Research Article
Journal of Intelligent Systems: Theory and Applications 5(2) (2022) 168-177
DOI: 10.38016/jista.1153108

Auxiliary Learning of Non-Monotonic Hyperparameter Scheduling System Via Grid Search

Alaa Ali Hameed[1*]

[1] Istinye University, Computer Engineering, Istanbul, Türkiye

alaa.hameed@istinye.edu.tr

Abstract
Recent advancements in advanced neural networks have given rise to new adaptive learning strategies. Conventional learning strategies suffer from many issues, such as slow convergence and lack of robustness. To fully exploit its potential, these issues must be resolved. Both issues are related to the step-size, and momentum term, which is generally fixed and remains uniform for all weights associated with each network layer. In this study, the recently published Back-Propagation Algorithm with Variable Adaptive Momentum (BPVAM) algorithm has been proposed to overcome these issues and improve effectiveness for classification. The study was conducted on various hyperparameters based on the grid search approach, then the optimal values of hyperparameters have trained these algorithms. Six cases were considered with varying values of the hyperparameter to evaluate the impact of the hyperparameter on the training models. It is empirically proven that the convergence behavior of the model is improved in terms of the mean and standard deviation for accuracy and the sum of squared error (SSE). A comprehensive set of experiments indicated that the BPVAM is a robust and highly efficient algorithm.

Keywords: Adaptive neural networks; Hyperparameter; Steady-state error; Optimization.

Grid Arama Yoluyla Monotonik Olmayan Hiperparametre Planlama Sisteminin Yardımcı Öğrenimi

Öz
Gelişmiş sinir ağlarındaki son gelişmeler, yeni uyarlanabilir öğrenme stratejilerine yol açmıştır. Geleneksel öğrenme stratejileri, yavaş yakınsama ve sağlamlık eksikliği gibi birçok sorundan muzdariptir. Potansiyelinden tam olarak yararlanmak için bu sorunların çözülmesi gerekir. Her iki konu da adım boyutu ve genellikle sabit olan ve her ağ katmanıyla ilişkili tüm ağırlıklar için tek tip kalan momentum terimi ile ilgilidir. Bu çalışmada, bu sorunların üstesinden gelmek ve sınıflandırma etkinliğini artırmak için yakın zamanda yayınlanan Değişken Uyarlanabilir Momentumlu Geri Yayılım Algoritması (BPVAM) algoritması önerilmiştir. Çalışma grid arama yaklaşımına dayalı olarak çeşitli hiperparametreler üzerinde yürütülmüş, daha sonra hiperparametrelerin optimal değerleri bu algoritmaları eğitmiştir. Hiperparametrenin eğitim modelleri üzerindeki etkisini değerlendirmek için hiperparametrenin değişen değerlerine sahip altı durum ele alındı. Modelin yakınsama davranışının, doğruluk için ortalama ve standart sapma ve karesel hatanın toplamı (SSE) açısından iyileştirildiği deneysel olarak kanıtlanmıştır. Kapsamlı bir deney seti, BPVAM'nin sağlam ve yüksek verimli bir algoritma olduğunu gösterdi.

Anahtar Kelimeler: Uyarlanabilir sinir ağları; Hiperparametre; Kararlı durum hatası; Optimizasyon.

1. Introduction

Advanced Adaptive Neural Networks (AANNs) are the latest developments for classification that have shown their effectiveness in solving different problems in various domains. For instance, AANNs are employed for pattern recognition (Jain et al., 2018) (Jain et al., 2019), object detection (Erol et al., 2018) (Rahman et al., 2020), images classification (Sharma et al., 2018) (Patel et al., 2019), medical diagnosis (Sarvamangala and Kulkarni, 2022) (Yu et al., 2021) (Houssein et al., 2021), etc (Demircan Keskin et al., 2022) (Güney et al., 2022) (Gemirter and Goularas, 2021). Recently,

* Corresponding Author.
E-mail: alaa.hameed@istinye.edu.tr

Recieved : 03 Aug 2022
Revision : 09 Sep 2022
Accepted : 12 Sep 2022

AANNs have gained more attention due to their applicability to large datasets in an efficient manner.

