Improving Soybean Disease Prediction by Performing Late-Stage Re-Training Using Fireworks Algorithm

***Abstract*— This paper proposes a novel method of training and optimizing a model to get as optimal a result as possible. Contemporary models emphasize bigger dataset sizes and better preprocessing. This study aims to shed light on alternative methods of using multiple optimizers with different properties to get better results. This incremental increase is achieved by switching the optimizer after training once, hence performing late-stage re-training.**

The present study shows the use of 2 distinct optimizers for their unique properties to achieve Late-Stage Re-Training (LSRT). Here, a different optimizer is used to tune our parameters (i.e., "Re-Training") after a model has achieved a certain level of accuracy using another optimizer.

Existing methods of creating ANNs are utilized by using the APIs within Tensorflow and Keras. Furthermore, the Tensorflow Optimizer class is used to create our swarm algorithm within Tensorflow.

***Keywords*—Optimizers, Fireworks Algorithm, Adam, Soybean Dataset Classification, Artificial Neural Network, Artificial Intelligence, Machine Learning**

I. Introduction

A. Initial Motivation

At the beginning of the present study, the idea of late-stage training came from the need to create more accurate models using an amalgamation of presently used technologies to create a procedure to train a model that is better than the individual sum of its parts. Most of these models are essential in fields where accuracy or efficiency may be paramount and extract actionability [4]. Such models are used in sectors such as Healthcare, Finance, Automobiles, etc., where the Bayes error rate is minuscule (close to or less than 5%). During the first round of training, making big waves in accuracy and loss metrics might be more convenient than closing that last 10% gap. This is where the finding of the present study comes in.

The architecture to best experiment on such a method is best suited to be a classification model, a technology whose implementations are quickly permeating through modern life. Hence, for the present study, the `Soyabean Disease` Dataset [5][6] was chosen due to its large number of classes (~19).

Another critical step of this study was to choose a pair of optimization algorithms that might work well together for two different parts of the training process. The first part of the training is optimized using the Adam Optimizer [7] for its quick convergence and the use of momentum; conversely, the second part of the training uses an implementation of the swarm intelligence algorithm, that is, the Fireworks Algorithm.

The following study discusses and presents the outcomes of this idea.

B. Soyabean Disease Classification

The growing human population needs more and more resources each year. One of the key resources is food and grain. The growth demand thus means an increase in crop demand. Plant diseases and infections largely cause agricultural losses around the world. Plant ailments, weeds, insects, and chemicals have all been responsible for a decline in worldwide crop production. It is therefore of paramount importance that diseases are detected early to prevent their spread and reduce crop losses. Agriculture has for decades used cultural practices to protect crops from pests and diseases—mostly in conjunction with solarization and crop rotation—to provide some protection from harmful pests, as well as the development of pesticide-resistant cultivars, in conjunction with the use of biological agents.

A healthy agroecosystem depends on the effective detection of diseases and ailments. With the development of molecular biology and biotechnology, detecting plant diseases has become more efficient. Preventing and controlling soybean diseases requires an automated diagnostic system. The obvious goal is to maximize yield and minimize economic losses, which can in turn be achieved by curbing pesticide residues on the land and improving the quality of the crops. To predict soybean diseases at an early stage, it is necessary to classify and categorize the diseases effectively. Several learning algorithms have also been applied to predict pest attacks and disease infestations in crops. An algorithm has been developed to compare aerial parts of healthy and diseased plants using spectral imaging data. In addition to monitoring the crops' morphological traits, there is also a substantial amount of evidence that ML methods are successful as well.

Diaporthe Stem Canker	Charcoal Rot
Rhizoctonia Root rot	Phytophthora rot
Brown Stem rot	Powdery Mildew
Downy Mildew	Brown Spot
Bacterial Blight	Bacterial Pustule
Purple Seed Stain	Anthracnose
Phyllosticta Leaf Spot	Alternarialeaf Spot
Frog Eye Leaf Spot	Diaporthe Pod and Stem Blight
Cyst Nematode	2 4 d Injury

Fig 1: Soybean Diseases

Nevertheless, the varying nature of plant changes may lead to inaccurate predictions due to changes in symptomatology. Thus, physical or appearance-based disease identification cannot be relied on to identify diseases reliably—especially at the early stages of growth. To identify the causative agent of rot, it is important to have an appropriate detection method since the symptoms do not appear until the crops' midseason. The method for predicting the occurrence of charcoal rot disease used in the present study also includes the plants' morphological characteristics (including characteristics related to growth and yield) as well as their physiological features. To train and assess machine learning algorithms, a hybrid set of features derived from both healthy and diseased soybean plants are used.

