

# 인공지능 개인 과제

## - 삼목데이터 학습-

|    |       |           |            |
|----|-------|-----------|------------|
| 과  | 목 :   | 인공지능      |            |
| 담당 | 교수님 : | 김동근 교수님   |            |
| 학  | 번 :   | 201501900 |            |
| 제  | 출     | 자 :       | 김종희        |
| 제  | 출     | 일 :       | 2019.12.06 |

# 목차

|                         |     |
|-------------------------|-----|
| 1. 서론 .....             | p.2 |
| 2. 본론                   |     |
| 2-1. 사용기술               |     |
| 2-1-1. Tensorflow ..... | p.2 |
| 2-1-2. MLP .....        | p.2 |
| 2-1-3. Matplotlib ..... | p.2 |
| 2-2. 삼목학습               |     |
| 2-2-1. 데이터셋 .....       | p.3 |
| 2-2-2. MLP 설계도 .....    | p.3 |
| 2-3. 코드실험 .....         | p.4 |
| 3. 결론 .....             | p.6 |
| 4. 부록 .....             | p.7 |
| 참고문헌 .....              | p.9 |

## 1. 서론

이번 보고서의 과제는 인공지능을 가지고 틱택토 게임을 학습시키는 것이다. 인공지능은 인간이 지닌 지적 능력의 일부 또는 전체를 인공적으로 구현한 것으로 인공지능이란 용어가 처음 등장한 때는 1956년에 미국 다트머스에서 마빈 민스키, 클로드 새넌 등 인공지능 및 정보 처리 이론에 지대한 공헌을 한 사람들이 개최한 학회에서 존 매카시가 이 용어를 사용하면서 사용하면서부터이다. 하지만 인공지능이라는 개념 자체는 훨씬 예전부터 있었다. 예를 들면, 앨런 튜링이 '생각하는 기계'의 구현 가능성과 튜링 테스트를 제안한 것은 1950년의 일이며, 최초의 신경망 모델은 1943년에 제안되었다.

틱택토(Tic-Tac-Toe)란 우리가 흔히 삼목이라고 부르는 것으로 서양 놀이의 일종으로 두 명이 번갈아가며 O와 X를 3×3 판에 써서 같은 글자를 가로, 세로, 혹은 대각선 상에 놓이도록 하는 놀이다. (3,3,3) 게임으로 오목과 비슷한 형태의 추상전략 보드게임이다.



<그림 1> Tic-Tac-Toe 게임 구조

## 2. 본론

### 2-1. 사용 기술

#### 2-1-1. Tensorflow

다양한 작업에 대해 데이터 흐름 프로그래밍을 위한 오픈소스 소프트웨어 라이브러리이다. 심볼릭 수학 라이브러리이자, 뉴럴 네트워크같은 기계학습 응용프로그램에도 사용된다. 이것은 구글내 연구와 제품개발을 위한 목적으로 구글 브레인팀이 만들었고 2015년 11월 9일 아파치 2.0 오픈소스 라이선스로 공개되었다.

#### 2-1-2. MLP

다층 퍼셉트론(Multilayer Perceptron)이란 이제 인간이 생각하고 학습하는 방법을 인공지능이 흉내내기 위해 인공신경망이란 개념을 만들어냈고 이를 실현하기 위해 인간의 뉴런을 퍼셉트론으로 흉내를 내서 그 목적을 실현하려 했으나 인간이 생각하기에는 간단한 XOR문제도 해결을 못하는 난관에 봉착해서 이를 해결하기 위한 시도에서 나온 것이다. 그러나 이렇게 다층으로 퍼셉트론을 쌓아 XOR문제 및 여러 복잡한 문제를 해결할 것만 같았던 다층 퍼셉트론에도 문제가 하나 있었으니 바로 Output Layer의 Target value가 있는 것과 달리 중간에 Hidden Layer가 가지고 있는 node에는 Target Value가 존재하지 않아 학습을 시킬 방법이 없었던 것이다. 이러한 다층 퍼셉트론의 문제를 해결하기 위해 나온 것이 바로 역전파. delta rule을 통해 층이 많더라도 안되던 학습의 문제를 해결하게 된다.

