**Advance Data Mining**

**경희대학교 산업경영공학과**

Reading Assignment

**Chapter 9, 10**



고급데이터마이닝

진창호

산업경영공학과

2016100937

김성수

과목명

담당교수

학과

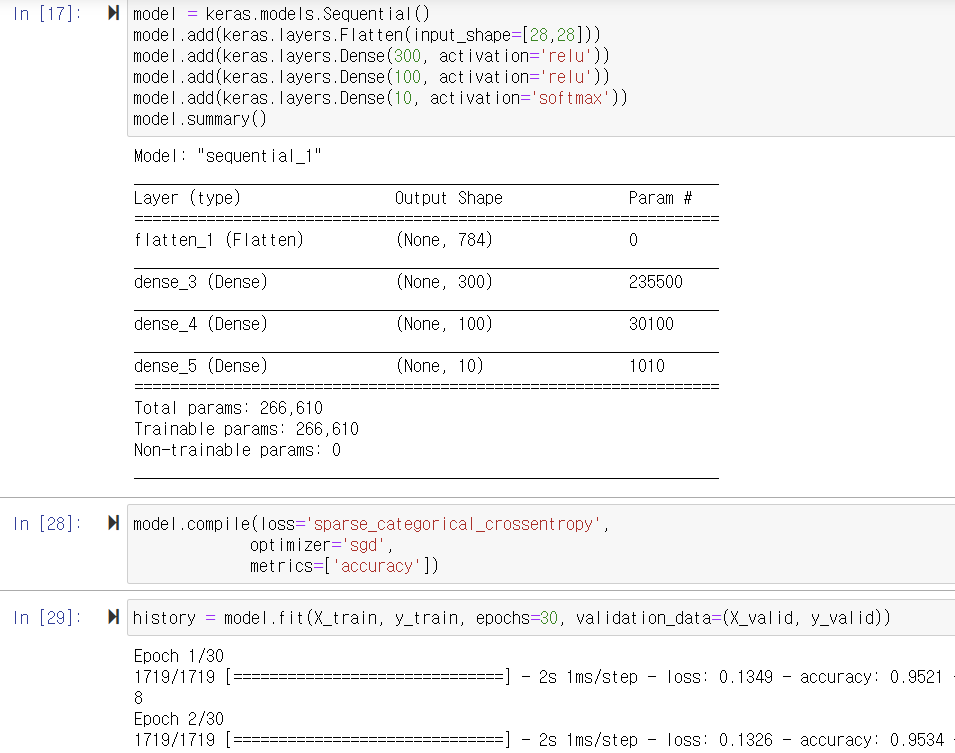
학번

이름

**Chapter9**

* 군집: 비지도 학습으로 비슷한 샘플간 클러스터로 모으는 것을 의미  
  ■ 활용방안  
   (1) 이미지 분할: 이미지를 세그먼트 여러 개로 분할하는 작업으로, 시맨틱 분할에서는   
   동일한 종류의 물체에 속한 모든 픽셀은 같은 세그먼트에 할당됨  
   (2) 군집을 활용한 전처리: MNIST dataset에서 차원축소에 있어 군집을 활용가능  
   (3) 군집을 사용한 준지도 학습: 레이블이 없는 데이터가 많고 레이블이 있는 데이터가 적을 때 사용. 즉, 동일한 클러스터에 속한 샘플들에게 레이블을 전파.  
  ■ 관련 알고리즘  
   (1) k-평균: 군집화 하고자 하는 군집의 수 k를 정한 후 초기 랜덤하게 중심점이 설정되고 중심점을 기준으로 가까운 샘플에 레이블을 할당, 그 후 나눠진 샘플들의 평균으로 중심점 재설정하는 알고리즘   
   a. 속도가 빠르고 확장이 용이  
   b. 불필요한 계산을 줄이기 위하여 삼각부등식을 활용한 속도개선 가능  
   c. k-평균++ 알고리즘을 활용하여 거리가 먼 중심점 초기화 가능  
   d. 전체 데이터셋을 반복하는 것이 아닌 미니배치 k-평균을 사용하여 초기점을 이동하여   
   속도 개선 가능. 그러나 이는 클러스터의 수가 증가할 때 이너셔가 나쁠 수 있음  
   e. 최적의 클러스터 개수 찾기: 실루엣 점수를 활용  
   - 샘플의 실루엣 계수 = (b-a)/max(a, b). (단, a는 동일한 클러스터에 있는 다른   
   샘플까지의 평균 거리, b는 가장 가까운 클러스터까지 평균거리)  
   - 실루엣 계수는 [-1, 1]의 범위를 갖고 \_1에 가까울수록 클러스터에 잘 속해 있으며   
   -1에 가까우면 이 샘플이 잘못된 클러스터에 할당된 것을 의미  
   - 실루엣 다이어그램을 활용하여 실루엣 계수(너비)와 샘플의 개수(높이)파악 가능  
   f. 단점  
   - 최적이 아닌 솔루션을 피하려면 알고리즘을 여러 번 실행해야 함  
   - 클러스터의 개수 지정 필요성  
   - 클러스터의 크기나 밀집도가 서로 다르거나 원형이 아니면 잘 작동하지 않음  
   (2) DBSCAN: 밀집된 연속적 지역을 클러스터로 정의하여 작은 거리내 샘플의 수를 세고   
   작은 거리내에 최소 샘플 수 만큼 있다면 이를 핵심 샘플로 간주하여 이를 동일한   
   클러스터로 간주. 만약 핵심샘플이 아니고 이웃도 아닌 샘플은 이상치가 됨  
   a. 장점: 클러스터가 충분히 밀집되어 있고 밀집되지 않은 지역을 구분 시 좋은 성능을   
   가지며 클러스터의 모양과 개수에 상관없이 감지할 수 있는 능력이 있고 하이퍼   
   파라미터가 2개뿐  
   b. 단점: 새로운 샘플에 대하여 예측할 수 없으며 클러스터 간 밀집도가 다르면   
   올바르게 잡는 것이 불가능하고 계산 복잡도가 선형증가.  
    
