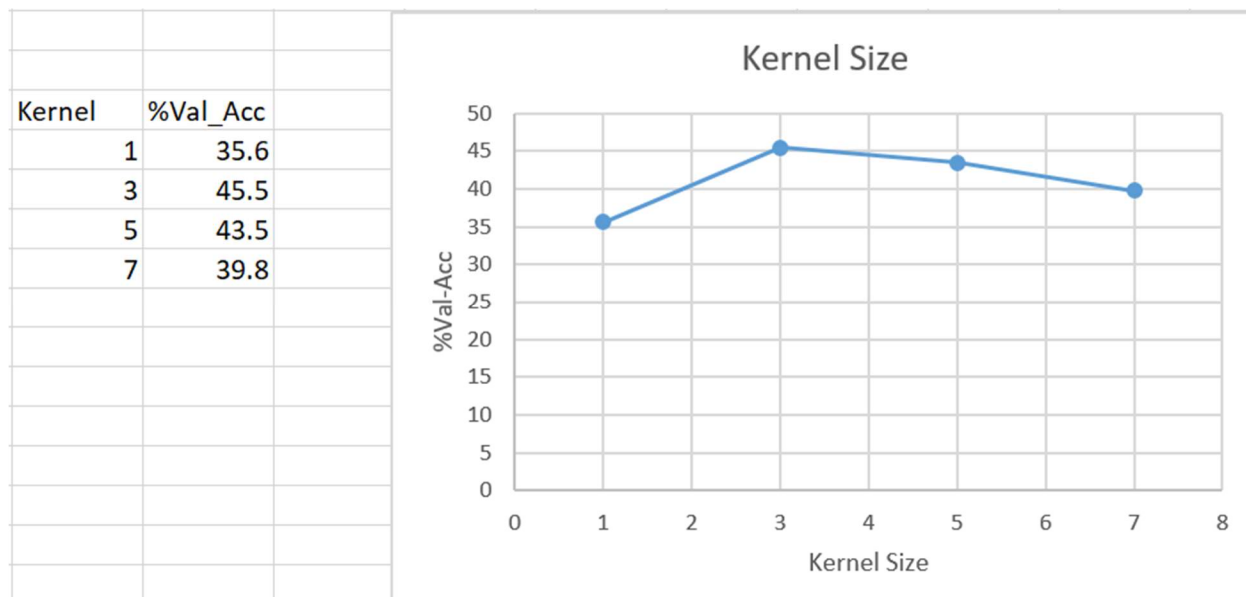
به طور مشابه برای مقادیر مختلف learning rate هم خروجی می گیریم که نتایج به در نوت بوک های LR پیوست شده و به شرح نمودار زیر است.

| LR | Learn-Rate | %Val-Acc |
|----|------------|----------|
| 0 | 1 | 7.4 |
| 1 | 0.1 | 41.5 |
| 2 | 0.01 | 46.7 |
| 3 | 0.001 | 45.5 |
| 4 | 0.0001 | 41.5 |
| 5 | 0.00001 | 27.3 |

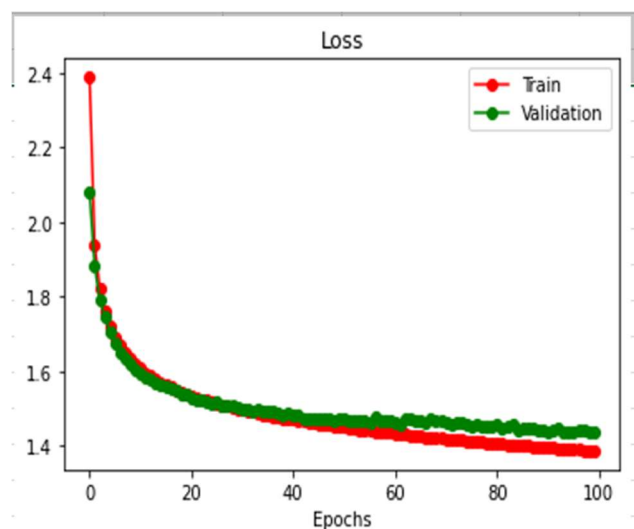


همانطور که در تصاویر مشخص است با کاهش نرخ یادگیری تا مقادیر ۰,۰۱ خروجی نیز بهتر شده و مقدار acc افزایش یافته است اما بعد از این مقدار نتایج نزولی شده در نتیجه یک مقدار اپتیممی را باید برای نرخ یادگیری در نظر گرفت.

