Experiments

To Do Probability of crossover and mutation

Crossover:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Crossprob(Union,Int,Symdiff,elemcrossover) | 1 | 2 | 3 | 4 | 5 |
| |  |  |  |  | | --- | --- | --- | --- | | 1 | 3 | 3 | 3 | | |  | | --- | | 0.841765 | | 0.774181 | | 0.86561 | | 0.865644 | | 0.833735 | | 0.836187 | |  |  |  |  |
| |  |  |  |  | | --- | --- | --- | --- | | 3 | 1 | 3 | 3 | | |  | | --- | | 0.873479 | | 0.893035 | | 0.815215 | | 0.86627 | | 0.869248 | | 0.863449 | |  |  |  |  |
| |  |  |  |  | | --- | --- | --- | --- | | 3 | 3 | 1 | 3 | | |  | | --- | | 0.832511 | | 0.838162 | | 0.839479 | | 0.886614 | | 0.81824 | | 0.843001 | |  |  |  |  |
| |  |  |  |  | | --- | --- | --- | --- | | 3 | 3 | 3 | 1 | | |  | | --- | | 0.857251 | | 0.822898 | | 0.843115 | | 0.815199 | | 0.84556 | | 0.836805 | |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mutation(Add, Remove) | 1 | 2 | 3 | 4 | 5 |
| |  |  | | --- | --- | | 0.1 | 0.9 | | |  | | --- | | 0.857251 | | 0.822898 | | 0.817963 | | 0.845447 | | 0.855886 | | 0.839889 | |  |  |  |  |
| |  |  | | --- | --- | | 0.4 | 0.6 | | |  | | --- | | 0.845434 | | 0.888146 | | 0.888064 | | 0.860227 | | 0.835217 | | 0.863417 | |  |  |  |  |
| |  |  | | --- | --- | | 0.6 | 0.4 | | |  | | --- | | 0.880696 | | 0.88201 | | 0.894685 | | 0.849242 | | 0.883944 | | 0.878115 | |  |  |  |  |
| |  |  | | --- | --- | | 0.9 | 0.1 | | |  | | --- | | 0.895338 | | 0.89542 | | 0.864754 | | 0.871058 | | 0.853241 | | 0.875962 | |  |  |  |  |

To Do Pc and pm combination

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| pc | 1 | 2 | 3 | 4 | 5 |
| |  |  | | --- | --- | | 0.1 | 0.5 | | |  | | --- | | 0.815624 | | 0.595713 | | 0.713909 | | 0.835045 | | 0.84393 | | 0.760844 | |  |  |  |  |
| |  |  | | --- | --- | | 0.4 | 0.5 | | |  | | --- | | 0.611457 | | 0.831538 | | 0.766624 | | 0.73654 | |  |  |  |  |
| |  |  | | --- | --- | | 0.7 | 0.5 | | |  | | --- | | 0.763236 | | 0.733775 | | 0.828185 | | 0.81496 | | 0.807577 | | 0.789547 | |  |  |  |  |
|  |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| pm | 1 | 2 | 3 | 4 | 5 |
| |  |  | | --- | --- | | 0.7 | 0.3 | | |  | | --- | | 0.86447 | | 0.798206 | | 0.864773 | | 0.831427 | | 0.839719 | |  |  |  |  |
| |  |  | | --- | --- | | 0.7 | 0.5 | | |  | | --- | | 0.763236 | | 0.733775 | | 0.828185 | | 0.81496 | | 0.807577 | | 0.789547 | |  |  |  |  |
| |  |  | | --- | --- | | 0.7 | 0.7 | | |  | | --- | | 0.861716 | | 0.887683 | | 0.88208 | | 0.844837 | | 0.870603 | | 0.869384 | |  |  |  |  |
| |  |  | | --- | --- | | 0.7 | 0.9 | | |  | | --- | | 0.886128 | | 0.882571 | | 0.882158 | | 0.882277 | | 0.84106 | | 0.874839 | |  |  |  |  |

