|  |  |
| --- | --- |
| 교육 제목 | AI 프로젝트 기반 S/W 전문가 양성과정 |
| 교육 일시 | 21.12.7 |
| 교육 장소 | 집 |
| **교육 내용** | |
| 오전 | 1. CNN review    1. (train\_input, train\_target), (test\_input, test\_target) = mnist.load\_data()    2. train\_test\_split()    3. Conv2D(padding = “same”) = 테두리 쳐서 원본사이즈 유지 28 \* 28    4. Maxpooling2D(2) Conv2D 한 것을 가로 세로 1/2 14 \* 14    5. Conv2D(padding = “same”) 14 \* 14    6. Maxpooling2D(2) Conv2D 한 것을 가로 세로 1/2 7 \* 7    7. flatten() 1차원으로    8. Dense    9. Dropout 🡪 일정 random % 노드 계산 안하기    10. Dense    11. compile    12. checkpoint    13. early\_stopping    14. model.fit(X\_t, Y\_t, epochs = n, validation\_data = (X\_v, Y\_v) 2. 텍스트의 토큰화    1. from keras.preprocessing.text import Tokenizer token = Tokenizer() fit\_on\_texts(docs) 🡪 model.fit 같은 개념인데 단어를 나누는 과정    2. x = token.texts\_to\_sequences([text]) 토큰의 인덱스로만 채워진 새로운 배열 만듦    3. padded\_x = pad\_sequences(x, 4) 길이가 길면 자름, 길이가 4보다 길면 앞에 것을 잘라 냄 길이가 짧으면 0을 넣음 CNN의 padding 개념이랑 같음    4. Embedding(word\_size, 8, input\_length = 4) 3. CNN은 순서 X 병렬처리 가능 RNN(Recurrent Neural Network, 순환 신경망)은 순서 매우 중요함    1. 오늘 주가가 얼마야? 오늘(기억) 🡪 오늘 주가가(기억) 🡪 오늘 주가가 얼마야? 🡪 출력 (기억)을 관리하는 것이 LTSM : Long-Short Term Memory FIFO 개념, |
| 오후 | 1. 순차적 개념 2. 순환 신경망 RNN    1. text가 들어감 🡪 문장을 단어로 쪼갬 tokeniza    2. (전처리) 각 문장마다 단어의 길이가 다르니까 단어의 길이를 동일하게 pad\_seq()    3. embedding    4. dropout    5. conv    6. maxpulling    7. lstm    8. sigmoid 3. 딥러닝 모델 학습법    1. 데이터를 학습, 검증, 테스트 데이터로 분리    2. 카테고리 데이터인 경우 원-핫 인코딩을 이용    3. 데이터의 특성을 극대화 합성곱을 이용하여 특성에 가중치를 곱해서 극대화 시킴 이미지 : conv2d <- 커널 사이즈 2차원 배열 기존의 사이즈를 변경하지 않기 위해서는 padding = “same” 텍스트 : conv1d <- 커널 사이즈의 1차원 배열 maxpooling1d(커널 사이즈) LSTM() : 타임스텝에 가중치를 부여, 기억, GRU()    4. 과적합 방지를 위해서 dropout(drop할 자료의 %)    5. 텍스트의 경우 문장마다 사이즈가 다르므로 문장의 사이즈 통일시킴 padding : pad\_sequences, 모자라면 0으로 채움, 넘치면 앞에서 자름 임베딩 : 정해진 단어를 주어지는 배열로 압축, 공간 낭비를 없앰    6. 콜백을 이용하여 손실함수 변동이 없을 시 조기 중단 가능    7. 여러 개의 층을 쌓을 경우 Dense()는 1차원 배열을 입력으로 받으므로 Flatten()을 사용하여 1차원 배열로 변환 후 Dense layer 처리    8. 마지막 출력층은 sigmoid 또는 softmax 활성함수를 사용 |