딥러닝 및 응용

dropout과 adam 적용

2015004475 김태훈

1. 구동 환경

   OS: Windows 10 64비트

   Language: Python 3.7

   환경설정은 conda를 이용하였으며 필요 패키지는 requirements.txt에 기재됨.

2. 코드 설명

   mnist 데이터 one-hot 인코딩 로드 (N개)

   X: mnist 이미지가 784픽셀이므로 [None, 784] 꼴의 인풋벡터를 placeholder로 구성. 추후 [N, 784] 형태가 될 것.

   Y: 각 샘플 당 0~9까지 라벨이 있으므로 길이 10짜리 라벨. 따라서 [N, 10]

   rate: 최신 tensorflow는 1 – keep_prob 의 값을 rate 라는 파라미터로 받아 사용.

   레이어 1, 2

   - sigmoid 함수를 activation 함수로 사용.
   - rate 만큼 뉴런을 disable.

   레이어 3

   - 출력 레이어. 10개의 뉴런으로 각각 라벨을 담당.
   - softmax 함수를 activation으로 사용.

cost: loss함수로 크로스 엔트로피를 사용.

opt: Adam 옵티마이저 사용. 각 파라미터는 흔히 사용하는 수치. cost를 minimize 하는 방향으로 작동함.

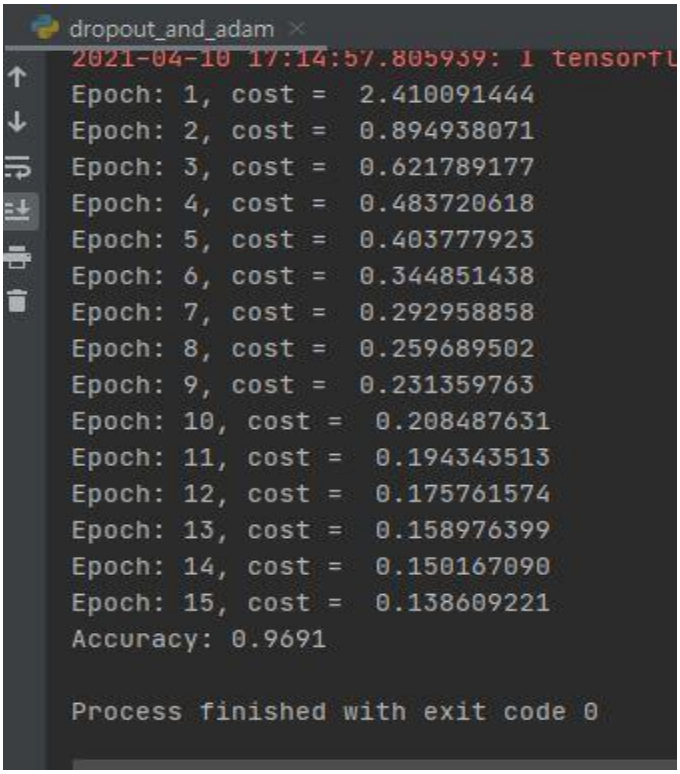
batch_size: 미니배치 사이즈. 100으로 설정 됨.

세션을 가동한 후...

