**17강 인공 신경망 🡪 간단한 인공 신경망 모델 만들기**

* 텍스트, 폰트, 스크린샷, 스케치이(가) 표시된 사진

  자동 생성된 설명딥러닝에서 대표적인 데이터셋 : MNIST
  + 0~9까지의 숫자를 손으로 쓴 이미지 데이터셋, 28\*28 픽셀, 총 6만개의 샘플
  + 더 발전된 형태 = 패션 mnist
  + 0~9까지의 숫자 (10개의 클래스), 28\*28, 6만개의 샘플
  + 타깃 데이터: 이미지가 분류 되는 옷의 종류에 따라 0~9까지 값 지정
  + Train\_input : 샘플 6만개의 배열, test\_input: 샘플 만개의 배열

\*tensorflow 라이브러리에 저장되있음 🡪 load\_data를 사용하면 훈련세트, 테스트세트와 타겟을 나누어진 채로 가져옴

* 텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

  자동 생성된 설명데이터셋의 간단한 통계 살펴보기 \*이미지파일은 몇 개를 직접 추출해서 보는 게 좋음

cmap = ‘gray\_r’을 통해 색깔 반전 🡪 높은 숫자의 값을 검게

타겟값이 0부터 9까지 존재한다는 것을 알 수 있음

Return\_counts=True에는 각 train\_target의 클래스 레이블에 따라 몇 개의 샘플이 있는 지 확인 가능 🡪 각각 6000개씩 존재 \*10개의 레이블이 6000개씩 = 6만개

\*일반적으로 타깃데이터에 레이블 마다 샘플 개수가 다를 수가 있으므로 해당 코드로 확인해 보는 과정을 거치는 것이 좋음

* 텍스트, 스크린샷, 폰트이(가) 표시된 사진

  자동 생성된 설명데이터셋에 로지스틱 회귀 사용하여 분류하기

이미지 데이터는 모두 255개의 픽셀 데이터를 기준으로 하기 때문에 표준화 과정을 거치지 않고 255로 나누어 0과 1 사이로 변환하는 정도로 갠찮음

28\*28 : 3차원 이미지 데이터를 2차원 배열로 변환

SGDClassifier: 경사하강법을 사용한 로지스특 회귀모델에

교차검증도 적용하여 검증폴드들의 평균 점수 출력

\*10개의 클래스를 분류하는 다중분류문제임 🡪 loss = log으로 지정하면 2개의 클래스 일 때에는 로지스틱 손실함수(z값 도출 후 시그모이드함수 적용)를 사용해 훈련, 다중분류일 경우엔 10개의 이진분류를 수행

Ex) 부츠-양성, 9개 음성으로 모델 하나 훈련—다음엔---티셔츠-양성, 9개 음성으로 모델 훈련 🡪 10개의 함수식이 도출되면 선형식을 통해 나오는10개의 z값을 소프트맥스 함수를 사용해 확률로 변환한다.

= OVR(OVA) = One Verses Rest(All)

* 텍스트, 폰트, 도표, 라인이(가) 표시된 사진

  자동 생성된 설명로지스틱 회귀모델 시각화 설명

픽셀 784개가 1차원 배열로 나열 되어 있는 데이터에서 각 레이블의 z값을 도출하기 위해 픽셀 하나하나에 가중치를 곱하고 마지막에 절편을 더하고 이러한 과정을 모든 레이블에 반복하면 각 레이블에 대한 z값이 도출된다. \*각 레이블의 각각의 픽셀에 대해 가중치가 서로 다르게 적용됨. Ex) 티셔츠 픽셀1 \*w1, 바지 픽셀1\*w1’

* 텍스트, 도표, 친필, 폰트이(가) 표시된 사진

  자동 생성된 설명로지스틱회귀모델을 인공신경망 모델과 동일하게 볼 수 있음

가장 간단한 인공신경망(=딥러닝)은 출력층이 하나 있는(z값) 모델을 일컬으며 그 과정은 위의 로지스틱 회귀모델과 완전히 동일 하다. \*심화 인공신경망(일반적 딥러닝)에선 출력층이 늘어남

각 뉴런(유닛)에 따라 포함하는 요소 각각에 가중치가 다르게 적용된다는 것이 중요함

\*절편도 항상 포함되지만 편의상 절편을 표기하지 않는 경우가 많음

\*사실상 실제 인간의 신경망과는 크게 유사성X 🡪 뉴런이 아니라 유닛이라고 불리는 이유

- 입력데이터가 놓여져 있는 층: 입력층

- 출력층에 있는 값을 뉴런(유닛)이라고 부르기도 함.

