

기존 하이퍼 파라미터

In [51]: `tf.keras.backend.clear_session()`

하이퍼파라미터

`D_MODEL = 256`

`NUM_LAYERS = 2`

`NUM_HEADS = 8`

`DFF = 512`

`DROPOUT = 0.1`

```
model = transformer(  
    vocab_size=VOCAB_SIZE,  
    num_layers=NUM_LAYERS,  
    dff=DFF,  
    d_model=D_MODEL,  
    num_heads=NUM_HEADS,  
    dropout=DROPOUT)
```

`(1, 8180, 256)`

`(1, 8180, 256)`

In [52]: `learning_rate = CustomSchedule(D_MODEL)`

```
optimizer = tf.keras.optimizers.Adam(  
    learning_rate, beta_1=0.9, beta_2=0.98, epsilon=1e-9)
```

```
def accuracy(y_true, y_pred):
```

레이블의 크기는 (batch_size, MAX_LENGTH - 1)

`y_true = tf.reshape(y_true, shape=(-1, MAX_LENGTH - 1))`

`return tf.keras.metrics.sparse_categorical_accuracy(y_true, y_pred)`

```
model.compile(optimizer=optimizer, loss=loss_function, metrics=[accuracy])
```

모델크기 키워보기

```
In [64]: # 하이퍼파라미터
D_MODEL = 512 # 임베딩 크기 증가
NUM_LAYERS = 4 # Transformer 레이어 증가
NUM_HEADS = 16 # Multi-Head Attention 헤드 수 증가
DFF = 2048 # Feed-Forward 네트워크 크기 증가
DROPOUT = 0.2 # 드롭아웃 비율 증가

model1 = transformer(
    vocab_size=VOCAB_SIZE,
    num_layers=NUM_LAYERS,
    dff=DFF,
    d_model=D_MODEL,
    num_heads=NUM_HEADS,
    dropout=DROPOUT)

learning_rate = CustomSchedule(D_MODEL)

optimizer = tf.keras.optimizers.Adam(
    learning_rate, beta_1=0.9, beta_2=0.98, epsilon=1e-9)

(1, 8180, 512)
(1, 8180, 512)
```

```
In [65]: '''from nltk.translate.bleu_score import sentence_bleu 에러나서 추가 못함

def bleu_score(y_true, y_pred):
    references = tf.cast(y_true, tf.int32).numpy()
    candidates = tf.argmax(y_pred, axis=-1).numpy()
    score = 0
    for ref, cand in zip(references, candidates):
        score += sentence_bleu([ref], cand)
    return score / len(references)
'''
```

```
In [70]: model1.compile(optimizer=optimizer, loss=loss_function, metrics=[accuracy])
```

초기학습

파라미터 조정

에포크 회수 증가

Epoch 27/50
185/185 [=====] - 10s 53ms/step - loss: 0.0223 - accuracy: 0.1695
Epoch 28/50
185/185 [=====] - 10s 53ms/step - loss: 0.0204 - accuracy: 0.1700
Epoch 29/50
185/185 [=====] - 10s 53ms/step - loss: 0.0178 - accuracy: 0.1707
Epoch 30/50
185/185 [=====] - 10s 53ms/step - loss: 0.0163 - accuracy: 0.1711
Epoch 31/50
185/185 [=====] - 10s 53ms/step - loss: 0.0153 - accuracy: 0.1714
Epoch 32/50
185/185 [=====] - 10s 53ms/step - loss: 0.0142 - accuracy: 0.1715
Epoch 33/50
185/185 [=====] - 10s 53ms/step - loss: 0.0130 - accuracy: 0.1719
Epoch 34/50
185/185 [=====] - 10s 53ms/step - loss: 0.0123 - accuracy: 0.1720
Epoch 35/50
185/185 [=====] - 10s 53ms/step - loss: 0.0113 - accuracy: 0.1723
Epoch 36/50
185/185 [=====] - 10s 53ms/step - loss: 0.0108 - accuracy: 0.1725
Epoch 37/50
185/185 [=====] - 10s 53ms/step - loss: 0.0099 - accuracy: 0.1727
Epoch 38/50
185/185 [=====] - 10s 53ms/step - loss: 0.0097 - accuracy: 0.1727
Epoch 39/50
185/185 [=====] - 10s 53ms/step - loss: 0.0091 - accuracy: 0.1729
Epoch 40/50
185/185 [=====] - 10s 53ms/step - loss: 0.0089 - accuracy: 0.1729
Epoch 41/50
185/185 [=====] - 10s 54ms/step - loss: 0.0080 - accuracy: 0.1731
Epoch 42/50
185/185 [=====] - 10s 54ms/step - loss: 0.0075 - accuracy: 0.1732
Epoch 43/50
185/185 [=====] - 10s 53ms/step - loss: 0.0074 - accuracy: 0.1733
Epoch 44/50
185/185 [=====] - 10s 53ms/step - loss: 0.0070 - accuracy: 0.1734
Epoch 45/50
185/185 [=====] - 10s 53ms/step - loss: 0.0070 - accuracy: 0.1735
Epoch 46/50
185/185 [=====] - 10s 53ms/step - loss: 0.0066 - accuracy: 0.1736
Epoch 47/50
185/185 [=====] - 10s 53ms/step - loss: 0.0063 - accuracy: 0.1736
Epoch 48/50
185/185 [=====] - 10s 53ms/step - loss: 0.0062 - accuracy: 0.1736
Epoch 49/50
185/185 [=====] - 10s 53ms/step - loss: 0.0058 - accuracy: 0.1737
Epoch 50/50
185/185 [=====] - 10s 53ms/step - loss: 0.0055 - accuracy: 0.1738