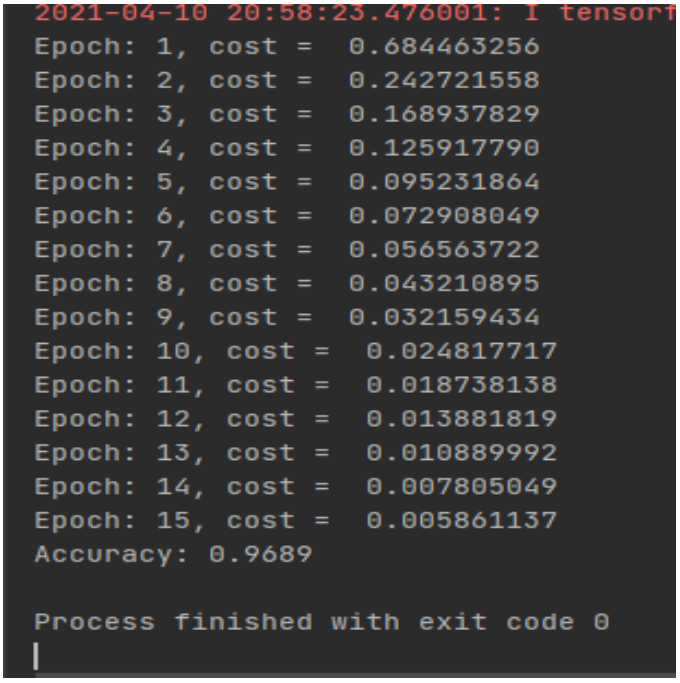
1. 텐서 variable을 랜덤 초기화.
2. 15회의 epoch을 반복하며
    A. 해당 epoch의 평균 코스트를 초기화.
    B. 샘플 총량을 batch 사이즈로 나눠 미니배치 개수 계산.
    C. 미니배치 개수만큼 반복하며
        i. 배치 사이즈만큼의 데이터를 뽑아냄
        ii. 미니배치만큼 뽑은 인풋데이터 batch_xs, 정답 데이터 batch_ys 그리고 dropout을 위한 rate를 지정하여 feed_dict로, fetch에 코스트와 옵티마이저를 파라미터로 세션을 작동.
        iii. 평균 코스트 계산.
    D. epoch 번호와 평균 코스트를 출력
3. is_correct에 hypothesis(소프트맥스 적용 후 출력값 = y햇)와 Y에 있는 정답이 서로 같은지 확인하여 T/F 할당.
4. is_correct를 이용해 정확도 계산식 정의.
5. 테스트 데이터를 모델에 적용하여 정확도 출력

3. 실험 결과

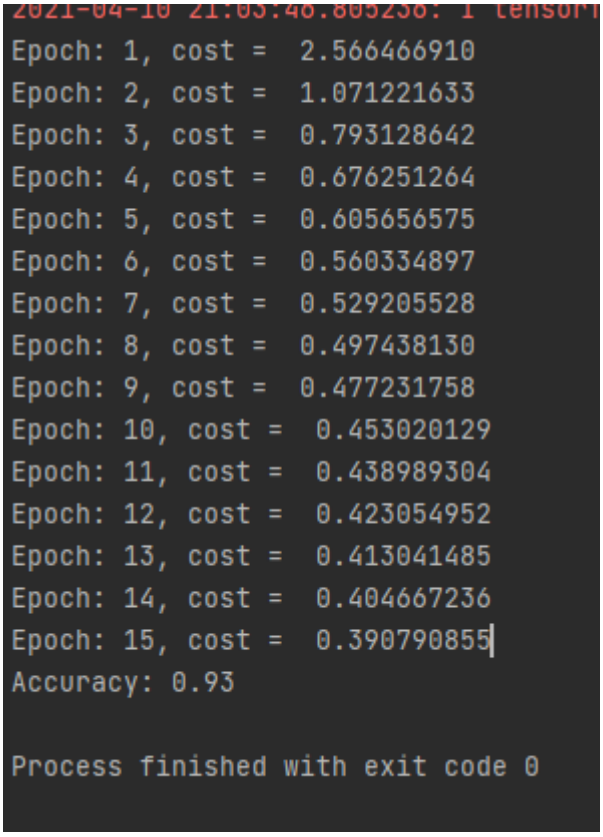
A. dropout과 adam 옵티마이저를 적용한 결과.



B. adam 옵티마이저만 적용한 결과.



C. dropout과 gradient descent 옵티마이저를 사용한 결과.



3종류의 코드를 실험한 결과 dropout과 adam 옵티마이저를 사용했을 때 정확도가 가장 높게 나타났다.