지금까지 테라스를 배웠지요

imdb
처음에는 데이터를 불러 x_train 과 y_train데이터 - x_train에는 여러개의 숫자가 있었다는 것을 기억합시다 - 숫자 몇개 냐는 질문을 했을때 - 정답은 모른다 - 여기 숫자는 단어의 숫자인데 - 이는 정해 져 있는 것이 아니니까 (단어의 숫자 만큼만 들어 오는 것)- 그 이후에 패딩등으로 작업만 하는거 - 이렇게 페어 데이터들이 있어서 12000개 정도로 훈련 - 이를 해석하고자 하면 word2dix라는 사전의 힘이 필요로 했다 - bad를 숫자로 표현해주고 하는 것 - 처음시작을 1로 x - train - 딕셔너리에는 이가 없으니까 추가 해주었다 1 = start / pad = 0 / 우리는 25000이라 쓰면 10000이상이면 unk처리를 해주어라로 10000의 사이즈로 만들어 놓으니까 - unk =2 번으로 만들어 놨었음 - 3번은 unused - 사전을 재 정의 했음 - x_train부분을 maxim 부분을 256로 잡아 두고 길이를 길이가 길면 또 짤라버리고 0번을 붙여 주는 작업까지 했음 - 시험문제 어떤 어떤 과정을 거치면 - 처음엔 data를 load 했을 때 - number of words - x train - 25675라는 숫자가 있을 까 없을까? - 없다가 정답
idx2word- 라는 거꾸로의 작업도 거쳤음 - create id blah blah - 이 만드는 부분에 대한 공부도 해야함
패딩을 해서 총 256개의 백터를 만들어줌 -
embedding- 의 의미와 적용은 무조건 시험문제!
원래 동그라미가 하나 있다고 가정 - 임베딩에는 voca size / 128(기존) - 하나를 추가해서 뒤에 256을 적는다 256은 input의 length임. - embedding은 입력이 원래 1개 이나 756이
일단 동그라미가 10000개 생기고(voca size) 한단어에 해당하는 거에 756이라는 숫자가 들어왔다고
10000중 756번째 숫자에 1이 들어간다고 생각하면 / 이것이 128개 - hidden node가 128개 아티클 하나를 가주고 있다(수 많은 숫자의 모임) 리뷰가 1 또는 0으로 평가 - x train을 받았을 때 총 256개의 숫자로 했어 놧던거 기억 + voca size 10000으로 해놧던것도 기억해내라 - 패딩으로 나머지는 0으로 했던거 기억 - 756해당이 여기 256개 중에 들어가고 이게 또 128에 들어가고 결과 값 하나에 들어가게 되는 것이다. -
엠베딩은 input(256)에서 히든 nod 128까지의 선까지를 포함

한 아티클은 1개 이상의 단어를 가지고 있는데 이것을 어떻게 해결 할 것인가? 하나의 숫자를 그 voca size만큼 input node를 만들고 원 핫코딩 - denseweight가 생기고 - hidden node(128)개가 생김
원핫코딩의 레이어는 dimension은 1 * 10000임. 756이라는 하나의 숫자는 1 * 1 그러나 아티클의 총 단어는 256개 그래서 756은 256 * 1 이 되는 것 256 * 1 - 256 * 10000 - 256 * 128 - 로 계속 되는 것
이어 주는 선은 10000 * 128 그대로 / 256은 input - length - 한 아티클의 개수
