인공지능 과제 리포트

과제 제목: KNN and Logistic Regression with MNIST Data

학번: C011006

이름: 강유진

1. 과제 개요

숫자 0~9까지의 손글씨 이미지의 집합으로 학습데이터와 테스트데이터로 구성된 MNIST data를 저번 과제에 구현한 KNN과 Logistic Regression을 적용해 Classification을 수행한다

2. 구현 환경

Pycharm

3. 알고리즘에 대한 설명

KNN은 hw1에서 구현한 알고리즘 그대로 사용했으며 최적의 k값 도출은 for문을 통해 k를 돌리면서 구한다.

Logistic Regression은 cost function, gradient descent를 사용해 각 epoch 당 cost 값을 구한다. multiclass classification은 softmax함수를 이용해서 클래스 확률을 계산하고 cost function으로 모든 클래스에 대한 비용을 계산하고 gradient descent를 이용해 가중치를 업데이트하고 학습된 모델을 사용해 다중 클래스 분류 예측을 수행한다.

4. 데이터에 대한 설명

4.1 Input Feature

손글씨 이미지로 사이즈는 28X28이며 한 픽셀당 0~255 값을 가지고 있다

Flatten=True : 이미지를 1차원 배열로 읽는다

Normalize=True : 픽셀값이 0~255이 아니라 0~1이 된다

4.2 Target Output

0~9중 어떤 숫자의 손글씨인지 예측한다

5. 소스코드에 대한 설명

KNN의 경우 KNNclass.py는 저번 과제와 생성자와 run()빼고 동일하고

Using_KNN.py은 최적의 k값 도출은 for문을 통해 k를 돌리면서 구한다.

Logistic Regression의 경우 LogisticRegressionClass.py에 LogisticRegression 클래스, MultiClassLogisticRegression 클래스가 구현되어 있다.

Using_LogisticRegression.py에는 LogisticRegression을 구하는 것과 MultiClassLogistic을 구하는 내용이 들어있는데 이 둘을 구분, 따로 출력하기 위해 LogisticRegression을 구하는 코드에 주석처리되있다. 주석처리를 빼면 LogisticRegression을 구할 수 있다.

둘다 전체적인 구조는 cost function과 gradient descent 등을 이용해 cost값을 출력한다. 그 뒤 각 클래스에 대한 비용을 색깔별로 분류해 모든 클래스의 그래프가 결과물로 나오게 한다.

6. 학습 과정에 대한 설명

KNN은 hw1과 동일하며 학습하는 과정이 없다
Logistic Regression은 cost function과 gradient descent를 이용해 적절한 가중치와 bias를 찾는 것이 모델을 학습시키는 과정이다. 이는 7. 결과 및 분석을 통해 뭐가 더 안정적으로 학습이 되는지 볼 수 있다.

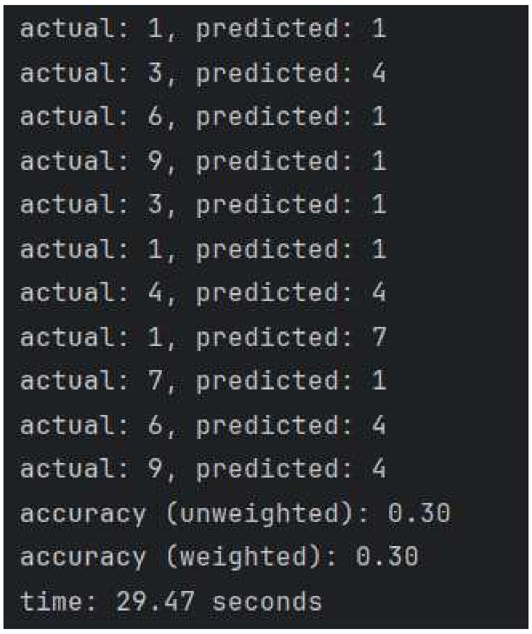
7. 결과 및 분석

1. KNN

1) 784개 input을 그대로 사용

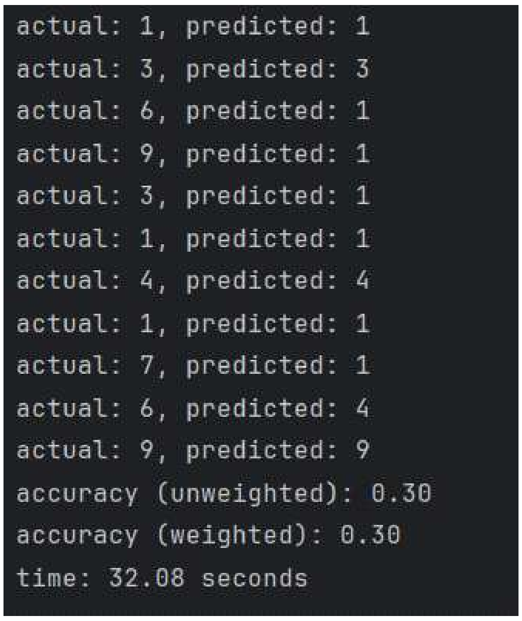
a. Normalize = False인 경우

k=1의 경우



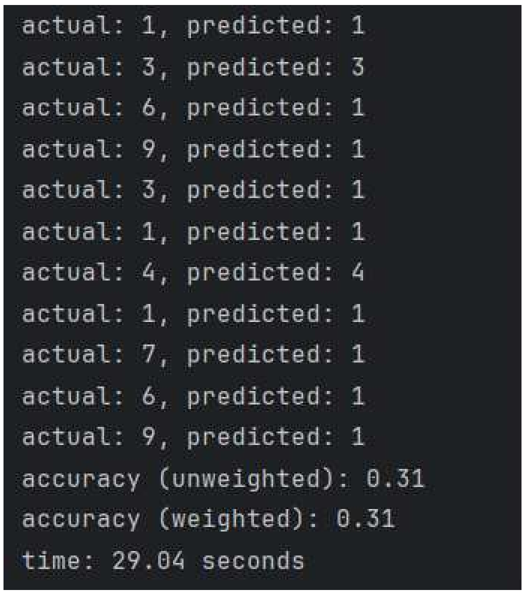
```
actual: 1, predicted: 1
actual: 3, predicted: 4
actual: 6, predicted: 1
actual: 9, predicted: 1
actual: 3, predicted: 1
actual: 1, predicted: 1
actual: 4, predicted: 4
actual: 1, predicted: 7
actual: 7, predicted: 1
actual: 6, predicted: 4
actual: 9, predicted: 4
accuracy (unweighted): 0.30
accuracy (weighted): 0.30
time: 29.47 seconds
```

k=5의 경우



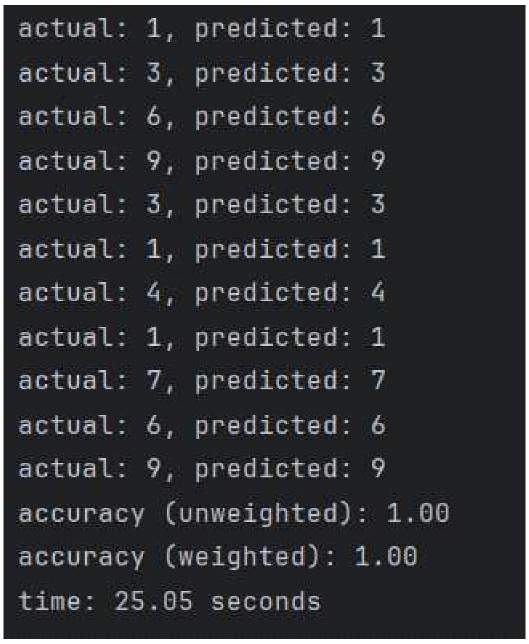
```
actual: 1, predicted: 1
actual: 3, predicted: 3
actual: 6, predicted: 1
actual: 9, predicted: 1
actual: 3, predicted: 1
actual: 1, predicted: 1
actual: 4, predicted: 4
actual: 1, predicted: 1
actual: 7, predicted: 1
actual: 6, predicted: 4
actual: 9, predicted: 9
accuracy (unweighted): 0.30
accuracy (weighted): 0.30
time: 32.08 seconds
```

k=10의 경우



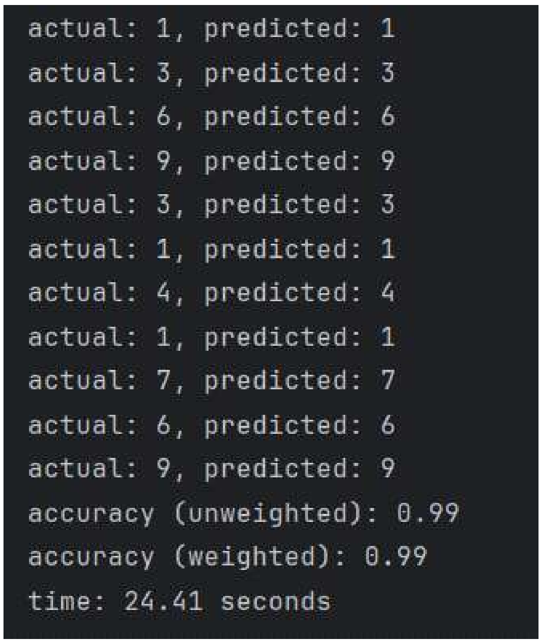
```
actual: 1, predicted: 1
actual: 3, predicted: 3
actual: 6, predicted: 1
actual: 9, predicted: 1
actual: 3, predicted: 1
actual: 1, predicted: 1
actual: 4, predicted: 4
actual: 1, predicted: 1
actual: 7, predicted: 1
actual: 6, predicted: 1
actual: 9, predicted: 1
accuracy (unweighted): 0.31
accuracy (weighted): 0.31
time: 29.04 seconds
```