#### 2-1-3. Matplotlib

Matplotlib는 파이썬에서 데이터를 차트나 플롯(Plot)으로 그려주는 라이브러리 패키지로써 가장 많이 사용되는 데이터 시각화(Data Visualization) 패키지로 알려져 있다. Matplotlib는 라인 플롯, 바 차트, 파이차트, 히스토그램, Box Plot, Scatter Plot 등을 비롯하여 다양한 차트와 플롯 스타일을 지원하며, Matplotlib.org 갤러리 웹페이지에서 다양한 샘플 차트를 볼 수 있다. Matplotlib를 이용하여 데이터 시각화(Data Visualization)를 사용하기 위해서 과학용 파이썬 배포판인 아나콘다(Anaconda)를 설치해서 Jupyter Notebook을 사용하면 편리하다.

## 2-2. 삼목 학습

### 2-2-1. 데이터 셋

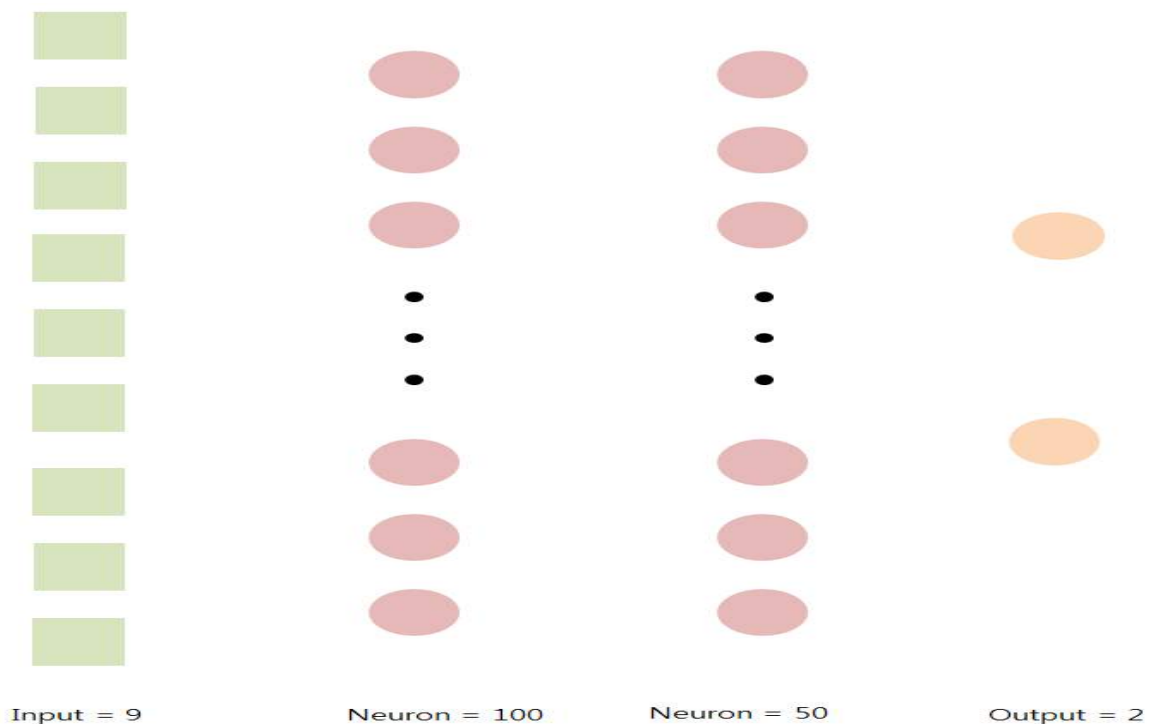
원래 제공된 틱택토 게임의 데이터 셋은 게임이 끝난 상태의 게임 판 모양, 그 상황에서 누가 이겼는지에 대한 정보를 나타낸다. 이 데이터 셋은 x, o, b(x : player1의 돌, o : player2의 돌, b : 빈 공간)를 사용해 게임 판 모양을, true, false(true : player1 승리, false : player2 승리)를 통해 게임의 승자를 표현했다. 그래서 데이터를 바로 처리하여 학습을 시키기에는 데이터가 부적절하다. 따라서 데이터를 학습을 시킬 수 있게 문자로 표현되어있는 데이터를 임의의 수치로 변경하여 데이터를 사용할 수 있게 하였다.

| 원래 데이터 | x  | o | b | true  | false |
|--------|----|---|---|-------|-------|
| 변환 데이터 | -1 | 1 | 0 | (1,0) | (0,1) |