The present study uses machine learning models to distinguish healthy plants from those that are unhealthy. It has further been shown that supervised machine learning algorithms can be used to model disease classification, and such algorithms can be used for disease prediction. The objective of this study is to propose a set of features for enhancing the prediction of soybean diseases. Agriculture crop yield is greatly affected by many pests and diseases. To improve the yield, early diagnosis and control of such diseases are essential.

For a long time, Soybean has been a rapidly growing crop in India, and it is considered a 'Kharif crop'. The leading producer of soybean in India is the state of Madhya Pradesh, followed by Maharashtra and Rajasthan [2]. Around the world, the Soybean is recognized as a 'wonder crop'. Soybean is an important food commodity because of its high nutritional content (especially protein, which is present at levels that are greater than 40%) and high oil presence (greater than 20%). Since the soya protein supplies enough amino acids, it is called a complete protein. Soybean oil does not contain any cholesterol. It is also used as animal feed. Low yield is a significant problem in the soya industry of the country, and crop diseases are one of the leading causes of such scarcity.

Soybeans are affected by a multitude of diseases like downy mildew, pod, stem blight, phytophthora root, stem rot, brown spot, Cercosporin leaf blight, purple seed stain, and frog eye leaf spot, to name a few. The present study deals with the classification of soybean diseases based on weather data, physical crop properties, plant properties, and crop management properties. The dataset is available in the UCI Machine Learning Repository. We will use machine learning techniques such as ANNs for this.

C. Fireworks Algorithm

Fireworks Algorithm (FWA) is a novel swarm intelligence algorithm introduced by Ying Tan [4]. FWA is one method of collaborated intelligence groups—a technique that simulates the behaviour of fireworks in the night sky. The algorithm is generally used for solving optimization by searching for the optimum value inside the solution space of specific problems.

An explosion of fireworks occurs when fireworks are shot up into the sky, and they move to an appropriate position and explode into shining lights of various shapes, depending upon the designs of the fireworks. Each firework has a specific

Commented [AS11]: Look again

explosion characteristic—for example, there can be a narrow explosion with a lot of sparks, a vast explosion with fewer sparks, a vast explosion with many sparks, and more. Each firework is defined by a unique explosion and the different positions at which they are set off. In this research, FWA is used as an algorithm for training ANNs and determining the proper weights and biases. The proper values are chosen from the processing of the error values between the predicted results and the actual results. This algorithm has the characteristics of a repeating cycle until a suitable result is achieved. The work done in the present study is divided into four main parts, as described in the Methodology section.

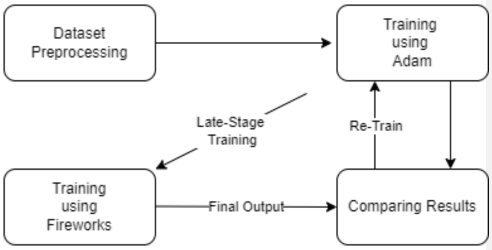
Algorithm : Pseudocode of training ANNs using FWA

```
Read weather datasets as input
Do filtering process
Do normalization process
Divide data into Training and Testing set
Initialize ANNs parameters, FWA parameters and maximum
number of training cycles
Generate random fireworks for the first generation
For i=1 to maximum number of fireworks do
    Calculate error;
End for
Initialize the best position
For j=1 to maximum number of training cycles do
    For i=1 to maximum number of fireworks do
        Calculate amplitude; Calculate number of sparks;
        Generate regular sparks;
    End for
    Generate Gaussian sparks;
    Evaluate sparks;
    Select number of sparks for next firework;
    Generate new number of fireworks;
End for
Return the best weights and biases value
```

Fig 2: Fireworks Pseudocode

II. Methodology

When deciding what algorithm to choose for the optimization step during the training of neural network models, there is also a need to take time into consideration. Thus, the present study considers various factors regarding training time and the best fit in terms of the data. However, it can also be intuitively understood that different sections of training can benefit most from different optimization techniques.



Methodology Flow

A. *Hypothesis*

In the present study, first, the Soybean classifier is trained using the Adam Optimizer, which has been used in several implementations of this problem statement. Nevertheless, as the algorithm gets trapped in a local minimum, the Firework algorithm is called upon to use built-in sparks to train further the model for better results, for both the training and testing data.

Commented [AS12]: instances

B. *Why Fireworks*

Fireworks is a swarm intelligence algorithm that uses 'explosions' to semi-randomly find out the next optimal step in our solution space. It is a great method to get out of any local minima that the study might have gotten trapped into, due to the linear descending structure of Adam.