To Do Tweaking generations

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| generations | 1 | 2 | 3 | 4 | 5 |
| 10 | |  | | --- | | 0.876004 | | 0.854726 | | 0.861652 | | 0.864127 | |  |  |  |  |
| 25 | |  | | --- | | 0.870716 | | 0.80717 | | 0.838943 | |  |  |  |  |
| 50 | |  | | --- | | 0.881941 | | 0.884881 | | 0.88737 | | 0.884731 | |  |  |  |  |
| 100 | |  | | --- | | 0.89342 | | 0.896608 | | 0.895014 | |  |  |  | But takes too much time so we took 50 |

To Do Individual size regularization

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Input\*(#x) | 1 | 2 | 3 | 4 | 5 |
| 5 | |  | | --- | | 0.858931 | | 0.844749 | | 0.832786 | | 0.855457 | | 0.8505 | | 0.848485 | |  |  |  |  |
| 10 | |  | | --- | | 0.827133 | | 0.83612 | | 0.844991 | | 0.84157 | | 0.8012 | | 0.830203 | |  |  |  |  |
| 15 | |  | | --- | | 0.811627 | | 0.770623 | | 0.872582 | | 0.813513 | | 0.835836 | | 0.820836 | |  |  |  |  |
| 20 | |  | | --- | | 0.86122 | | 0.843967 | | 0.858 | | 0.811074 | | 0.837099 | | 0.842272 | |  |  |  |  |

To Do Operators which performs the best(log, \*)

funList = ["np.log",

"\*",

"np.exp",

"np.sqrt",

"np.square",

"np.log10"]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | 0 | 25 | 25 | 25 | 25 | | |  | | --- | | 0.859516 | | 0.863652 | | 0.835268 | | 0.866672 | | 0.818173 | | 0.848656 | |  |  |  |  |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | 25 | 0 | 25 | 25 | 25 | | |  | | --- | | 0.88937 | | 0.825451 | | 0.829017 | | 0.825817 | | 0.827518 | | 0.839435 | |  |  |  |  |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | 25 | 25 | 0 | 25 | 25 | | |  | | --- | | 0.832152 | | 0.822365 | | 0.829941 | | 0.862963 | | 0.854042 | | 0.840293 | |  |  |  |  |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | 25 | 25 | 25 | 0 | 25 | | |  | | --- | | 0.834631 | | 0.824561 | | 0.839549 | | 0.868863 | | 0.842336 | | 0.841988 | |  |  |  |  |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | 25 | 25 | 25 | 25 | 0 | | |  | | --- | | 0.855582 | | 0.835026 | | 0.825875 | | 0.849232 | | 0.854764 | | 0.844096 | |  |  |  | 1614.4497702121735 seconds for two runs |
| Log10 = 0 | |  | | --- | | 0.856327 | | 0.856685 | | 0.827338 | | 0.851294 | | 0.809545 | | 0.840238 | |  |  |  |  |

To Do Tweaking population

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PopulationSize | 1 | 2 | 3 | 4 | 5 |
| 25 | |  | | --- | | 0.845426 | | 0.79506 | | 0.84077 | | 0.853631 | | 0.843382 | | 0.835654 | |  |  |  |  |
| 50 | |  | | --- | | 0.763236 | | 0.733775 | | 0.828185 | | 0.81496 | | 0.807577 | | 0.789547 | |  |  |  |  |
| 75 | |  | | --- | | 0.823209 | | 0.819948 | | 0.846153 | | 0.825017 | | 0.830845 | | 0.829034 | |  |  |  |  |
| 100 |  |  |  |  |  |

Neural Networks:

classifier.compile(optimizer = 'rmsprop', loss = 'mean\_squared\_error', metrics = ['accuracy'])

# Fitting the ANN to the Training set

classifier.fit(X\_train, y\_train, batch\_size = 1, epochs = 500)

* + Train Accuracy - 0.9070914442620417
  + Test Accuracy - 0.8472214613289865

Random Forest:

regr = RandomForestRegressor(random\_state=0)

* + Train Accuracy - 0.9671507420359712
  + Test Accuracy - 0.8646091463165809