머신러닝실습 레포트

-텀프로젝트-

제출일 : 2022.06.14.

학 번 : 2017142012

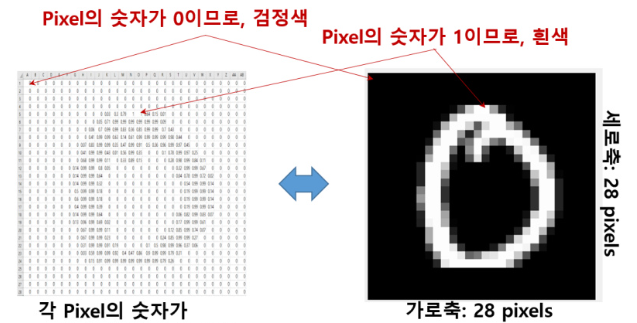
이 름 : 김효건

# **1. 프로젝트 개요**

**1. 이미지 데이터 셋(csv파일)을 통한 이미지 분류**

Csv 데이터 파일은 28x28크기의 csv 파일입니다 각각은 하나의 픽셀 값을 나타냅니다.

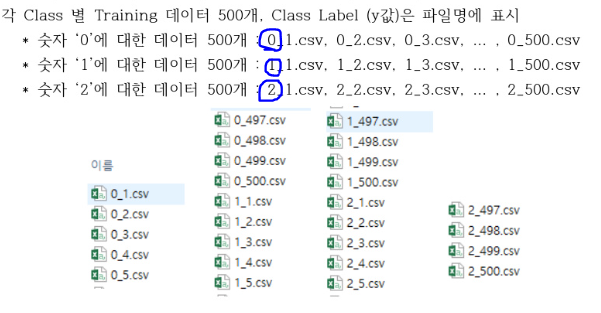
0에 가까울수록 검정, 1에 가까울수록 흰색을 나타내 아래와 같은 그림이 나타납니다.



해당 이미지들을 분류하기 위해 본 프로젝트에서 각 activation function은 sigmoid function을 사용했고 네트워크 층은 two-layer neural network라서 출력 층 1개, 은닉 층 1개, 입력 층 1개로 구성되어 있습니다. 또한 은닉 층 노드의 개수는 따로 지정해도 상관없으며 입력 층은 총 10개의 특성 중 5개의 특성을 골라 사용하며, 출력 노드의 개수는 (0, 1, 2) 3개를 구분해야 하므로 3개 입니다.

제공된 데이터 셋은 0,1,2 각각 500개의 데이터씩 총 1500개의 데이터를 입력받아 사용합니다.

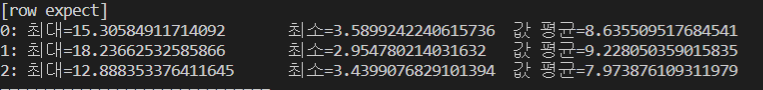
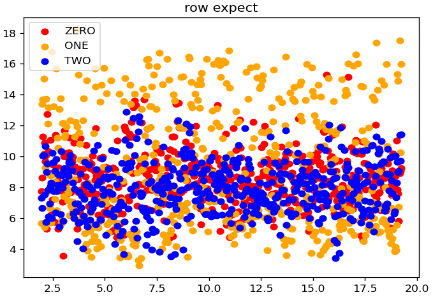
따라서 28x28인 데이터 셋은 총 10개중 5개의 데이터 특성을 찾아서 사용해야 하며, csv파일의 이름은 라벨로 정의할 수 있기 때문에 데이터를 받아올 때 가져오면서 one hot encoding을 사용하여 1500 x 3으로 만들어 줍니다.