b. Normalize = True인 경우
k=1의 경우

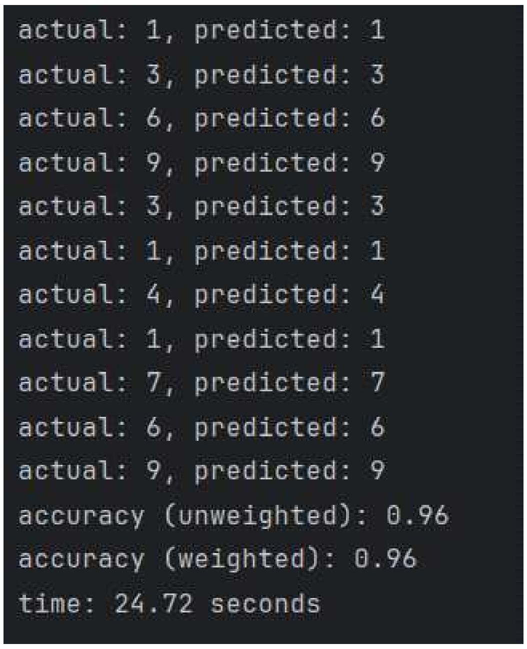


```
actual: 1, predicted: 1
actual: 3, predicted: 3
actual: 6, predicted: 6
actual: 9, predicted: 9
actual: 3, predicted: 3
actual: 1, predicted: 1
actual: 4, predicted: 4
actual: 1, predicted: 1
actual: 7, predicted: 7
actual: 6, predicted: 6
actual: 9, predicted: 9
accuracy (unweighted): 1.00
accuracy (weighted): 1.00
time: 25.05 seconds
```

k=5의 경우


```
actual: 1, predicted: 1
actual: 3, predicted: 3
actual: 6, predicted: 6
actual: 9, predicted: 9
actual: 3, predicted: 3
actual: 1, predicted: 1
actual: 4, predicted: 4
actual: 1, predicted: 1
actual: 7, predicted: 7
actual: 6, predicted: 6
actual: 9, predicted: 9
accuracy (unweighted): 0.99
accuracy (weighted): 0.99
time: 24.41 seconds
```

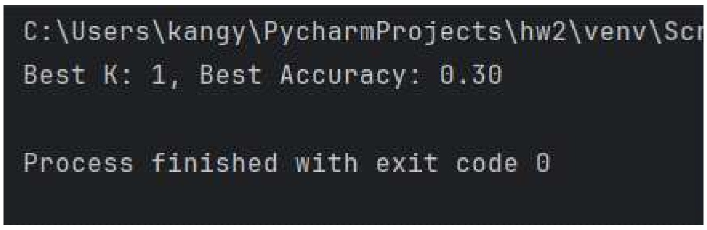
k=10의 경우


```
actual: 1, predicted: 1
actual: 3, predicted: 3
actual: 6, predicted: 6
actual: 9, predicted: 9
actual: 3, predicted: 3
actual: 1, predicted: 1
actual: 4, predicted: 4
actual: 1, predicted: 1
actual: 7, predicted: 7
actual: 6, predicted: 6
actual: 9, predicted: 9
accuracy (unweighted): 0.96
accuracy (weighted): 0.96
time: 24.72 seconds
```

2) 최적의 K값 도출

a. Normalize = False인 경우

k값의 범위 1부터 9까지, 테스트 데이터 100개

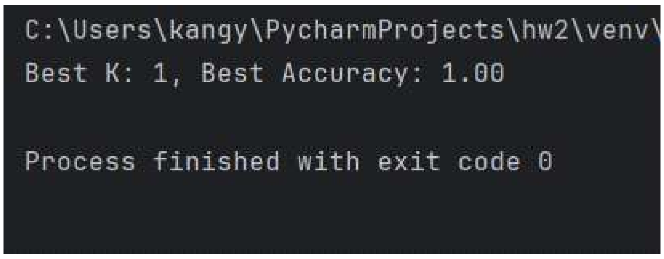

```
C:\Users\kangy\PycharmProjects\hw2\venv\Scr
Best K: 1, Best Accuracy: 0.30

Process finished with exit code 0
```

Best K = 1 / Accuracy : 0.3

b. Normalize = True인 경우

k값의 범위 1부터 9까지, 테스트 데이터 100개


```
C:\Users\kangy\PycharmProjects\hw2\venv\
Best K: 1, Best Accuracy: 1.00

Process finished with exit code 0
```

Best K = 1 / Accuracy : 1.0

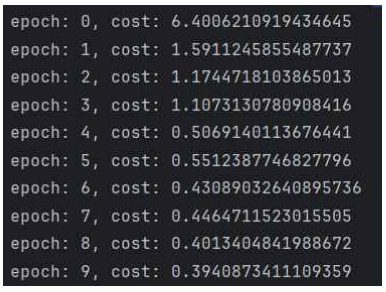
1)의 결과를 보았을 때 2)에서 best K의 값일 때 정확도가 가장 높은 것을 볼 수 있다.
또한 Normalize를 하기 전에는 best accuracy가 0.3일 정도로 낮지만 Normalize 후에는 accuracy가 1에 가까운 수가 나온다. 즉 Normalize를 해서 같은 위치의 픽셀끼리 비슷한 분산으로 정규화되어 성능이 향상된다는 사실을 알 수 있다.

2. Logistic Regression

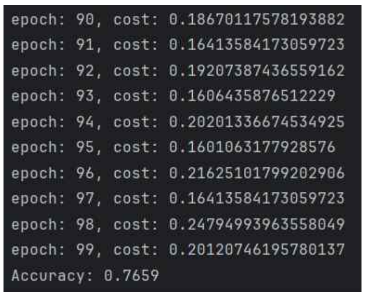
1) Single Class

a. Target = 0인 경우

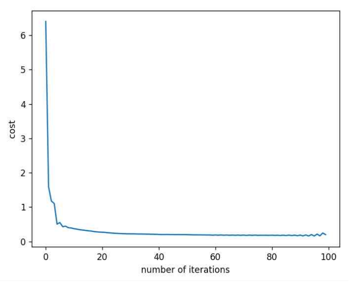
Normalize = False, Epoch = 100, lr = 0.01



```
epoch: 0, cost: 6.4006210919434645
epoch: 1, cost: 1.5911245855487737
epoch: 2, cost: 1.1744718103865013
epoch: 3, cost: 1.1073130780908416
epoch: 4, cost: 0.5069140113676441
epoch: 5, cost: 0.5512387746827796
epoch: 6, cost: 0.43089032640895736
epoch: 7, cost: 0.4464711523015505
epoch: 8, cost: 0.4013404841988672
epoch: 9, cost: 0.3940873411109359
```

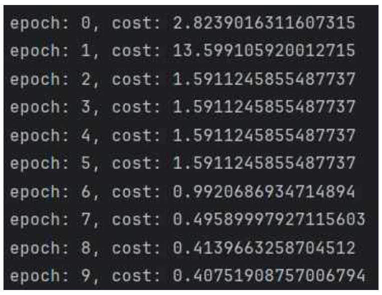


```
epoch: 90, cost: 0.18670117578193882
epoch: 91, cost: 0.16413584173059723
epoch: 92, cost: 0.19207387436559162
epoch: 93, cost: 0.1606435876512229
epoch: 94, cost: 0.20201336674534925
epoch: 95, cost: 0.1601063177928576
epoch: 96, cost: 0.21625101799202906
epoch: 97, cost: 0.16413584173059723
epoch: 98, cost: 0.24794993963558049
epoch: 99, cost: 0.20120746195780137
Accuracy: 0.7659
```

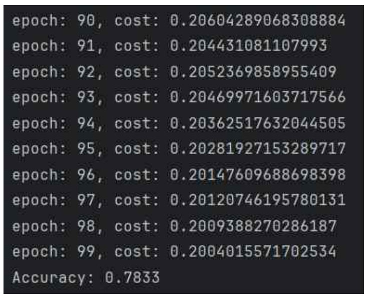


Normalize가 False임에도 Accuracy가 크게 측정된다

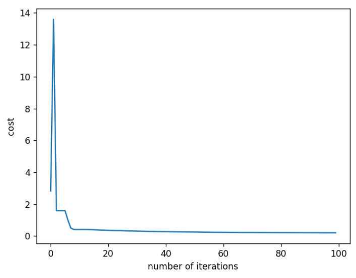
Normalize = False, Epoch = 100, lr = 0.1



```
epoch: 0, cost: 2.8239016311607315
epoch: 1, cost: 13.599105920012715
epoch: 2, cost: 1.5911245855487737
epoch: 3, cost: 1.5911245855487737
epoch: 4, cost: 1.5911245855487737
epoch: 5, cost: 1.5911245855487737
epoch: 6, cost: 0.9920686934714894
epoch: 7, cost: 0.49589997927115603
epoch: 8, cost: 0.4139663258704512
epoch: 9, cost: 0.40751908757006794
```

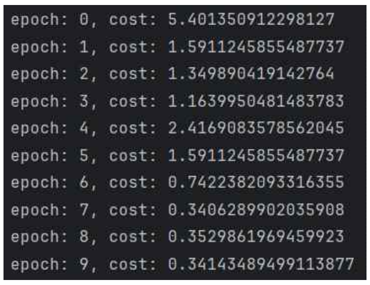


```
epoch: 90, cost: 0.20604289068308884
epoch: 91, cost: 0.204431081107993
epoch: 92, cost: 0.2052369858955409
epoch: 93, cost: 0.20469971603717566
epoch: 94, cost: 0.20362517632044505
epoch: 95, cost: 0.20281927153289717
epoch: 96, cost: 0.20147609688698398
epoch: 97, cost: 0.20120746195780131
epoch: 98, cost: 0.2009388270286187
epoch: 99, cost: 0.2004015571702534
Accuracy: 0.7833
```

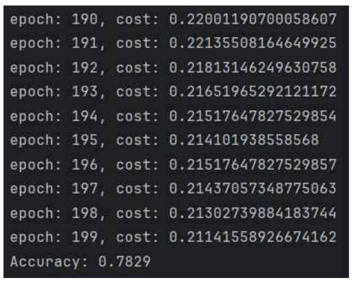


학습률 lr을 증가시키면 cost값이 빠르게 감소할 수 있으며, 작게하면 더 안정적으로 수렴할 수 있다.

