**스키마**

* 이전시간 활용 개념 복습
  + 크로스엔트로피손실함수의 과정:
    - 소프트 맥스를 거쳐 도출한 클래스별 확률값에 로그를 취하고 타깃값들과 곱함
      * 타깃값들 중 확률값의 클래스에 해당하는 타깃값들은 1로, 그 외의 타깃값들은 0으로 구성되어야 함 = 원핫인코딩
  + 에포크(epochs) : 확률적 경사하강법에서 전체 데이터를 프로세스하는 횟수
* 딥러닝관련 개념
  + 딥러닝에서는 교차검증을 사용하지 않고 훈련세트와 테스트 세트로만 분리
    - 데이터의 양이 충분한 경우가 많으며 계산양이 많아 자원을 절약해야 함
  + **패션 MNIST**: 가장 대표적인 딥러닝 연습용 데이터셋
    - 0~9까지의 숫자로 지정된 10개의 의류 클래스와 28\*28의 픽셀값으로 구성된 6만개의 샘플을 포함 (60000, 28, 28)
    - 사용방법:
      * 불러와서 훈련세트와 테스트세트로 분리
        + From tensorflow import keras

(train\_input, train\_target), (test\_input, test\_target) = \keras.datasets.fashion.mnist.load\_data()

* + - * 입력과 타깃 샘플 살펴보기
        + Fig, axs = plt.subplot()와 for루프를 활용하여 10개의 이미지 출력해보기
        + Train\_target 도 10개정도 값 출력해 보기 \*이미지데이터에서 필수
        + 클래스별로 샘플이 몇 개씩 들어있는 지 확인해보기 \*필수

Np.unique(train\_target, return\_counts=True))

* + **인공신경망:** 딥러닝의 기반이 되는 알고리즘으로 로지스틱회귀모델과 완전히 동일한 체계를 지님
    - 로지스틱회귀를 사용한 분류 과정 및 원리:
      * 훈련 데이터 스케일링:
        + 이미지 파일 같은 경우엔 굳이 스케일링을 할 필요없이 편의를 위해 255로 나누어 0과 1 사이의 값으로 변화하는 정도로
      * 훈련 데이터에 reshape()을 사용하여 1차원배열의 묶음으로
      * SGDClassifier: 확률적경사하강법으로 진행되는 로지스틱손실함수를 이용한 로지스트 회귀 모델 적용
        + 과정과 관련한 개념:

로지스틱 회귀 적용

픽셀 하나하나에 가중치를 곱하고 마지막에 절편을 더하는 과정을 모든 클래스에 대해 반복하여 z값 도출

\*각 클래스의 각각의 픽셀에 대해 가중치가 서로 다르게 적용됨

Ex) 티셔츠 – 픽셀1\*w1, 바지 픽셀2\*w1’

현재 다중분류이므로 10개의 이진분류를 수행

클래스 하나를 양성 / 나머지를 음성으로 두는 과정을 모든 클래스에 한해 반복하여 클래스의 개수만큼 z값을 도출(OVR/OVA)

소프트맥스 함수를 적용하여 확률로 변환한다.

* + - 인공신경망과 비교
      * 인공신경망의 구조
        + 입력층과 출력층으로 구분되어 입력층을 대상으로 학습

입력층: 입력데이터(현 과정에선 픽셀) \*객체로 지정하여 입력시킬 떄도 있음

출력층: 출력데이터로 뉴런/유닛이라고 불림(현과정에선 클래스에 따른 z값들)

* + - * + 학습과정

클래스에 따라 입력층에 있는 픽셀들 하나하나에 각자 다른 가중치를 부여하는 로지스틱 회귀 과정을 거쳐 여러 z값 도출. \*절편도 포함되지만 편의상 표기하지 않는 경우가 많음

* + - * + 주의사항:

실제 인간의 신경망과 크게 유사성 X

* + - 주요 라이브러리(2021년 기준):
      * 텐서플로/케라스
    - **케라스 모델**: 가장 기본적인 딥러닝 모델
      * 생성 과정(패션 MNIST 기반)
        + Train\_test\_split()을 활용하여 훈련세트(80) 테스트 세트(20%) 분리

매개변수: test\_size = : 테스트세트의 비율 설정

* + - * + 출력층의 유닛 개수 지정 \*항상 클래스의 개수와 동일해야함

Kera.layers.Dense()

Dense(밀집층, 완전연결층): 케라스의 가장 기본이 되는 층

케라스의 층들은 모두 layers 안에 포함됨

매개변수

10 : 시작부분에 작성하여 유닛의 개수를 지정해줌

Activation = : 적용할 분류기를 지정하는 것으로 현 상황에선 다중분류이므로 소프트맥스함수 사용

‘softmamx’ : 소프트맥스

‘sigmoid’ : 시그모이드

Input\_shape=(,) : 입력층을 객체로 지정하여 요소를 추가하는 경우 그에 따른 입력층의 크기를 설정. 현 상황에선 input\_shape=(784, ) = 784개의 원소로 이루어진 1차원 배열 \*객체 지정 안해도 관례로 행함

* + - * + 모델 변수를 객체로 지정 🡪 훈련모델 생성하기

Model = keras.sequential(dense) : 인공신경망 모델에 객체를 지정하는 것으로 매개변수로 dense라는 출력층 입력

* + - * + 모델 설정

Model.compile() : 매개변수를 통해 모델에 설정을 추가한다.

매개변수:

Loss = : 손실값을 기록 및 출력하라는 매개변수

Loss = ‘categorical\_crossentropy’ : 크로스엔트로피손실함수(다중분류일 경우)

‘binary crossentropy’ : 로지스틱손실함수(이중분류일 경우)

‘sparse\_categorical\_crossentropy’ : 타깃값들이 원핫인코딩에 용이하게 0과 1의 형태가 아닌 그 외의 정수값들을 포함할 때 사용

Metrics=’accuracy’ : 추가적으로 정확도를 측정

* + - * + 모델 훈련

Model.fit(train\_scaled, train\_target, epochs=5)

* + - * + 모델을 테스트세트에 적용한 뒤 결과 출력

Model.evaluate(val\_scaled, val\_target)

* + - * 사이킷런과 케라스 API비교
        + 사이킷런은 클래스를 만들 때 가능한 많은 매개변수 사용
        + 케라스는 세가지 부분으로 분화

층을 만드는 부분(.layers)

모델객체를 지정하고 출력지표(compile())를 설정

훈련 및 평가를 하는 부분