Note: All Graphs Are in The Python Package File under Folder (Pictures)
Linear Classifications

Breast Data:

```
import LinearClassification
import DataPreprocessing

# Breast Classification
etavalues = [0.01, 0.015, 0.02, 0.03, 0.035]
for x in etavalues:
    test =
LinearClassification.CrossValidation(DataPreprocessing.breastTuningNP
, 5, 300, x, 3, 0.001, 0.05)
```

Output Mini Example for 1 Fold:

```
Fold 1
Testing set size : (28, 10)
Training set size : (112, 10)

0     ->    TrainL : 0.3196183     |     TestL : 0.0541085 |     TrainAcc :
0.8928571   |     TestAcc: 0.7500000
1     ->    TrainL : 0.2244270     |     TestL : 0.0134993 |     TrainAcc :
0.9017857   |     TestAcc: 0.7500000
2     ->    TrainL : 0.1705966     |     TestL : 0.0048895 |     TrainAcc :
0.9017857   |     TestAcc: 0.7500000
3     ->    TrainL : 0.1366442     |     TestL : 0.0023815 |     TrainAcc :
0.9017857   |     TestAcc: 0.7500000
4     ->    TrainL : 0.1178088     |     TestL : 0.0014698 |     TrainAcc :
0.9017857   |     TestAcc: 0.7500000
5     ->    TrainL : 0.1031771     |     TestL : 0.0010813 |     TrainAcc :
0.9107143   |     TestAcc: 0.7500000
6     ->    TrainL : 0.0923754     |     TestL : 0.0008974 |     TrainAcc :
0.9107143   |     TestAcc: 0.7500000
7     ->    TrainL : 0.0838621     |     TestL : 0.0008055 |     TrainAcc :
0.9107143   |     TestAcc: 0.7500000
8     ->    TrainL : 0.0773272     |     TestL : 0.0007597 |     TrainAcc :
0.9107143   |     TestAcc: 0.7500000
9     ->    TrainL : 0.0721648     |     TestL : 0.0007390 |     TrainAcc :
```

```
0.9107143   |     TestAcc: 0.7500000
10    ->    TrainL : 0.0679964      |     TestL : 0.0007327 |     TrainAcc :
0.9196429   |     TestAcc: 0.7500000
11    ->    TrainL : 0.0645718      |     TestL : 0.0007351 |     TrainAcc :
0.9196429   |     TestAcc: 0.7500000
12    ->    TrainL : 0.0616755      |     TestL : 0.0007429 |     TrainAcc :
0.9196429   |     TestAcc: 0.7500000
13    ->    TrainL : 0.0592104      |     TestL : 0.0007542 |     TrainAcc :
0.9196429   |     TestAcc: 0.7500000
14    ->    TrainL : 0.0571140      |     TestL : 0.0007676 |     TrainAcc :
0.9196429   |     TestAcc: 0.7500000
15    ->    TrainL : 0.0553150      |     TestL : 0.0007824 |     TrainAcc :
0.9285714   |     TestAcc: 0.7500000
16    ->    TrainL : 0.0537589      |     TestL : 0.0007980 |     TrainAcc :
0.9285714   |     TestAcc: 0.7500000
17    ->    TrainL : 0.0524035      |     TestL : 0.0008141 |     TrainAcc :
0.9285714   |     TestAcc: 0.7500000
18    ->    TrainL : 0.0512155      |     TestL : 0.0008304 |     TrainAcc :
0.9285714   |     TestAcc: 0.7500000
19    ->    TrainL : 0.0501686      |     TestL : 0.0008468 |     TrainAcc :
0.9285714   |     TestAcc: 0.7500000
20    ->    TrainL : 0.0492412      |     TestL : 0.0008632 |     TrainAcc :
0.9285714   |     TestAcc: 0.7500000
21    ->    TrainL : 0.0484160      |     TestL : 0.0008794 |     TrainAcc :
0.9285714   |     TestAcc: 0.7500000
22    ->    TrainL : 0.0476786      |     TestL : 0.0008953 |     TrainAcc :
0.9285714   |     TestAcc: 0.7500000
23    ->    TrainL : 0.0470173      |     TestL : 0.0009111 |     TrainAcc :
0.9285714   |     TestAcc: 0.7142857
24    ->    TrainL : 0.0464180      |     TestL : 0.0009265 |     TrainAcc :
0.9285714   |     TestAcc: 0.7142857
25    ->    TrainL : 0.0458598      |     TestL : 0.0009417 |     TrainAcc :
0.9285714   |     TestAcc: 0.7142857
```

House Classification Linear Classifier

```
# House Classification
for eta in etavalues:
   votingTuning =
LinearClassification.CrossValidation(dp.houseTuningNP, 5, 300, eta,
3, 0.001, 0.05)
```

Output Mini Example for 1 Fold:

```
Fold 1
Testing set size : (18, 17)
Training set size : (69, 17)

0     ->     TrainL : 0.5765269      |     TestL : 0.4630579 |     TrainAcc :
0.8985507    |     TestAcc: 0.8888889
1     ->     TrainL : 0.4262632      |     TestL : 0.3293719 |     TrainAcc :
0.8985507    |     TestAcc: 0.8888889
2     ->     TrainL : 0.3336425      |     TestL : 0.2472146 |     TrainAcc :
0.8840580    |     TestAcc: 0.8888889
3     ->     TrainL : 0.2728103      |     TestL : 0.1934049 |     TrainAcc :
0.8840580    |     TestAcc: 0.8888889
4     ->     TrainL : 0.2304152      |     TestL : 0.1562511 |     TrainAcc :
0.8840580    |     TestAcc: 0.8888889
5     ->     TrainL : 0.1994731      |     TestL : 0.1294834 |     TrainAcc :
0.8840580    |     TestAcc: 0.8888889
6     ->     TrainL : 0.1760617      |     TestL : 0.1095250 |     TrainAcc :
0.8985507    |     TestAcc: 0.8888889
7     ->     TrainL : 0.1578349      |     TestL : 0.0942201 |     TrainAcc :
0.8985507    |     TestAcc: 0.8888889
8     ->     TrainL : 0.1433131      |     TestL : 0.0822067 |     TrainAcc :
0.8985507    |     TestAcc: 0.8888889
9     ->     TrainL : 0.1314738      |     TestL : 0.0725899 |     TrainAcc :
0.8985507    |     TestAcc: 0.8888889
10    ->     TrainL : 0.1211228      |     TestL : 0.0647613 |     TrainAcc :
0.8985507    |     TestAcc: 0.8888889
11    ->     TrainL : 0.1124197      |     TestL : 0.0582955 |     TrainAcc :
0.8985507    |     TestAcc: 0.8888889
12    ->     TrainL : 0.1050140      |     TestL : 0.0528873 |     TrainAcc :
0.9130435    |     TestAcc: 0.8888889
```

Voting LC Testing

```
## eta = 0.03 or 0.035 gives the highest accuracy and precision on the testing

votingTesting = LinearClassification.CrossValidation(dp.houseNP80, 5, 300, 0.03, 3, 0.001, 0.05)
```

Sample output:

```
Fold 1
Testing set size : (70, 17)
Training set size : (278, 17)

0     ->    TrainL : 0.2512830     |     TestL : 0.0438223 |     TrainAcc : 0.9028777   |     TestAcc: 0.9000000
1     ->    TrainL : 0.0801011     |     TestL : 0.0186876 |     TrainAcc : 0.9100719   |     TestAcc: 0.9000000
2     ->    TrainL : 0.0527060     |     TestL : 0.0117252 |     TrainAcc : 0.9244604   |     TestAcc: 0.9142857
3     ->    TrainL : 0.0409860     |     TestL : 0.0087789 |     TrainAcc : 0.9316547   |     TestAcc: 0.9428571
4     ->    TrainL : 0.0344158     |     TestL : 0.0070857 |     TrainAcc : 0.9424460   |     TestAcc: 0.9714286
5     ->    TrainL : 0.0302392     |     TestL : 0.0061464 |     TrainAcc : 0.9460432   |     TestAcc: 0.9714286
6     ->    TrainL : 0.0273005     |     TestL : 0.0056057 |     TrainAcc : 0.9532374   |     TestAcc: 0.9714286
7     ->    TrainL : 0.0250695     |     TestL : 0.0052978 |     TrainAcc : 0.9532374   |     TestAcc: 0.9714286
8     ->    TrainL : 0.0233598     |     TestL : 0.0051356 |     TrainAcc : 0.9568345   |     TestAcc: 0.9571429
9     ->    TrainL : 0.0219946     |     TestL : 0.0050693 |     TrainAcc : 0.9604317   |     TestAcc: 0.9571429
10    ->    TrainL : 0.0209031     |     TestL : 0.0050682 |     TrainAcc : 0.9604317   |     TestAcc: 0.9571429
```

Car Linear Classification

```
# Car Classification
for eta in etavalues:
    carTuning = LinearClassification.CrossValidation(dp.carTuningNP,
5, 300, eta, 3, 0.001, 0.05)
```

Tuning Output:

```
Fold 1
Testing set size : (70, 7)
Training set size : (276, 7)

0     ->    TrainL : 0.7049381      |     TestL : 0.2880506 |     TrainAcc :
0.6847826   |     TestAcc: 0.7142857
1     ->    TrainL : 0.4115942      |     TestL : 0.1927253 |     TrainAcc :
0.6775362   |     TestAcc: 0.7285714
2     ->    TrainL : 0.3626638      |     TestL : 0.1330814 |     TrainAcc :
0.6630435   |     TestAcc: 0.7285714
3     ->    TrainL : 0.3299704      |     TestL : 0.1000083 |     TrainAcc :
0.6666667   |     TestAcc: 0.7428571
4     ->    TrainL : 0.3055379      |     TestL : 0.0798352 |     TrainAcc :
0.6630435   |     TestAcc: 0.7285714
5     ->    TrainL : 0.2878898      |     TestL : 0.0666733 |     TrainAcc :
0.6557971   |     TestAcc: 0.7428571
6     ->    TrainL : 0.2741907      |     TestL : 0.0576117 |     TrainAcc :
0.6630435   |     TestAcc: 0.7428571
7     ->    TrainL : 0.2633573      |     TestL : 0.0510953 |     TrainAcc :
0.6702899   |     TestAcc: 0.7428571
8     ->    TrainL : 0.2547681      |     TestL : 0.0462396 |     TrainAcc :
0.6775362   |     TestAcc: 0.7428571
9     ->    TrainL : 0.2476736      |     TestL : 0.0425142 |     TrainAcc :
0.6811594   |     TestAcc: 0.7428571
10    ->    TrainL : 0.2419694      |     TestL : 0.0395862 |     TrainAcc :
0.6811594   |     TestAcc: 0.7428571
```

Linear Regression:

Abalone

```
import DataPreprocessing as dp
import LinearRegression as LR

etavalues = [0.01, 0.015, 0.02, 0.03, 0.035]

# Abalone
for eta in etavalues:
   abaloneTuning = LR.CrossValidation(dp.abaloneTuningNP, 5, 500,
eta, 5, 0.001, 0.05)
```

Tuning Sample Output:

```
Fold 5
Testing set size : (167, 9)
Training set size : (668, 9)

0     ->    TrainL : -0.3077025     |     TestL : -0.3358578      |
TrainMSE : 0.0393765    |     TestMSE: 0.0461868
1     ->    TrainL : -0.3497624     |     TestL : -0.3362035      |
TrainMSE : 0.0383154    |     TestMSE: 0.0450168
2     ->    TrainL : -0.3500821     |     TestL : -0.3359588      |
TrainMSE : 0.0373277    |     TestMSE: 0.0439760
3     ->    TrainL : -0.3502196     |     TestL : -0.3356640      |
TrainMSE : 0.0364366    |     TestMSE: 0.0430473
4     ->    TrainL : -0.3503249     |     TestL : -0.3353694      |
TrainMSE : 0.0356330    |     TestMSE: 0.0422158
5     ->    TrainL : -0.3504193     |     TestL : -0.3350828      |
TrainMSE : 0.0349081    |     TestMSE: 0.0414710
6     ->    TrainL : -0.3505067     |     TestL : -0.3348055      |
TrainMSE : 0.0342542    |     TestMSE: 0.0408037
7     ->    TrainL : -0.3505882     |     TestL : -0.3345377      |
TrainMSE : 0.0336641    |     TestMSE: 0.0402059
8     ->    TrainL : -0.3506644     |     TestL : -0.3342792      |
TrainMSE : 0.0331315    |     TestMSE: 0.0396706
9     ->    TrainL : -0.3507357     |     TestL : -0.3340301      |
TrainMSE : 0.0326507    |     TestMSE: 0.0391913
10    ->    TrainL : -0.3508025     |     TestL : -0.3337900      |
TrainMSE : 0.0322167    |     TestMSE: 0.0387623
```

Testing Sample Output:

Forest Tuning

```
# Forest Fire

forestetavalues = [0.01, 0.0125, 0.015, 0.0175, 0.02]
for eta in forestetavalues:
   forestTuning = LR.CrossValidation(dp.forestTuningNP,5,500, eta, 5, 
0.001, 0.05)
```

Mini Output Sample

```
Fold 5
Testing set size : (20, 13)
Training set size : (83, 13)

0     ->    TrainL : -0.2492854     |     TestL : -0.4362389      |
TrainMSE : 0.2055510    |     TestMSE: 0.2316211
1     ->    TrainL : -0.5158734     |     TestL : -0.5835453      |
TrainMSE : 0.1409924    |     TestMSE: 0.1779354
2     ->    TrainL : -0.6151974     |     TestL : -0.6481687      |
TrainMSE : 0.1355167    |     TestMSE: 0.1798555
3     ->    TrainL : -0.6617123     |     TestL : -0.6817206      |
TrainMSE : 0.1394773    |     TestMSE: 0.1890732
4     ->    TrainL : -0.6865798     |     TestL : -0.7009985      |
TrainMSE : 0.1434967    |     TestMSE: 0.1969906
5     ->    TrainL : -0.7009501     |     TestL : -0.7128771      |
TrainMSE : 0.1461396    |     TestMSE: 0.2026389
6     ->    TrainL : -0.7097009     |     TestL : -0.7206247      |
TrainMSE : 0.1476241    |     TestMSE: 0.2065269
7     ->    TrainL : -0.7152592     |     TestL : -0.7259526      |
TrainMSE : 0.1483897    |     TestMSE: 0.2092804
8     ->    TrainL : -0.7189348     |     TestL : -0.7298133      |
TrainMSE : 0.1487853    |     TestMSE: 0.2113700
9     ->    TrainL : -0.7214713     |     TestL : -0.7327580      |
TrainMSE : 0.1490360    |     TestMSE: 0.2131004
10    ->    TrainL : -0.7233043     |     TestL : -0.7351138      |
TrainMSE : 0.1492706    |     TestMSE: 0.2146538
```

Testing:

```
forestTesting = LR.CrossValidation(dp.forestNP80, 5, 500, 0.2, 5,
0.001, 0.05)
```

Mini Output Sample

```
Fold 5
Testing set size : (82, 13)
Training set size : (332, 13)

0     ->    TrainL : -0.6578927     |      TestL : -0.7514058      |
TrainMSE : 0.2402547    |      TestMSE: 0.2857232
1     ->    TrainL : -0.6855510     |      TestL : -0.7479705      |
TrainMSE : 0.2389314    |      TestMSE: 0.2838773
2     ->    TrainL : -0.6854317     |      TestL : -0.7456886      |
TrainMSE : 0.2378968    |      TestMSE: 0.2821988
3     ->    TrainL : -0.6854406     |      TestL : -0.7440510      |
TrainMSE : 0.2372088    |      TestMSE: 0.2810381
4     ->    TrainL : -0.6854715     |      TestL : -0.7428451      |
TrainMSE : 0.2368042    |      TestMSE: 0.2802887
5     ->    TrainL : -0.6855015     |      TestL : -0.7419442      |
TrainMSE : 0.2366086    |      TestMSE: 0.2798401
6     ->    TrainL : -0.6855253     |      TestL : -0.7412633      |
TrainMSE : 0.2365534    |      TestMSE: 0.2795992
7     ->    TrainL : -0.6855425     |      TestL : -0.7407429      |
TrainMSE : 0.2365856    |      TestMSE: 0.2794951
8     ->    TrainL : -0.6855541     |      TestL : -0.7403409      |
TrainMSE : 0.2366678    |      TestMSE: 0.2794774
9     ->    TrainL : -0.6855611     |      TestL : -0.7400273      |
TrainMSE : 0.2367749    |      TestMSE: 0.2795116
10    ->    TrainL : -0.6855646     |      TestL : -0.7397802      |
TrainMSE : 0.2368911    |      TestMSE: 0.2795746
```

Computer Linear Regression Tuning

```
computeretavalues = [0.01, 0.0125, 0.015, 0.0175, 0.02]
for eta in computeretavalues:
   computerTuning = LR.CrossValidation(dp.computerTuningNP,5, 100,
eta, 3, 0.001, 0.05)
```

Sample Output :

```
Fold 5
Testing set size : (8, 9)
Training set size : (34, 9)

0     ->    TrainL : -0.2454299     |     TestL : -0.4956086      |
TrainMSE : 0.1128266    |     TestMSE: 0.1198185
1     ->    TrainL : -0.5326167     |     TestL : -0.6681041      |
TrainMSE : 0.0148567    |     TestMSE: 0.0094541
2     ->    TrainL : -0.6458731     |     TestL : -0.7459052      |
TrainMSE : 0.0140654    |     TestMSE: 0.0106190
3     ->    TrainL : -0.7026092     |     TestL : -0.7883109      |
TrainMSE : 0.0369147    |     TestMSE: 0.0387470
4     ->    TrainL : -0.7356251     |     TestL : -0.8142845      |
TrainMSE : 0.0641161    |     TestMSE: 0.0716492
5     ->    TrainL : -0.7566471     |     TestL : -0.8314505      |
TrainMSE : 0.0898518    |     TestMSE: 0.1026481
6     ->    TrainL : -0.7708216     |     TestL : -0.8434084      |
TrainMSE : 0.1124671    |     TestMSE: 0.1298618
7     ->    TrainL : -0.7807493     |     TestL : -0.8520635      |
TrainMSE : 0.1317381    |     TestMSE: 0.1530554
8     ->    TrainL : -0.7878834     |     TestL : -0.8585130      |
TrainMSE : 0.1479370    |     TestMSE: 0.1725627
9     ->    TrainL : -0.7930988     |     TestL : -0.8634310      |
TrainMSE : 0.1614820    |     TestMSE: 0.1888838
10    ->    TrainL : -0.7969532     |     TestL : -0.8672532      |
TrainMSE : 0.1728029    |     TestMSE: 0.2025291
```

