

گزارش پیاده سازی

تمرین سوم

توسط: کتایون کبرائی

در این تمرین باید داده های اینتل را با استفاده از مدل های معروف رزنت و ترنسفر کردن آن
پیش بینی کنیم.

اولین کار بعد از دانلود دیتا و آپلود آن نیاز است که یک سری تغییراتی روی دیتا انجام دهیم. برای مثال باید روی آن با استفاده از توابع ماثول ترنسفرم تغییراتی انجام دهیم تا دیتاهای بیشتری داشته باشیم. چون میدانیم خاصیت مدل این است که هرچه دیتای بیشتری داشته باشیم به دقت پیش بینی بالاتری میرسیم. از جمله کارهای دیگری که دیتای ما را آماده میکند نرمالایز کردن دیتا است که بعد از ترنسفر کردن روی دیتای تست و ترین انجام میدهیم.

```
data_transforms = {  
    train: transforms.Compose([  
        transforms.CenterCrop(224),  
        transforms.RandomHorizontalFlip(0.5),  
        transforms.RandomRotation(degrees=(0,180)),  
        transforms.ToTensor(),  
        transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])  
    ]),  
    test: transforms.Compose([  
        transforms.RandomResizedCrop(224),  
        transforms.CenterCrop(224),  
        transforms.ToTensor(),  
        transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])  
    ]),  
}
```

نکته قابل توجه در این بخش ترنسفر این است که با توجه ورودی ها تابعی انتخاب کنیم که در تعداد ورودی ها دست کاری انجام ندهد چون در غیر این صورت در اجرا به مشکل بر خواهیم خورد.

در گام بعدی مدل مورد نظر خود را دائلود میکنیم. همان طور که در صورت تمرین هست با استفاده از ماژول مدل میتوانیم مدل های مختلف مثل Resnet , Densenet, inception-resnet و غیره را دائلود و پیاده سازی کنیم.

برای استفاده از این مدل ها و پیاده سازی ترنسفر لرنینگ نیاز داریم ابتدا برای اپیاک های ابتدایی وزن های مدل را فریز کنیم تا دیتای جدید وزن های مدل اصلی را به هم نزنند. تنها تغییری که نیاز هست در لایه های مدل هم بدهیم تعویض لایه های فولی کاکنتند نهایی میباشد.

برای طراحی دوباره بخش فولی کاکنتند از حداقل دو لایه لینیر استفاده میکنیم که لایه های میانی تر را با اکتیویشن فانکشن رلو و لایه ی اخر را با لاگ سافت مکس یا سافت مکس قرار میدهیم تا بهترین نتیجه را به ما دهد.

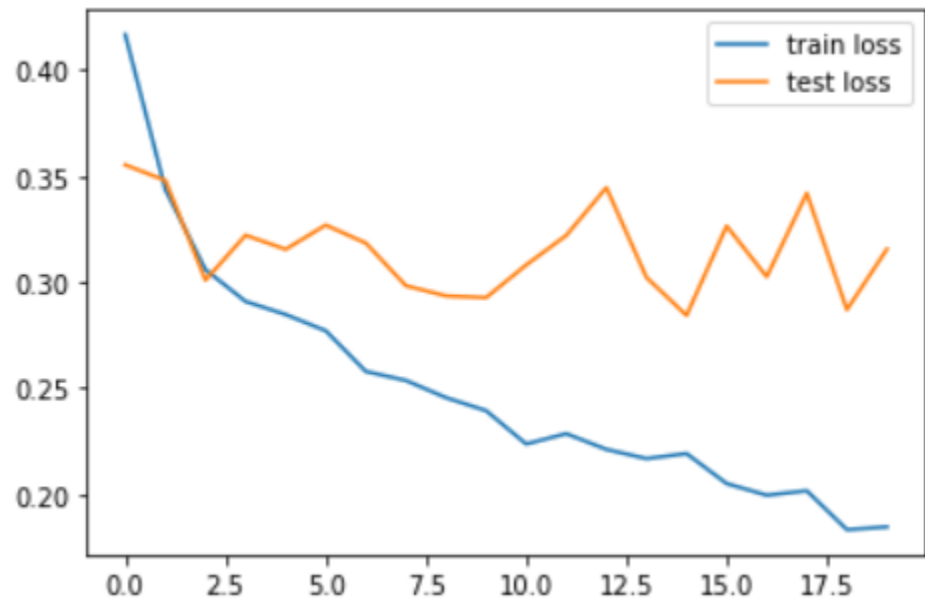
```
for param in resnet.parameters():  
    param.requires_grad = False  
  
resnet.fc = nn.Sequential(nn.Linear(2048, 1024),  
                        nn.ReLU(),  
                        nn.Dropout(0.25),  
                        nn.Linear(1024, 256),  
                        nn.ReLU(),  
                        nn.Dropout(0.25),  
                        nn.Linear(256, 6),  
                        nn.LogSoftmax(dim=1))
```

