[딥러닝분석 기말 프로젝트]

20192940 강준규

**[Way 1] Brute Force**

1. **제시하는 문제를 선형회귀, 분류, CNN, RNN 모델 순으로 별다른 전처리 없이 해결하는 방법이다. Nan 값들에 대해서만 0으로 바꿔주는 간단한 함수를 적용(아래 그림 참조) 후 바로 모델 설계에 돌입하였다.**

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

1. **선형 회귀 모델**
   1. 많은 실험을 해야 하는 관계로 Optimizer는 SGD로, Loss는 MSE로 고정한 후 실험을 진행하였다. Epoch 역시 1000번으로 고정하였다.
   2. MSE 값은 아래 그림과 같이 Nan이 나왔다.



1. **분류 모델**
   1. 활성 함수는 Sigmoid로, Optimizer는 SGD로, Epoch는 1000번으로 고정하였다.
   2. Test Acc.의 값은 다음과 같다.



1. **CNN 모델**
   1. 활성 함수는 ReLu로, Optimizer는 Adam으로, Epoch와 Batch Size는 50과 32로 고정하였다.
   2. Test Acc.의 값은 다음과 같다.



1. RNN 모델
   1. 활성 함수는 ReLu와 Sigmoid로, Optimizer는 Adam으로, Epoch와 Batch Size는 50과 32로 고정하였다.
   2. Test Acc.의 값은 다음과 같다.



1. **3-5 모델의 AUC Score는 다음과 같다.**

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

1. **Visualization**
   1. 3-5 모델의 정확도와 손실 그래프이다.

텍스트, 도표, 그래프, 라인이(가) 표시된 사진

자동 생성된 설명

**[Way 2] SNP Selection**

1. **데이터 전처리 과정에서의 SNP를 Variance가 높은 순으로 100개를 선택했으며, 그 이외의 모든 실험 파라미터는 고정으로 하여 비교 실험을 하였다.**

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

1. **선형 회귀 모델**
   1. MSE 값은 아래 그림과 같이 Nan이 나왔다.



1. **분류 모델**
   1. Test Acc.의 값은 다음과 같다.



1. **CNN 모델**
   1. Test Acc.의 값은 다음과 같다.



1. **RNN 모델**
   1. Test Acc.의 값은 다음과 같다.



1. **3-5 모델의 AUC Score는 다음과 같다.**

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

1. **Visualization**
   1. 3-5 모델의 정확도와 손실 그래프이다.

텍스트, 도표, 라인, 그래프이(가) 표시된 사진

자동 생성된 설명

1. **이전 방식(Way 1)과의 차이점**
   1. 입력 데이터가 정갈하지 못해 정갈하지 못했던 정확도와 손실 그래프가 나왔던 (Way 1 – 7번 참고) 이전 방법에 비해, 해당 그래프들이 어느정도 형태를 갖추기 시작했다. 하지만 여전히 Test에서 손실 값이 수렴하지 못하고 치솟는 경향을 보아 아직은 많은 시도가 필요한 듯하다.

**[Way 3] Standardizing - Scaling**

1. **문제점 착안**
   1. 이전 두 방법에서 선형 회귀 모델의 MSE 값이 계속 Nan으로 나왔다. 아래 두 그림처럼 처음부터 손실 값들이 치솟는 경향이 확인되었다.

텍스트, 스크린샷, 블랙, 대수학이(가) 표시된 사진

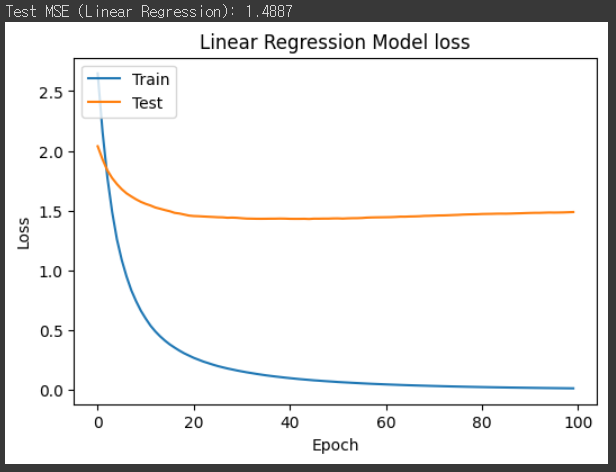
자동 생성된 설명

텍스트, 스크린샷, 디스플레이, 라인이(가) 표시된 사진

자동 생성된 설명

* 1. 이에 문제를 인지하고, 데이터를 표준화하는 전처리를 추가하기로 결정하였다. 데이터를 스케일링 하면 학습의 안정성과 성능을 향상시킬 수 있다.