Testing:

```
computerTesting = LR.CrossValidation(dp.computerNP80,5, 100, 0.0125,
3, 0.001, 0.05)
```

Sample Output:

```
Fold 1
Testing set size : (34, 9)
```

```
Training set size : (133, 9)

0     ->    TrainL : -0.4367102     |     TestL : -0.7011504      |
TrainMSE : 0.0258217    |     TestMSE: 0.0086524
1     ->    TrainL : -0.7110948     |     TestL : -0.8061879      |
TrainMSE : 0.0850272    |     TestMSE: 0.0698787
2     ->    TrainL : -0.7695829     |     TestL : -0.8426840      |
TrainMSE : 0.1539805    |     TestMSE: 0.1456636
3     ->    TrainL : -0.7919355     |     TestL : -0.8598753      |
TrainMSE : 0.2031903    |     TestMSE: 0.2010823
4     ->    TrainL : -0.8021696     |     TestL : -0.8692291      |
TrainMSE : 0.2365332    |     TestMSE: 0.2395182
5     ->    TrainL : -0.8070828     |     TestL : -0.8747693      |
TrainMSE : 0.2592856    |     TestMSE: 0.2664025
6     ->    TrainL : -0.8093348     |     TestL : -0.8782524      |
TrainMSE : 0.2752641    |     TestMSE: 0.2857674
7     ->    TrainL : -0.8101659     |     TestL : -0.8805525      |
TrainMSE : 0.2869633    |     TestMSE: 0.3002863
8     ->    TrainL : -0.8102179     |     TestL : -0.8821422      |
TrainMSE : 0.2959646    |     TestMSE: 0.3116738
9     ->    TrainL : -0.8098442     |     TestL : -0.8832912      |
TrainMSE : 0.3032577    |     TestMSE: 0.3210165
10    ->    TrainL : -0.8092462     |     TestL : -0.8841590      |
TrainMSE : 0.3094579    |     TestMSE: 0.3289995
```

Abalone FFBP Neural Network Tuning:

```
# FeedForward Neural Network w/ hidden layers
# 8 Inputs -> 1 Ouput
# Neural Network Architecture -> [8,9,8,1]
# Tuning:
for eta in etavalues:
   abaloneTuning = NNReg.CrossValidation(dp.abaloneTuningNP,5,
[8,9,8,1], 500, eta, 0.001, 0.05)
```

Output Sample:

```
Fold 3
Testing set size : (167, 9)
Training set size : (668, 9)

0     ->     TrainL : 0.1159370     |     TestL : 0.0855237 |     TrainMSE :
1.2557431    |     TestMSE: 1.2377189
1     ->     TrainL : 0.1158462     |     TestL : 0.0884556 |     TrainMSE :
1.2881313    |     TestMSE: 1.2750692
2     ->     TrainL : 0.1157323     |     TestL : 0.0912513 |     TrainMSE :
1.3208910    |     TestMSE: 1.3129593
3     ->     TrainL : 0.1155991     |     TestL : 0.0938796 |     TrainMSE :
1.3534263    |     TestMSE: 1.3507423
4     ->     TrainL : 0.1154467     |     TestL : 0.0963191 |     TrainMSE :
1.3851848    |     TestMSE: 1.3878202
5     ->     TrainL : 0.1152749     |     TestL : 0.0985532 |     TrainMSE :
1.4156289    |     TestMSE: 1.4236012
6     ->     TrainL : 0.1150835     |     TestL : 0.1005706 |     TrainMSE :
1.4442671    |     TestMSE: 1.4575279
7     ->     TrainL : 0.1148721     |     TestL : 0.1023653 |     TrainMSE :
1.4706820    |     TestMSE: 1.4891099
8     ->     TrainL : 0.1146405     |     TestL : 0.1039369 |     TrainMSE :
1.4945511    |     TestMSE: 1.5179496
9     ->     TrainL : 0.1143884     |     TestL : 0.1052900 |     TrainMSE :
1.5156562    |     TestMSE: 1.5437596
10    ->     TrainL : 0.1141154     |     TestL : 0.1064333 |     TrainMSE :
1.5338826    |     TestMSE: 1.5663689
```

Testing:

```
# Testing:
```

```
forestTesting = LR.CrossValidation(dp.forestNP80, 5, 500, 0.2, 5,
0.001, 0.05)
```

Output Sample:

```
Fold 5
Testing set size : (668, 9)
Training set size : (2674, 9)

0     ->    TrainL : 0.1218784     |     TestL : 0.0720568 |     TrainMSE :
1.0656505   |     TestMSE: 1.0433330
1     ->    TrainL : 0.1206241     |     TestL : 0.0864299 |     TrainMSE :
1.1976453   |     TestMSE: 1.1889644
2     ->    TrainL : 0.1185985     |     TestL : 0.0954057 |     TrainMSE :
1.3108492   |     TestMSE: 1.3123753
3     ->    TrainL : 0.1156202     |     TestL : 0.0991638 |     TrainMSE :
1.3686796   |     TestMSE: 1.3772591
4     ->    TrainL : 0.1115809     |     TestL : 0.0993578 |     TrainMSE :
1.3721202   |     TestMSE: 1.3855161
5     ->    TrainL : 0.1065562     |     TestL : 0.0973674 |     TrainMSE :
1.3377178   |     TestMSE: 1.3544599
6     ->    TrainL : 0.1007936     |     TestL : 0.0939728 |     TrainMSE :
1.2791186   |     TestMSE: 1.2980869
7     ->    TrainL : 0.0946239     |     TestL : 0.0895873 |     TrainMSE :
1.2048569   |     TestMSE: 1.2250348
8     ->    TrainL : 0.0883649     |     TestL : 0.0844496 |     TrainMSE :
1.1203128   |     TestMSE: 1.1407313
9     ->    TrainL : 0.0822641     |     TestL : 0.0787222 |     TrainMSE :
1.0293232   |     TestMSE: 1.0490984
10    ->    TrainL : 0.0764841     |     TestL : 0.0725396 |     TrainMSE :
0.9350718   |     TestMSE: 0.9534630
```

Forest FFBP Network:
Tuning:

```
# FeedForward Neural Network w/ hidden layers
# 12 Inputs -> 1 Output
# Neural Network Architecture -> [12,13,12,1]
# Tuning:
for eta in forestetavalues:
   forestNNTuning = NNReg.CrossValidation(dp.forestTuningNP, 5,
[12,13,12,1], 500, eta, 0.001, 0.05)
```

Output Sample:

```
Fold 5
Testing set size : (20, 13)
Training set size : (83, 13)

0     ->    TrainL : 0.0802097      |     TestL : 0.0636194 |     TrainMSE :
2.1710075   |     TestMSE: 2.2415515
1     ->    TrainL : 0.0798755      |     TestL : 0.0659144 |     TrainMSE :
2.2281500   |     TestMSE: 2.2797088
2     ->    TrainL : 0.0794757      |     TestL : 0.0678249 |     TrainMSE :
2.2803563   |     TestMSE: 2.3161754
3     ->    TrainL : 0.0790195      |     TestL : 0.0692941 |     TrainMSE :
2.3245192   |     TestMSE: 2.3476338
4     ->    TrainL : 0.0785029      |     TestL : 0.0703300 |     TrainMSE :
2.3587466   |     TestMSE: 2.3716222
5     ->    TrainL : 0.0779218      |     TestL : 0.0709835 |     TrainMSE :
2.3823824   |     TestMSE: 2.3871990
6     ->    TrainL : 0.0772729      |     TestL : 0.0713213 |     TrainMSE :
2.3958929   |     TestMSE: 2.3947055
7     ->    TrainL : 0.0765545      |     TestL : 0.0714080 |     TrainMSE :
2.4004567   |     TestMSE: 2.3951429
8     ->    TrainL : 0.0757662      |     TestL : 0.0712970 |     TrainMSE :
2.3974945   |     TestMSE: 2.3896724
9     ->    TrainL : 0.0749096      |     TestL : 0.0710296 |     TrainMSE :
2.3883593   |     TestMSE: 2.3793626
10    ->    TrainL : 0.0739878      |     TestL : 0.0706372 |     TrainMSE :
2.3742120   |     TestMSE: 2.3651104
```

Testing: Eta = 0.0175

```
forestNNTest = NNReg.CrossValidation(dp.forestNP80, 5, [12,13,12,1],
500, 0.017, 0.001, 0.05)
```

Output Sample:

```
486   ->    TrainL : -0.0144043     |     TestL : -0.0030254      |
TrainMSE : 0.6028241    |     TestMSE: 0.6724523
487   ->    TrainL : -0.0143680     |     TestL : -0.0029984      |
TrainMSE : 0.6031358    |     TestMSE: 0.6727922
488   ->    TrainL : -0.0143320     |     TestL : -0.0029715      |
TrainMSE : 0.6034462    |     TestMSE: 0.6731305
489   ->    TrainL : -0.0142962     |     TestL : -0.0029448      |
TrainMSE : 0.6037551    |     TestMSE: 0.6734672
490   ->    TrainL : -0.0142606     |     TestL : -0.0029182      |
TrainMSE : 0.6040626    |     TestMSE: 0.6738024
491   ->    TrainL : -0.0142253     |     TestL : -0.0028918      |
TrainMSE : 0.6043687    |     TestMSE: 0.6741360
492   ->    TrainL : -0.0141901     |     TestL : -0.0028655      |
TrainMSE : 0.6046734    |     TestMSE: 0.6744681
493   ->    TrainL : -0.0141551     |     TestL : -0.0028394      |
TrainMSE : 0.6049766    |     TestMSE: 0.6747986
494   ->    TrainL : -0.0141204     |     TestL : -0.0028135      |
TrainMSE : 0.6052785    |     TestMSE: 0.6751275
495   ->    TrainL : -0.0140859     |     TestL : -0.0027877      |
TrainMSE : 0.6055790    |     TestMSE: 0.6754550
496   ->    TrainL : -0.0140515     |     TestL : -0.0027620      |
TrainMSE : 0.6058781    |     TestMSE: 0.6757809
497   ->    TrainL : -0.0140174     |     TestL : -0.0027365      |
TrainMSE : 0.6061759    |     TestMSE: 0.6761054
498   ->    TrainL : -0.0139835     |     TestL : -0.0027111      |
TrainMSE : 0.6064722    |     TestMSE: 0.6764283
499   ->    TrainL : -0.0139497     |     TestL : -0.0026859      |
TrainMSE : 0.6067673    |     TestMSE: 0.6767497
```

Computer Hardware Deep Neural Network

Tuning

```
# FeedForward Neural Network w/ hidden layers
# 8 Inputs -> 1 Ouput
# Neural Network Architecture -> [8,9,8,1]
# Tuning:
for eta in computeretavalues:
   computerNNTuning = FFBBReg.CrossValidation(dp.computerTuningNP,5,
[8,9,8,1], 100, eta, 0.001, 0.05)
```

Sample Output:

```
90    ->    TrainL : -0.1201450     |     TestL : -0.1184019      |
TrainMSE : 0.0393752    |     TestMSE: 0.0582644
91    ->    TrainL : -0.1201543     |     TestL : -0.1184367      |
TrainMSE : 0.0394660    |     TestMSE: 0.0584584
92    ->    TrainL : -0.1201634     |     TestL : -0.1184705      |
TrainMSE : 0.0395545    |     TestMSE: 0.0586471
93    ->    TrainL : -0.1201723     |     TestL : -0.1185034      |
TrainMSE : 0.0396407    |     TestMSE: 0.0588307
94    ->    TrainL : -0.1201810     |     TestL : -0.1185354      |
TrainMSE : 0.0397248    |     TestMSE: 0.0590092
95    ->    TrainL : -0.1201895     |     TestL : -0.1185665      |
TrainMSE : 0.0398067    |     TestMSE: 0.0591830
96    ->    TrainL : -0.1201978     |     TestL : -0.1185968      |
TrainMSE : 0.0398865    |     TestMSE: 0.0593521
97    ->    TrainL : -0.1202059     |     TestL : -0.1186262      |
TrainMSE : 0.0399644    |     TestMSE: 0.0595167
98    ->    TrainL : -0.1202138     |     TestL : -0.1186549      |
TrainMSE : 0.0400404    |     TestMSE: 0.0596771
99    ->    TrainL : -0.1202215     |     TestL : -0.1186828      |
TrainMSE : 0.0401145    |     TestMSE: 0.0598333

Process finished with exit code 0
```

Testing Eta = 0.015

```
# Testing:
```

```
computerTesting = FFBBReg.CrossValidation(dp.computerNP80, 5,
[8,9,8,1], 100, 0.015, 0.001, 0.05)
```

Sample output:

```
87    ->    TrainL : -0.0881616     |     TestL : -0.0751349      |
TrainMSE : 0.0856140    |     TestMSE: 0.0945456
88    ->    TrainL : -0.0883832     |     TestL : -0.0754257      |
TrainMSE : 0.0848235    |     TestMSE: 0.0935010
89    ->    TrainL : -0.0886018     |     TestL : -0.0757115      |
TrainMSE : 0.0840524    |     TestMSE: 0.0924825
90    ->    TrainL : -0.0888173     |     TestL : -0.0759926      |
TrainMSE : 0.0832998    |     TestMSE: 0.0914893
91    ->    TrainL : -0.0890298     |     TestL : -0.0762690      |
TrainMSE : 0.0825652    |     TestMSE: 0.0905204
92    ->    TrainL : -0.0892394     |     TestL : -0.0765408      |
TrainMSE : 0.0818479    |     TestMSE: 0.0895752
93    ->    TrainL : -0.0894462     |     TestL : -0.0768083      |
TrainMSE : 0.0811474    |     TestMSE: 0.0886528
94    ->    TrainL : -0.0896502     |     TestL : -0.0770716      |
TrainMSE : 0.0804631    |     TestMSE: 0.0877526
95    ->    TrainL : -0.0898515     |     TestL : -0.0773307      |
TrainMSE : 0.0797944    |     TestMSE: 0.0868738
96    ->    TrainL : -0.0900503     |     TestL : -0.0775857      |
TrainMSE : 0.0791410    |     TestMSE: 0.0860159
97    ->    TrainL : -0.0902465     |     TestL : -0.0778369      |
TrainMSE : 0.0785023    |     TestMSE: 0.0851782
98    ->    TrainL : -0.0904401     |     TestL : -0.0780842      |
TrainMSE : 0.0778779    |     TestMSE: 0.0843602
99    ->    TrainL : -0.0906314     |     TestL : -0.0783277      |
TrainMSE : 0.0772673    |     TestMSE: 0.0835612

Process finished with exit code 0
```

Classification Deep Neural Networks:

Breast:
Tuning:

```
# Deep Neural Network
etavalues = [0.01, 0.015, 0.02, 0.03, 0.035]

# Neural Network Architecture 9 Input -> 9 Hidden 1 Inputs -> 9 Hidden 2
Inputs - > 1 Output
for eta in etavalues:
   breastTuning = DC.CrossValidation(dp.breastTuningNP,5,[9,9,9,1],500,
eta, 10, 0.001, 0.5,'sigmoid', 'momentum')
```

Tuning Sample output:

```
Fold 5
Testing set size : (28, 10)
Training set size : (112, 10)

1     ->    TrainLoss : 1.8533925   |     TestLoss : 1.2676903    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
2     ->    TrainLoss : 1.8007152   |     TestLoss : 1.2142989    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
3     ->    TrainLoss : 1.7902678   |     TestLoss : 1.2064270    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
4     ->    TrainLoss : 1.7870111   |     TestLoss : 1.2011644    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
5     ->    TrainLoss : 1.7848280   |     TestLoss : 1.1980578    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
6     ->    TrainLoss : 1.7833330   |     TestLoss : 1.1966375    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
7     ->    TrainLoss : 1.7822473   |     TestLoss : 1.1963623    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
8     ->    TrainLoss : 1.7814132   |     TestLoss : 1.1965948    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
9     ->    TrainLoss : 1.7807541   |     TestLoss : 1.1968895    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
10    ->    TrainLoss : 1.7802254   |     TestLoss : 1.1970891    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
```

Breast DNN Testing:

```
# DNN Testing
breastDNNtest =  DC.CrossValidation(dp.breastNP80, 5, [9,9,9,1], 500,
0.015, 10, 0.001, 0.05, 'sigmoid', 'momentum')
```

Sample output:

```
490   ->    TrainLoss : 2.2516799   |     TestLoss : 1.5315540    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
491   ->    TrainLoss : 2.2516815   |     TestLoss : 1.5315560    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
492   ->    TrainLoss : 2.2516830   |     TestLoss : 1.5315580    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
493   ->    TrainLoss : 2.2516846   |     TestLoss : 1.5315600    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
494   ->    TrainLoss : 2.2516861   |     TestLoss : 1.5315620    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
495   ->    TrainLoss : 2.2516877   |     TestLoss : 1.5315640    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
496   ->    TrainLoss : 2.2516892   |     TestLoss : 1.5315660    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
497   ->    TrainLoss : 2.2516907   |     TestLoss : 1.5315680    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
498   ->    TrainLoss : 2.2516923   |     TestLoss : 1.5315700    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
499   ->    TrainLoss : 2.2516938   |     TestLoss : 1.5315719    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
500   ->    TrainLoss : 2.2516953   |     TestLoss : 1.5315739    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000

Process finished with exit code 0
```

VOTING DNN CLASSIFICATION

TUNING

```
# Deep Neural Network
# Neural Network Architecture 16 Input -> 16 Hidden 1 Inputs -> 16
Hidden 2 Inputs - > 1 Output
for eta in etavalues:
   votingTuning =
DC.CrossValidation(dp.houseTuningNP,5,[16,16,16,1],500, eta, 10,
0.001, 0.5,'sigmoid', 'momentum')
```

```
Sample output:


TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
64    ->     TrainLoss : 1.5798163   |     TestLoss : 1.0008395    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
65    ->     TrainLoss : 1.5798169   |     TestLoss : 1.0008407    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
66    ->     TrainLoss : 1.5798175   |     TestLoss : 1.0008419    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
67    ->     TrainLoss : 1.5798182   |     TestLoss : 1.0008431    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
68    ->     TrainLoss : 1.5798190   |     TestLoss : 1.0008443    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
69    ->     TrainLoss : 1.5798198   |     TestLoss : 1.0008455    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
70    ->     TrainLoss : 1.5798206   |     TestLoss : 1.0008467    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
71    ->     TrainLoss : 1.5798215   |     TestLoss : 1.0008480    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
72    ->     TrainLoss : 1.5798224   |     TestLoss : 1.0008492    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
73    ->     TrainLoss : 1.5798234   |     TestLoss : 1.0008504    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
74    ->     TrainLoss : 1.5798244   |     TestLoss : 1.0008516    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
75    ->     TrainLoss : 1.5798254   |     TestLoss : 1.0008528    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
76    ->     TrainLoss : 1.5798264   |     TestLoss : 1.0008540    |
```