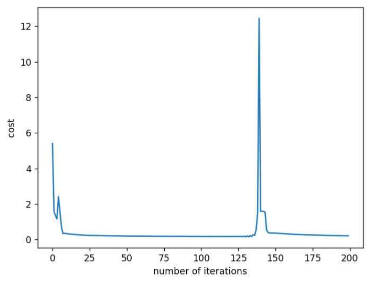
Normalize = False, Epoch = 200, lr = 0.1



```
epoch: 0, cost: 5.401350912298127
epoch: 1, cost: 1.5911245855487737
epoch: 2, cost: 1.349890419142764
epoch: 3, cost: 1.1639950481483783
epoch: 4, cost: 2.4169083578562045
epoch: 5, cost: 1.5911245855487737
epoch: 6, cost: 0.7422382093316355
epoch: 7, cost: 0.3406289902035908
epoch: 8, cost: 0.3529861969459923
epoch: 9, cost: 0.34143489499113877
```

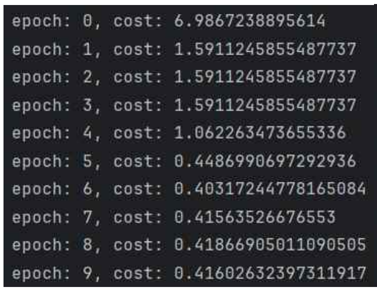


```
epoch: 190, cost: 0.22001190700058607
epoch: 191, cost: 0.22135508164649925
epoch: 192, cost: 0.21813146249630758
epoch: 193, cost: 0.21651965292121172
epoch: 194, cost: 0.21517647827529854
epoch: 195, cost: 0.214101938558568
epoch: 196, cost: 0.21517647827529857
epoch: 197, cost: 0.21437057348775063
epoch: 198, cost: 0.21302739884183744
epoch: 199, cost: 0.21141558926674162
Accuracy: 0.7829
```

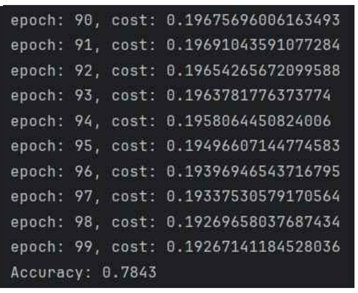


epoch를 증가하면 모델은 더 많이 훈련되어 cost값이 감소할 수 있다
하지만 이 경우 중간에 cost값이 튀어버리는 경우가 생긴다

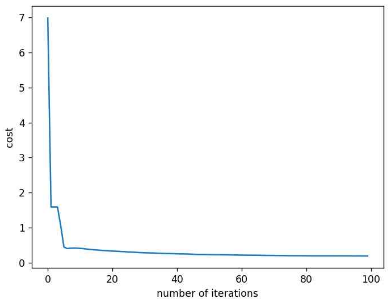
Normalize = True, Epoch = 100, lr = 0.01



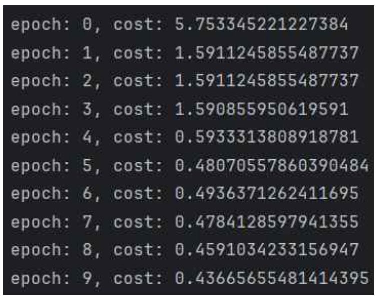
```
epoch: 0, cost: 6.9867238895614
epoch: 1, cost: 1.5911245855487737
epoch: 2, cost: 1.5911245855487737
epoch: 3, cost: 1.5911245855487737
epoch: 4, cost: 1.062263473655336
epoch: 5, cost: 0.4486990697292936
epoch: 6, cost: 0.40317244778165084
epoch: 7, cost: 0.41563526676553
epoch: 8, cost: 0.41866905011090505
epoch: 9, cost: 0.41602632397311917
```



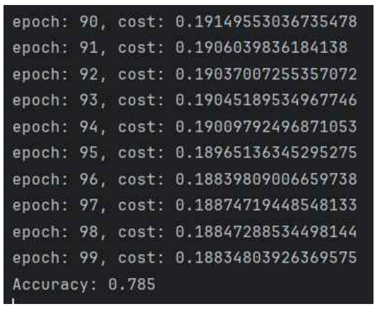
```
epoch: 90, cost: 0.19675696006163493
epoch: 91, cost: 0.19691043591077284
epoch: 92, cost: 0.19654265672099588
epoch: 93, cost: 0.1963781776373774
epoch: 94, cost: 0.1958064450824006
epoch: 95, cost: 0.19496607144774583
epoch: 96, cost: 0.19396946543716795
epoch: 97, cost: 0.19337530579170564
epoch: 98, cost: 0.19269658037687434
epoch: 99, cost: 0.19267141184528036
Accuracy: 0.7843
```



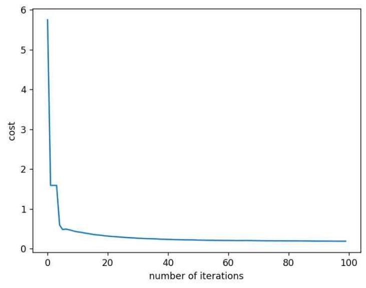
Normalize = True, Epoch = 100, lr = 0.1



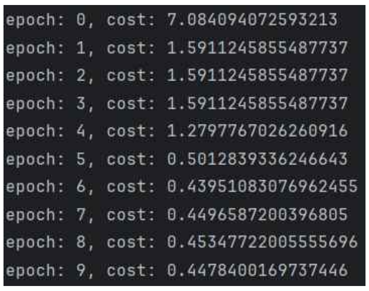
```
epoch: 0, cost: 5.753345221227384
epoch: 1, cost: 1.5911245855487737
epoch: 2, cost: 1.5911245855487737
epoch: 3, cost: 1.590855950619591
epoch: 4, cost: 0.5933313808918781
epoch: 5, cost: 0.48070557860390484
epoch: 6, cost: 0.4936371262411695
epoch: 7, cost: 0.4784128597941355
epoch: 8, cost: 0.4591034233156947
epoch: 9, cost: 0.43665655481414395
```



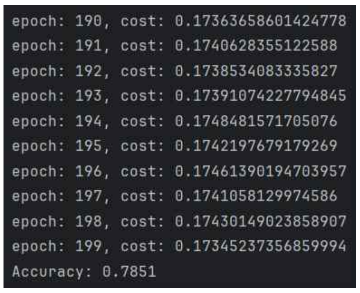
```
epoch: 90, cost: 0.19149553036735478
epoch: 91, cost: 0.1906039836184138
epoch: 92, cost: 0.19037007255357072
epoch: 93, cost: 0.19045189534967746
epoch: 94, cost: 0.19009792496871053
epoch: 95, cost: 0.18965136345295275
epoch: 96, cost: 0.18839809006659738
epoch: 97, cost: 0.18874719448548133
epoch: 98, cost: 0.18847288534498144
epoch: 99, cost: 0.18834803926369575
Accuracy: 0.785
```



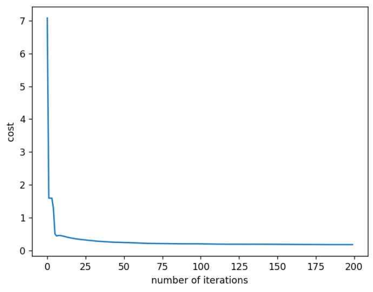
Normalize = True, Epoch = 200, lr = 0.1



```
epoch: 0, cost: 7.084094072593213
epoch: 1, cost: 1.5911245855487737
epoch: 2, cost: 1.5911245855487737
epoch: 3, cost: 1.5911245855487737
epoch: 4, cost: 1.2797767026260916
epoch: 5, cost: 0.5012839336246643
epoch: 6, cost: 0.43951083076962455
epoch: 7, cost: 0.4496587200396805
epoch: 8, cost: 0.45347722005555696
epoch: 9, cost: 0.4478400169737446
```



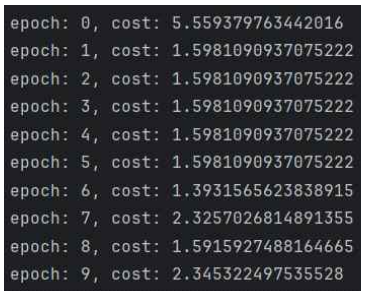
```
epoch: 190, cost: 0.17363658601424778
epoch: 191, cost: 0.1740628355122588
epoch: 192, cost: 0.1738534083335827
epoch: 193, cost: 0.17391074227794845
epoch: 194, cost: 0.1748481571705076
epoch: 195, cost: 0.1742197679179269
epoch: 196, cost: 0.17461390194703957
epoch: 197, cost: 0.1741058129974586
epoch: 198, cost: 0.17430149023858907
epoch: 199, cost: 0.17345237356859994
Accuracy: 0.7851
```



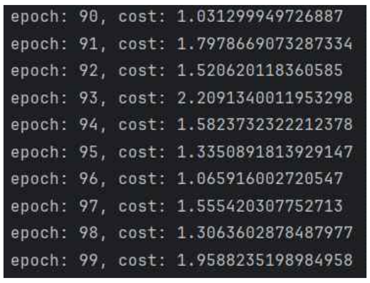
Normalize된 데이터는 Normalize=False보다 cost값을 낮출 수 있다

b. Target = 9인 경우

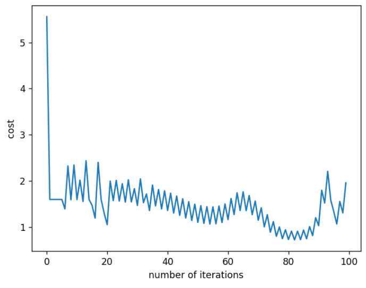
Normalize = True, Epoch = 100, lr = 0.01



