인공 신경망 해석 및 분석 과제

담당과목: 자율사물시스템설계

담당교수: 고봉환 교수님

학번: 2019112369

이름: 황규민

목차

[서론]

인공 신경망, Artificial Neural Network(ANN)이란 간략히 신경망(Neural Network)이라고도 하는데, 인간의 뉴런이 연결된 형태를 수학적으로 모방한 모델이다. 인공 신경망은 컴퓨터 과학과 인공지능 연구에서 중요한 역할을 하는데, 주로 패턴 인식, 분류, 예측, 데이터 처리 및 의사 결정에 사용된다. 인공 신경망은 인간의 신경계처럼 많은 양의 뉴런들이 모여 계층을 이루고 자극과 신호를 공유한다.

우리가 신경망을 구축하려면 적어도 다음 세가지 기능을 가져야 한다. 입력, 은닉, 출력 노드의 수 설정을 하는 "초기화" 기능, 학습 데이터들을 통해 학습하고 이에 따라 가중치를 업데이트 하는 "학습" 기능, 그리고 입력을 받아 연산한 후 출력 노드에서 답을 전달하는 "질의" 기능이 있다. 다른 기능들도 존재하지만 이 세가지 기능을 통해 신경망을 구축해 본다.

'신경망 첫걸음' 책과 다르게 Visual Studio Code 프로그램을 사용했으며 주피터 노트북 환경을 사용하기 위해 ipynb 파일을 사용하였다. 신경망을 구축한 후 본 실험에서는 학습률(learning rate), 주기(epoch), 그리고 은닉 노드의 수(hidden nodes) 이 세 변수를 각각 변화해가며 신경망의 성능을 확인하고 최적의 성능 값을 가지는 조합을 찾아낸다. 그리고 실제 손 글씨 데이터를 받았을 때 인공 신경망의 성능을 확인해 본다.

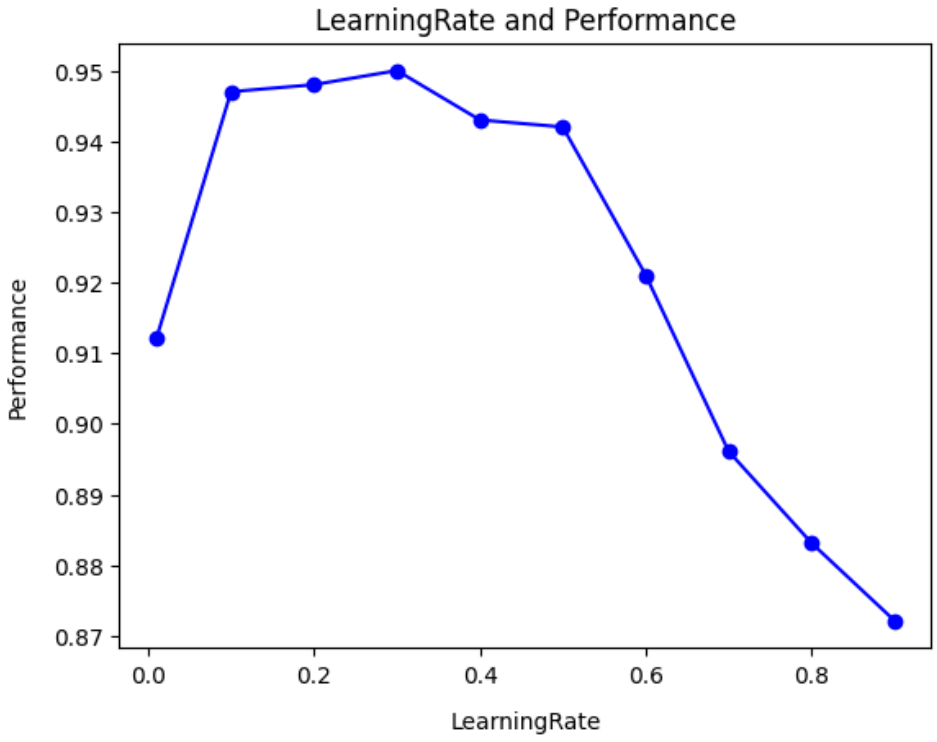
[인공 신경망]

이번 설계 과정에서 사용된 코드의 경우, 책 '신경망 첫걸음'(저자: 타리크 라시드)의 신경망 코드를 참고하여 작성하였다. 코드는 보고서의 맨 마지막 장에 첨부하였다.