House Testing ETA = 0.02

```
voteDNNTEST = DC.CrossValidation(dp.houseNP80,5,[16,16,16,1],500,
0.02, 10, 0.001, 0.5,'sigmoid', 'momentum')
```

Sample Output:

```
490   ->    TrainLoss : 2.2484288   |     TestLoss : 1.1809602    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
491   ->    TrainLoss : 2.2484483   |     TestLoss : 1.1809716    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
492   ->    TrainLoss : 2.2484679   |     TestLoss : 1.1809831    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
493   ->    TrainLoss : 2.2484874   |     TestLoss : 1.1809946    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
494   ->    TrainLoss : 2.2485070   |     TestLoss : 1.1810060    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
495   ->    TrainLoss : 2.2485266   |     TestLoss : 1.1810175    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
496   ->    TrainLoss : 2.2485461   |     TestLoss : 1.1810289    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
497   ->    TrainLoss : 2.2485657   |     TestLoss : 1.1810403    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
498   ->    TrainLoss : 2.2485852   |     TestLoss : 1.1810517    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
499   ->    TrainLoss : 2.2486048   |     TestLoss : 1.1810631    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
500   ->    TrainLoss : 2.2486244   |     TestLoss : 1.1810745    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000

Process finished with exit code 0
```

Car DNN Tunning:

```
# Deep Neural Network
# Neural Network Architecture 6 Input -> 6 Hidden 1 Inputs -> 6
Hidden 2 Inputs - > Output
for eta in etavalues:
   carTuning = DC.CrossValidation(dp.carTuningNP,5,[6,6,6,1],500,
eta, 10, 0.001, 0.5,'sigmoid', 'momentum')
```

Sample Output:

```
490   ->    TrainLoss : 2.6481012   |     TestLoss : 1.4529070    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
491   ->    TrainLoss : 2.6482819   |     TestLoss : 1.4530739    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
492   ->    TrainLoss : 2.6484626   |     TestLoss : 1.4532408    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
493   ->    TrainLoss : 2.6486432   |     TestLoss : 1.4534075    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
494   ->    TrainLoss : 2.6488238   |     TestLoss : 1.4535741    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
495   ->    TrainLoss : 2.6490043   |     TestLoss : 1.4537406    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
496   ->    TrainLoss : 2.6491848   |     TestLoss : 1.4539070    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
497   ->    TrainLoss : 2.6493652   |     TestLoss : 1.4540733    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
498   ->    TrainLoss : 2.6495457   |     TestLoss : 1.4542395    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
499   ->    TrainLoss : 2.6497261   |     TestLoss : 1.4544056    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
500   ->    TrainLoss : 2.6499064   |     TestLoss : 1.4545716    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
```

Car DNN Testing:

```
carTesting = DC.CrossValidation(dp.carNP80,5,[6,6,6,1],500, 0.03, 10,
0.001, 0.5,'sigmoid', 'momentum')
```

Sample Output:

```
490   ->    TrainLoss : 3.1987046   |     TestLoss : 2.8571934    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
491   ->    TrainLoss : 3.1993960   |     TestLoss : 2.8581238    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
492   ->    TrainLoss : 3.2000874   |     TestLoss : 2.8590540    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
493   ->    TrainLoss : 3.2007789   |     TestLoss : 2.8599839    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
494   ->    TrainLoss : 3.2014706   |     TestLoss : 2.8609135    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
495   ->    TrainLoss : 3.2021623   |     TestLoss : 2.8618428    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
496   ->    TrainLoss : 3.2028541   |     TestLoss : 2.8627718    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
497   ->    TrainLoss : 3.2035460   |     TestLoss : 2.8637005    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
498   ->    TrainLoss : 3.2042379   |     TestLoss : 2.8646290    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
499   ->    TrainLoss : 3.2049300   |     TestLoss : 2.8655571    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000
500   ->    TrainLoss : 3.2056222   |     TestLoss : 2.8664850    |
TrainAccuracy : 1.0000000     |     TestAccuracy: 1.0000000

Process finished with exit code 0
```

AUTOENCODER AND DNN

ABALONE

```
etavalues = [0.01, 0.015, 0.02, 0.03, 0.035]

# AUTOENCODER THEN TO NETWORK
# Autoencoder Network Architecture 9 Input -> 8 Hidden 1 Inputs -> 9
Output
# DNN Network Architecture 8 Inputs -> 9 H1 Nodes -> 8 H2 Nodes -> 1
Ouput
for eta in etavalues:
   abaloneAENN = AE.AENN(dp.abaloneTuningNP, 5, 200, eta, 0.001,
0.05)
   abaloneTuningAE = abaloneAENN.fitALL(dp.abaloneTuningNP)
```

Experiences Vanishing gradients problem depending on gradient

```
Sample output:

This is how many Features X has
 8
This is the Autoencoding Layers
 [8, 7, 8]
This is the DNN Layers
 [7, 8, 7, 1]
Initializing Traditional DNN

Best Weights After Training Autoencoder
 [[nan nan nan nan nan nan nan nan]
 [nan nan nan nan nan nan nan nan]
 [nan nan nan nan nan nan nan nan]
 [nan nan nan nan nan nan nan nan]
 [nan nan nan nan nan nan nan nan]
 [nan nan nan nan nan nan nan nan]
 [nan nan nan nan nan nan nan nan]]
X Originally
 [[ 0.         -0.05405405 -0.17647059 ... -0.81170141 -0.79723502
  -0.76382661]
 [ 1.          0.12162162  0.09243697 ... -0.70880968 -0.75115207
```

```
  -0.6243149 ]
 [-1.         0.24324324  0.17647059 ... -0.63483524 -0.55102041
  -0.59441953]
 ...
 [ 1.         0.31081081  0.2605042  ... -0.46334902 -0.5483871
  -0.54658695]
 [-1.         0.32432432  0.32773109 ... -0.50369872 -0.37195523
  -0.50672646]
 [ 1.         0.71621622  0.68067227 ...  0.27034297 -0.00987492
  -0.01644245]]
X Encoded from Trained Autoencoders
 [[nan nan nan ... nan nan nan]
 [nan nan nan ... nan nan nan]
 [nan nan nan ... nan nan nan]
 ...
 [nan nan nan ... nan nan nan]
 [nan nan nan ... nan nan nan]
 [nan nan nan ... nan nan nan]]
Now attached Attaching To Other Network

Dataframe
       0   1   2   3   4   5   6         Y
0   NaN NaN NaN NaN NaN NaN NaN -0.500000
1   NaN NaN NaN NaN NaN NaN NaN -0.285714
2   NaN NaN NaN NaN NaN NaN NaN -0.357143
3   NaN NaN NaN NaN NaN NaN NaN -0.571429
4   NaN NaN NaN NaN NaN NaN NaN -0.285714
..   ..  ..  ..  ..  ..  ..  ..       ...
830 NaN NaN NaN NaN NaN NaN NaN -0.642857
831 NaN NaN NaN NaN NaN NaN NaN -0.285714
832 NaN NaN NaN NaN NaN NaN NaN -0.500000
833 NaN NaN NaN NaN NaN NaN NaN -0.285714
834 NaN NaN NaN NaN NaN NaN NaN -0.214286

[835 rows x 8 columns]
Numpy
 [[        nan         nan         nan ...         nan         nan
  -0.5       ]
 [        nan         nan         nan ...         nan         nan
  -0.28571429]
 [        nan         nan         nan ...         nan         nan
  -0.35714286]
 ...
```

```
 [         nan          nan          nan ...          nan          nan
  -0.5       ]
 [         nan          nan          nan ...          nan          nan
  -0.28571429]
 [         nan          nan          nan ...          nan          nan
  -0.21428571]]
Fold 1
Testing set size : (167, 8)
Training set size : (668, 8)
```

Best Eta at 0.015

```
# Best Eta is 0.015
abaloneTestAENN = AE.AENN(dp.abaloneNP80, 5, 200, 0.015, 0.001, 0.05)
abaloneTestAENN.fitALL(dp.abaloneNP80)
```

Testing abalone:

```
This is how many Features X has
 8
This is the Autoencoding Layers
 [8, 7, 8]
This is the DNN Layers
 [7, 8, 7, 1]
Initializing Traditional DNN

Best Weights After Training Autoencoder
 [[4.60211574e+09 4.60211573e+09 4.60211574e+09 4.60211573e+09
  4.60211573e+09 4.60211573e+09 4.60211573e+09 4.60211573e+09]
 [4.60212077e+09 4.60212077e+09 4.60212077e+09 4.60212077e+09
  4.60212077e+09 4.60212077e+09 4.60212077e+09 4.60212077e+09]
 [4.60210008e+09 4.60210009e+09 4.60210009e+09 4.60210009e+09
  4.60210008e+09 4.60210009e+09 4.60210008e+09 4.60210008e+09]
 [4.60211244e+09 4.60211244e+09 4.60211244e+09 4.60211244e+09
  4.60211244e+09 4.60211244e+09 4.60211244e+09 4.60211244e+09]
 [4.60211069e+09 4.60211069e+09 4.60211069e+09 4.60211069e+09
  4.60211069e+09 4.60211069e+09 4.60211069e+09 4.60211069e+09]
 [4.60212660e+09 4.60212660e+09 4.60212660e+09 4.60212660e+09
  4.60212660e+09 4.60212660e+09 4.60212660e+09 4.60212660e+09]
```

```
 [4.60210723e+09 4.60210723e+09 4.60210723e+09 4.60210723e+09
  4.60210723e+09 4.60210723e+09 4.60210723e+09 4.60210723e+09]]
X Originally
 [[-1.          0.48648649  0.46218487 ... -0.10356422 -0.31928901
  -0.34529148]
 [-1.          0.77027027  0.66386555 ... -0.27303295 -0.06254115
   0.18584953]
 [ 1.          0.52702703  0.46218487 ... -0.3597848  -0.29295589
  -0.32536124]
 ...
 [ 0.          0.35135135  0.24369748 ... -0.51176866 -0.48518762
  -0.5266567 ]
 [ 1.          0.09459459  0.04201681 ... -0.58170814 -0.69321922
  -0.72396612]
 [ 0.         -0.02702703 -0.09243697 ... -0.78749159 -0.7801185
  -0.81365222]]
X Encoded from Trained Autoencoders
 [[-7.38391337e+09 -7.38392146e+09 -7.38388826e+09 ... -7.38390528e+09
  -7.38393081e+09 -7.38389973e+09]
 [-1.22393239e+07 -1.22393373e+07 -1.22392821e+07 ... -1.22393109e+07
  -1.22393528e+07 -1.22393015e+07]
 [ 3.83779943e+08  3.83780363e+08  3.83778638e+08 ...  3.83779522e+08
   3.83780849e+08  3.83779234e+08]
 ...
 [-9.78800712e+09 -9.78801784e+09 -9.78797384e+09 ... -9.78799639e+09
  -9.78803023e+09 -9.78798904e+09]
 [-1.02408232e+10 -1.02408345e+10 -1.02407884e+10 ... -1.02408120e+10
  -1.02408474e+10 -1.02408043e+10]
 [-1.86140290e+10 -1.86140494e+10 -1.86139657e+10 ... -1.86140086e+10
  -1.86140729e+10 -1.86139946e+10]]
Now attached Attaching To Other Network

Dataframe
                  0             1  ...             6         Y
C:\Users\wsven\AppData\Local\Programs\Python\Python38\lib\site-packages\num
py\lib\function_base.py:4454: VisibleDeprecationWarning: Creating an
ndarray from ragged nested sequences (which is a list-or-tuple of
lists-or-tuples-or ndarrays with different lengths or shapes) is
deprecated. If you meant to do this, you must specify 'dtype=object' when
creating the ndarray.
  arr = asarray(arr)
0    -7.383913e+09 -7.383921e+09  ... -7.383900e+09 -0.357143
1    -1.223932e+07 -1.223934e+07  ... -1.223930e+07 -0.071429
```

```
2     3.837799e+08  3.837804e+08  ...  3.837792e+08 -0.428571
3    -1.821319e+10 -1.821321e+10  ... -1.821316e+10 -0.428571
4     8.095362e+09  8.095371e+09  ...  8.095347e+09 -0.214286
...            ...           ...  ...           ...       ...
3337 -1.733969e+09 -1.733970e+09  ... -1.733965e+09 -0.500000
3338 -1.630077e+10 -1.630079e+10  ... -1.630074e+10 -0.642857
3339 -9.788007e+09 -9.788018e+09  ... -9.787989e+09 -0.357143
3340 -1.024082e+10 -1.024083e+10  ... -1.024080e+10 -0.571429
3341 -1.861403e+10 -1.861405e+10  ... -1.861399e+10 -0.571429

[3342 rows x 8 columns]
Numpy
 [[-7.38391337e+09 -7.38392146e+09 -7.38388826e+09 ... -7.38393081e+09
  -7.38389973e+09 -3.57142857e-01]
 [-1.22393239e+07 -1.22393373e+07 -1.22392821e+07 ... -1.22393528e+07
  -1.22393015e+07 -7.14285714e-02]
 [ 3.83779943e+08  3.83780363e+08  3.83778638e+08 ...  3.83780849e+08
   3.83779234e+08 -4.28571429e-01]
 ...
 [-9.78800712e+09 -9.78801784e+09 -9.78797384e+09 ... -9.78803023e+09
  -9.78798904e+09 -3.57142857e-01]
 [-1.02408232e+10 -1.02408345e+10 -1.02407884e+10 ... -1.02408474e+10
  -1.02408043e+10 -5.71428571e-01]
 [-1.86140290e+10 -1.86140494e+10 -1.86139657e+10 ... -1.86140729e+10
  -1.86139946e+10 -5.71428571e-01]]
Fold 1
Testing set size : (669, 8)
Training set size : (2673, 8)

0     ->    TrainL : -0.1080319     |     TestL : -0.0659500      |
TrainMSE : 0.3489202    |     TestMSE: 0.3926883
1     ->    TrainL : -0.1057847     |     TestL : -0.0638129      |
TrainMSE : 0.3377872    |     TestMSE: 0.3819313
2     ->    TrainL : -0.1032184     |     TestL : -0.0615009      |
TrainMSE : 0.3263924    |     TestMSE: 0.3709219
3     ->    TrainL : -0.1004432     |     TestL : -0.0590269      |
TrainMSE : 0.3149412    |     TestMSE: 0.3598577
4     ->    TrainL : -0.0974743     |     TestL : -0.0564025      |
TrainMSE : 0.3036298    |     TestMSE: 0.3489279
5     ->    TrainL : -0.0943260     |     TestL : -0.0536431      |
TrainMSE : 0.2926607    |     TestMSE: 0.3383275
6     ->    TrainL : -0.0910166     |     TestL : -0.0507671      |
TrainMSE : 0.2822335    |     TestMSE: 0.3282488
```

```
7     ->    TrainL : -0.0875680     |     TestL : -0.0477952      |
TrainMSE : 0.2725343    |     TestMSE: 0.3188708
8     ->    TrainL : -0.0840052     |     TestL : -0.0447495      |
TrainMSE : 0.2637261    |     TestMSE: 0.3103508
9     ->    TrainL : -0.0803545     |     TestL : -0.0416525      |
TrainMSE : 0.2559426    |     TestMSE: 0.3028174
10    ->    TrainL : -0.0766430     |     TestL : -0.0385262      |
TrainMSE : 0.2492836    |     TestMSE: 0.2963668
```

FOREST FIRE AUTOENCODER PLUS DNN

```
# AUTOENCODER THEN TO NETWORK
# Autoencoder Network Architecture 12 Input -> 11 Hidden 1 Inputs ->
12 Output
# DNN Network Architecture 8 Inputs -> 11 H1 Nodes -> 10 H2 Nodes ->
1 Ouput
for eta in forestetavalues:
   forestAENN = AE.AENN(dp.forestTuningNP, 5, 200, eta, 0.001, 0.05)
   forestAENN.fitALL(dp.forestTuningNP)
```

