

# 9주차 과제

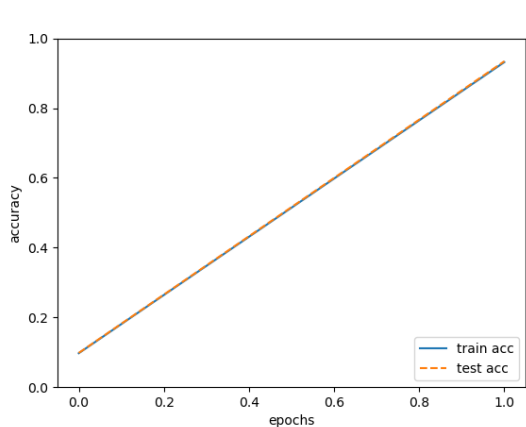
## <미니 배치 학습 구현하기>

1. Batch 사이즈 변경에 따른 훈련데이터와 테스트데이터의 정확도 추이 변화
2. Hidden Size 변화에 따른 정확도 추이 변화
3. 오버피팅 문제 기술

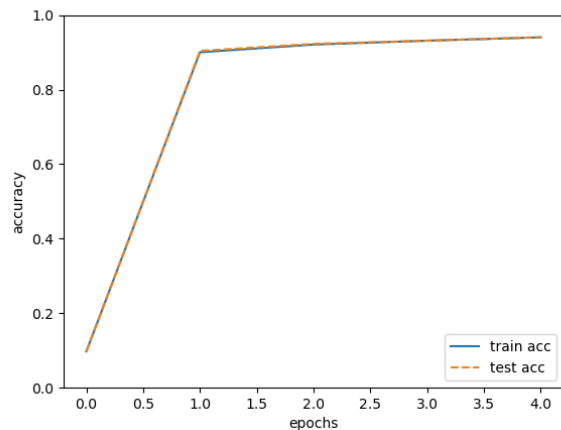
# 1. Batch 사이즈 변경에 따른 훈련데이터와 데이트데이터의 정확도 추이 변화

: Batch size가 작아질수록 정확도가 낮아지고, 커질수록 정확도는 올라가나 학습 시간이 오래걸린 것을 확인할 수 있었다.