표 1. 데이터 변환

위 표와 같이 데이터를 변환해 프로그램에서 저장한 데이터를 사용할 수 있도록 했고 true, false 부분은 현재 주어진 문제의 출력층이 2개이기 때문에 출력층 노드의 개수인 2만큼 데이터를 채워준 (1,0), (0,1)로 변환해 저장했다. 입력받은 데이터 셋은 player1, player2의 승리가 정렬되어 저장됐기 때문에 바로 traindata, testdata를 6:4 비율로 나누면 각 데이터들은 한쪽으로 편향된 데이터만 저장된다. 따라서 이를 방지하기 위해 shuffle 메소드를 사용해 저장된 데이터들을 섞어 데이터가 한쪽으로 편향돼 저장되는 것을 방지했다.

### 2-2-2. MLP 설계도



<그림2> 3층 MLP 구조 (가중치 생략)

## 2-3. 코드 실험

- 다양한 최적화 방법에 따른 학습 결과

| SGD                                                                                                                                                                                                                                                                                     |  | Adam                                                                                                                                                                                                                                                                                                                   | RMSprop                                                                                                                                                                                                                                                                                                                  |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 손실함수 및 학습률                                                                                                                                                                                                                                                                              |  | 0.01 ( 최적화 방법의 차이를 보기 위해 모든 최적화 방법의 학습률은 0.01 )                                                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                          |
|                                                                                                                                                                                                                                                                                         |  | epoch = 10001<br>손실함수 : Cross-Entropy                                                                                                                                                                                                                                                                                  |                                                                                                                                                                                                                                                                                                                          |
|                                                                                                                                                                                                                                                                                         |  |                                                                                                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                          |
| <그림3> SGD 학습결과 그래프                                                                                                                                                                                                                                                                      |  | <그림4> Adam 학습결과 그래프                                                                                                                                                                                                                                                                                                    | <그림5> RMSprop 학습결과 그래프                                                                                                                                                                                                                                                                                                   |
| Test_Hypothesis: Squeezed text (384 lines).<br>Test_Accuracy: 0.9791667<br>Train_Hypothesis: [[0.03538042 0.96461964]<br>[0.9753549 0.02464512]<br>[0.9981804 0.00181968]<br>[0.95038074 0.04961923]<br>[0.91294205 0.08705797]<br>[0.0377183 0.9622817 ]]<br>Train_Accuracy: 0.9965157 |  | Test_Hypothesis: Squeezed text (384 lines).<br>Test_Accuracy: 0.9817708<br>Train_Hypothesis: [[5.3820353e-09 1.0000000e+00]<br>[9.999998e-01 8.2969521e-08]<br>[1.0000000e+00 9.993524e-10]<br>[5.5526037e-09 1.0000000e+00]<br>[1.0000000e+00 5.5310647e-08]<br>[4.1696686e-09 1.0000000e+00]]<br>Train_Accuracy: 1.0 | Test_Hypothesis: Squeezed text (384 lines).<br>Test_Accuracy: 0.9921875<br>Train_Hypothesis: [[9.9999976e-01 2.5115148e-07]<br>[1.0000000e+00 1.2206837e-06]<br>[9.9999986e-01 9.2307602e-06]<br>[1.0000000e+00 1.7427354e-08]<br>[7.4378147e-10 1.0000000e+00]<br>[1.0000000e+00 6.9398015e-11]]<br>Train_Accuracy: 1.0 |
| <그림6> SGD 학습결과 값                                                                                                                                                                                                                                                                        |  | <그림7> Adam 학습결과 값                                                                                                                                                                                                                                                                                                      | <그림8> RMSprop 학습결과 값                                                                                                                                                                                                                                                                                                     |
| 파란색 그래프선(cost), 주황색 그래프선(Accuracy)                                                                                                                                                                                                                                                      |  |                                                                                                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                          |