   (3) 가우시안 혼합: 샘플의 파라미터가 알려지지 않은 여러 개의 혼합된 가우시안 분포에서 생성되었다고 가정하는 확률모델로 하나의 가우시안 분포에서 생성된 샘플은 하나의 클러스터를 생성하고 일반적으로 이 클러스터는 타원형  
   a. 파라미터: 각 클러스터의 평균, 분산 그리고 각 클러스터의 가중치  
   b. 파라미터 추정 시 활용되는 알고리즘은 EM 알고리즘으로 이는 k-평균과 비슷하나   
   소프트 클러스터 할당을 활용  
   c. 이는 생성 모델이며, 샘플이 주어지면 확률밀도함수의 로그를 예측하여 점수를 계산.   
   점수가 높을수록 밀도가 높음  
   d. 장단점: 타원형 클러스터에 작동을 잘하지만, 다른 모양을 가진 데이터셋에 훈련 시   
   결과가 좋지 않으며 클러스터의 개수가 많거나 샘플의 수가 적으면 최적의 솔루션으로   
   수렴하지 않을 수 있다.  
   e. 가우시안 혼합을 사용한 이상치 및 특이치 탐지: 임의의 임계값에 대하여 밀도가 낮은  
   지역의 샘플을 이상치로 구분하고 만약 이상치가 없는 깨끗한 데이터셋이라고   
   가정한다면 특이치 탐색에 활용 가능  
   f. 클러스터의 개수 정하기: 학습할 파라미터가 많은 모델에는 패널티를 가하고 데이터   
   학습을 잘하는 모델에 보상을 주는 BIC와 AIC척도를 활용. BIC와 AIC는 종종 동일한   
   모델을 선택하지만 BIC가 선택한 모델이 AIC가 선택한 모델보다 간단한 경향이 있으나  
   데이터에 잘 맞지 않을 수 있음.  
   e. 베이즈 가우시안 혼합모델: 최적의 클러스터 개수를 수동으로 찾지 않고 클러스터의   
   가중치를 0으로 가깝게 만들며 자동으로 불필요한 클러스터를 제거  
   (4) 기타 군집 알고리즘: 병합 군집, BIRCH(Balanced Iterative Reducing and Clustering using Hierarchies),  
   평균-이동, 유사도 전파, 스펙트럼 군집 등