* 텍스트, 폰트, 스크린샷, 도표이(가) 표시된 사진

  자동 생성된 설명텐서플로와 케라스

Pytorch의 인기가 증가하는 추세(2021년 기준)

이 두 라이브러리를 공부하는 것이 좋음

* 텍스트, 도표, 친필, 폰트이(가) 표시된 사진

  자동 생성된 설명케라스 모델 만들기

딥러닝에서는 교차검증을 잘 사용하지 않음:

* 데이터가 많기 때문에 검증안정 + 게산비용이 높기 때문에 부가과정을 줄여야 함

Train\_test\_slit()에서 test\_size = 0.2 는 테스트세트를 전체 데이터의 20퍼센트로 설정

Kera.layers.Dense(10 ~) : 10개 뉴런으로 구성된 출력층 지정(출력층의 뉴런의 개수 지정) \*항상 클래스의 개수와 동일해야 함

Activation = : 매개변수로 다중분류이므로 소프트맥스함수를 사용해 각 레이블의 확률 출력 \*이중분류는 ‘sigmoid’ 기입

Input\_shape = : 매개변수로 입력층의 경우 객체를 따로 지정하여 추가하는 경우가 있으므로 첫번째 모델에 추가되는 층(입력층)의 크기를 지정해주는 것이 관례적으로 행하는 과정임

ex)데이터의 샘플 크기(픽셀 개수)에 따라 튜플로 감싼 784개의 원소로 이루어진 1차원 배열로 지정

케라스의 가장 기본이 되는 층 = Dense층 : 밀집층, 완전연결층(fully connected layer)

케라스의 층들은 layers 안에 모두 포함됨

Model = keras.Sequential(dense) : 인공신경망을 모델에 객체로 지정하는 과정으로 dense라는 출력층을 매개변수로 입력 🡪 얘를 이요해서 모델 훈련

* 모델 설정:

텍스트, 폰트, 스크린샷, 도표이(가) 표시된 사진

자동 생성된 설명우선 손실값을 기록하는 과정이 필요 🡪 model.compile(loss = ) 사용 : \*이중분류일경우 로지스틱손실함수를 사용하고 다중분류일 경우엔 크로스엔트로피손실함수를 사용하는데 매개변수로 들어가는 형식은 각각 ‘binary\_crossentropy’와 ‘categorical\_crossentropy’의 형식으로 들어간다.

크로스엔트로피손실함수의 과정: 10개의 확률값(소프트맥스를 거친)에 로그를 취하고 타깃값과 곱해짐 🡪 한 레이블의 값을 도출할 때 곱해지는 타깃값 중 해당 레이블의 타깃값만 1로 살고 나머지는 0으로 구성되어야함 = 원핫인코딩. EX) 티셔츠와 바지일경우의 원핫인코딩이 그림으로 보여짐 BUT 타깃값을 그냥 정수값으로 사용하고 싶으면 sparse를 붙여 크로스엔트로피 매개변수를 사용하면됨

\*추가적으로 정확도를 측정할 수도 있음 🡪 metrics 매개변수 사용

* 모델훈련

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명에포크 🡪 확률적경사하강법에서 그 과정을 반복하는 횟수 🡺 횟수에 따라 손실값이 줄어들고 정확도가 높아진 다는 것을 알 수 있음 \*최적의 에포크에 따라 결과가 다를 수도 있음

Mode.evaluate(val\_scaled, val\_target): 훈련이 끝나면 검증세트로 최종검증 = 85퍼센트의 정확도를 보임

* 사이킷런과 케라스 API 비교

텍스트, 폰트, 친필, 라인이(가) 표시된 사진

자동 생성된 설명사이킷런은 클래스를 만들 때 가능한 많은 매개변수 사용

케라스는 층을 만들어 따로 추가하고 compile 매써드를 통해 따로 손실함수들과 같은 지표들을 추가하고 훈련과 평가는 사이킷런과 동일한 과정을 거침

케라스 모델은 층을 만드는 부분, 모델을 만드는 부분, 출력지표를 설정 하고 훈련 및 평가하는 세 부분으로 분화되어 있음

\*API:  Application Programming Interface. In the context of APIs, the word Application refers to any software with a distinct function. Interface can be thought of as a contract of service between two applications. This contract defines how the two communicate with each other using requests and responses.

텍스트, 도표, 친필, 라인이(가) 표시된 사진

자동 생성된 설명**18강 심층 신경망 🡪 인공 신경망에 층을 추가하여 심층 신경망 만들어보기**

입력층은 층을 셀 때 포함되지 않음 🡪 현 상황에선 은닉충과 출력층을 포함하여 2개의 층이라고 일컬음

은닉층의 개수는 정해진 것이 아님 = 100이 아니여도 됨

\*출력층 보다는 큰 은닉층을 사용해야 데이터손실이 일어나지 않음

이진분류: 시그모이드 / 다중분류: 소프트맥스

은닉충이 선형식이라면 굳이 존재할 필요가 없음 🡪 비선형식을 포함한 활성화함수를 만들 때에만 은닉층을 사용 \*시그모이드 함수, relu함수, tanh 함수를 은닉층에 많이 사용