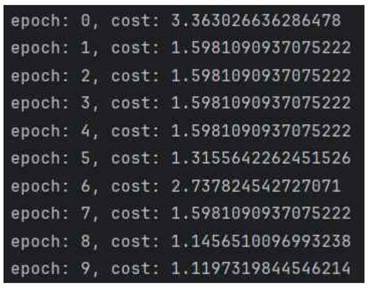
```
epoch: 0, cost: 5.559379763442016
epoch: 1, cost: 1.5981090937075222
epoch: 2, cost: 1.5981090937075222
epoch: 3, cost: 1.5981090937075222
epoch: 4, cost: 1.5981090937075222
epoch: 5, cost: 1.5981090937075222
epoch: 6, cost: 1.3931565623838915
epoch: 7, cost: 2.3257026814891355
epoch: 8, cost: 1.5915927488164665
epoch: 9, cost: 2.345322497535528
```



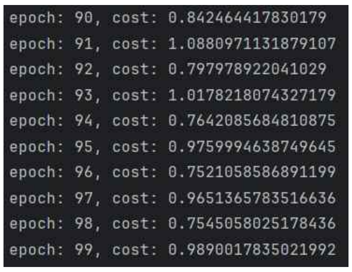
```
epoch: 90, cost: 1.031299949726887
epoch: 91, cost: 1.7978669073287334
epoch: 92, cost: 1.520620118360585
epoch: 93, cost: 2.2091340011953298
epoch: 94, cost: 1.5823732322212378
epoch: 95, cost: 1.3350891813929147
epoch: 96, cost: 1.065916002720547
epoch: 97, cost: 1.555420307752713
epoch: 98, cost: 1.3063602878487977
epoch: 99, cost: 1.9588235198984958
```



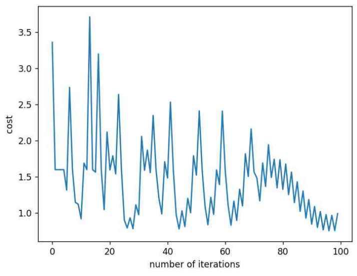
Normalize = True, Epoch = 100, lr = 0.1



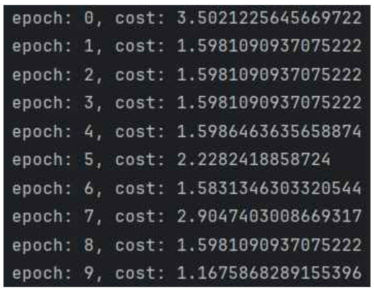
```
epoch: 0, cost: 3.363026636286478
epoch: 1, cost: 1.5981090937075222
epoch: 2, cost: 1.5981090937075222
epoch: 3, cost: 1.5981090937075222
epoch: 4, cost: 1.5981090937075222
epoch: 5, cost: 1.3155642262451526
epoch: 6, cost: 2.737824542727071
epoch: 7, cost: 1.5981090937075222
epoch: 8, cost: 1.1456510096993238
epoch: 9, cost: 1.1197319844546214
```



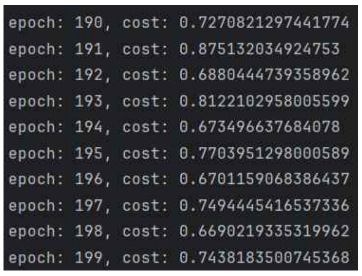
```
epoch: 90, cost: 0.842464417830179
epoch: 91, cost: 1.0880971131879107
epoch: 92, cost: 0.797978922041029
epoch: 93, cost: 1.0178218074327179
epoch: 94, cost: 0.7642085684810875
epoch: 95, cost: 0.9759994638749645
epoch: 96, cost: 0.7521058586891199
epoch: 97, cost: 0.9651365783516636
epoch: 98, cost: 0.7545058025178436
epoch: 99, cost: 0.9890017835021992
```



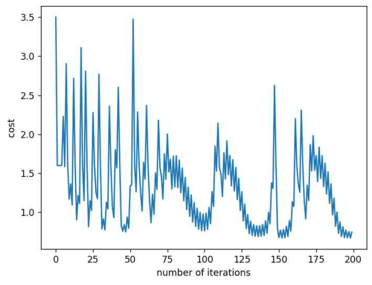
Normalize = True, Epoch = 200, lr = 0.1



```
epoch: 0, cost: 3.5021225645669722
epoch: 1, cost: 1.5981090937075222
epoch: 2, cost: 1.5981090937075222
epoch: 3, cost: 1.5981090937075222
epoch: 4, cost: 1.5986463635658874
epoch: 5, cost: 2.2282418858724
epoch: 6, cost: 1.5831346303320544
epoch: 7, cost: 2.9047403008669317
epoch: 8, cost: 1.5981090937075222
epoch: 9, cost: 1.1675868289155396
```



```
epoch: 190, cost: 0.7270821297441774
epoch: 191, cost: 0.875132034924753
epoch: 192, cost: 0.6880444739358962
epoch: 193, cost: 0.8122102958005599
epoch: 194, cost: 0.673496637684078
epoch: 195, cost: 0.7703951298000589
epoch: 196, cost: 0.6701159068386437
epoch: 197, cost: 0.7494445416537336
epoch: 198, cost: 0.6690219335319962
epoch: 199, cost: 0.7438183500745368
```

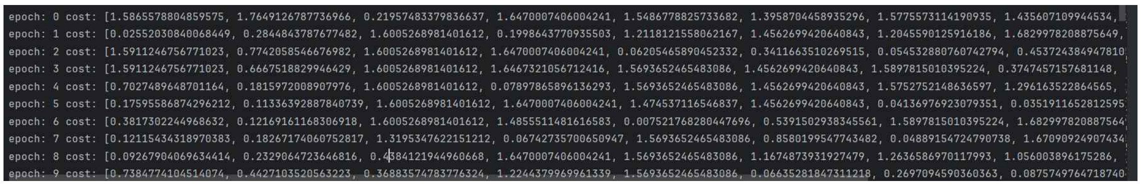


target을 0에서 9로 증가하면서 그래프값, 데이터의 클래스 불균형이 발생하는 것을 알 수 있다. 예를 들어 class 3이 다른 class에 비해 더 많이 나타날 때, class 3으로 예측하면 정확도는 높아지지만, 다른 class들에 대한 예측은 부정확하다

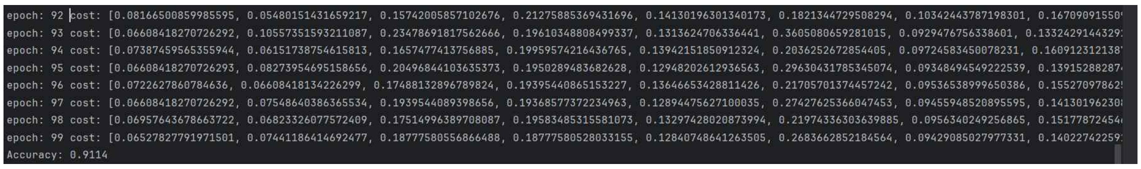
2) Multi-class

a. Normalize = False인 경우

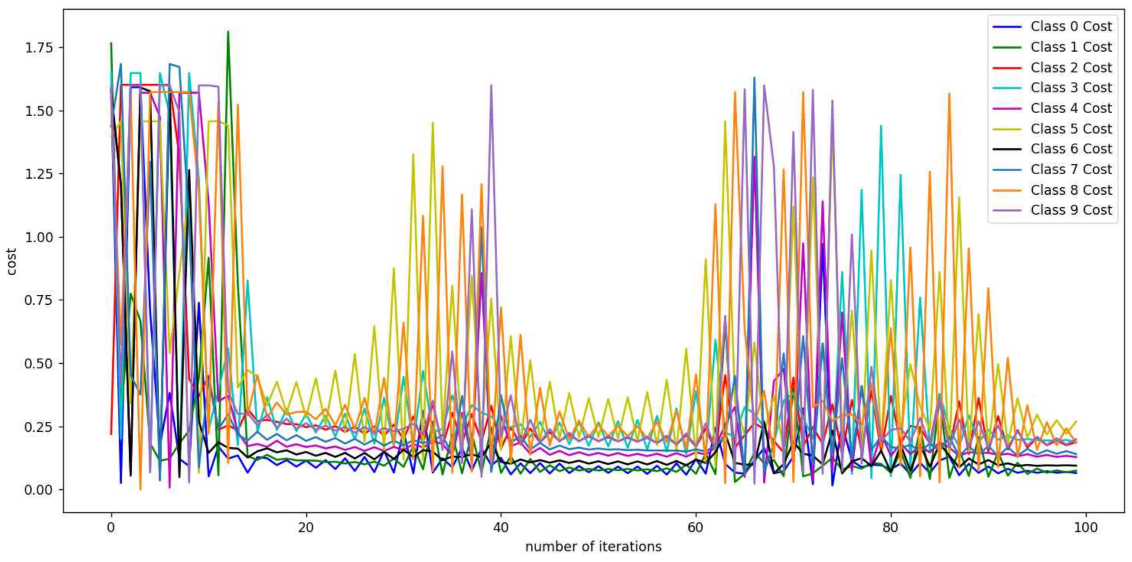
Normalize = False, Epoch = 100, lr = 0.1



