University of Waterloo
Faculty of Engineering
Department of Electrical and Computer Engineering

Gradient Free, Gradient Descent and Hybrid Algorithms
-- ECE 457A Bonus Project

Prepared by
Jinming Zhang (jm3zhang)
Ruizhou Xu (r66xu)
Boyang Cheng (b9cheng)
Yuanxin Wang (y2469wan)

4A ComputerEngineering
5 August 2019

Table of Content

Abstract

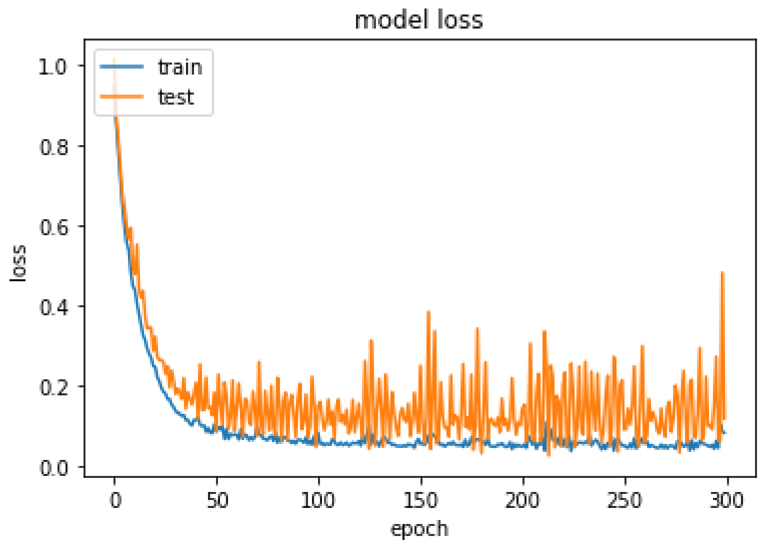
The scope of this project is to examine the behaviour and characteristics of Population Based Algorithms with special focus on the Gradient Based Algorithms. The three algorithms that will be chosen for the project will be the Gradient Descent, Gradient Free and Hybrid Algorithm. This report will compare the three algorithms listed above in terms of their performance and accuracy with qualitative and quantitative analysis. Theoretically, The Gradient Descent Algorithm should have an excellent accuracy, but due to the huge search space, the performance will not be optimal. The Gradient Free Algorithm will have a fast convergence, but the accuracy will not be optimal. This is the motive for choosing to experiment on a Hybrid Algorithm that consists of both Gradient Descent and Gradient Free Algorithm and hopefully it will take advantage of both algorithms strength. As discussed with course instructor, although we are not specifically analyzing the cooperation and adaption nature of the three algorithms, we will point out why we choose certain parameters and show the process of fine-tuning. Due to time limitation, the Hybrid Algorithm's design is referenced from some online repositories.

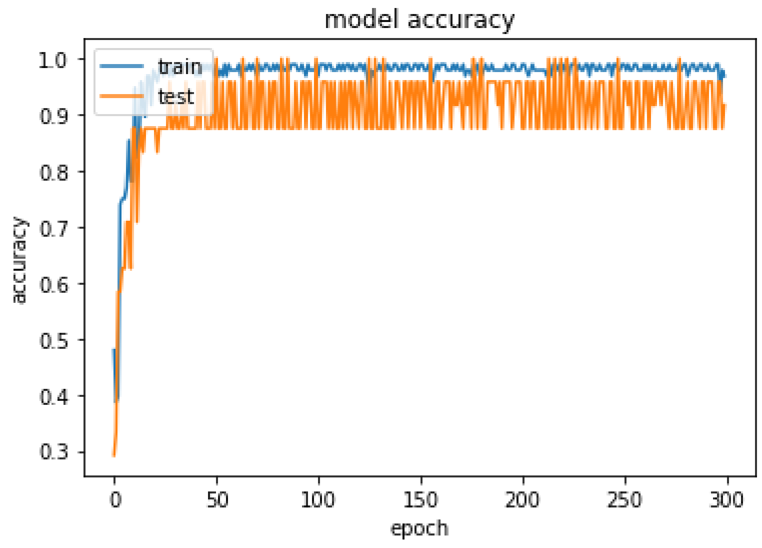
1 Gradient Descent Algorithm

We built a simple tensorflow model for iris classification task. As discussed with course instructor, we only include small number of layers for this project. We only have one hidden layer of size (3, 4). Even with such few layers, the model can still converge to the near-optimal value perfectly.

Quantitative Analysis

In this section of the report, we will focus on the quantitative analysis for the behavior of the gradient based Algorithm. With only 100 epochs / 800 iterations (since we have a batch size of 8), the model can get a validation accuracy of 99% and validation loss of 0.00832. The plots for train and validation are as follows:





Qualitative Analysis

In this section of the report, we will focus on the Qualitative analysis for the behavior of the gradient based Algorithm. It is obvious that as the number of iterations increases, the algorithms will reach lower loss and better accuracy before overfitting happens. It is also worth noting that increasing the complexity of the neural network will also improve the performance. However, 800 iterations may not be a great converge rate for such a simple dataset. Let’s explore more on quick convergence algorithm - PSO in the next section.

2 Gradient Free Algorithm

The Gradient Free Algorithm that will be implemented and analysed in this project will be a Particle Swarm Optimization (PSO) Algorithm. PSO is a population based Meta-Huristics Algorithm. The advantage of PSO Algorithm is that it has a great performance which means this algorithm converge fast. It is based on the idea of simulating the collective behavior of social animals without not leader. The simulation is resulted from the motion equation below.

$$v_{t+1}^{id} = w * v_t^{id} + c_1 r_1^{id}\left(pbest_t^{id} - x_t^{id}\right) + c_2 r_2^{id}\left(Nbest_t^{id} - x_t^{id}\right)$$
$$x_{t+1}^{id} = x_t^{id} + v_{t+1}^{id}$$

where

- *w* is the inertia weight,
- *c_1, c_2* are the acceleration coefficients,
- *r_1,r_2* are randomly generated numbers in [0, 1],
- *t* is the iteration number,
- *i* and *d* are the particle number and the dimension.