در نهایت بعد از تغییر لایه های نهایی مدل را به جی پی یوی در دسترس می‌رسانیم. برای تمام مدل ها از کرایتیرین کراس انتروپی و برای تابع اپتیمایزر از ادام استفاده می‌کنیم. در یکی از ترین ها ابتدا به مدل لرنینگ ریت 0.1 دادیم. چون بالا بود این عدد دیدیم لاس های تست و ترین یک بار خیلی بالا و یک بار خیلی کم میشوند که در نهایت مارا به بهترین دقت نمی‌رساند. پس به لرنینگ های 0.001 و 0.0001 می‌رسیم.

بعد از تمامی این ها مدل هارا ترین می‌کنیم که با نمودارهایی تک تک انها را توصیف می‌کنیم:

ابتدا بهترین مدل ما مدل رزنت 101 بود که به بالاترین اکیوریسی یعنی 89.93 رسید. طرز کار این مدل به این شکل بود که ابتدا مدل اصلی را برای 10 الی 15 اپپاک مثلا فریز می‌کردیم که دیتای جدید وزن های بدست آمده را خراب نکند. بعد از ده اپپاک که تمام شد می‌گذاشتیم دوباره کل مدل از فریز در اید و همه با هم شروع به ترین شدن کنند. این کار در نهایت بیشترین دقت را به ما داد.

با استفاده از نگهداری مقادیر بدست آمده لاس در هر اپپاک نمودار ان را رسم می‌کنیم.



در نهایت با استفاده از تابع های نوشته شده برای به دست آوردن اکيوريسى، اکيوريسى ديتاى
ترين و ديتاى تست و اکيوريسى نهايى را بدست مى اوريم.

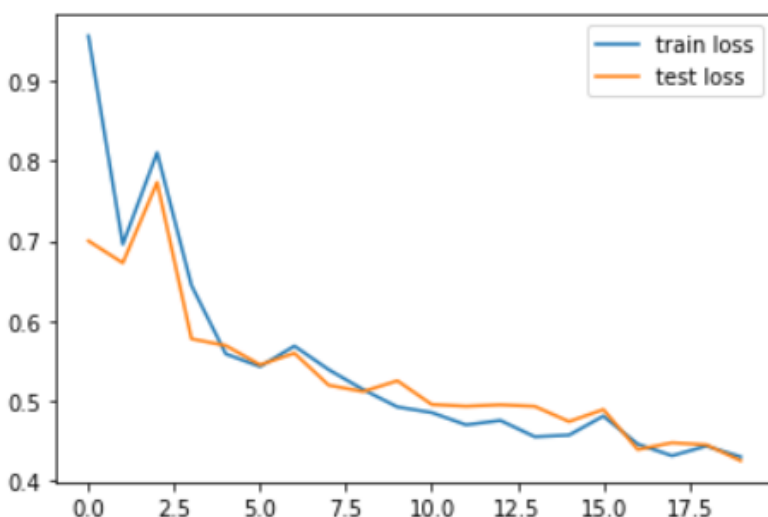
```
accuracy(dataloader[test_data], model)
```

```
0%|          | 0/94 [00:00<?, ?it/s]  
Got 2689/3000 with accuracy 89.6333
```

یکی از دیگر کارهایی که میتوانیم بکنیم گذاشتن اسکچولر میباشد که ترین مارا بهتر میکند. البته در مدل ما به کار کردن بیشتری احتیاج داشت که اسکچولر در متد ترین کردن جواب دهد ولی میبینیم که شروع لاس ایپاک ها مقادیر خیلی بهتری میباشد.

```
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=1e-4)
exp_lr_scheduler = lr_scheduler.StepLR(optimizer, step_size=7, gamma=0.1)
```

یکی دیگر از مدل ها مدل دنس نت بود که اکيوريسى خوبى داد و قابل در نظر گرفتن بود:



در این مدل لاس دیتای ترین و تست بسیار نزدیک به هم و این کیفیت مدل را به ما نشان میدهد. در نهایت هم با اندازه گیری این لاس ها در هر ایپاک به اکيوريسى 85.73 رسیدیم.