* 텍스트, 도표, 폰트, 라인이(가) 표시된 사진

  자동 생성된 설명심층 신경망

은닉층 = 입력층과 출력층에서 접근이 안되고 가운데 숨어 있어 은닉충이라고 불림

은닉충의 객체 = dense1, 출력층의 객체 dense2

Desnse1과 dense2를 keras.Sequential에 넣으면 모델이 만들어짐

🡪 은닉충이 없는 모델과 비교했을 때 모델을 구성하는 방식만 달라짐 \*활성함수를 무조건 지정해줘야 유의미함

* 텍스트, 도표, 친필, 라인이(가) 표시된 사진

  자동 생성된 설명Summary() method

두개의 층에 자동으로 dense와 desne\_1과 같이 부여

앞의 차원이 none이라고 되어 있는 이유는 fit을 할 때 batchsize= 매개변수를 사용하면 임의경사하강법으로 32개씩 잘라서 모델을 훈련함 🡪 디폴트는 32개이며 바꿀 수 있음

Param 매개변수는 각 층에 있는 가중치와 절편,모델 파라미터의 개수를 나타냄 🡪 모델이 잘 수행하는 가 계산하기 위해서는 모델 파라미터의 개수를 직접 계산하여 출력값과 비교.은닉충의 파라미터를 살펴보면 은닉층에는 입력층이 784개의 뉴런 은닉충에는 100개로 완전연결(=모든 입력과 모든 뉴런이 연결되어 있음) 🡪 파라미터의 개수는 784\*100 + 100(각 선형식마다 절편이 존재함 = 100개)

절편이 따로 표기하지 않는다는 건 그림에서의 의미

출력층에서도 마찬가지로 100(은닉층의 출력개수)\*10(출력층) + 10(절편의 개수)

* 텍스트, 폰트, 스크린샷, 라인이(가) 표시된 사진

  자동 생성된 설명층을 추가하는 다른 방법:

객체화 하자마자 바로 sequential클래스에 dense 관련코드 전달 🡪 Sequential안의 리스ㅡ 형식으로)

But 실제로 많이 사용하는 방법은 객체를 먼저 만들고 add매써드를 사용해 dense를 활용한 층을 하나하나 추가함 🡪 if구문 등 조금 더 동적으로 모델 구성 가능

* 모델 훈련

이전과 동일하게 모델 compile하고 훈련

* 렐루 함수와 Flatten 층

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명시그모이드 함수의 단점: 출력값이 너무 커지거나 작아질 경우에 시그모이드 함수의 변화정도가 너무 작아짐. 🡪 완만하게 이어지는 구간은 신경망모델이 대응하기가 어려움

* 렐루 함수의 개발

렐루함수: z값이 0보다 크면 z출력, 0보다 작으면 0으로 잘라버림 \*은닉층에 사용하려면 activation = ‘relu’ 사용

Flatten층: 모델을 편리하게 운영하기 위해 케라스에만 존재하는 유틸리티층으로 학습되는 파라미터는 없지만 이미지 배열을 일차원 배열로 펄쳐주는 작업을 해줌

Ex) 28\*28의 이미지 배열을 784의 1차원 배열로 flatten해줌

* 옵티마이저: 하이퍼파라미터 중에서 하나로 효과적인 학습방식 선택 (like경사하강법 )

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명Sgd: 확률적 경사 하강법을 의미하지만 실제로는 임의경사하강법을 사용(32개의 샘플씩) \*대신 배치의 크기를 따로 설정하지 못함

기본 경사 하강법 옵티마이저: 모멘텀 / 네스테로프 모멘텀

적응적 학습 옵티마이저(최적값(수렴점)에 가까워질수록 학습율을 줄이는(조금씩조금씩 이동) 모델): RMS /아담 /Adagrad

사용방법: Sgd 클래스로 만들어서 기입(기본 매개변수가 아닌 다른 매개변수를 사용하고 싶을 때)하거나 문자열로 sgd라고 사용해도 됨

모멘텀: SGD클래스에 모멘텀 매개변수와 nestrov 매개변수의 값을 지정해줘야 함

\*모맨텀과 네스테로크 모멘텀은 조금 복잡하므로 현 책에는 나오지 않음

아담: 옵티마이저 매개변수로 문자열 기입

* 머신러닝과 딥러닝 구분

랜덤포레스트는 트리들이 완전히 독립적으로 훈련하여 나중에 합산됨:

인공신경망 같은 경우는 은닉층에서 한번에 훈련되며 층을 추가하여 단계별 학습을 가능하게 함

그레이디언트 부스팅이나 스태킹(앙상블 모델에 여러 단계를 추가하여 학습하는 모델 – 여러 단계를 추가하기 어려움) 훈련도 모델들이 순찬적으로 훈련하게 됨

인공신경망은 여러 층(깊이를 깊게=deep)을 학습 가능 + 각층이 비정형 데이터의 특징을 뽑아 다음층으로 전달하는 방식을 사용(=특성공학을 수동적으로 뽑았던 역할을 어느정도 자동으로 진행해줌) = 표현학습 모델