**과목명: 시스템프로그래밍**

**홀수**

**<<Project #4>>**

**서강대학교 컴퓨터공학과**

**20181593**

**계인혜**

목 차

1. **프로젝트 개요**
2. **코드 설명**
   1. Reshaping the data
   2. Creating the model
   3. Compiling and training the model(loss function)
   4. Confusion matrix
   5. Test accuracy

**3 정확도 향상 방법**

1. **프로젝트 개요**

MNIST 예제를 바탕으로 프로젝트 4에서는 Google Colab에서 Keras를 이용하여 이미지 분류를 하는 머신러닝 모델을 개발한다. CIFAR-10이라는 dataset을 이용하여 이미지 분류 성능을 측정하고 직접 모델의 파라미터들을 바꿔가며 테스트 셋에 대한 정확도 80% 이상을 얻는 것을 목표로 한다.

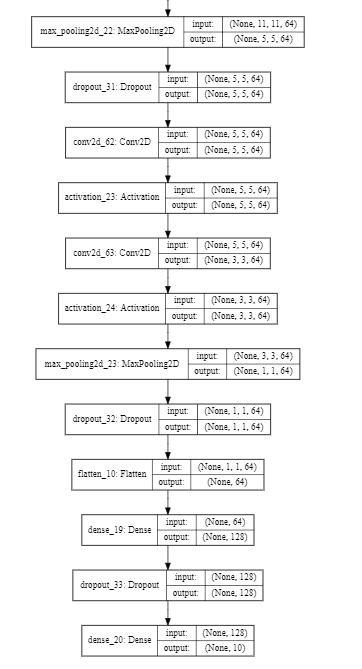
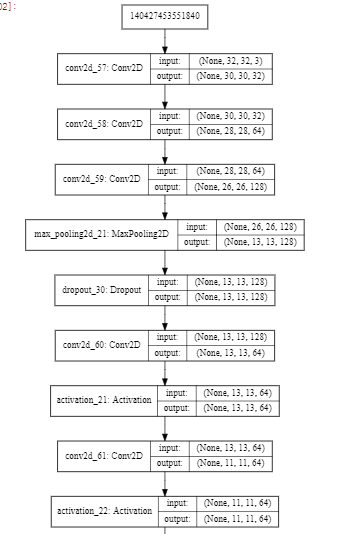
1. **코드 설명**
   1. Reshaping the data(rank = 4)

먼저, 텐서란 임의의 차원 개수를 가지는 행렬의 일반화된 모습으로 텐서의 축 개수를 rank라고 부른다.

Mnist 이미지를 담은 배열의 rank가 4인 이유는 이미지는 (sample, height, width, channels)크기의 4D 텐서를 갖기 때문이다. 이때 channel은 컬러 채널로 흑백 이미지의 경우 채널은 1이고, 컬러 이미지에 대한 채널은 3이다.

* 1. Creating the model

다음은 프로그램을 실행하였을 때 출력되는 최종 layer이다. 왼쪽에서 오른쪽으로 이어진다.



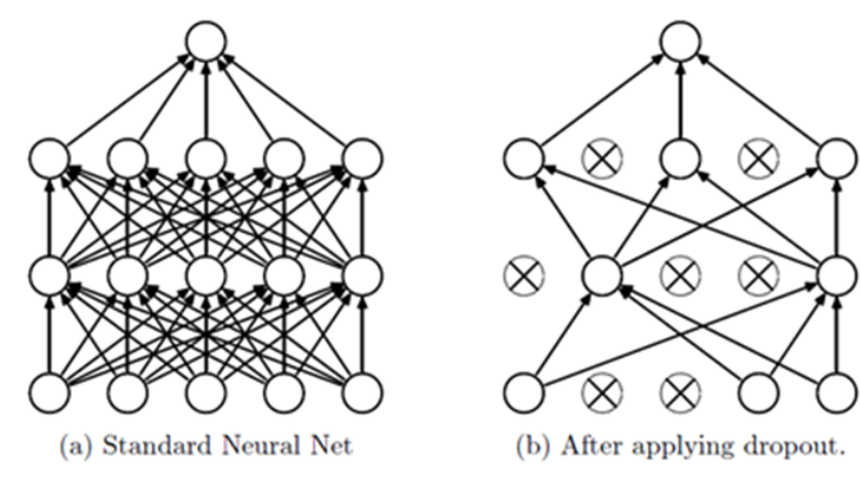
Layer 구조에 대한 설명을 다음 표를 통해 살펴보자.

|  |  |  |
| --- | --- | --- |
| Conv2D | 1st para | 컨볼루션 필터 수 |
| 2nd para | 컨볼루션 커널 (행,열) |
| padding | 경계 처리 방법 |
| Input shape | 샘플 수 제외한 입력 형태 |
| Max Pooling | 사소한 변화를 무시해준다. | |
| Dropout | 뉴런을 확률적으로 사용하지 않으면서 overfitting을 방지한다. | |
| Activation | 활성화 함수를 설정한다. | |
| Flatten | 1차원으로 바꿔준다. | |
| Dense | 입력 뉴런과 출력 뉴런을 연결한다. | |

