<Ch.07 MLP와 케라스 라이브러리>

2 행렬로 미니 배치 구현하기 → 시험에는 안 나올듯!

- 한 번에 여러 개의 샘플을 처리해도 각각 계산되어서 행렬에 쌓임

- 미니배치의 구현

3 학습률

- 학습률(learning rate): 한 번에 가중치를 얼마나 변경할 것인가를 나타낸 것

- 모멘텀(momentum): 운동량, 학습 속도를 가속화 ← 시험 X
  - 모멘텀 도입시 가중치 변경 수식: $W_{t+1} = W_t - \eta \frac{\partial E}{\partial W} + momentum * W_t$

- 적절한 학습률 설정하기 : 현재 gradient 값, Weight의 크기, 학습의 정도 등을 고려하며 적응형 학습률을 채택
  - Adagrad : SGD 방법을 개량한 최적화 방법. 학습률 감쇠(learning rate decay) 채택
    - 학습의 횟수가 많아지면 감쇠가 커져 학습이 안 됨
  - RMSprop : Adagrad의 수정판. gradient 누적이 아닌, 지수가중이동평균 사용
  - Adam (Adaptive Moment Estimation) : RMSprop + 모멘텀

4 케라스 (keras) 시작하기

- 텐서플로우와 케라스
  - TensorFlow: 저수준 Deep Learning 라이브러리
  - Keras: 고수준 Deep Learning 라이브러리

- Keras로 신경망을 작성하는 절차
  - Keras의 핵심 데이터 구조는 model
    - Sequential 선형 model: 가장 간단한 모델 유형

5 케라스를 사용하는 3가지 방법

- Sequential 모델을 만들고 모델에 필요한 레이어를 추가하는 방법
  - add()를 사용해 모델에 점진적으로 레이어를 추가

- 함수형 API를 사용하는 방법
  - 레이어와 레이어를 변수로 연결

- Model 클래스 상속하기
  - Model 클래스를 상속받아 나름대로의 클래스를 정의

6 케라스를 이용한 MNIST 숫자 인식

7 케라스의 입력 데이터

- 입력 데이터 유형 : numpy 배열, Tensorflow Dataset 객체, 파이썬 제너레이터

- 텐서(tensor) : 신경망의 입력, 다차원 넘파이 배열임.

- 텐서의 속성
  - 차원 : tensor에 존재하는 축의 개수
    - ndim 속성으로 알 수 있음
  - 형상 (shape) : 텐서의 각 축
  - 데이터 타입 (data type) : 텐서 요소의 자료형
    - dtype 속성으로 알 수 있음.

- 훈련 데이터의 형상 : data를 쪼개야 하는 경우
  - batch = x_train[:256]

- 많이 사용되는 훈련 데이터의 형상 : 벡터 데이터, 이미지 데이터, 시계열 데이터, 동영상 데이터

8 케라스의 클래스들

- keras의 요소 : 모델, 레이어, 입력 데이터, 손실 함수, 옵티마이저 (학습을 수행하는 최적화 알고리즘)

- Sequantial 모델 : Feed-Forward 신경망을 구현하는 가장 기초적인 모델
  - Compile, fit 등

- 레이어
  - Iput, Dense 등

- 손실함수 : MSE, BinaryCrossentropy, CategoricalCrossentropy, SparseCategoricalCrossentropy (one-Hot encoding 안했을 때)

- 측정항목 : Accuracy 등

- 옵티마이저 : SGD, Adam, Adagrad 등

- 활성화함수 : sigmoid, relu, softmax, tanh, selu, softplus 등

9 하이퍼 매개변수

- 하이퍼 매개변수 : 학습률, 모멘텀의 가중치, 은닉층의 개수, 유닛의 개수, 미니 배치의 크기 등

- 하이퍼 매개변수를 찾는데 사용되는 방법 : 기본값 사용, 수동 검색, 그리드 검색, 랜덤 검색
  - 그리드 검색 : 각 하이퍼 매개변수에 대하여 몇 개의 값을 지정해 가장 좋은 조합을 찾는 알고리즘.