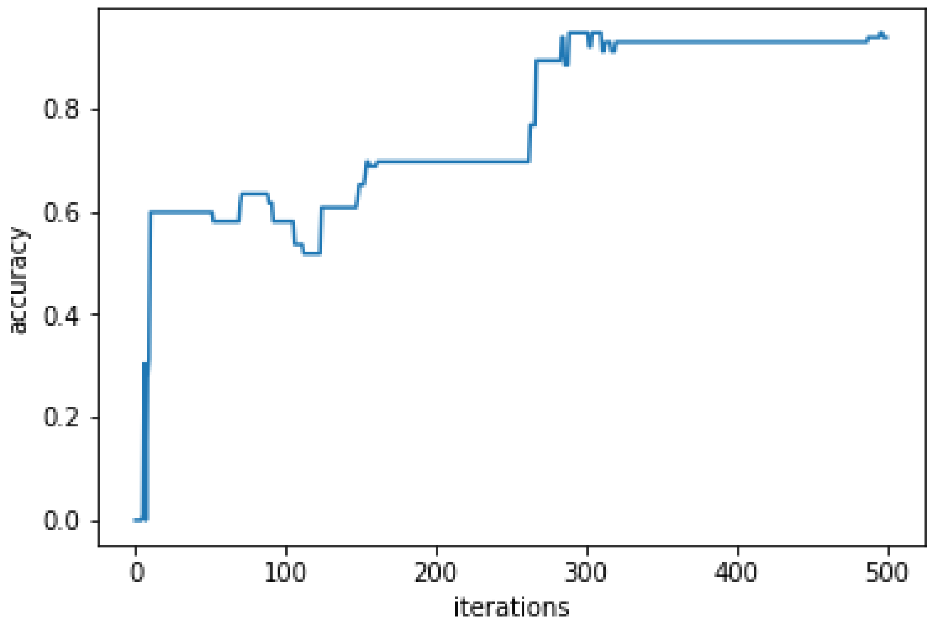
With the motion equation defined, we can simulate the PSO Algorithm with python. The PSO implementation is attached with this report. The quantitative analysis of the PSO Algorithm will be examined in the next section of the report.

Quantitative Analysis

In this section of the report, we will focus on the quantitative analysis for the behavior of the PSO Algorithm. Firstly, we will have a reference run of the PSO Algorithm. With the reference plot, we can compare the effectiveness of different parameters on the performance of the algorithm. The parameters that will be tuned in this section will be the particle size, previous best and global best.

Initial Reference Plot

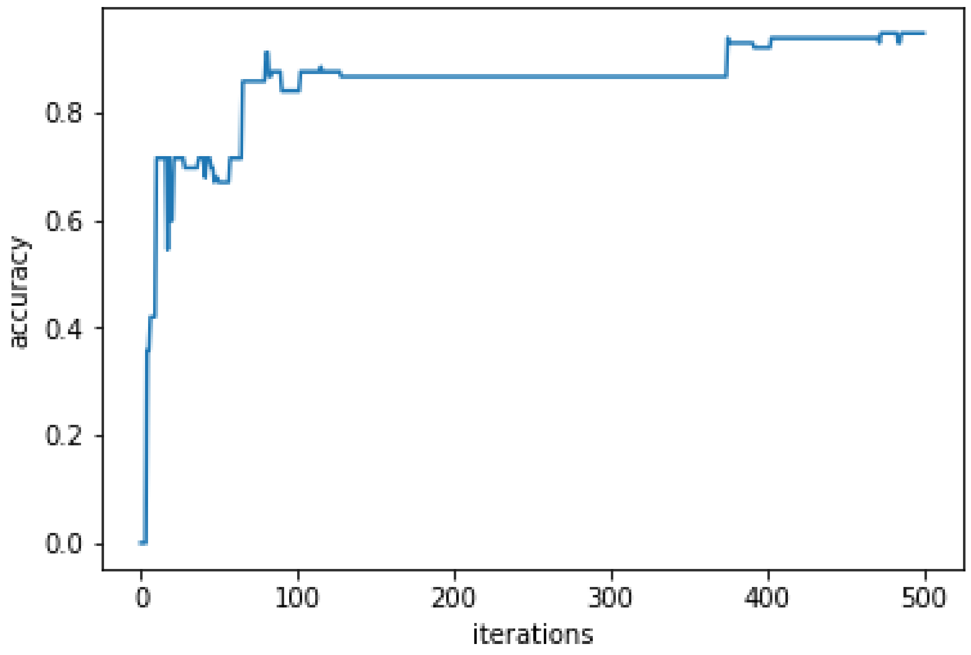
This initial reference plot has a particle size of 32, previous best of 2.05 and batch size of 500. The convergence of the algorithm will be around 300 iterations which shows the characteristic of PSO that it converge fast. The initial reference plot is shown below.



The training accuracy is 0.9375 and the evolution of the accuracy is 0.868421052631579

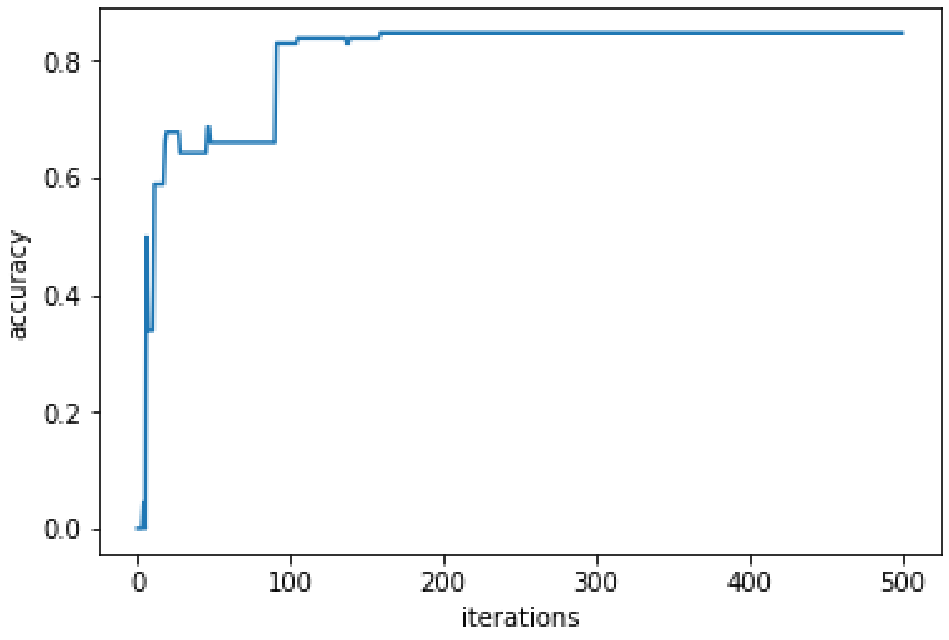
Tuning on Particle Size

The particle size we used for the quantitative analysis will be 10, 60 and 100. The particle size 10 plot is shown below.



