(1), (2) 구현 방법 및 부연 설명

12191656 이채연

(1) 모델을 Input layer -> Convolutional layer -> MaxPooling layer -> Fully-connected layer 층으로 구성하여 hidden layer 층을 1개로 하였다.

In [1]: 구현을 할 때는 딥러닝 프레임워크 Keras를 기반으로 Python에서 TensorFlow를 사용하였다.

In [2]: 학습 데이터 셋을 Train_set과 Test_set으로 나누고 Train_set과 Test_set 이미지를 28*28의 1개의 색깔을 가지도록 배열 차원을 reshape 하였으며 이것이 Input이 된다.

In [3]: Label이 10개의 값을 가지도록 one-hot encoding을 적용하였다.

In [4]: 학습률은 0.001, 에포크는 15 그리고 batch_size는 64로 설정하여 전체 데이터 셋 중에서 64개씩 총 15번 학습시켰다.

In [5]: Keras의 Sequential 모델은 차례로 layer을 쌓아 나가는 구조이다.

In [6]: Neural network의 hidden layer에서 input shape를 (28,28,1) 로 지정하였으며 Conv, MaxPooling2D 층을 거치게 된다.

In [7]: Flatten() 은 이미지를 1차원으로 바꿔주며 마지막 레이어의 결과 값은 softmax activation 함수를 거친다.

In [8]: 모델을 학습시키기 전에 학습 프로세스를 구성한다. 손실함수와 옵티마이저의 종류, 학습과 테스트 중 모델을 평가할 지표를 정의한다. 그 후, 모델을 학습시킨다.

In [9]: 학습시킨 모델의 10개의 임의의 Test_set 이미지에 대한 예측이 실제와 같다는 것을 확인하였다. 또한, 모델의 손실 값 및 정확도를 나타내어주었다.

(2) MNIST 데이터셋에 대해 linear-model을 설계하여 학습시킨 뒤, adversarial example을 만들어 모델이 이를 잘못 예측하는 것을 확인하였다.

In [1]: 구현을 할 때는 딥러닝 프레임워크 Keras를 기반으로 Python에서 TensorFlow를 사용하였다.

In [2]: 학습 데이터 셋을 Train_set과 Test_set으로 나누었다.

In [3]: MNIST의 숫자를 나타내는 단어 Label을 지정하였다.

In [4]: Train_set과 Test_set 이미지를 28*28의 1개의 색깔을 가지도록 배열 차원을 reshape 하였으며 이것이 Input이 된다. 그리고 Label이 10개의 값을 가지도록 one-hot encoding을 적용하였다.

In [5]: Softmax activation 함수를 사용하였고, input 차원을 784로 하고 unit을 10개로 나누어 linear 모델을 설계하였다.

In [6]: 에포크는 15 그리고 batch_size는 64로 설정하여 전체 데이터 셋 중에서 64개씩 총 15번 학습시켰다.

In [7]: 모델의 perturbation 적용 전의 정확도를 나타내었다.

In [8]: image와 label을 파라미터로 가지고 adversarial pattern을 만드는 함수를 선언하였다. dtype=float32를 사용하여 image를 Tensor로 변환하였다. 그 뒤, GradientTape 객체를 사용하여 특정 Tensor에 대한 그레이디언트를 자동으로 계산하도록 하였다. 그리고 계산된 그레디언트의 부호를 가져와 adversarial pattern을 만들어 반환하였다.

In [9]: adversarial image를 만들기 위해 x_train set의 이미지와 Label을 저장하였다.

In [10]: 이를 가지고 adversarial_pattern 함수를 호출하여 perturbation를 만들었다.

In [11]: 입실론의 크기를 0.25로 하여 Image에 perturbation을 추가하였다.

In [12]: 만들어진 adversarial example을 그림으로 나타내었다.

In [13]: 모델은 만들어진 adversarial example에 대해 label이 'four'인 이미지를 'eight'로 잘못 예측하는 것을 확인하였다.