딥러닝 실제와 응용 과제 – 2022920024 박동찬

1. Pretrained_BERT
   A. 가설 설정

초기에 주어진 하이퍼파라미터 값(learning_rate=1e-5, batch_size=32, epochs=10, pretrained_model='bert-base-uncased')으로 코드를 실행하는 경우, 결과값은 아래 표와 같다.

Epoch	Training Loss	Validation Accuracy	Validation Loss	Test Accuracy
1	0.57	0.80	0.48	N/A
2	0.40	0.83	0.41	N/A
3	0.29	0.84	0.42	N/A
4	0.21	0.84	0.46	N/A
5	0.17	0.84	0.59	N/A
6	0.13	0.84	0.60	N/A
7	0.11	0.82	0.71	N/A
8	0.10	0.85	0.67	N/A
9	0.08	0.83	0.73	N/A
10	0.07	0.84	0.71	0.84

Training Loss는 epoch가 진행함에 따라 감소하고 있으므로 모델이 훈련 데이터에 대해 점점 더 적합해지고 있다는 것을 의미한다. Validation Accuracy의 경우, 증가하다가 일정한 약 0.84에서 정체되고 있다. Validation Loss의 경우, 손실이 꾸준히 증가하고 있는 것을 볼 수 있다. 이는 모델이 훈련 데이터에 의해 overfitting 되고 있어 검증 데이터에 대한 일반화 성능이 감소하고 있음을 알 수 있다. 최종 Test Accuracy는 0.84가 도출되었다. 모델이 overfitting 되지 않도록 하이퍼파라미터를 조절한다면 더 좋은 성능을 얻을 수 있을 것이다. Test Accuracy를 0.84로 기준으로 하여 이보다 높은 성능의 모델을 얻도록 한다.

Learning rate는 너무 낮으면 학습이 느리고 너무 높으면 학습이 불안정해질 수 있다. Validation Loss가 증가하는 것을 방지하기 위해 약간의 값을 더 높여 설정한다. Batch size의 경우, overfitting을 방지하기 위해 크기를 줄여 더 자주 가중치를 업데이트할 수 있도록 한다. 또한 epoch 수를 감소시켜 overfitting을 방지할 수 있다. 너무 적은 epoch를 적용하면 underfitting 될 수 있으므로 적당히 조절해야한다. 또한 다양한 사전 훈련된 BERT 모델을 적용해서 좋은 결과값을 찾아야한다. bert-base-uncased를 제외한 다른 사전 훈련된 모델을 적용하여 더 좋은 결과를 얻을 수 있을지도 테스트해보았다.

B. 실험

learning rate = 1e-5, batch_size = 32, epochs = 10, pretrained_model = 'bert-base-uncased'

Epoch	Training Loss	Validation Accuracy	Validation Loss	Test Accuracy
1	0.57	0.80	0.48	N/A
2	0.40	0.83	0.41	N/A
3	0.29	0.84	0.42	N/A
4	0.21	0.84	0.46	N/A
5	0.17	0.84	0.59	N/A
6	0.13	0.84	0.60	N/A
7	0.11	0.82	0.71	N/A
8	0.10	0.85	0.67	N/A
9	0.08	0.83	0.73	N/A
10	0.07	0.84	0.71	0.84

learning rate = 2e-5, batch_size = 16, epochs = 10, pretrained_model = 'bert-base-uncased'

Epoch	Training Loss	Validation Accuracy	Validation Loss	Test Accuracy
1	0.49	0.82	0.41	N/A
2	0.27	0.83	0.48	N/A
3	0.19	0.85	0.50	N/A
4	0.12	0.85	0.71	N/A
5	0.09	0.84	0.88	N/A
6	0.06	0.84	0.92	N/A
7	0.03	0.83	1.03	N/A
8	0.03	0.84	1.10	N/A
9	0.02	0.84	1.17	N/A
10	0.01	0.84	1.19	0.82

learning rate = 2e-5, batch_size = 16, epochs = 5, pretrained_model = 'bert-base-uncased'

Epoch	Training Loss	Validation Accuracy	Validation Loss	Test Accuracy
1	0.49	0.82	0.41	N/A
2	0.27	0.83	0.48	N/A
3	0.17	0.85	0.59	N/A
4	0.12	0.86	0.69	N/A
5	0.08	0.86	0.77	0.83

learning rate = 3e-5, batch_size = 16, epochs = 5, pretrained_model = 'bert-base-uncased'

