

بخش 1:

(الف)

برای این قسمت نیاز است که فقط pretrained مدل resnet50 را برابر False قرار دهیم تا مقدار وزن های خود مدل لحاظ نشود و مدل از اول با مقدار های تصادفی آموزش داده شود و در هر 3 epoch وزن ها را سیو می کنیم (منبع: [pytorch](#))

```
res_mod = models.resnet50(pretrained=False)

# saving weights
if epoch % 3 == 0:
    checkpoint = {'model_state_dict': model.state_dict(), 'optimizer_state_dict': optimizer.state_dict()}
    save_checkpoint(checkpoint, f'/content/checkpoint_{epoch}.pt')
```

در نهایت بعد از epoch10 به دقت تقریبی 4.20 رسیدیم

```
Epoch 9/9
-----
Iterating through data...
=> saving checkpoint
train Loss: 146.3324 Acc: 4.7549
Iterating through data...
=> saving checkpoint
val Loss: 150.5951 Acc: 4.1915

Training complete in 34m 15s
Best val Acc: 4.191469
```

(ب)

در این قسمت مقدار pretrained را برابر True می گذاریم تا از مقدار های pretrain شده استفاده شود و مقدار انها را Freeze می کنیم تا مقدار انها نیز عوض نشود سپس یک لایه fully connected برای دسته بندی اضافه می کنیم که مقدار وزن های این لایه آموزش داده میشود.

```

resnet_pretrained = models.resnet50(pretrained=True)
#freeze part model and train the rest
for param in resnet_pretrained.parameters():
    param.requires_grad = False

resnet_pretrained.fc = nn.Linear(resnet_pretrained.fc.in_features, len(classes))

```

در این جا به 40 درصد دقت در داده های تست و 80 درصد دقت در داده های آموزشی رسیدیم که نشان دهنده ان است که overfit شدیم اما دقت نیز به دلیل استفاده از مدل pretrain شده افزایش داشته است

```

Epoch 9/9
-----
Iterating through data...
=> saving checkpoint
train Loss: 33.9759 Acc: 80.9927
Iterating through data...
=> saving checkpoint
val Loss: 79.5736 Acc: 40.1538

Training complete in 30m 46s
Best val Acc: 40.240578

```

منابع: [medium](#)، [stackoverflow](#)

(پ)

در این حالت ابتدا با مدل pretrain شده فیچر ها را استخراج کردیم و به مدل svm دادیم به دقت آموزش 99 درصد و 11 درصد تست رسیدیم که پر واضح است مدل overfit شده است.

(ت)

در این مورد ، لایه fully connected ، layer2 و در layer4 هم Bottleneck دوم را از حالت freeze خارج کردیم تا روی مدل آموزش داده شود به دقت 99 درصد در داده های آموزش و 72 درصد در داده های تست رسیدیم

```
[18]
for name, child in resnet_pretrained.named_children():
    if name in ['layer2', 'fc']:
        print(f"{name} - unfreezed")
        unfreeze(child)
    elif name in ['layer4'] :
        unfreeze(child[0], False)
        unfreeze(child[1], False)
        unfreeze(child[2])
    else:
        unfreeze(child, False)

pytorch_total_params = sum(p.numel() for p in resnet_pretrained.parameters())
pytorch_total_trainable_params = sum(p.numel() for p in resnet_pretrained.parameters())
print('pytorch_total_params= ', pytorch_total_params )
print('pytorch_total_trainable_params= ', pytorch_total_trainable_params)
```

در این [لینک](#) هم یک finetuner معرفی شده که می توان مدل را با آن finetune کرد

