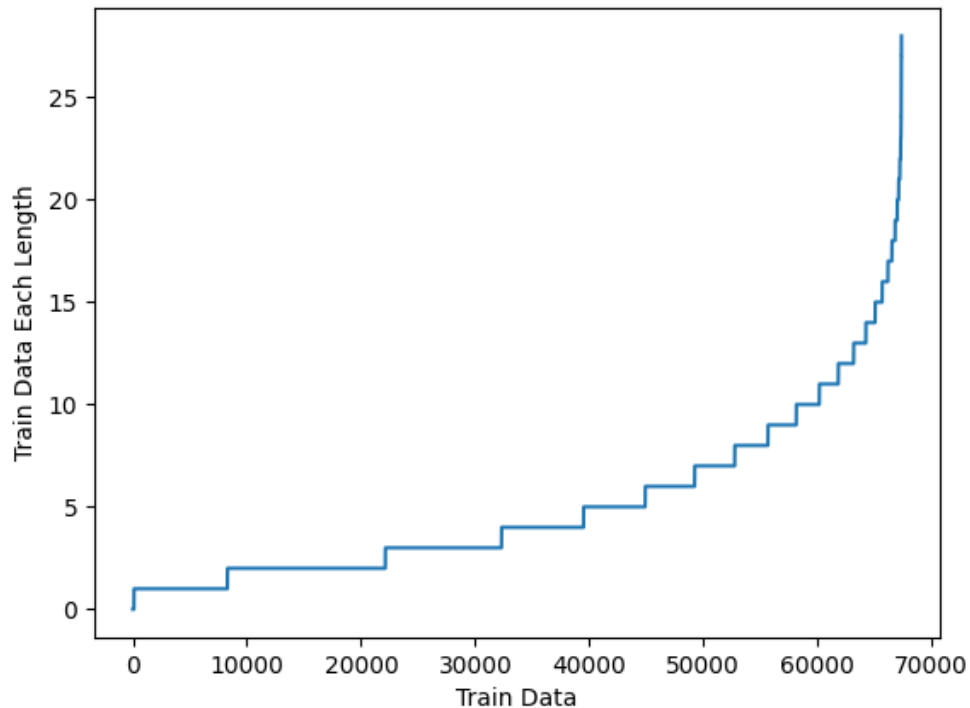


Transformer기반 모델 sentiment classification 구현 과제

1. 구두점 제거
 - a. Noise로 간주되는 구두점을 제거하였습니다.
 - b. 이는, text의 실제 의미와 관계가 없는 요소로 판단하였으며, 토큰화의 일관성을 향상시켰습니다.
2. 소문자 변환
 - a. 대소문자의 구분으로 똑같은 단어이지만, 대문자와 소문자 때문에 다르게 해석될 위험이 있어 소문자로 통일된 형식으로 변환하여 일관성을 부여했습니다.
 - b. 이는, Dataset의 어휘 크기를 줄여줄 수 있습니다.
3. 토큰화
 - a. text를 작은 단위로 분할하여 구조화할 수 있게끔 했습니다.
 - b. Likelihood를 계산하기 위해 토큰화를 진행을 하기도 했습니다.
4. 불용어 제거
 - a. 문법적으로는 중요한 단어이지만, text analysis에서는 큰 영향을 미치지 않는다고 생각했습니다.
 - b. 이는, 핵심 단어에 집중할 수 있도록 했습니다.
5. 표제어 추출
 - a. 단어의 다양한 형태를 하나의 표제어로 통일하게 되면, 어휘의 일관성이 향상 된다고 합니다. 이를 통해, text를 더 효율적으로 분석할 수 있게 되었습니다.
 - b. 어휘의 다양성을 줄였습니다. 이는 data의 dimension을 낮추기 때문에 inference과정에서 더 효율적입니다.
6. 단어의 길이
 - a. network에 input datat를 넣기 위해서는, data의 크기(문장의 길이)는 동일해야 한다. train data의 최대 단어 길이는 28이었으며, 단어의 길이를 28을 기준으로 모든 vector를 28로 길이를 맞춰 주었다. (padding을 적용하였다.)