All eta except 0.02 experienced the vanishing gradient problem

```
This is how many Features X has
 12
This is the Autoencoding Layers
 [12, 11, 12]
This is the DNN Layers
 [11, 12, 11, 1]
Initializing Traditional DNN

Best Weights After Training Autoencoder
 [[-6.00910161e+13 -6.00910161e+13 -6.00910161e+13 -6.00910161e+13
  -6.00910161e+13 -6.00910161e+13 -6.00910161e+13 -6.00910161e+13
  -6.00910161e+13 -6.00910161e+13 -6.00910161e+13 -6.00910161e+13]
 [-6.00910177e+13 -6.00910177e+13 -6.00910177e+13 -6.00910177e+13
  -6.00910177e+13 -6.00910177e+13 -6.00910177e+13 -6.00910177e+13
  -6.00910177e+13 -6.00910177e+13 -6.00910177e+13 -6.00910177e+13]
 [-6.00910184e+13 -6.00910184e+13 -6.00910184e+13 -6.00910184e+13
  -6.00910184e+13 -6.00910184e+13 -6.00910184e+13 -6.00910184e+13
  -6.00910184e+13 -6.00910184e+13 -6.00910184e+13 -6.00910184e+13]
 [-6.00910167e+13 -6.00910167e+13 -6.00910167e+13 -6.00910167e+13
  -6.00910167e+13 -6.00910167e+13 -6.00910167e+13 -6.00910167e+13
  -6.00910167e+13 -6.00910167e+13 -6.00910167e+13 -6.00910167e+13]
 [-6.00910166e+13 -6.00910166e+13 -6.00910166e+13 -6.00910166e+13
  -6.00910166e+13 -6.00910166e+13 -6.00910166e+13 -6.00910166e+13
  -6.00910166e+13 -6.00910166e+13 -6.00910166e+13 -6.00910166e+13]
 [-6.00910175e+13 -6.00910175e+13 -6.00910175e+13 -6.00910175e+13
  -6.00910175e+13 -6.00910175e+13 -6.00910175e+13 -6.00910175e+13
  -6.00910175e+13 -6.00910175e+13 -6.00910175e+13 -6.00910175e+13]
 [-6.00910134e+13 -6.00910134e+13 -6.00910134e+13 -6.00910134e+13
```

```
  -6.00910134e+13 -6.00910134e+13 -6.00910134e+13 -6.00910134e+13
  -6.00910134e+13 -6.00910134e+13 -6.00910134e+13 -6.00910134e+13]
 [-6.00910158e+13 -6.00910158e+13 -6.00910158e+13 -6.00910158e+13
  -6.00910158e+13 -6.00910158e+13 -6.00910158e+13 -6.00910158e+13
  -6.00910158e+13 -6.00910158e+13 -6.00910158e+13 -6.00910158e+13]
 [-6.00910179e+13 -6.00910179e+13 -6.00910179e+13 -6.00910179e+13
  -6.00910179e+13 -6.00910179e+13 -6.00910179e+13 -6.00910179e+13
  -6.00910179e+13 -6.00910179e+13 -6.00910179e+13 -6.00910179e+13]
 [-6.00910168e+13 -6.00910168e+13 -6.00910168e+13 -6.00910168e+13
  -6.00910168e+13 -6.00910168e+13 -6.00910168e+13 -6.00910168e+13
  -6.00910168e+13 -6.00910168e+13 -6.00910168e+13 -6.00910168e+13]
 [-6.00910146e+13 -6.00910146e+13 -6.00910146e+13 -6.00910146e+13
  -6.00910146e+13 -6.00910146e+13 -6.00910146e+13 -6.00910146e+13
  -6.00910146e+13 -6.00910146e+13 -6.00910146e+13 -6.00910146e+13]]
X Originally
 [[ 0.5        -0.42857143  0.81818182 ... -0.57647059 -0.88888889
  -1.        ]
 [ 0.75        0.14285714  0.27272727 ...  0.92941176 -0.2
  -0.9375    ]
 [ 0.25       -0.14285714  1.         ... -0.31764706  0.11111111
  -1.        ]
 ...
 [-0.75       -0.42857143 -0.81818182 ... -0.69411765 -0.28888889
  -1.        ]
 [ 0.25        0.14285714 -0.81818182 ... -0.36470588 -0.6
  -1.        ]
 [ 0.25       -0.71428571  0.63636364 ... -0.62352941 -0.08888889
  -1.        ]]
X Encoded from Trained Autoencoders
 [[ 6.07781562e+13  6.07781578e+13  6.07781585e+13 ...  6.07781580e+13
   6.07781569e+13  6.07781547e+13]
 [ 1.23807455e+14  1.23807459e+14  1.23807460e+14 ...  1.23807459e+14
   1.23807457e+14  1.23807452e+14]
 [-1.64624800e+13 -1.64624805e+13 -1.64624807e+13 ... -1.64624805e+13
  -1.64624802e+13 -1.64624796e+13]
 ...
 [ 1.67372998e+14  1.67373002e+14  1.67373004e+14 ...  1.67373003e+14
   1.67373000e+14  1.67372994e+14]
 [ 7.97180769e+13  7.97180791e+13  7.97180800e+13 ...  7.97180793e+13
   7.97180779e+13  7.97180750e+13]
 [ 2.04503760e+14  2.04503765e+14  2.04503768e+14 ...  2.04503766e+14
   2.04503762e+14  2.04503755e+14]]
Now attached Attaching To Other Network
```

```
Dataframe
                  0             1  ...            10         Y
0    6.077816e+13  6.077816e+13  ...  6.077815e+13 -1.000000
1    1.238075e+14  1.238075e+14  ...  1.238075e+14 -1.000000
2   -1.646248e+13 -1.646248e+13  ... -1.646248e+13 -1.000000
3    5.672570e+13  5.672570e+13  ...  5.672570e+13 -1.000000
4    2.341109e+13  2.341109e+13  ...  2.341109e+13 -1.000000
..            ...           ...  ...           ...       ...
98   2.108257e+13  2.108257e+13  ...  2.108257e+13 -0.590401
99   9.832300e+13  9.832300e+13  ...  9.832300e+13  0.610468
100  1.673730e+14  1.673730e+14  ...  1.673730e+14 -0.679320
101  7.971808e+13  7.971808e+13  ...  7.971807e+13 -1.000000
102  2.045038e+14  2.045038e+14  ...  2.045038e+14 -1.000000

[103 rows x 12 columns]
C:\Users\wsven\AppData\Local\Programs\Python\Python38\lib\site-packages\num
py\lib\function_base.py:4454: VisibleDeprecationWarning: Creating an
ndarray from ragged nested sequences (which is a list-or-tuple of
lists-or-tuples-or ndarrays with different lengths or shapes) is
deprecated. If you meant to do this, you must specify 'dtype=object' when
creating the ndarray.
  arr = asarray(arr)
Numpy
 [[ 6.07781562e+13  6.07781578e+13  6.07781585e+13 ...  6.07781569e+13
   6.07781547e+13 -1.00000000e+00]
 [ 1.23807455e+14  1.23807459e+14  1.23807460e+14 ...  1.23807457e+14
   1.23807452e+14 -1.00000000e+00]
 [-1.64624800e+13 -1.64624805e+13 -1.64624807e+13 ... -1.64624802e+13
  -1.64624796e+13 -1.00000000e+00]
 ...
 [ 1.67372998e+14  1.67373002e+14  1.67373004e+14 ...  1.67373000e+14
   1.67372994e+14 -6.79320025e-01]
 [ 7.97180769e+13  7.97180791e+13  7.97180800e+13 ...  7.97180779e+13
   7.97180750e+13 -1.00000000e+00]
 [ 2.04503760e+14  2.04503765e+14  2.04503768e+14 ...  2.04503762e+14
   2.04503755e+14 -1.00000000e+00]]
Fold 1
Testing set size : (21, 12)
Training set size : (82, 12)

0     ->    TrainL : 0.0904077      |     TestL : 0.0665626 |     TrainMSE :
2.6659721   |     TestMSE: 2.6649089
```

```
1      ->    TrainL : 0.0902426      |      TestL : 0.0690841 |     TrainMSE :
2.6581492   |     TestMSE: 2.6274184
2      ->    TrainL : 0.0898865      |      TestL : 0.0749513 |     TrainMSE :
2.6747316   |     TestMSE: 2.5914598
3      ->    TrainL : 0.0890267      |      TestL : 0.0783447 |     TrainMSE :
2.5618825   |     TestMSE: 2.4491895
4      ->    TrainL : 0.0785857      |      TestL : 0.0680319 |     TrainMSE :
2.2012482   |     TestMSE: 2.0966604
5      ->    TrainL : 0.0652530      |      TestL : 0.0657510 |     TrainMSE :
2.1291283   |     TestMSE: 2.0255771
6      ->    TrainL : 0.0627549      |      TestL : 0.0697852 |     TrainMSE :
2.2617504   |     TestMSE: 2.1555442
7      ->    TrainL : 0.0677959      |      TestL : 0.0732837 |     TrainMSE :
2.3829946   |     TestMSE: 2.2740771
8      ->    TrainL : 0.0723805      |      TestL : 0.0779193 |     TrainMSE :
2.5543859   |     TestMSE: 2.4414275
9      ->    TrainL : 0.0787983      |      TestL : 0.0794438 |     TrainMSE :
2.6299607   |     TestMSE: 2.5152701
10     ->    TrainL : 0.0823773      |      TestL : 0.0748734 |     TrainMSE :
2.6305873   |     TestMSE: 2.5393082
```

Testing foret data on Eta = 0.02

```
# Best Eta is 0.02
forestTestAENN = AE.AENN(dp.forestNP80, 5, 200, 0.02, 0.001, 0.05)
forestTestAENN.fitALL(dp.forestNP80)
```

Mini output

```
Fold 1
Testing set size : (83, 12)
Training set size : (331, 12)

0      ->    TrainL : -0.0860314     |      TestL : -0.0567672      |
TrainMSE : 0.6459747    |     TestMSE: 0.5232859
1      ->    TrainL : -0.0841130     |      TestL : -0.0533022      |
TrainMSE : 0.6290264    |     TestMSE: 0.5097785
2      ->    TrainL : -0.0799834     |      TestL : -0.0466436      |
TrainMSE : 0.6065482    |     TestMSE: 0.4938912
3      ->    TrainL : -0.0720583     |      TestL : -0.0358700      |
TrainMSE : 0.5974830    |     TestMSE: 0.4955540
4      ->    TrainL : -0.0592437     |      TestL : 0.0242499 |     TrainMSE :
```

```
1.1581331   |     TestMSE: 1.1192019
5     ->    TrainL : 0.0123617      |     TestL : 0.0809670 |     TrainMSE :
2.6486223   |     TestMSE: 2.7038493
6     ->    TrainL : 0.0850107      |     TestL : 0.0796243 |     TrainMSE :
2.6220765   |     TestMSE: 2.6951179
7     ->    TrainL : 0.0866748      |     TestL : 0.0728484 |     TrainMSE :
2.5313648   |     TestMSE: 2.6574312
8     ->    TrainL : 0.0891586      |     TestL : 0.0739718 |     TrainMSE :
2.5413869   |     TestMSE: 2.6599228
9     ->    TrainL : 0.0889358      |     TestL : 0.0750350 |     TrainMSE :
2.5527146   |     TestMSE: 2.6638480
10    ->    TrainL : 0.0886863      |     TestL : 0.0750296 |     TrainMSE :
2.5519859   |     TestMSE: 2.6629177
```

COMPUTER AUTOENCODER TO DNN

TUNING

```
# AUTOENCODER THEN TO NETWORK
# Autoencoder Network Architecture 8 Input -> 7 Hidden 1 Inputs -> 8 Output
# DNN Network Architecture 7 Inputs -> 8 H1 Nodes -> 8 H2 Nodes -> 1 Output
for eta in computeretavalues:
    computerAENN = AE.AENN(dp.computerTuningNP, 5, 200, eta, 0.001, 0.05)
    computerAENN.fitALL(dp.computerTuningNP)
```

NO VANISHING GRADIENT PROBLEMS:

MINI SAMPLE OUTPUT:

```
This is how many Features X has
 8
This is the Autoencoding Layers
 [8, 7, 8]
This is the DNN Layers
 [7, 8, 7, 1]
Initializing Traditional DNN

Best Weights After Training Autoencoder
 [[-6.9092296e+24 -6.9092296e+24 -6.9092296e+24 -6.9092296e+24
  -6.9092296e+24 -6.9092296e+24 -6.9092296e+24 -6.9092296e+24]
 [-6.9092296e+24 -6.9092296e+24 -6.9092296e+24 -6.9092296e+24
  -6.9092296e+24 -6.9092296e+24 -6.9092296e+24 -6.9092296e+24]
 [-6.9092296e+24 -6.9092296e+24 -6.9092296e+24 -6.9092296e+24
  -6.9092296e+24 -6.9092296e+24 -6.9092296e+24 -6.9092296e+24]
 [-6.9092296e+24 -6.9092296e+24 -6.9092296e+24 -6.9092296e+24
  -6.9092296e+24 -6.9092296e+24 -6.9092296e+24 -6.9092296e+24]
 [-6.9092296e+24 -6.9092296e+24 -6.9092296e+24 -6.9092296e+24
  -6.9092296e+24 -6.9092296e+24 -6.9092296e+24 -6.9092296e+24]
 [-6.9092296e+24 -6.9092296e+24 -6.9092296e+24 -6.9092296e+24
  -6.9092296e+24 -6.9092296e+24 -6.9092296e+24 -6.9092296e+24]
 [-6.9092296e+24 -6.9092296e+24 -6.9092296e+24 -6.9092296e+24
```

```
  -6.9092296e+24 -6.9092296e+24 -6.9092296e+24 -6.9092296e+24]]
X Originally
 [[-0.93103448 -0.375      -0.98381659 -0.50300601 -0.5015015  -0.75
  -0.69230769 -0.81818182]
 [-0.93103448 -0.36538462 -0.98786244 -0.50300601 -0.001001   -0.5
  -0.69230769 -0.63636364]
 [-0.86206897  0.42307692 -0.48347943 -0.94138277 -0.90815816 -1.
  -0.96153846 -0.97727273]
 [-0.65517241  0.24038462 -0.79770735 -0.97119238 -0.93943944 -0.9375
  -0.84615385 -0.82954545]
 [-0.65517241  0.27884615 -0.83007417 -0.91007014 -0.80492993 -1.
  -0.80769231 -0.63636364]
 [-0.5862069   0.78846154 -0.59136885 -0.97194389 -0.93943944 -0.96875
  -0.96153846 -0.96590909]
 [-0.44827586  0.34615385 -0.98921106 -0.92196894 -0.92004505  0.0234375
  -0.53846154 -0.72727273]
 [-0.44827586  0.75961538 -0.95549562 -0.94138277 -0.87687688 -0.9375
  -0.96153846 -0.94318182]
 [-0.44827586  0.76923077 -0.95549562 -0.87875752 -0.75175175 -0.9375
  -0.96153846 -0.94318182]
 [-0.51724138 -0.92307692 -0.95549562 -0.94138277 -0.87687688 -0.9375
  -0.88461538 -0.94318182]
 [-0.37931034  0.98076923 -0.75320297 -0.97194389 -0.75175175 -0.9375
  -0.96153846 -0.90909091]
 [-0.24137931  0.63461538  0.05596763 -0.98797595 -0.75175175 -1.
  -0.96153846 -0.95454545]
 [-0.10344828  0.31730769 -0.92178018 -0.87875752 -0.75175175 -0.5
  -0.96153846 -0.56818182]
 [ 0.10344828 -0.85576923 -0.90155091 -0.98797595 -0.97072072 -1.
  -0.88461538 -0.88636364]
 [-0.03448276 -0.09615385 -0.6183412  -0.97995992 -0.62662663 -0.953125
  -0.96153846 -0.72727273]
 [-0.03448276 -0.08653846 -0.6183412  -0.99198397 -0.97797798 -0.953125
  -0.76923077 -0.72727273]
 [ 0.03448276  0.48076923 -0.57788267 -0.94138277 -0.87687688 -1.
  -0.88461538 -0.93181818]
 [ 0.03448276  0.49038462 -0.83412003 -0.87875752 -0.87687688 -1.
  -0.88461538 -0.93181818]
 [ 0.17241379 -0.78846154 -0.94605529 -0.75350701 -0.5015015  -0.9921875
  -0.76923077 -0.86363636]
 [ 0.17241379 -0.61538462 -0.48347943 -0.75350701 -0.75175175 -1.
  -0.96153846 -0.98863636]
 [ 0.17241379 -0.60576923  0.1908294  -0.94138277 -0.97072072 -1.
```

```
 -0.96153846 -0.97727273]
[ 0.17241379 -0.58653846  0.1908294  -0.94138277 -0.87687688 -0.96875
 -0.96153846 -0.97727273]
[ 0.17241379 -0.55769231 -0.71948753 -0.87875752 -0.87687688 -0.9375
 -0.88461538 -0.93181818]
[ 0.17241379 -0.54807692 -0.71948753 -0.87875752 -0.87687688 -0.9375
 -0.88461538 -0.93181818]
[ 0.17241379  0.          1.         -0.95591182 -0.93943944 -1.
 -1.         -1.        ]
[ 0.31034483  0.72115385 -0.95549562 -0.87875752 -0.5015015  -0.8125
 -0.76923077 -0.81818182]
[ 0.44827586  0.05769231 -0.86783547 -0.87875752 -0.75175175 -0.875
 -0.96153846 -0.96590909]
[ 0.44827586  0.06730769 -0.86783547 -0.87875752 -0.87687688 -0.984375
 -0.96153846 -0.94318182]
[ 0.44827586  0.07692308 -0.89885367 -0.87875752 -0.75175175 -0.75
 -0.96153846 -0.93181818]
[ 0.44827586  0.16346154 -0.92582603 -0.87875752 -0.75175175 -0.875
 -0.76923077 -0.90909091]
[ 0.44827586  0.19230769 -0.97572488 -0.50300601 -0.001001   -0.5
 -0.69230769 -0.72727273]
[ 0.44827586  0.22115385 -0.97167903 -0.50300601 -0.001001   -0.5
 -0.69230769 -0.72727273]
[ 0.51724138  0.85576923 -0.94740391 -0.94138277 -0.87687688 -1.
 -0.96153846 -0.93181818]
[ 0.51724138  0.90384615 -0.94740391 -0.75350701 -0.5015015  -1.
 -0.96153846 -0.90909091]
[ 0.51724138  0.95192308 -0.97167903 -0.75350701 -0.5015015   0.
 -0.38461538 -0.63636364]
[ 0.72413793 -0.33653846 -0.80714767 -0.94138277 -0.87687688 -0.9375
 -0.96153846 -0.84090909]
[ 0.79310345 -0.22115385 -0.69925826 -0.97194389 -0.93943944 -0.9375
 -0.96153846 -0.94318182]
[ 0.86206897 -0.99038462 -0.86648685 -0.87875752 -0.75175175 -0.75
 -0.80769231 -0.68181818]
[ 0.86206897 -0.98076923 -0.95549562 -0.87875752 -0.001001   -0.8125
 -0.76923077 -0.70454545]
[ 0.86206897 -0.94230769 -0.98246797 -0.50300601  1.         -0.25
 -0.53846154  1.        ]
[ 0.86206897 -0.03846154 -0.85569791 -0.94138277 -0.75175175 -1.
 -0.96153846 -0.90909091]
[ 0.86206897  0.04807692 -0.89076197 -0.94138277 -0.75175175 -0.75
 -0.92307692 -0.90909091]]
```

```
X Encoded from Trained Autoencoders
 [[3.83797209e+25 3.83797209e+25 3.83797209e+25 3.83797209e+25
  3.83797209e+25 3.83797209e+25 3.83797209e+25]
 [3.18996358e+25 3.18996358e+25 3.18996358e+25 3.18996358e+25
  3.18996358e+25 3.18996358e+25 3.18996358e+25]
 [3.94573914e+25 3.94573914e+25 3.94573914e+25 3.94573914e+25
  3.94573914e+25 3.94573914e+25 3.94573914e+25]
 [3.96335953e+25 3.96335953e+25 3.96335953e+25 3.96335953e+25
  3.96335953e+25 3.96335953e+25 3.96335953e+25]
 [3.70711705e+25 3.70711705e+25 3.70711705e+25 3.70711705e+25
  3.70711705e+25 3.70711705e+25 3.70711705e+25]
 [3.59051595e+25 3.59051595e+25 3.59051595e+25 3.59051595e+25
  3.59051595e+25 3.59051595e+25 3.59051595e+25]
 [2.88504819e+25 2.88504819e+25 2.88504819e+25 2.88504819e+25
  2.88504819e+25 2.88504819e+25 2.88504819e+25]
 [3.66509482e+25 3.66509482e+25 3.66509482e+25 3.66509482e+25
  3.66509482e+25 3.66509482e+25 3.66509482e+25]
 [3.52873028e+25 3.52873028e+25 3.52873028e+25 3.52873028e+25
  3.52873028e+25 3.52873028e+25 3.52873028e+25]
 [4.82220750e+25 4.82220750e+25 4.82220750e+25 4.82220750e+25
  4.82220750e+25 4.82220750e+25 4.82220750e+25]
 [3.23598542e+25 3.23598542e+25 3.23598542e+25 3.23598542e+25
  3.23598542e+25 3.23598542e+25 3.23598542e+25]
 [2.90644199e+25 2.90644199e+25 2.90644199e+25 2.90644199e+25
  2.90644199e+25 2.90644199e+25 2.90644199e+25]
 [3.01805534e+25 3.01805534e+25 3.01805534e+25 3.01805534e+25
  3.01805534e+25 3.01805534e+25 3.01805534e+25]
 [4.41053957e+25 4.41053957e+25 4.41053957e+25 4.41053957e+25
  4.41053957e+25 4.41053957e+25 4.41053957e+25]
 [3.45288786e+25 3.45288786e+25 3.45288786e+25 3.45288786e+25
  3.45288786e+25 3.45288786e+25 3.45288786e+25]
 [3.56443898e+25 3.56443898e+25 3.56443898e+25 3.56443898e+25
  3.56443898e+25 3.56443898e+25 3.56443898e+25]
 [3.24548892e+25 3.24548892e+25 3.24548892e+25 3.24548892e+25
  3.24548892e+25 3.24548892e+25 3.24548892e+25]
 [3.37261648e+25 3.37261648e+25 3.37261648e+25 3.37261648e+25
  3.37261648e+25 3.37261648e+25 3.37261648e+25]
 [3.76011757e+25 3.76011757e+25 3.76011757e+25 3.76011757e+25
  3.76011757e+25 3.76011757e+25 3.76011757e+25]
 [3.71846712e+25 3.71846712e+25 3.71846712e+25 3.71846712e+25
  3.71846712e+25 3.71846712e+25 3.71846712e+25]
 [3.51917513e+25 3.51917513e+25 3.51917513e+25 3.51917513e+25
  3.51917513e+25 3.51917513e+25 3.51917513e+25]
```

```
 [3.41945794e+25 3.41945794e+25 3.41945794e+25 3.41945794e+25
  3.41945794e+25 3.41945794e+25 3.41945794e+25]
 [3.87907226e+25 3.87907226e+25 3.87907226e+25 3.87907226e+25
  3.87907226e+25 3.87907226e+25 3.87907226e+25]
 [3.87242877e+25 3.87242877e+25 3.87242877e+25 3.87242877e+25
  3.87242877e+25 3.87242877e+25 3.87242877e+25]
 [2.57226297e+25 2.57226297e+25 2.57226297e+25 2.57226297e+25
  2.57226297e+25 2.57226297e+25 2.57226297e+25]
 [2.55929510e+25 2.55929510e+25 2.55929510e+25 2.55929510e+25
  2.55929510e+25 2.55929510e+25 2.55929510e+25]
 [3.31285407e+25 3.31285407e+25 3.31285407e+25 3.31285407e+25
  3.31285407e+25 3.31285407e+25 3.31285407e+25]
 [3.45252931e+25 3.45252931e+25 3.45252931e+25 3.45252931e+25
  3.45252931e+25 3.45252931e+25 3.45252931e+25]
 [3.21107872e+25 3.21107872e+25 3.21107872e+25 3.21107872e+25
  3.21107872e+25 3.21107872e+25 3.21107872e+25]
 [3.10771590e+25 3.10771590e+25 3.10771590e+25 3.10771590e+25
  3.10771590e+25 3.10771590e+25 3.10771590e+25]
 [1.90606904e+25 1.90606904e+25 1.90606904e+25 1.90606904e+25
  1.90606904e+25 1.90606904e+25 1.90606904e+25]
 [1.88334320e+25 1.88334320e+25 1.88334320e+25 1.88334320e+25
  1.88334320e+25 1.88334320e+25 1.88334320e+25]
 [2.96130243e+25 2.96130243e+25 2.96130243e+25 2.96130243e+25
  2.96130243e+25 2.96130243e+25 2.96130243e+25]
 [2.52321905e+25 2.52321905e+25 2.52321905e+25 2.52321905e+25
  2.52321905e+25 2.52321905e+25 2.52321905e+25]
 [1.22880794e+25 1.22880794e+25 1.22880794e+25 1.22880794e+25
  1.22880794e+25 1.22880794e+25 1.22880794e+25]
 [3.43924549e+25 3.43924549e+25 3.43924549e+25 3.43924549e+25
  3.43924549e+25 3.43924549e+25 3.43924549e+25]
 [3.37233435e+25 3.37233435e+25 3.37233435e+25 3.37233435e+25
  3.37233435e+25 3.37233435e+25 3.37233435e+25]
 [3.36121739e+25 3.36121739e+25 3.36121739e+25 3.36121739e+25
  3.36121739e+25 3.36121739e+25 3.36121739e+25]
 [2.92967269e+25 2.92967269e+25 2.92967269e+25 2.92967269e+25
  2.92967269e+25 2.92967269e+25 2.92967269e+25]
 [2.44707119e+24 2.44707119e+24 2.44707119e+24 2.44707119e+24
  2.44707119e+24 2.44707119e+24 2.44707119e+24]
 [3.17538130e+25 3.17538130e+25 3.17538130e+25 3.17538130e+25
  3.17538130e+25 3.17538130e+25 3.17538130e+25]
 [2.94051176e+25 2.94051176e+25 2.94051176e+25 2.94051176e+25
  2.94051176e+25 2.94051176e+25 2.94051176e+25]]
Now attached Attaching To Other Network
```

```
Dataframe
                 0             1  ...             6         Y
0   3.837972e+25  3.837972e+25  ...  3.837972e+25 -0.779720
1   3.189964e+25  3.189964e+25  ...  3.189964e+25 -0.454545
2   3.945739e+25  3.945739e+25  ...  3.945739e+25 -0.944056
3   3.963360e+25  3.963360e+25  ...  3.963360e+25 -0.977273
4   3.707117e+25  3.707117e+25  ...  3.707117e+25 -0.958042
5   3.590516e+25  3.590516e+25  ...  3.590516e+25 -0.889860
6   2.885048e+25  2.885048e+25  ...  2.885048e+25 -0.531469
7   3.665095e+25  3.665095e+25  ...  3.665095e+25 -0.959790
8   3.528730e+25  3.528730e+25  ...  3.528730e+25 -0.886364
9   4.822208e+25  4.822208e+25  ...  4.822208e+25 -0.965035
10  3.235985e+25  3.235985e+25  ...  3.235985e+25 -0.902098
11  2.906442e+25  2.906442e+25  ...  2.906442e+25 -0.989510
12  3.018055e+25  3.018055e+25  ...  3.018055e+25 -0.758741
13  4.410540e+25  4.410540e+25  ...  4.410540e+25 -0.980769
14  3.452888e+25  3.452888e+25  ...  3.452888e+25 -0.917832
15  3.564439e+25  3.564439e+25  ...  3.564439e+25 -0.947552
16  3.245489e+25  3.245489e+25  ...  3.245489e+25 -0.972028
17  3.372616e+25  3.372616e+25  ...  3.372616e+25 -0.959790
18  3.760118e+25  3.760118e+25  ...  3.760118e+25 -0.779720
19  3.718467e+25  3.718467e+25  ...  3.718467e+25 -0.954545
20  3.519175e+25  3.519175e+25  ...  3.519175e+25 -0.991259
21  3.419458e+25  3.419458e+25  ...  3.419458e+25 -0.979021
22  3.879072e+25  3.879072e+25  ...  3.879072e+25 -0.940559
23  3.872429e+25  3.872429e+25  ...  3.872429e+25 -0.951049
24  2.572263e+25  2.572263e+25  ...  2.572263e+25 -0.979021
25  2.559295e+25  2.559295e+25  ...  2.559295e+25 -0.916084
26  3.312854e+25  3.312854e+25  ...  3.312854e+25 -0.923077
27  3.452529e+25  3.452529e+25  ...  3.452529e+25 -0.940559
28  3.211079e+25  3.211079e+25  ...  3.211079e+25 -0.902098
29  3.107716e+25  3.107716e+25  ...  3.107716e+25 -0.826923
30  1.906069e+25  1.906069e+25  ...  1.906069e+25 -0.363636
31  1.883343e+25  1.883343e+25  ...  1.883343e+25 -0.440559
32  2.961302e+25  2.961302e+25  ...  2.961302e+25 -0.980769
33  2.523219e+25  2.523219e+25  ...  2.523219e+25 -0.930070
34  1.228808e+25  1.228808e+25  ...  1.228808e+25 -0.639860
35  3.439245e+25  3.439245e+25  ...  3.439245e+25 -0.944056
36  3.372334e+25  3.372334e+25  ...  3.372334e+25 -0.991259
37  3.361217e+25  3.361217e+25  ...  3.361217e+25 -0.888112
38  2.929673e+25  2.929673e+25  ...  2.929673e+25 -0.811189
39  2.447071e+24  2.447071e+24  ...  2.447071e+24  0.589161
```

```
40  3.175381e+25  3.175381e+25  ...  3.175381e+25 -0.937063
41  2.940512e+25  2.940512e+25  ...  2.940512e+25 -0.930070

[42 rows x 8 columns]
Numpy
 [[ 3.83797209e+25  3.83797209e+25  3.83797209e+25  3.83797209e+25
   3.83797209e+25  3.83797209e+25  3.83797209e+25 -7.79720280e-01]
 [ 3.18996358e+25  3.18996358e+25  3.18996358e+25  3.18996358e+25
   3.18996358e+25  3.18996358e+25  3.18996358e+25 -4.54545455e-01]
 [ 3.94573914e+25  3.94573914e+25  3.94573914e+25  3.94573914e+25
   3.94573914e+25  3.94573914e+25  3.94573914e+25 -9.44055944e-01]
 [ 3.96335953e+25  3.96335953e+25  3.96335953e+25  3.96335953e+25
   3.96335953e+25  3.96335953e+25  3.96335953e+25 -9.77272727e-01]
 [ 3.70711705e+25  3.70711705e+25  3.70711705e+25  3.70711705e+25
   3.70711705e+25  3.70711705e+25  3.70711705e+25 -9.58041958e-01]
 [ 3.59051595e+25  3.59051595e+25  3.59051595e+25  3.59051595e+25
   3.59051595e+25  3.59051595e+25  3.59051595e+25 -8.89860140e-01]
 [ 2.88504819e+25  2.88504819e+25  2.88504819e+25  2.88504819e+25
   2.88504819e+25  2.88504819e+25  2.88504819e+25 -5.31468531e-01]
 [ 3.66509482e+25  3.66509482e+25  3.66509482e+25  3.66509482e+25
   3.66509482e+25  3.66509482e+25  3.66509482e+25 -9.59790210e-01]
 [ 3.52873028e+25  3.52873028e+25  3.52873028e+25  3.52873028e+25
   3.52873028e+25  3.52873028e+25  3.52873028e+25 -8.86363636e-01]
 [ 4.82220750e+25  4.82220750e+25  4.82220750e+25  4.82220750e+25
   4.82220750e+25  4.82220750e+25  4.82220750e+25 -9.65034965e-01]
 [ 3.23598542e+25  3.23598542e+25  3.23598542e+25  3.23598542e+25
   3.23598542e+25  3.23598542e+25  3.23598542e+25 -9.02097902e-01]
 [ 2.90644199e+25  2.90644199e+25  2.90644199e+25  2.90644199e+25
   2.90644199e+25  2.90644199e+25  2.90644199e+25 -9.89510490e-01]
 [ 3.01805534e+25  3.01805534e+25  3.01805534e+25  3.01805534e+25
   3.01805534e+25  3.01805534e+25  3.01805534e+25 -7.58741259e-01]
 [ 4.41053957e+25  4.41053957e+25  4.41053957e+25  4.41053957e+25
   4.41053957e+25  4.41053957e+25  4.41053957e+25 -9.80769231e-01]
 [ 3.45288786e+25  3.45288786e+25  3.45288786e+25  3.45288786e+25
   3.45288786e+25  3.45288786e+25  3.45288786e+25 -9.17832168e-01]
 [ 3.56443898e+25  3.56443898e+25  3.56443898e+25  3.56443898e+25
   3.56443898e+25  3.56443898e+25  3.56443898e+25 -9.47552448e-01]
 [ 3.24548892e+25  3.24548892e+25  3.24548892e+25  3.24548892e+25
   3.24548892e+25  3.24548892e+25  3.24548892e+25 -9.72027972e-01]
 [ 3.37261648e+25  3.37261648e+25  3.37261648e+25  3.37261648e+25
   3.37261648e+25  3.37261648e+25  3.37261648e+25 -9.59790210e-01]
 [ 3.76011757e+25  3.76011757e+25  3.76011757e+25  3.76011757e+25
   3.76011757e+25  3.76011757e+25  3.76011757e+25 -7.79720280e-01]
```

```
[ 3.71846712e+25  3.71846712e+25  3.71846712e+25  3.71846712e+25
  3.71846712e+25  3.71846712e+25  3.71846712e+25 -9.54545455e-01]
[ 3.51917513e+25  3.51917513e+25  3.51917513e+25  3.51917513e+25
  3.51917513e+25  3.51917513e+25  3.51917513e+25 -9.91258741e-01]
[ 3.41945794e+25  3.41945794e+25  3.41945794e+25  3.41945794e+25
  3.41945794e+25  3.41945794e+25  3.41945794e+25 -9.79020979e-01]
[ 3.87907226e+25  3.87907226e+25  3.87907226e+25  3.87907226e+25
  3.87907226e+25  3.87907226e+25  3.87907226e+25 -9.40559441e-01]
[ 3.87242877e+25  3.87242877e+25  3.87242877e+25  3.87242877e+25
  3.87242877e+25  3.87242877e+25  3.87242877e+25 -9.51048951e-01]
[ 2.57226297e+25  2.57226297e+25  2.57226297e+25  2.57226297e+25
  2.57226297e+25  2.57226297e+25  2.57226297e+25 -9.79020979e-01]
[ 2.55929510e+25  2.55929510e+25  2.55929510e+25  2.55929510e+25
  2.55929510e+25  2.55929510e+25  2.55929510e+25 -9.16083916e-01]
[ 3.31285407e+25  3.31285407e+25  3.31285407e+25  3.31285407e+25
  3.31285407e+25  3.31285407e+25  3.31285407e+25 -9.23076923e-01]
[ 3.45252931e+25  3.45252931e+25  3.45252931e+25  3.45252931e+25
  3.45252931e+25  3.45252931e+25  3.45252931e+25 -9.40559441e-01]
[ 3.21107872e+25  3.21107872e+25  3.21107872e+25  3.21107872e+25
  3.21107872e+25  3.21107872e+25  3.21107872e+25 -9.02097902e-01]
[ 3.10771590e+25  3.10771590e+25  3.10771590e+25  3.10771590e+25
  3.10771590e+25  3.10771590e+25  3.10771590e+25 -8.26923077e-01]
[ 1.90606904e+25  1.90606904e+25  1.90606904e+25  1.90606904e+25
  1.90606904e+25  1.90606904e+25  1.90606904e+25 -3.63636364e-01]
[ 1.88334320e+25  1.88334320e+25  1.88334320e+25  1.88334320e+25
  1.88334320e+25  1.88334320e+25  1.88334320e+25 -4.40559441e-01]
[ 2.96130243e+25  2.96130243e+25  2.96130243e+25  2.96130243e+25
  2.96130243e+25  2.96130243e+25  2.96130243e+25 -9.80769231e-01]
[ 2.52321905e+25  2.52321905e+25  2.52321905e+25  2.52321905e+25
  2.52321905e+25  2.52321905e+25  2.52321905e+25 -9.30069930e-01]
[ 1.22880794e+25  1.22880794e+25  1.22880794e+25  1.22880794e+25
  1.22880794e+25  1.22880794e+25  1.22880794e+25 -6.39860140e-01]
[ 3.43924549e+25  3.43924549e+25  3.43924549e+25  3.43924549e+25
  3.43924549e+25  3.43924549e+25  3.43924549e+25 -9.44055944e-01]
[ 3.37233435e+25  3.37233435e+25  3.37233435e+25  3.37233435e+25
  3.37233435e+25  3.37233435e+25  3.37233435e+25 -9.91258741e-01]
[ 3.36121739e+25  3.36121739e+25  3.36121739e+25  3.36121739e+25
  3.36121739e+25  3.36121739e+25  3.36121739e+25 -8.88111888e-01]
[ 2.92967269e+25  2.92967269e+25  2.92967269e+25  2.92967269e+25
  2.92967269e+25  2.92967269e+25  2.92967269e+25 -8.11188811e-01]
[ 2.44707119e+24  2.44707119e+24  2.44707119e+24  2.44707119e+24
  2.44707119e+24  2.44707119e+24  2.44707119e+24  5.89160839e-01]
[ 3.17538130e+25  3.17538130e+25  3.17538130e+25  3.17538130e+25
```

```
   3.17538130e+25  3.17538130e+25  3.17538130e+25 -9.37062937e-01]
 [ 2.94051176e+25  2.94051176e+25  2.94051176e+25  2.94051176e+25
   2.94051176e+25  2.94051176e+25  2.94051176e+25 -9.30069930e-01]]
C:\Users\wsven\AppData\Local\Programs\Python\Python38\lib\site-packages\num
py\lib\function_base.py:4454: VisibleDeprecationWarning: Creating an
ndarray from ragged nested sequences (which is a list-or-tuple of
lists-or-tuples-or ndarrays with different lengths or shapes) is
deprecated. If you meant to do this, you must specify 'dtype=object' when
creating the ndarray.
  arr = asarray(arr)
Fold 1
Testing set size : (9, 8)
Training set size : (33, 8)

0     ->    TrainL : 0.1409196      |     TestL : 0.1407626 |     TrainMSE :
3.4677465   |     TestMSE: 3.0730376
1     ->    TrainL : 0.1407626      |     TestL : 0.1405686 |     TrainMSE :
3.4628920   |     TestMSE: 3.0684476
2     ->    TrainL : 0.1405686      |     TestL : 0.1403329 |     TrainMSE :
3.4569688   |     TestMSE: 3.0628473
3     ->    TrainL : 0.1403329      |     TestL : 0.1400401 |     TrainMSE :
3.4495774   |     TestMSE: 3.0558593
4     ->    TrainL : 0.1400401      |     TestL : 0.1396675 |     TrainMSE :
3.4401485   |     TestMSE: 3.0469456
5     ->    TrainL : 0.1396675      |     TestL : 0.1391847 |     TrainMSE :
3.4279035   |     TestMSE: 3.0353708
6     ->    TrainL : 0.1391847      |     TestL : 0.1385506 |     TrainMSE :
3.4118136   |     TestMSE: 3.0201634
7     ->    TrainL : 0.1385506      |     TestL : 0.1377134 |     TrainMSE :
3.3905844   |     TestMSE: 3.0001017
8     ->    TrainL : 0.1377134      |     TestL : 0.1347619 |     TrainMSE :
3.3159512   |     TestMSE: 2.9296034
9     ->    TrainL : 0.1347619      |     TestL : 0.1266500 |     TrainMSE :
3.1149993   |     TestMSE: 2.7400278
10    ->    TrainL : 0.1266500      |     TestL : 0.1173500 |     TrainMSE :
2.8925117   |     TestMSE: 2.5305844
```