Epoch	Training Loss	Validation Accuracy	Validation Loss	Test Accuracy
1	0.49	0.83	0.39	N/A
2	0.28	0.83	0.48	N/A
3	0.17	0.85	0.61	N/A
4	0.10	0.84	0.68	N/A
5	0.05	0.84	0.84	0.84

learning rate = 2e-5, batch_size = 32, epochs = 5, pretrained_model = 'bert-base-uncased'

Epoch	Training Loss	Validation Accuracy	Validation Loss	Test Accuracy
1	0.50	0.80	0.47	N/A
2	0.30	0.83	0.42	N/A
3	0.19	0.84	0.49	N/A
4	0.13	0.84	0.56	N/A
5	0.09	0.85	0.65	0.83

learning rate = 3e-5, batch_size = 16, epochs = 5, pretrained_model = 'bert-large-uncased'

Epoch	Training Loss	Validation Accuracy	Validation Loss	Test Accuracy
1	0.48	0.82	0.39	N/A
2	0.29	0.84	0.49	N/A
3	0.19	0.83	0.66	N/A
4	0.13	0.84	0.66	N/A
5	0.09	0.85	0.75	0.84

learning rate = 5e-5, batch_size = 16, epochs = 5, pretrained_model = 'bert-base-uncased'

Epoch	Training Loss	Validation Accuracy	Validation Loss	Test Accuracy
1	0.51	0.79	0.44	N/A
2	0.29	0.81	0.48	N/A
3	0.15	0.84	0.69	N/A
4	0.09	0.84	0.84	N/A
5	0.04	0.84	0.96	0.82

learning rate = 3e-5, batch_size = 16, epochs = 5, pretrained_model = 'roberta-base'

Epoch	Training Loss	Validation Accuracy	Validation Loss	Test Accuracy
1	0.53	0.82	0.43	N/A
2	0.35	0.84	0.53	N/A
3	0.24	0.85	0.42	N/A
4	0.18	0.84	0.68	N/A
5	0.12	0.86	0.76	0.83

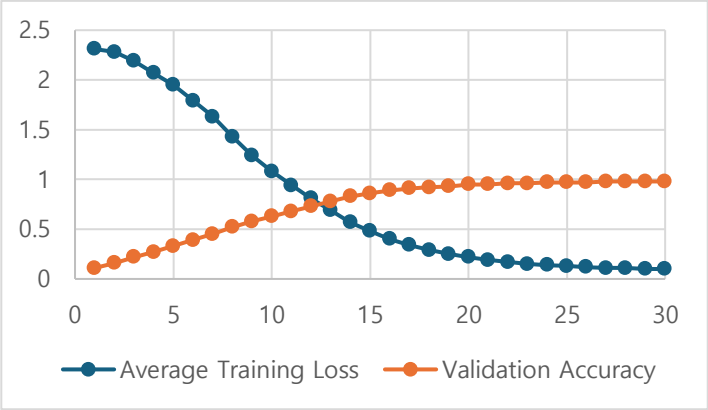
C. 분석

하이퍼파라미터를 바꾸어가며 여러 경우를 테스트 해보았지만 초기값보다 높은 Test Accuracy를 얻지 못했다. 하지만 위 실험 결과를 통해 Learning Rate를 크게 할수록 초기 학습 단계에서 손실이 더 빠르게 줄일 수 있음을 알 수 있었다. 또한 epoch가 커질수록 Training Loss가 계속 줄어들어 모델이 잘 학습하고 있음을 알았다. 모든 경우에서 Validation Loss가 크게 증가하여 Overfitting의 위험은 존재했고 이에 대해 더 좋은 하이퍼파라미터 세팅은 얻지 못했다.

2. Transformer

A. 가설 설정

hidden_state_dimension = 20, batch_size = 20, num_multi_heads = 2, max_epoch = 30, max_grad = 5.0, learning_rate = 1e-4, num_layers = 1으로 하이퍼파라미터를 설정하면 Test Accuracy 값이 0.98이 나온다. Average Training Loss와 Validation Accuracy를 epoch에 따라 정리한 그래프는 다음과 같다.