표 2. 최적화 방법에 따른 학습 결과

- epoch에 따른 학습 결과

| epoch = 1000                                                                                                                                                                                                                                                                           |  | epoch = 5000                                                                                                                                                                                                                                                                                                           | epoch = 10000                                                                                                                                                                                                                                                                         |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 최적화 방법 및 손실함수                                                                                                                                                                                                                                                                          |  | 손실함수 : Cross-Entropy<br>최적화 방법 : SGD                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                       |
|                                                                                                                                                                                                                                                                                        |  | (epoch에 따른 학습결과와 차이를 보기 위해 최적화 방법 및 손실함수 동일,<br>학습률 또한 0.01로 동일)                                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                                                                       |
|                                                                                                                                                                                                                                                                                        |  |                                                                                                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                       |
| <그림9> epoch(1000) 학습결과 그래프                                                                                                                                                                                                                                                             |  | <그림10> epoch(5000) 학습결과 그래프                                                                                                                                                                                                                                                                                            | <그림11> epoch(10000) 학습결과 그래프                                                                                                                                                                                                                                                          |
| Test_Hypothesis: Squeezed text (384 lines).<br>Test_Accuracy: 0.7786458<br>Train_Hypothesis: [[0.9985977 0.00140236]<br>[0.4206972 0.5793028]<br>[0.99101514 0.00898487]<br>[0.62395406 0.376046 ]<br>[0.01774066 0.9822594]<br>[0.19228728 0.80771273]]<br>Train_Accuracy: 0.78745645 |  | Test_Hypothesis: Squeezed text (384 lines).<br>Test_Accuracy: 0.9817708<br>Train_Hypothesis: [[5.3820353e-09 1.0000000e+00]<br>[9.999998e-01 8.2969521e-08]<br>[1.0000000e+00 9.993524e-10]<br>[5.5526037e-09 1.0000000e+00]<br>[1.0000000e+00 5.5310647e-08]<br>[4.1696686e-09 1.0000000e+00]]<br>Train_Accuracy: 1.0 | Test_Hypothesis: Squeezed text (384 lines).<br>Test_Accuracy: 0.984375<br>Train_Hypothesis: [[0.99221605 0.007784 ]<br>[0.85679644 0.14320359]<br>[0.9950956 0.0049044 ]<br>[0.9944829 0.00551712]<br>[0.01736161 0.98263836]<br>[0.00319625 0.99680376]]<br>Train_Accuracy: 0.989547 |
| <그림12> epoch(1000) 학습결과 값                                                                                                                                                                                                                                                              |  | <그림13> epoch(5000) 학습결과 값                                                                                                                                                                                                                                                                                              | <그림14> epoch(10000) 학습결과 값                                                                                                                                                                                                                                                            |
| 파란색 그래프선(cost), 주황색 그래프선(Accuracy)                                                                                                                                                                                                                                                     |  |                                                                                                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                       |

