NeuralNetsGeneticAlgorithms

March 20, 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 [5]: from GANN import *

        n_cases = 1000
        height = 5
        width = 5
        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 [8]: results = dict()
        for n in range(1,11):
            n = n * 100
            ga = GenAlWeightsNN(n_pop=n,max_gen=100)
```

```
        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[8]:        Accuracy on test  Accuracy validate
        100              54.97              54.94
        200              57.53              56.57
        300              57.94              58.04
        400              58.84              58.48
        500              57.37              57.39
        600              58.51              57.70
        700              58.87              58.06
        800              58.85              58.51
        900              58.98              59.14
        1000             59.38              59.01
```
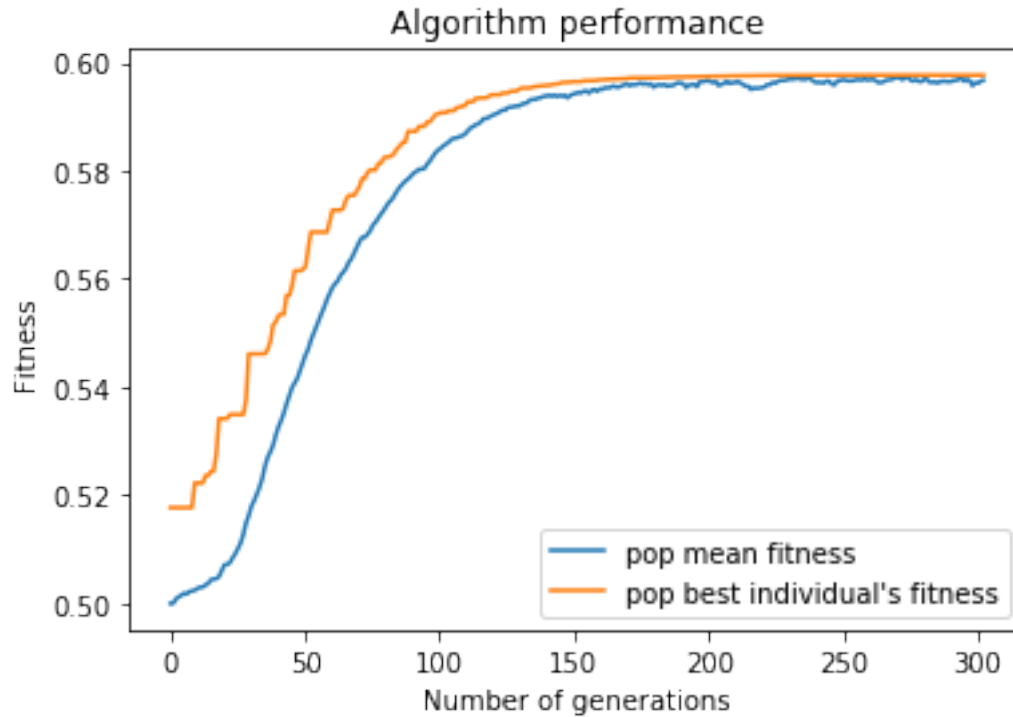
2   The performance of algorithm

After testing some paramters I have found the fine parameters that allows for achieving best results. In purpose of maximazing the search space I used 1000 initial population.

```
In [10]: ga.max_gen =300
         # ga = GenAlWeightsNN(n_pop=1000,max_gen=250,desired_fit=0.7)
         # 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:59.77
Achieved accuracy on validate dataset:60.62
```

2.1 Summary

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

2.1.1 TO DO

Next thing to do is to adapt the algorithm so it evolves the architecture of neural nets.

2.1.2 References

Above implementation used a parts of numpy-based neural net taken from:
    https://www.kaggle.com/niyipop/2-layer-neural-network-from-scratch-using-numpy/data/
    Genetic algorithm originates from:
    https://github.com/stakar/Sig_Feat_Selector