Machine learning models required training samples to learn the patterns in the data. The performance of the machine learning models is evaluated using a cost function. It will determine how accurately a model learns patterns from data. In addition, the model has many hyper-parameters that should be selected to minimize the cost function. The learning process is repeated over several epochs to obtain an optimal set of these parameters, generally termed learning. Therefore, the choice of the cost function is subjective as it depends on the model and the training data (Hinton et al., 2012) (Mestres et al., 2017). There are various methods that can be employed for training the neural networks, however, gradient-based methods are most commonly used due to their simplicity and efficiency. It aims to reduce the gradient of the cost function to obtain optimal weights during training (Krizhevsky et al., 2012) (Park et al., 2020). Although neural networks are prevalent, several issues must be addressed to carry out the training process smoothly (Hertel et al., 2020) (Sandha et al., 2020) (Sun et al., 2022). The most common issues include vanishing and exploding gradients (Bengio et al., 1994) (Glorot and Bengio, 2010.) and overfitting (Liu et al., 2021).

Another problem that can affect the neural network's performance is the presence of local minima. This situation may occur when training the model on a large dataset using more complex models. The gradient descent algorithm may face a gradient vanishing problem if it gets stuck in local minima. In addition, selecting an optimal learning rate is crucial for obtaining good accuracy for the model. Research has shown that too small value for the learning rate results in slow convergence of the model. In contrast, if a large value of learning rate is selected, then it may cause the model to skip the global optima (Jagtap et al., 2020) (Jin et al., 2022).

Recent research has shown that instead of using a fixed learning rate, an adaptive learning rate offers faster convergence with good accuracy (Seong et al., 2018) (Yan et al., 2020). Moreover, a large learning rate should not be used, which can lead to super-convergence and have regularizing effects (Smith and Topin, 2019).

The literature review reveals that researchers have proposed different solutions to the gradient vanishing problem (Liu et al., 2021). For instance, adding a momentum term can accelerate the weight updating processing that may help the model to push out of the local optima. The momentum term will keep changing the weights continuously with an appropriate ratio. During the training, it is possible that the derivative of the cost function produces zero value. Even in such a situation the model continues to update weights using the previous iteration's values of the cost function (Sutskever et al., 2013). It is interesting to note that during learning it is not possible to determine whether the solution obtained is optimal or reached a local. In both cases, the model will stopped as there will be no change in the parameter values over consecutive iterations. The model depends on several parameters that affect its performance. Learning rate (LR) also known as step-size is one of the crucial parameters. Fine tuning LR plays crucial role in obtaining optimal solution. Selection of a small value may allow the model to reach the optimal solution very slowly. In contrast, a large value may allow the model to reach the optimal solution faster. However, there is a trade-off between selection of a large/small value with the optimal solution. Therefore, care must be taken in selection of this crucial parameter. This problem can be solved using a scheduled rate. The most commonly used technique is to multiply the gradient with a constant during training of the model. The main issue with such technique is that the LR may not scale well during training. There are various solutions proposed to overcome this problem, such as time-based techniques where the LR is altered as the training proceeds (Li and Arora, 2019). Some other techniques, such as Adagrad and RMSProp are also proposed to solve this problem. These techniques apply adaptive optimization on the LR to adapt its value during the training (Duchi JDUCHI and Singer, 2011) (Reddi et al., 2019) (Yi et al., 2020). Some research proposed to combine both adaptive optimization adaptive LR schedules to further improve the accuracy of the model. However, these methods only apply a function in such as way that it decreases the LR as the model training proceeds. The main drawback of such techniques is that it may stuck in local minimum due to small gradient changes (Rumelhart et al., 1986) (Sohl-Dickstein et al.,2014).

Other advanced techniques to solve these bottlenecks include different activation functions (Klambauer et al., 2017) (Nair and Hinton, 2010), batch normalization (Ioffe and Szegedy, 2015), novel initialization schemes (He et al., 2015), and dropout (Srivastava et al., 2014). The main drawback of these methods is the higher computational overhead, which limits the performance improvements in terms of CPU cost, convergence rate, and optimal error.

The most common techniques for optimizing the deep neural networks (DNN) include batch gradient (BGD) and stochastic gradient descent (SGD) algorithms. BGD is usually slower and is more suitable for a small size dataset. On the other hand, SGD is faster and is more suitable to process large size data. Typically, SGD produces less reliable results which may also lead to bad convergence. In (Yang, 2021), authors proposed a new method based on the Kalman filter for better optimization of the network using adaptive filtering. The method employed the historical state of the optimization, which helped reduce the estimation variance in the SGD algorithm. This led to faster convergence and resulted in better gradient direction estimation even in the presence of noise.