표 3. epoch에 따른 학습 결과

- Learning\_rate에 따른 학습 결과

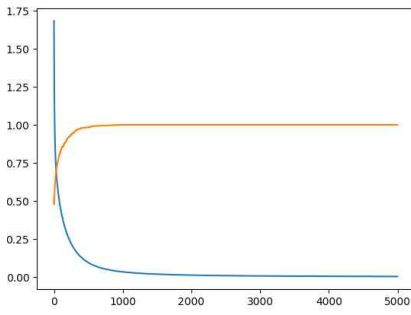
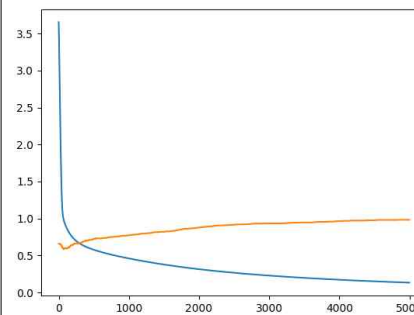
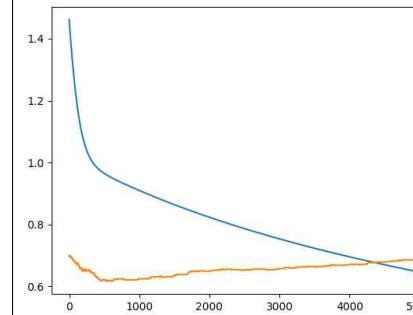
| Learning_rate = 0.1                                                                                                                                                                                                                                                                                                     |  | Learning_rate = 0.01                                                                                                                                                                                                                                                                   |  | Learning_rate = 0.001                                                                                                                                                                                                                                                                |  |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| 최적화 방법 및<br>손실함수                                                                                                                                                                                                                                                                                                        |  | Cross-entropy<br>SGD                                                                                                                                                                                                                                                                   |  |                                                                                                                                                                                                                                                                                      |  |
|                                                                                                                                                                                                                                                                                                                         |  | (Learning_rate에 따른 학습결과 차이를 보기 위해 최적화 방법 및 손실함수 동일,<br>epoch 또한 5000로 동일)                                                                                                                                                                                                              |  |                                                                                                                                                                                                                                                                                      |  |
|                                                                                                                                                                                                                                        |  |                                                                                                                                                                                                       |  |                                                                                                                                                                                                   |  |
| <그림15> Learning_rate(0.1)학습결과 그래프                                                                                                                                                                                                                                                                                       |  | <그림16> Learning_rate(0.01) 학습결과 그래프                                                                                                                                                                                                                                                    |  | <그림17> Learning_rate(0.001) 학습결과 그래프                                                                                                                                                                                                                                                 |  |
| Test_Hypothesis: Squeezed text (384 lines).<br>Test_Accuracy: 0.9765625<br>Train_Hypothesis: [[1.6878698e-02 9.8312128e-01]<br>[4.0551373e-03 9.9594492e-01]<br>[9.8991090e-01 1.0089062e-02]<br>[9.8653817e-01 1.3461822e-02]<br>[9.992085e-01 7.9132078e-05]<br>[9.9747735e-01 2.5226080e-03]]<br>Train_Accuracy: 1.0 |  | Test_Hypothesis: Squeezed text (384 lines).<br>Test_Accuracy: 0.9453125<br>Train_Hypothesis: [[0.9795154 0.02048464]<br>[0.41052556 0.58947444]<br>[0.04079062 0.9592093]<br>[0.14961477 0.85038525]<br>[0.9462716 0.05372841]<br>[0.58660305 0.4133969]]<br>Train_Accuracy: 0.9825784 |  | Test_Hypothesis: Squeezed text (384 lines).<br>Test_Accuracy: 0.6953125<br>Train_Hypothesis: [[0.99010134 0.0098987]<br>[0.82514036 0.17485963]<br>[0.20768887 0.7923111]<br>[0.99835783 0.00164216]<br>[0.98703593 0.0129641]<br>[0.44524002 0.55476]]<br>Train_Accuracy: 0.6864114 |  |
| <그림18> Learning_rate(0.1)학습결과 값                                                                                                                                                                                                                                                                                         |  | <그림19> Learning_rate(0.01) 학습결과 값                                                                                                                                                                                                                                                      |  | <그림20> Learning_rate(0.001)학습결과 값                                                                                                                                                                                                                                                    |  |
| 파란색 그래프선(cost), 주황색 그래프선(Accuracy)                                                                                                                                                                                                                                                                                      |  |                                                                                                                                                                                                                                                                                        |  |                                                                                                                                                                                                                                                                                      |  |

표 4. Learning\_rate에 따른 학습결과

- TestData, TrainData 비교

| TestData                                                                                                                                                                                                                                                                                                                                           | TrainData                                                                                                                                                                    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 해당 학습 결과는 epoch : 5000, 학습률 : 0.1<br>MSE, SGD를 사용한 결과이다.                                                                                                                                                                                                                                                                                           |                                                                                                                                                                              |
| Test_Hypothesis: [[9.93241072e-01 6.75897719e-03]<br>[1.41089976e-01 8.58909965e-01]<br>[1.95956081e-02 9.80404437e-01]<br>[9.53010678e-01 4.69892435e-02]<br>[9.40277457e-01 5.97225428e-02]<br>....<br>[9.98506486e-01 1.49349554e-03]<br>[9.90016162e-01 9.98381525e-03]<br>[1.62793938e-02 9.83720660e-01]<br>[9.92064059e-01 7.93592166e-03]] | Train_Hypothesis: [[0.01566616 0.9843338]<br>[0.12596329 0.8740367]<br>[0.9334935 0.06650652]<br>[0.00252602 0.997474]<br>[0.8407343 0.15926571]<br>[0.03580721 0.96419275]] |
| <그림21> TestData 결과 값 일부                                                                                                                                                                                                                                                                                                                            | <그림22> TrainData 결과 값 일부                                                                                                                                                     |
| [1,0], [0,1], [0,1], [1,0] [1,0] ..... [1,0]<br>1승리, 2승리, 2승리, 1승리, 1승리 ..... 1승리                                                                                                                                                                                                                                                                  | [0,1], [0,1], [0,1] ..... [1,0], [1,0], [0,1]<br>2승리, 2승리, 2승리 ..... 1승리, 1승리, 2승리                                                                                           |
| TestData_Hypothesis 결과 분석                                                                                                                                                                                                                                                                                                                          | TrainData_Hypothesis 결과 분석                                                                                                                                                   |