همچنین به طور مشابه مقادیر مختلف را نیز برای کرنل تست می کنیم و بازنمایی آن به شکل زیر است.

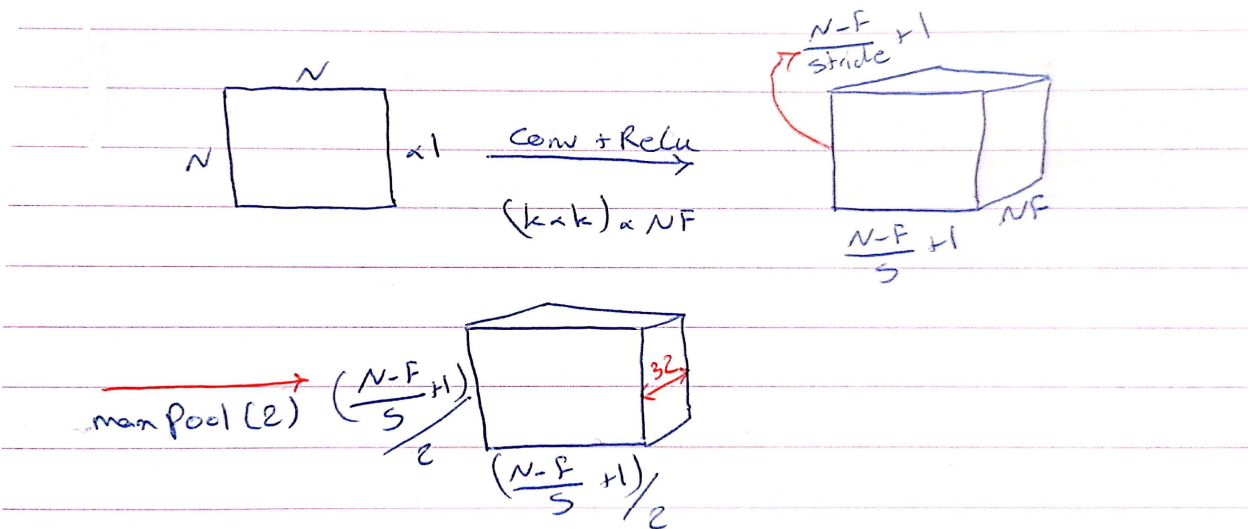


همانطور که مشخص است مقدار ۳ برای کرنل حالت بهینه را دارد.



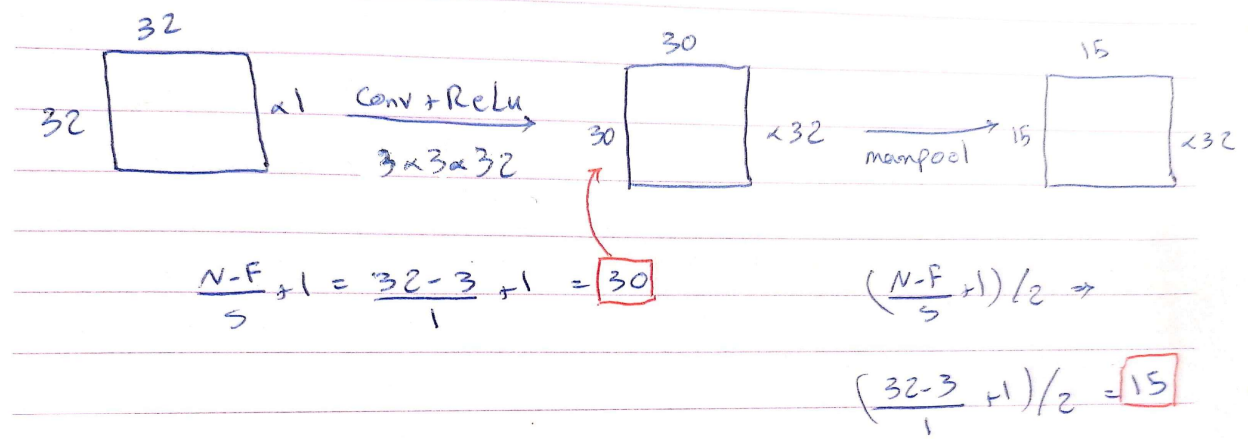
همچنین طبق نمودار زیر مشخص است که با افزایش تعداد ایپاک ها مقدار تابع loss کاهش یافته است.