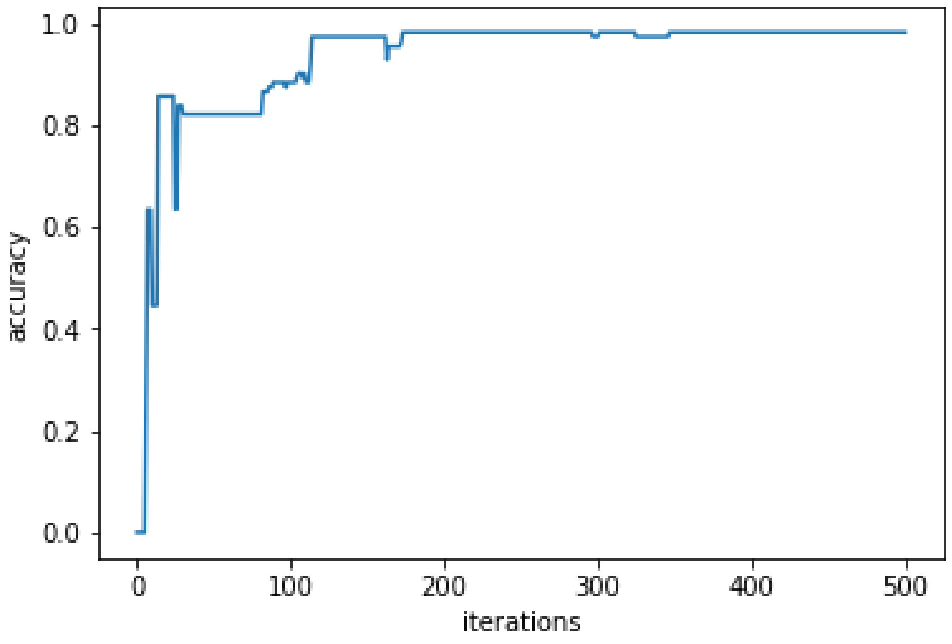
The training accuracy is 0.9464285714285714 and the evolution of the accuracy is 0.8947368421052632.

The particle size 60 plot is shown below.



The training accuracy is 0.9732142857142857 and the evolution of the accuracy is 0.9736842105263158.

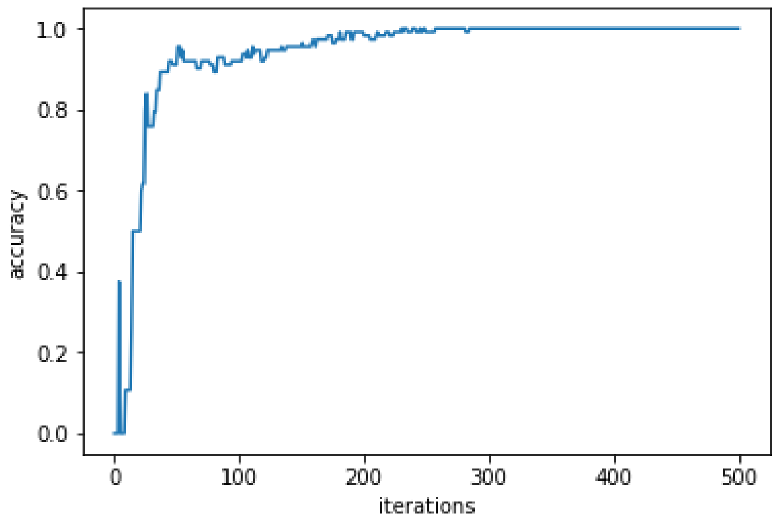
The particle size 100 plot is shown below.



The training accuracy is 0.9821428571428571 and the evolution of the accuracy is 1.0.

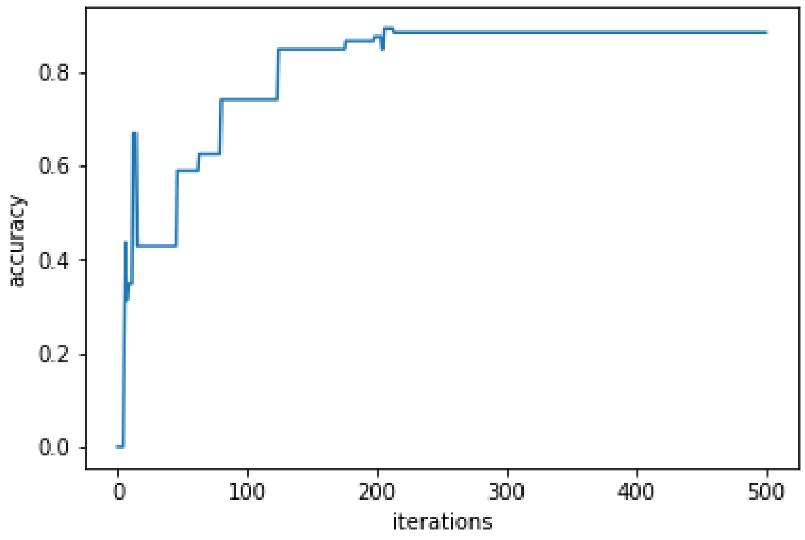
Tuning on previous best

The previous best we used for the quantitative analysis will be 1, 2.5 and 3. The previous best 1 plot is shown below.



