DNN 입문 예제

: MNIST를 이용한 분류문제 해결

49p

0~9까지 10개의 클래스

영상 하나가 28\*28이다.

영상 데이터는 글자가 있는 부분은 0~1사이의 값으로, 나머지는 다 0으로 채워져 있다.

# 사전 지식 쌓기

1. Batch size

답을 맞춰본다 = 가중치를 갱신.

Batch size만큼의 학습을 수행하고 가중치를 갱신한다. 이를 전체 데이터에 대해서 반복

메모리에 따라 가능한 만큼 batch사이즈를 설정하는 것이 최선.

1. Epochs
2. Optimizer

Global minimum을 찾는 것.

1. Gradient Descent (GD)

학습데이터 전체를 사용하는 최적화

1. Stochastic Gradient Descent (SGD)

학습데이터 일부(mini-batch)를 사용하는 최적화

배치 사이즈별로 최적화한다.

그렇다고 가장 성능이 좋은 것은 아니다.

48p 대표적인 최적화 방법론들

SGD로부터 탐색방향과 스탭 사이즈의 장점을 고려하여 나온 것이 “Adam”이다.

54p pytorch 메뉴얼

우리가 위에서 배운 라이브러리.

torch: 하나의 언어. 과학,수학쪽에서 머신러닝을 사용하기 위한 언어

pytorch: 이를 python으로.

55p Torch vision 패키지

torch비전 라이브러리에 대한 설명

57p 다양한 실험의 방향성을 통해 성능 비교

1, 2)는 NN 1layer=Perceptron이다.

1. SGD optimizer를 통해 숫자분류

성능이 가장 낮을 것이다.

1. Adam optimizer

SGD보다 Adam 최적화가 분류문제를 잘 해결할 수 있다는 결론을 도출

1. Layer 3개(NN)을 Adam optimizer를 이용해 해결

Layer를 많이 쌓아서 성능이 오름

1. 3번문제에서 초기화 방법을 Xavier로 변경

성능이 향상되었다.

1. Layer를 5개로 깊게 쌓았다. DNN
2. Dropout 기법을 적용

해당 문제에서는 성능이 오르진 않았다.

하지만 다른 데이터셋에 대해 깊게 layer를 쌓으면 5번의 상태는 학습이 안될 것이다.

이때 dropout을 하면 성능이 오르거나, 학습이 되는 안정적으로 수렴할 것이다.