برای محاسبه feature map به روش زیر عمل می شود.



Scanned with CamScanner

بنابراین با توجه به مقادیر لایه مشخص شده feature map را در آن لایه به شکل زیر محاسبه می کنیم.

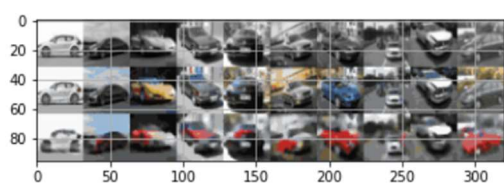


Scanned with CamScanner

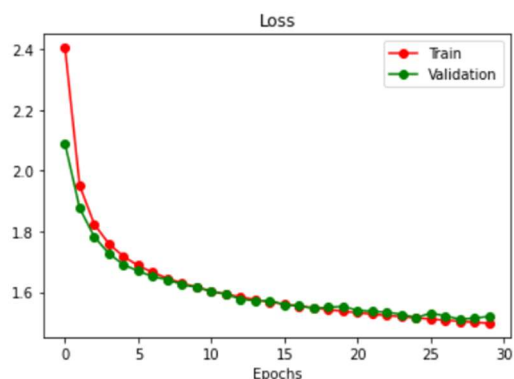
نتایج بدست آمده بعد از آموزش شبکه base model:

همانطور که در خروجی های موجود در نوت بوک مشخص است نتایج این مدل در ۳۰ اپیاک به شکل زیر بدست آمده است.

```
-----
CNN
Beginning training ...
Epoch [1/30], Loss: 2.4039, Time (s): 19
Epoch [1/30], Val Loss: 2.0882, Val Acc: 30.5%, Time(s): 21
Epoch [2/30], Loss: 1.9507, Time (s): 40
Epoch [2/30], Val Loss: 1.8760, Val Acc: 35.4%, Time(s): 42
Epoch [3/30], Loss: 1.8231, Time (s): 61
Epoch [3/30], Val Loss: 1.7819, Val Acc: 38.2%, Time(s): 63
Epoch [4/30], Loss: 1.7588, Time (s): 82
Epoch [4/30], Val Loss: 1.7267, Val Acc: 39.8%, Time(s): 84
Epoch [5/30], Loss: 1.7170, Time (s): 103
Epoch [5/30], Val Loss: 1.6899, Val Acc: 40.6%, Time(s): 105
Epoch [6/30], Loss: 1.6873, Time (s): 124
Epoch [6/30], Val Loss: 1.6696, Val Acc: 41.1%, Time(s): 126
Epoch [7/30], Loss: 1.6643, Time (s): 145
Epoch [7/30], Val Loss: 1.6505, Val Acc: 41.6%, Time(s): 147
Epoch [8/30], Loss: 1.6453, Time (s): 166
Epoch [8/30], Val Loss: 1.6412, Val Acc: 41.6%, Time(s): 167
Epoch [9/30], Loss: 1.6295, Time (s): 186
Epoch [9/30], Val Loss: 1.6246, Val Acc: 42.1%, Time(s): 188
Epoch [10/30], Loss: 1.6159, Time (s): 207
Epoch [10/30], Val Loss: 1.6161, Val Acc: 42.1%, Time(s): 209
Epoch [11/30], Loss: 1.6035, Time (s): 228
Epoch [11/30], Val Loss: 1.6017, Val Acc: 42.6%, Time(s): 229
Epoch [12/30], Loss: 1.5930, Time (s): 249
Epoch [12/30], Val Loss: 1.5926, Val Acc: 42.8%, Time(s): 250
Epoch [13/30], Loss: 1.5836, Time (s): 271
Epoch [13/30], Val Loss: 1.5770, Val Acc: 43.5%, Time(s): 272
Epoch [14/30], Loss: 1.5749, Time (s): 292
Epoch [14/30], Val Loss: 1.5708, Val Acc: 43.6%, Time(s): 293
Epoch [15/30], Loss: 1.5672, Time (s): 313
Epoch [15/30], Val Loss: 1.5721, Val Acc: 43.4%, Time(s): 314
Epoch [16/30], Loss: 1.5601, Time (s): 334
Epoch [16/30], Val Loss: 1.5581, Val Acc: 43.9%, Time(s): 335
Epoch [17/30], Loss: 1.5537, Time (s): 355
Epoch [17/30], Val Loss: 1.5549, Val Acc: 43.9%, Time(s): 357
Epoch [18/30], Loss: 1.5478, Time (s): 376
Epoch [18/30], Val Loss: 1.5459, Val Acc: 44.2%, Time(s): 378
Epoch [19/30], Loss: 1.5421, Time (s): 397
Epoch [19/30], Val Loss: 1.5496, Val Acc: 44.0%, Time(s): 399
Epoch [20/30], Loss: 1.5369, Time (s): 418
Epoch [20/30], Val Loss: 1.5521, Val Acc: 43.8%, Time(s): 420
Epoch [21/30], Loss: 1.5319, Time (s): 439
Epoch [21/30], Val Loss: 1.5403, Val Acc: 44.2%, Time(s): 441
Epoch [22/30], Loss: 1.5274, Time (s): 461
Epoch [22/30], Val Loss: 1.5361, Val Acc: 44.4%, Time(s): 462
Epoch [23/30], Loss: 1.5229, Time (s): 482
Epoch [23/30], Val Loss: 1.5337, Val Acc: 44.4%, Time(s): 484
Epoch [24/30], Loss: 1.5187, Time (s): 503
Epoch [24/30], Val Loss: 1.5264, Val Acc: 44.7%, Time(s): 505
Epoch [25/30], Loss: 1.5147, Time (s): 525
Epoch [25/30], Val Loss: 1.5159, Val Acc: 45.1%, Time(s): 527
Epoch [26/30], Loss: 1.5109, Time (s): 546
Epoch [26/30], Val Loss: 1.5301, Val Acc: 44.4%, Time(s): 548
Epoch [27/30], Loss: 1.5072, Time (s): 568
Epoch [27/30], Val Loss: 1.5215, Val Acc: 44.7%, Time(s): 570
Epoch [28/30], Loss: 1.5037, Time (s): 590
Epoch [28/30], Val Loss: 1.5100, Val Acc: 45.3%, Time(s): 592
Epoch [29/30], Loss: 1.5004, Time (s): 612
Epoch [29/30], Val Loss: 1.5140, Val Acc: 45.0%, Time(s): 614
Epoch [30/30], Loss: 1.4972, Time (s): 633
Epoch [30/30], Val Loss: 1.5197, Val Acc: 44.7%, Time(s): 635
```