Epoch 27/50
185/185 [=====] - 38s 207ms/step - loss: 0.1259 - accuracy: 0.1448
Epoch 28/50
185/185 [=====] - 38s 207ms/step - loss: 0.1135 - accuracy: 0.1471
Epoch 29/50
185/185 [=====] - 38s 207ms/step - loss: 0.1032 - accuracy: 0.1493
Epoch 30/50
185/185 [=====] - 38s 207ms/step - loss: 0.0942 - accuracy: 0.1510
Epoch 31/50
185/185 [=====] - 38s 207ms/step - loss: 0.0867 - accuracy: 0.1526
Epoch 32/50
185/185 [=====] - 38s 207ms/step - loss: 0.0804 - accuracy: 0.1540
Epoch 33/50
185/185 [=====] - 38s 207ms/step - loss: 0.0734 - accuracy: 0.1558
Epoch 34/50
185/185 [=====] - 38s 207ms/step - loss: 0.0678 - accuracy: 0.1571
Epoch 35/50
185/185 [=====] - 38s 207ms/step - loss: 0.0616 - accuracy: 0.1587
Epoch 36/50
185/185 [=====] - 38s 207ms/step - loss: 0.0571 - accuracy: 0.1600
Epoch 37/50
185/185 [=====] - 38s 207ms/step - loss: 0.0540 - accuracy: 0.1606
Epoch 38/50
185/185 [=====] - 38s 207ms/step - loss: 0.0499 - accuracy: 0.1615
Epoch 39/50
185/185 [=====] - 38s 207ms/step - loss: 0.0468 - accuracy: 0.1624
Epoch 40/50
185/185 [=====] - 38s 207ms/step - loss: 0.0432 - accuracy: 0.1635
Epoch 41/50
185/185 [=====] - 38s 207ms/step - loss: 0.0420 - accuracy: 0.1637
Epoch 42/50
185/185 [=====] - 38s 206ms/step - loss: 0.0382 - accuracy: 0.1648
Epoch 43/50
185/185 [=====] - 38s 206ms/step - loss: 0.0368 - accuracy: 0.1650
Epoch 44/50
185/185 [=====] - 38s 206ms/step - loss: 0.0338 - accuracy: 0.1658
Epoch 45/50
185/185 [=====] - 38s 206ms/step - loss: 0.0320 - accuracy: 0.1664
Epoch 46/50
185/185 [=====] - 38s 206ms/step - loss: 0.0312 - accuracy: 0.1667
Epoch 47/50
185/185 [=====] - 38s 207ms/step - loss: 0.0289 - accuracy: 0.1672
Epoch 48/50
185/185 [=====] - 38s 206ms/step - loss: 0.0279 - accuracy: 0.1674
Epoch 49/50
185/185 [=====] - 38s 207ms/step - loss: 0.0266 - accuracy: 0.1677
Epoch 50/50
185/185 [=====] - 38s 206ms/step - loss: 0.0248 - accuracy: 0.1684

Epoch 27/50
185/185 [=====] - 38s 206ms/step - loss: 0.0106 - accuracy: 0.1722
Epoch 28/50
185/185 [=====] - 38s 207ms/step - loss: 0.0107 - accuracy: 0.1722
Epoch 29/50
185/185 [=====] - 38s 207ms/step - loss: 0.0105 - accuracy: 0.1722
Epoch 30/50
185/185 [=====] - 38s 206ms/step - loss: 0.0100 - accuracy: 0.1724
Epoch 31/50
185/185 [=====] - 38s 206ms/step - loss: 0.0096 - accuracy: 0.1724
Epoch 32/50
185/185 [=====] - 38s 206ms/step - loss: 0.0098 - accuracy: 0.1724
Epoch 33/50
185/185 [=====] - 38s 206ms/step - loss: 0.0090 - accuracy: 0.1727
Epoch 34/50
185/185 [=====] - 38s 206ms/step - loss: 0.0095 - accuracy: 0.1725
Epoch 35/50
185/185 [=====] - 38s 207ms/step - loss: 0.0089 - accuracy: 0.1726
Epoch 36/50
185/185 [=====] - 38s 206ms/step - loss: 0.0087 - accuracy: 0.1728
Epoch 37/50
185/185 [=====] - 38s 206ms/step - loss: 0.0084 - accuracy: 0.1729
Epoch 38/50
185/185 [=====] - 38s 206ms/step - loss: 0.0088 - accuracy: 0.1726
Epoch 39/50
185/185 [=====] - 38s 207ms/step - loss: 0.0079 - accuracy: 0.1730
Epoch 40/50
185/185 [=====] - 38s 206ms/step - loss: 0.0087 - accuracy: 0.1727
Epoch 41/50
185/185 [=====] - 38s 206ms/step - loss: 0.0081 - accuracy: 0.1729
Epoch 42/50
185/185 [=====] - 38s 206ms/step - loss: 0.0075 - accuracy: 0.1730
Epoch 43/50
185/185 [=====] - 38s 207ms/step - loss: 0.0085 - accuracy: 0.1729
Epoch 44/50
185/185 [=====] - 38s 205ms/step - loss: 0.0073 - accuracy: 0.1731
Epoch 45/50
185/185 [=====] - 38s 206ms/step - loss: 0.0073 - accuracy: 0.1731
Epoch 46/50
185/185 [=====] - 38s 206ms/step - loss: 0.0078 - accuracy: 0.1730
Epoch 47/50
185/185 [=====] - 38s 205ms/step - loss: 0.0079 - accuracy: 0.1729
Epoch 48/50
185/185 [=====] - 38s 207ms/step - loss: 0.0074 - accuracy: 0.1731
Epoch 49/50
185/185 [=====] - 38s 206ms/step - loss: 0.0068 - accuracy: 0.1732
Epoch 50/50
185/185 [=====] - 38s 207ms/step - loss: 0.0072 - accuracy: 0.1731

