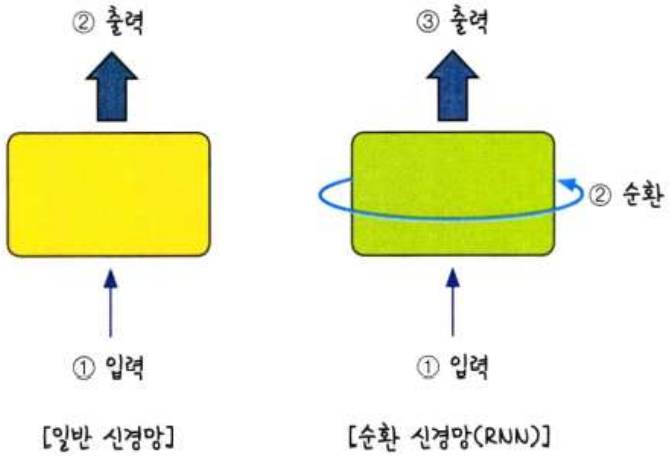
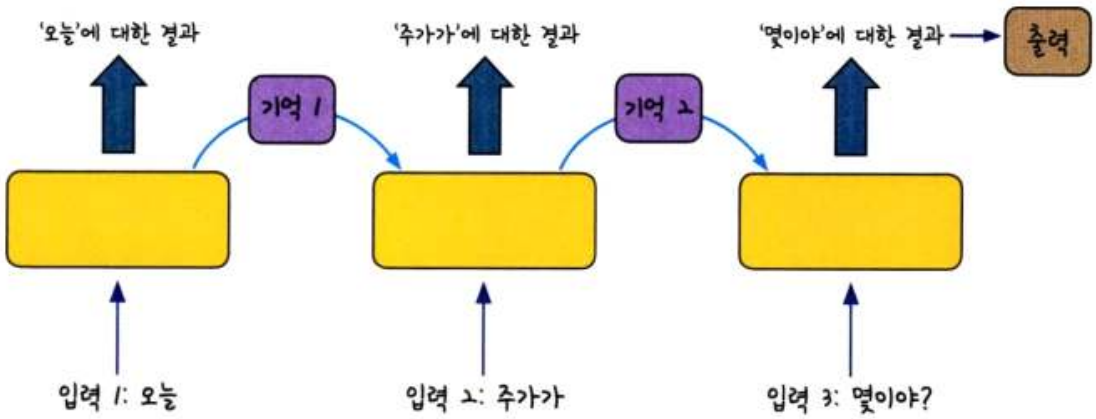
시퀀스 배열로 다루는 순환 신경망(RNN)

문장은 여러 개의 단어로 이루어져 있는데 그 의미를 전달하려면 각 단어가 정해진 순서대로 입력되어야 함. -> 과거에 입력된 데이터와 나중에 입력되 데이터 사이의 관계를 고려해야 하는 문제가 생김 -> **순환 신경망(RNN)**

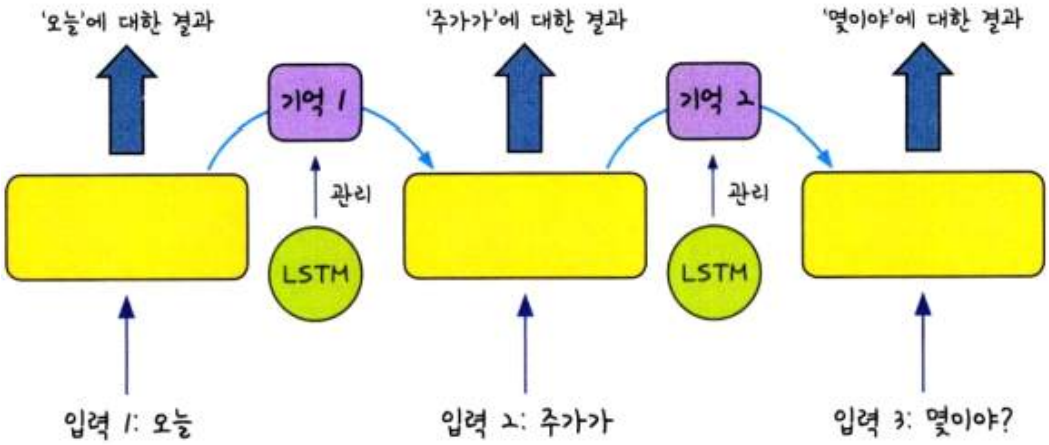
순환 신경망(RNN) - 여러 개의 데이터가 순서대로 입력되었을 때 앞서 입력받은 데이터를 잠시 기억해 놓는 방법

기억된 데이터가 얼마나 중요한지를 판단하여 별도의 가중치를 줘서 다음 데이터로 넘김. 모든 입력 값에 이 작업을 순서대로 실행하므로 다음 층으로 넘어가기 전에 같은 층을 맴도는 것처럼 보임





순환이 되는 가운데 앞서 나온 입력에 대한 결과가 뒤에 나오는 입력 값에 영향을 줌. 이렇게 해야지만, 비슷한 두 문장이 입력되었을 때 그 차이를 구별하여 출력 값에 반영할 수가 있음.



RNN의 결과를 더욱 개선하기 위한 노력 -> LSTM

LSTM(Long Short Term Memory) - 한 층 안에서 반복을 많이 해야 하는 RNN의 특성상 일반 신경망보다 기울기 소실 문제가 더 많이 발생하고 이를 해결하기 어렵다는 단점을 보완한 방법

반복되기 직전에 다음 층으로 기억된 값을 넘길지 안 넘길지를 관리하는 단계를 하나 더 추가하는 것임!