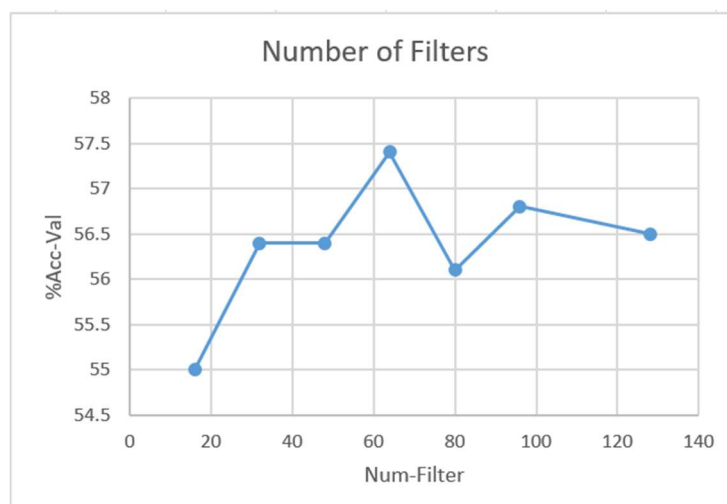
که خروجی آن در تصویر زیر و روند تغییر تابع loss در نمودار زیر بازنمایی شده است.



پیاده سازی شبکه U-net

در این حالت مانند شبکه قبل نتایج را برای مقادیر مختلف learning rate, number of filter و kernel بدست آورده ایم که به شرح نمودار های زیر است:

| N-Filter | Val-Acc% |
|----------|----------|
| 16 | 55 |
| 32 | 56.4 |
| 48 | 56.4 |
| 64 | 57.4 |
| 80 | 56.1 |
| 96 | 56.8 |
| 128 | 56.5 |

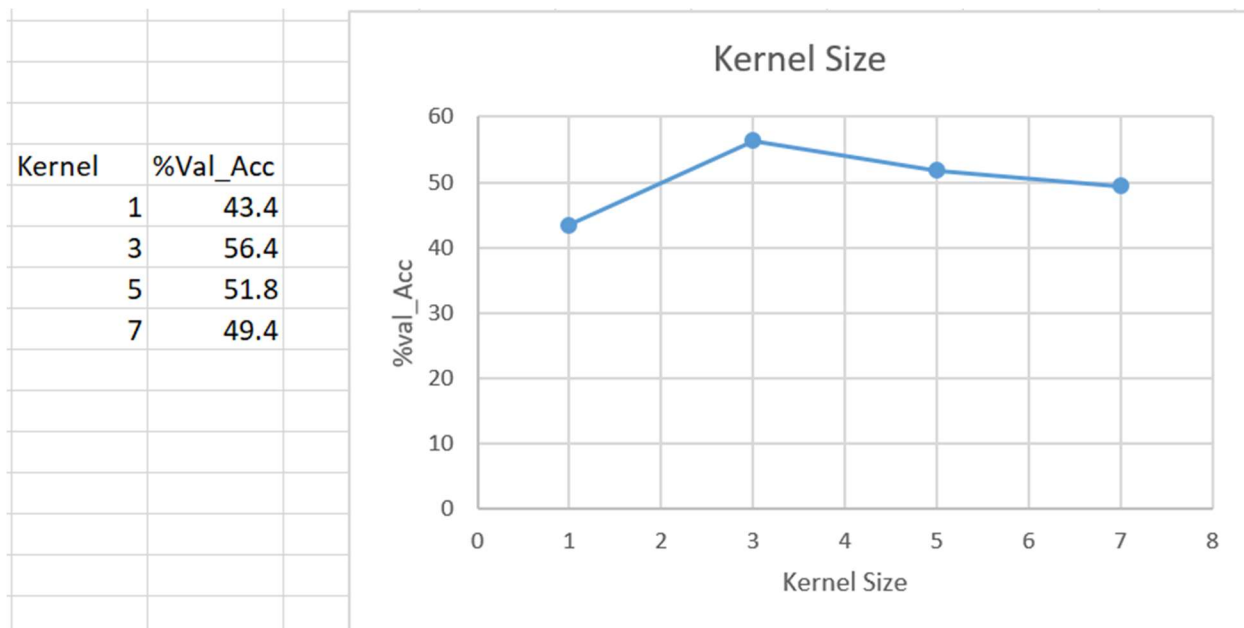


بنا بر این مقدار ۶۴ برای number of filter در این شبکه مناسب به نظر می رسد.

| LR | Learn-Rate | %Val-Acc |
|----|------------|----------|
| 0 | 10 | 39.8 |
| 1 | 1 | 61.2 |
| 2 | 0.1 | 54 |
| 3 | 0.01 | 57.6 |
| 4 | 0.001 | 56.4 |
| 5 | 0.0001 | 45.7 |



مقدار ۱ برای نرخ یادگیری خروجی بهتری داشته.