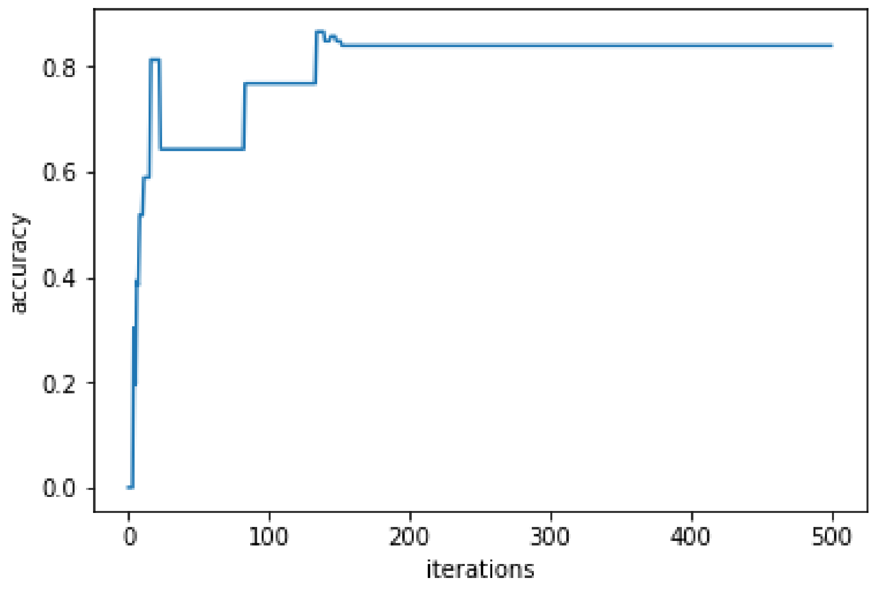
The training accuracy is 1.0 and the evolution of the accuracy is 0.9473684210526315.

The previous best 2.5 plot is shown below.



The training accuracy is 0.8839285714285714 and the evolution of the accuracy is 0.8421052631578947.

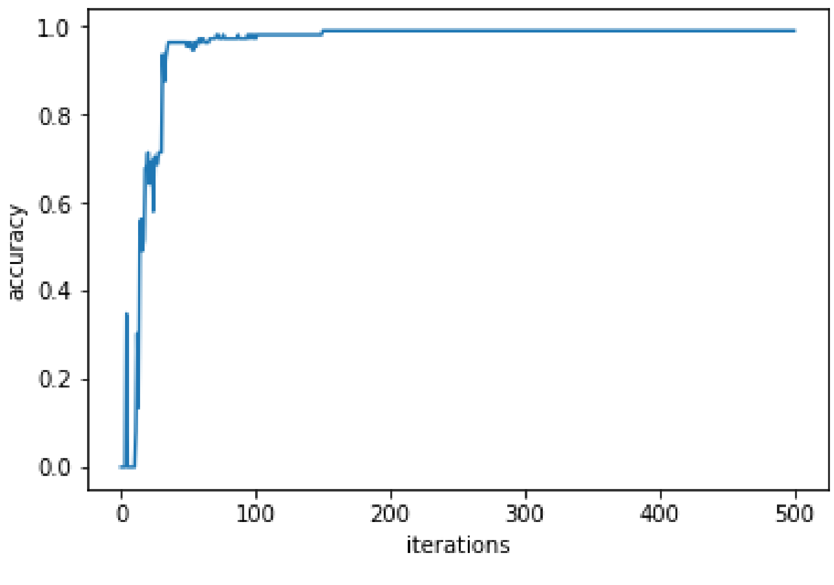
The previous best 3 plot is shown below.



The training accuracy is 0.8392857142857143 and the evolution of the accuracy is 0.868421052631579.

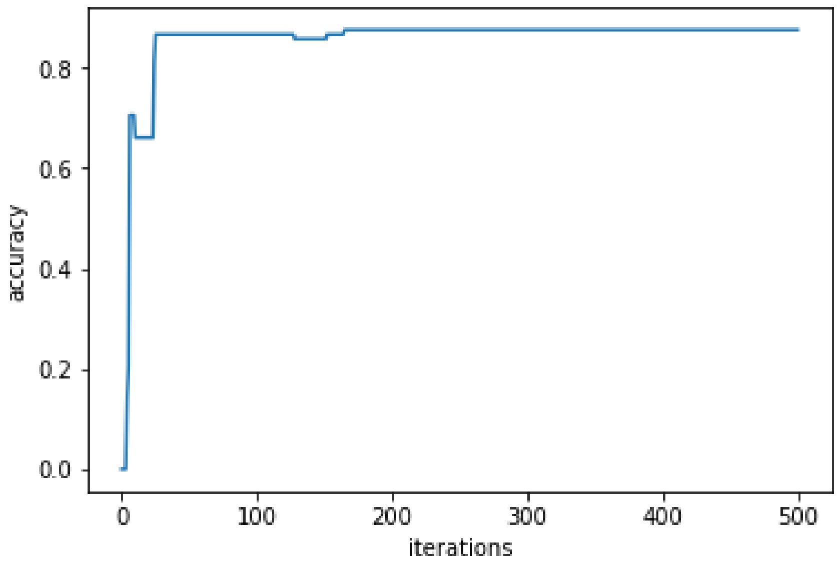
Tuning on global best

The global best we used for the quantitative analysis will be 1, 2.5 and 3. The global best 1 plot is shown below.



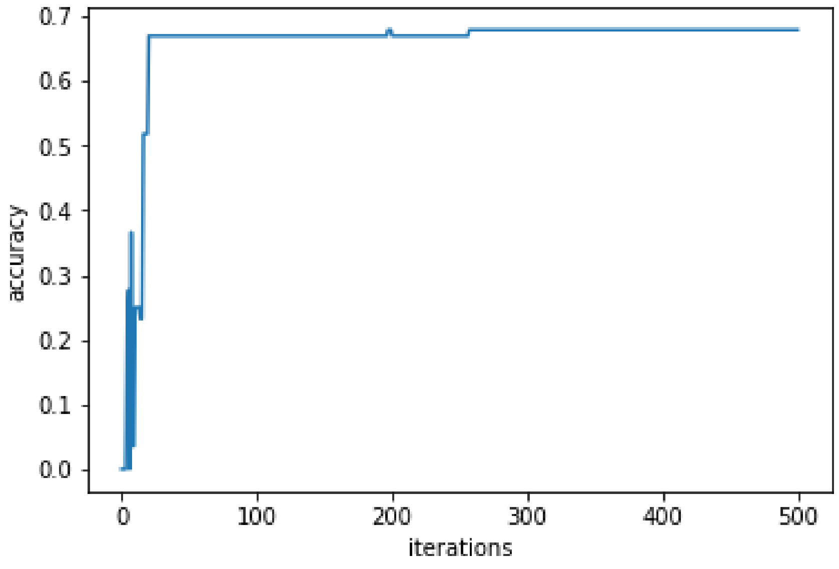
The training accuracy is 0.9910714285714286 and the evolution of the accuracy is 0.9736842105263158.