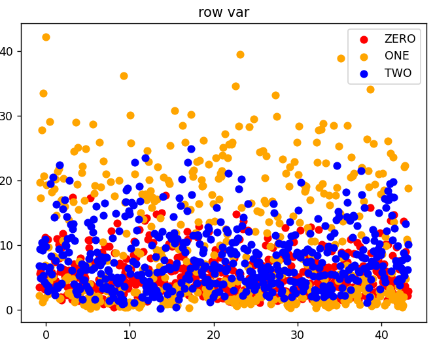


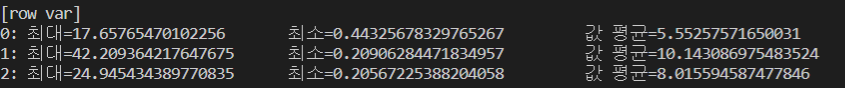
# **2. 해결 과정**

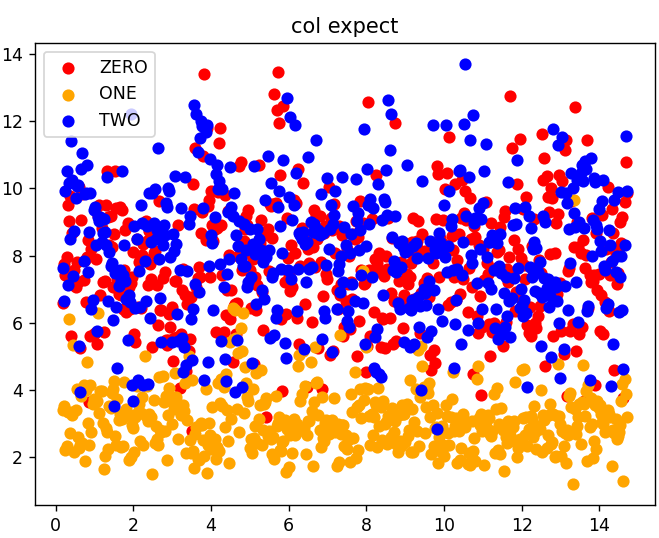
먼저 입력 받은 데이터 셋은 1500개이며 각각 28x28입니다. 학습 전 최적화해야 하는 하이퍼파라미터로는 **특성**, **은닉 층의 노드 수**, learning rate가 있습니다. 먼저 총 10개의 특성 중 5개를 선택하기 위해 데이터의 분포도 및 특성을 살펴보았습니다.

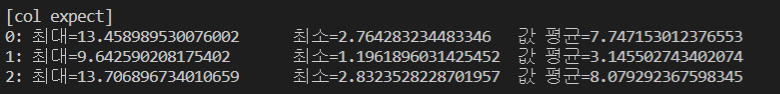
데이터 셋의 분포도와 평균과 최소 최대를 비교하여 선택하였습니다.

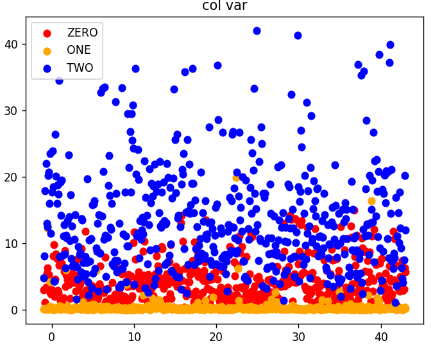


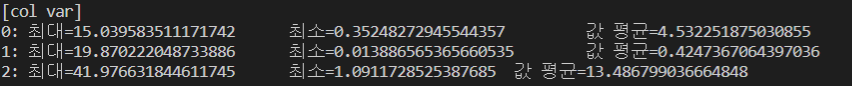




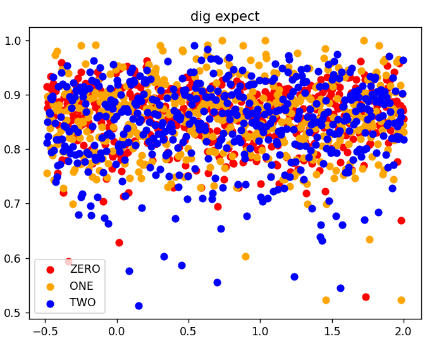
가로축 projection특성의 경우 기대치보다는 분산이 평균에 대한 차이가 더 커서 우선순위를 두었습니다.

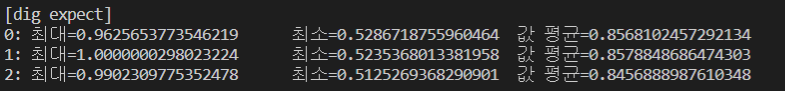


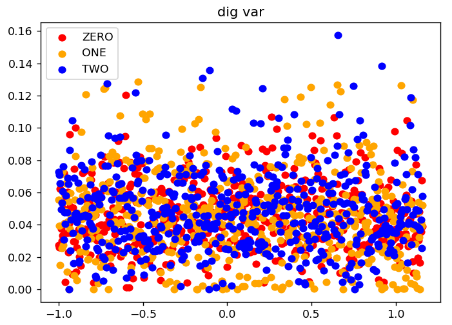


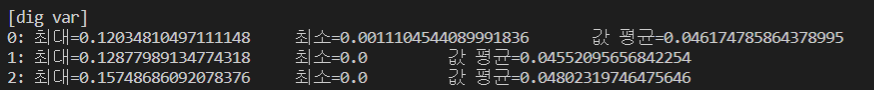


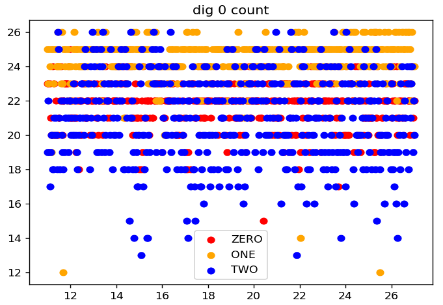
세로축 projection의 경우 그래프로 보기에 기대치의 경우 1을 분리하기 쉽고, 분산의 경우 2를 분리하기에 적합한 특성으로 보아 둘 다 사용하기 적합하다 생각했습니다.