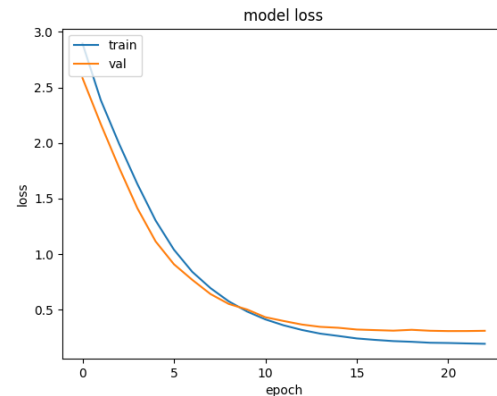
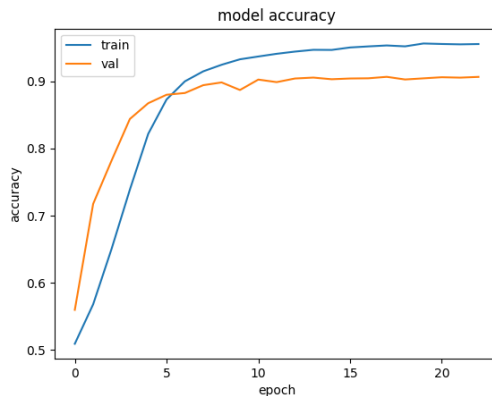


b. padding은 post로 정해주었습니다.

Modeling Layers

- token의 vector dimension은 28로 맞추었습니다.
 - 이는 SST-dataset의 각 token의 최대 길이가 28이어서, 각각의 token의 vector는 28로 맞추어줬습니다.
 - vector size는 padding을 통해서 채웠습니다.
 - 따라서 transformer의 embedding vector size는 28로 맞추었습니다.
 - 논문에서 attention head는 $N = 6$ 을 사용하였지만, 저는 2를 사용했습니다.
 - transformer의 각각의 sub layer의 Feed Forward의 hidden layer의 수는 32로 사용하였습니다.
 - transformer block을 거치고 난 뒤의 Linear, softmax를 적용하였으며, 이때의 Linear의 hidden layer의 노드 개수는 20으로 설정하였으며, overfitting을 막기 위해 L2 Regularization과 Dropout을 사용하였습니다.
-

Metrics



모델이 학습할 때, 이전까지는 Adam의 optimizer를 이용하였지만 이번에는 Adamw의 optimizer를 이용하였으며, 이는 기존의 Adam보다 local minimum으로 더 빨리 수렴하였습니다.

Learning Rate가 0.1일 때, epoch가 2~4에서 끝나 발산 할 가능성과 overfitting이 일어날 확률을 생각하여 1e-4로 설정하였으며, 마찬가지로 overfitting을 방지하기 위하여 weight decay를 설정해주었습니다.

Learning Rate를 작게 설정하였기 때문에 epoch를 30으로 설정하였으며, batch size는 256, validation dataset은 train dataset의 20%를 split해주었습니다.

위의 그래프를 보시면 accuracy와 loss를 나타내고 있는데, 총 30번의 epoch를 설정하였지만 Early Stopping으로 인해 23번의 epoch 학습이 진행되었습니다.

모델이 학습할수록 accuracy와 loss가 점진적으로 좋아지는 것을 볼 수 있습니다.

최종 test data의 대한 Loss는 0.69, Accuracy : 0.78(Loss, Accuracy 모두 3번째 자리에서 반올림)이 나왔으며 target 값의 label인 0(Negative), 1(Positive)에 대한 precision, recall, f1-score의 정보는 아래와 같습니다.

	precision	recall	f1-score	support
0	0.81	0.73	0.77	428
1	0.76	0.84	0.80	444
accuracy			0.78	872
macro avg	0.79	0.78	0.78	872
weighted avg	0.79	0.78	0.78	872