The global best 2.5 plot is shown below.



The training accuracy is 0.875 and the evolution of the accuracy is 0.7368421052631579.

The global best 3 plot is shown below.



The training accuracy is 0.6785714285714286 and the evolution of the accuracy is 0.631578947368421.

Qualitative Analysis

Comment on particle size

Generally, when particle size increases, the algorithm converges at smaller number of iterations. Also, the final accuracy on both training and eval set increases. However, the computational cost for each iteration increases so that more time is needed for each iteration to complete.

Comment on previous best

Generally, decreasing the previous best coefficient slightly brings up the accuracy on both training and eval sets and makes the curve smoother. When the previous best coefficient is set greater than 3, a significant decrease on the accuracy will occur.

Comment on global best

Generally, decreasing the previous best coefficient slightly brings up the accuracy on both training and eval sets and makes the curve smoother. When the previous best coefficient is set greater than 3, a significant decrease on the accuracy will occur.

Overally, PSO converges faster than gradient based method but it often converges to suboptimal solutions. Is it a way to combine the speed of PSO and accuracy of gradient descent? We will explore such algorithm in the next section.

3 Hybrid Aglorithm

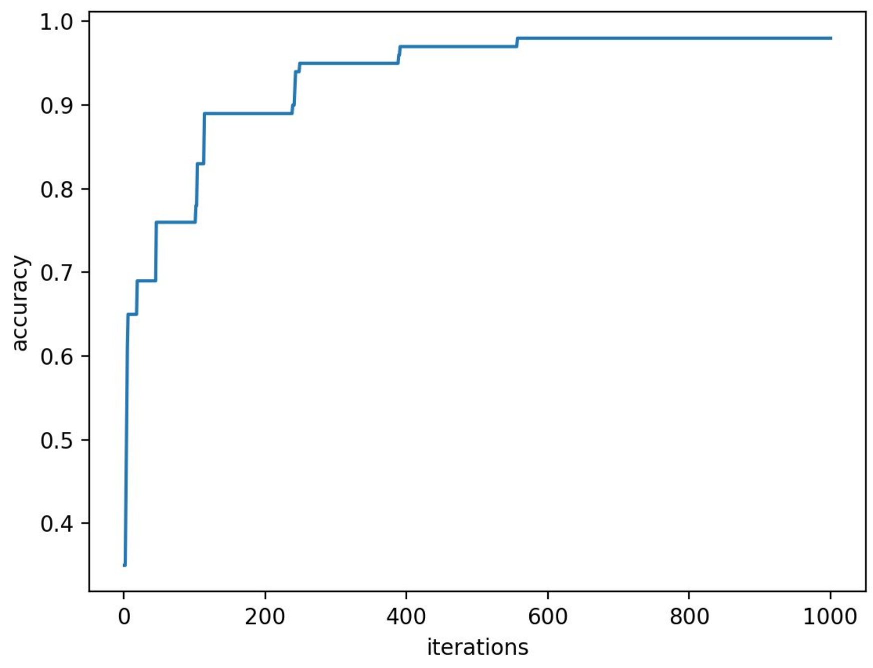
The Hybrid Algorithm that will be implemented and analysed in this project will be a Hybrid of Particle Swarm Optimization (PSO) Algorithm and Deep Neural Network. The dynamics of the PSO Algorithm is thoroughly examined in Section 2 of the report. The advantage of the PSO Algorithm is that it has a great performance which means this algorithm converge fast. However, the trade off for the performance of PSO is that its accuracy is not quite optimal. This disadvantage in accuracy has pushed us to look for a type of algorithm that is not only fast, but also accuracy. With the cost of programming complexity, we found that a Hybrid Algorithm that consists both Particle Swarm Optimization (PSO) Algorithm and Deep Neural Network will have a better accuracy than PSO Algorithm and a better performance than Gradient Descent Algorithm. We will simulate the Hybrid Algorithm with python. The Hybrid Algorithm implementation is attached with this report. The quantitative analysis of the Hybrid Algorithm will be examined in the next section of the report.

Quantitative Analysis

In this section of the report, we will focus on the quantitative analysis for the behavior of the Hybrid Algorithm. Firstly, we will have a reference run of the Hybrid Algorithm. With the reference plot, we can compare the effectiveness of different parameters on the performance of the algorithm. The parameters that will be tuned in this section will be the learning rate, number of particles and global best.

Initial Reference Plot

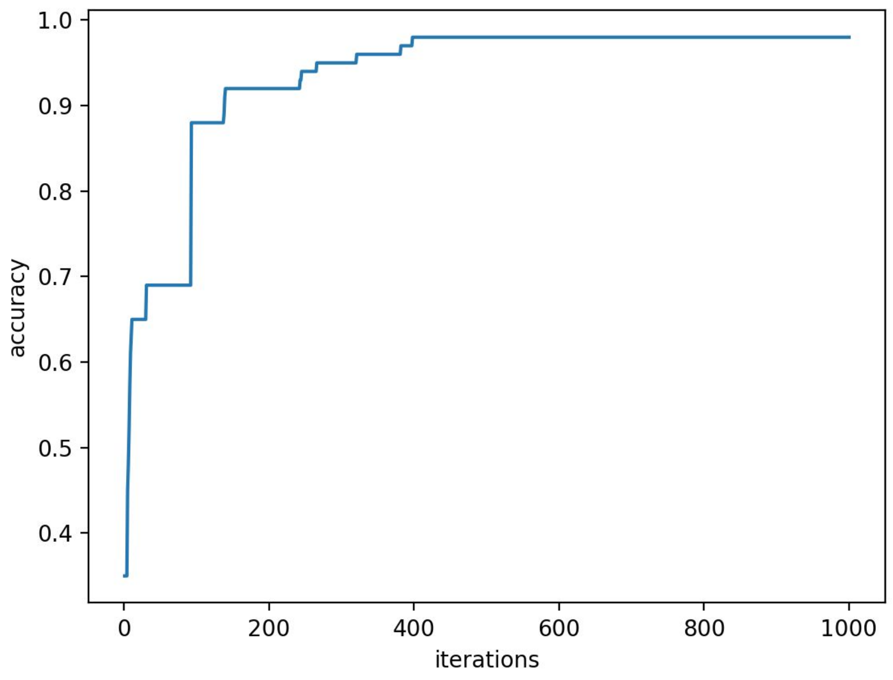
This initial reference plot has a particle size of 32, global best of 0.8, individual best of 0.7 and maximum velocity of 0.005. The performance of the algorithm will be around 600 iterations. It might not be as fast as the PSO algorithm in terms of performance, but it is certainly way faster than the Gradient Descent Algorithm and has a way better accuracy than the PSO algorithm itself. The initial reference plot is shown below.



