

목차

- MNIST
 - Layer를 추가했을 때와 아닐 때의 차이
 - 기존에 추가된 부분
 - 기존 결과
 - 2개의 Layer를 추가한 결과
 - Drop out을 적용했을 때
 - Drop out 25%
 - Layer를 5개로 늘렸을때
 - 이유가 무엇일까
 - 훈련에 제약을 더 걸어주면?(Drop Out 비율 증가)
 - Drop Out 비율: 30%
 - 결론
 - Layer
 - Drop Out
- CIFAR10
 - 기존 모델의 결과(No Layer, No Dropout)
 - 2-Layer, All Dropout(0.2)
 - 5-Layer, 2 Dropout (비율: 0.1)
 - 훈련량을 늘렸을 때, 과적합도는 어떻게 변화할까

MNIST

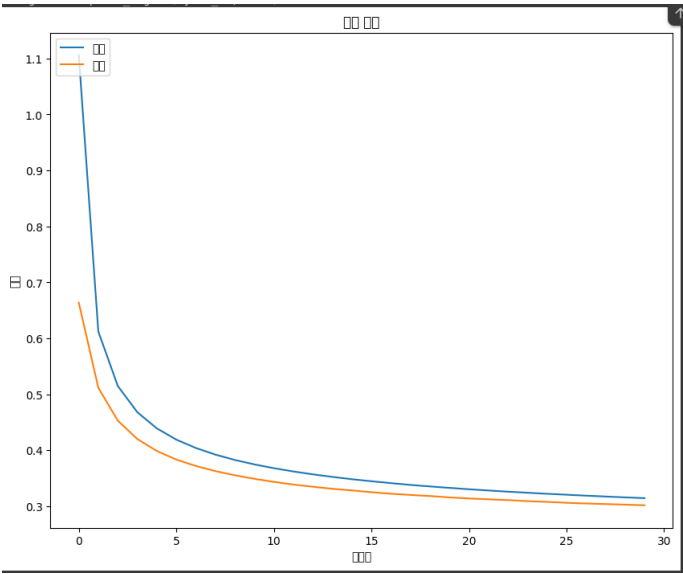
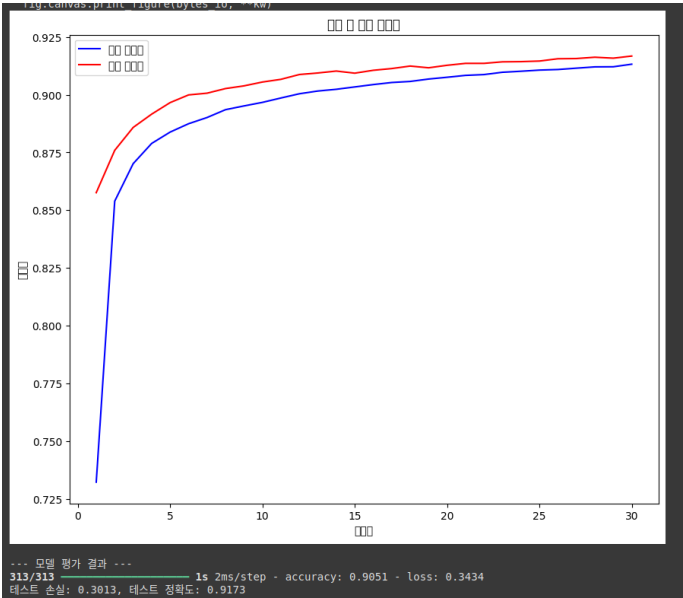
Layer를 추가했을 때와 아닐 때의 차이

기존에 추가된 부분

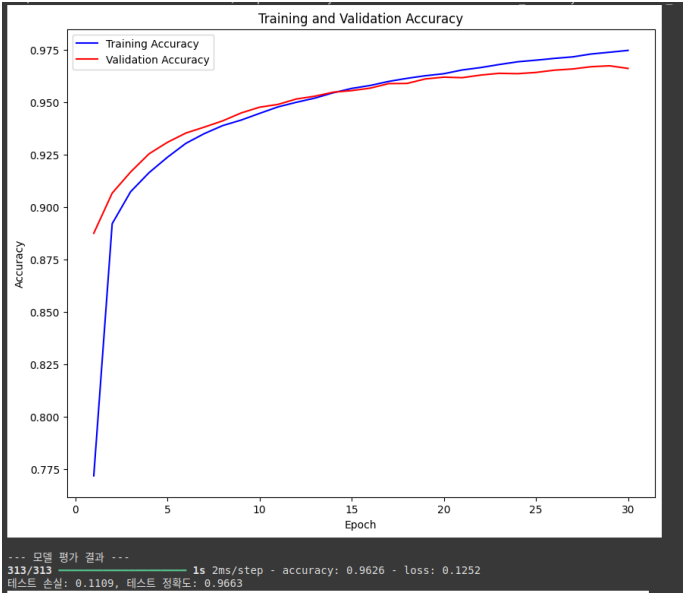
```
md = Sequential()
# 첫 번째 은닉층
md.add(Dense(128, activation='relu', input_shape=(28 * 28,)))
# 두 번째 은닉층
md.add(Dense(64, activation='relu'))
# 출력층: 10개의 클래스 (숫자 0-9), 활성화 함수: softmax (다중 클래스 분류)
md.add(Dense(10, activation='softmax'))
md.summary()
```