-손실함수에 따른 학습 결과

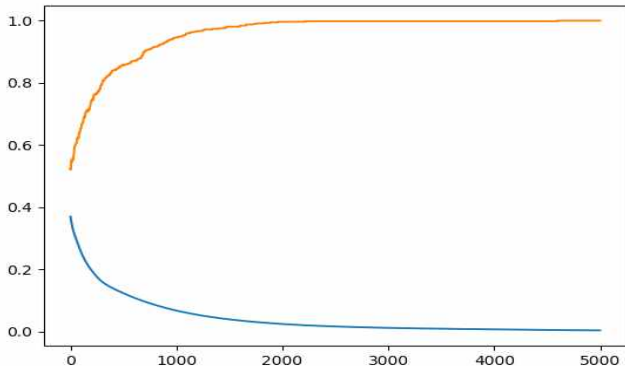
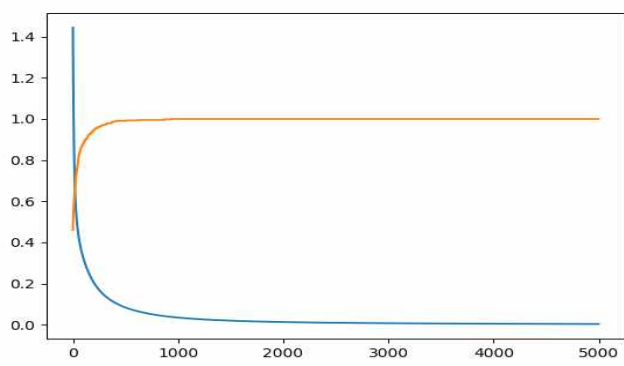
| MSE                                                                                                                                                                                                                                                                  |                                                                                                           | Cross-Entropy                                                                                                                                                                                                                                                                                               |  |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| 최적화 방법 및<br>학습률                                                                                                                                                                                                                                                      | 최적화 방법 : SGD                                                                                              |                                                                                                                                                                                                                                                                                                             |  |
|                                                                                                                                                                                                                                                                      | (표3, 4에 따라 가장 학습정도가 좋았던 학습률과 epoch를 사용해 학습률은 0.1, epoch 5000사용, 5000번의 cost 값은 많기 때문에 1000번 마다 cost 값 출력) |                                                                                                                                                                                                                                                                                                             |  |
|                                                                                                                                                                                     |                                                                                                           |                                                                                                                                                                                                                           |  |
| <그림23> MSE 사용 학습 결과 그래프                                                                                                                                                                                                                                              |                                                                                                           | <그림24> Cross-Entropy 사용 학습 결과 그래프                                                                                                                                                                                                                                                                           |  |
| <pre>Test_Hypothesis: Squeezed text (384 lines). Test_Accuracy: 0.9765625 Train_Hypothesis: [[0.01566616 0.9843338 ] [0.12596329 0.8740367 ] [0.9334935 0.06650652] [0.00252602 0.997474 ] [0.8407343 0.15926571] [0.03580721 0.96419275]] Train_Accuracy: 1.0</pre> |                                                                                                           | <pre>Test_Hypothesis: Squeezed text (384 lines). Test_Accuracy: 0.9895833 Train_Hypothesis: [[9.9990034e-01 9.9618810e-05] [9.9037021e-01 9.6297814e-03] [9.9936432e-01 6.3565391e-04] [2.7554308e-03 9.9724460e-01] [9.9881661e-01 1.1834608e-03] [2.5554202e-03 9.9744463e-01]] Train_Accuracy: 1.0</pre> |  |
| <그림25> MSE 사용 학습 결과 값                                                                                                                                                                                                                                                |                                                                                                           | <그림26> Cross-Entropy 사용 학습 결과 값                                                                                                                                                                                                                                                                             |  |
| <pre>0 0.37044427 Squeezed text (369 lines). 1000 0.06810777 Squeezed text (244 lines). 2000 0.02493057 Squeezed text (369 lines). 3000 0.0124666495 Squeezed text (244 lines). 4000 0.0076666744 Squeezed text (369 lines).</pre>                                   |                                                                                                           | <pre>0 1.4440123 Squeezed text (369 lines). 1000 0.034858547 Squeezed text (289 lines). 2000 0.013392289 Squeezed text (369 lines). 3000 0.0076507884 Squeezed text (369 lines). 4000 0.0051475014 Squeezed text (369 lines).</pre>                                                                         |  |
| <그림27> MSE 손실함수 값                                                                                                                                                                                                                                                    |                                                                                                           | <그림28> Cross-Entropy 손실함수 값                                                                                                                                                                                                                                                                                 |  |
| 파란색 그래프선(cost), 주황색 그래프선(Accuracy)                                                                                                                                                                                                                                   |                                                                                                           |                                                                                                                                                                                                                                                                                                             |  |