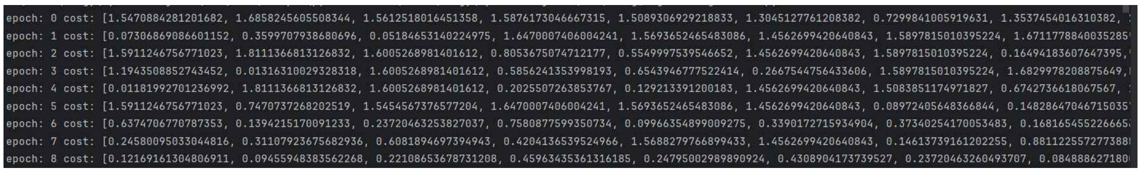
```
epoch: 0 cost: [1.5865578804859575, 1.7649126787736966, 0.21957483379836637, 1.6470007406004241, 1.5486778825733682, 1.3958704458935296, 1.5775573114190935, 1.435607109944534,
epoch: 1 cost: [0.02552030840068449, 0.2844843787677482, 1.6005268981401612, 0.1998643770935503, 1.2118121558062167, 1.4562699420640843, 1.2045590125916186, 1.6829978208875649,
epoch: 2 cost: [1.5911246756771023, 0.7742058546676982, 1.6005268981401612, 1.6470007406004241, 0.06205465890452332, 0.3411663510269515, 0.054532880760742794, 0.453724384947810
epoch: 3 cost: [1.5911246756771023, 0.6667518829946429, 1.6005268981401612, 1.6467321056712416, 1.5693652465483086, 1.4562699420640843, 1.5897815010395224, 0.3747457157681148,
epoch: 4 cost: [0.7027489648701164, 0.1815972008907976, 1.6005268981401612, 0.07897865896136293, 1.5693652465483086, 1.4562699420640843, 1.5752752148636597, 1.296163522864565,
epoch: 5 cost: [0.17595586874296212, 0.11336392887840739, 1.6005268981401612, 1.6470007406004241, 1.474537116546837, 1.4562699420640843, 0.04136976923079351, 0.0351911652812595
epoch: 6 cost: [0.3817302244968632, 0.12169161168306918, 1.6005268981401612, 1.4855511481616583, 0.007521768280447696, 0.5391502938345561, 1.5897815010395224, 1.682997820887564
epoch: 7 cost: [0.12115434318970383, 0.18267174060752817, 1.3195347622151212, 0.06742735700650947, 1.5693652465483086, 0.8580199547743482, 0.04889154724790738, 1.67090924907434
epoch: 8 cost: [0.09267904069634414, 0.2329064723646816, 0.4384121944960668, 1.6470007406004241, 1.5693652465483086, 1.1674873931927479, 1.2636586970117993, 1.056003896175286,
epoch: 9 cost: [0.7384774104514074, 0.4427103520563223, 0.36883574783776324, 1.2244379969961339, 1.5693652465483086, 0.06635281847311218, 0.2697094590360363, 0.0875749764718740
```



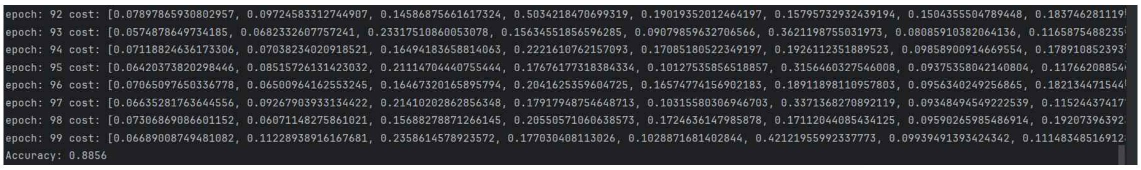
```
epoch: 92 cost: [0.08166500859985595, 0.05480151431659217, 0.15742005857102676, 0.21275885369431696, 0.14130196301340173, 0.1821344729508294, 0.10342443787198301, 0.16709091550
epoch: 93 cost: [0.06608418270726292, 0.10557351593211087, 0.23478691817562666, 0.19610348808499337, 0.1313624706336441, 0.3605080659281015, 0.0929476756338601, 0.133242914432
epoch: 94 cost: [0.07387459565355944, 0.06151738754615813, 0.1657477413756885, 0.19959574216436765, 0.13942151850912324, 0.2036252672854405, 0.09724583450078231, 0.160912312138
epoch: 95 cost: [0.06608418270726293, 0.08273954695158656, 0.20496844103635373, 0.1950289483682628, 0.12948202612936563, 0.29630431785345074, 0.09348494549222539, 0.13915288287
epoch: 96 cost: [0.0722627860784636, 0.06608418134226299, 0.17488132896789824, 0.19395440865153227, 0.13646653428811426, 0.21705701374457242, 0.09536538999650386, 0.15527097862
epoch: 97 cost: [0.06608418270726292, 0.07548640386365534, 0.1939544089398656, 0.19368577372234963, 0.12894475627100035, 0.27427625366047453, 0.09455948520895595, 0.14130196230
epoch: 98 cost: [0.06957643678663722, 0.06823326077572409, 0.17514996389708087, 0.19583485315581073, 0.13297428020873994, 0.21974336303639885, 0.0956340249256865, 0.15177872454
epoch: 99 cost: [0.06527827791971501, 0.07441186414692477, 0.18777580556866488, 0.18777580528033155, 0.12840748641263505, 0.2683662852184564, 0.09429085027977331, 0.14022742259
Accuracy: 0.9114
```



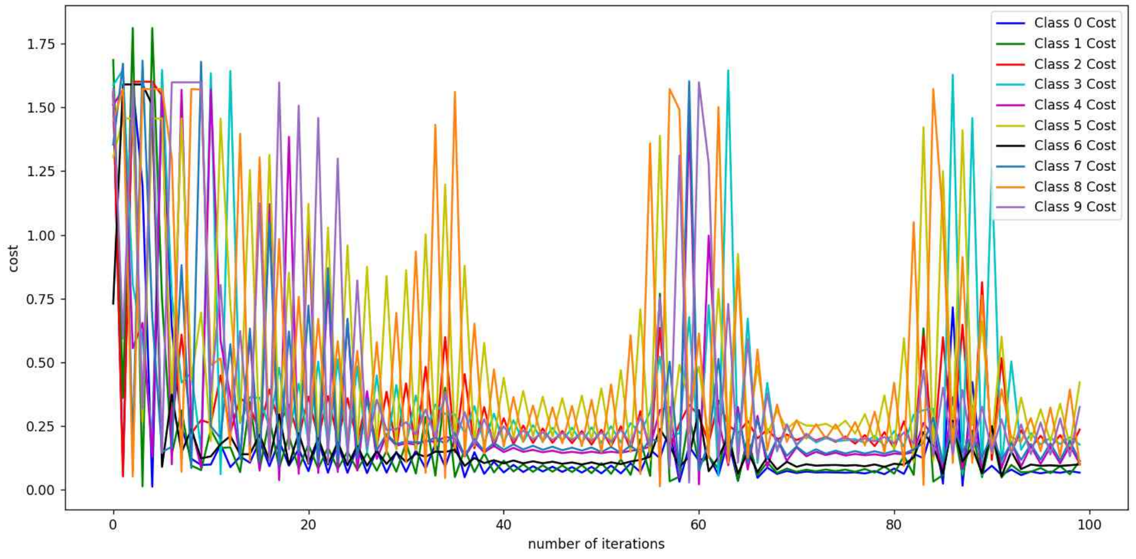
Normalize = False, Epoch = 100, lr = 0.01



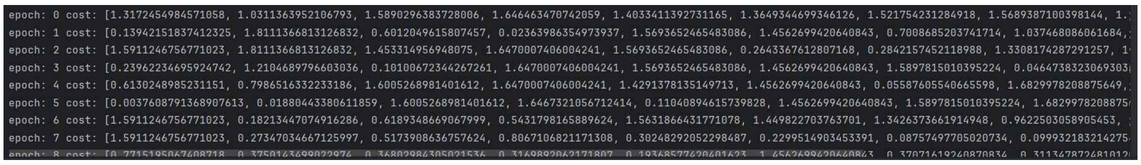
```
epoch: 0 cost: [1.5470884281201682, 1.6858245605508344, 1.5612518016451358, 1.5876173046667315, 1.5089306929218833, 1.3045127761208382, 0.7299841005919631, 1.3537454016310382,
epoch: 1 cost: [0.07306869086601152, 0.3599707938680696, 0.05184653140224975, 1.6470007406004241, 1.5693652465483086, 1.4562699420640843, 1.5897815010395224, 1.6711778840035285
epoch: 2 cost: [1.5911246756771023, 1.8111366813126832, 1.6005268981401612, 0.8053675074712177, 0.5549997539546652, 1.4562699420640843, 1.5897815010395224, 0.16494183607647395,
epoch: 3 cost: [1.1943508852743452, 0.01316310029328318, 1.6005268981401612, 0.5856241353998193, 0.6543946777522414, 0.2667544756433606, 1.5897815010395224, 1.6829978208875649,
epoch: 4 cost: [0.01181992701236992, 1.8111366813126832, 1.6005268981401612, 0.2025507263853767, 0.129213391200183, 1.4562699420640843, 1.5083851174971827, 0.6742736618067567,
epoch: 5 cost: [1.5911246756771023, 0.7470737268202519, 1.5454567376577204, 1.6470007406004241, 1.5693652465483086, 1.4562699420640843, 0.08972405648366844, 0.14828647046715035
epoch: 6 cost: [0.6374706770787353, 0.1394215170091233, 0.23720463253827037, 0.7580877599350734, 0.09966354899009275, 0.3390172715934904, 0.37340254170053483, 0.168165455226665
epoch: 7 cost: [0.24580095033044816, 0.31107923675682936, 0.6081894697394943, 0.4204136539524966, 1.5688279766899433, 1.4562699420640843, 0.14613739161202255, 0.881122557277388
epoch: 8 cost: [0.12169161304806911, 0.09455948383562268, 0.22108653678731208, 0.45963435361316185, 0.24795002989890924, 0.4308904173739527, 0.23720463260493707, 0.084888627180
```