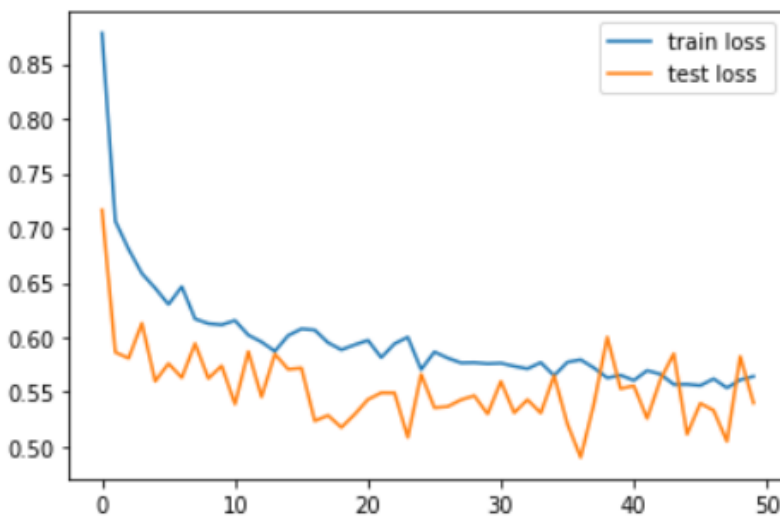
```
In [53]: accuracy(test_loader, model)
0%|          | 0/47 [00:00<?, ?it/s]
Got 2656/3000 with accuracy 88.5333
```

```
In [ ]:
```

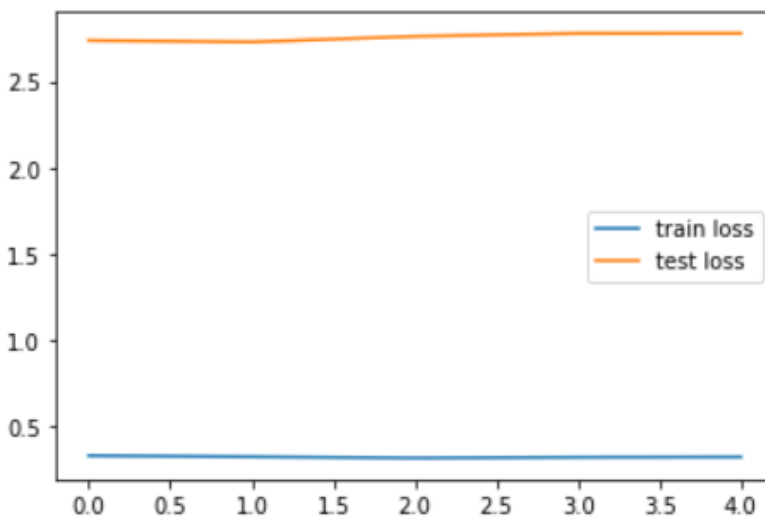
یک مدل دیگری که پیاده سازی شد مدل اینسپشن رزنت بود که برخلاف تصور از دنس نت و رزنت 101 اکیوریسی خیلی پایین تری دریافت کردیم:

```
In [83]: accuracy(data_loader[test], inception_resnet)
0%|          | 0/94 [00:00<?, ?it/s]
Got 2505/3000 with accuracy 83.5000
```

```
In [ ]:
```



در بسیاری از مدل ها مشکل اورفیتینگ هم داشتیم به طوری که اختلاف بین لاس دیتای ترین و تست خیلی زیاد میشد که این مشکل با دیدن نمودار ان کاملا قابل تشخیص بود:



```
Epoch 1/5, Train_Loss: 0.3340, \Test_Loss: 2.7370, Duration: 0:01:45.264610
Epoch 2/5, Train_Loss: 0.3278, \Test_Loss: 2.7281, Duration: 0:01:44.532685
Epoch 3/5, Train_Loss: 0.3204, \Test_Loss: 2.7605, Duration: 0:01:42.578312
Epoch 4/5, Train_Loss: 0.3247, \Test_Loss: 2.7776, Duration: 0:01:41.351926
Epoch 5/5, Train_Loss: 0.3264, \Test_Loss: 2.7779, Duration: 0:01:41.551378
```

در نتیجه با تغییر گام ها و عوض کردن اعداد و لود درست دیتای تست و ترین تمامی این مشکلات برطرف شد.