초기 모델 답변

```
output = predict("영화 볼래?")
```

Input: 영화 볼래?
Output: 최신 영화가 좋을 것 같아요 .

```
output = predict("고민이 있어")
```

Input: 고민이 있어
Output: 제가 고민을 들어드릴게요 .

```
output = predict("이번에 플러터로 ai앱개발을 해야 되는데 무슨 주제가 좋을까?")
```

Input: 이번에 플러터로 ai앱개발을 해야 되는데 무슨 주제가 좋을까?
Output: 제가 들어드릴게요 .

```
output = predict("물음표쓰면 다 제가 들어드릴게요로 답하는거야?")
```

Input: 물음표쓰면 다 제가 들어드릴게요로 답하는거야?
Output: 쉬고 나면 힘이 날거예요 맛있는 것도 드세요

```
output = predict("공부할까?")
```

Input: 공부할까?
Output: 나중에 도움이 될 거예요 .

파라미터 조정 모델

```
output = predict("영화 볼래?")
```

Input: 영화 볼래?
Output: 가능하다면요 .

```
output = predict("고민이 있어")
```

Input: 고민이 있어
Output: 많이 만나보는 게 좋겠지만 연락을 더 자주 해보세요 .

```
output = predict("이번에 플러터로 ai앱개발을 해야 되는데 무슨 주제가 좋을까?")
```

Input: 이번에 플러터로 ai앱개발을 해야 되는데 무슨 주제가 좋을까?
Output: 나중에 후회할 수도 있을 것 같아요 .

```
output = predict("물음표쓰면 다 제가 들어드릴게요로 답하는거야?")
```

Input: 물음표쓰면 다 제가 들어드릴게요로 답하는거야?
Output: 사랑은 한 번 떠난 건 어떨까요 .

```
output = predict("공부할까?")
```

Input: 공부할까?
Output: 나중에 도움이 될 거예요 .

파라미터 조정후 추가학습 모델

```
output = predict("영화 볼래?")
```

Input: 영화 볼래?
Output: 최신 영화가 좋을 것 같아요 .

```
output = predict("고민이 있어")
```

Input: 고민이 있어
Output: 얼른 알아보세요 .

```
output = predict("이번에 플러터로 ai앱개발을 해야 되는데 무슨 주제가 좋을까?")
```

Input: 이번에 플러터로 ai앱개발을 해야 되는데 무슨 주제가 좋을까?
Output: 구두로 먹을 내보세요 .

```
output = predict("물음표쓰면 다 제가 들어드릴게요로 답하는거야?")
```

Input: 물음표쓰면 다 제가 들어드릴게요로 답하는거야?
Output: 저처럼 연락이 늦는 사람을 이해해 주세요 .

```
output = predict("공부할까?")
```

Input: 공부할까?
Output: 나중에 도움이 될 거예요 .

gpt 3.5 터보

localhost:60499

localhost:60499

Chatbot

내가 안녕디지몬이라고 말하면 넌 나와 함께하고 싶어라고 해줘

안녕디지몬! 나와 함께하고 싶어요. 함께 무엇을 할까요?

안녕디지몬

너와 함께하고 싶어요. 함께 무엇을 할까요?

너와 함께하고 싶어 딱 여기까지만 말해줘

안녕디지몬

너와 함께하고 싶어요. 함께 무엇을 할까요?

안녕디지몬

너와 함께하고 싶어요. 함께 무엇을 할까요?

뒤에 요. 함께 무엇을 할까요? 붙이지 말아줘

네, 알겠어요. 함께 무엇을 할까요?

메시지를 입력하세요

gpt 4o

