SUID : 387781563

**Evolutionary Machine Learning – HW4**

**HW4:**  Use a Learning Classifier System for learning a collection of rules for the same classification problem. Compare with the previous results (HW1-3).

Link:

https://www.kaggle.com/chaithanya96/bankmarketing

The bank marketing dataset is the csv file with users’ details used to predict the marketing decision. The decision yes or no is represented with binary numbers 1 and 0. The sample columns included are:

1. 'age',
2. 'job',
3. 'marital',
4. 'education',
5. 'default',
6. 'balance',
7. 'housing',
8. 'loan',
9. 'contact',
10. 'day',
11. 'month',
12. 'duration',
13. 'campaign',
14. 'pdays',
15. 'previous',
16. 'poutcome',
17. 'market?'

The screenshot of the sample dataset is attached below:

A close up of text on a white surface

Description automatically generated

**Code execution steps:**

Learning Classifier System

I have implemented the Learning Classifier System using XCS algorithm to train a shallow feedforward neural network for a 2-class classification task using the xcs library available in python. The accuracy is then compared with the accuracy calculated using Genetic Algorithm, CMA Evolutionary Strategy and particle swarm optimization implemented in previous HWs.

1. XCS python library is installed to implement the Learning Classifier System approach to train the data.
2. Pandas data frame available in python is used to read the data set.
3. The input data set is the continuous data which is converted to discreet data and then into binary data.
4. BankDataset Class is defined which takes the number of training cycles, input size and the input array as parameters.
5. XCS algorithm is applied on the dataset by specifying the default parameters. Few of the parameters are modified to obtain the efficient result. The parameters are modified as shown below:

* algorithm.ga\_threshold = 1
* algorithm.crossover\_probability = .5
* algorithm.wildcard\_probability = .5
* algorithm.deletion\_threshold = 2
* algorithm.subsumption\_threshold = 10
* algorithm.mutation\_probability = .03

1. The prediction accuracy is calculated for the model based on each column.

The snapshot of data set after converting it to discreet data:

A close up of a piece of paper

Description automatically generated

The snapshot of data set after converting it to binary data:

A screenshot of a social media post

Description automatically generated

The snapshot of the model summary:

A screenshot of a cell phone

Description automatically generated

The snapshot of the predicted model accuracy:

A screenshot of a cell phone

Description automatically generated

**Conclusion:**

The **Learning Classifier System** resulted in the accuracy of 99%, 88%, 99% and 99% based on the columns. **Particle Swarm Optimization** resulted in the accuracy of 0.88 i.e. **88**%. **CMA-ES** resulted in 0.715 test accuracy. **Genetic Neural Network** resulted in a test accuracy of 0.89 and the **Sequential Neural Network** resulted in a test accuracy of 0.88.

References

1. [http](http://citeseer.ist.psu.edu/517270.html)://citeseer.ist.psu.edu/517270.html
2. <https://pythonhosted.org/xcs/>
3. Discussed with classmates.