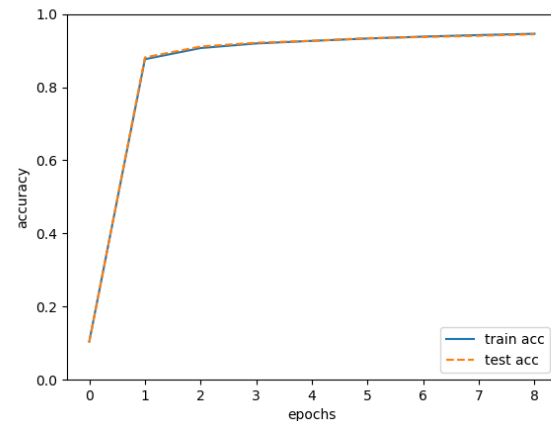
## <Batch Size별 학습 결과 그래프>



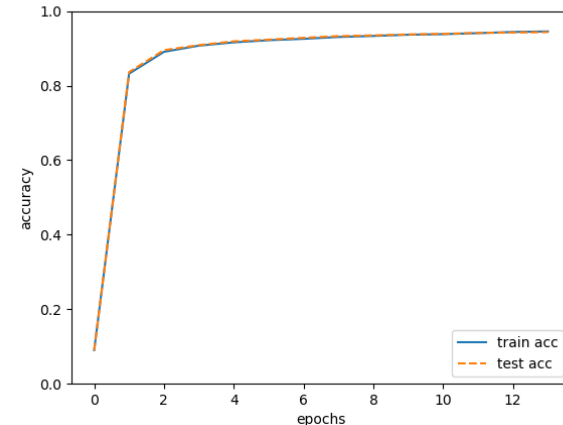
Batch Size : 10



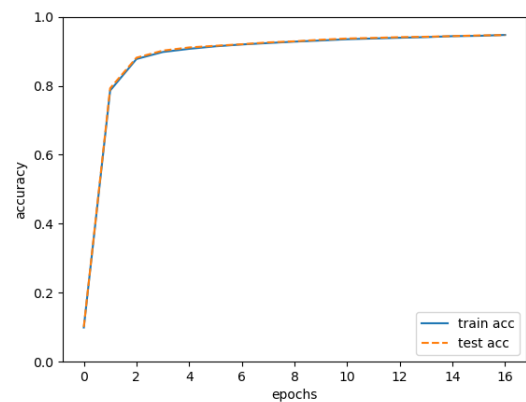
Batch Size : 30



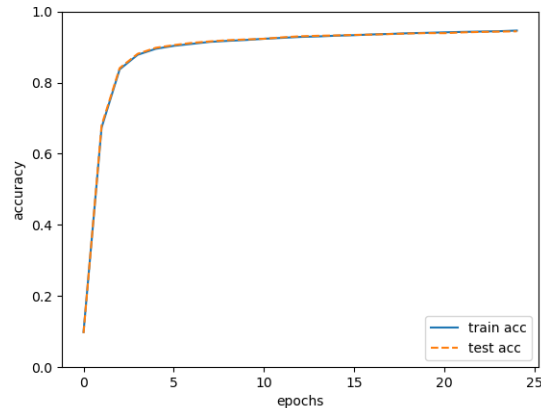
Batch Size : 50



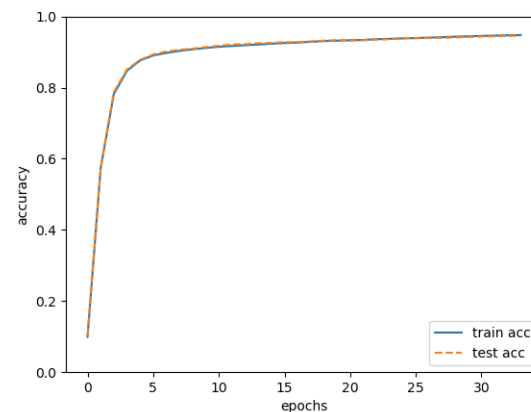
Batch Size : 80



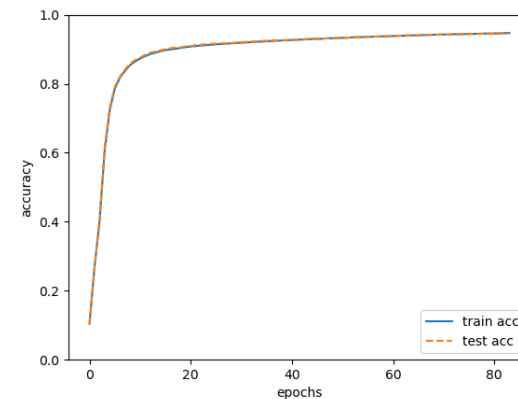
Batch Size : 100



Batch Size : 150



Batch Size : 200

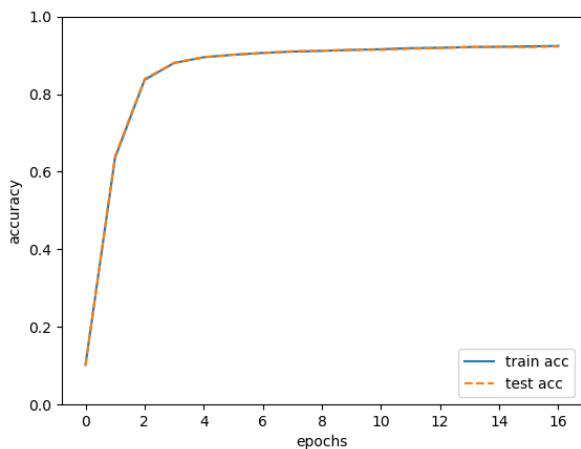


Batch Size : 500

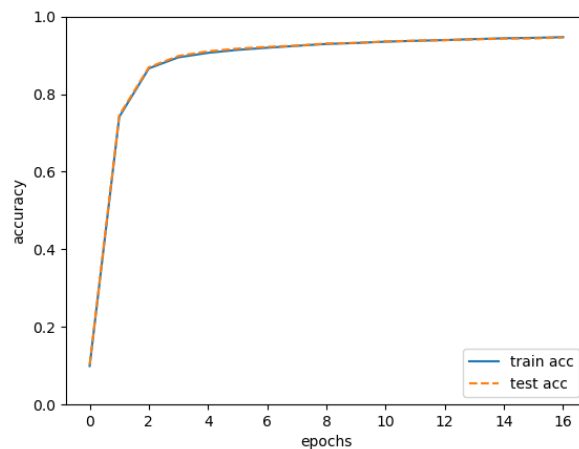
## 2. Hidden Size 변화에 따른 정확도 추이 변화

Batch Size의 변화에서의 정확도 추이보다는 변화폭이 작지만 Hidden Size가 작을수록 정확도가 살짝 떨어지는 것을 볼 수 있다. 하지만 일정 Size 이상이 되면 정확도의 변화 차이가 적다.

