**과목명: 시스템프로그래밍**

**1분반**

**<<Project #4>>**

**서강대학교 [컴퓨터공학과]**

**[학번] 20161571**

**[이름] 김도연**

목 차

1. **프로그램 목표**
2. **프로젝트 요구사항과 그 답변**

**1. 프로젝트 목표**

MNIST를 위한 예제코드가 주어져있을때, 그 코드를 적절히 개선하여 CIFAR-10의 데이터 셋에 대해서도 80%이상의 정확도(예제코드를 그대로 가져다 쓴다면 50~60%의 정확도를 얻을 수 있다)를 얻는 것이 목표이다.

**2. 프로젝트 요구사항과 그 답변**

-모델향상을 위해 사용한 방법들

1) optimizer함수의 변경

기존 SGD함수를 RMSprop함수로 변경해 주었다.

2) layer의 추가(가장 핵심)

convolution layer, pooling layer의 개수를 늘리고 구성도 다시 하였다.

자세한 설명은 뒤에 서술하였다.

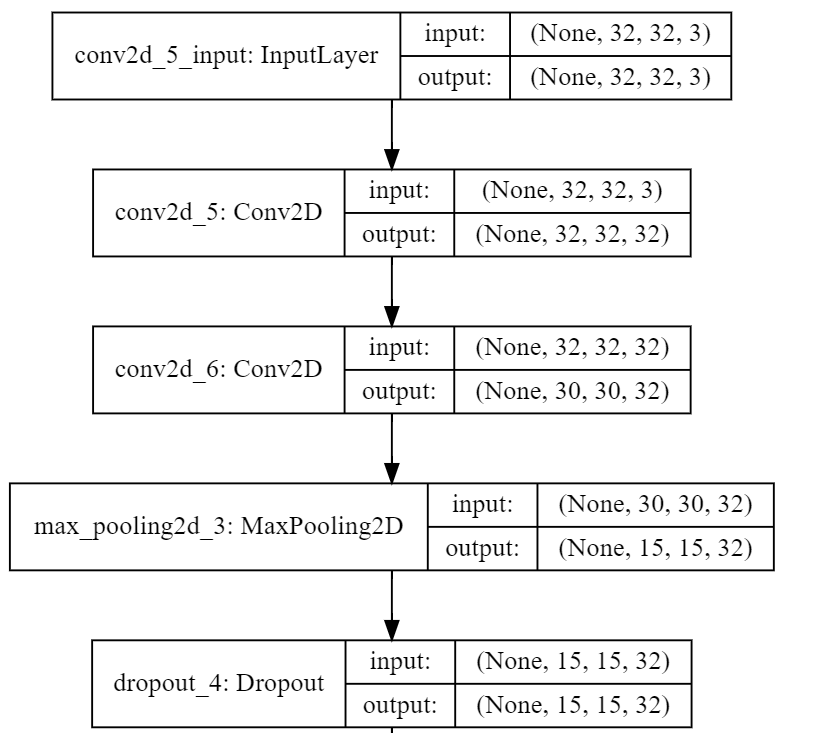
3) 기존 parameter들의 변경

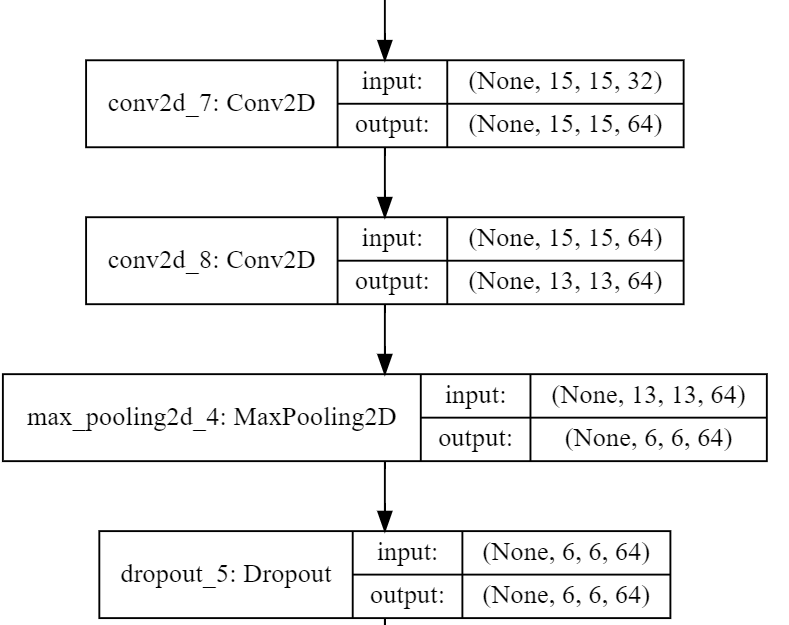
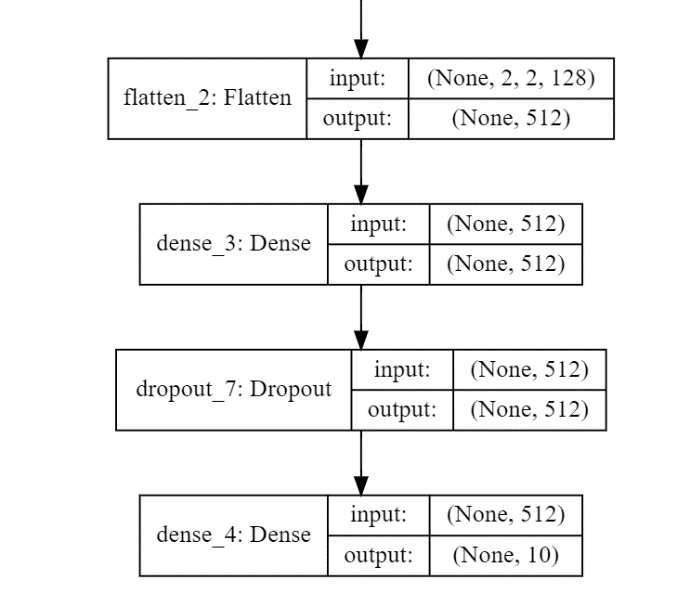
epochs는 10->50 으로 (문제조건)