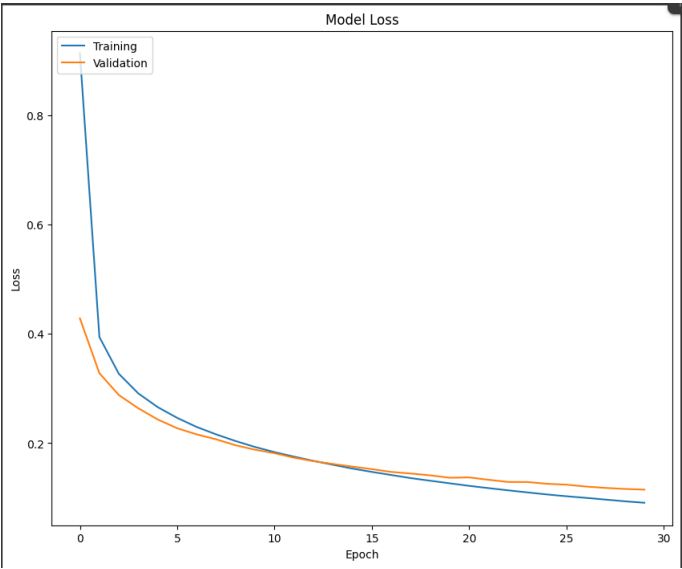
- 기존에는 단층 구조였음
- 2개의 Layer를 추가

기존 결과



2개의 Layer를 추가한 결과





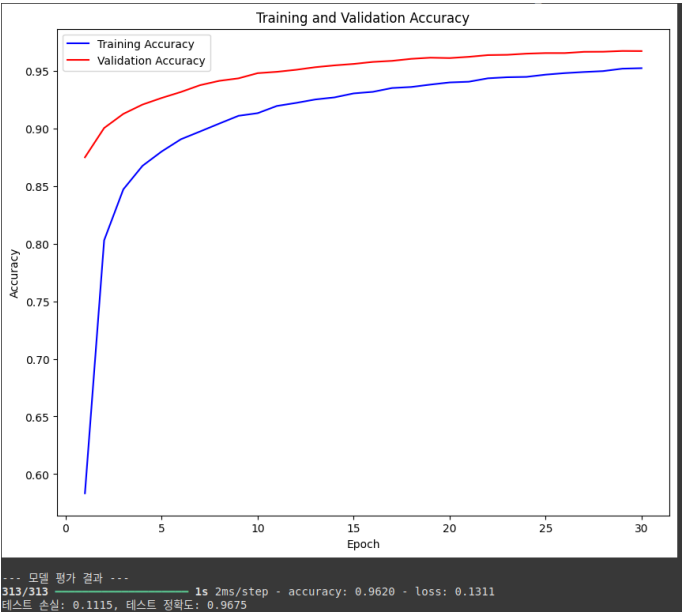
0	단층	2-Layers
ACC	0.9173	0.9663
손실도	0.301	0.11

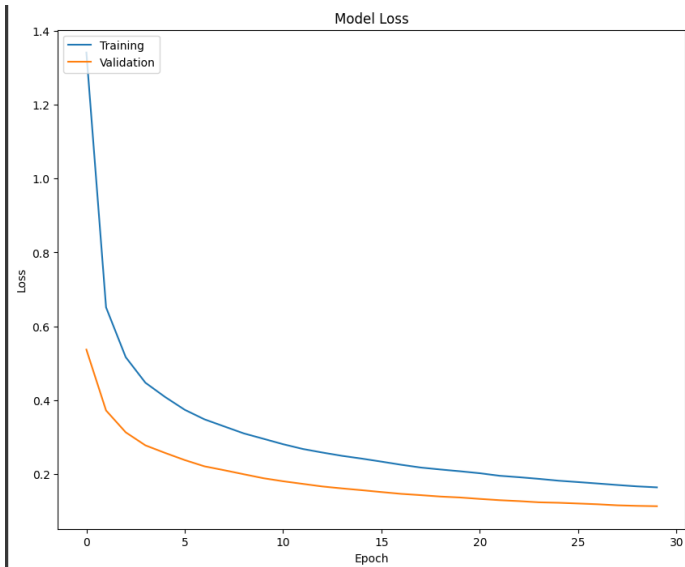
Drop out을 적용했을 때

Drop out 25%

2개의 Layer에 모두 추가

```
md = Sequential()
# 첫 번째 은닉층
md.add(Dense(128, activation='relu', input_shape=(28 * 28,)))
md.add(Dropout(0.25)) # Dropout 레이어 추가
# 두 번째 은닉층
md.add(Dense(64, activation='relu'))
md.add(Dropout(0.25)) # Dropout 레이어 추가
# 출력층: 10개의 클래스 (숫자 0-9), 활성화 함수: softmax (다중 클래스 분류)
md.add(Dense(10, activation='softmax'))
md.summary()
```