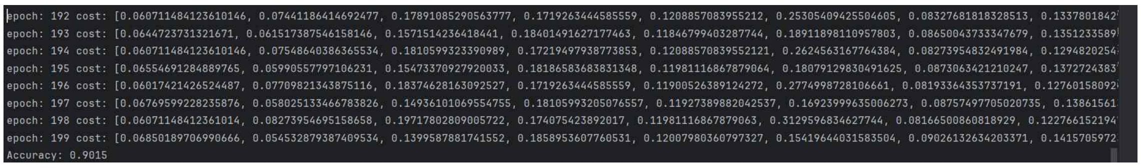
```
epoch: 92 cost: [0.07897865930802957, 0.09724583312744907, 0.14586875661617324, 0.5034218470699319, 0.19019352012464197, 0.15795732932439194, 0.1504355504789448, 0.183746281119
epoch: 93 cost: [0.0574878649734185, 0.0682332607757241, 0.23317510860053078, 0.15634551856596285, 0.09079859632706566, 0.3621198755031973, 0.08085910382064136, 0.1165875488235
epoch: 94 cost: [0.07118824636173306, 0.07038234020918521, 0.16494183658814063, 0.2221610762157093, 0.17085180522349197, 0.1926112351889523, 0.09858900914669554, 0.178910852393
epoch: 95 cost: [0.06420373820298446, 0.08515726131423032, 0.21114704440755444, 0.17676177318384334, 0.10127535856518857, 0.3156460327546008, 0.09375358042140804, 0.11766208854
epoch: 96 cost: [0.07065097650336778, 0.06500964162553245, 0.16467320165895794, 0.2041625359604725, 0.16574774156902183, 0.18911898110957803, 0.0956340249256865, 0.182134471544
epoch: 97 cost: [0.06635281763644556, 0.09267903933134422, 0.21410202862856348, 0.17917948754648713, 0.10315580306946703, 0.3371368270892119, 0.09348494549222539, 0.11524437417
epoch: 98 cost: [0.07306869086601152, 0.06071148275861021, 0.15688278871266145, 0.20550571060638573, 0.1724636147985878, 0.17112044085434125, 0.09590265985486914, 0.19207396392
epoch: 99 cost: [0.06689008749481082, 0.11228938916167681, 0.2358614578923572, 0.177030408113026, 0.1028871681402844, 0.42121955992337773, 0.09939491393424342, 0.11148348516912
Accuracy: 0.8856
```



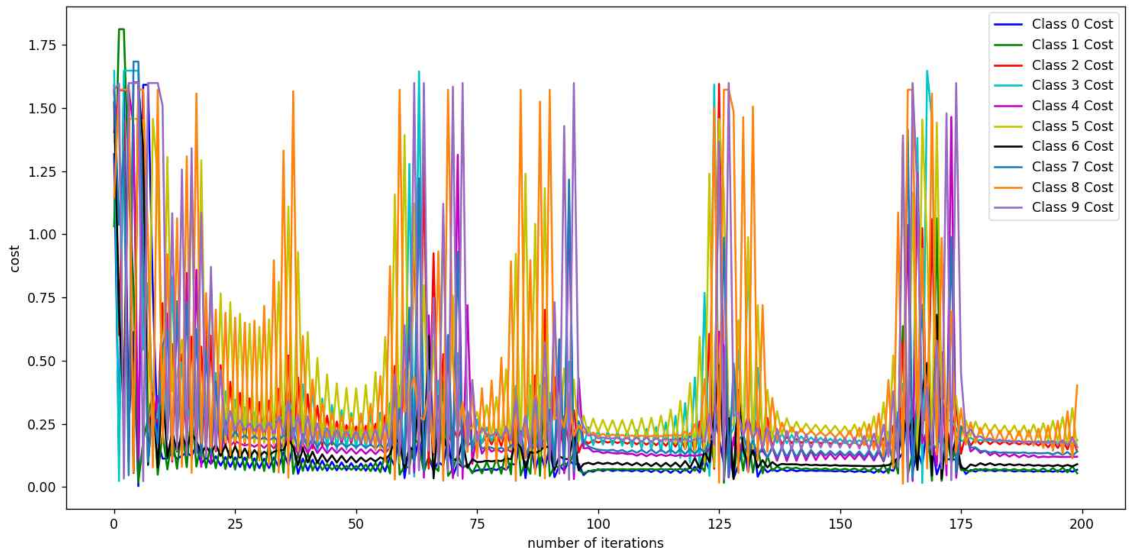
Normalize = False, Epoch = 200, lr = 0.01



```
epoch: 0 cost: [1.3172454984571058, 1.0311363952106793, 1.5890296383728006, 1.646463470742059, 1.4033411392731165, 1.3649344699346126, 1.521754231284918, 1.5689387100398144, 1.
epoch: 1 cost: [0.13942151837412325, 1.8111366813126832, 0.6012049615807457, 0.02363986354973937, 1.5693652465483086, 1.4562699420640843, 0.7008685203741714, 1.037468086061684,
epoch: 2 cost: [1.5911246756771023, 1.8111366813126832, 1.453314956948075, 1.6470007406004241, 1.5693652465483086, 0.2643367612807168, 0.2842157452118988, 1.3308174287291257, 1
epoch: 3 cost: [0.23962234695924742, 1.2104689796603036, 0.10100672344267261, 1.6470007406004241, 1.5693652465483086, 1.4562699420640843, 1.5897815010395224, 0.0464738323069303
epoch: 4 cost: [0.6130248985231151, 0.7986516332233186, 1.6005268981401612, 1.6470007406004241, 1.4291378135149713, 1.4562699420640843, 0.05587605540665598, 1.6829978208875649,
epoch: 5 cost: [0.0037608791368907613, 0.01880443380611859, 1.6005268981401612, 1.6467321056712414, 0.11040894615739828, 1.4562699420640843, 1.5897815010395224, 1.682997820887
epoch: 6 cost: [1.5911246756771023, 0.18213447074916286, 0.6189348669067999, 0.5431798165889624, 1.5631866431771078, 1.449822703763701, 1.3426373661914948, 0.9622503058905453,
epoch: 7 cost: [1.5911246756771023, 0.27347034667125997, 0.5173908636757624, 0.8067106821171308, 0.30248292052298487, 0.2299514903453391, 0.08757497705020734, 0.099932183214275
epoch: 8 cost: [0.7715195067408718, 0.3750143499022974, 0.36802984305021536, 0.3169892062171807, 0.19368577420601623, 1.4562699420640843, 0.37071619240870834, 0.311347872481012
```



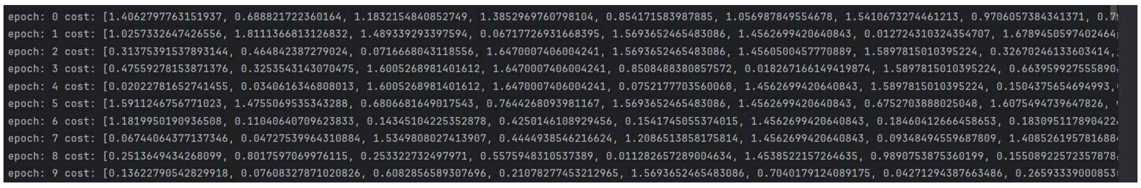
```
epoch: 192 cost: [0.060711484123610146, 0.07441186414692477, 0.17891085290563777, 0.1719263444585559, 0.1208857083955212, 0.25305409425504605, 0.08327681818328513, 0.1337801842
epoch: 193 cost: [0.06447237313216711, 0.061517387546158146, 0.1571514236418441, 0.18401491627177463, 0.11846799403287744, 0.18911898110957803, 0.08650043733347679, 0.1351233589
epoch: 194 cost: [0.060711484123610146, 0.07548640386365534, 0.1810599323390989, 0.17219497938773853, 0.12088570839552121, 0.2624563167764384, 0.08273954832491984, 0.1294820254
epoch: 195 cost: [0.06554691284889765, 0.05990557797106231, 0.15473370927920033, 0.18186583683831348, 0.11981116867879064, 0.18079129830491625, 0.0873063421210247, 0.1372724383
epoch: 196 cost: [0.06017421426524487, 0.07709821343875116, 0.18374628163092527, 0.1719263444585559, 0.11900526389124272, 0.2774998728106661, 0.08193364353737191, 0.12760158092
epoch: 197 cost: [0.06769599228235876, 0.058025133466783826, 0.14936101069554755, 0.18105993205076557, 0.11927389882042537, 0.16923999635006273, 0.08757497705020735, 0.13861561
epoch: 198 cost: [0.06071148412361014, 0.08273954695158658, 0.19717802809005722, 0.174075423892017, 0.11981116867879063, 0.3129596834627744, 0.08166500860818929, 0.122766152194
epoch: 199 cost: [0.06850189706990666, 0.054532879387409534, 0.1399587881741552, 0.1858953607760531, 0.12007980360797327, 0.15419644031583504, 0.09026132634203371, 0.1415705972
Accuracy: 0.9015
```



