FashionMNIST data를 활용하여 아래 질문들에 대한 코드와 답을 작성하세요.

하이퍼 파라미터 – 기본 셋팅

텍스트, 폰트, 스크린샷, 라인이(가) 표시된 사진

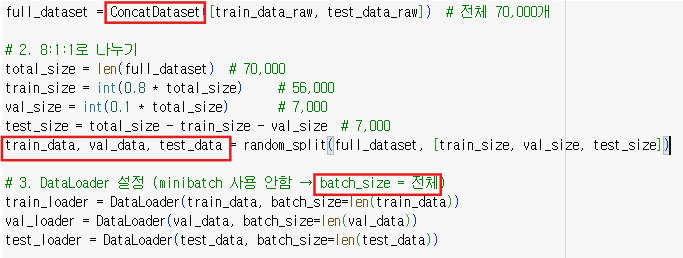
AI가 생성한 콘텐츠는 부정확할 수 있습니다.

모델 – 기본 셋팅 (추후 바뀜)

텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

– data를 8:1:1로 train / validation / test로 분리하고 2 hidden layer MLP 모델을 training 시키고 train / validation / test에 대한 accuracy를 각각 구하세요 (10)



문제 1번의 코드에서는 ConcatDataset를 사용하여 trian\_data,test\_data를 합친 70,000개의 데이터를 train:valid:test -> 8:1:1로 나눴고 그대로 모델의 돌려 train\_data로 (epoch=10)g 학습시킨 후 , valid와 test에 대한 accuracy도 측정하였다. 결과는 다음과 같다

텍스트, 폰트, 영수증, 화이트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

– minibatch를 사용한 SGD로 코드를 수정하고 train / validation / test에 대한 accuracy를 구하세요 (10)

텍스트, 폰트, 스크린샷, 라인이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

문제 2번에서는 batch사이즈 64로 minibatch 적용 + optimizer는 SGD를 사용하였다.

결과는 다음과 같다.

텍스트, 폰트, 화이트, 영수증이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

결과를 봤을 때 minibatch를 사용하는 것이 데이터를 한번에 올리지 않고 작은 크기로 나눠 GPU의 병렬 처리 성능을 잘 활용할 수 있어서 학습시간도 현저히 줄었고 수치를 봤을 때 accuracy도 향상 되었다.

– data normalization을 적용 했을 때 어떤 변화가 있는지 epoch / loss plot을 그려서 비교하세요 (10)

텍스트, 폰트, 스크린샷, 라인이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

위 코드처럼 정규화를 통해 각 픽셀의 0부터 255사이의 정수를 -1.0 ~ 1.0 사이의 실수로 바꿨다.

결과를 보면,

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.라인, 그래프, 도표, 경사이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

Train 과 test 모두 loss가 지속적으로 줄어들고 있고 acc도 증가하여 정규화함으로써 더 안정적이고 빠른 수렴을 보였다.

– optimizer를 SGD / SGD+Momentum / SGD+Nestrov Momentum / AdaGrad / RMSProp / Adam 를 사용 하였을 때 어떤 결과가 있는지 epoch / loss plot을 그려서 비교하세요. (하나의 plot에 나오도록 해야함) (20)

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

위 리스트로 opimizer를 하나씩 바꿔가며 각 opimizer로 학습한 결과를 plot으로 그린결과

라인, 그래프, 텍스트, 도표이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

SGD보다 다른 Optimizer가 더 빠른 수렴을 보였고 momentum과 Nesterov는 비슷한 양상을 보였지만 AdaGrad,RmsProp, Adam은 그 보다 더 빠른 수렴을 보여줬다.

수업에서 배웠던 결과를 교과서적으로 보여주는 plot임을 확인할 수 있다

– Dropout을 적용했을 때 성능 변화를 epoch / loss / accuracy를 사용하여 비교 하세요 (10)

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

Dropout은 비선형 활성화된 feature 중 일부를 제거해야 의미있다고 생각하여 저 위치에 뒀다. 결과를 확인해보니

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

라인, 그래프, 도표, 텍스트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

네트워크가 특정뉴런에 의존하지 않고 일반화 성능이 올라간 것을 Accuracy가 85% 이상으로 올라갔다 즉 , 네트워크가 특정뉴런에 의존하지 않고 일반화 성능이 올라간 것으로 판단했다.

– MLP model에 batch normalization layer를 추가 했을 때 어떤 변화가 있는지 답하세요 (10)

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

다음과 같이 nn.BatchNorm1d를 ReLu() 전에 두어서 **Covariate Shift**를 방지, 매 스텝마다 hidden layer에 입력으로 들어오는 데이터의 분포가 달라지는 것을 방지한 결과

텍스트, 폰트, 스크린샷, 번호이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

Test데이터에 대한 Accuracy가 88.76%로 가장 높은 수지를 기록했다.

– 위의 방법들과 hyper parameter tuning을 통해 모델의 성능을 향상시키고 결과를 report 하세요 (30)

지금까지 optimizer,minibatch,dropout,Batchnorm1d 등 다양한 방법으로 과적합을 방지하고 모델을 빠르게 수렴하는 과정을 통해 accuracy를 높였고

Hyper parameter tuning 같은 경우 leaning rate를 0.005으로 epoch = 20으로 하니 87로 떨어져서 lr은 0.001로 유지 epoch만 20으로 바꿔서

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

텍스트, 폰트, 영수증, 화이트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

train에서는 90%의 accuracy를 test에서도 그에 준하는 accuracy를 얻었다.