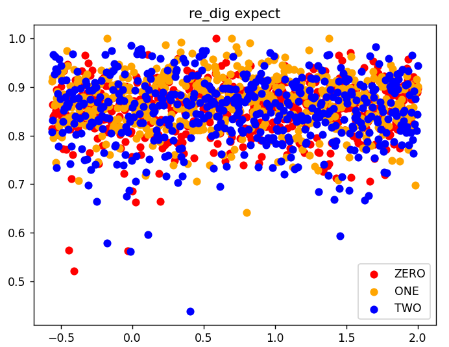


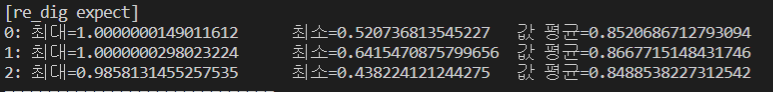


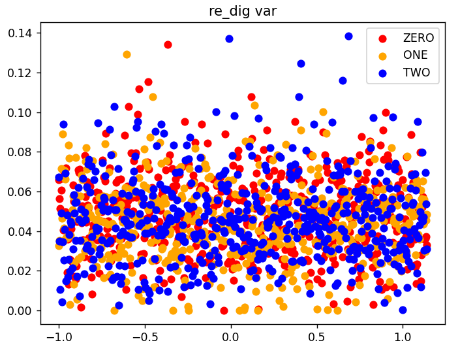
텍스트이(가) 표시된 사진

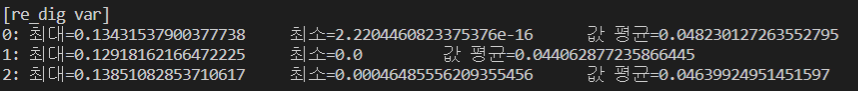
자동 생성된 설명

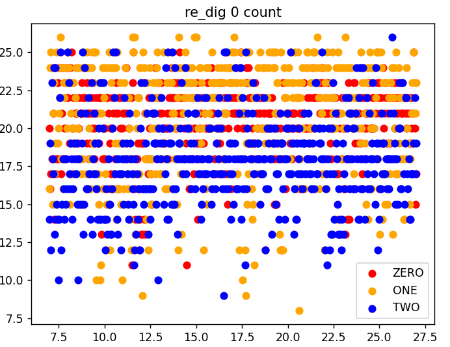
대각선에 대한 기대 값과 분산, 0의 개수를 출력한 결과 기대 값과 분산은 평균과 최소 최대가 구분하기 힘들지만 그나마 count 개수에서 차이가 보이므로 해당 특성을 사용할 수 있습니다

