# NeuralNetsGeneticAlgorithms

March 19, 2019

## 1  Genetic Algorithm evolving Neural Net

This notebook presents the performance of genetic algorithm that aims in evolving neural net's weights.

### 1.1  Dataset

Firstly, we need to prepare dataset.

```
In [1]: from GANN import *

        n_cases = 1000
        height = 6
        width = 6
        placeholders_target = np.zeros((n_cases,height,width))
        placeholders_noise = np.zeros((n_cases,height,width))

        placeholders_target = np.array([create_instance(placeholders_target[n])
                                            for n in range(n_cases)])
        placeholders_noise = np.array([create_random_inst(placeholders_noise[n])
                                            for n in range(n_cases)])

        target = [[1,0] for n in range(n_cases)]
        noise = [[0,1] for n in range(n_cases)]

        dataset = np.vstack([placeholders_target,placeholders_noise])
        target = np.vstack([target,noise])
        dataset = dataset.reshape((n_cases*2,height*width))
```

### 1.2  Setting parameters

I am checking which initial number of population provides best results.

```
In [19]: results = dict()
         for n in range(1,11):
             n = n * 100
             ga = GenAlWeightsNN(n_pop=n,max_gen=50)
```

```
        ga.fit(dataset,target)
        ga.transform()
        results[n] = (ga.best_ind,ga.validate())

    result_dict = pd.DataFrame(index = results.keys())
    result_dict['Accuracy on test'] = [round(n[0]*100,2) for n in results.values()]
    result_dict['Accuracy validate'] = [round(n[1],2) for n in results.values()]
    result_dict
```

Out[19]:       Accuracy on test  Accuracy validate
        100              53.09               53.02
        200              53.25               53.32
        300              55.61               55.71
        400              55.18               55.37
        500              54.84               54.54
        600              54.94               54.55
        700              54.66               54.02
        800              55.83               55.36
        900              55.35               55.37
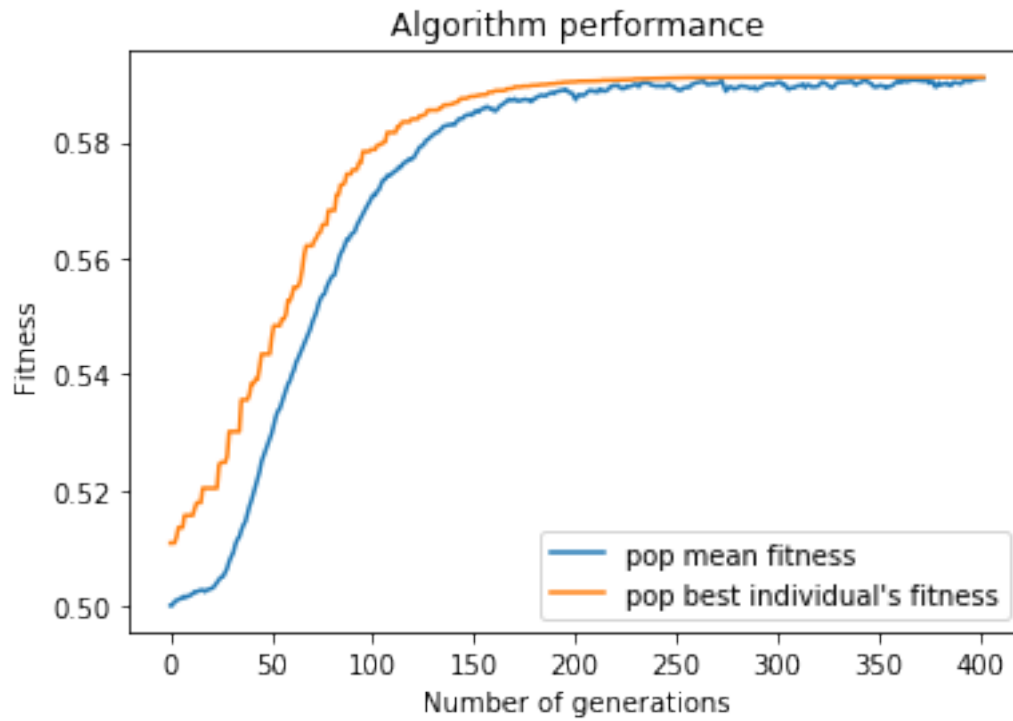        1000             55.42               54.56

## 2 The performance of algorithm

After testing some paramters I have found the fine parameters that allows for

```
In [26]: ga = GenAlWeightsNN(n_pop=800,max_gen=400)
        ga.fit(dataset,target)
        ga.transform()
        ga.plot_fitness()
        print('Achieved accuracy:{}'.format(round(ga.best_ind*100,2)))
        print('Achieved accuracy on validate dataset:{}'.format(
            round(ga.validate(),2)))
```

Algorithm performance

```
Achieved accuracy:59.12
Achieved accuracy on validate dataset:59.14
```

## 2.1 Summary

Even though algorithm performed properly, it didn't allow for as good results as the backpropagation algorithm.

In [ ]: