

## <중간과제 - CIFAR10 인식 정확도 챌린지>

휴먼지능정보공학과 201910803 박채희

**최종정확도: 76.02999877929688**

(Batch\_size=16, Learning rate=0.02, num\_epoch=50, data normalization, dropout 0.2(linear 0.4), weight initialization, SGD, scheduler step size=20)

내 코드에서의 순위

순위	정확도	특징
1	76.02999877929688	Weight initialization, drop out, data normalization
2	75.38999938964844	Weight initialization, drop out, data normalization
3	75.18000030517578	Weight initialization, drop out, batch normalization, data normalization
4	75.0199966430664	Weight initialization, drop out, data normalization, learning rate decay
5	74.65999603271484	drop out, data normalization
6	74.50999450683594	Weight initialization, drop out, data normalization, learning rate decay

실험 과정

1. drop out 비율을 바꿔가며 학습시켜보았다. 비율이 달라질 때 마다 정확도가 꽤 큰 폭으로 달라지는 것을 확인하였고 drop out 비율을 잘 조절해야 할 것이라고 예측하였다.

```
#epoch 50, data 정규화, SGD, learning rate 0.001, drop out 0.2, weight initialization
model.eval()
ComputeAccr(test_loader,model)
```

Accuracy of Test Data: 71.5199966430664

```
[ ] #epoch 50, data 정규화, SGD, learning rate 0.001, drop out 0.3, weight initialization
model.eval()
ComputeAccr(test_loader,model)
```

Accuracy of Test Data: 62.89999771118164

```
[ ] #epoch 50, data 정규화, SGD, learning rate 0.001, drop out 0.25, weight initialization
model.eval()
ComputeAccr(test_loader,model)
```

Accuracy of Test Data: 66.69999694824219

2. learning rate 0.001에서 0.002로 바꿔가며 학습시켜 보았고 알맞은 learning rate도 찾는 것이 중

요하다고 느꼈다.

```
[ ] #epoch 50, data 정규화, SGD, learning rate 0.002, drop out 0.2, weight initialization
model.eval()
ComputeAccr(test_loader,model)
```

Accuracy of Test Data: 72.1500015258789

```
[ ] #epoch 50, data 정규화, SGD, learning rate 0.002, drop out 0.25, weight initialization
model.eval()
ComputeAccr(test_loader,model)
```

Accuracy of Test Data: 69.81999969482422

3. drop out을 바꿨을 때 정확도에 영향을 끼치는 것 같아서 CNN의 마지막 Linear 층에만 추가시켜서 학습시켜 보았다. 정확도가 상승하는 것을 확인하였다.

```
▶ #epoch 50, data 정규화, SGD, learning rate 0.002, 마지막에만 drop out 0.2, weight initialization
model.eval()
ComputeAccr(test_loader,model)
```

Accuracy of Test Data: 70.63999938964844

```
[ ] #epoch 50, data 정규화, SGD, learning rate 0.002, 마지막에만 drop out 0.25, weight initialization
model.eval()
ComputeAccr(test_loader,model)
```

Accuracy of Test Data: 70.62999725341797

```
[ ] #epoch 50, data 정규화, SGD, learning rate 0.002, 마지막에만 drop out 0.3, weight initialization
model.eval()
ComputeAccr(test_loader,model)
```

Accuracy of Test Data: 71.06999969482422

```
[ ] #epoch 50, data 정규화, SGD, learning rate 0.002, 마지막에만 drop out 0.4, weight initialization
model.eval()
ComputeAccr(test_loader,model)
```

Accuracy of Test Data: 72.19999694824219

4. 그래서 Linear층과 아닌 층의 dropout 비율을 다르게 설정해보았다. 이때 정확도가 가장 상승하였다.

```
▶ #drop out 0.25, 마지막에는 dropout 0.4 나머지는 같게
model.eval()
```

```
ComputeAccr(test_loader,model)
```

Accuracy of Test Data: 75.38999938964844

아! drop out 비율이 핵심이구나

```
[ ] # drop out 0.3, 마지막에는 dropout 0.4 나머지는 같게
model.eval()
ComputeAccr(test_loader,model)
```

Accuracy of Test Data: 74.18000030517578

```
[ ] #drop out 0.2, 마지막에는 dropout 0.4 나머지는 같게 , weight initialization 안함
model.eval()
ComputeAccr(test_loader,model)
```

Accuracy of Test Data: 74.65999603271484

5. epoch를 늘리면 정확도가 올라갈 것이라고 예상해서 epoch를 조정하여 학습시켜 보았다. 하지만 별다른 차이는 없는 것 같았다.

```
[ ] # drop out 0.2, 마지막에는 dropout 0.4 나머지는 같게 , weight initialization 하고, learning rate 0.02, epoch 70
model.eval()
ComputeAccr(test_loader,model)
```

Accuracy of Test Data: 74.58000183105469

```
[ ] #drop out 0.2, 마지막에는 dropout 0.4 나머지는 같게 , weight initialization 하고, learning rate 0.02, epoch 30
model.eval()
ComputeAccr(test_loader,model)
```

Accuracy of Test Data: 74.27999877929688

```
[ ] #drop out 0.2, 마지막에는 dropout 0.4 나머지는 같게 , weight initialization 하고, learning rate 0.02, epoch 50
model.eval()
ComputeAccr(test_loader,model)
```

Accuracy of Test Data: 75.16999816894531

6. scheduler의 step size를 다르게 해서 학습시켜 보았다. 큰 변화는 없었지만 step size도 영향을 조금 끼치는 것 같았다.

```
[ ] #drop out 0.2, 마지막에는 dropout 0.4 나머지는 같게 , weight initialization 하고, learning rate 0.02, epoch 50 scheduler 25
model.eval()
ComputeAccr(test_loader,model)
```

Accuracy of Test Data: 75.0199966430664

```
[ ] # drop out 0.2, 마지막에는 dropout 0.4 나머지는 같게 , weight initialization 하고, learning rate 0.02, epoch 50 scheduler 30
model.eval()
ComputeAccr(test_loader,model)
```

Accuracy of Test Data: 74.50999450683594

7. 그래서 drop out 비율을 linear 층에는 0.4, 나머지는 0.2로 고정하였고, weight initialization 은 안 한 것 보다 한 것이 더 정확도가 높았다. Learning rate 도 0.02 일 때가 가장 높아서 결국 Batch\_size=16, Learning rate=0.02, num\_epoch=50, data normalization, dropout 0.2(linear 0.4), weight initialization, SGD, scheduler step size=20 옵션의 모델을 만들었다.

```
[9] # dropout 0.2마지막에는 dropout 0.4 나머지는 같게 , weight initialization 안하고, learning rate 0.02, epoch 50
model.eval()
ComputeAccr(test_loader,model)
```

Accuracy of Test Data: 74.29999542236328

```
[ ] # dropout 0.2마지막에는 dropout 0.4 나머지는 같게 , weight initialization 하고, learning rate 0.02, epoch 50
model.eval()
ComputeAccr(test_loader,model)
```

Accuracy of Test Data: 76.02999877929688