Max pooling을 비유적으로 설명하자면 사람마다 눈, 코, 입 위치가 조금씩 다른데 이러한 차이가 사람이라고 인식하는 데 있어서는 큰 영향을 미치지 않게 하는 것이다.

|  |  |
| --- | --- |
| Padding |  |
| Input shape |  |
| Max pooling |  |
| Dropout |  |
| Activation |  |
| Flatten |  |
| Dense |  |

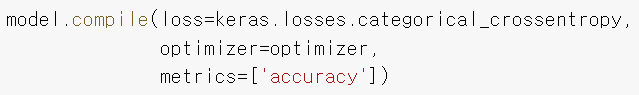
먼저 padding = ‘same’으로 정의하였다. 이는 출력 이미지 사이즈와 입력 이미지 사이즈가 동일한 것을 의미한다. Input shape는 모델의 첫 레이어에서만 (행, 열, 채널 수)로 정의한다. Max pooling의 parameter를 (2,2)로 지정함으로써 출력이 입력의 절반 크기로 줄어들게 된다. Dropout을 0.25로 설정함으로써 overfitting을 어느정도 방지하였다. 다음은 dropout을 간단하게 설명할 수 있는 그림이다.



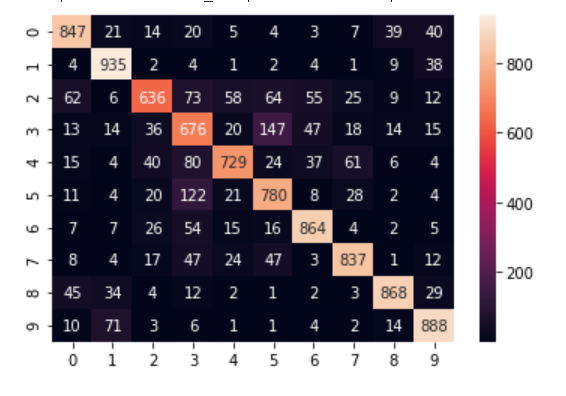
이어서 activation은 활성화 함수로 이번 프로젝트에서는 relu(rectifier), softmax 두 개만을 사용하였다. 컨볼루션 레이어와 맥스풀링 데이터는 주로 2차원 자료를 다루지만 이를 다시 전결합층에 전하기 위해서는 1차원 자료로의 변환이 필요하다. 이를 해결해주는 것이 바로 flatten layer이다. 마지막으로 dense layer는 입력 출력을 연결하는 전결합층이다.

* 1. Compiling and training the model(loss function)

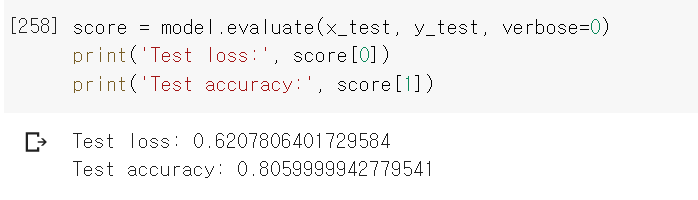
Loss function(손실 함수)란 출력값과 정답의 오차를 정의하는 함수로 보통 SSE와 Cross Entropy Error가 많이 사용된다. 이번 프로젝트에서는 cross entropy함수를 사용하였다. Cross entropy error는 자연로그를 예측값에 씌워 실제 값과 곱한 후 전체 값을 더해 음수로 변환한다. cross entropy error는 출력이 1일 때 0이 되며 x가 커질수록 0에 가까워지고 x가 작아질수록 값이 음의 방향으로 작아진다. 카테고리가 2개면 binary\_crossentropy를 사용하고, 3개 이상이면 categorical\_crossentropy를 사용한다. 이번 프로젝트에서는 카테고리가 3개 이상이기 때문에 categorical)crossentropy를 사용하였다.



* 1. Confusion matrix



* 1. Test accuracy



1. **정확도 향상 방법**

Learning rate를 0~1사이의 범위에서 여러 번 바꿔본 결과, 0.08일 때 가장 높은 정확도를 보였다. 또한 이 방법만으로는 정확도 80%를 보이기 어려워서 좀더 많은 Pooling layer층을 쌓았다. 마지막으로 base 코드에는 ‘float 32’로 설정되어 있던 것을 ‘float 64’로 바꾸면서 정확도가 아주 약간 더 올라갔다. 프로그램을 실행하는데 걸리는 시간이 너무 오래 걸려서 batch size를 더 크게 바꿔보기도 했지만, 정확도가 떨어져 다시 원래의 128로 설정하였다.