When fireworks itself is used as the only optimizer used during training, it does not give appropriate results, nearing only a measly 60% train accuracy on average and therefore being much behind industry-standard results that are generated from legacy algorithms. On further inspection, however, the present study finds that after the initial burst, the rate of increase of the 'train' accuracy decreases exponentially. This makes Fireworks a great secondary algorithm, while it performs poorly as a primary optimization algorithm.

C. *Limitations & Alternatives*

One of the known drawbacks of the Adam optimizer is that in some scenarios, Adam does not converge to the optimal solution. Also, slow convergence and low accuracy are some major issues with the Fireworks Algorithm. But if these operations are used in tandem, they proceed to rectify each other's drawbacks, i.e., The initial training that is done using Adam provides a quick and accurate starting point. Further passes using Fireworks takes care of any local minimums and brings the solution to the global minimum.

This approach still has several issues, such as the Weight Decay problem in Adam and Explosion Tuning in Fireworks. These problems can be circumvented using alternative versions of these algorithms themselves, iti.e., using the dynamic search firework algorithm (dynFWA) and AdamW or AMSGrad.

Furthermore, the pair of optimizers used in the present study might not be the best in terms of all-rounded use with other datasets as well. This makes the process of choosing the appropriate pair of optimizers more difficult.

III. Experimental setup

A. Dataset

The soybean dataset is retrieved using the 'pmbl' library, where the data has more than 30 columns (such as: 'date', 'plant-stand', 'precip', 'temp', 'hail', 'crop-hist'). This data is imported, converted into a data frame, and then the necessary splits (X, Y, test, train) are made.

B. Creating Custom Optimizer

Using the built-in Optimizer class of TensorFlow, the present study uses a custom-created implementation of the Fireworks Algorithm as a new class that can be called inside the main driver code. To create an optimizer function, an Object Function is needed to create and evaluate the sparks array inside the Fireworks class that is used to initialize new possible optimal values. The Object Function and the number of dimensions and maximum iterations are taken as parameters.

C. Training Model

Now, the process of creating and training the model begins. As this happens, two instances of optimizers are created—one being Adam and the other being Fireworks. A sequential model is created with three dense layers with 56, 28, and 19 nodes, respectively.

Then, a new TensorFlow session is started, followed by compiling, training, and finally evaluating the results and data.

IV. Results

Operation	Train Accuracy	Test Accuracy
1st Pass (Adam)	94.47%	90.58%
2nd Pass (Adam)	97.12%	89.69%
3rd Pass (Fireworks)	98.89%	92.03%

Fig 3: The results of the 3-step training

In Fig. 3, the study shows that during training, the following processes take place:

A. First Pass (using Adam)

During the first training pass, the model accuracy begins from around ~40% accuracy in the first epoch; this increases to nearly ~94.5% by the last epoch. The speed and accuracy of Adam are utilized to get a well-trained model as the starting point.

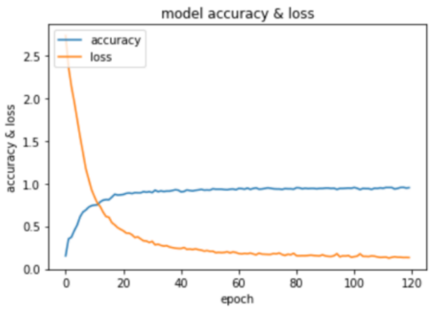


Fig 4: The results of First pass of the training

B. Second Pass (using Adam)

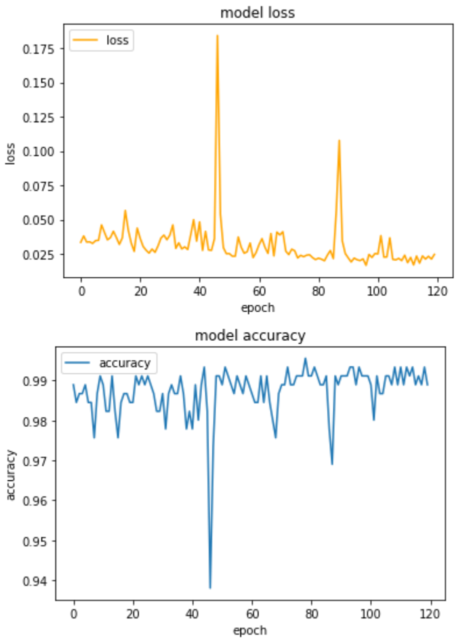


Fig 5: The results of the Second pass of the training

In this next phase of the training, in the current study, the `model.fit()` function is run again to re-train the existing model on the dataset using Adam once more. This operation gives a slightly better train accuracy but nearly the same test accuracy (if not lower). Subsequently, no matter how many times the model is re-trained, the test accuracy does not budge.