Lowest MSE achieved at eta = 0.0125 with it being 0.03433 on the training and 0.0331 on testing while tuning

Computer Testing:

```
# # Best Eta is 0.0125
computerTestAENN = AE.AENN(dp.computerNP80, 5, 200, 0.0125, 0.001,
0.05)
computerTestAENN.fitREG(dp.computerNP80)
```

Sample Output

```
This is how many Features X has
 8
This is the Autoencoding Layers
 [8, 7, 8]
This is the DNN Layers
 [7, 8, 7, 1]
Initializing Traditional DNN
```

Best Weights After Training Autoencoder

```
 [[-20313.43134487 -20313.72336947 -20313.34326271 -20313.29683072
  -20313.65412322 -20313.38941077 -20313.67537443 -20313.58807961]
 [-20361.7251663  -20362.02062233 -20361.82941916 -20361.75013048
  -20361.65684895 -20361.70828645 -20361.85359267 -20361.84316593]
 [-20325.02140063 -20325.22335063 -20324.95475689 -20325.24620882
  -20324.99064919 -20325.2653241  -20325.35632745 -20325.28937165]
 [-20366.83860561 -20366.72319303 -20366.50249416 -20366.83946411
  -20366.62573223 -20366.68793814 -20366.52747412 -20366.84211238]
 [-20404.6401162  -20404.41633843 -20404.68721988 -20404.75178815
  -20404.45342978 -20404.83114262 -20404.56221979 -20404.39881917]
 [-20259.35571753 -20259.3757007  -20259.56486361 -20259.35633246
  -20259.46830531 -20259.14548656 -20259.11503799 -20259.10326595]
 [-20307.11770997 -20306.96182164 -20307.17406019 -20306.93098189
  -20306.84106182 -20307.09234467 -20306.77957744 -20306.81872823]]
X Originally
 [[-0.5862069   0.80769231 -0.59136885 ... -0.96875    -0.96153846
  -0.96590909]
 [ 0.17241379 -0.74038462 -0.98786244 ... -1.         -0.69230769
  -0.81818182]
```

```
 [-0.10344828  0.31730769 -0.92178018 ... -0.5        -0.96153846
  -0.56818182]
 ...
 [ 0.44827586  0.15384615 -0.92582603 ... -0.5        -0.69230769
  -0.81818182]
 [-0.51724138 -0.89423077 -0.95549562 ... -0.9375     -0.88461538
  -0.93181818]
 [-0.93103448 -0.27884615 -0.99190829 ... -0.5        -0.38461538
  -0.63636364]]
X Encoded from Trained Autoencoders
 [[104862.42886462 105111.61182778 104922.94118256 ... 105333.16775633
  104582.683435   104828.79466659]
 [103006.39140911 103251.32128517 103065.61049691 ... 103468.25017433
  102731.44934201 102973.15103703]
 [ 88732.29947759  88943.23102292  88783.39362423 ...  89130.49068327
   88495.70251208  88703.76408772]
```

```
Fold 1
Testing set size : (34, 8)
Training set size : (133, 8)

0     ->    TrainL : 0.1391464      |     TestL : 0.1388369 |     TrainMSE :
3.2697892   |     TestMSE: 3.3669603
1     ->    TrainL : 0.1388369      |     TestL : 0.1384459 |     TrainMSE :
3.2601774   |     TestMSE: 3.3571627
2     ->    TrainL : 0.1384459      |     TestL : 0.1379582 |     TrainMSE :
3.2481665   |     TestMSE: 3.3449193
3     ->    TrainL : 0.1379582      |     TestL : 0.1373344 |     TrainMSE :
3.2327876   |     TestMSE: 3.3292419
4     ->    TrainL : 0.1373344      |     TestL : 0.1365195 |     TrainMSE :
3.2127039   |     TestMSE: 3.3087673
5     ->    TrainL : 0.1365195      |     TestL : 0.1338153 |     TrainMSE :
3.1461686   |     TestMSE: 3.2409280
6     ->    TrainL : 0.1338153      |     TestL : 0.1321132 |     TrainMSE :
3.1047025   |     TestMSE: 3.1986421
7     ->    TrainL : 0.1321132      |     TestL : 0.1300166 |     TrainMSE :
3.0539771   |     TestMSE: 3.1469061
8     ->    TrainL : 0.1300166      |     TestL : 0.1274734 |     TrainMSE :
2.9929880   |     TestMSE: 3.0846904
9     ->    TrainL : 0.1274734      |     TestL : 0.1174542 |     TrainMSE :
2.7584938   |     TestMSE: 2.8453566
10    ->    TrainL : 0.1174542      |     TestL : 0.0987821 |     TrainMSE :
```

```
2.3475947   |     TestMSE: 2.4254356
```

Breast AUTOENCODER

```
this is y Matrix [[1. 0.]
 [1. 0.]
 [0. 1.]
 [0. 1.]
 [0. 1.]
 [0. 1.]
 [1. 0.]
 [0. 1.]
 [0. 1.]
 [1. 0.]
 [1. 0.]
 [1. 0.]
 [1. 0.]
 [0. 1.]
 [1. 0.]
 [0. 1.]
 [0. 1.]
 [0. 1.]
 [1. 0.]
 [0. 1.]
 [1. 0.]
 [1. 0.]
 [1. 0.]
 [0. 1.]
 [1. 0.]
 [0. 1.]
 [1. 0.]
 [0. 1.]]
This is the Loss [[9.34096736e-05 0.00000000e+00]
 [9.34096736e-05 0.00000000e+00]
 [0.00000000e+00 9.34096736e-05]
 [0.00000000e+00 9.34096736e-05]
 [0.00000000e+00 9.34096736e-05]
 [0.00000000e+00 9.34096736e-05]
 [9.34096736e-05 0.00000000e+00]
 [0.00000000e+00 9.34096736e-05]
 [0.00000000e+00 9.34096736e-05]
```

```
 [9.34096736e-05 0.00000000e+00]
 [9.34096736e-05 0.00000000e+00]
 [9.34096736e-05 0.00000000e+00]
 [9.34096736e-05 0.00000000e+00]
 [0.00000000e+00 9.34096736e-05]
 [9.34096736e-05 0.00000000e+00]
 [0.00000000e+00 9.34096736e-05]
 [0.00000000e+00 9.34096736e-05]
 [0.00000000e+00 9.34096736e-05]
 [9.34096736e-05 0.00000000e+00]
 [0.00000000e+00 9.34096736e-05]
 [9.34096736e-05 0.00000000e+00]
 [9.34096736e-05 0.00000000e+00]
 [9.34096736e-05 0.00000000e+00]
 [0.00000000e+00 9.34096736e-05]
 [9.34096736e-05 0.00000000e+00]
 [0.00000000e+00 9.34096736e-05]
 [9.34096736e-05 0.00000000e+00]
 [0.00000000e+00 9.34096736e-05]]
0.0015434870249310538
This is the gradient
[[ 2.44982234 -2.44981979]
 [ 1.79923244 -1.79922534]
 [ 1.84172933 -1.8417276 ]
 [ 1.54708651 -1.54708145]
 [ 1.73332692 -1.73331901]
 [ 2.62829866 -2.6282911 ]
 [ 1.73679708 -1.73678919]
 [ 1.90694898 -1.90694579]
 [ 0.95174048 -0.95173474]]
499   ->    TrainL : 0.0455132      |     TestL : 0.0015435 |     TrainAcc :
0.7857143   |     TestAcc: 0.6785714
```

Breast testing on 80%

```
# Testing on 80%
# Best accuracy at eta = 0.01 with accuracy being 0.8214
breastAENNTest= AE.AENN(dp.breastNP80, 5, 500, 0.01, 0.001, 0.05)
breastAENNTest.fitCLASS(dp.breastNP80)
```

Sample output

```
This is how many Features X has
 9
This is the Autoencoding Layers
 [9, 8, 9]
This is the DNN Layers
 [8, 9, 8, 1]
Initializing Traditional DNN

Best Weights After Training Autoencoder
 [[ 0.00821113  0.0690875   0.38719683  0.23396454  0.35455252  0.45310184
    0.41483783  0.21466091  0.30638825]
 [ 0.25768174  0.09562562  0.37957415  0.10387488  0.30346954  0.0600836
    0.21671466  0.19536755  0.08376999]
 [ 0.19103462  0.25567551  0.38730273  0.23455716  0.21094869  0.41356367
    0.13828499  0.0114883   0.32630317]
 [ 0.23775383  0.30181568  0.28574429  0.22771574  0.40067495  0.0496934
    0.19678719  0.223546    0.09768878]
 [ 0.35470728  0.42447231  0.1409958   0.3654743   0.25583167  0.14483413
    0.34958353  0.22869772  0.01861316]
 [-0.01714553  0.10496461  0.27973908  0.4024662   0.19429232  0.27959687
    0.28487011  0.31218075  0.29319338]
 [ 0.22673793  0.24070872 -0.01121168  0.34598881  0.30522699  0.28196703
    0.20662382  0.03179066  0.05551376]
 [ 0.37083205  0.14892192  0.05900859  0.42127336  0.34474255  0.42834936
    0.43409727  0.3219858   0.0984141 ]]
X Originally
 [[ 4  1  1 ...  2  1  1]
 [ 3  1  1 ...  2  1  1]
 [10  5  7 ...  8  9  1]
 ...
 [ 3  1  1 ...  5  8  1]
 [ 3  1  1 ...  2  1  1]
 [ 4  7  8 ...  9  1  1]]
X Encoded from Trained Autoencoders
 [[ 4.14222877  3.10955837  3.9186237  ...  3.12107739  3.43934469
    5.37565968]
 [ 3.22781396  2.73170942  2.90046175 ...  2.57902917  2.64867271
    4.14812891]
 [15.58005234 11.51797351 13.3534807  ... 12.87632519 10.81159487
   18.68261184]
 ...
```

```
 [ 8.57019804  6.77799321  5.13054123 ...  7.58959107  6.01441833
  10.61532334]
 [ 3.22781396  2.73170942  2.90046175 ...  2.57902917  2.64867271
   4.14812891]
 [14.51974343  9.09260785 12.91774819 ... 10.85382453  9.52768277
  14.25140947]]
Now attached Attaching To Other Network

Dataframe
              0          1          2  ...          6          7  Y
C:\Users\wsven\AppData\Local\Programs\Python\Python38\lib\site-packages\num
py\lib\function_base.py:4454: VisibleDeprecationWarning: Creating an
ndarray from ragged nested sequences (which is a list-or-tuple of
lists-or-tuples-or ndarrays with different lengths or shapes) is
deprecated. If you meant to do this, you must specify 'dtype=object' when
creating the ndarray.
  arr = asarray(arr)
0     4.142229   3.109558   3.918624  ...   3.439345   5.375660  0
1     3.227814   2.731709   2.900462  ...   2.648673   4.148129  0
2    15.580052  11.517974  13.353481  ...  10.811595  18.682612  1
3    15.481759  11.548195  14.586931  ...  12.100267  18.494892  1
4     3.626230   2.433061   2.656677  ...   2.401821   3.840562  0
..         ...        ...        ...  ...        ...        ... ..
554   3.615011   3.111284   3.287764  ...   2.637461   4.207137  0
555   3.211392   2.216346   2.518393  ...   2.195197   3.406465  0
556   8.570198   6.777993   5.130541  ...   6.014418  10.615323  0
557   3.227814   2.731709   2.900462  ...   2.648673   4.148129  0
558  14.519743   9.092608  12.917748  ...   9.527683  14.251409  1

[559 rows x 9 columns]
Numpy
 [[ 4.14222877  3.10955837  3.9186237  ...  3.43934469  5.37565968
   0.        ]
 [ 3.22781396  2.73170942  2.90046175 ...  2.64867271  4.14812891
   0.        ]
 [15.58005234 11.51797351 13.3534807  ... 10.81159487 18.68261184
   1.        ]
 ...
 [ 8.57019804  6.77799321  5.13054123 ...  6.01441833 10.61532334
   0.        ]
 [ 3.22781396  2.73170942  2.90046175 ...  2.64867271  4.14812891
   0.        ]
 [14.51974343  9.09260785 12.91774819 ...  9.52768277 14.25140947
```

```
   1.       ]]
Fold 1
Testing set size : (112, 10)
Training set size : (447, 10)

0     ->    TrainL : 0.2074516      |      TestL : 0.0040784 |     TrainAcc :
0.8434004   |     TestAcc: 0.8928571
1     ->    TrainL : 0.1087564      |      TestL : 0.0012973 |     TrainAcc :
0.8478747   |     TestAcc: 0.9107143
2     ->    TrainL : 0.0848944      |      TestL : 0.0008445 |     TrainAcc :
0.8456376   |     TestAcc: 0.9017857
3     ->    TrainL : 0.0735528      |      TestL : 0.0007363 |     TrainAcc :
0.8501119   |     TestAcc: 0.9017857
4     ->    TrainL : 0.0667052      |      TestL : 0.0007117 |     TrainAcc :
0.8523490   |     TestAcc: 0.9017857
5     ->    TrainL : 0.0623452      |      TestL : 0.0007118 |     TrainAcc :
0.8523490   |     TestAcc: 0.9017857
6     ->    TrainL : 0.0593852      |      TestL : 0.0007193 |     TrainAcc :
0.8590604   |     TestAcc: 0.9017857
7     ->    TrainL : 0.0573400      |      TestL : 0.0007285 |     TrainAcc :
0.8590604   |     TestAcc: 0.9017857
8     ->    TrainL : 0.0558743      |      TestL : 0.0007374 |     TrainAcc :
0.8568233   |     TestAcc: 0.9017857
9     ->    TrainL : 0.0548012      |      TestL : 0.0007453 |     TrainAcc :
0.8568233   |     TestAcc: 0.9017857
10    ->    TrainL : 0.0539826      |      TestL : 0.0007520 |     TrainAcc :
0.8568233   |     TestAcc: 0.9017857
11    ->    TrainL : 0.0533645      |      TestL : 0.0007577 |     TrainAcc :
0.8568233   |     TestAcc: 0.9017857
12    ->    TrainL : 0.0528877      |      TestL : 0.0007624 |     TrainAcc :
0.8568233   |     TestAcc: 0.9017857
13    ->    TrainL : 0.0525195      |      TestL : 0.0007663 |     TrainAcc :
0.8568233   |     TestAcc: 0.9017857
14    ->    TrainL : 0.0522329      |      TestL : 0.0007695 |     TrainAcc :
0.8612975   |     TestAcc: 0.9017857
```

House Autoencoder to DNN

```
# AUTOENCODER THEN TO NETWORK
# Autoencoder Network Architecture 9 Input -> 8 Hidden 1 Inputs -> 9
Output
# DNN Network Architecture 8 Inputs -> 8 H1 Nodes -> 1 Output
for eta in etavalues:
   houseAENN = AE.AENN(dp.houseTuningNP, 5, 500, eta, 0.001, 0.05)
   houseAENN.fitCLASS(dp.houseTuningNP)
```

Sample output:

```
This is the Loss [[0.00000000e+00 4.04501287e-05]
 [4.04501287e-05 0.00000000e+00]
 [4.04501287e-05 0.00000000e+00]
 [4.04501287e-05 0.00000000e+00]
 [0.00000000e+00 4.04501287e-05]
 [0.00000000e+00 4.04501287e-05]
 [4.04501287e-05 0.00000000e+00]
 [4.04501287e-05 0.00000000e+00]
 [0.00000000e+00 4.04501287e-05]
 [4.04501287e-05 0.00000000e+00]
 [0.00000000e+00 4.04501287e-05]
 [0.00000000e+00 4.04501287e-05]
 [0.00000000e+00 4.04501287e-05]
 [4.04501287e-05 0.00000000e+00]
 [4.04501287e-05 0.00000000e+00]
 [0.00000000e+00 4.04501287e-05]
 [4.04501287e-05 0.00000000e+00]]
0.013614828377254926
This is the gradient
[[-0.03604817  0.03605442]
 [ 0.0007864  -0.00077811]
 [-0.00320936  0.00321288]
 [ 0.00338483 -0.00337944]
 [-0.04061737  0.04062595]
 [-0.03922609  0.03923073]
 [ 0.00511379 -0.00510379]
 [ 0.00601349 -0.00600478]
```

```
 [ 0.005497   -0.00549029]
 [ 0.00200516 -0.00199605]
 [-0.00213178  0.00213884]
 [-0.00351111  0.0035126 ]
 [ 0.00236527 -0.00235594]
 [-0.03469114  0.03469505]
 [ 0.00033572 -0.00032674]
 [ 0.00215066 -0.002145  ]]
499   ->    TrainL : 0.0143858      |     TestL : 0.0136148 |     TrainAcc :
1.0000000   |     TestAcc: 0.9411765

Process finished with exit code 0
```