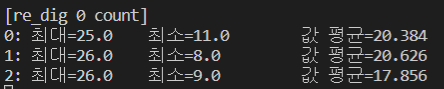










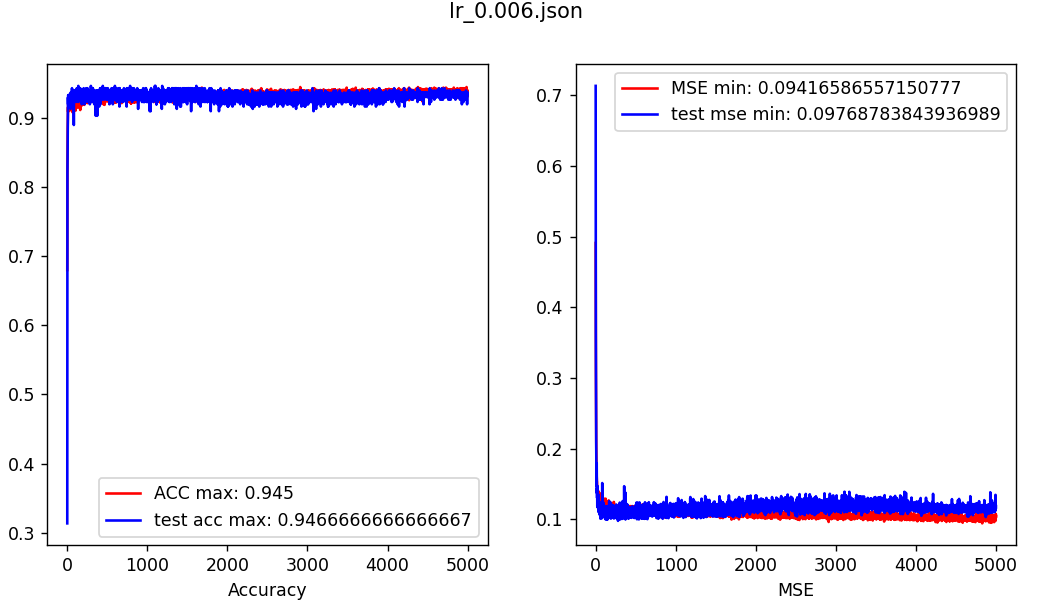


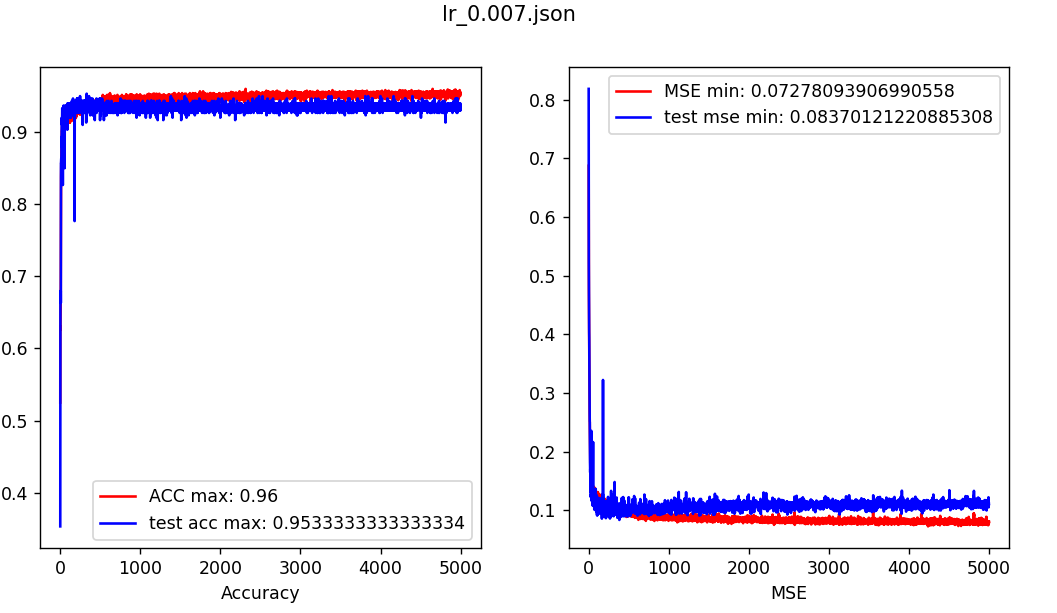
역 대각선 특성의 경우에도 기대 값이나 분산의 경우 별 차이가 없으나, 0의 개수는 차이가 조금은 있으므로 해당 특성 또한 활용할 수 있습니다.

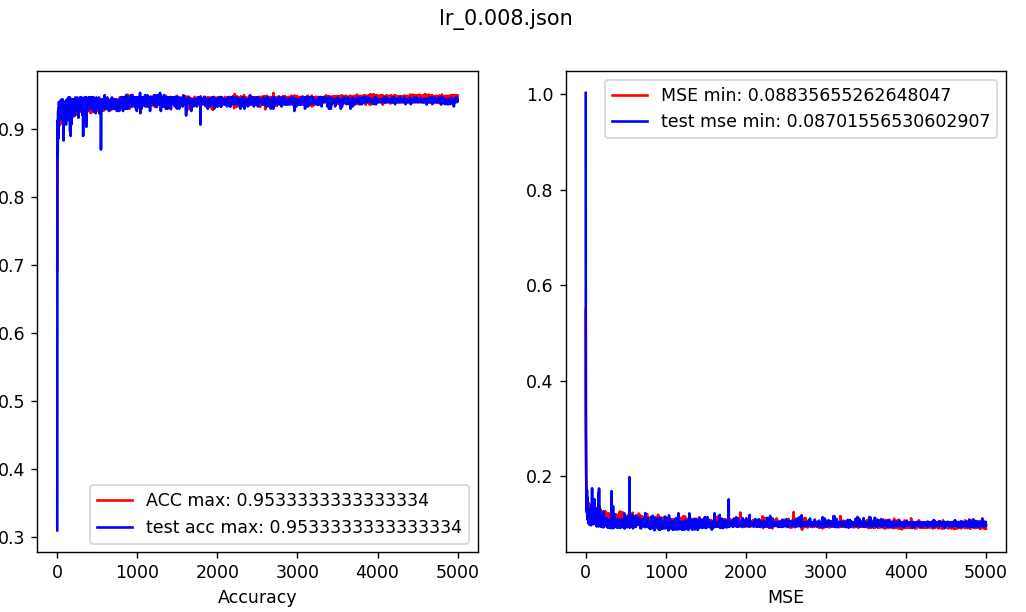
따라서 특성2(가로축 분산), 특성3(세로축 기대 값), 특성4(세로축 분산), 특성7(대각선 0 개수), 특성10(역 대각선 0 개수)를 선택하여 학습을 진행하였습니다. 또한 특성2, 3, 4를 보면 숫자1과 2를 쉽게 구별할 수 있겠다 예측이 가며, 숫자 0은 인식하기 힘들 거 같다고 생각됩니다.

