A Brief Comparison of Optimization Algorithms in ANN Using Simple Evaluation Metrics

Shahidul Islam
Department of CSE
Fareast International University
Dhaka, Bangladesh
badhon279@gmail.com

Riashad Hassan
Department of CSE
Fareast International University
Dhaka, Bangladesh
riashadhassan1@gmail.com

Shabab Affanul Hoque
Department of CSE
Fareast International University
Dhaka, Bangladesh
shababaffanul@gmail.com

***Abstract*— In several fields, deep neural networks have demonstrated their effectiveness. However, as neural networks get more complex and datasets get larger, optimizing these networks has gotten harder. This research paper aims to compare the performance of different optimization algorithms in an artificial neural network (ANN). The algorithms evaluated include five popular optimizers currently in use. The effectiveness of these algorithms has been compared on the Breast Cancer Wisconsin dataset by figuring out their differences in evaluation metrics including accuracy and loss functions.**

Keywords—Optimization algorithms, Artificial Neural Network, Loss Function, Accuracy, Stochastic gradient descent, Adam, Breast cancer classification, Adam optimizer.

I. INTRODUCTION

Artificial neural networks (ANNs) excel at learning complex patterns and relationships in data, making them essential in machine learning and deep learning. ANNs can model nonlinear relationships, automatically learn relevant features from data, and achieve state-of-the-art performance in various fields. Optimization algorithms are computational methods used to adjust the parameters of a neural network model in order to minimize or maximize a specific objective function. They play a crucial role in various fields, including machine learning and deep learning. The goal of optimization is to find the optimal set of parameter values that yield the best performance or accuracy for a given task.

II. RELATED WORK

In Neural Network literature, performance analysis of optimization algorithms is vastly studied. As example, a study [1] compares the performance of different optimizers, including SGD, Adam, RMSprop, and Adagrad, on various deep learning tasks. The authors evaluate the optimizers based on accuracy metrics such as classification accuracy and mean squared error. Another paper [2] focuses on the performance comparison of optimizers for convolutional neural networks (CNNs). The study investigates optimizers such as SGD, Adam, RMSprop, and Adadelta and analyses their impact on accuracy metrics, with a specific emphasis on image classification tasks. Furthermore, a group of authors [3] analyse the effects of optimizers, including Adam, AMSGrad, and SGD, on the convergence and generalization of artificial neural networks. They examine accuracy metrics and highlight the importance of considering both optimization dynamics and generalization performance. Another study [4] compares the performance of various optimizers, such as SGD, Adam, RMSprop, and Adagrad, on different architectures of artificial neural networks. The authors evaluate the optimizers using accuracy metrics and provide insights into the selection of optimizers for specific tasks.

III. METHODOLOGY

A. Optimization algorithms

The foundation for a machine's ability to learn through experience is rooted upon a set of optimization algorithms. They are used to measure gradients and minimize the loss function. The algorithms studied in this study are described below.

1) ***Stochastic Gradient Descent**:* Stochastic Gradient Descent (SGD) is an efficient optimization algorithm for training machine learning models. It updates model parameters using small randomly selected subsets of data, known as mini-batches. By computing gradients and iteratively adjusting parameters, SGD converges towards an optimal solution. Its speed and memory efficiency make it widely used in large-scale models.

2) ***Adam:*** Adam (Adaptive Moment Estimation) is an optimization algorithm that combines adaptive learning rates and momentum. It efficiently updates model parameters by maintaining averages of past gradients and squared gradients. The algorithm incorporates bias correction to alleviate initial bias towards low values. Adam is widely used in machine learning and deep learning for its effectiveness and robustness.

3) ***Adagrad:*** Adagrad (Adaptive Gradient Algorithm) dynamically adjusts the learning rate for each parameter based on the historical accumulation of squared gradients. Adagrad is particularly beneficial for handling sparse data, as it assigns larger updates to infrequent parameters and smaller updates to more frequent ones. This adaptability allows Adagrad to effectively optimize model performance without the need for manual learning rate tuning.