1. **선형 회귀 모델**
   1. MSE 값은 아래 그림과 같다. 이전 두 방법과 다르게 손실 값이 어느정도 안정화되기 시작했다.



1. **분류 모델**
   1. Test Acc.의 값은 다음과 같다.



1. **CNN 모델**
   1. Test Acc.의 값은 다음과 같다.



1. **RNN 모델**
   1. Test Acc.의 값은 다음과 같다.



1. **3-5 모델의 AUC Score는 다음과 같다.**

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

1. **Visualization**
   1. 3-5 모델의 정확도와 손실 그래프이다.

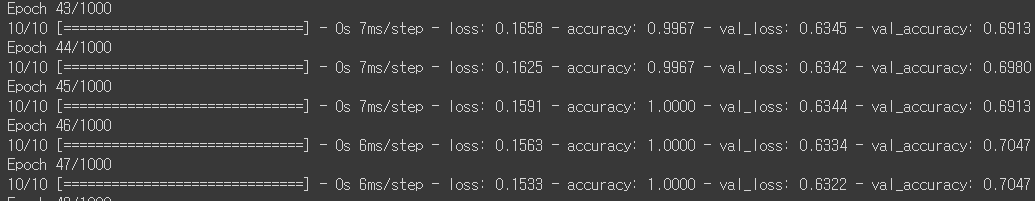
텍스트, 도표, 라인, 그래프이(가) 표시된 사진

자동 생성된 설명

1. **이전 방식(Way 1-2)과의 차이점**
   1. 겉보기에는 수치나 그래프에 큰 차이가 없어 보인다. 오히려 RNN의 정확도는 이전 방식보다 낮아졌다(그러나 RNN의 AUC Score는 이전보다 상승하였다). 하지만 이번 방식의 목적인, 선형 회귀 모델에서의 발산하는 손실 값을 잡아내는데 성공하였다. 현재 새로 발견된 문제는 모델이 학습 단계에서 발생시키는 과적합이다.

**[Way 4] Regularization – Dropout, Early Stopping**

1. **문제점 착안**
   1. Way 3에서 모델이 학습단계에서 너무 일찍 정확도를 1.0을 만들고 이후 과적합 문제가 발생하는 것을 확인하였다.



(45번째 Epoch부터 정확도가 1.0에 들어서고 이후 학습이 계속되는 모습)

* 1. 이에 Dropout과 Early Stopping 같은 Regularization 방법들을 활용하여 문제를 해결해보고자 한다.
  2. 본 실험부터는 정확도를 측정하지 않는 선형 회귀 모델을 제외하였다. 또한, 많은 실험을 관계로 RNN 모델도 실험에서 제외하였다. Dropout 확률은 0.5로 고정하였고, Early Stopping에는 특별한 파라미터를 주지 않은 채 실험을 진행하였다.

1. **분류 모델**
   1. Epoch 12 만에 학습을 완료하였으며, Test ACC.의 값은 다음과 같다.





1. **CNN 모델**
   1. Epoch 21 만에 학습을 완료하였으며, Test Acc.의 값은 다음과 같다.





1. 2-3 모델의 AUC Score는 다음과 같다.

텍스트, 폰트, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

1. **Visualization**
   1. 2-3 모델의 정확도와 손실 그래프이다.

텍스트, 도표, 그래프, 라인이(가) 표시된 사진

자동 생성된 설명

1. **이전 방식(Way 3)과의 차이점**
   1. 손실 그래프를 보면 알 수 있듯, 학습이 많이 안정화되었다. Test 단계에서도 과적합 문제를 일으키지 않고 적절하게 낮은 방향으로 수렴하는 것을 확인할 수 있다.

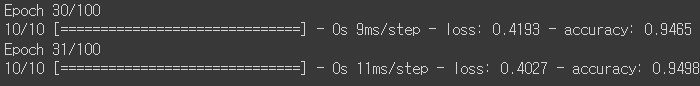
**[Way 5] L2 Regularization**

1. **문제점 착안**
   1. Way 4에서 보다 학습 안정성을 부여하기 위해 모델 학습 단계에서 L2 정규화를 사용하기로 하였다.
   2. 본 실험 역시 분류 모델과 CNN 모델에서 실험을 진행하였다.
2. **분류 모델**
   1. 모델의 설계는 다음과 같다(L2 정규화 추가).

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

* 1. Epoch 31 만에 학습을 완료하였으며, Test ACC.의 값은 다음과 같다.