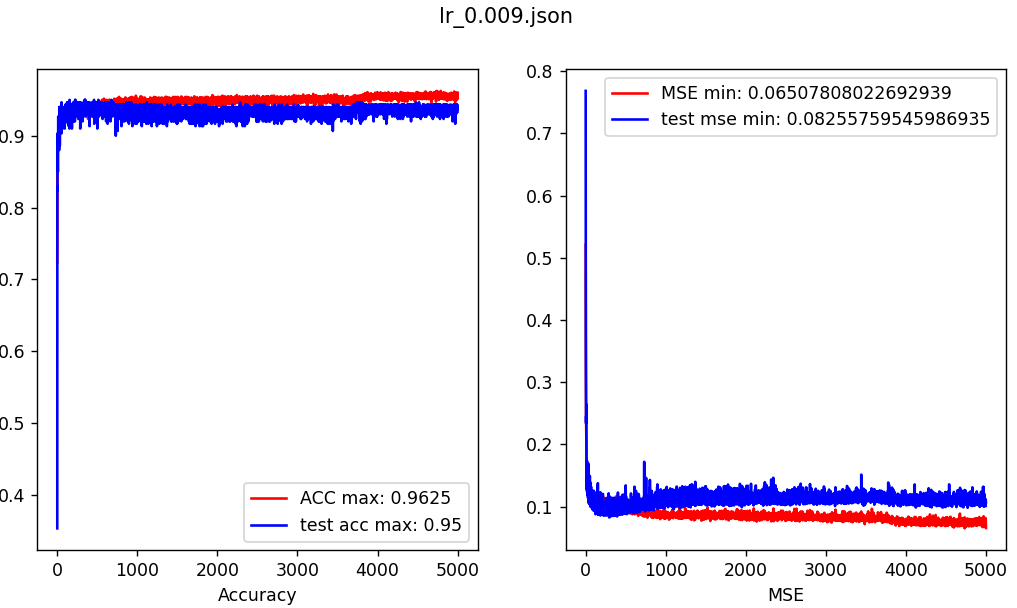
이제 예측을 위해서 learning rate와 은닉 층의 노드 수가 남았습니다. 저는 먼저 learning rate를 최적화 한 후 은닉 층 노드 수를 최적화 하겠습니다.

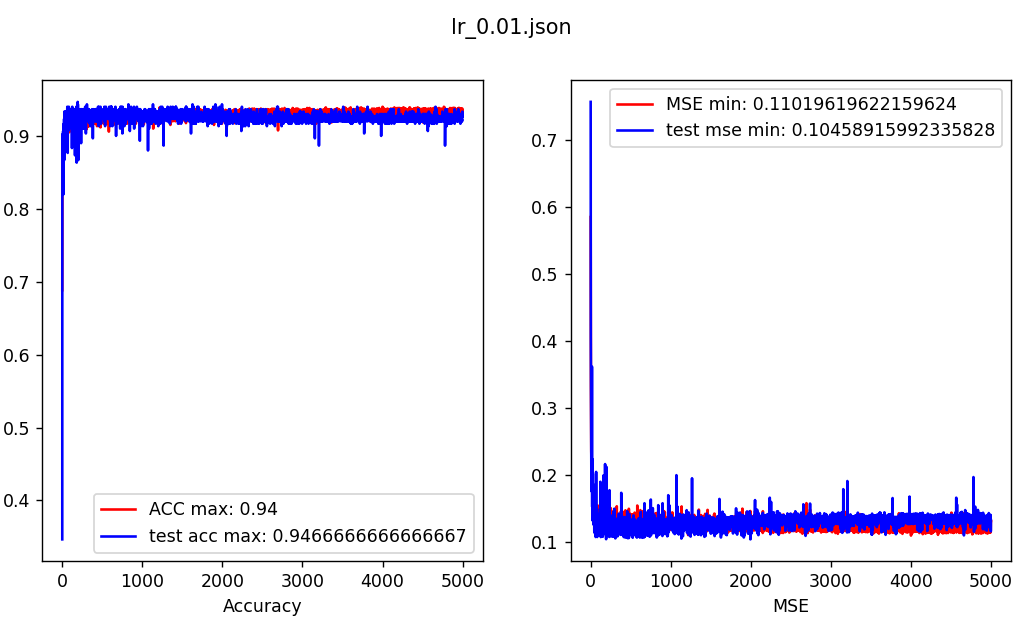
Learning rate의 경우 0.006 ~ 0.01, 총 5개를 실행하였습니다.