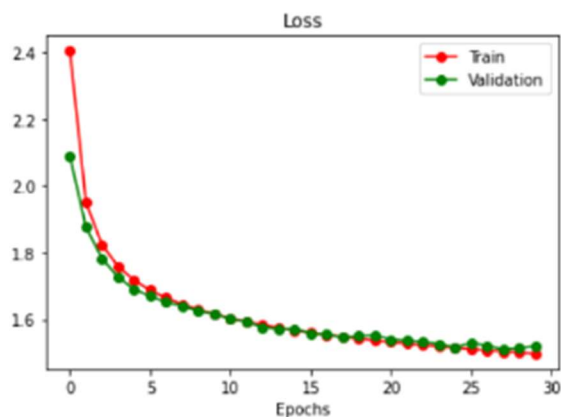
در این شبکه نیز مقدار ۳ برای کرنل بهینه ترین حالت بوده است.

برای مقایسه نتایج این شبکه با شبکه قبل داریم:

basemodel Epoch [30/30], Loss: 1.4972, Time (s): 633
Epoch [30/30], Val Loss: 1.5197, Val Acc: 44.7%, Time(s): 635

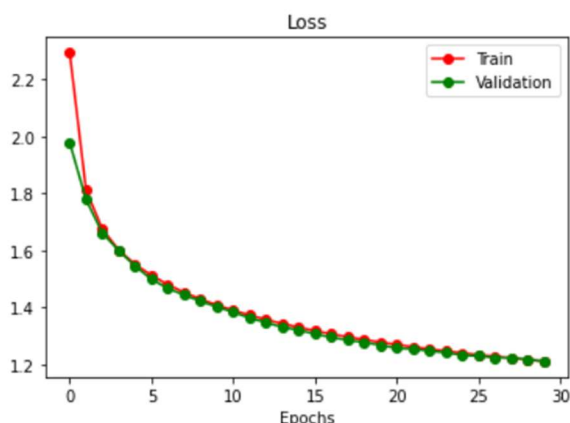
U-net Epoch [30/30], Loss: 1.2096, Time (s): 692
Epoch [30/30], Val Loss: 1.2101, Val Acc: 56.4%, Time(s): 694

همانطور که مشخص است در ایپاک های آخر یادگیری این دو شبکه درصد accuracy به میزان قابل توجهی افزایش یافته که نشان می دهد skip connection های بین لایه nام و لایه اول، لایه (1)-nام و لایه دوم، و... اثرگذاری قابل توجهی روی عملکرد شبکه داشته است همچنین این اثرگذاری را می توان در توابع loss دو شبکه مشاهده کرد.



شبکه اولیه base model

شبکه U-net



پیاده سازی بخش residual block

در این بخش در هر لایه از یک مسیری استفاده شده است که علاوه بر اینکه هر لایه ورودی خود را پس انجام یک سری عملیات از لایه قبل می گیرد بایدد ورودی های لایه قبل نیز بدون آن تغییرات به ورودی لایه بعد اضافه شود که با افزایش پارامترها باعث می شود که خروجی مناسب تری در این شبکه حاصل شود.

U-net

Epoch [30/30], Loss: 1.2096, Time (s): 692

Epoch [30/30], Val Loss: 1.2101, Val Acc: 56.4%, Time(s): 694

residual
block

Epoch [25/25], Loss: 0.9064, Time (s): 4349

Epoch [25/25], Val Loss: 0.8474, Val Acc: 70.5%, Time(s): 4359

همانطور که مشخص چنین معماری باعث افزایش قابل توجه مقدار accuracy شده است.

همچنین مقدار تابع loss نیز به نسبت شبکه های قبل در ایپاک های آخر کاهش داشته است.