표 5. 손실함수에 따른 학습 결과

### 3. 결론

본문 2-3과 같이 인공지능의 학습은 많은 요소로 영향을 받는다. 학습률, 학습횟수, 최적화 방법, 손실함수 등에 따라 다르며 데이터마다 상황마다 컴퓨터마다 실행할 때마다 그 값은 천차만별이다. 즉, 특정 학습률과 학습 횟수로 학습이 잘됐다고 해서 그 수치들이 절대적인 수치가 아니라는 것을 얘기한다. 흔히 학습 횟수를 늘리면 좋은 학습결과를 가져올 것이라 예상하고 실제로도 좋은 학습결과가 나온다. 하지만 학습 횟수가 높을수록 좋은 학습결과가 나온다고 장담할 수 없다. 본문의 경우 epoch 5000이 10000보다 좋은 학습결과를 보여주고 있다. 따라서 epoch값이 높을수록 학습결과가 좋을 수 있지만 절대적으로 좋다는 것은 아니라는 것을 알 수 있으며 학습률 또한 마찬가지이다. 그리고 학습률은 최적화 방법마다 좋은 학습결과를 위해 요구하는 수치가 다르다.

제공된 삼목 데이터는 player1의 승리 데이터가 player2의 승리 데이터의 2배가량 더 많이 있었다. 또 삼목을 진행하다보면 무승부가 나는 경우도 있지만 무승부에 관한 데이터가 없었다는 부분이 아쉬웠고 실제 삼목게임에서 일어나는 경우의 수와 비교해 봤을 때 데이터의 수가 현저히 적다. 그렇기 때문에 player2에 대한 데이터 양이 player1보다 적다라는 부분은 많이 안타까웠으며 만약 현재 학습시킨 데이터로 게임을 진행한다면 player1



의 승리 결과를 많이 볼 수 있었을 것으로 예상되고 그 결과 자체도 적은 데이터로 학습을 시켰기 때문에 올바르게 판단을 했다고 말 할 수 있을 것 같지는 않다.

처음 인공지능을 들을 때 학습을 시킨다는 것 자체가 무엇인가 의문을 가졌다. 그래서 단순히 모든 상황에 대해서 사용자가 상황과 답을 저장해 그 저장된 답에 따라 결과를 보여주게 하는 것이 학습일거라 생각했다. 하지만 인공지능의 학습은 주어진 데이터에 있는 답만 활용하는 것이 아닌 정답이 없는 데이터도 학습이 가능하였다. 따라서 무조건 사용자가 상황에 따른 답을 주어져야만 학습할 수 있는 것이 아닌 답이 없어도 스스로가 답을 내릴 수 있는 것이다. 이번 과제를 통해 인공지능학습을 경험하면서 인공지능 학습에 대한 생각이 바뀌었고 현재 4차 산업의 핵심이라고 할 수 있는 인공지능을 부족하지만 할 수 있는 선에서 데이터를 직접 학습시킬 수 있는 좋은 경험이었다.