Other gradient-based methods, such as adaptive gradient methods (AGMs), can also be employed to

optimize nonconvex problems in machine learning, specifically deep learning. In (Tong et al., 2022), two improvements of AGMs are proposed to enhance the model's accuracy further. It was observed that the anisotropic scale of the adaptive learning rate (A-LR) has high variations across multiple dimensions of the nonconvex optimization problem. This variation may lead to slower convergence and the model may get stuck in the local minima. The literature shows that a number of research are dedicated to improving the AGMs using A-LR. Another main bottleneck that plays vital role in obtaining the optimal accuracy is finding optimal values for its hyperparameters used in the A-LR. In some works, authors proposed adding activation functions in A-LR such as softplus function for AGM's improvement. Two such methods, namely SADAM and SAMSGRAD are also proposed to improve the model accuracy. Results showed that SAMSGRAD exhibit faster convergence than the AMSGRAD under various conditions such as nonconvex, non-strongly convex, and Polyak-Łojasiewicz conditions.

Another adaptive gradient descent algorithm that is commonly used in backpropagation (BP) for training feed-forward neural networks (FFNNs) is called Adam. The Adam algorithm's main issue is that it might fail to reach global optima. Solutions based on metaheuristic methods exist, which help train FFNNs to overcome the local minima issue. However, the solutions also have compromise on the convergence efficiency of the model compared to the Adam optimizer. A solution was proposed in terms of an ensemble of differential evolution and Adam (EDEAdam), combining both Adam optimizer and differential evolution algorithm, which forms a robust and efficient search mechanism to achieve better results in both global and local search. The integration of these two methods not only helped improve results but also showed faster convergence speed (Xue et al., 2022).

Hameed et al. (Hameed et al., 2016), proposed a BP algorithm with variable adaptive momentum (BPVAM). The algorithm improves the convergence behavior by achieving faster convergence, optimal error, and lower mathematical complexity, reducing the overall CPU cost and processing time. The learning rate is a crucial parameter that controls the model. The learning rate parameter depends on the input data's eigenvalues of the autocorrelation matrix.

This study investigates the learning performance of BPVAM algorithm. An adaptive momentum scheduler is introduced to overcome the gradient vanishing problem. A detailed set of experiments are performed on various benchmark datasets to evaluate the performance of the proposed model. The main contributions of this study are highlighted as follows:

- Introduction of a variable adaptive momentum term in the weight update equation.
- Fine-tuning the hyperparameters for computing an optimal momentum in stochastic gradient descent in BPVAM algorithm
- Investigate the model's behavior with adaptive momentum term and compare it with models with a fixed learning rate.
- Diverse set of experiments on different benchmark datasets are performed to test the efficiency and robustness of the proposed model.

The paper is organized as follows. In Section 2, details about the adaptive learning rate algorithms are presented. Extensive set of experiments are presented in Section 3. Finally, the paper is completed with conclusion.

2. Backpropagation Algorithm with Variable Adaptive Momentum (BPVAM)

Hameed et al (Hameed et al., 2016), introduced the BPVAM algorithm see fig. 1, where α (the adaptive momentum) is controlled by the learning rate parameter η. In this case, if given initial weights Ψ^0 and Ψ^1, and a momentum factor $\alpha\ \epsilon(0,1)$, BPVAM updates the weight vector iteratively which means that equation (28) can now be represented as

$$\Delta\Psi_i = \eta\delta_w\Psi_i + \alpha_\Psi^i\Delta\Psi_{i-1},\ i = 1, 2, ..., \quad (1)$$

Where $\eta > 0$ is the learning rate which is assumed to be a constant in this work and $\alpha_\Psi^i = (\alpha_{\Psi_0}^i, \alpha_{\Psi_1}^i, \alpha_{\Psi_2}^i, \dots\dots\dots, \alpha_{\Psi_q}^i)$ is the momentum coefficient vector at the i^{th} training iteration which is constituted by the coefficient α_Ψ^i for every $\Delta\Psi_i^i$ (i= 0, 1, 2, ….,q) and for each α_Ψ^i, it is adjusted after each training epoch by

$$\alpha_{\Psi_i}^i = \begin{cases} \alpha \cdot \frac{-\eta\delta_{\Psi_i}\Psi_i \cdot \Delta\Psi_i^{i-1}}{\left\|\Delta\Psi_i^{i-1}\right\|^2} & if\ \delta_{\Psi_i}\Psi_i \cdot \Delta\Psi_i^{i-1} < 0 \\ 0 & otherwise \end{cases} \quad (2)$$

In the BPAM, α (the adaptive momentum) was controlled by the learning rate η, where η is dependent on the eigenvalues of the autocorrelation matrix of the input.