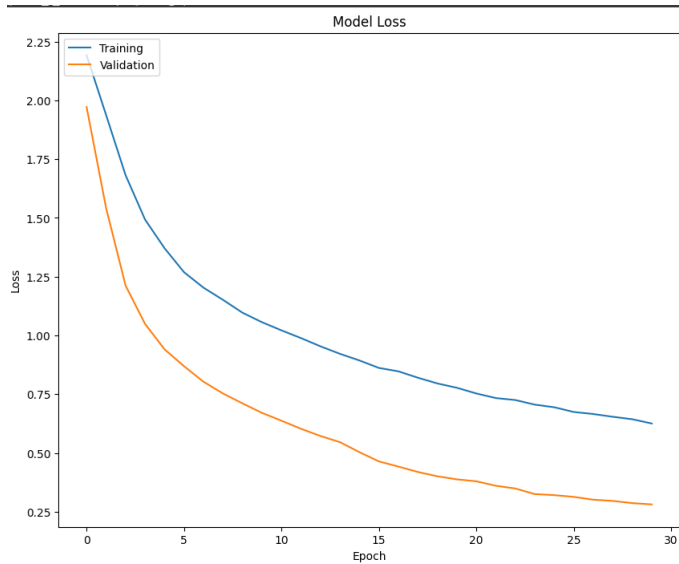
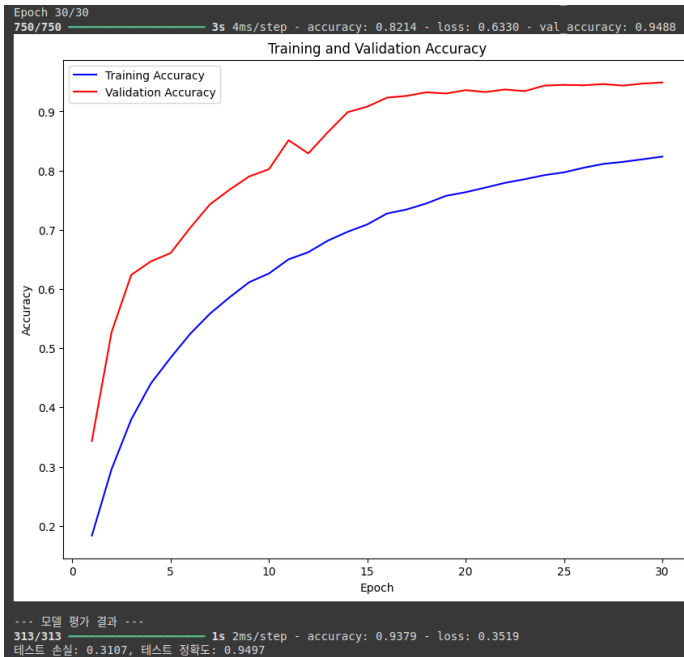


0	DropOut X	DropOut O
ACC	0.9663	0.9675

- Drop Out은 전체 뉴런이 아닌 일부 뉴런만 활성화 시키는 것을 의미함
- 과적합을 방지하기 위해 사용
- 모든 뉴런을 사용하지 않기 때문에 훈련 정확도는 떨어짐
- 검증/ 테스트 단계에서는 모든 뉴런을 활용
 - 훈련 시 비활성화했던 뉴런들을 활성화시킴
 - 때문에 더 안정적이고 성능이 향상되는 모습을 보임
- 훈련을 반복하면 과적합 문제가 발생할 수 있음
- Drop out이 없는 경우
 - 훈련횟수가 반복될수록 Training Accuracy와 Validation Accuracy 차이가 벌어짐
 - 과적합 문제
- Drop Out이 있는 경우
 - 훈련횟수가 증가할수록 두 간격의 차이가 줄어들음
 - Drop Out으로 인해 training에 제약이 걸려 없는 경우보다 항상 속도가 더딤
 - 대신 검증 단계에서 모든 뉴런들이 활성화되어 제약받았던 모델이 잠재력을 발휘함
 - 때문에 훈련을 반복할수록 두 간극의 차이가 줄어들음

Layer를 5개로 늘렸을때

동일하게 모든 층에 Dropout 존재



0	2-Layer	5-Layer
ACC	0.9675	0.9497

- 레이어 수를 늘린 것이 오히려 정확도가 떨어진 것을 확인 할 수 있다
- 심지어 검증 정확도와 훈련 정확도의 차이가 2-Layer에 비해 확연히 커진 것을 확인할 수 있다

이유가 무엇일까

- 레이어 수의 증가
 - 훈련 모델의 복잡도가 증가
 - 훈련 데이터의 미세한 패턴까지 학습할 수 있음
 - == 노이즈까지 학습할 수 있는 가능성 증가
- 훈련 데이터 암기 능력 증가
 - 복잡해질수록 암기 능력 향상
 - 훈련 횟수가 증가할수록 훈련 정확도가 계속 상승하는 모습을 보임
 - 2-Layer에서는 epoch 20부터 더딤

- 5-Layer에서는 epoch 30까지 계속 상승하는 모습을 보임
- 그러나 암기력이 향상되면
 - 훈련 데이터에 대해 특화됨
 - 새 데이터에 대해 성능이 저하됨
 - 과적합 문제 발생