#### 4. 부록

```
import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt

###Data Handling###

# x : -1 / o : 1 / b : 0 / true : (1,0), false : (0,1)
data_all = np.genfromtxt('data1.csv', delimiter=',', dtype=np.float32)

np.random.shuffle(data_all)

#데이터 나누기 6:4
train_num = int(len(data_all)*0.6)
test_num = len(data_all)-train_num

data_train = data_all[:train_num]
data_test = data_all[train_num:]
x_train = data_train[:, 0:9]
y_train = data_train[:, 9:11]
x_test = data_test[:, 0:9]
y_test = data_test[:, 9:11]

#학습 데이터 설정
iteration = 20000
learning_rate = 0.1

X = tf.placeholder(tf.float32, [None, 9])
Y = tf.placeholder(tf.float32, [None, 2])

#입력 ~ layer1
```



```

W = tf.Variable(tf.random_normal([9,100]))
B = tf.Variable(tf.random_normal([100]))
layer1 = tf.sigmoid(tf.matmul(X, W)+B)

#layer1 ~ layer2
W2 = tf.Variable(tf.random_normal([100,50]))
B2 = tf.Variable(tf.random_normal([50]))
layer2 = tf.sigmoid(tf.matmul(layer1, W2)+B2)

#layer2 ~ 출력
W3 = tf.Variable(tf.random_normal([50,2]))
B3 = tf.Variable(tf.random_normal([2]))
result = tf.matmul(layer2, W3)+B3

#sigmoid
#hypothesis = tf.sigmoid(result)
#softmax
hypothesis = tf.nn.softmax(result)

#MSE
cost = tf.reduce_mean(tf.square(hypothesis-Y))
#softmax를 사용한 Cross-Entropy
#cost = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits(logits=result, labels=Y))

#Adam
#train = tf.train.AdamOptimizer(learning_rate).minimize(cost)
#SGD
train = tf.train.GradientDescentOptimizer(learning_rate).minimize(cost)
#RMSprop
#train = tf.train.RMSpropOptimizer(learning_rate,0.9).minimize(cost)

# 출력층이 1개일 때
#predicted = tf.cast(hypothesis > 0.5, dtype=tf.float32)

# 출력층이 2개 이상일 때
predicted = tf.argmax(hypothesis, 1)
correct_prediction = tf.equal(predicted, tf.argmax(Y, 1))
accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))

#교육 데이터와 실제 값 비교한 결과값(얼마나 실제값과 비슷했는가) / 출력층이 1개일 때
#accuracy = tf.reduce_mean(tf.cast(tf.equal(predicted, Y), dtype=tf.float32))

```

```

cost_list = []
accu_list = []

with tf.Session() as sess:
    sess.run(tf.global_variables_initializer())

    for step in range(iteration):
        sess.run(train, feed_dict={X: x_train, Y: y_train})
        c , a= sess.run([cost, accuracy], feed_dict={X: x_train, Y: y_train})
        accu_list.append(a)
        cost_list.append(c)

        if step % 1000 == 0:
            print(step, sess.run(cost, feed_dict={X: x_train, Y: y_train}), sess.run([W, W2, W3]))

    h, a = sess.run([hypothesis, accuracy], feed_dict={X: x_test, Y: y_test})
    print("\nTest_Hypothesis: ", h, "\nTest_Accuracy: ", a)
    h, a = sess.run([hypothesis, accuracy], feed_dict={X: x_train, Y: y_train})
    print("\nTrain_Hypothesis: ", h, "\nTrain_Accuracy: ", a)

    plt.plot(range(iteration), cost_list)
    plt.plot(range(iteration), accu_list)
    plt.show()

```

## 참고문헌

### 1. 나무위키 ‘인공지능’

<https://namu.wiki/w/%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5>

### 2. 위키백과 ‘딥러닝’

<https://ko.wikipedia.org/wiki/%ED%8B%B1%ED%83%9D%ED%86%A0>

### 3. 위키백과 ‘텐서플로’

<https://ko.wikipedia.org/wiki/%ED%85%90%EC%84%9C%ED%94%8C%EB%A1%9C>

### 4. 마팜스 블로그 ‘다층 퍼셉트론’

<https://mafams.tistory.com/15>

### 5. 예제로 배우는 파이썬 프로그래밍 ‘Matplotlib 차트/플롯 그리기’

<http://pythonstudy.xyz/python/article/407-Matplotlib-%EC%B0%A8%ED%8A%B8-%ED%94%8C%EB%A1%AF-%EA%B7%B8%EB%A6%AC%EA%B8%B0>