4) ***RMSProp:*** RMSProp (Root Mean Square Propagation) is an optimization algorithm widely used in machine learning and deep learning. It addresses the limitations of Adagrad by alleviating the rapid decrease in learning rates. RMSProp maintains a moving average of squared gradients and utilizes this information to normalize the learning rate for each parameter update. By adapting the learning rates based on the magnitude of recent gradients, RMSProp improves the convergence and stability of the optimization process. It is particularly effective in scenarios with sparse gradients or non-stationary objectives.

5) ***Nadam:*** Nadam (Nesterov-accelerated Adaptive Moment Estimation) is an advanced optimization algorithm used in machine learning and deep learning. It combines the benefits of Nesterov accelerated gradient (NAG) and Adam, leading to efficient parameter updates. Nadam incorporates adaptive learning rates, similar to Adam, and integrates NAG to adjust the update direction based on a future gradient estimate. This combination enables Nadam to converge faster and achieve better optimization performance compared to traditional algorithms. It is particularly effective for deep learning models, offering accelerated convergence and improved results.

B. *Dataset*

The performances of the aforementioned algorithms are evaluated on a breast cancer classification dataset inbuilt in *scikit-learn* library. It can also be found in *UCI ML repository* as the "Breast Cancer Wisconsin (Diagnostic Data set)". The dataset contains 569 instances with 30 attributes and zero missing value. The target values consist of two outputs of "Malignant" (0) and "Benign" (0).

C. *Evaluation metrics*

In machine learning, *accuracy* is the most common evaluation metric used to measure the performance of a model. It represents the proportion of correct predictions made by the model out of all predictions.

$$Accuracy = \frac{Correct\ Predictions\ Count}{Total\ Predictions}$$

For the loss function, sparse categorical cross-entropy function has been used. It's similar to standard categorical cross-entropy loss function, but more memory efficient and computationally robust than that.

$$L_{\mathrm{CE}} = -\sum_{i=1}^{n} t_i \log(p_i), \quad \text{for n classes,}$$

where t_i is the truth label and p_i is the Softmax probability for the i^{th} class.

IV. Results

A. *Training the dataset*

The training process commences by importing necessary libraries, putting the dataset into a structured data frame. Data standardization had been done for better accuracy in results. Then, the dataset is put through a different model for each optimizer. But the activation function for the layers, epoch size, loss function etc. are kept same for every optimizer to focus on the actual performance of each.

B. *Model Evaluation*

The model is evaluated using accuracy and loss function, briefly structured below.

Algorithm	Accuracy	Loss Function
SGD	95.61%	0.1669
Adam	97.37%	0.1273
Adagrad	70.18%	0.6419
RMSProp	95.61%	0.1276
Nadam	95.61%	0.1389

Table 1: Evaluation of the algorithms

The evaluation can be better understood by having a look at the gradual increase through the epochs as stated below.