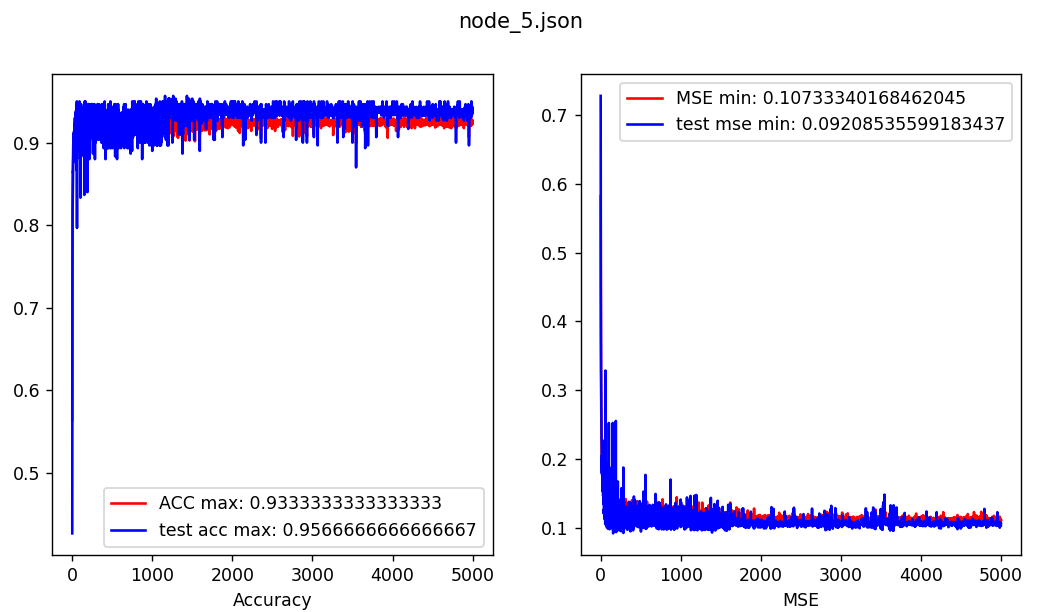


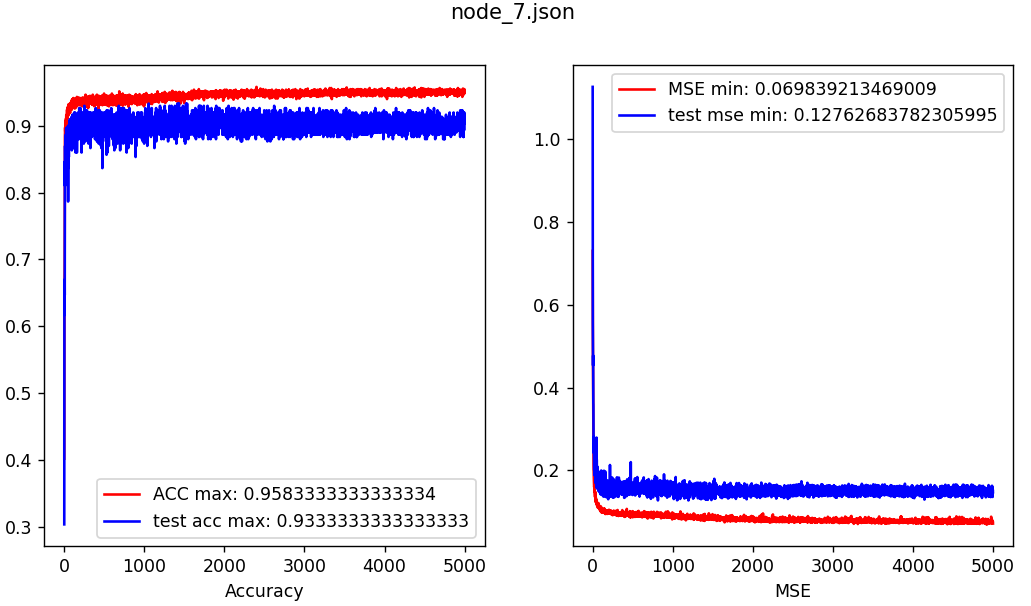


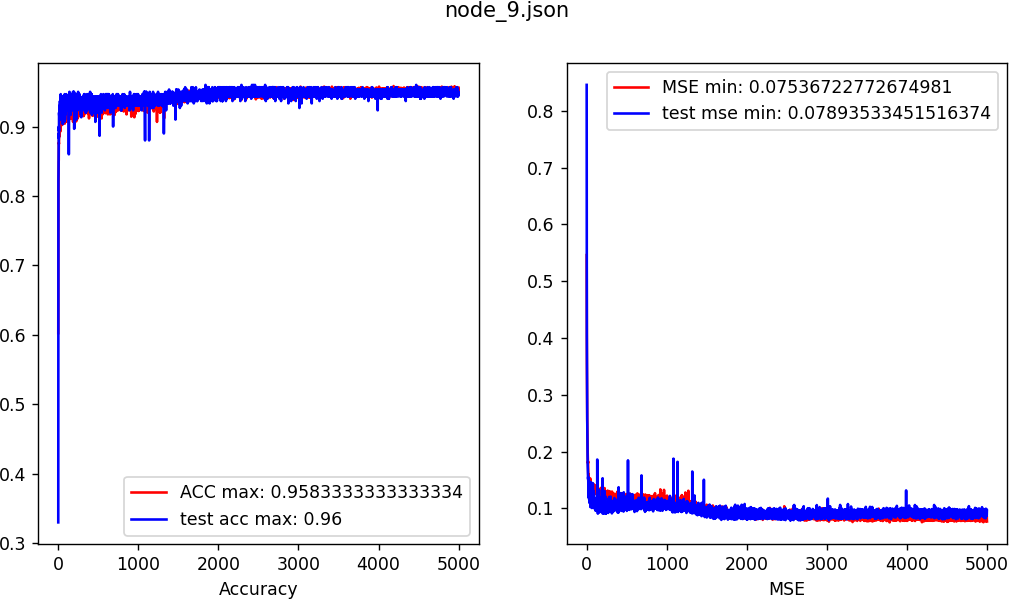


저는 이중에서 가장 정확도가 높게 나왔던 0.009를 사용하고자 했지만 test(valid)와 train의 차가 커지는 것을 보고 좀 더 안정적인 0.008을 사용했습니다.