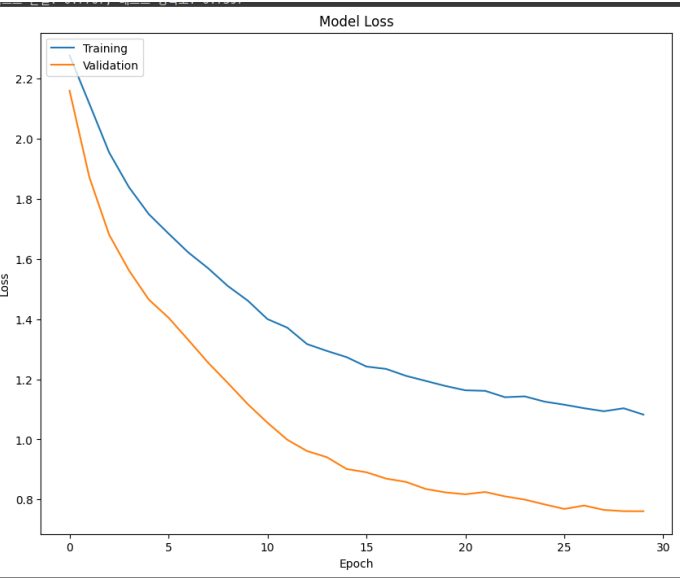
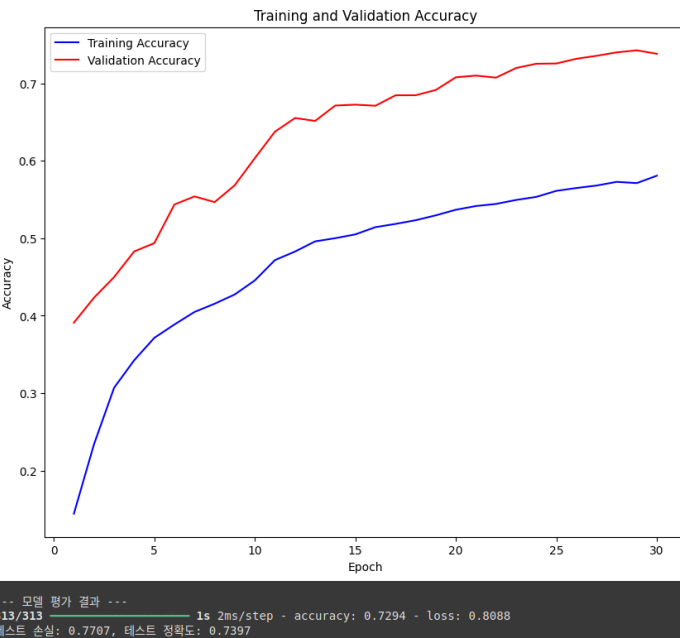
[결론]

- Layer 수를 늘리면, 훈련 능력은 좋아짐
- 대신 훈련 모델에 특화되어 새 데이터에 대한 성능이 떨어짐
- 과적합 문제가 심화된다

훈련에 제약을 더 걸어주면?(Drop Out 비율 증가)

기존 Drop Out 비율: 25%

40%까지 증가

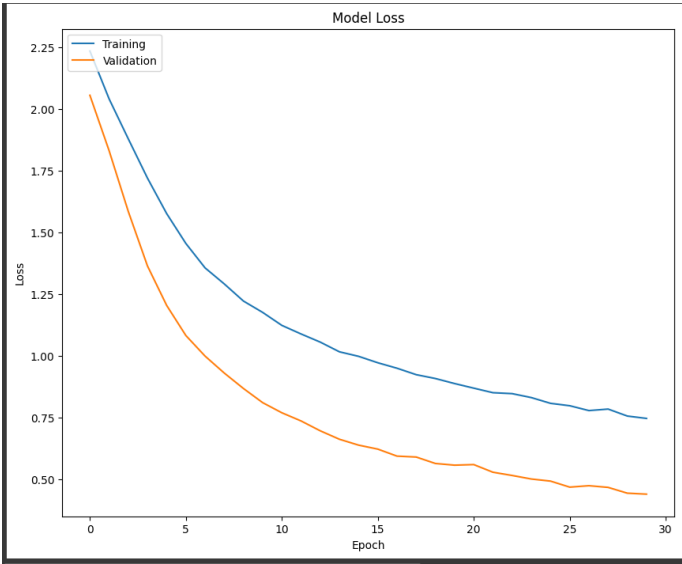
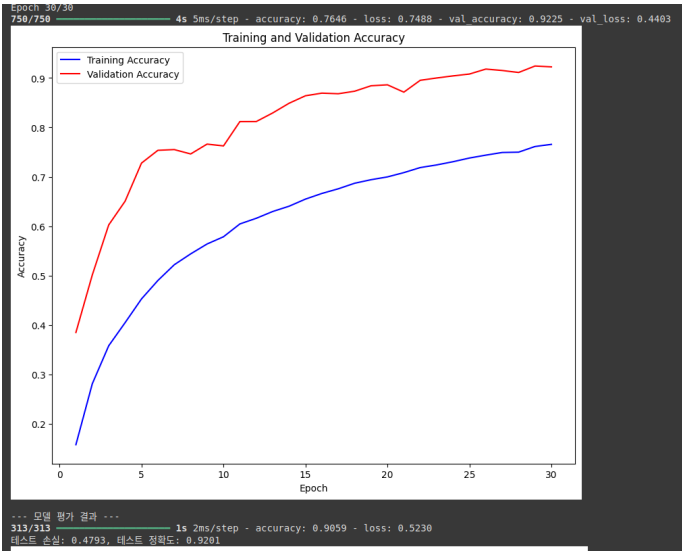