House Testing on 80% Data
Best performance on eta = 0.03 and 0.035 both at 0.97701 % accuracy and 0.91% precision, Chose eta = 0.03

```
houseAENNTest= AE.AENN(dp.houseNP80, 5, 500, 0.03, 0.001, 0.05)
houseAENNTest.fitCLASS(dp.houseNP80)
```

Sample Output:

```
This is how many Features X has
 16
This is the Autoencoding Layers
 [16, 15, 16]
This is the DNN Layers
 [15, 16, 15, 1]
Initializing Traditional DNN
```

```
this is the Probabilites [[1.37308846e-02 9.86269115e-01]
 [1.67989276e-02 9.83201072e-01]
 [9.98997626e-01 1.00237393e-03]
 [9.93179764e-01 6.82023608e-03]
 [9.99806563e-01 1.93437012e-04]
 [9.98054571e-01 1.94542854e-03]
 [3.01365409e-02 9.69863459e-01]
 [1.02374714e-01 8.97625286e-01]
```

```
 [9.99981761e-01 1.82391191e-05]
 [9.74548796e-01 2.54512041e-02]
 [9.03155079e-01 9.68449211e-02]
 [1.73084165e-01 8.26915835e-01]
 [9.80035847e-01 1.99641532e-02]
 [4.49889870e-01 5.50110130e-01]
 [9.99338858e-01 6.61141584e-04]
 [9.81395125e-01 1.86048751e-02]
 [6.02138834e-03 9.93978612e-01]
 [7.58797492e-01 2.41202508e-01]
 [3.02835176e-04 9.99697165e-01]
 [1.37308846e-02 9.86269115e-01]
 [2.04032849e-02 9.79596715e-01]
 [9.99715271e-01 2.84728593e-04]
 [9.99152559e-01 8.47440695e-04]
 [9.98999280e-01 1.00072047e-03]
 [9.99818779e-01 1.81220500e-04]
 [9.96379801e-01 3.62019854e-03]
 [9.99952881e-01 4.71187057e-05]
 [9.99008584e-01 9.91416219e-04]
 [6.02138834e-03 9.93978612e-01]
 [9.96424128e-01 3.57587214e-03]
 [9.99820466e-01 1.79534224e-04]
 [9.91719060e-01 8.28093963e-03]
 [9.93722685e-01 6.27731524e-03]
 [9.99152559e-01 8.47440695e-04]
 [9.96379801e-01 3.62019854e-03]
 [9.91191998e-01 8.80800155e-03]
 [9.96379801e-01 3.62019854e-03]
 [9.95913941e-01 4.08605889e-03]
 [2.92183568e-01 7.07816432e-01]
 [9.12617870e-01 8.73821304e-02]
 [6.25992455e-01 3.74007545e-01]
 [2.00791550e-03 9.97992084e-01]
 [2.10865469e-01 7.89134531e-01]
 [1.10229632e-02 9.88977037e-01]
 [3.93622285e-03 9.96063777e-01]
 [9.99459629e-01 5.40370642e-04]
 [9.99353038e-01 6.46961847e-04]
 [9.98736174e-01 1.26382552e-03]
 [3.77829402e-02 9.62217060e-01]
 [3.56394297e-01 6.43605703e-01]
 [9.99834742e-01 1.65258333e-04]
```

```
 [8.70456654e-01 1.29543346e-01]
 [9.84740110e-01 1.52598900e-02]
 [6.02138834e-03 9.93978612e-01]
 [9.74548796e-01 2.54512041e-02]
 [9.99278016e-01 7.21984482e-04]
 [9.99935747e-01 6.42531502e-05]
 [2.58696522e-02 9.74130348e-01]
 [9.99910949e-01 8.90510842e-05]
 [9.97823534e-01 2.17646629e-03]
 [2.81174235e-01 7.18825765e-01]
 [9.84740110e-01 1.52598900e-02]
 [9.98997626e-01 1.00237393e-03]
 [9.98969734e-01 1.03026573e-03]
 [6.02138834e-03 9.93978612e-01]
 [9.77339525e-01 2.26604754e-02]
 [9.99353038e-01 6.46961847e-04]
 [9.27728650e-01 7.22713500e-02]
 [4.31634058e-02 9.56836594e-01]]
this is y Matrix [[0. 1.]
 [0. 1.]
 [1. 0.]
 [1. 0.]
 [1. 0.]
 [1. 0.]
 [0. 1.]
 [1. 0.]
 [1. 0.]
 [1. 0.]
 [1. 0.]
 [0. 1.]
 [1. 0.]
 [1. 0.]
 [1. 0.]
 [1. 0.]
 [0. 1.]
 [1. 0.]
 [0. 1.]
 [0. 1.]
 [0. 1.]
 [1. 0.]
 [1. 0.]
 [1. 0.]
 [1. 0.]
```

```
 [1. 0.]
 [1. 0.]
 [1. 0.]
 [0. 1.]
 [1. 0.]
 [1. 0.]
 [1. 0.]
 [1. 0.]
 [1. 0.]
 [1. 0.]
 [1. 0.]
 [1. 0.]
 [1. 0.]
 [0. 1.]
 [1. 0.]
 [0. 1.]
 [0. 1.]
 [0. 1.]
 [0. 1.]
 [0. 1.]
 [1. 0.]
 [1. 0.]
 [1. 0.]
 [0. 1.]
 [0. 1.]
 [1. 0.]
 [1. 0.]
 [1. 0.]
 [0. 1.]
 [1. 0.]
 [1. 0.]
 [1. 0.]
 [0. 1.]
 [1. 0.]
 [1. 0.]
 [0. 1.]
 [1. 0.]
 [1. 0.]
 [1. 0.]
 [0. 1.]
 [1. 0.]
 [1. 0.]
 [1. 0.]
```

```
 [1. 0.]]
This is the Loss [[0.00000000e+00 1.82392854e-05]
 [0.00000000e+00 1.82392854e-05]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]
 [0.00000000e+00 1.82392854e-05]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]
 [0.00000000e+00 1.82392854e-05]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]
 [0.00000000e+00 1.82392854e-05]
 [1.82392854e-05 0.00000000e+00]
 [0.00000000e+00 1.82392854e-05]
 [0.00000000e+00 1.82392854e-05]
 [0.00000000e+00 1.82392854e-05]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]
 [0.00000000e+00 1.82392854e-05]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]
 [0.00000000e+00 1.82392854e-05]
 [1.82392854e-05 0.00000000e+00]
 [0.00000000e+00 1.82392854e-05]
 [0.00000000e+00 1.82392854e-05]
```

```
 [0.00000000e+00 1.82392854e-05]
 [0.00000000e+00 1.82392854e-05]
 [0.00000000e+00 1.82392854e-05]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]
 [0.00000000e+00 1.82392854e-05]
 [0.00000000e+00 1.82392854e-05]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]
 [0.00000000e+00 1.82392854e-05]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]
 [0.00000000e+00 1.82392854e-05]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]
 [0.00000000e+00 1.82392854e-05]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]
 [0.00000000e+00 1.82392854e-05]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]
 [1.82392854e-05 0.00000000e+00]]
0.020373897192852278
This is the gradient
[[-0.01333684  0.01333807]
 [-0.02206623  0.02206791]
 [ 0.01302711 -0.01302494]
 [-0.00828176  0.00828406]
 [-0.00648344  0.00648548]
 [-0.00767645  0.00767992]
 [-0.0219426   0.02194579]
 [-0.01032332  0.01032719]
 [-0.01295919  0.01296367]
 [ 0.00665526 -0.00665401]
 [-0.02355043  0.0235506 ]
 [-0.01172713  0.01172894]
 [ 0.00632935 -0.00632566]
 [-0.00908552  0.00908935]
```

```
 [ 0.01071391 -0.01071283]
 [-0.01598364  0.01598606]]
499   ->    TrainL : 0.0213077      |     TestL : 0.0203739 |     TrainAcc :
0.9856631   |     TestAcc: 0.9420290

Process finished with exit code 0
```

Car Tuning Autoencoder

```
# AUTOENCODER THEN TO NETWORK
# Autoencoder Network Architecture 6 Input -> 5 Hidden 1 Inputs -> 6
Output
# DNN Network Architecture 5 Inputs -> 4 H1 Nodes -> 1 Output
for eta in etavalues:
   carAENN = AE.AENN(dp.carTuningNP, 5, 500, eta, 0.001, 0.05)
   carAENN.fitCLASS(dp.carTuningNP)
```