마지막 파라미터로 노드 수가 남았습니다. 노드 수의 경우 위의 결과는 8개의 노드로 했기 때문에 노드 5, 7, 9일 때를 실행하여 비교하였습니다.



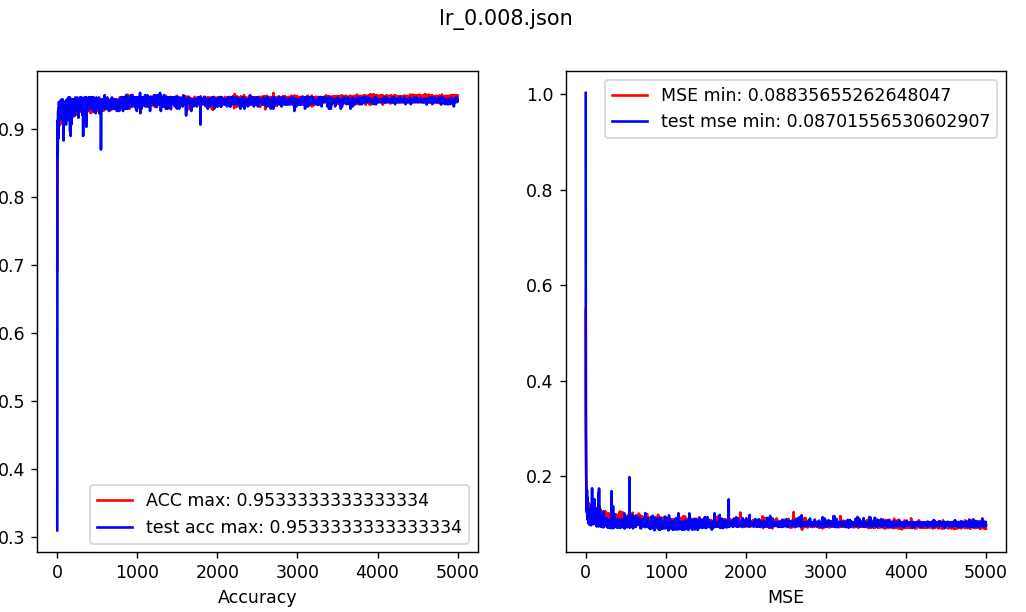




노드 5, 7,8,9를 비교하여 가장 안정적이었던 노드8을 사용했습니다.

# **3. 결과**

특성 2,3,4,7,10 이렇게 5개와 하이퍼파라미터 learning rate=0.008, 노드 수=8개를 설정한 결과는

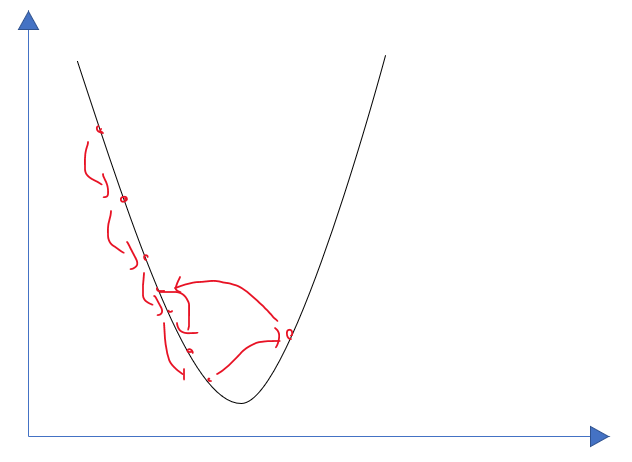


와 같이 나왔습니다.

# **4. 분석**

**문제점1.**

그래프에는 도중에 튀는 현상을 볼 수 있는데 이는 learning rate가 커서



손실함수가 2차원이라 했을 때 위와 같이 global minimum으로 수렴하다 다시 밖으로 튀었다가 안으로 수렴하게 되는 현상으로 보았습니다.

**문제점\_2**

최종적으로 최적의 weight를 통해 구한 결과 값에서 어느 부분의 값이 잘못 나왔는지 확인하기

위해 값을 출력했습니다.

텍스트, 컴퓨터이(가) 표시된 사진

자동 생성된 설명

위 그림은 최종 모델을 통해 최적의 가중치를 구하였고 해당 가중치로 valid set을 돌린 결과 실패한 목록 전체입니다. 결과를 보면 데이터가 틀린 부분은 전부 이미지가 0일때와 2일때를 잘 구분 못하여 생기는 현상이었습니다.

