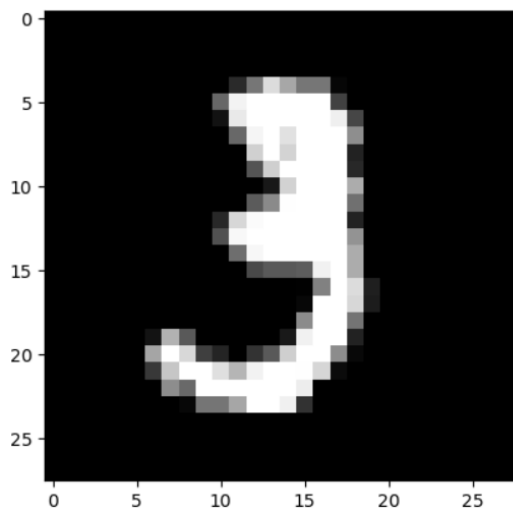


گزارش کار

قدم اول:

در قدم اول فایل های MNIST در پروژه قرار داده شد و کدهای پردازش و خواندن تست شد.



```
[[0.]  
 [0.]  
 [0.]  
 [1.]  
 [0.]  
 [0.]  
 [0.]  
 [0.]  
 [0.]  
 [0.]
```

Process finished with exit code 0

قدم دوم:

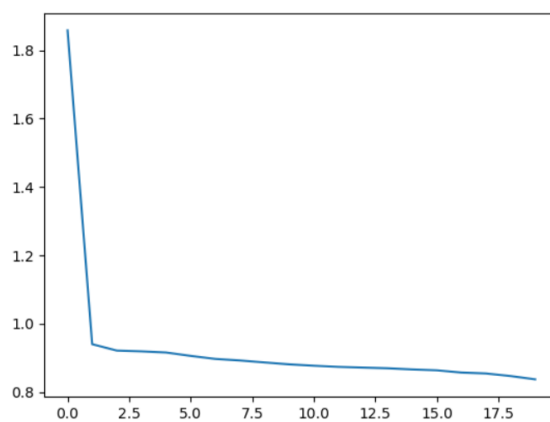
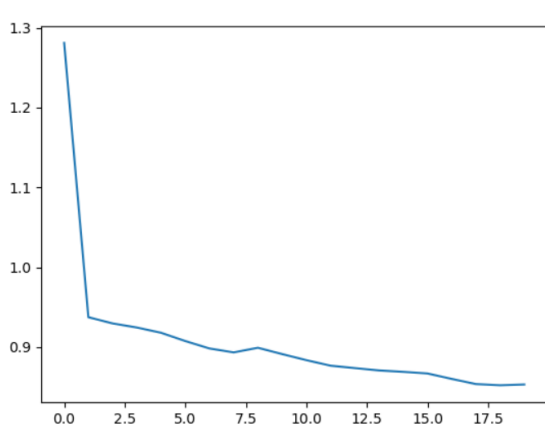
در این مرحله Feedforward انجام شد و خروجی هایی رندوم مانند عکس زیر از شبکه گرفته شد. عدد آخر (16%) دقت مدل در این مرحله است.

```
1  
7  
1  
7  
7  
1  
9  
7  
7  
7  
7  
1  
7  
1  
0.16
```

قدم سوم:

در این مرحله , SGD به مدل اضافه شده ودقت مدل با یادگیری و تست روی 100 عکس اول به محدوده ی 28 الی 41 درصد رسید.

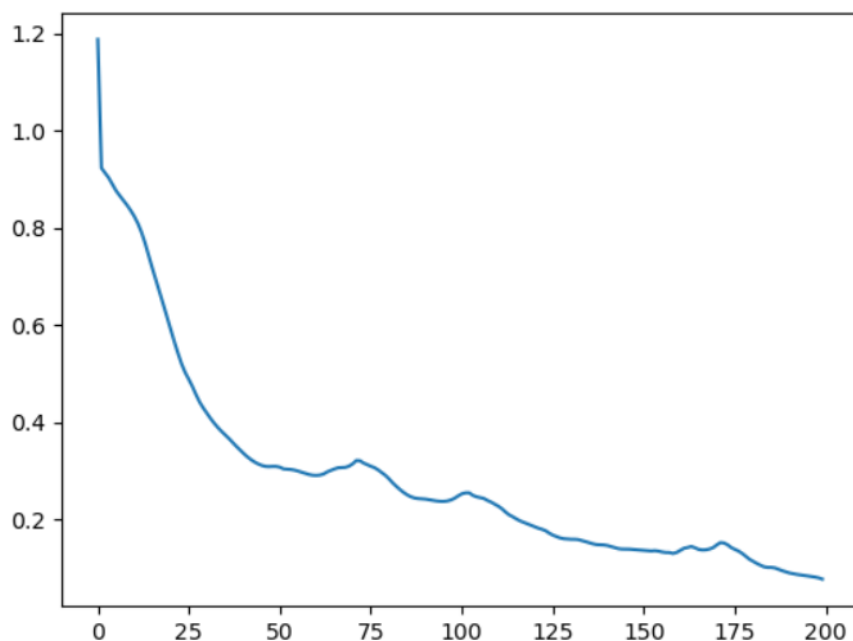
همانطور که در عکس های زیر مشاهده می شود cost ها در طول روند یادگیری سیر نزولی داشتند. دقت در شبکه ی سمت راست 35 و در چپی 28 درصد بود.



