Assignment1

2016025196

김동규

실행 환경

Google colab, pytorch, python

코드 설명

Assignment1 코드는 기존 assignment1.ipynb파일에서 Classifier와 main실행코드를 수정하여 만들어졌습니다.

해당 코드에는 Pytorch의 Dropout, BatchNormal, Adam API를 통하여 구현되었습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

Classifier는 총 4개의 layer로 되어있으며, Layer1(32\*32\*3,32\*32), Layer2 (32\*32,512), Layer3(512,256), Layer4(256,10)으로 구성되었습니다.

이에 따라 총 세개의 BatchNormal이 선언되어 있으며, 이미지 분류라는 특성상 1,2,3 Layer는 ReLU activation을 사용합니다. 실험결과 layer 3에도 ReLU를 사용하는 것이 가장 정확도가 높았습니다.

Dropout의 확률은 여러 실험 결과 35%가 가장 적절한 수치임을 알 수 있었습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

학습 코드는 main문 내에서 구현되었으며, 이전 수업에서 나온 예시 코드를 활용하여 작성하였습니다. gpu가 사용가능한지 여부를 확인하여 사용하는 코드와, train\_dataloader를 추가한 뒤, batch size와 epochs에 따라 모델의 train을 진행합니다. Optimizer로는 Adam을 사용하였으며, lr=0.001, betas=(0.9,0.999)일 때 가장 성능이 높은 것을 확인하여, 이 수치는 조정하지 않았습니다. 진행상황과 cost값을 보기 위해 매 epoch마다 진행상황을 출력합니다.

실험 결과

아래는 첫 작성 코드입니다.

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

이러한 결과를 얻은 뒤, 여러 수치들을 조정해가며 결과값이 좋아지던 결과만 작성했습니다.

epochs=30

ReLU사용

batchsize=64

54%(캡처본 사라짐) 드랍아웃 적용(30%)

linear2 256,256로 수정

레이어3 256,128 추가

epoch 25

텍스트이(가) 표시된 사진

자동 생성된 설명1레이어 32\*32\*3, 32\*32 수정

2레이어 32\*32, 512 3레이어 512,256 수정

drop30% , epoch 29 및 layer3에 sigmoid사용

input nomalization적용, dropout 32%

dropout 35%

최종 모델 : epochs=29, Dropout Probability=35%, batch size=64, Layer={(32\*32\*3, 32\*32), (32\*32,256), (256,128), (128,10)}

여러 차례의 실험 결과 59.35%의 정확도를 갖는 최적화된 모델을 생성하였습니다.

결론

1. epochs는 30을 기준으로 점차 overfitting으로 인해 정확도가 낮아진다.

2. 이미지 분류기 특성상 activation함수로 ReLU함수가 성능이 높다.

3. batch size는 32~256등 여러 결과를 사용해본 결과 64가 가장 성능이 좋다.

4. dropout은 25~35%가 성능이 가장 좋고, 최종 layer의 크기 기준으로 35%가 가장 성능이 높다.

5. Layer의 크기는 input size 32\*32\*3을 기준으로 Layer1(32\*32\*3, 32\*32), Layer2

(32\*32, 256), Layer3(256,128), Layer4(128,10)이 성능과 학습시간상으로 가장 성능이 좋다.