Drop_Out	0.25	0.4
ACC	0.9497	0.7397

- Drop out을 증가시켰더니 오히려 정확도가 떨어짐
- 검증 및 훈련 정확도의 차이도 더 증가함
- Underfitting 문제가 발생한 것으로 유추됨
 - 드롭아웃 비율이 너무 높아 훈련 자체를 제대로 수행하지 못함

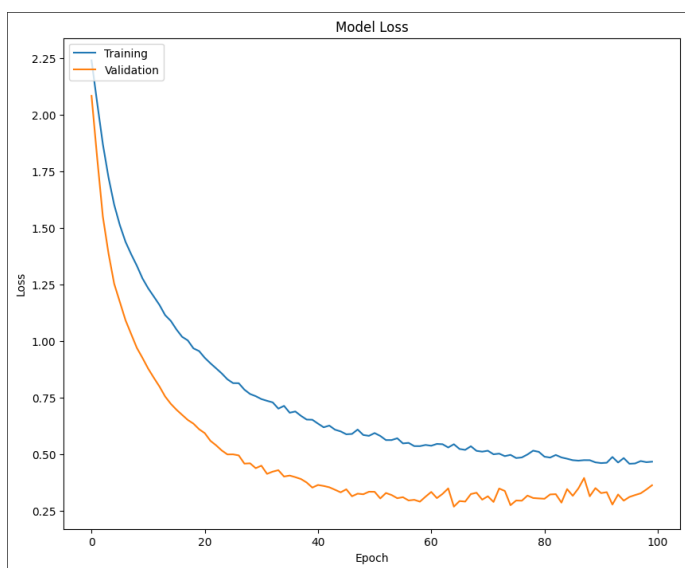
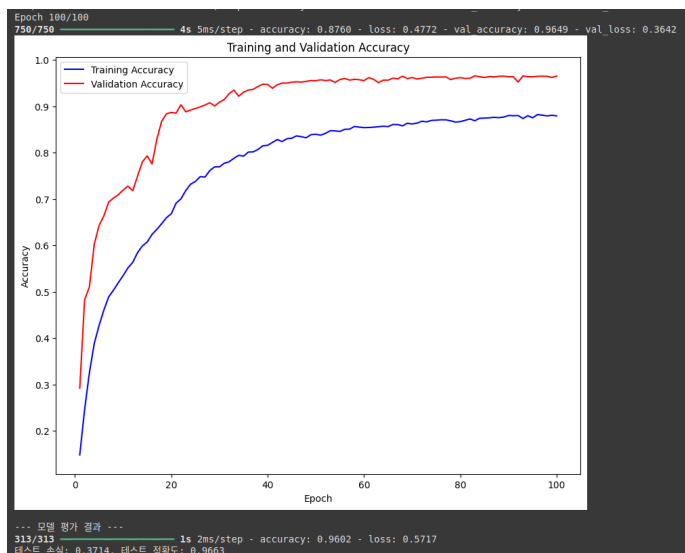
Drop Out 비율: 30%

훈련 30회 경우



- Drop Out이 40%때 보다 정확도가 증가한 것을 확인 가능
- 아직 검증 및 훈련 정확도의 차이가 크긴 하지만 발산하는 정도는 아님

훈련 100회 경우



- 훈련횟수가 증가하니 과적합도가 줄어든 것을 확인할 수 있음
- 다만, 2-Layer에 비해 과적합도가 아직 큰 것을 확인 가능

결론

Layer

- Layer를 추가하면 훈련 정확도가 늘어남
- 단, Layer의 수 증가가 정확도가 떨어짐
 - Layer 수 증가에 따른 복잡도가 증가하면
 - 훈련 모델의 암기력 증가
 - 훈련 모델에 대해서만 특화됨
 - 새 데이터에 대한 판단 능력이 저하됨
- 과적합도가 올라감

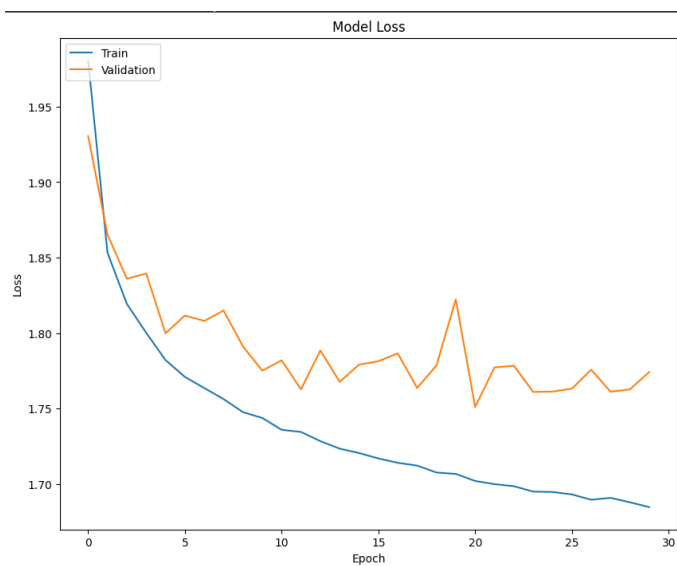
Drop Out

- Drop out을 통해 훈련과정에 제약을 걸어줌
- 적정 Drop Out은 과적합도를 해소함
- 그러나 Drop Out의 비율이 너무 높아지면 훈련량이 줄어듦