یادگیری در شبکه ی مربوط به عکس سمت راست 178 ثانیه طول کشید.

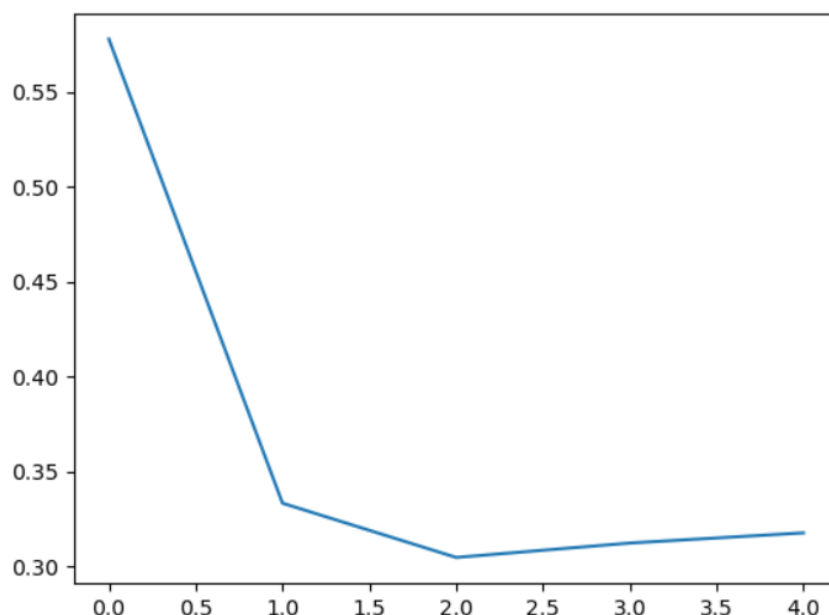
قدم چهارم:

در این قدم مراحل مربوط به یادگیری vectorize شد. سپس یادگیری و تست با epoch 200 روی 100 عکس اول انجام شد و نتیجه ی آن دقت 98 درصدی در مدت [یادگیری] 45 ثانیه بود. عکس زیر روند کاست ها در epoch 200 را نشان می دهد.



قدم پنجم:

و در آخر در این مرحله شبکه روی کل عکس های مجموعه ی training آموزش داده و روی همان تست شد. زمان یادگیری در حدود 3 دقیقه (170 تا 220 ثانیه) و دقت حدود 90 (82 تا 92) درصد به دست آمد. عکس زیر پلات کاست را برای مجموعه train نشان می دهد که دقت این شبکه 84 درصد اندازه گیری شد.



همچنین بعد از آموزش روی مجموعه آموزشی , دقت روی مجموعه ی تست در محدوده ی 80 تا 90 بود.

امتیازی ها

```
shift_mode = True
```

1. **شیفت به راست:** برای شیفت تصاویر تگ shift_mode را فعال می کنیم. ابتدا با استفاده از تابع shift_image زمان ورودی دادن به به شبکه ی عصبی هر عکس را 4 پیکسل به سمت راست شیفت داده و سپس وارد شبکه می کنیم. در نهایت می بینیم که دقت شبکه ی عصبی به حدود 15 تا 30 درصد می رسد. دلیل آن می تواند این باشد که پایه ی تشخیص اعداد توسط انسان نحوه ی قرارگیری خانه های سفید نسبت به سیاه است در حالی که شبکه ی عصبی از طریق قرارگیری هر خانه در صفحه مختصات عدد را می فهمد و با جابجایی خانه های سفید نوروں های اشتباهی فعال می شوند.

2. **Tanh:** برای تغییر activation function از sigmoid به Tanh باید در ابتدای کد sigmoid را False

```
sigmoid = False
```

کرد. با استفاده از tanh به جای sig دقت مدل افزایش محسوسی پیدا نمی کند (حدود 80 درصد) اما سرعت آموزش بیشتر می شود (به حدود 160 ثانیه می رسد)

3. **Momentum_SGD**: برای استفاده از momentum ابتدا لازم است تگ آن TRUE شود.

```
momentum_SGD = True  
momentum = 0.9
```

momentum را 0.9 در نظر میگیریم. ایده ی این آپگرید این است که هر بار هنگام آپدیت ضریبی از تغییرات دفعه ی قبل را به تغییرات این مرحله اضافه کنیم. به این ترتیب متغیر هایی که خیلی از مقدار بهینه ی خود فاصله دارند سریعتر به آن مقدار می رسند. با این شیوه دقت مدل افزایش پیدا کرد (حدود 90 درصد) و همچنین همانطور که از عکس زیر پیداست روند کاهشی هزینه ها بسیار محسوس تر است.