The training accuracy is 0.98.

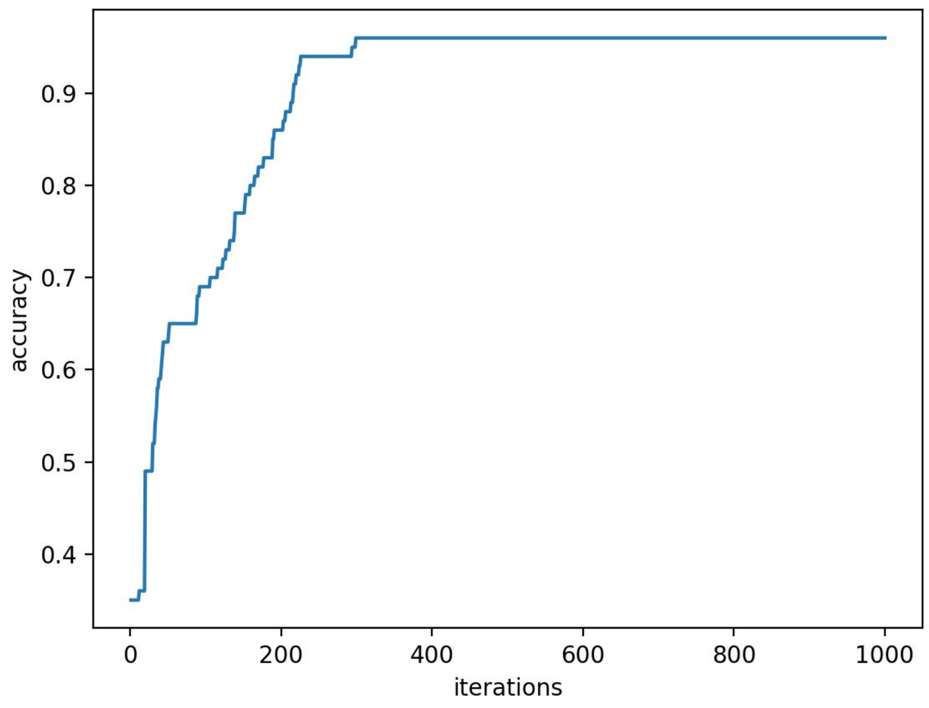
Tuning on learning rate

The learning rate we used for the quantitative analysis will be 0.05, 0.01 and 0.001. The learning rate 0.05 plot is shown below.



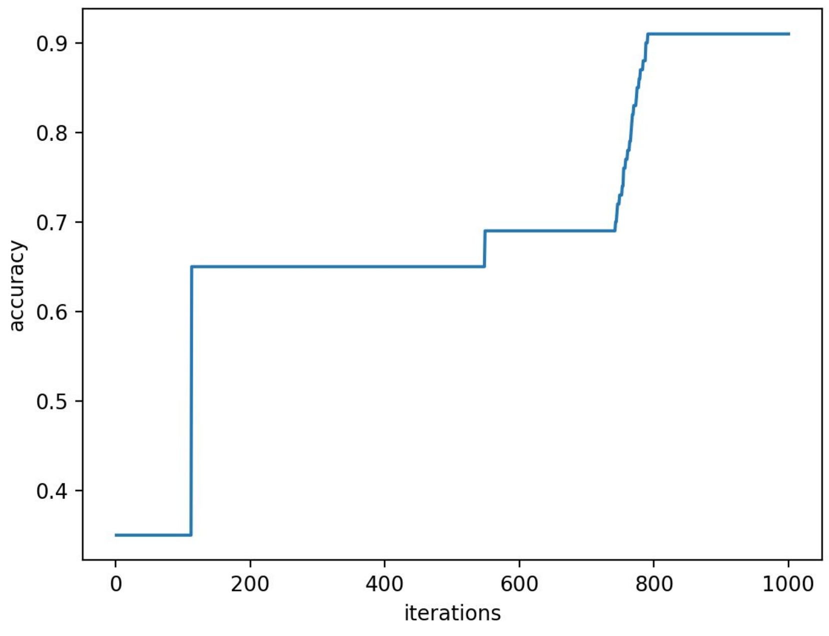
The training accuracy is 0.98.

The learning rate 0.01 plot is shown below.



The training accuracy is 0.96.

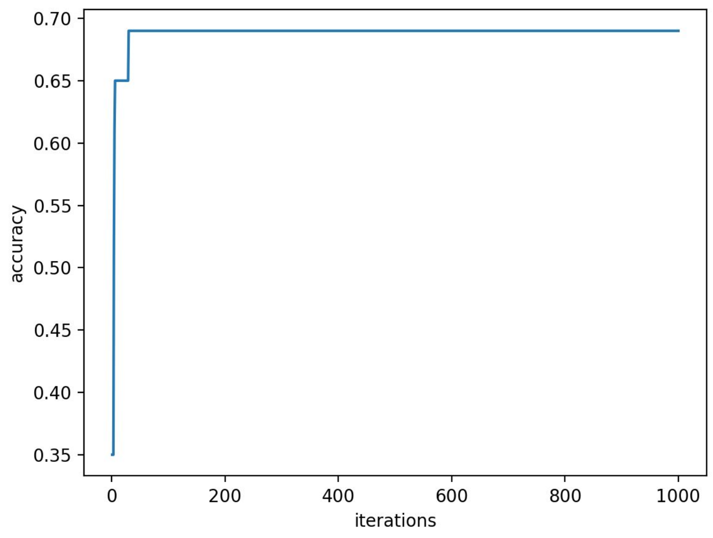
The learning rate 0.001 plot is shown below.