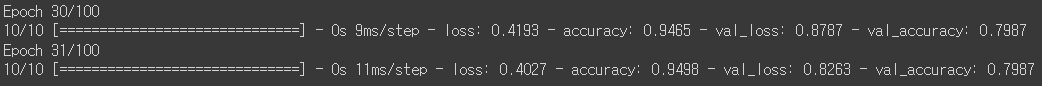


1. **CNN 모델**
   1. 모델의 설계는 다음과 같다(L2 정규화 추가).

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

* 1. Epoch 31 만에 학습을 완료하였으며, Test Acc.의 값은 다음과 같다.





1. **2-3 모델의 AUC Score는 다음과 같다.**

텍스트, 폰트, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

1. **Visualization**
   1. 2-3 모델의 정확도와 손실 그래프이다.

텍스트, 도표, 그래프, 라인이(가) 표시된 사진

자동 생성된 설명

1. 이전 방식(Way 4)과의 차이점
   1. 분류 모델의 AUC Score가 확연히 높아지는 성능을 보였다. 분류 모델은 AUC Score가 0.55에서 0.70으로, CNN 모델은 0.65에서 0.65로 변화한 것을 확인할 수 있다.
   2. 그 외에는 크게 변화한 지점을 확인할 수 없었다.

**[Way 6] Standardization + SNP Selection + Regularization**

1. **실험의 마무리로, 이전의 모든 방법을 모두 동원하여 선형 회귀부터 RNN까지 모든 모델의 성능을 확인하기로 하였다.**
2. **선형 회귀 모델**
   1. MSE와 손실 값은 다음과 같다. MSE 값도 이전의 방식(Way 3)보다 낮아졌으며, 손실은 Test에서 이전 방식(Way 3)보다 더 수려하게 하강하는 것을 확인할 수 있다.

텍스트, 스크린샷, 디스플레이, 라인이(가) 표시된 사진

자동 생성된 설명

1. **분류 모델**
   1. Epoch 92 만에 학습을 완료하였으며, Test Acc.의 값은 다음과 같다.





1. **CNN 모델**
   1. Epoch 46 만에 학습을 완료하였으며, Test Acc.의 값은 다음과 같다.

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명



1. **RNN 모델**
   1. Early Stopping이 발생하지 않았으며, Test Acc.의 값은 다음과 같다.



1. **3-5 모델의 AUC Score는 다음과 같다.**

텍스트, 폰트, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

1. **Visualization**
   1. 3-5 모델의 정확도와 손실 그래프이다.

텍스트, 도표, 라인, 그래프이(가) 표시된 사진

자동 생성된 설명

1. **이전 방식(Way 1-5)과의 차이점**
   1. 선형 회귀 모델에서의 손실 값은 이전의 방식들보다 확연하게 안정적인 그래프를 보였다.
   2. 분류 모델과 CNN 모델은 적절히 Early Stopping 이 동작하였지만, 학습의 성능에는 크게 영향을 주지 않았다.
   3. RNN의 경우에는 Dropout을 사용하였을 때 실험의 진행 시간이 확연히 줄어들었다. 하지만 Early Stopping은 작동하지 않았다. Dropout의 확률을 너무 크게 준 탓인지, 가시화된 그래프에서 확인할 수 있듯이 Test 데이터의 정확도와 손실 값이 너무나도 이상적으로 찍혔다. 본 연구자는 이에 RNN 방식에서 데이터를 정규화하는 과정이 유효하게 작용하지 않았다고 결론지었다.

**[Conclusion]**

1. RNN을 제외한 나머지 모델에서, 학습의 안정화를 위해 하나씩 정규화 방식을 채택할 때마다 정확도와 손실 값에 대한 그래프는 정갈하게 변화하는 것을 알 수 있었다.
2. RNN의 경우에는 너무 이상적인 값이 도출되어 실험에 문제가 있다고 결론 지었으며, 이는 RNN과 정규화 방식이 맞지 않거나 혹은, 본 연구자의 미숙한 설계로 해당 결과가 나왔다고 판단하고 있다.
3. 모든 방식에서 Test Acc.의 값은 극적으로 변화하지 않았다. 따라서, 여러 정규화 방식과 과제의 명세에서 제시한 SNP를 일부 채택하는 방식은 모델의 정확도를 올리는데 효과적이기 보다, 학습 단계에서의 안정성을 부여하는데 큰 역할을 했음을 확인했다.
4. 시각화된 그래프로 보아 학습의 안정성은 잘 통제하였다. 그러나 결국 실험의 정확도와 AUC Score가 크게 차이가 나지 않고 성공적이지 못한 값으로 실험이 마무리된 것으로 보아, 주어진 데이터의 수치가 고르지 못하고 한쪽으로 크게 편향되었다는 사실을 유추해볼 수 있었다.