위 그래프와 같이 Average Training Loss는 감소하고 Validation Accuracy는 계속 증가했다. 이는 모델이 훈련 데이터에 대해 점차 학습을 잘하고 있으며 validation set에 대해서도 더 잘 예측함을 뜻한다. 모델이 overfitting 없이 데이터에 대한 학습을 잘하고 있음을 알 수 있고 Test Accuracy 값 역시 매우 높은 0.98이 도출되었다. 이미 높은 성능을 보이고 있지만 더 높이기 위해 하이퍼파라미터의 조정을 고려해야한다. hidden_state_dimension을 높여 더 많은 정보를 학습할 수 있도록 하고 batch size를 늘려 더 빠른 학습이 되도록 조절해본다. multi head 수를 늘려 입력 시퀀스의 다양한 부분에 집중할 수 있도록 하고 epoch도 증가시켜 충분한 시간동안 학습을 시킨다. learning rate는 너무 높은 값은 불안정해지고 너무 낮은 값은 학습 속도가 느려지므로 1e-3, 1e-5 두 가지 경우를 테스트하여 어느 경우가 더 좋은지 확인한다. gradient clipping은 크게 설정할 경우 거의 모든 그래디언트 값이 clipping 되지 않으므로 더 작은 값으로 실험했다. layer 수도 키워서 모델이 더 복잡한 패턴을 학습할 수 있도록 했다.

B. 실험

(hidden_state_dimension, batch_size, num_multi_heads, max_epoch, max_grad, learning_rate, num_layers 순서)

i. (20, 20, 2, 30, 5.0, 1e-3, 1)으로 설정한 경우

learning rate가 1e-4에 비해 훨씬 빠르게 수렴했으며 Test Accuracy는 1.00이 도출되었다. epoch에 대해 속도가 빨라 overfitting이나 불안정한 학습이 가능하다.

ii. (20, 20, 2, 30, 5.0, 1e-5, 1)으로 설정한 경우

Test Accuracy가 0.14가 도출되었고 Training Loss의 감소 속도가 매우 느리

며 2.28까지밖에 학습하지 못했다. learning rate가 너무 느려 underfitting이 심하여 좋은 모델이 아니다.

iii. (64, 20, 2, 30, 5.0, 1e-4, 1)

Training Loss의 감소 속도가 초기값보다 빠르며 6번째 epoch 이후 급격히 낮아져 0에 가까워진다. 7번째 epoch부터 Validation Accuracy가 1.00이 되었고 Test Accuracy가 1.00이 되었다.

iv. (20, 64, 2, 30, 5.0,1e-4, 1)

Training Loss의 감소 속도가 초기값과 같으며 걸린 시간이 크게 줄었다. Validation Accuracy와 Test Accuracy도 초기값과 같은 값이 도출되었다.

v. (20, 20, 4, 30, 5.0, 1e-4, 1)

Test Accuracy가 0.89가 도출되었고 초기값보다 학습 속도가 느렸다.

vi. (20, 20, 4, 50, 5.0, 1e-4, 1)

위 실험에서 epoch를 증가시켰는데 Training Loss는 0에 도달했고 Validation Accuracy와 Test Accuracy는 1.00에 도달했다.

vii. (20, 20, 2, 30, 5.0, 1e-4, 2)

초기값에 비해 Training Loss가 큰 폭으로 감소하고 Validation Accuracy도 큰 폭으로 증가해 4번째 epoch에서 1.00이 됐다. Test Accuracy도 1.0이 도출되어 overfitting의 가능성이 있다.

viii. (20, 20, 2, 30, 1.0, 1e-4, 1)

초기값과 같았다.

ix. (64, 64, 4, 50, 1.0, 1e-4, 2)으로 설정한 경우

Training Loss가 꾸준히 감소하고 Validation Accuracy는 1.00으로 일정하게 유지되었다. Test Accuracy는 1.00이 도출되었다. 데이터가 매우 단순하거나 모델이 overfitting 되었다고 판단할 수 있다.

C. 분석

hidden state dimension를 크게 하면 더 복잡한 패턴 등을 학습할 수 있지만 Validation Accuracy가 매우 급격하게 커지는 것을 보아 overfitting 될 수 있다. batch size를 키우면 메모리 사용량이 증가하고 epoch당 걸리는 시간이 크게 줄어들었다. multi-heads 수를 키우면 다양한 각도에서 패턴을 학습할 수 있지만 계산 비용이 증가해 학습 속도가 떨어졌다. 이 경우 epoch를 50으로 늘려 실험해보니 더 학습되어 정확도가 1.00에 도달해 overfitting 되는 경향이 있었다. learning rate가 1e-3의 경우 속도가 너무 빨라 불안정학 학습을 보였고 1e-5의 경우 너무 느려 underfitting 경향을 보였다. number of layers의 경우, 더 복잡한 패턴을 학습할 수 있지만 더 많은 파라미터를 가지면서 데이터에 overfitting 되는 경향을 보였다. max gradient로 clipping이 잘 되도록 작게 설정했지만 초기값과 차이가 없었다.