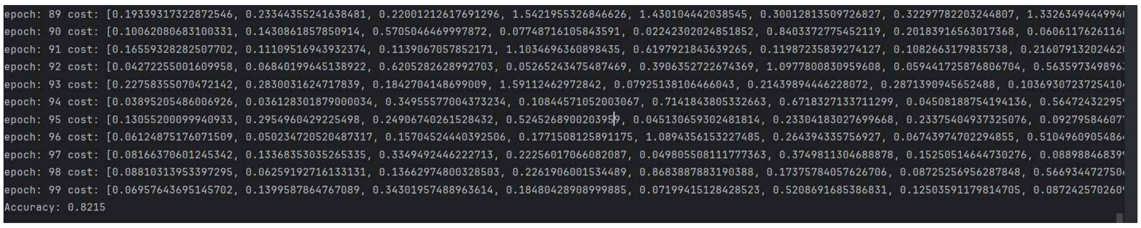
epoch를 증가해 모델이 더 많은 훈련 데이터를 볼 수 있다

b. Normalize = True인 경우

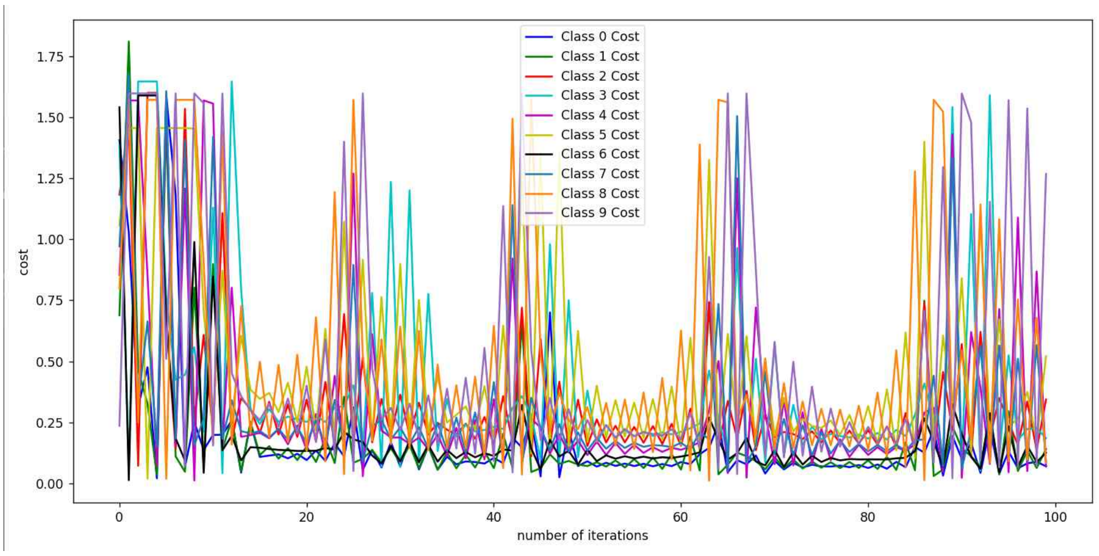
Normalize = True, Epoch = 100, lr = 0.1



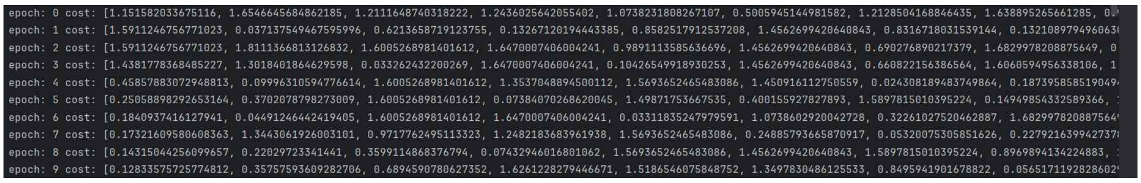
```
epoch: 0 cost: [1.4062797763151937, 0.688821722360164, 1.1832154840852749, 1.3852969760798104, 0.854171583987885, 1.056987849554678, 1.5410673274461213, 0.9706057384341371, 0.7
epoch: 1 cost: [1.0257332647426556, 1.8111366813126832, 1.489339293397594, 0.06717726931668395, 1.5693652465483086, 1.4562699420640843, 0.012724310324354707, 1.678945059740246
epoch: 2 cost: [0.31375391537893144, 0.464842387279024, 0.0716668043118556, 1.6470007406004241, 1.5693652465483086, 1.4560500457770889, 1.5897815010395224, 0.3267024613360341
epoch: 3 cost: [0.47559278153871376, 0.3253543143070475, 1.6005268981401612, 1.6470007406004241, 0.8508488380857572, 0.018267166149419874, 1.5897815010395224, 0.663959927555890
epoch: 4 cost: [0.02022781652741455, 0.0340616346808013, 1.6005268981401612, 1.6470007406004241, 0.0752177703560068, 1.4562699420640843, 1.5897815010395224, 0.1504375654694993,
epoch: 5 cost: [1.5911246756771023, 1.4755069535343288, 0.6806681649017543, 0.7644268093981167, 1.5693652465483086, 1.4562699420640843, 0.6752703888025048, 1.6075494739647826,
epoch: 6 cost: [1.1819950190936508, 0.1104064070962383, 0.14345104225352878, 0.4250146108929456, 0.1541745055374015, 1.4562699420640843, 0.18460412666458653, 0.183095117890422
epoch: 7 cost: [0.06744064377137346, 0.04727539964310884, 1.5349808027413907, 0.4444938546216624, 1.2086513858175814, 1.4562699420640843, 0.09348494559687809, 1.408526195781688
epoch: 8 cost: [0.251364943426809 9, 0.8017597069976115, 0.253322732497971, 0.5575948310537389, 0.011282657289004634, 1.4538522157264635, 0.9890753875360199, 0.1550892257235787
epoch: 9 cost: [0.13622790542829918, 0.07608327871020826, 0.6082856589307696, 0.21078277453212965, 1.5693652465483086, 0.7040179124089175, 0.04271294387663486, 0.265933390008535
```



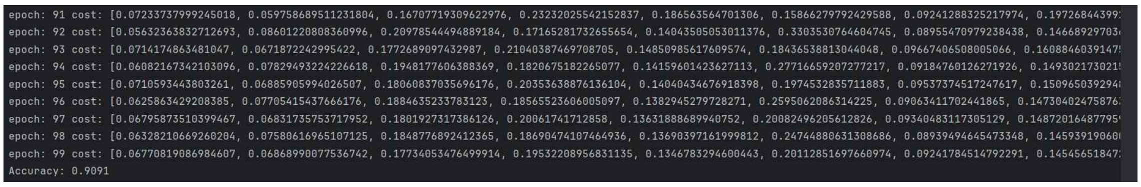
```
epoch: 89 cost: [0.19339317322872546, 0.23344355241638481, 0.22001212617691296, 1.5421955326846626, 1.430104442038545, 0.30012813509726827, 0.32297782203244807, 1.332634944499
epoch: 90 cost: [0.10062080683100331, 0.1430861857850914, 0.5705046469997872, 0.07748716105843591, 0.02242302024851852, 0.8403372775452119, 0.20183916563017368, 0.06061176261168
epoch: 91 cost: [0.16559328282507702, 0.11109516943932374, 0.1139067057852171, 1.1034696360898435, 0.6197921843639265, 0.11987235839274127, 0.1082663179835738, 0.216079132024620
epoch: 92 cost: [0.04272255001609958, 0.06840199645138922, 0.620528262 8992703, 0.05265243475487469, 0.3906352722674369, 1.0977800830959608, 0.05944172587680670 4, 0.5635973498963
epoch: 93 cost: [0.22758355070472142, 0.283003162471783 9, 0.184270414869900 9, 1.59112462972842, 0.07925138106466043, 0.21439894446228072, 0.2871390945652488, 0.1036930723725410
epoch: 94 cost: [0.03895205486006926, 0.036128301879000034, 0.3495557700437323 4, 0.10844571052003067, 0.7141843805332663, 0.6718327133711299, 0.04508188754194136, 0.564724322955
epoch: 95 cost: [0.13055200099940933, 0.2954960429225498, 0.2490674026152843 2, 0.5245268900203959, 0.04513065930248181 4, 0.23304183027699668, 0.23375404937325076, 0.092795846077
epoch: 96 cost: [0.06124875176071509, 0.05023472052048731 7, 0.15704524440392506, 0.1771508125891175, 1.0894356153227485, 0.264394335756927, 0.06743974702294855, 0.510496090548 6
epoch: 97 cost: [0.08166370601245342, 0.13368353035265335, 0.334949244622271 3, 0.22256017066082087, 0.04980550811177736 3, 0.374981130468887 8, 0.1525051464473027 6, 0.0889884683 9
epoch: 98 cost: [0.08810313953397295, 0.06259192716133131, 0.13662974800328503, 0.2261906001534489, 0.8683887883190388, 0.17375784057626706, 0.08725256956287848, 0.566934472750
epoch: 99 cost: [0.06957643695145702, 0.1399587864767089, 0.34301957488963614, 0.18480428908999885, 0.07199415128428523, 0.5208691685386831, 0.12503591179814705, 0.087242570260
Accuracy: 0.8215
```



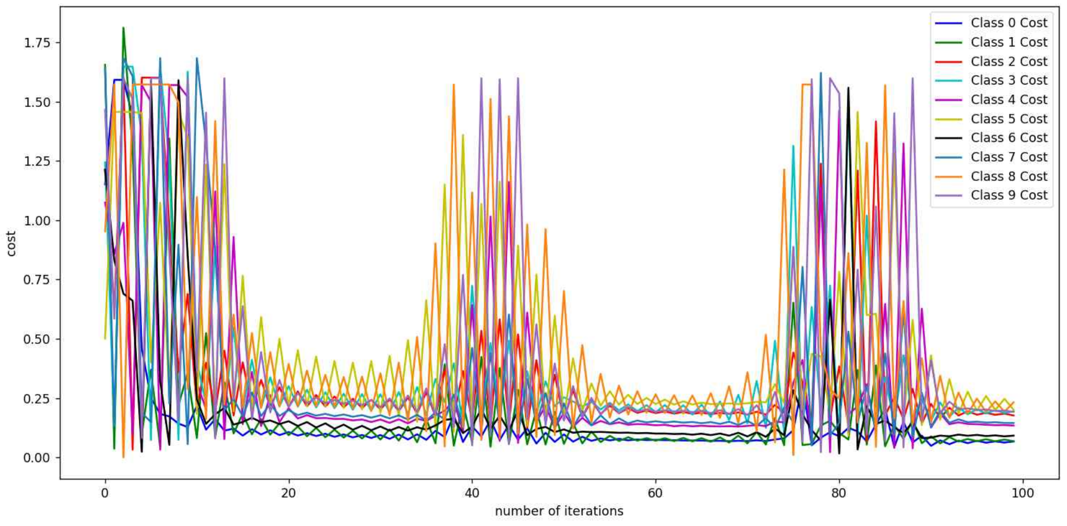
Normalize = True, Epoch = 100, lr = 0.01



```
epoch: 0 cost: [1.151582033675116, 1.6546645684862185, 1.2111648740318222, 1.2436025642055402, 1.0738231808267107, 0.5005945144981582, 1.2128504168846435, 1.638895265661285, 0.
epoch: 1 cost: [1.5911246756771023, 0.037137549467595996, 0.6213658719123755, 0.13267120194443385, 0.8582517912537208, 1.4562699420640843, 0.8316718031539144, 0.132108979496063
epoch: 2 cost: [1.5911246756771023, 1.8111366813126832, 1.6005268981401612, 1.6470007406004241, 0.9891113585636696, 1.4562699420640843, 0.690276890217379, 1.6829978208875649, 0
epoch: 3 cost: [1.4381778368485227, 1.3018401864629598, 0.033262432200269, 1.6470007406004241, 0.10426549918930253, 1.4562699420640843, 0.660822156386564, 1.6060594956338106, 1
epoch: 4 cost: [0.45857883072948813, 0.09996310594776614, 1.6005268981401612, 1.3537048894500112, 1.5693652465483086, 1.450916112750559, 0.024308189483749864, 0.187395858519049
epoch: 5 cost: [0.25058898292653164, 0.3702078798273009, 1.6005268981401612, 0.07384070268620045, 1.49871753667535, 0.400155927827893, 1.5897815010395224, 0.1494985433258936 6,
epoch: 6 cost: [0.1840937416127941, 0.0449124644241940 5, 1.6005268981401612, 1.6470007406004241, 0.03311835247979591, 1.0738602920042728, 0.32261027520462887, 1.68299782088756 4
epoch: 7 cost: [0.1732160958060836 3, 1.3443061926003101, 0.9717762495113323, 1.2482183683961938, 1.5693652465483086, 0.24885793665870917, 0.05320075305851626, 0.2279216399427378
epoch: 8 cost: [0.14315044256099657, 0.22029723341441, 0.3599114868376794, 0.07432946016801062, 1.5693652465483086, 1.4562699420640843, 1.5897815010395224, 0.8969894134224883,
epoch: 9 cost: [0.12833575725774812, 0.35757593609282706, 0.6894590780627352, 1.6261228279446671, 1.5186546075848752, 1.3497830486125533, 0.8495941901678822, 0.0565171192828602
```



