**데이터와 패턴분류 보고서**

**이름 : 이 준 희**

**1. 과제 수행 내용**

주어진 데이터셋에는 불필요한 속성이 있다고 생각하였다. 학생 성별, 보호자 성별, 학부모 설문 같은 속성의 경우는 학점과는 전혀 연관이 없어 보였지만 혹시 모르니 데이터의 값을 면밀히 분석해보기로 하였다.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

데이터셋을 분석할 때 학점을 기준으로 속성의 평균값을 계산하였다. 높은 점수를 받은 학생의 속성과 중간 점수를 받은 학생의 속성의 차(1)와 중간 점수를 받은 학생의 속성과 낮은 점수를 받은 학생의 속성의 차(2)를 계산하여 (1)과 (2)가 양수가 나와야 클래스 간의 차이가 존재한다는 것을 알 수 있다. 또한 각 클래스간의 차이가 명확하게 보여야 학습을 하였을 때 정답값을 더 잘 나타낼 수 있다고 생각하였다. 그러하여 3, 4, 5, 6번의 속성을 이용하여 신경망을 구성해야겠다고 생각하였다.

하지만 내가 잘못 생각했을 수도 있어 각 속성에 대한 학점의 분포도를 spyder를 통하여 나타내 보았다.

도표, 평면도, 지도, 스크린샷이(가) 표시된 사진

자동 생성된 설명

속성의 값이 0과 1로 이루어진 데이터는 1, 2, 7, 8, 9번이고, 1번과 2번은 학점의 관계 없이 무분별하게 분포되어 있는 것을 볼 수 있었고 그와 반해 7, 8, 9번의 데이터는 학점과 관계가 있어 보이지만 중복된 클래스가 많아 정답을 찾기에 부족한 데이터로 보인다. 하지만 3, 4, 5, 6번 속성의 데이터는 학점이 낮은 학생, 중간인 학생, 높은 학생이 구분되어 있는 데이터로 보인다. 이러한 이유로 3, 4, 5, 6번의 속성을 이용하여 데이터를 학습시키려고 하였다.

이 속성으로 신경망을 적용하기 위해 X = dataset[:,[2,3,5,6]].astype(float)를 하여 3, 4, 5, 6번의 속성을 X에 넣었고 Y\_obj = dataset[:,4]를 하여 Y\_obj에 정답 클래스를 넣었다. 또한, X/100을 하여 X의 범위를 0~1 사이의 값을 갖도록 구성하여 정규화를 하였다. 정규화를 한 이유는 경사하강을 하였을 때, 더 빨리 최소점을 찾을 수 있고 정확도가 튀지 않기에 사용하였다.

신경망의 구성은 모델이 학점을 예측하는 것이므로 학점 클래스의 H, M, L의 예측값을 출력하도록 출력층을 3개의 노드로 구성하였고 출력의 활성화 함수는 softmax함수를 사용하여 대소관계가 바뀌지 않으면서 편향되어 있는 데이터를 높은 확률로 맞출 수 있게 설정하였다. 입력층에서는 X\_train의 4개의 속성을 입력하게 하였다. 또한, 출력층에서의 활성화 함수를 제외한 모든 은닉층의 활성화 함수는 Relu 함수로 설정하였는데 그 이유는 역전파를 하였을 때, 기울기 소실의 문제를 극복하기 위하여 사용하였다. 노드의 갯수와 은닉층의 갯수를 설정하는 것은 많은 시도 끝에 과제에 제시된 테스트셋에서의 성능이 74%가 넘기는 것을 목표로 임의대로 설정하였다. 모델의 손실함수는 MSE를 사용하였고, 최적화 함수는 adam을 사용하였다.

**2. 신경망 학습 결과**

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 그래프, 도표이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

모델의 신경망이 이렇게 구성되어 있을 때, 학습 데이터의 정확도는 76.94%가 나왔고, 테스트셋의 정확도는 75.83%가 나왔다.

텍스트, 스크린샷, 그래프, 라인이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

다음 그림은 모든 속성을 넣었을 때의 학습 데이터의 정확도와 테스트셋의 정확도이다. 학습 데이터의 정확도는 84.72%로 증가하였지만 테스트셋의 정확도는 75%로 감소한 것을 볼 수 있다. 속성의 값이 많이 포함되어 있을 수록 좋지만 정답값을 예측하기에 불필요한 속성이 있고 모델이 그것을 학습하므로 이러한 결과가 나왔을 것을 예측하였다.

신경망을 model.add(Dense(32, input\_dim=X\_train.shape[1], activation='relu'))로 유지하고 노드가 10개 있는 은닉층이 추가될 때마다 학습 데이터의 정확도와 테스트셋의 정확도가 어떻게 변하는지 궁금하여 시각적으로 확인할 수 있는 그래프를 그려보았다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 그래프, 라인이(가) 표시된 사진

자동 생성된 설명

X축은 은닉층 추가 갯수이고, Y축은 정확도이다.

텍스트, 스크린샷, 책, 메뉴이(가) 표시된 사진

자동 생성된 설명

이 과제에서 성능을 도출할 때, 테스트셋의 정확도가 학습 데이터의 정확도보다 낮거나 일의 자리가 같을 때를 가정하여 테스트셋에서의 정확도가 높은 것을 도출하고 만약 정확도가 같다면 은닉층의 갯수가 적은 것을 좋은 성능으로 뽑는다. 이 모델은 17개의 추가된 은닉층을 가진 학습 데이터 정확도: 72.78%, 테스트셋 정확도: 71.67%의 성능을 가진다.