C. *Third Pass (using Fireworks)*

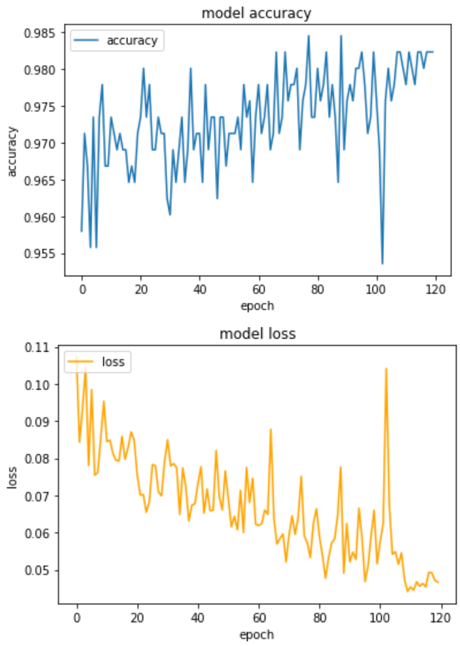


Fig 6: The results of the Third pass of the training

In the next phase of the training, Fireworks is now used as the optimizer, and the model is re-trained. After training, there is an equivalent increase in both the test and train accuracy of the dataset. While with Adam, the accuracy and loss were stagnant, Fireworks provides the last stage of training to get better results.

V. Conclusion and Discussion

In conclusion, a new way of getting better results was discovered on both our test and train data with the help of the hybrid implementation of optimizers. Not only does the train accuracy increase after initially plateauing while using the original optimizer but the test accuracy increases by a huge margin after initially re-training with the Adam optimizer. Hence, using the Fireworks Algorithm and, therefore, LSRT (Late-Stage Re-Training), the present study was to create a model that was ~4% more accurate on train data and nearly ~2% on test data.

(TODO: Discussion, i.e., comparing other studies)

VI. Future work

In the future, the focus will be on more hybrid optimization combinations and further research on consequent training. Furthermore, an optimizer can be created to use the multiple initializations of a swarm algorithm with the concept of momentum, alpha, and beta from traditional gradient descent algorithms.

The hypothesis provided by the current study needs to experiment on further datasets that need LSRT (Last-Stage Re-Training), such as COVID-19 prediction classification problems.

VII. Acknowledgment

This paper and its research would not have been possible without the exceptional support of my supervisor, Prof. Rajashree Krishna. Her enthusiasm, knowledge, and exacting attention to detail have been an inspiration and kept this work on track from its first encounter to the final draft of this paper.

I am also grateful to MIT Manipal for providing me with the resources to conduct this study.

Another group of fellow researchers that I would like to thank are Weidi Xu and Yifeng Li from the Computational Intelligence Laboratory (CIL), Peking University, for publishing the working code for their implementation of the Fireworks Algorithm from their research paper.

VIII. References

[1] [Dr. Nanda Ashwin1, Uday Kumar Adusumilli2.] A Machine Learning Approach to Prediction of Soybean Disease, (ISSN: 2394-4099)

[2] [Rajashree Krishna, Prema K V.] Soybean crop disease classification using machine learning techniques, (INSPEC Accession Number: 2027-9158)

[3] [Saktaya Suksri, Warangkhana Kimpan.] Neural Network Training Model for Weather Forecasting Using Fireworks Algorithm, (INSPEC Accession Number: 16692911)

[4] [Qiang Lyu, Yixin Chen, Zhaorong Li, Zhicheng Cui, Ling Chen, Xing Zhang, Haihua Shen.] Extracting Actionability from Machine Learning Models by Sub-optimal Deterministic Planning (arXiv:1611.00873 [cs.AI])

[5] [Tan, M., & Eshelman, L. (1988).] Using weighted networks to represent classification knowledge in noisy domains. (Proceedings of the Fifth International Conference on Machine Learning (pp. 121-134). Ann Arbor, Michigan: Morgan Kaufmansn.)

[6] [Fisher, D.H. & Schlimmer, J.C. (1988).] Concept Simplification and Predictive Accuracy. (Proceedings of the Fifth International Conference on Machine Learning (pp. 22-28). Ann Arbor, Michigan: Morgan Kaufmann.)

[7] [Diederik P. Kingma, Jimmy Ba] Adam: A Method for Stochastic Optimization. (arXiv:1412.6980 [cs.LG])

[8] [Haoran Luo; Weidi Xu, Ying Tan] A Discrete Fireworks Algorithm for Solving Large-Scale Travel Salesman Problem. (INSPEC Accession Number: 18147040)