The work presented by (Hameed et al., 2016) estimates the autocorrelation matrix *R(i)* of the input recursively as

$$R_i = \beta R_i + Rxx \quad (3)$$

Where β is the forgotten factor (0<<β<1), and $Rxx = E\{X(i)X^T(i)\}$, E is the expectation operator. Tacking the expected value of both sides of equation (32) produces

$$\bar{R}_i = \frac{1-\beta^i}{1-\beta}Rxx \quad (4)$$

Where $\bar{R}_i = E\{R_i\}$. Solving equation (32) in the steady state ($i \to \infty$) yields

$$\bar{R}_i = \frac{1}{1-\beta} \tag{5}$$

In this case, equation (34) implies that the eigenvalues of the estimated autocorrelation matrix increase exponentially, and in the limit they become $\frac{1}{1-\beta}$ times the original value.

The work done by (Hameed et al., 2016) also proposed a variable momentum, which is expressed by

$$\alpha_i = \frac{\lambda}{1-\beta^i} \tag{6}$$

Where $\lambda < \frac{2-2\beta}{max\ eigen\ value\ of \mathbf{R_{xx}}}$ and this case β is the forgetting factor ($0 \ll \beta < 1$),

Assuming that β is large, this will force the term $1 - \beta^i$ to reach unity, and assuming that the initial $\alpha(i)$ is relatively large, to provide fast convergence of the weights. By updating equations (27) and (28), with time it becomes very close to λ (a small positive constant) hence it provides law error, equation (27) and (28) can then be represented as

$$\Delta\Psi_{ji}(i+1) = \eta\delta_y \mathbf{x}_i(i) + \left(\frac{\lambda}{1-\beta^i}\right)\Delta\Psi_{ji}(i) \tag{7}$$

$$\Delta\Psi_{kj}(i+1) = \eta\delta_o y_i(i) + \left(\frac{\lambda}{1-\beta^i}\right)\Delta\Psi_{kj}(i),\ i = 0,1,\ldots \tag{8}$$

Where i represents the number of iterations and $\Delta\Psi$ is defined as updating the weights.