설계의 초기 설정 값은 학습률 0.1, 주기 5, 은닉 노드의 수는 200으로 잡았고, 가중치의 경우 random 함수를 사용하여 랜덤으로 정하였다. 인공 신경망의 학습 데이터로는 mnist_train_10k.csv, mnist_test_1k.csv 파일을 테스트 데이터로 사용하였다. 초기 값으로 성능을 평가하였을 때 약 0.947, 94.7%의 값을 가졌다. 이 또한 낮지 않은 수치이지만 학습률, 주기, 은닉노드의 수를 조절해 가면 더 높은 성능을 얻을 수 있을 것이라 기대된다. 또한 코드는 VSC에서 ipynb파일을 사용하여 주피터 환경을 구현하였다. py파일이 아닌 ipynb파일을 사용한 이유는 ipynb파일의 경우 코드를 따로따로 실행시키기가 용이하고 가중치 값을 random함수를 통해 정해두고 고정시켜 두어야 하기 때문이다.

[인공 신경망의 성능과 학습률(learning rate)의 상관 관계]

먼저, 학습률에 따른 성능 차이를 확인해본다. 학습률은 0.01부터 0.9사이의 값을 사용하여 plot하였고 몇 개의 점을 찍어 대략적인 그래프를 그려보았다.

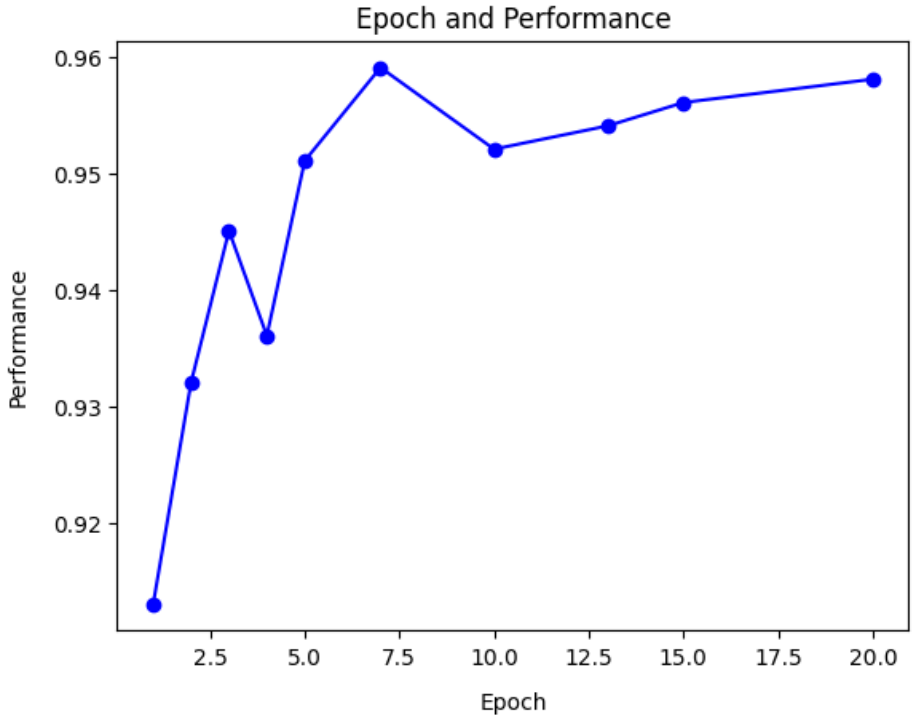


학습률이 0.3일 때 성능 값이 0.95005, 95%로 최대값을 가졌다. 그래프를 분석해

보았을 때 학습률은 값이 커질수록 경사 하강 과정에서 오버피팅(Overfitting)이 일어나면서 튕김 현상이 발생하는 것 같다. 결과값에 따라 최적의 학습률은 0.3으로 설정한다.