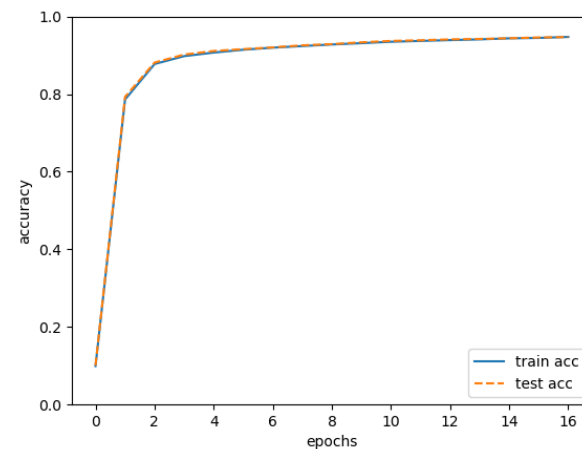
### <Hidden Size 별 학습 결과 그래프>



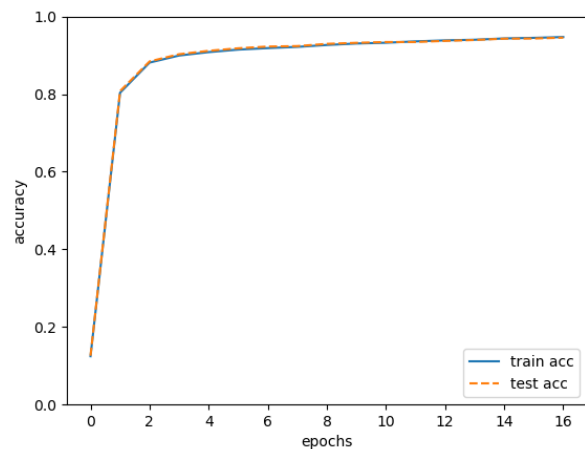
Hidden Size : 10



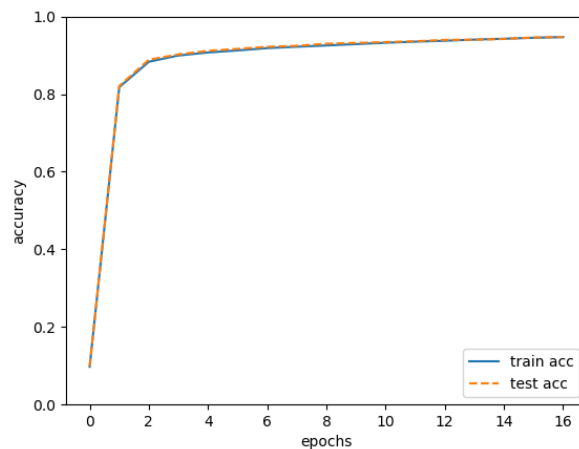
Hidden Size : 30



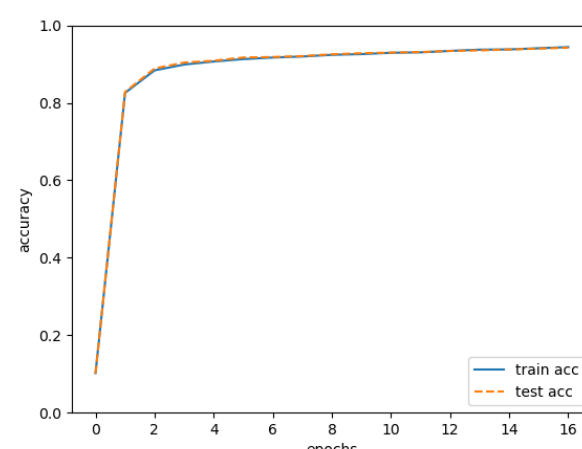
Hidden Size : 50



Hidden Size : 80



Hidden Size : 100



Hidden Size : 150

### 3. (1)과 (2)에서의 오버피팅 문제 기술

Batch Size와 Hidden Size가 작으면 정확도가 떨어지고 크면 정확도가 상승되나, 두 사이즈 모두 어느 순간이 넘어가면 변화의 차이가 적어짐을 확인할 수 있다.

(1)번 문제의 슬라이드에 보면 Batch\_Size 500일때는 train\_acc와 test\_acc가 거의 똑같은 값의 그래프로 나타난다. 이것이 오버피팅으로 보이며 학습 데이터가 너무 많아지면 그것에 포함된 것만 제대로 구분하는 결과가 나타나는 것이다.

오버피팅을 피하기 위해서는 적정선의 학습 Size를 찾아 적용해야 한다.