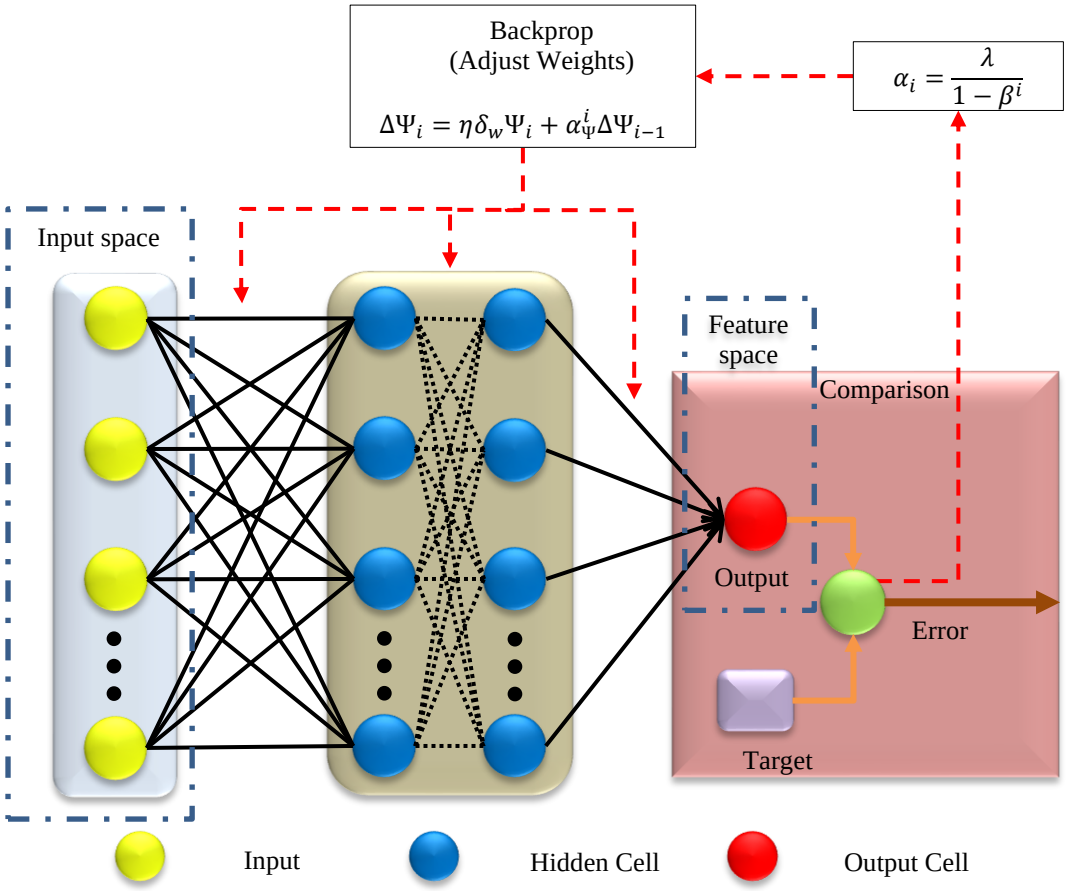


Figure 1. BPVAM architecture

3. Experimental Results

The experiments were performed on four different data sets obtained from various domains. A diverse set of datasets were considered for testing the application of the proposed method for different types of data. These datasets include Breast cancer, Heart Disease, Lung Cancer, and Iris. Each dataset has a varying number of samples, attributes, and classes (Asuncion and Newman,2007).

3.1 Preprocessing and Experimental Setup

All data in the dataset was normalized between 0 and 1 using the Min-Max normalization method. The main advantage of the normalization is maintaining stability in the network by allowing all the weights to converge almost simultaneously. Moreover, missing data were replaced with the mean value of the attribute.

All the experiments were performed in the Matlab™ environment. The models were executed on a Dell machine with Intel core i7, 2.10 GHz processor with 16 GB of RAM and NVIDIA™ GeForce TMGTX 1080. The dataset was divided into training (70%) and testing (30%) for each experiment. Since the dataset was balanced, therefore, no augmentation was performed.

3.2 Evaluation

The performance evaluation of BPVAM and its comparison with conventional BP was carried out in terms of accuracy and SSE on four benchmark datasets. Moreover, the models were also compared in terms of mean and standard deviation behaviors over the whole training process. Since the models depend on various hyperparameters, therefore, the optimal values of these hyperparameters were obtained using the Grid Search algorithm. The obtained optimal values of hyperparameters were then used to train the models. Six different cases were considered with varying values of the hyperparameter to evaluate the impact of the hyperparameter on the model accuracy. The experimental setup was similar to the one presented by the authors in (Hameed et al., 2016). The evaluation results for each dataset are described in detail as below.

Table 1 summarizes the results obtained on the **breast cancer** dataset. As it can be seen, the error convergence for the BPVAM (4.678) is better than the conventional BP (4.707) algorithm in terms of SSE for case 6. For other cases (1-5), the performance of BPVAM was also higher than conventional BP as it produced less error. Similarly, in terms of accuracy, the BPVAM algorithm produced higher accuracy compared BP in general overall six cases. It is observed that BPVAM obtained optimal results with $\eta = 0.9$, $\lambda = 0.0085$, and $\beta = 0.992$, whereas conventional BP produced best results with $\alpha = 0.01$, and $\eta = 0.9$.

Table 2 summarizes the evaluation results obtained **heart disease** dataset. These results showed that the BPVAM always produced better results than the conventional BP algorithm. Highest accuracy (61.96%) was obtained for BPVAM and lowest error (3.961) with parameter values of η = 0.03, λ = 0.022, β = 0.995. It is interesting to note that the accuracy of models tend to become close to each other as the parameter values were decreased from case-1 to case-6.

Table 3 shows the experimental results obtained by the models on the **Lung Cancer** dataset. The SSE and accuracy of BPVAM were BP 0.0054 and 60.00%, respectively. Similarly, for BP the SSE and accuracy remained 0.0063 and 60.00, respectively. The cases show that the convergence behavior of BP is very slow and very sensitive to the hyperparameter selection compared to BPVAM. The best results were obtained for BPVAM with parameters $\eta = 0.1$, $\lambda = 0.005$, and $\beta = 0.9980$. For BP BP optimal results were obtained with $\alpha = 0.05$, and $\eta = 0.1$.

Table 4 summarizes the results obtained on the **Iris** dataset. The results show that case 1 produced the optimal results for BPVAM with an accuracy of 84.44% and SSE of 0.853, while for BP the accuracy was 77.78% and SSE of 0.991 for BP. Following all cases from 1 to 6, it shows the BPVAM is more robust and can keep improving the network model steadily.