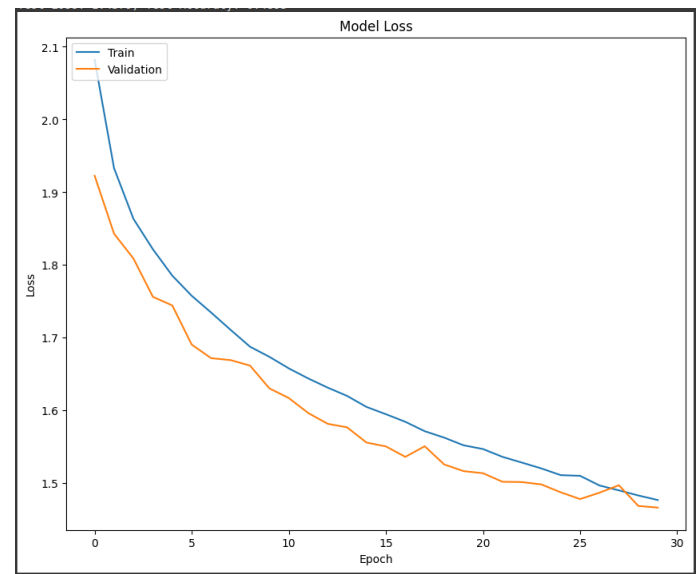
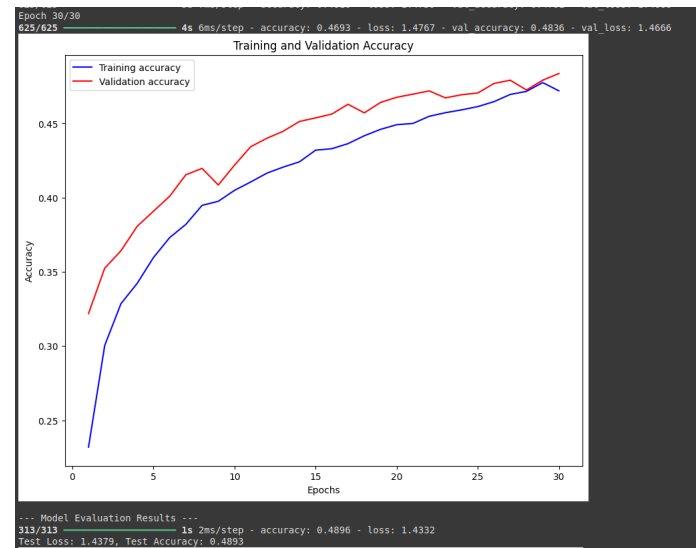
- 제대로된 훈련이 안됨
- Underfitting 문제
- 오히려 성능이 떨어짐
- 적정 Drop Out을 주어야 성능 향상과 과적합도 해소를 할 수 있다

CIFAR10

기존 모델의 결과(No Layer, No DropOut)



2-Layer, All DropOut(0.2)

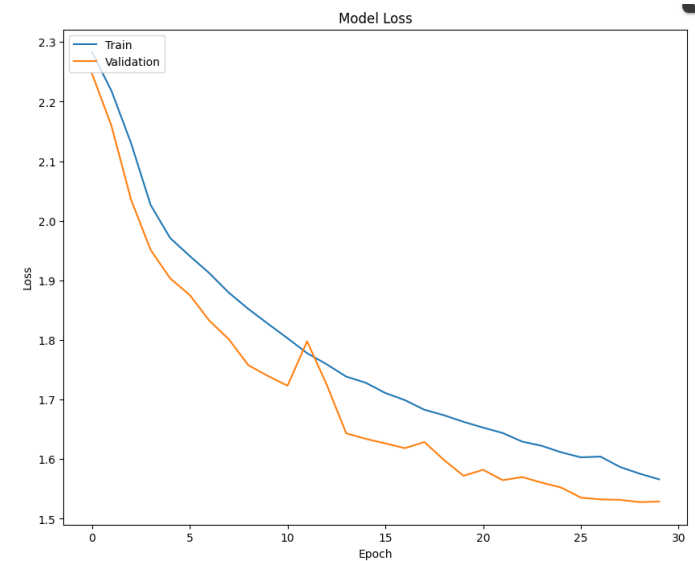
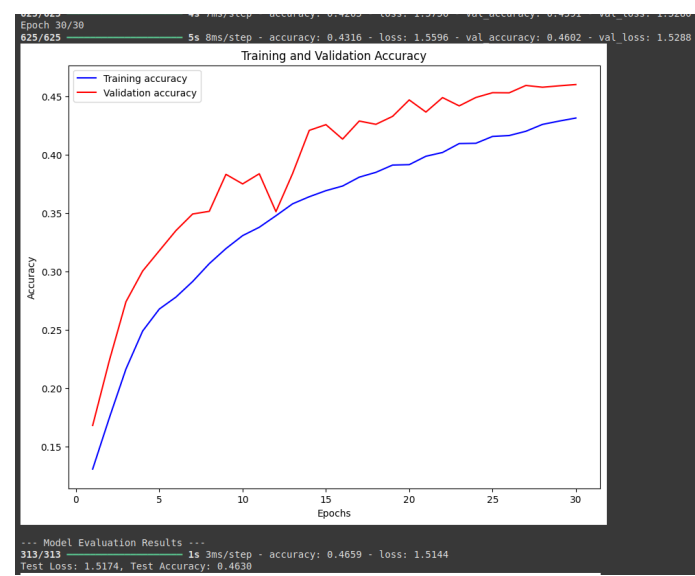