[인공 신경망의 성능과 주기의 상관 관계]

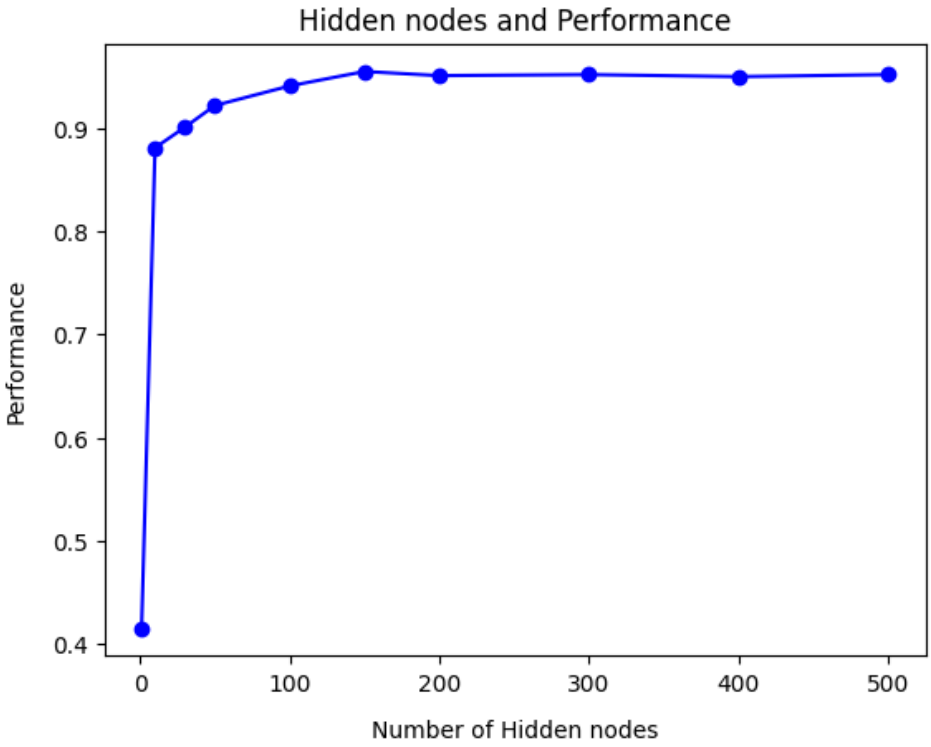
성능과 주기를 앞서 실험을 진행하였던 학습률과 동일한 방식으로 그래프를 그려보면 다음과 같다. 학습 주기를 1부터 20사이의 값을 대입해가며 주기와 성능 값을 plot해 보았다.



주기, 즉 반복 수행 값이 클수록 성능이 좋아지는 것을 볼 수 있지만, 코드를 돌리는 소모 시간 또한 증가하기 때문에 최적 주기는 7로 설정한다.

[인공 신경망의 성능과 은닉 노드의 상관 관계]

마지막으로 은닉 노드의 수와 성능과의 상관 관계를 그래프로 그려보면 다음과 같다. 은닉 노드의 수는 5부터 500까지의 값을 사용하였다.



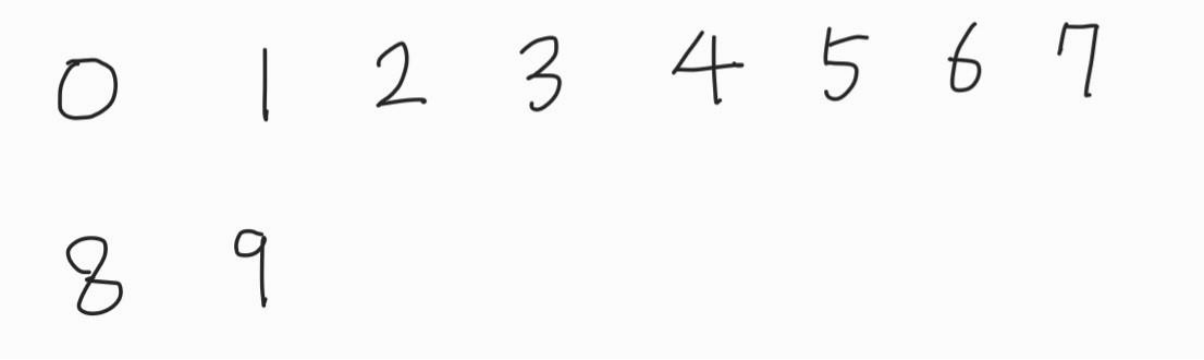
은닉 노드는 실제 학습이 일어나는 계층이기에 은닉 노드의 수는 결과 값에 큰 영향을 미친다. 결과를 분석해 보았을 때 은닉 노드의 수가 5개일 때 0.4155의 성능 값을 가졌다. 그리고 은닉 노드의 수가 10일 때 0.8811, 30일 때 0.9011로 비교적 안정적인 성능을 내었다. Plot 결과에 따라 최적의 은닉 노드의 수는 150으로 설정한다.

