



6주차

Chapter 7. 딥러닝을 시작합니다

▼ 7-1 인공 신경망

PCA : 원본 데이터셋의 벡터를 찾는 것 - 분산이 가장 큰 벡터 (값들을 대표)

→ 주성분 = 원본 공간 차원

주성분을 찾을 때는 항상 앞선 선분의 수직 방향

각 주성분의 차원은 원본 데이터셋

transform() : 차원 수를 변환하는 메소드

inverse_transform() : 차원 재구성 메소드

패션 MNIST 데이터셋 - 로지스틱 회귀 적용

- 이미지 데이터 - 직관
- train_target의 클래스 분포
- 각 정숫값에 들어간 클래스 개수

모델 제작

경사하강법을 사용한 로지스틱 회귀 모델

1)

부츠 - 양성

나머지 - 음성

모델 훈련

식 도출

2)

티셔츠 - 양성

나머지 - 음성

모델 훈련

식 도출

...

→ 10개의 선형 방정식의 계산값을 softmax 함수를 이용해서 확률로 변환

→ 이진 분류의 경우 하나의 함수 훈련 뒤 sigmoid 함수를 이용해 확률로 변환

OvR(OvA) : 다중 분류를 여러 개의 이진 분류로 나눠 훈련

패션 분류 - 입력 픽셀이 10개의 선형 방정식을 구성할 때 **다른 가중치와 곱해진다**는 것이 중요 (→ 인공 신경망에 동일 적용)

인공 신경망

- 경사하강법을 사용한 선형 회귀 모델과 같음
- 입력층은 알고리즘 상에서 데이터 자체, 특별한 의미 없음 (편의를 위한 구현 가능)
- 절편이 표기되지 않는 경우가 많지만 유닛에 항상 더해짐
- 생물학적 뉴런에서 영감을 받아 만들어짐

딥러닝 : 출력층 외 다른 층을 넣어둔 모델

가장 기본적인 인공 신경망 = 로지스틱, 선형 회귀

텐서플로 = 케라스

파이토치 or 텐서플로우 배우면 됨

케라스 모델

- 입력층 - 유닛(뉴런) 개수 (입력 데이터)
- 밀집층 (완전 연결층) - 활성화 함수로 확률 계산
- 출력층 - 클래스 개수

모델 설정

- 손실 함수 정의
 - 이진 분류 - loss : binary crossentropy
 - 다중 분류 - loss : categorical crossentropy
- 훈련 도중 손실값, 정확도 지정 (matrix method, 컴파일 메소드)

loss 함수

- 타겟값 = 정수

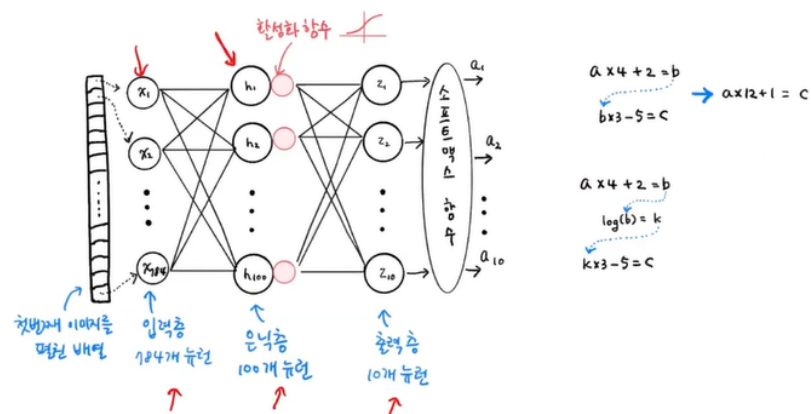
- 원핫 인코딩 여부

(원핫 인코딩 - 인덱스 1을 제외하고 나머지 모두 0)

에폭 : 반복 횟수

▼ 7-2 심층 신경망

2개의 층 추가



출력층 > 은닉층 → 정보 손실

이진 분류 : 시그모이드 함수, relu 함수, tanh 함수 등 사용 (활성화 함수)

은닉층과 출력층

`keras.sequential([dense1, dense2])`

각각 객체를 만들어 sequential 전달

summary() 메서드

고정된 배치 사이즈로 모델 형성

dense 객체를 여러 개 만들어서 sequential 클래스에 차례로 추가

층 추가 방법 : add 메소드를 이용해 model에 dense를 추가하여 유연하게 대처 가능

ReLU 함수

시그모이드 함수의 단점인 완만한 구간에서 깊은 층을 못 쌓는 것을 보완

Flatten 층

배열을 펼치는 과정 등 전처리에 도움, 실제 학습은 X

모델 안에 전처리 코드를 넣는 것이 효율적

옵티마이저

확률적 경사 하강법 사용

- 기본 경사 하강법 옵티마이저
- 적응적 학습률 옵티마이저

sgd 클래스를 이용해 모멘텀 방식 적용 가능

(어떻게 가중치를 업데이트 하는지는 자료 참고)

랜덤 포레스트 - 앙상블

신경망 - 여러 층을 쌓아 순차적 학습

스태킹 모델 - 앙상블을 쌓아 훈련

▼ 7-3 신경망 모델 훈련

손실 곡선

손실값을 토크마다 저장해 반환

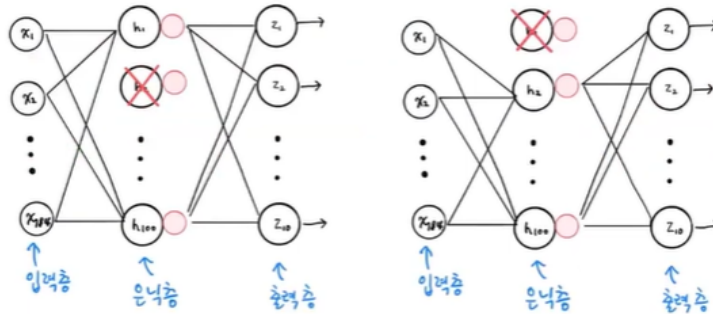
에포크를 늘리면 복잡한 모델이 만들어짐

훈련 세트 & 검증 세트의 적절한 절충점을 찾기 위해 손실값 비교하여 파악

검증 손실

에포크가 증가함에 따라 훈련 세트의 손실값이 줄어듦

과대적합을 규제하기 위해 드롭아웃 사용



드롭아웃 : 일부 뉴런을 끄으로써 과대적합을 막음

모델 저장과 복원

최적의 모델 클래스 객체를 디스크에 저장하는 작업이 라이브러리에 제공됨

save weight: 가중치 (모델 파라미터)만 저장

load weight : 모델 객체를 만든 다음 파라미터를 바꿔 학습

save : 모델 구조와 가중치 함께 저장

load model : 모델 파일 지정 뒤 로드

콜백

지정한 작업 수행 메소드

ModelCheckpoint : fir 훈련 중 가장 낮은 손실값 가중치 저장

callbacks 매개변수를 통해 checkpoint_cb 전달

조기 종료

keras 콜백으로 제공

early stopping class 매개변수

- patience : 검증 세트 손실이 2번의 에포크로 증가하면 멈춤
- restore_best_weight : 손실이 낮았던 가중치로 되돌림