Tuning Sample Output:

```
this is y Matrix [[1. 0. 0. 0.]
 [0. 1. 0. 0.]
 [1. 0. 0. 0.]]
This is the Loss [[0.35626203 0.         0.         0.        ]
 [0.         0.35626203 0.         0.        ]
 [0.35626203 0.         0.         0.        ]]
0.3586344693896866
This is the gradient
[[ 0.3259769  -0.28671304 -0.00052893 -0.03873089]
 [-0.0694733   0.11054233 -0.02220165 -0.01886404]
 [ 0.06944399 -0.11069592  0.0225911   0.01866427]
 [-0.34732531  0.17541456  0.04982027  0.1220959 ]
 [ 0.23832175 -0.28715478  0.02464445  0.02419228]
 [ 0.33714732 -0.35326519  0.00525173  0.0108695 ]]
This is the Scores [[ 3.20554904  1.19288056 -1.85573716 -2.52602486]
 [ 0.29134008  0.83074446 -0.5303433  -0.57883044]
 [ 1.77951505  1.809665   -2.23498588 -1.33191565]]
X [[-1  0  3  0  1  2]
 [ 1 -1  1  1  0  1]
 [-1  1  1  1  2  2]]
this is the Probabilites [[0.87472918 0.11689146 0.00554351 0.00283585]
 [0.2798333  0.47990994 0.12304015 0.11721661]
 [0.47773784 0.49236095 0.00862411 0.0212771 ]]
this is y Matrix [[1. 0. 0. 0.]
```

```
 [1. 0. 0. 0.]
 [0. 1. 0. 0.]]
This is the Loss [[0.13384095 0.         0.         0.        ]
 [0.13384095 0.         0.         0.        ]
 [0.         0.13384095 0.         0.        ]]
0.13621727369174633
This is the gradient
[[-0.35820829  0.2901053   0.03677222  0.0313348 ]
 [ 0.39859126 -0.3293407  -0.03766624 -0.03158098]
 [-0.20539983  0.10765291  0.04934858  0.04840177]
 [-0.08105168 -0.00881546  0.04361978  0.04625278]
 [ 0.27762246 -0.29910635  0.00684783  0.01463976]
 [-0.0052713  -0.10017356  0.0502705   0.05517771]]
This is the Scores [[ 2.12651232  1.47534395 -1.95733385 -1.62626998]
 [ 1.64924839  2.33054146 -2.49717808 -1.45351045]
 [-0.93790409  1.5503202  -0.92559134  0.33004043]]
X [[ 1 -1  1  0  2  2]
 [ 1 -1  1  2  2  2]
 [-1  1  0  2  0  2]]
this is the Probabilites [[0.64031507 0.33388345 0.01078455 0.01501693]
 [0.32925305 0.65074617 0.00520906 0.01479172]
 [0.05679944 0.68385831 0.05750312 0.20183913]]
this is y Matrix [[1. 0. 0. 0.]
 [0. 1. 0. 0.]
 [1. 0. 0. 0.]]
This is the Loss [[0.38000453 0.         0.         0.        ]
 [0.         0.38000453 0.         0.        ]
 [0.38000453 0.         0.         0.        ]]
0.38238531531518627
This is the gradient
[[ 0.30360448 -0.23319976 -0.01335641 -0.05704427]
 [-0.30498034  0.23292988  0.0143103   0.05774349]
 [-0.00945627 -0.00512256  0.00524642  0.00933584]
 [-0.40953698  0.22349725  0.0415382   0.14450695]
 [-0.01941055 -0.00988008  0.00991272  0.01938162]
 [-0.64928243  0.44601759  0.04880809  0.1544601 ]]
This is the Scores [[ 3.00248198  2.37980356 -3.07667184 -2.2778229 ]]
X [[-1  0  2  2  2  2]]
this is the Probabilites [[0.647716   0.34750321 0.00148334 0.00329744]]
this is y Matrix [[0. 1. 0. 0.]]
This is the Loss [[0.         0.43430295 0.         0.        ]]
0.4366689447681718
This is the gradient
```

```
[[-6.48377777e-01  6.52380943e-01 -1.00285119e-03 -2.99627638e-03]
 [-7.14129447e-04 -1.53931235e-04  4.73366464e-04  3.98032747e-04]
 [ 1.29612038e+00 -1.30499268e+00  2.88163158e-03  5.99409903e-03]
 [ 1.29520788e+00 -1.30457380e+00  2.69522986e-03  6.67611712e-03]
 [ 1.29630955e+00 -1.30462584e+00  2.21663394e-03  6.10336741e-03]
 [ 1.29526057e+00 -1.30465001e+00  2.77516087e-03  6.61763595e-03]]
This is the Scores [[ 1.23309583  1.60380159 -1.88372175 -0.93905219]
 [ 1.73838147  0.84289172 -1.49307739 -1.08008819]
 [-1.78445142  0.48431851  0.57045031  0.74377436]
 [-0.05349703  1.19681879 -0.34484451 -0.7754437 ]
 [ 2.93398313  1.7292834  -2.41761811 -2.2316165 ]
 [ 1.86871007  1.51998461 -1.39110859 -1.97920034]
 [ 0.50994883  2.07530623 -1.36662047 -1.18647295]
 [ 1.16253177  1.49607097 -1.49556291 -1.14214987]
 [ 3.76616262  2.14272456 -3.16776635 -2.72338488]
 [ 1.99622958  0.33917522 -1.039045   -1.30030808]
 [ 0.03155156  0.96065762 -0.63564396 -0.34298861]
 [ 2.50271997  2.50620268 -2.80104339 -2.18713193]
 [ 0.48730883  1.6186792  -1.5630046  -0.52068933]
 [-0.03949347  1.47285728 -1.20829635 -0.21420008]
 [ 1.97319461  1.00278034 -1.50035854 -1.46846499]
 [ 1.85510148  0.12371397 -0.26272733 -1.70650342]
 [ 2.07531023  2.07116858 -2.52416133 -1.60835593]
 [ 1.7541046   1.4829612  -2.08744909 -1.14657361]
 [ 1.33361173  0.86448465 -1.01981121 -1.16709581]
 [ 1.50677238  1.42067324 -1.47096161 -1.43577527]
 [ 1.61297649  1.26749995 -1.31113142 -1.55276895]
 [ 0.37084925  0.60334196 -0.75217506 -0.22063006]
 [ 1.14447356  0.49761795 -0.35483239 -1.27634572]
 [ 2.6477875   2.01001241 -2.22117192 -2.41247188]
 [ 2.38142279  0.88946618 -1.57824681 -1.68110648]
 [ 1.25004761  1.47428497 -0.98325727 -1.71588575]
 [ 1.33521144  2.67218984 -2.90752454 -1.07894313]
 [ 2.05227526  2.7347737  -2.98547488 -1.77651284]
 [ 2.73135161  1.40021209 -2.25772586 -1.86368194]
 [ 1.96810261  1.23077243 -1.46893479 -1.70691359]
 [ 2.2469864   2.25371801 -2.72106623 -1.76070054]
 [ 2.79794702  1.28583005 -1.6727242  -2.39936324]
 [ 4.462306    2.11271237 -3.17302067 -3.3829    ]
 [ 1.68699898  1.28350935 -1.30391234 -1.64932038]
 [ 0.43086818  0.34331287  0.11849586 -0.8784251 ]
 [ 2.96059177  2.4080231  -3.19439448 -2.15862115]
 [ 2.3475023   1.51440107 -1.85715569 -1.98874415]
```

```
 [ 2.02466311  1.06246456 -0.99364178 -2.06505954]
 [ 4.24667442  2.50117201 -3.36473331 -3.36065772]
 [ 3.6036331   2.45459755 -3.27956389 -2.75963943]
 [ 0.39137471  1.81617015 -1.08980049 -1.09262519]
 [ 2.80191566  1.50794272 -2.64588469 -1.66058427]
 [ 2.47164939  0.92561477 -1.2045882  -2.17602952]
 [ 0.76161544  0.30571053 -0.3619894  -0.69486605]
 [ 3.84575808  1.31081158 -2.2311598  -2.9133739 ]
 [ 1.27156426  2.38101677 -1.72860988 -1.88133901]
 [ 2.12841229  1.99458194 -2.44424624 -1.66685277]
 [ 2.87641913  0.90402183 -1.67785439 -2.0890219 ]
 [ 0.38791628  1.90789138 -1.4851784  -0.79297608]
 [ 2.31358181  2.13933597 -2.13606457 -2.29638183]
 [ 1.13246469  0.9090525  -1.11416446 -0.91549611]
 [ 1.54578487  0.56774624 -1.22347649 -0.88968899]
 [ 0.65319147  1.53456632 -1.60892859 -0.5740661 ]
 [ 2.56659231  1.03422021 -1.27006514 -2.31063554]
 [ 1.1834522   1.58866702 -1.56825893 -1.18020446]
 [ 1.43918578  1.84115168 -1.64823609 -1.60663585]
 [ 0.79614446  1.79457722 -1.56306667 -1.00561757]
 [ 0.06507709  1.45595487 -0.62166449 -0.86929146]
 [ 0.82263787  1.03925088 -0.70637521 -1.13710975]
 [ 2.46607641  1.77353715 -2.13397568 -2.08259192]
 [ 0.38445784  1.9996126  -1.88055632 -0.49332698]
 [-1.22100556  1.36280595 -0.45132566  0.3327451 ]
 [ 1.25401626  1.69639764 -1.95641776 -0.97710679]
 [ 2.35096074  1.42267985 -1.46177777 -2.28839326]
 [ 0.00851659  1.62426274 -1.0969575  -0.51114552]
 [ 2.20291576  1.39066105 -1.47621594 -2.0952904 ]
 [-0.19471114  0.48105568 -0.26440865 -0.01581231]
 [ 1.20302874  1.01678312 -1.5023233  -0.71239844]
 [ 1.02477609  0.68858664 -1.01974914 -0.68216758]]
X [[ 0  0  0  0  2  2]
 [ 1 -1  0  0  2  0]
 [ 1  1  0  0  0  2]
 [ 1  0  2  1  0  2]
 [-1 -1  2  2  1  0]
 [-1  0  3  1  0  2]
 [ 1  0  2  2  1  2]
 [ 0  1  1  0  2  2]
 [-1 -1  2  2  2  0]
 [-1 -1  1  0  0  0]
 [-1  1  1  2  0  0]
```

```
[-1 -1  2  2  1  2]
[ 1  0  0  2  2  0]
[ 0 -1  0  2  0  1]
[ 0 -1  1  0  1  1]
[-1  1  3  0  0  0]
[-1 -1  1  2  1  1]
[-1 -1  0  0  1  2]
[ 1 -1  1  0  1  1]
[-1  1  2  2  1  0]
[-1  1  2  0  1  2]
[-1  0  0  1  0  0]
[ 1 -1  2  0  0  1]
[-1  0  3  2  1  1]
[ 1 -1  1  0  2  0]
[ 1 -1  3  2  0  1]
[-1  0  0  2  2  2]
[ 0 -1  1  2  2  2]
[-1  0  1  0  2  1]
[ 0  1  2  1  2  0]
[-1  0  1  1  2  2]
[-1 -1  3  1  0  1]
[-1 -1  3  1  2  1]
[ 0  0  2  0  1  2]
[ 1  0  2  0  0  1]
[-1 -1  1  1  2  2]
[ 0 -1  2  1  1  1]
[ 1  1  3  0  2  1]
[-1 -1  3  1  2  2]
[-1 -1  2  1  2  2]
[ 1 -1  2  2  0  2]
[-1 -1  0  0  2  1]
[-1  1  3  0  1  1]
[ 0  1  1  0  1  0]
[-1 -1  3  1  1  0]
[ 1  1  3  2  2  2]
[-1 -1  1  1  1  2]
[-1  1  2  0  2  0]
[ 0 -1  1  2  0  2]
[-1 -1  3  2  0  2]
[-1  1  1  1  1  0]
[ 0 -1  0  0  1  0]
[-1 -1  0  1  0  2]
[ 1 -1  3  1  1  0]
```

```
 [ 1  0  1  1  2  1]
 [ 1 -1  2  2  1  1]
 [ 1 -1  1  2  1  1]
 [ 1  1  2  1  1  2]
 [ 1 -1  2  2  0  0]
 [ 0  0  2  1  2  1]
 [-1 -1  0  2  0  2]
 [ 1  1  0  1  1  2]
 [ 1 -1  0  1  2  1]
 [ 1 -1  3  1  1  1]
 [ 0  1  1  2  1  1]
 [-1  1  3  1  1  1]
 [ 1 -1  0  1  0  0]
 [-1  0  0  1  1  0]
 [ 1  0  0  0  2  0]]
this is the Probabilites [[3.83584533e-01 5.55720678e-01 1.69919967e-02
4.37027925e-02]
 [6.63303895e-01 2.70898307e-01 2.62003924e-02 3.95974051e-02]
 [2.96420589e-02 2.86564657e-01 3.12341140e-01 3.71452144e-01]
 [1.74687530e-01 6.09911982e-01 1.30536301e-01 8.48641861e-02]
 [7.63223499e-01 2.28800669e-01 3.61810232e-03 4.35773002e-03]
 [5.66473891e-01 3.99696510e-01 2.17499692e-02 1.20796302e-02]
 [1.63376336e-01 7.81655226e-01 2.50152853e-02 2.99531523e-02]
 [3.89746252e-01 5.44046700e-01 2.73140496e-02 3.88929988e-02]
 [8.33532768e-01 1.64388900e-01 8.11999176e-04 1.26633325e-03]
 [7.83838848e-01 1.49477979e-01 3.76724427e-02 2.90107307e-02]
 [2.11282684e-01 5.35019411e-01 1.08418794e-01 1.45279112e-01]
 [4.95626182e-01 4.97355310e-01 2.46467170e-03 4.55383654e-03]
 [2.17696789e-01 6.74838632e-01 2.80163926e-02 7.94481861e-02]
 [1.49525408e-01 6.78454275e-01 4.64633324e-02 1.25556985e-01]
 [6.93508110e-01 2.62788272e-01 2.15033725e-02 2.22002451e-02]
 [7.54304624e-01 1.33541032e-01 9.07372066e-02 2.14171370e-02]
 [4.92355021e-01 4.90320078e-01 4.95168777e-03 1.23732129e-02]
 [5.43787601e-01 4.14641931e-01 1.16698086e-02 2.99006592e-02]
 [5.54748610e-01 3.47021933e-01 5.27251262e-02 4.55043314e-02]
 [4.94769655e-01 4.53952775e-01 2.51877640e-02 2.60898060e-02]
 [5.54391884e-01 3.92444555e-01 2.97778222e-02 2.33857384e-02]
 [3.18416210e-01 4.01758867e-01 1.03579046e-01 1.76245878e-01]
 [5.44714600e-01 2.85261522e-01 1.21626642e-01 4.83972364e-02]
 [6.48301504e-01 3.42605894e-01 4.97983350e-03 4.11276886e-03]
 [7.92891267e-01 1.78346713e-01 1.51199716e-02 1.36420486e-02]
 [4.14928458e-01 5.19228199e-01 4.44693965e-02 2.13739470e-02]
 [2.03610733e-01 7.75252202e-01 2.92552519e-03 1.82115394e-02]
```

```
 [3.32552966e-01 6.58060969e-01 2.15771332e-03 7.22835180e-03]
 [7.80585353e-01 2.06211961e-01 5.31730514e-03 7.88538099e-03]
 [6.51084801e-01 3.11472084e-01 2.09387706e-02 1.65043451e-02]
 [4.92153500e-01 4.95477664e-01 3.42375545e-03 8.94508057e-03]
 [8.08137054e-01 1.78148006e-01 9.24479275e-03 4.47014750e-03]
 [9.12173532e-01 8.70285846e-02 4.40653502e-04 3.57229892e-04]
 [5.70192326e-01 3.80879893e-01 2.86473916e-02 2.02803889e-02]
 [3.42713628e-01 3.13983326e-01 2.50766662e-01 9.25363841e-02]
 [6.31483236e-01 3.63399477e-01 1.34055693e-03 3.77672968e-03]
 [6.83662266e-01 2.97187368e-01 1.02042678e-02 8.94609906e-03]
 [6.90760440e-01 2.63906410e-01 3.37671417e-02 1.15660078e-02]
 [8.50666487e-01 1.48490018e-01 4.20888390e-04 4.22607261e-04]
 [7.57753211e-01 2.40164045e-01 7.76592718e-04 1.30615176e-03]
 [1.78217962e-01 7.40853386e-01 4.05214760e-02 4.04071765e-02]
 [7.75182635e-01 2.12539046e-01 3.33775459e-03 8.94056445e-03]
 [8.01286442e-01 1.70747160e-01 2.02870123e-02 7.67938588e-03]
 [4.56197162e-01 2.89171572e-01 1.48312309e-01 1.06318957e-01]
 [9.23598599e-01 7.32099288e-02 2.11988041e-03 1.07159219e-03]
 [2.42412263e-01 7.35163076e-01 1.20668944e-02 1.03577660e-02]
 [5.24233792e-01 4.58567456e-01 5.41557590e-03 1.17831754e-02]
 [8.64588233e-01 1.20284054e-01 9.09731977e-03 6.03039389e-03]
 [1.65761513e-01 7.57880098e-01 2.54688315e-02 5.08895567e-02]
 [5.37136193e-01 4.51242932e-01 6.27519968e-03 5.34567616e-03]
 [4.91512098e-01 3.93104062e-01 5.19799154e-02 6.34039243e-02]
 [6.55173825e-01 2.46376772e-01 4.10848413e-02 5.73645619e-02]
 [2.62367935e-01 6.33413025e-01 2.73202392e-02 7.68988002e-02]
 [8.03079779e-01 1.73483407e-01 1.73188691e-02 6.11794469e-03]
 [3.76290853e-01 5.64295020e-01 2.40143425e-02 3.53997843e-02]
 [3.86408171e-01 5.77587615e-01 1.76277150e-02 1.83764990e-02]
 [2.51665548e-01 6.83026582e-01 2.37810700e-02 4.15267996e-02]
 [1.69078210e-01 6.79419780e-01 8.50823670e-02 6.64196428e-02]
 [3.84688346e-01 4.77730386e-01 8.33809974e-02 5.42002703e-02]
 [6.57489580e-01 3.28944702e-01 6.60863302e-03 6.95708511e-03]
 [1.52713783e-01 7.67946625e-01 1.58560529e-02 6.34835393e-02]
 [4.73131238e-02 6.26782274e-01 1.02152584e-01 2.23752019e-01]
 [3.69801143e-01 5.75561907e-01 1.49174459e-02 3.97195046e-02]
 [7.00778644e-01 2.76970549e-01 1.54785246e-02 6.77228177e-03]
 [1.43731519e-01 7.23205383e-01 4.75828791e-02 8.54802190e-02]
 [6.74448008e-01 2.99357911e-01 1.70263629e-02 9.16771835e-03]
 [1.96304941e-01 3.85845140e-01 1.83088888e-01 2.34761031e-01]
 [4.89190030e-01 4.06061817e-01 3.27007817e-02 7.20473717e-02]
 [4.93742119e-01 3.52772686e-01 6.39107845e-02 8.95744105e-02]]
this is y Matrix [[1. 0. 0. 0.]
```

```
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [0. 0. 1. 0.]
 [1. 0. 0. 0.]
 [0. 1. 0. 0.]
 [0. 0. 0. 1.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [0. 1. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [0. 1. 0. 0.]
 [0. 0. 0. 1.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [0. 1. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [0. 1. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [0. 1. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
```

```
 [1. 0. 0. 0.]
 [0. 0. 0. 1.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [0. 1. 0. 0.]
 [0. 1. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [0. 0. 1. 0.]
 [0. 1. 0. 0.]
 [0. 1. 0. 0.]
 [0. 0. 0. 1.]
 [1. 0. 0. 0.]
 [0. 0. 1. 0.]
 [1. 0. 0. 0.]
 [0. 1. 0. 0.]
 [0. 1. 0. 0.]
 [0. 1. 0. 0.]
 [0. 0. 1. 0.]
 [0. 1. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]]
This is the Loss [[0.07947772 0.         0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.         0.         0.07947772 0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.         0.07947772 0.         0.        ]
 [0.         0.         0.         0.07947772]
 [0.07947772 0.         0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.07947772 0.         0.         0.        ]
```

```
 [0.07947772 0.         0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.         0.07947772 0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.         0.07947772 0.         0.        ]
 [0.         0.         0.         0.07947772]
 [0.07947772 0.         0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.         0.07947772 0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.         0.07947772 0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.         0.07947772 0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.         0.         0.         0.07947772]
 [0.07947772 0.         0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.         0.07947772 0.         0.        ]
 [0.         0.07947772 0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.07947772 0.         0.         0.        ]
 [0.         0.         0.07947772 0.        ]
 [0.         0.07947772 0.         0.        ]
 [0.         0.07947772 0.         0.        ]
 [0.         0.         0.         0.07947772]
 [0.07947772 0.         0.         0.        ]
 [0.         0.         0.07947772 0.        ]
 [0.07947772 0.         0.         0.        ]
```

```
 [0.         0.07947772 0.         0.         ]
 [0.         0.07947772 0.         0.         ]
 [0.         0.07947772 0.         0.         ]
 [0.         0.         0.07947772 0.         ]
 [0.         0.07947772 0.         0.         ]
 [0.07947772 0.         0.         0.         ]
 [0.07947772 0.         0.         0.         ]
 [0.07947772 0.         0.         0.         ]]
0.08183819866337681
This is the gradient
[[ 0.03988389  0.00415821 -0.01309915 -0.03093891]
 [ 0.03837369 -0.01455307 -0.01104559 -0.0127717 ]
 [-0.05891646  0.18298042 -0.03949054 -0.08456999]
 [-0.0979613   0.20099967 -0.04328665 -0.05974629]
 [-0.11653288  0.22680842 -0.05164102 -0.05863082]
 [-0.10614138  0.21016299 -0.03392291 -0.07009534]]
499   ->    TrainL : 0.2521933      |     TestL : 0.0818382 |     TrainAcc :
0.6389892   |     TestAcc: 0.6086957

Process finished with exit code 0
```

Car Testing:

```
# Best Accuracy at 0.6560 with eta = 0.015

carAENNTest= AE.AENN(dp.carNP80, 5, 500, 0.03, 0.001, 0.05)
carAENNTest.fitCLASS(dp.carNP80)
```

Sample output:

```
This is how many Features X has
 6
This is the Autoencoding Layers
 [6, 5, 6]
This is the DNN Layers
 [5, 6, 5, 1]
Initializing Traditional DNN

Best Weights After Training Autoencoder
 [[nan nan nan nan nan nan]
 [nan nan nan nan nan nan]
 [nan nan nan nan nan nan]
 [nan nan nan nan nan nan]
 [nan nan nan nan nan nan]]
X Originally
```

```
 [[-1  0  0  0  2  0]
 [ 0  1  3  1  1  2]
 [ 0 -1  2  2  2  2]
 ...
 [-1 -1  3  2  2  2]
 [-1  1  1  2  2  0]
 [-1  1  0  1  2  2]]
X Encoded from Trained Autoencoders
 [[nan nan nan nan nan]
 [nan nan nan nan nan]
 [nan nan nan nan nan]
 ...
 [nan nan nan nan nan]
 [nan nan nan nan nan]
 [nan nan nan nan nan]]
Now attached Attaching To Other Network

Dataframe
        0   1   2   3   4  Y
C:\Users\wsven\AppData\Local\Programs\Python\Python38\lib\site-packages\numpy
\lib\function_base.py:4454: VisibleDeprecationWarning: Creating an ndarray
from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or
ndarrays with different lengths or shapes) is deprecated. If you meant to do
this, you must specify 'dtype=object' when creating the ndarray.
  arr = asarray(arr)
0     NaN NaN NaN NaN NaN  0
1     NaN NaN NaN NaN NaN  3
2     NaN NaN NaN NaN NaN  1
3     NaN NaN NaN NaN NaN  0
4     NaN NaN NaN NaN NaN  0
...    ..  ..  ..  ..  .. ..
1377 NaN NaN NaN NaN NaN  0
1378 NaN NaN NaN NaN NaN  1
1379 NaN NaN NaN NaN NaN  0
1380 NaN NaN NaN NaN NaN  0
1381 NaN NaN NaN NaN NaN  1

[1382 rows x 6 columns]
Numpy
 [[nan nan nan nan nan  0.]
 [nan nan nan nan nan  3.]
 [nan nan nan nan nan  1.]
 ...
```

```
 [nan nan nan nan nan  0.]
 [nan nan nan nan nan  0.]
 [nan nan nan nan nan  1.]]
Fold 1
Testing set size : (277, 7)
Training set size : (1105, 7)

0     ->    TrainL : 0.4790501     |     TestL : 0.0797067 |     TrainAcc :
0.6497738   |     TestAcc: 0.7292419
1     ->    TrainL : 0.3296098     |     TestL : 0.0455600 |     TrainAcc :
0.6343891   |     TestAcc: 0.7075812
2     ->    TrainL : 0.3027349     |     TestL : 0.0369344 |     TrainAcc :
0.6352941   |     TestAcc: 0.7075812
3     ->    TrainL : 0.2926254     |     TestL : 0.0337316 |     TrainAcc :
0.6334842   |     TestAcc: 0.7039711
4     ->    TrainL : 0.2880036     |     TestL : 0.0323070 |     TrainAcc :
0.6334842   |     TestAcc: 0.7039711
5     ->    TrainL : 0.2855690     |     TestL : 0.0315829 |     TrainAcc :
0.6334842   |     TestAcc: 0.7039711
6     ->    TrainL : 0.2841465     |     TestL : 0.0311711 |     TrainAcc :
0.6334842   |     TestAcc: 0.7039711
7     ->    TrainL : 0.2832403     |     TestL : 0.0309148 |     TrainAcc :
0.6334842   |     TestAcc: 0.7039711
8     ->    TrainL : 0.2826299     |     TestL : 0.0307449 |     TrainAcc :
0.6343891   |     TestAcc: 0.7039711
9     ->    TrainL : 0.2822040     |     TestL : 0.0306272 |     TrainAcc :
0.6361991   |     TestAcc: 0.7039711
10    ->    TrainL : 0.2818999     |     TestL : 0.0305436 |     TrainAcc :
0.6361991   |     TestAcc: 0.7039711
11    ->    TrainL : 0.2816797     |     TestL : 0.0304832 |     TrainAcc :
0.6361991   |     TestAcc: 0.7039711
12    ->    TrainL : 0.2815188     |     TestL : 0.0304391 |     TrainAcc :
0.6361991   |     TestAcc: 0.7039711
13    ->    TrainL : 0.2814005     |     TestL : 0.0304068 |     TrainAcc :
0.6361991   |     TestAcc: 0.7039711
14    ->    TrainL : 0.2813131     |     TestL : 0.0303829 |     TrainAcc :
0.6361991   |     TestAcc: 0.7039711
15    ->    TrainL : 0.2812483     |     TestL : 0.0303653 |     TrainAcc :
0.6361991   |     TestAcc: 0.7039711
16    ->    TrainL : 0.2812002     |     TestL : 0.0303522 |     TrainAcc :
0.6361991   |     TestAcc: 0.7039711
17    ->    TrainL : 0.2811644     |     TestL : 0.0303425 |     TrainAcc :
0.6361991   |     TestAcc: 0.7039711
```

```
18    ->    TrainL : 0.2811377      |     TestL : 0.0303353 |     TrainAcc :
0.6361991    |     TestAcc: 0.7039711
19    ->    TrainL : 0.2811177      |     TestL : 0.0303299 |     TrainAcc :
0.6361991    |     TestAcc: 0.7039711
20    ->    TrainL : 0.2811027      |     TestL : 0.0303259 |     TrainAcc :
0.6361991    |     TestAcc: 0.7039711
21    ->    TrainL : 0.2810914      |     TestL : 0.0303230 |     TrainAcc :
0.6361991    |     TestAcc: 0.7039711
22    ->    TrainL : 0.2810830      |     TestL : 0.0303207 |     TrainAcc :
0.6361991    |     TestAcc: 0.7039711
23    ->    TrainL : 0.2810766      |     TestL : 0.0303191 |     TrainAcc :
0.6361991    |     TestAcc: 0.7039711
```