이 때, 속성 3번 대신 9번을 넣어 보았다. 속성 9번은 100으로 나누어진 값으로 X에 저장되었다.

텍스트, 스크린샷, 그래프, 라인이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 책, 패턴이(가) 표시된 사진

자동 생성된 설명

이 모델의 추가된 은닉층은 19개이고 학습 데이터 정확도: 78.61%, 테스트셋의 정확도: 73.33%의 성능을 가지게 되고 그 전의 속성을 학습한 모델보다 좋은 성능을 지녔다.

텍스트, 스크린샷, 그래프, 도표이(가) 표시된 사진

자동 생성된 설명

(속성 9) / 100을 하지 않은 은닉층 갯수에 따른 정확도

텍스트, 스크린샷, 메뉴, 책이(가) 표시된 사진

자동 생성된 설명

이 모델의 추가된 은닉층은 17번째이고, 학습 데이터의 정확도: 77.22%, 테스트셋의 정확도: 77.5%로 속성 9의 값을 그대로 X에 저장하였을 때는 성능이 더 좋게 나왔다.

텍스트, 스크린샷, 그래프, 도표이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

결론적으로 속성을 4, 5, 6, 9번의 조합으로 입력층의 노드의 갯수 4개와 은닉층 18개, 출력층 노드의 갯수 3개의 모델을 만들어 테스트셋의 정확도를 74.17%를 달성하였다.

속성 9번이 속성 2번보다 좋은 정확도를 가지게 된 이유에 대해서는 L 클래스와 H 클래스를 확실하게 나눠주기 때문에 더 좋은 속성이라고 보인다. 또한 속성 9를 100으로 나누는 것보다 속성 9의 정확도가 좋은 이유는 속성 9는 이미 0~1 사이의 값이므로 100을 나누는 것 보다 좋은 성능을 보여주는 것으로 보인다.

훈련 데이터의 정확도와 데이터셋의 정확도가 유사한 이유는 훈련 데이터와 테스트셋의 데이터가 그만큼 닮았기 때문에 은닉층이 많아졌을 때에도 테스트셋의 정확도가 낮아지지 않고 훈련 데이터와 유사하게 나온 것으로 보인다.

**3. 고찰**

은닉층의 갯수에 따라 정확도가 어떻게 변하는지를 수행하면서 많은 것을 알게 되었다. 은닉층이 많아지면 과적합이 일어날 줄 알았지만 일어나지 않았다. 과적합은 훈련 데이터에 대한 성능은 좋지만 테스트셋에 대한 성능은 안 좋은 결과를 보여줘야 하는데 훈련 데이터의 정확도가 낮아지면 테스트셋의 정확도도 낮아지고 훈련 데이터의 정확도가 높아지면 테스트셋의 정확도도 같이 높아지는 것을 보면서 훈련 데이터와 테스트셋의 데이터가 명확한 속성을 갖고 있다면(상관관계가 있는 데이터를 가지고 있다면) 과적합이 일어나지 않을 수 있겠다고 생각이 되었다. 또, 은닉층이 많아지면서 정확도가 떨어졌다가 다시 높아지는 것을 볼 수 있는데, 은닉층의 갯수에 따라 기울기 손실로 인하여 정확도가 낮아지는 것을 일반적으로 알고 있었지만 다시 높아지는 것은 무엇 때문인지는 검색해도 나오지 않아 알 수 없었지만 이런 현상이 있을 수 있다는 것을 알게 되었다.

텍스트, 스크린샷, 그래프, 라인이(가) 표시된 사진

자동 생성된 설명

4, 5, 6, 9번의 속성이 1개 ~ 101개(32개의 노드 + 20개의 노드)의 은닉층을 가질 때의 정확도

그래프는 40개의 추가 은닉층을 가질 때 부터 일정한 값을 가지게 되는 것과, 훈련 데이터의 정확도와 테스트셋의 정확도가 매우 유사하다는 것을 볼 수 있고, 정확도가 들쑥날쑥하다는 3가지의 특징이 있다. 첫번째 특징으로 알 수 있는 것은 40번 이후로 가중치의 값이 변하지 않았다는 것이고 이는 기울기 소실이 일어났다는 것을 알 수 있다. 두번째 특징으로 알 수 있는 것은 속성의 조합이 정답과 상관관계가 있다는 것을 알 수 있다. 마지막 특징은 알 수 없었다.

과제를 수행하면서 은닉층의 노드의 갯수를 설정하는 것이 어려웠다. 이를 명확하게 설명하는 자료는 없었고 많은 시도를 해야한다는 자료만이 존재하여 임의로 설정하는데 시간이 많이 걸렸다. 시간이 더 주어졌다면 더 효율적인 노드의 갯수를 설정하고 은닉층의 갯수를 정할 수 있었겠다는 아쉬움이 남는다. 또한, 은닉층 갯수에 따른 정확도 그래프에서 17번째가 최고의 성능을 보였는데, 따로 17개의 은닉층을 추가하고 정확도를 살펴봤을 때 정확도가 다르게 나왔다. 이 또한 시간이 더 주어졌다면 왜 이런 결과가 나왔는지 분석하고 싶은 마음이 있다. 그리고 초반에 생각한 속성보다 후에 9번의 속성으로 대체하는 것이 더 좋은 성능을 보여 놀랐던 기억이 있다. 조금 더 신중하게 분석해야겠다는 생각도 들었다.