[최적의 신경망 조건]

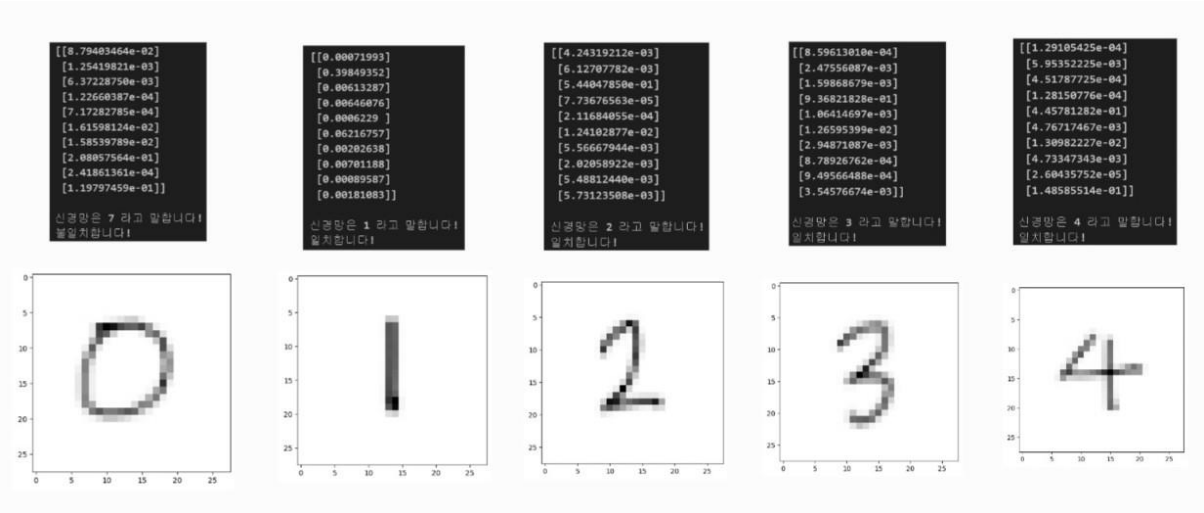
위 결과를 통해 최적의 성능을 내는 인자의 조합을 알아보았을 때 학습률이 0.3, 주기가 7, 은닉 노드의 수가 150개일 때 최대의 성능을 낸다는 사실을 알아내었다. 그리고 성능값은 0.957, 95.7%의 정확도를 내었다.

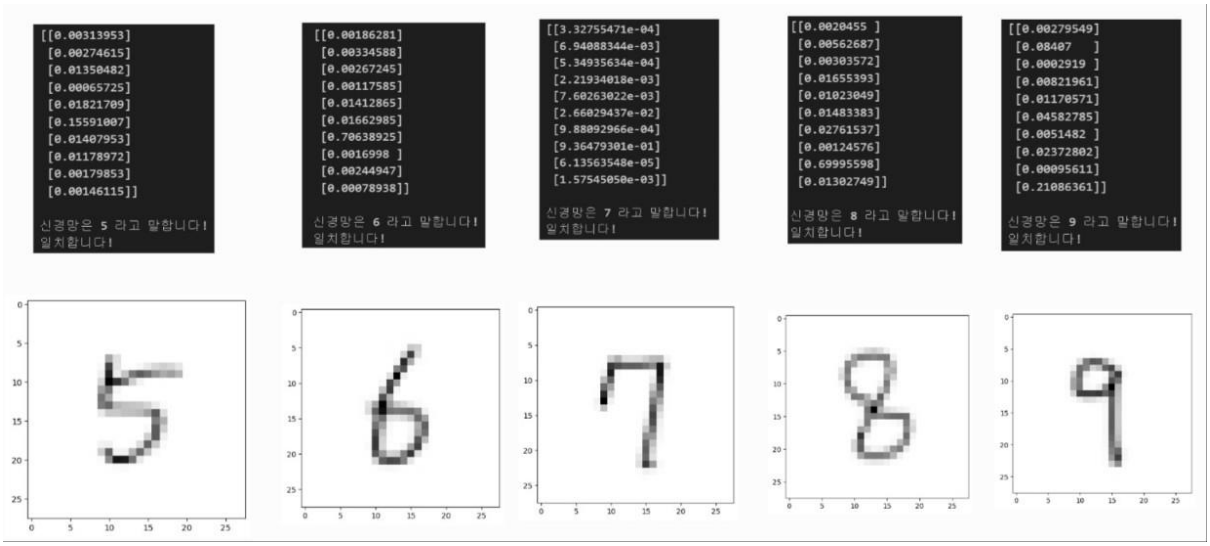
[인공 신경망 - 손 글씨 데이터]