The training accuracy is 0.91.

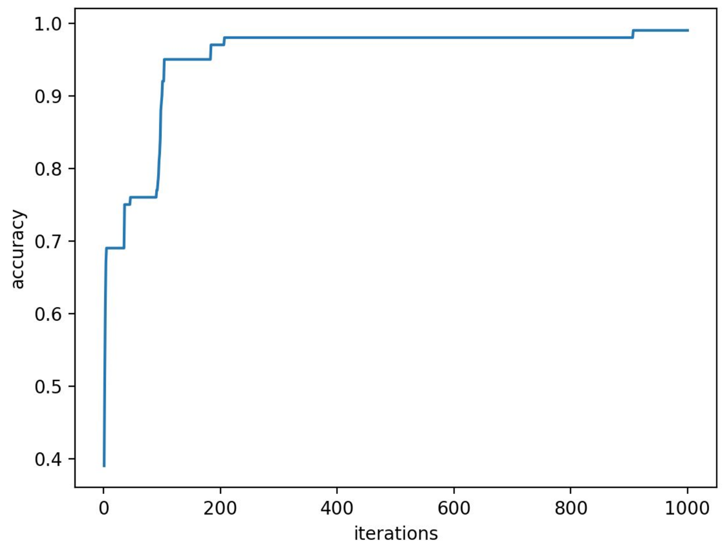
Tuning on number of particles

The number of particles we used for the quantitative analysis will be 10, 60 and 100. The number of particles 10 plot is shown below.



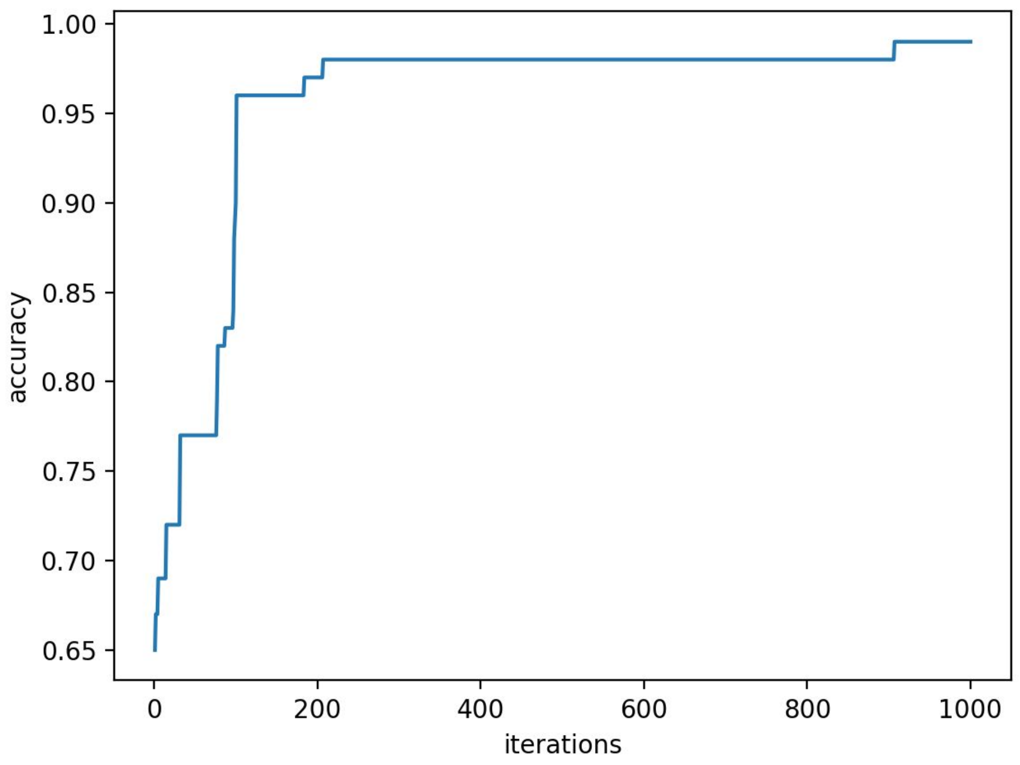
The training accuracy is 0.69.

The number of particles 60 plot is shown below.



The training accuracy is 0.99.

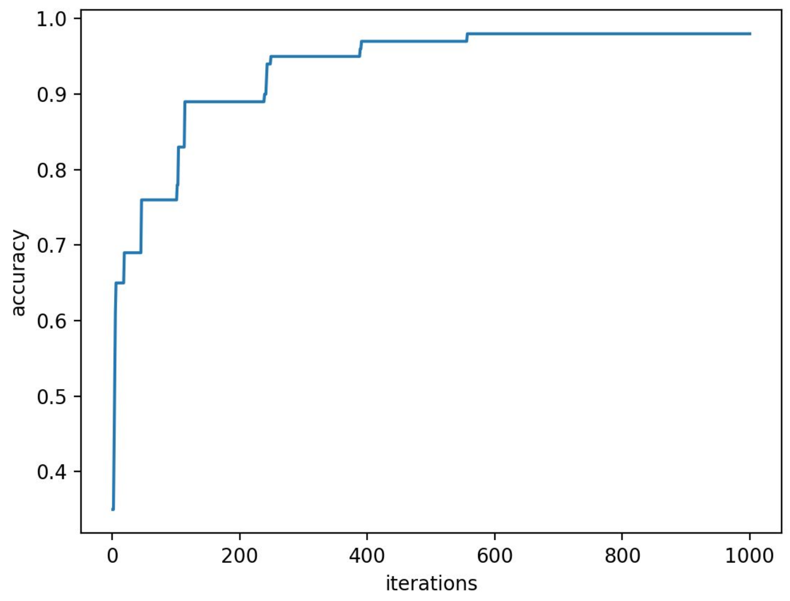
The number of particles 100 plot is shown below.



The training accuracy is 0.99.