	0	No Layer_DropOut	With Layer_DropOut
ACC		0.38	0.49

- 정확도가 0.1정도 개선된 것을 확인할 수 있다
- 기존에 layer층이 없고, Drop out이 없을 때는 과적합도가 불규칙한 모습을 보였다
- 각 조건들을 추가한 후에는 부드러운 곡선 형태로 정확도가 개선되었으며, 과적합도 또한 많이 줄어든 것을 확인할 수 있다

5-Layer, 2 DropOut (비율: 0.1)

MNIST 모델에서 겪었던 문제들을 개선하여 CIFAR10에 적용해봄

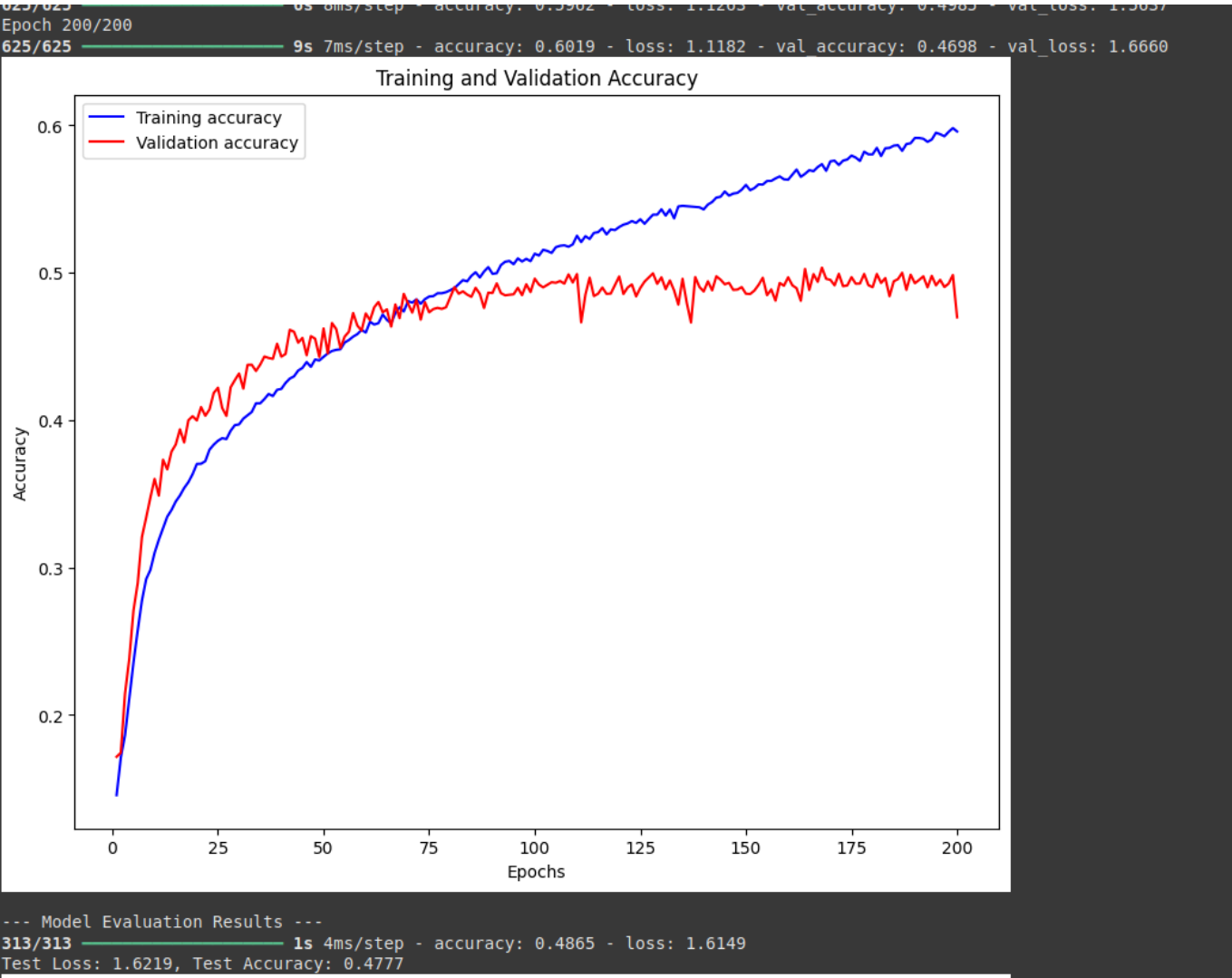


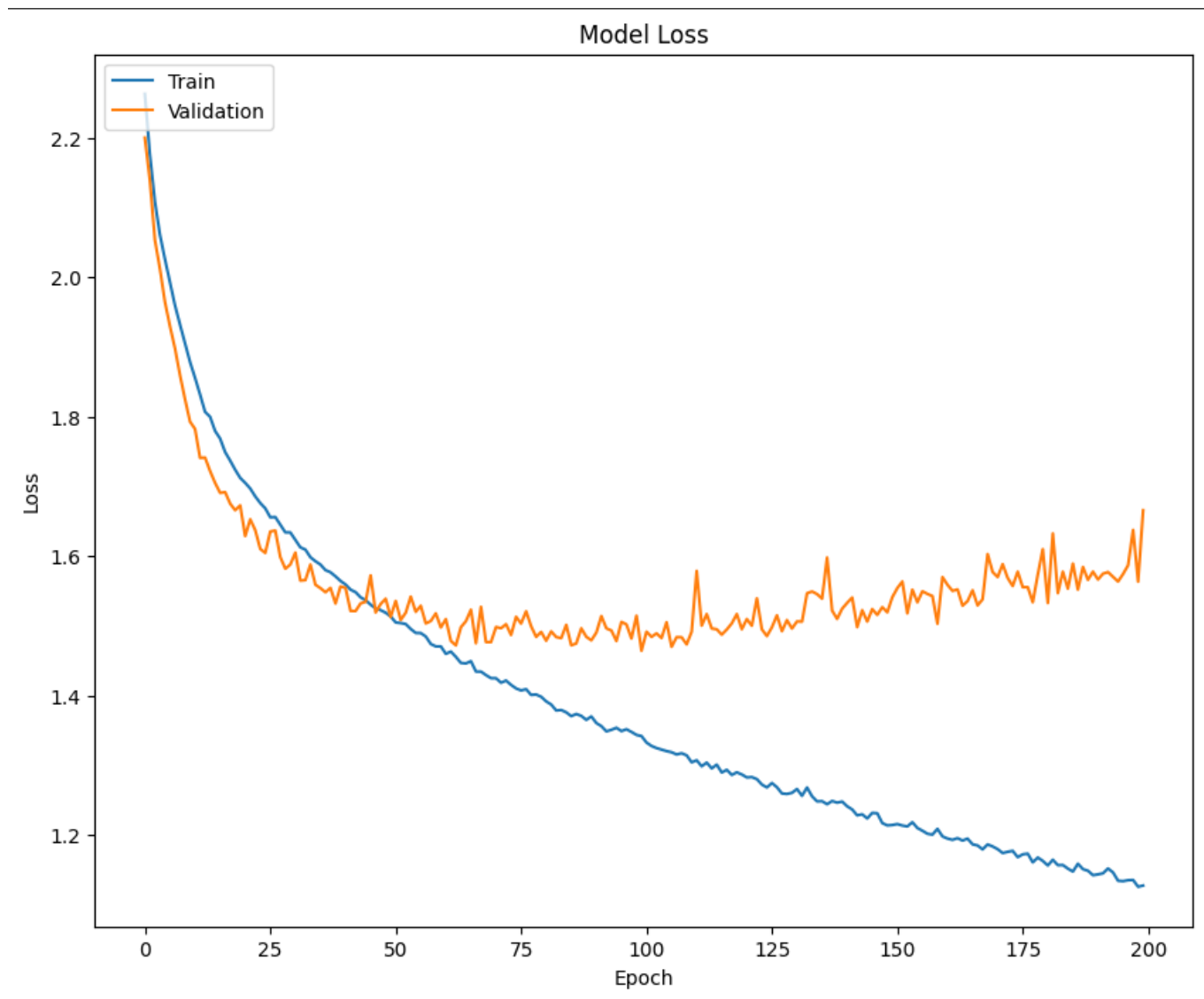
0	2-layer, 0.2_all	5-layer, 0.1-2drop
ACC	0.49	0.463

- 성능은 0.03 정도 저하됨
- 과적합도는 이전에 비해 훈련량이 늘어날수록 줄어드는 추세를 보임
- MNIST에서 조건들을 개선하여 넣은 결과
 - 성능 저하도 덜하고
 - 과적합도 추세선도 감소하는 방향으로 진행

