

CIFAR10 인식 정확도 챌린지 (중간 대체 과제)

[21-1] 심층학습 / 201810793 이연우 (21.04.23 ~ 21.05.08)

1. 최종 정확도

최종 정확도 82.69

```
netname = './gdrive/MyDrive/201810793_Model.pkl'  
model = torch.load(netname) #로드 가져오기  
ComputeAccr(test_loader,model)
```

/usr/local/lib/python3.7/dist-packages/ipykernel_

Accuracy of Test Data: 82.69999694824219

2. 실험 일지

- 실험 방법 :

상명대학교 심층학습과목의 7주차에서 제공된 '[04실습]심층신경망 훈련'을 커스터마이징 하였다. '[04]심층신경망 훈련'에서 배운 훈련 방법 조합하여 test 세트에 대한 정확도를 최대한 높였다..

훈련 방법에는 Drop Out, Data Augmentation, Weight Initalization, Data Normalization, Batch Normalization, Adam optimizer, Learning rate decay를 배웠으며, epoch, Learning rate, Batch_size, Activation function등을 조합하여 약 130번의 실험을 진행하였다.

- 시도한 방법 간단 서술 :

Epoch = 60, Learning rate = 0.002, Batch_size = 16, Activation Func = ReLU 기본값 설정

(1) 수업에서 배운 각각의 기법들 하나만 선택하여 훈련

기법을 하나씩 사용하여 훈련한 결과 특출나게 정확도가 오른 기법들이 있었다. Weight Initialization, Batch Normalization, Adam optimizer, Learning rate decay를 사용했을 때 각각 약 **50이 넘는 정확도**가 나왔고, 앞으로의 훈련들을 위 4개의 기법들을 중심으로 훈련을 하고자 결정하였다.

(2) (1)에서 선정한 기법들을 조합하여 가장 높은 정확도가 나온 조합 선정

Weight Initialization, Batch Normalization, Adam optimizer, Learning rate decay를

모두 사용하여 정확도를 측정했을 때 가장 높은 **정확도 75.55**가 나왔다.

(3) Drop Out의 수치 선정

(2)에서 사용한 기법들에 Drop Out을 추가하여 훈련을 시켜보았다. 0.1, 0.2, 0.25, 0.3의 확률을 넣어 실험한 결과, **0.25에서 정확도 76.37**로 가장 높은 결과가 나왔다.

(4) Activation Function 변경

'[04실습]심층신경망 훈련'에서 제공된 코드에 기본 설정된 ReLU를 LeakyReLU, RReLU, PReLU, ELU 함수로 수정해 훈련을 해 보았다. 활성화 함수 중에 **PReLU를 사용했을 때 정확도 76.97**로 가장 높은 결과가 나왔고 이후로는 PReLU만 사용해 실험을 하였다.

(5) Epoch, Batch_size 변경

Epoch과 Batch_size를 크게 하면 정확도가 더 높게 올라 갈 것이라 예측하여 Epoch은 60 ~ 200, Batch_size는 16, 32, 64로 다양하게 조합하여 정확도를 측정해보았지만 **모든 조합의 훈련들이 결과가 비슷**하다는 결론이 나왔다. 가장 짧은 시간이 걸리는 Epoch = 60, Batch_size = 16으로 앞으로의 실험을 해보고 가장 높은 정확도가 나왔을 때 Epoch의 크기를 크게 해보기로 하였다.

(6) Image Data 변형

Data Augmentation과 Data Normalization의 값을 바꿔보며 각각 훈련을 시켜보았다. Data Augmentation은 좌우반전, 회전, 확대, 축소등을 조합해보았고, Data Normalization은 평균과, 표준편차의 수치를 데이터의 결과를 보며 수정해보았다. 수많은 실험을 해 본 결과 Data Augmentation에서는 좋은 조합을 찾기가 어려웠고, **Data Normalization**에서 평균(mean) = (0.5, 0.5, 0.5), 표준편차(std) = (0.2, 0.5, 0.8)로 설정했을 때 **정확도 77.18**이라는 결과가 나왔다.

(7) Learning rate decay 수치 선정

Step_size와 gamma의 크기를 각각 5, 10, 20, 30와 0.1, 0.2로 바꿔보며 실험해 보았다. 수치를 조합하여 훈련 결과 많은 정확도는 오르지 않지만 Step_size = 10, gamma = 0.2 일 때 **정확도가 77.47**로 전보다 0.3 오른 것을 볼 수 있었다.

(8) Drop Out 재실험

Drop Out을 모든 레이어에 사용할 필요가 없다는 것을 알고 Drop out을 사용할 부분과 안 할 부분을 구분하여 실험을 해봤다. 결과 가장 마지막 레이어와 그 전 레이어에 Drop Out을 사용하면 정확도가 대폭 오른 것을 알 수 있었다.

실습 코드에는 Drop Out이 0.2로 설정 돼있지만 보통 0.5을 많이 사용한다고 하여 Drop Out 할 비율을 0.5로 바꿔 보았다. (3)번에서 구한 0.25일 때 보다 0.5가 훨씬 좋은 결과가 나왔다. 마지막 레이어 2개에 **Drop Out을 0.5로 적용했을 때 정확도 79.25**가 나왔다.

(9) Linear 늘리기

마지막 실험으로 Linear을 2개를 더 늘렸더니 훈련을 시켰더니 **정확도 82.69**로 지금까지 중에 가장 높은 정확도가 나왔다.

3. 최종 결과

- Epoch = 120, Learning rate = 0.002, Batch_size = 16
- Activation Function : PReLU
- Data Normalization : mean (0.5, 0.5, 0.5), std (0.2,0.5,0.8)
- Drop Out : 0.5 (부분만 사용)
- Weight initialization
- Batch Normalization
- Optimizer : Adam optimizer
- Scheduler : StepLR (step_size = 10, gamma = 0.2) (Learning rate decay)

을 이용하여 훈련한 결과 **정확도 82.69**로 가장 높게 나왔다.

4. 실험 후기

수업을 들었을 때는 정확도를 올리는 게 이렇게 어려운 일인지 몰랐었다. 조금만 수치를 수정해도 정확도가 급격하게 변해서 처음에는 어떻게 하면 정확도가 오를지 보이지 않아 막막했는데 챌린지를 진행하는 과정에서 실험을 거듭하며 점차 감을 잡을 수 있었다.

결과적으로 82.69라는 정확도를 구할 수 있었지만 지난 학기 수강생 중에 87.4의 정확도를 한 사람을 보고 어떻게 하면 저렇게 높은 정확도가 나올 수 있었는지 궁금증이 생겼고 좀 더 인공지능에 관해 공부해야 할 필요성을 느꼈다.

약 2주 동안 심층신경망 훈련을 커스터마이징 하는 과정에서 이론 수업으로는 알지 못했던 부분을 실험해보며 인공지능 학습의 정확도를 높이기 위한 방법에 좀 더 친숙해질 수 있었다.