과제 코드 jupyter notebook file : Assignment\_2.ipynb

1. 사용 데이터: MNIST

2. 사용 라이브러리: Tensorflow2

3. hold-out validation

- train data 개수: 50000

- validation data 개수: 10000

- test data 개수: 10000

4. accuracy 90%이상 달성 유무: ㅇ

5. data transformation

- 빠른 연산을 위한 channel-wise normalization 실시

- label one-hot encoding 실시

6. model architecture

Table

Description automatically generated

- Alexnet을 기반으로 설계를 진행하였습니다. Alexnet은 ImageNet 학습을 위해 image 224x224x3의 input으로 받고 있으며 classification을 해야하는 label의 수가 1000개이기 때문에 mnist data에 맞춰 dimension와 layer의 수를 낮춰주었습니다. 가장 일반적으로 사용되는 conv2d + activation function (relu) + max pooling 형태를 띄고 있으며 최종 dense layer에 들어가기 전 dropout을 통해 regularization을 적용했습니다.

# 7. 비교 실험

1) learning rate에 따른 결과

- optimizer: Adam

- learning rate: 0.01

- test accuracy: 97.5%

Chart, line chart

Description automatically generated

- optimizer: Adam

- learning rate: 0.001

- test accuracy: 99.3%

Chart, line chart

Description automatically generated

- learning rate의 크기에 따라 모델이 수렴하는 속도를 볼 수 있었습니다. Learning rate가 큰 경우, 학습 초반에 train loss가 급격하게 떨어지는 모습을 볼 수 있었습니다. 하지만 train loss가 일정 값 이하로 떨어지지 않는 결과가 나타났습니다. 이를 통해, local minima에 빠진 듯한 모습을 확인할 수 있었습니다. 그래서 learning rate를 조금 더 낮게 설정하여 실험을 진행하였고 2%나 높은 성능을 보여주었습니다. 위의 결과를 통해 적절한 learning rate를 정하는게 중요한 것임을 확인할 수 있었습니다.

2) optimizer 종류에 따른 결과

- optimizer: Adam

- learning rate: 0.001

- test accuracy: 99.3%

Chart, line chart

Description automatically generated

- optimizer: SGD

- learning rate: 0.001

- momentum: 0.9

- test accuracy: 98.5%

Chart, line chart

Description automatically generated

- optimizer의 종류에 따라 결과가 달라지는 것을 확인하기 위해 SGD optimizer를 사용했습니다. Learning rate는 동일하게 적용하였고 SGD에서 momentum값으로 가장 많이 사용되는 0.9로 설정했습니다. 학습을 통해 나오는 loss값의 추이를 살펴보았을 때 오히려 SGD가 안정적으로 학습이 진행되는 결과를 관찰할 수 있었습니다. 다만, 최종 test accuracy에서는 Adam optimizer의 성능이 더 좋은 것을 확인했습니다. 이를 통해, 데이터와 모델의 특징 마다 적절한 hyper-parameter와 구현 옵션을 설정하는 것이 중요하다는 것을 알 수 있었습니다.

# 8. 결과 시각화

Graphical user interface, application

Description automatically generated

- 모델이 예측한 결과에 맞는 것을 눈으로 확인할 수 있었습니다.