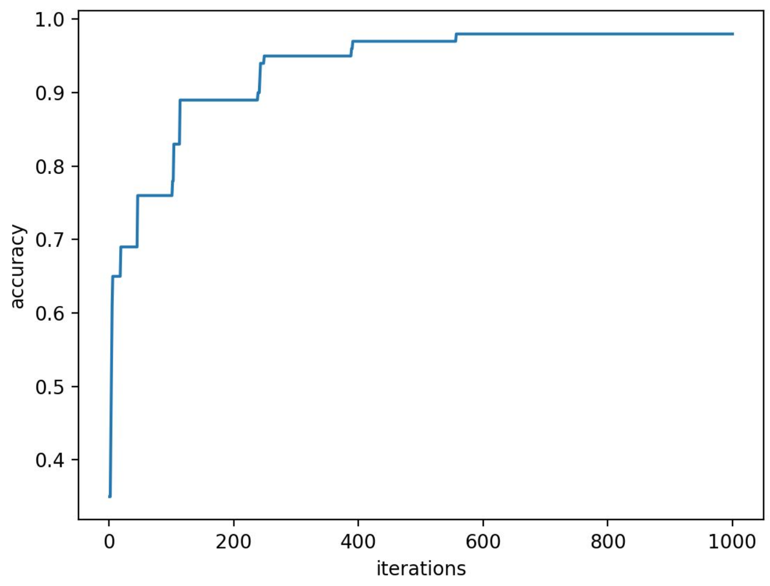
Tuning on Global Best

The Global Best we used for the quantitative analysis will be 0.1, 0.2 and 0.5. The Global Best 0.1 plot is shown below.



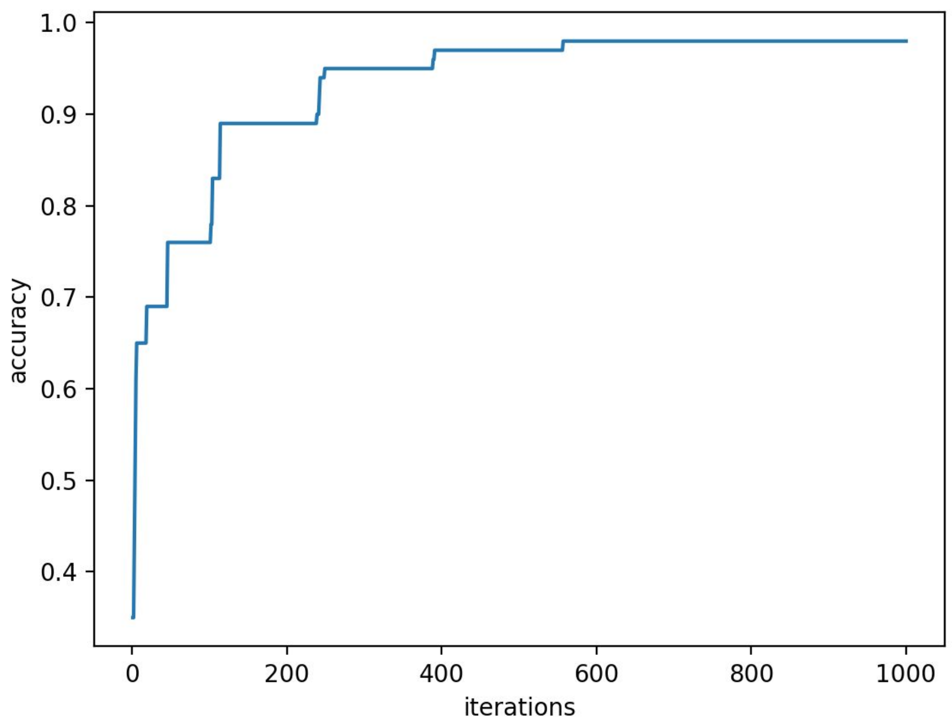
The training accuracy is 0.98.

The Global Best 0.2 plot is shown below.



The training accuracy is 0.98.

The Global Best 0.5 plot is shown below.



The training accuracy is 0.98.

Qualitative Analysis

Comment on Learning Rate

In this experiment, decreasing the Learning Rate slightly will result in a decrease the accuracy on training Accuracy. We should try larger learning rate as first and apply a learning rate decay schedule in the last few epochs. The initial large learning rate will allow the algorithm to do more exploration in the search space and in the last few epochs smaller learning rate will help us get more exploitations.

Comment on number of particles

Generally, when the number of particles is small, the algorithm converges at smaller number of iterations. Also, the final accuracy on training increases when the number of particles increases. However, the computational cost for each iteration increases so that more time is needed for each iteration to complete.

Comment on Global Best

Generally, when the Global Best has little or even no effect on both the accuracy and the converge speed of the Hybrid Algorithm.

Overally, we can see that the hybrid approach converges a little bit faster than the gradient based method and reaches more accurate values than PSO algorithm. The differences are not very large since this task is very simple and it is nearly impossible for any algorithm to perform very bad on it. However, we still see an improvement which agrees with our hypothesis.

4 Comparison and Conclusion

In conclusion, the experiment results agrees with our hypothesis. On the one hand, Gradient Descent algorithms are very accurate in terms of finding the global optimum but it takes many runs to reach there and sometimes even suffer from local optimum problem. On the other hand, Gradient Free algorithms such as PSO can converge really fast to a solution, but this solution might not be desirable in terms of optimality. The Hybrid approach works both quick and accurate in this Iris Dataset experiment but it runs a little slower due to high computation cost.

Also, readers may question that in some cases the gradient descent approach in our experiment is nearly as fast as the other two, our argument is that since we are using Keras, a mature deep learning framework for the gradient descent method while using only numpy and some simple tensorflow ops, the GPU acceleration built for Keras may be much better than that of numpy and thus it is normal that the gradient descent method runs unexpected fast.

In summary, each of the three algorithms has its own advantages and disadvantages. It would be advisable to carry out more research into how to combine the meta-heuristics algorithms with back propagation to improve the efficiency in the machine learning industry.

Reference

[1] https://github.com/asbudhkar/ANN-PSO
[2] https://github.com/piyush2896/PSO-for-Neural-Nets
[3] https://ieeexplore.ieee.org/document/8068573