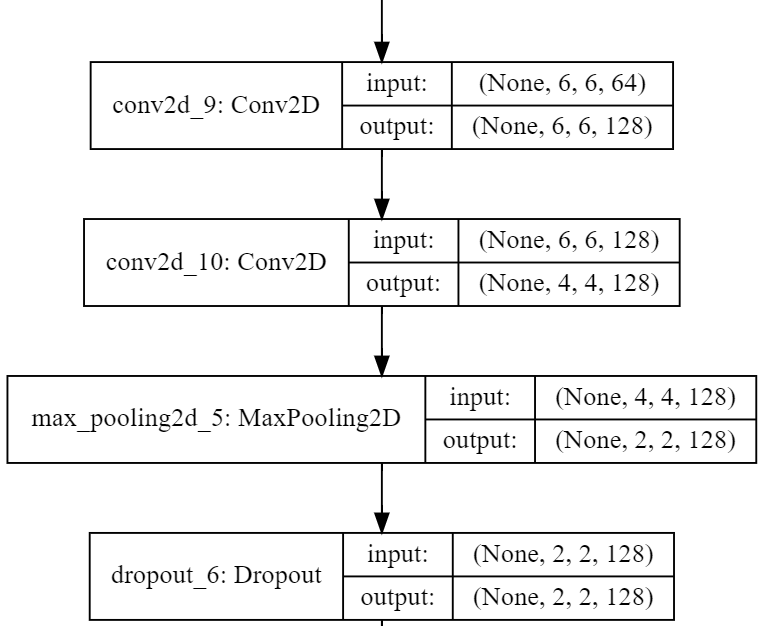
batch\_size는 128->32로

learning\_rate는 0.01->0.001로 변경하였다.

**-최종 Layer그림**







**-최종 Layer 구조에 대한 설명**

1)입력layer이다. 처음에 3x3의 크기를 가진 32개의 마스크를 적용해준다.

2)convolution layer로 3x3의 크기를가진 32개의 마스크를 적용해준다.

3)pooling layer로 maxpooling을 수행한다

4)은닉 layer에 있는 노드중 25%를 추가해준다(이때 추가할 노드는 임의로 결정)

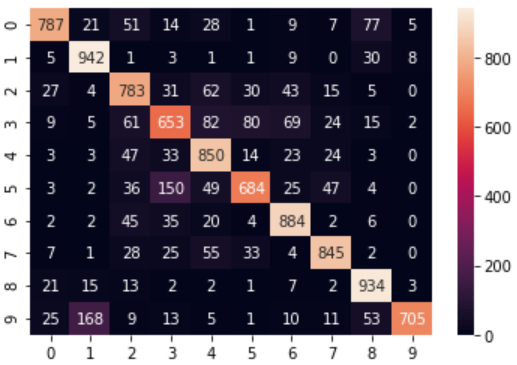
5)위 내용을 각각 64개, 128개의 마스크에 대해서 반복한다

6)Flatten()을 통해서 2차원 배열을 1차원 배열로 바꾸어준다.

7)512개의 노드를 가진 완전연결 layer이고 dropout를 다시 사용해준다.

8)마지막의 dense는 출력 layer생성을 위한 식이다.

**-confustion matrix 그림**



**-loss function을 설명**

loss function은 실제 데이터와 학습을 통해 얻은 데이터의 차이를 평가하기 위한 함수로서 쓰인다. 이번 프로젝트에서 선택한 loss function은 cross-Entropy형태의 함수중 하나인 categorical\_crossentropy인데, 이때의 함수식은 다음과 같다.

(y:실제값, y-hat: 예측값)

특징은 낮은 확률의 예측이 맞거나, 높은확률의 예측이 틀리는 경우 loss값이 더 크게 나온다.

**-전처리 과정에서 MNIST 이미지를 담은 배열의 rank를 4로 맞춰주는 이유**

배열의 구조가 {image의 개수, channel의 개수, hegith의 개수, width의 개수} 이기 때문에 당연히 rank를 4로 맞춰야 한다.

**-프로그램 실행에 대한 방법**

완성된 .ipynb 프로그램을 실행하기 위해서는 파일이 구글 드라이브나 github에 업로드 되있어야 한다. 그 상태에서 colab의 화면에서 파일->노트열기 로 들어가서 실행하고자하는 파일(여기서는 20161571.ipynb)을 찾아서 실행시킬 수 있다.