훈련량을 늘렸을 때, 과적합도는 어떻게 변화할까

훈련량(epoch)를 200으로 늘리고 추세를 살펴본다





- 정확도의 경우 0.45정도로 감소함
- 훈련 정확도의 경우 학습횟수가 증가할수록 계속해서 상승하는 모습을 보임
- 단, 검증 정확도의 경우 약 50번의 학습횟수부터 일정한 값으로 유지되는 모습을 보임
- 학습 모델이 훈련 데이터셋에 최적화가 계속 이뤄지는 반면 실제 데이터에 대해선 일반화 성능이 떨어짐
 - Drop Out을 적용했지만 과적합도가 해소되지않음
- 학습량이 적을때는 두 간극의 차이가 줄어든 추세로 확인됐으나 실제 학습량을 늘리니 과적합도가 해소되지 않은 것을 확인할 수 있다

[결론]

- Layer의 추가나 Drop Out의 추가등 모델에 특성을 추가하면 분명 성능이 개선됨
- 단, 기능을 추가할 때, 그 수를 적정수준 유지하지 못하면 오히려 성능이 퇴보하는 결과를 가져옴
- 이러한 **적정수준**을 찾는 것이 관건임
 - 정확한 Trade-Off로 설명될 수 없는 부분이 학습모델의 성능 개선을 어렵게 만든다