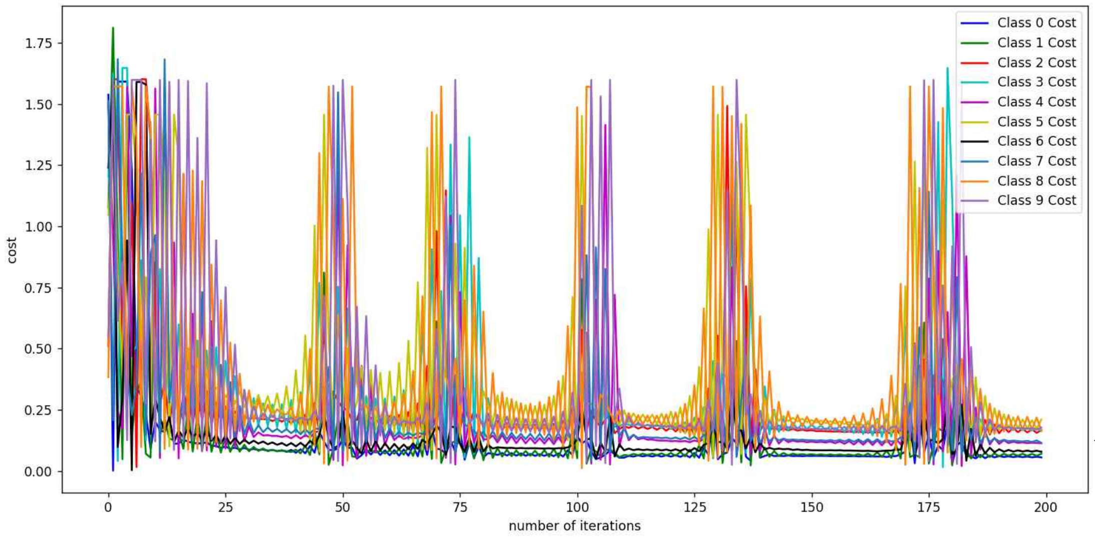
```
epoch: 91 cost: [0.07233737999245018, 0.059758689511231804, 0.16707719309622976, 0.23232025542152837, 0.186563564701306, 0.15866279792429588, 0.09241288325217974, 0.19726844399
epoch: 92 cost: [0.05632363832712693, 0.08601220808360996, 0.20978544494889184, 0.17165281732655654, 0.14043505053011376, 0.3303530764604745, 0.08955470979238438, 0.14668929703
epoch: 93 cost: [0.0714174863481047, 0.0671872242995422, 0.1772689097432987, 0.21040387469708705, 0.14850985617609574, 0.18436538813044048, 0.09667406508005066, 0.16088460391475
epoch: 94 cost: [0.06082167342103096, 0.07829493224226618, 0.1948177606388369, 0.18206751822650 77, 0.14159601423627113, 0.27716659207277217, 0.09184760126271926, 0.149302173021
epoch: 95 cost: [0.071059344380326 1, 0.06885905994026507, 0.18060837035696176, 0.20353638876136104, 0.14040434676918398, 0.1974532835711883, 0.09537374517247617, 0.150965039294
epoch: 96 cost: [0.0625863429208385, 0.07705415437666176, 0.1884635233783123, 0.18565523606005097, 0.1382945279728271, 0.2595062086314225, 0.09063411702441865, 0.14730402475876
epoch: 97 cost: [0.06795873510399467, 0.06831735753717952, 0.1801927317386126, 0.20061741712858, 0.13631888689940752, 0.20082496205612826, 0.09340483117305129, 0.14872016487795
epoch: 98 cost: [0.06328210669260204, 0.07580616965107125, 0.1848776892412365, 0.18690474107464936, 0.13690397161999812, 0.24744880631308686, 0.08939494645473348, 0.14593919060
epoch: 99 cost: [0.06770819086984607, 0.06868990077536742, 0.17734053476499914, 0.19532208956831135, 0.1346783294600443, 0.20112851697660974, 0.09241784514792291, 0.14545651847
Accuracy: 0.9091
```



Normalize = True, Epoch = 200, lr = 0.01

```
epoch: 0 cost: [1.5375094514677894, 1.0767221997512975, 0.5098032308199402, 1.2036064579524195, 1.4932287183726272, 1.0465316712638435, 1.2390539927636173, 1.5115932811267365,
epoch: 1 cost: [0.0016119284955404684, 1.8111366813126832, 1.6005268981401612, 1.624556437705937, 0.9601398786420231, 1.4562699420640843, 1.5897815010395224, 0.2074655932747347
epoch: 2 cost: [1.5911246756771023, 0.9484085941285463, 1.6005268981401612, 0.04081664929550362, 0.24179882563800184, 0.8092356759734096, 0.09799933759472293, 1.682997820887564
epoch: 3 cost: [1.5911246756771023, 0.04825641448776404, 0.49292250618731004, 1.6470007406004241, 0.17386921184645665, 0.39313400781094154, 0.298979323331921, 0.867932277508991
epoch: 4 cost: [1.5911246756771023, 1.4561255192262257, 0.4327575318980417, 1.6470007406004241, 1.5693652465483086, 1.4562699420640843, 0.941977311380381, 0.40476392004003314,
epoch: 5 cost: [1.2323650021492119, 0.15384452542859606, 1.0967913795846402, 0.48885064895231667, 0.4685402547107632, 1.4562699420640843, 0.003492773961632014, 0.22950864810498
epoch: 6 cost: [0.3478613067379666, 0.318017103042298, 0.016276371237769906, 0.5081067986812255, 1.2535663401825003, 1.2121385883602445, 1.5897815010395224, 0.3566064625396282,
epoch: 7 cost: [0.3024677602695796, 0.36898953801975953, 1.6005268981401612, 0.860782808674348, 0.19512751140435725, 0.5387993308148187, 1.5897815010395224, 1.2156298323914805,
epoch: 8 cost: [0.23550584126920393, 0.0701983670385809, 1.6005268981401612, 0.1255280730154123, 1.5693652465483086, 0.792392754730382, 1.5785862643664543, 0.1482745359847514,
epoch: 9 cost: [0.09090445211358271, 0.05498181557989653, 1.3550008465469576, 1.426819966960895, 0.1973205838143205, 0.22330365826204016, 0.15170824272297176, 0.88777190962583
```

```
epoch: 190 cost: [0.06076825325240031, 0.06029877013958229, 0.15440309513130546, 0.1876345040884436, 0.11064404541269918, 0.16843357540791196, 0.08874378495740298, 0.1209579201
epoch: 191 cost: [0.057560712760767074, 0.07047402818005308, 0.17856877546959474, 0.16900890489601172, 0.1256604577606666, 0.23187523008290026, 0.07894535879856615, 0.127691093
epoch: 192 cost: [0.05979459740662497, 0.06253854015881723, 0.1589428517644928, 0.1819746400576374, 0.11379830082489391, 0.17199035737025684, 0.08593929606790027, 0.12166957665
epoch: 193 cost: [0.05705452484755778, 0.0700899681599645, 0.17512681480381165, 0.16738515812476104, 0.11997910572497508, 0.21922329632276746, 0.07970945931175052, 0.1243757011
epoch: 194 cost: [0.05964355198240243, 0.06246755550780441, 0.15889164427390165, 0.17776098251602668, 0.11513720595105566, 0.17408586781432056, 0.08411814505966551, 0.122050723
epoch: 195 cost: [0.05683633040173366, 0.07045363255632128, 0.17297850283560326, 0.16660173027099995, 0.11682933558966165, 0.21134565186960114, 0.0799372765206134, 0.1206481862
epoch: 196 cost: [0.05929922151035014, 0.062139786487373755, 0.1574841981959973, 0.17598085461703014, 0.11361082513540148, 0.17476167871850484, 0.0833472295135696, 0.1229790043
epoch: 197 cost: [0.056085315189355014, 0.0701117928934997, 0.17051484936513156, 0.16540001732310095, 0.11601399332055845, 0.2075045513684416, 0.07943707560184363, 0.1186049257
epoch: 198 cost: [0.05826890036892765, 0.061909968739708374, 0.1557052047669025, 0.17596880513654253, 0.11445527607482289, 0.17250250961595506, 0.08299048972279263, 0.125082431
epoch: 199 cost: [0.05582210281268466, 0.07064219413723216, 0.16733930410526351, 0.16337898538950596, 0.113555154348333, 0.2103926241607482, 0.07887644779825862, 0.1150861800668
Accuracy: 0.9172
```



데

이터를 Normalize하면 모델 학습을 안정화할 수 있다
또한 learning_rate를 줄이고 epoch를 늘리면 accuracy가 증가하고 epoch가 크다보니 cost가 자주 튀는 것을 볼 수 있다.