작성일자 : 2021/01/14

2020 인공지능 학습용 데이터 해커톤 대회

학습 모델 설명

한양대학교 에리카 캠퍼스 학부생 박태민

0. 제원

Python을 이용하였고,

신경망 모듈 개발을 위한 주 라이브러리는 pytorch의 torch를 이용하였습니다.

부가적인 라이브러리로는 mfcc를 만들기 위해 librosa와 numpy등을 이용하였습니다.

오픈소스로는

Kor2vec 모델을 사용하기 위해

네이버 클로바 AI 팀이 개발한 kor2vec 라이브러리를 이용하였습니다.

[GitHub - naver/kor2vec: OOV없이 빠르고 정확한 한국어 Embedding 라이브러리](https://github.com/naver/kor2vec)

1. 전체 과정

학습은 input은 pcm파일, output은 [단어길이, feature]만큼의 embedding vector입니다.

단어길이는 21, feature은 128로 설정했습니다.

왜냐하면 word-embedding을 구현 할 때, 오픈소스 kor2vec를 이용하였는데,

여기에서 feature를 128로 설정했기 때문입니다.

따라서 첫번째로 제공된 학습데이터에서 전체 문장들을 단어로 쪼개고, 중복단어를 없애서 단어만 모여있는 txt파일을 만듭니다.

그 다음, kor2vec모델을 이용해 단어를 전부 tensor vector로 변경합니다. (mapping\_8.txt)

pcm\_dataloader\_8.py는 데이터로더인데, 여기서 동일한 kor2vec모델을 이용해서 정답 문장을 [21, 128] 차원의 tensor vector로 변경합니다.

(단어의 개수는 21개보다 작은 문장이 있을 것입니다. 이는 eof라는 영어로 따로 만들어 준 후, mapping\_8.txt에 추가합니다.)

다음 자체적으로 개발한 Speak2Embed 모델(stt\_model\_8)을 통해 PCM파일을 입력, 출력 [21, 128]의 tensor vector를 생성합니다.

정답 문장의 tensor vector와 입력값 pcm 파일에 Speak2Embed모델을 거친 output vector의 loss를 계산합니다. 이때 손실함수로 SmoothL1Loss를 이용합니다.

최종적으로 테스트는 batch는 1이며, 하나의 pcm파일이 들어가면 [21, 128] tensor vector가 생성됩니다.

이를 미리 kor2vec를 통해 임베딩을 거친 벡터(vectormap\_8.pt)와 for문으로 제일 loss가 낮은 vector의 인덱스를 찾습니다.

그 인덱스는 mapping\_8.txt의 단어의 인덱스와 동일합니다.

따라서 mapping\_8.txt의 인덱스에 해당하는 단어를 뽑아냅니다.

단어의 길이가 21이므로 21번 반복합니다. ‘eof’가 나오면, 공백으로 대체합니다.

마지막으로 해당 문장을 txt파일로 저장합니다.

2. Speak2Embed 모델 설명

Speak2Embed 모델은 pcm 음성 파일을 길이 21, feature 128의 output 벡터로 만들어냅니다.

크게 두가지 파트가 었습니다.

첫번째는 sequence 자체를 1D convolution을 통해 channel 128의 vector(A 벡터라고 하겠습니다)를 만듭니다.

그리고, 주파수 특성의 학습도 고려했습니다.

따라서 Speak2Embed 모델 내부에서 librosa를 이용해 mfcc를 만듭니다. 여기서 mfcc는

‘Mel Frequency Cepstral Coefficient’로 mel spectrogram을 DCT(Discrete Cosine Transform) 처리하면 얻게되는 coefficient를 말합니다.  
따라서 sequence를 넣어 만든 mfcc에 대해 2D convolution을 수행, channel 128의 vector를 만듭니다(B 벡터라고 하겠습니다)

최종적으로 A벡터와 B벡터를 더한 후, 정규화 Transposed 1D convolution을 거쳐 [21, 128]의 output tensor를 생성합니다.