**Chapter10**

* 퍼셉트론: 가장 간단한 인공신경망 구조 중 하나로 입력과 출력이 어떤 숫자이며 입력의 가중합을 수행한 후 계단함수를 적용하여 결과를 출력. 이는 하나의 TLU층으로 구성됨.
* 다층 퍼셉트론: 퍼셉트론이 여러 개 쌓아 올린 신경망을 의미하며 XOR문제를 풀 수 있음. 이는 입력층과 하나이상의 은닉층 그리고 출력층으로 구성됨.  
  a. 역전파 알고리즘의 활용: 다층 퍼셉트론의 훈련을 위한 방법론으로 네트워크의 오차를 계산하여 그레이디언트를 생성하는 것을 반복함. 이에 연쇄법칙이 활용되고 여러 번 에포크를 통하여 개선  
  b. 다층 퍼셉트론의 활성함수: 로지스틱 시그모이드 🡪 계단함수는 수평선 밖에 없어 그레이디언트가 없기 때문에 그레이디언트를 정의하기 위함  
  c. 활용방안  
   (1) 회귀: 활성함수를 사용하지 않거나 ReLU를 활용하여 범위의 값을 출력하게 하며,   
   손실함수에는 주로 평균제곱오차가 활용됨. 파라미터로는 은닉층의 수, 뉴런의 수,   
   활성함수, 손실함수 등이 존재함.  
   (2) 분류: 이진분류의 경우 활성함수가 로지스틱 활성함수를 활용하며 출력은 0과 1사이의  
   실수로 예측확률로 해석 가능. 만약 여러 클래스 중 한 클래스에만 속해야 한다면   
   소프트 맥스 활성함수로 대체 가능. 손실함수로는 크로스 엔트로피가 활용되며 하이퍼  
   파라미터는 층의 수, 뉴런의 수, 활성함수, 손실함수 등이 있음
* 다층 퍼셉트론 구현  
  - 시퀀셜 API로 다층퍼셉트론을 만드는 과정  
   (1) 데이터셋 적재  
   (2) 시퀀셜 API를 활용하여 모델 생성: 순서대로 연결된 층을 일렬로 쌓아 구성. 은닉층을   
   추가함에 있어 뉴런의 수 및 활성함수의 종류를 지정 가능. summary() 메소드로 모델 내  
   모든 층의 수, 출력 크기, 파라미터 개수 출력 가능. get\_weights() 메서드로 가중치를   
   출력 가능  
   (3) 모델 컴파일: 손실함수와 옵티마이저를 지정하는 단계로 손실함수가 레이블의 형태에   
   따라 sparse\_categorical\_crossentropy 등으로 지정되며 optimizer는 sgd 등으로 정의   
   그리고 향상시키고자 하는 철도를 metric에 기재함.  
   (4) 모델 훈련 및 평가: fit() 메서드를 활용하여 훈련시키며 class\_weight 매개변수를   
   지정함으로써 훈련 데이터 내 클래스 분포에 대한 패널티를 부여할 수 있음. 모델 성능이  
   만족스럽지 못하면 하이퍼 파라미터의 튜닝이 필요하며, 학습률이나 옵티마이저 등의   
   개선이 필요  
   (5) 모델을 사용하여 예측: 최종적으로 예측한 클래스의 부류 및 각 클래스일 확률을 추출  
  - 시퀀셜 API 외에도 함수형 API를 사용해 복잡한 모델이나 서브클래싱 API로 동적 모델을  
   만들 수 있음  
    
  - 모델의 저장과 복원: 케라스는 HDF5 포맷을 활용하여 모델의 구조 및 층의 모든 모델 파라미터와 옵티마이저까지 저장. 훈련이 오래 걸릴 경우 체크포인트를 통해 중간중간 저장가능하며 콜백 메소드로 다시금 체크포인트를 불러오는 것이 가능  
  - 텐서보드를 활용한 시각화: 학습 곡선을 그리거나 모델이 생성한 이미지, 3D의 복잡한 다차원 데이터를 시각화 및 자동으로 클러스터링 가능
* 신경망 하이퍼파라미터 튜닝  
  - 은닉층의 개수: 복잡한 문제에서는 심층 신경망이 파라미터 효율성이 더 좋음. 계층 구조는 심층 신경망이 좋은 솔루션으로 빨리 수렴하게 도와주며 새로운 데이터에 대하여 일반화 성능을 향상시킴. 뉴런의 층의 수를 점진적으로 늘려가다가 과대적합 직전 조기종료하는 방식을 선택.  
  - 은닉층의 뉴런 개수: 입력층과 출력층의 뉴련 개수는 입력과 출력의 형태에 따라 결정됨. 은닉층은 일반적으로 점점 줄여가며 깔때기 형태로 구성됨. 일단은 과대적합이 되기 전까지 점진적으로 뉴런의 수를 늘리며 그 후 조기 종료나 규제기법을 활용하는 것이 바람직. 일반적으로는 층의 뉴런 수 보다 층의 수를 늘리는 것이 이득이 됨.  
  - 학습률: 가장 중요한 파라미터로 매우 낮은 학습률에서 시작하여 점진적으로 큰 학습률까지 반복하는 것이 필요. 학습률이 낮으면 손실이 줄어들다가 크게 하면 손실이 증가하는 경우가 있음. 이 두 값 사이가 최적의 학습률이 될 것.  
  - 옵티마이저: 고전적인 평범한 미니배치 경사 하강법이 있으나, 더욱 발전한 고급 옵티마이저 방식이 존재함(11장)  
  - 배치크기: 모델의 성능과 훈련시간에 큰 영향을 끼침. 큰 배치는 종종 불안전하게 훈련하는 경우가 있으며 작은 배치라면 일반화성능을 내지 못하는 경우가 있음. 배치크기가 커질수록 훈련시간이 획기적으로 줄어드는 장점이 있음.  
  - 활성화 함수: 수행하는 작업에 따라 활성함수가 달라지며 일반적으로 ReLU가 모든 은닉층의 default값.  
  - 반복횟수: 대부분 튜닝이 필요 없으며 이를 조기 종료하는 방식을 선택하는 것을 선호