

이강욱 교수님에게

교수님 메일 잘 받았습니다. 정말 교수님 말씀처럼 작은 GPT 코드를 따라하고 공부해보면서 개념만 알았던 몇일 전과는 다르게 그 세부 구성이 어떻게 작동되는지를 공부할 수 있었습니다.

어려웠던 점이라고 한다면 그 세부 내용의 대부분이 새로 접하는 것이었다는 것이었습니다. 클래스나 함수를 구성하고 변수를 정의하는 등 python을 사용한다는 점에서는 친숙했지만 N-gram 언어모델이나 기존에 배웠던 RNN이 아닌 Attention Transformer 등 아직 잘 알지 못하는 개념이 나와서 어색했고 동시에 수월했던 점은 유튜브와 구글에 검색을 하니 상세히 잘 알기 쉽게 설명이 되어 있어서 공부 하면서 코드를 따라할 수 있었다는 점이었습니다.

우선 텍스트를 Tokenization 하는 것부터 새로웠습니다. 기존에 제가 해왔던 머신러닝에서는 csv 표 안에 포함 되어있는 숫자들과 null값들 뿐이었는데 텍스트 자체를 분석하고 다루기 위해서는 텍스트를 하나의 정수 값으로 대치하는 것이 필요하다는 것을 배웠습니다. 또한 직관적으로 다음에 올 단어에 대해 생각하는 우리 뇌구조와 다르게 확률적 접근[P(다음에 올 단어 | 이전 단어)]을 통해 다음 단어를 선택하는 N-gram Language 모델과 기존 RNN과는 다르게 훨씬 빠르고, 기울기 소실, 거리가 떨어진 지점의 있는 것과의 상관관계를 다루기 어렵다는 단점을 극복한 Attention Transformer라는 것을 사용한다는 것도 알게 되었습니다. Transformer의 세부 과정 중 토큰 Embedding과 위치값을 더해서 문장 내에 순서에 따른 차이를 인식하기도 하고 Attention 변환을 진행 할 때 대각행렬 위의 값들은 -inf 값을 통해 softmax의 넣었을때 전부 0으로 만들어 현재 예측 되는 값이 전에 있는 행렬 원소에게만 영향 받게 한다는 것도 배웠고 문장의 positive/negative를 구분하는 감정분석을 진행할 시에는 지금과는 다르게 다음에 오는 값에도 영향을 받게 하여 더 넓은 범위에서 관계를 분석해 낸다는 것도 흥미로웠습니다. 결국 이 Attention 변환을 통해 가장 주목해야할 단어가 무엇인지를 찾아내고 loss값을 줄여가는 방향으로 데이터를 학습시켜 가장 최적의 답변을 생성해 나아가는 과정의 GPT를 만들어보면서 머신러닝이라는 것은 인간의 관점이 아닌 기계의 장점을 최대화 할 수 있는 관점에서 학습을 시키고 우리가 필요로 하는 것을 얻어내는 것이 아닐까 생각했습니다.

저는 셰익스피어 텍스트 대신 kaggle에서 라디오헤드, 저스틴비버, 브루노 마스의 가사 정보가 담긴 텍스트를 이용해 학습을 시켜봤습니다. 락과 RnB를 섞은 느낌의 가사가 나오지 않을까 생각하며 학습을 시켰지만 결론적으로 그 정도의 산출물을 얻어내지는 못했습니다. 학습을 시키고 검증하는 과정에서 생각보다 train loss 값이 val loss 값보다 많이 작아서 과적합이 된다고 판단했고 dropout=0.5 로 설정하고 학습데이터와 검증데이터의 비율을 7:3으로 조정하여 그 차이를 줄였습니다. 그 이후에도 결과값에 말이 안되는 단어들이 많이 나와서 max iteration을 10000번까지 증가시키고 Batch size 와 block size도 유튜브 영상에서처럼 늘리고 싶었지만 제 노트북으로는 연산이 너무 오래 걸리게 되어 32와 64 정도로 증가시키니 가사라고 하기는 유의미하지 않지만 love me, take you hevere, You'd better 등 주어 다음 동사가 철자를 맞추며 써지는 수준의 결과를 확인 할 수 있었습니다.

저의 부족한점을 많이 느꼈고 동시에 그 부족한 부분을 조금씩 채워가면서 이 분야에 대한 흥미를 더 느꼈고 앞으로도 이렇게 찾아보면서 열심히 공부하면 되겠다는 자신감을 얻을 수 있었던 것 같습니다. 이런 좋은 기회 주시고 긴 답변 읽어 주셔서 감사합니다.