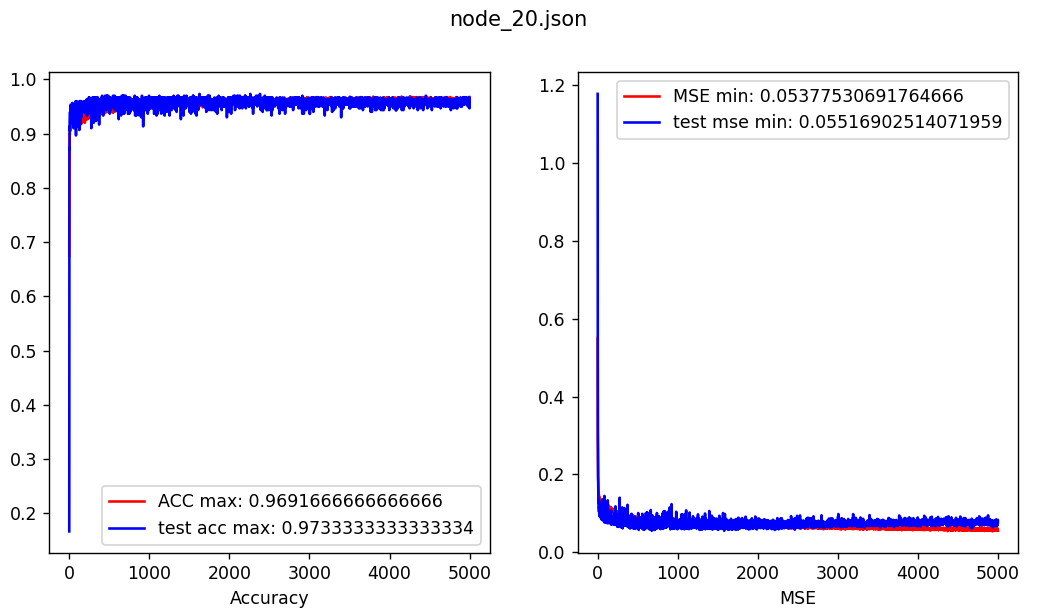
**궁금증 1.**

Node 개수에 따른 연관관계

Node는 입력 데이터에 대한 특성의 개수입니다. 은닉 층 또한 하나의 입력 노드라 보면 모두 특성이라 볼 수 있다고 생각했습니다.

이를 전제로 제가 맨 처음 데이터의 분포도 및 평균과 최대 최소 값을 보았을 때 데이터들이 한쪽으로 치우쳐져 있는 현상이 있었습니다. 따라서 위와 같은 상황에서 제한된 특성을 입력 특성으로 주게 되었을 때 더 높은 성능을 나타나게 하려면 데이터의 분포를 구분 짓게 만들어야 하고 이를 저는 노드의 개수를 늘리면 그 만큼 특성을 세부적으로 가중치들을 쌓으면 분리할 수 있다고 생각을 했고 노드 개수를 5개에서 9개까지 늘렸었습니다.

이때 들었던 생각은 만약 노드를 더 높게 잡게 된다면 train 데이터 셋에 대해서 더 fit하게 될 것이고 이는 overfitting으로 이어질 것이라 생각을 하고 node를 20개로 늘렸지만 overfitting이 일어나지 않았습니다.



# **5. 결론**

딥러닝에서 너무 높은 training 정확도는 오히려 train 데이터 셋에만 적합한 overfitting 현상이 생긴다. 따라서 valid set으로 지속적인 overfitting을 확인해주는 과정이 필요하다.

95%란 결과가 나왔지만 오류 값들을 보면 0하고 2가 제대로 분리가 안된 것을 볼 수 있다. 이를 해결하기 위해서는 제대로 된 특성을 구하는 것부터 해야할 것 같다. 혹은 특성을 변환 시켜 학습을 해야 한다 들었습니다. 따라서 데이터 셋에 한 곳에 분포 되어 있는 데이터 셋을 제곱을 하게 된다면 각각의 데이터가 분산되는 효과가 있지 않을까 하여 제곱하여 계산을 해봤지만 제대로된 학습 값이 나오지 않았습니다.

보통은 입력 노드와 출력 노드 사이로 은닉층의 노드를 한다고 들었지만 이럴 때에는 node의 개수를 늘리는 것도 좋은 방법이었던 거 같습니다. 노드를 늘릴수록 더 높은 valid 정확도가 나오는 것을 확인할 수 있었습니다.

이번 텀프로젝트의 경우 데이터 학습 전 많은 필터링 과정을 거치며 학습해야 하는 경우의 수를 줄이고 이를 통해 결과를 도출했습니다. 하지만 각 데이터 셋을 훈련시키는 시간 또한 많이 들었기에 이를 줄이기 위해 learning rate 자체를 크게 잡아 빠르게 결과를 확인해 보며 이후, learning rate를 조절해 가며 변경하는 것이 생각보다 많은 도움이 되었던 거 같습니다.