앞서 최적의 성능을 내는 인자 조합 학습률 0.3, 주기 7, 은닉 노드의 수 150개를 사용하여 인공 신경망이 손 글씨 데이터를 받았을 때 어느 정도 성능을 내는지 확인해 본다.



교수님께서 제공해주신 사이트를 통해 위 손 글씨 데이터를 28*28 픽셀 크기의 png 파일로 추출하였다. 그리고 인공신경망에 질의를 하여 입력한 손 글씨 데이터 값이 어떤 것인지 답을 하게 하였다. 여기서 알아 두어야 하는 점은 돌아오는 답은 확률이라는 점이다. 그리고 신경망은 짜인 코드에 따라 맨 위(숫자 0에 해당)에서 맨 아래(숫자 9에 해당) 중 확률이 가장 큰 값에 해당하는 칸에 따라 주어진 숫자를 반환하게 된다.





95%의 성능을 가진 신경망은 결과적으로 질의를 하였을 때 0을 제외한 모든 데이터를 잘 파악하였다.

[결과 및 고찰]

이번 설계에서는 인공신경망을 구축하기 필요한 최소 조건인 초기화, 학습, 질의를 통해 신경망을 구축하였고, 학습률, 주기, 은닉 노드의 수를 변화시켜가며 최적의 성능을 낼 수 있는 신경망을 찾아내 보았다. 하지만 최소 설계 조건으로만 설계를 하여서 그런지 최대 성능 값은 0.957를 얻을 수 있었고 설계 과정 중 변수를 바꿔가며 성능을 측정하였지만 성능 값이 99%와 같이 100%에 근접한 값까지 치솟지는 않았다.

이미지 전처리 과정 중 작가의 github에 나와있는 코드를 사용하여 사진을 흑백으로 바꾸기 위해 imageio 함수의 (as_gray=True)를 사용하려 했지만 이 옵션이 만료가 되어 코드를 쓸 수 없게 되었다. 따라서 코드 수정이 불가피해졌고 이를 해결하기 위해 OpenCV와 imagio를 혼용하여 문제를 해결하였다. 이렇게 설계를 한 이유는 OpenCV를 잘 다루지 못한 이유가 크다. 이번 설계 과정을 진행하면서 OpenCV를 처음 접해 보았기 때문에 아는 만큼 사용하였다. 또 아쉬웠던 점은 학습률, 주기, 은닉 노드의 수를 하나하나 산출하여 plot하였는데, 코드를 더 잘 다룰 줄 알았더라면 for 문을 통해 plot을 더 잘 할 수도 있었기에 아쉬움이 든다.

본 실험에서 사용하였던 ANN방식 외에 DNN(Deep Neural Network) 와 CNN(Convolutional Neural Network) 등 다양한 학습모델이 존재한다. 공부를 하면서 CNN, DNN과 같은 다른 학습모델의 학습 과정 및 다른 사람들의 코드들도 살펴보았는데 본 설계 과정처럼 mnist 데이터 csv파일을 통해 받아오는 것이 아닌, TensorFlow 서버를

통해 불러와서 train을 하는 방식을 사용하기도 한다는 것을 알게 되었다. 그리고 TensorFlow로 잘 짜인 CNN 설계를 하였을 때 성능이 99%까지 도달하는 것을 보았다. 이를 보며 TensorFlow를 이용한 신경망 구축도 시간이 된다면 해보고 싶다는 생각이 들었다.

이번 설계를 진행하면서 인공 신경망이 작동하는 원리와 과정, 그리고 프로그래밍 지식을 배웠고, 여러 신경망과 코드를 찾아보면서 많은 것 들을 배워 갈 수 있었다.

[사용 코드]