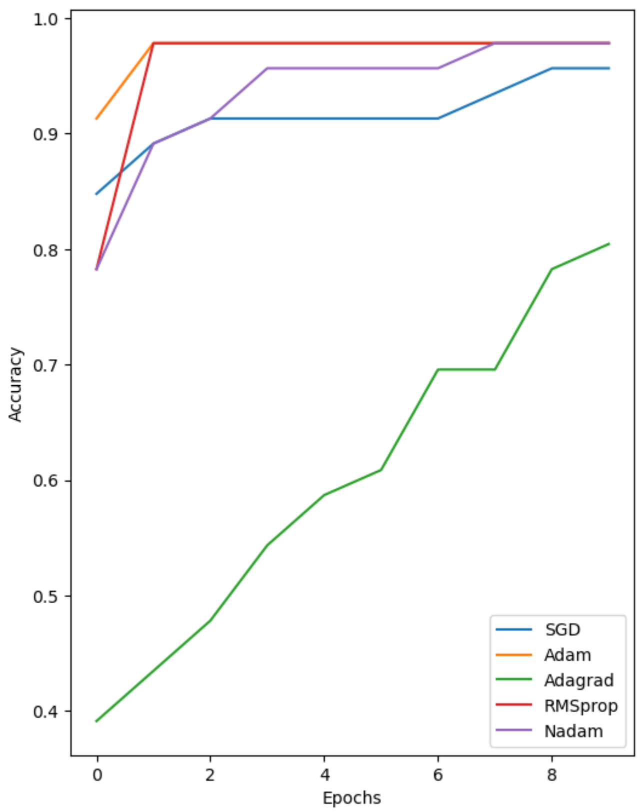

Fig 1: The behavior of the optimizers during training, as shown in accuracy

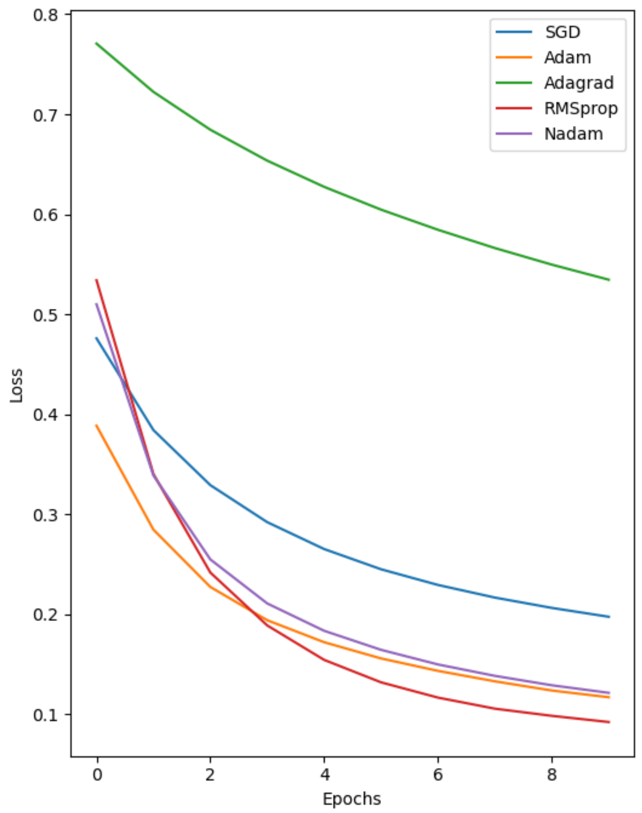

Fig 2: The behavior of the optimizers during training, as shown in loss function

From the above table and figures, The Adam optimizer is clearly the most powerful optimizer in this study with the highest accuracy of 97.37% and the lowest loss function of 0.1273. SGD, RMSProp, Nadam are almost head-to-head in effectiveness with the same accuracy of 95.61%, having loss function of 0.1669, 0.1276, 0.1389 respectively. Among them, Adagrad performed the worst with an accuracy of only 70.18% with a loss function of 0.6419. Not defining a learning rate and absence of hyperparameter tuning can be culpable for such low accuracy.

V. Conclusion

In this study, most commonly used optimizers have been examined. The differences between their working structures, along with robustness have been briefly summarized. But few uncommon optimization algorithms have been left out. Also, working with only a single ANN model, ignoring the other neural networks models like CNN, RNN is also a culpable notation of this study. Also, other important evaluation metrics like F-1 score, recall and precision were also not used. These gaps create an ample opportunity for further intensive study. So, the research for finding better adaptive algorithms for deep learning will continue.

References

[1] Oyelade, O., Isewon, I., & Oladipupo, F. (2018). A Comparative Study of Optimization Algorithms for Deep Learning. In 2018 International Conference on Computational Science and Computational Intelligence (CSCI) (pp. 225-230). IEEE.

[2] Ruder, S. (2019). Optimization Methods for Convolutional Neural Networks. arXiv preprint arXiv:1912.08957.

[3] Luo, L., Xiong, Y., Liu, Y., & Sun, X. (2019). On the Variance of the Adaptive Learning Rate and Beyond. In Proceedings of the 36th International Conference on Machine Learning (ICML) (Vol. 97, pp. 4036-4045).

[4] Kaur, A., & Singh, A. K. (2020). A Comparative Analysis of Optimization Algorithms for Deep Neural Networks. In Proceedings of the 2020 5th International Conference on Computing, Communication and Security (ICCCS) (pp. 1-6). IEEE.