RNN의 장점: 입력 값과 출력 값을 어떻게 설정하느냐에 따라 여러 가지 상황에 적용할 수 있음!

1. LSTM을 이용한 로이터 뉴스 카테고리 분류하기

(실습코드는 github 참고)

입력된 문장의 의미를 파악하는 것은 곧 모든 단어를 종합하여 하나의 카테고리로 분류하는 작업

안녕. 오늘 날씨가 참 좋네 -> 인사

중부 지방은 대체로 맑겠으나, 남부 지방은 구름이 많겠습니다. -> 날씨

올 초부터 유동성의 힘으로 주가가 일정하게 상승했습니다. -> 주식

이번 선거에서는 누가 이길 것 같아? -> 정치

퍼셉트론의 한계를 극복한 신경망이 다시 뜨고 있대. -> 딥러닝

총 11228개의 뉴스 기사가 46개의 카테고리로 나누어진 대용량 텍스트 데이터, 로이터 뉴스 데이터를 사용하여 학습을 시켜보자.

numpy.max() 함수 - 배열의 길이를 세어 줌(0부터 세기 때문에 1을 더해야 함)

print(X_train[0]) 으로 기사를 출력해 보니 단어가 나오는 게 아니라 숫자가 나옴. 딥러닝은 단어를 그대로 사용하지 않고 숫자로 변환한 다음 학습할 수 있음(해당 단어가 몇 번이나 나타나는지 세어 빈도에 따라 번호를 붙였음)

(X_train, Y_train), (X_test, Y_test) = reuters.load_data(num_words=1000, test_split=0.2)
-> 데이터의 20% 테스트셋으로 이용, 빈도가 1~1000에 해당하는 단어만 선택해서 불러오기

각 기사의 단어 수가 제각각 이므로 이를 동일하게 맞춰야 함 -> **sequence() 사용**

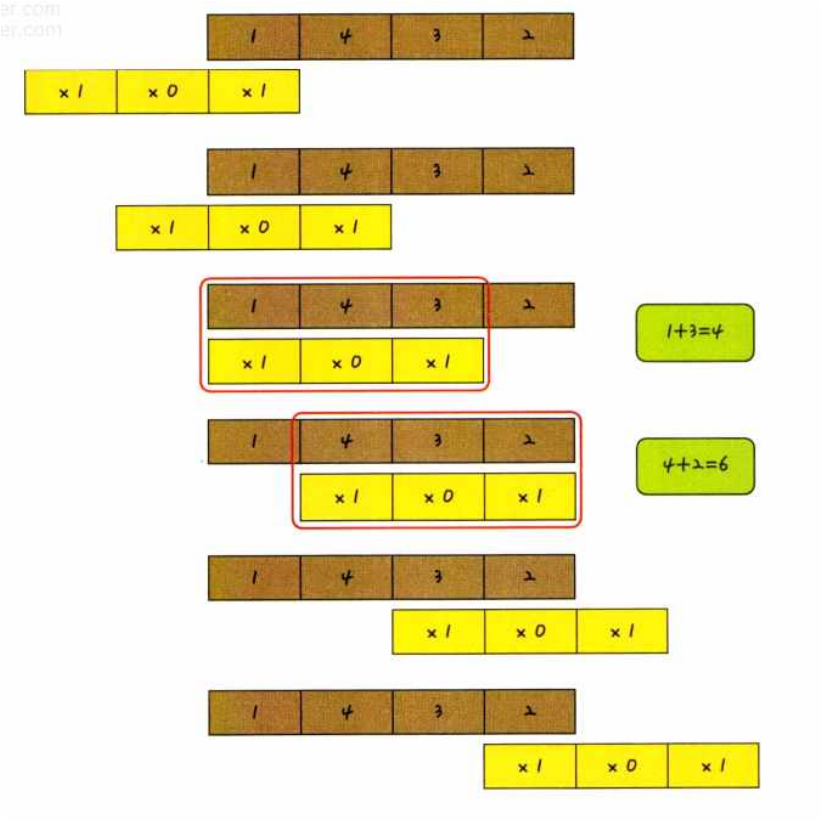
LSTM은 RNN에서 기억 값에 대한 가중치를 제어하며, LSTM(기사당 단어 수, 기타 옵션)의 형식으로 적용됨.

2. LSTM과 CNN의 조합을 이용한 영화 리뷰 분류하기
(실습코드는 github 참고)
영화 데이터베이스(IMDB) - 여오하와 관련된 정보와 출연진 정보, 개봉 정보, 영화 후기, 평점에 이르기까지 매우 폭넓은 데이터가 저장된 자료

앞서 다루었던 로이터 뉴스 데이터와 마찬가지로 각 단어에 대한 전처리를 마친 상태, 데이터셋에서 나타나는 빈도에 따라 번호가 정해지므로 빈도가 높은 데이터를 불러와 학습시킬 수 있음.

데이터 전처리 과정은 로이터 뉴스 데이터와 거의 같으나 클래스가 긍정 또는 부정 두 가지뿐이라 원-핫 인코딩 과정이 없음

Conv1D - Conv2D의 개념을 1차원으로 옮긴 것, 컨볼루션 층이 1차원이고 이동하는 배열도 1차원



(노란색 레이어가 커널을 의미, 이 커널이 지나가면서 원래의 1차원 배열에 가중치를 각각 곱하여 새로운 층인 컨볼루션 층을 만듬)

MaxPooling1D - 2차원 배열이 1차원으로 바뀌어 정해진 구역 안에서 가장 큰 값을 다음 층을 넘기고 나머지는 버린다.