Further experiments were performed to compare the performance of the two models in terms of mean and standard deviation. Figure 2 and 3 shows the comparison of models in terms of the mean and standard deviation obtained for accuracy and SSE, respectively. It is evident that despite the improvement of the BP algorithm, the significant change indicates the sensitivity of the algorithm to its selection of parameters. On the other hand, the BPVAM algorithm shows its superiority from the first case until the sixth case. It increases the mean accuracy of the model while decreasing the standard deviation over all cases. We can deduce that the overall BPVAM model outperformed BP in terms of accuracy and SSE.

Table 1. Performance comparison metrics of the tested algorithms for Breast Cancer dataset

Case	Algorithm	α	η	λ	β	SSE	Accuracy (%)
1	BP	0.06	0.4	-	-	7.244	60.34
	BPVAM	-	0.4	0.0090	0.997	6.873	63.79
2	BP	0.05	0.5	-	-	5.691	67.24
	BPVAM	-	0.5	0.0089	0.996	5.685	67.24
3	BP	0.04	0.6	-	-	5.656	68.97
	BPVAM	-	0.6	0.0088	0.995	5.053	70.69
4	BP	0.03	0.7	-	-	4.924	72.41
	BPVAM	-	0.7	0.0087	0.994	4.787	75.86
5	BP	0.02	0.8	-	-	4.801	74.14
	BPVAM	-	0.8	0.0086	0.993	4.780	75.86
6	BP	0.01	0.9	-	-	4.707	77.59
	BPVAM	-	0.9	0.0085	0.992	4.678	77.59

Table 2. Performance comparison metrics of the tested algorithms for Heart Disease dataset

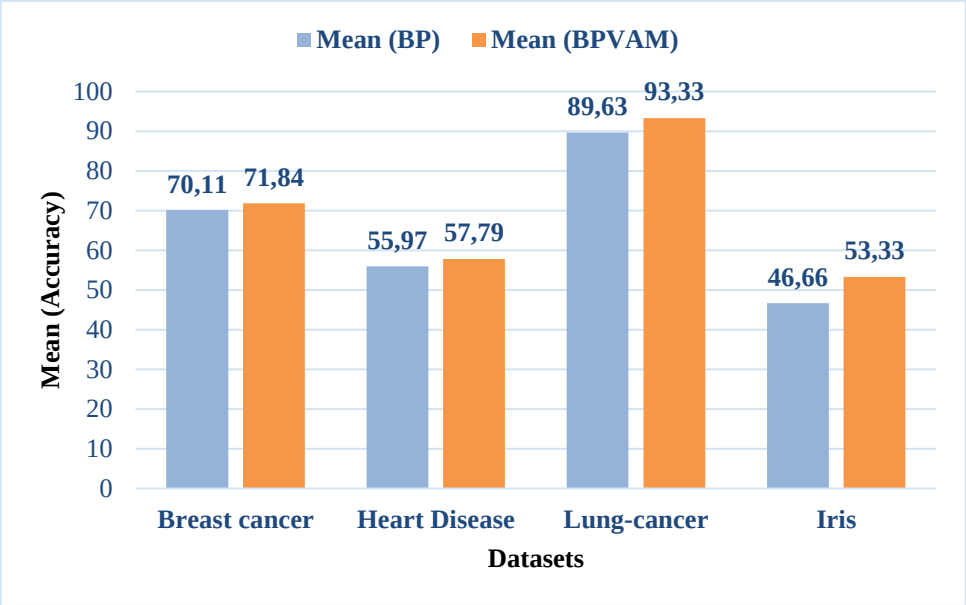
Case	Algorithm	α	η	λ	β	Training Cost	Accuracy Performance
1	BP	0.06	0.08	-	-	5.306	48.91
	BPVAM	-	0.08	0.027	0.999	4.900	51.09
2	BP	0.05	0.07	-	-	4.648	52.17
	BPVAM	-	0.07	0.026	0.999	4.615	54.35
3	BP	0.04	0.06	-	-	4.586	55.43
	BPVAM	-	0.06	0.025	0.998	4.022	58.70
4	BP	0.03	0.05	-	-	4.332	57.61
	BPVAM	-	0.05	0.024	0.997	4.017	59.78
5	BP	0.02	0.04	-	-	4.020	59.78
	BPVAM	-	0.04	0.023	0.996	4.001	60.87
6	BP	0.01	0.03	-	-	3.969	61.96
	BPVAM	-	0.03	0.022	0.995	3.961	61.96

Table 3. Performance comparison metrics of the tested algorithms for Lung-Cancer dataset

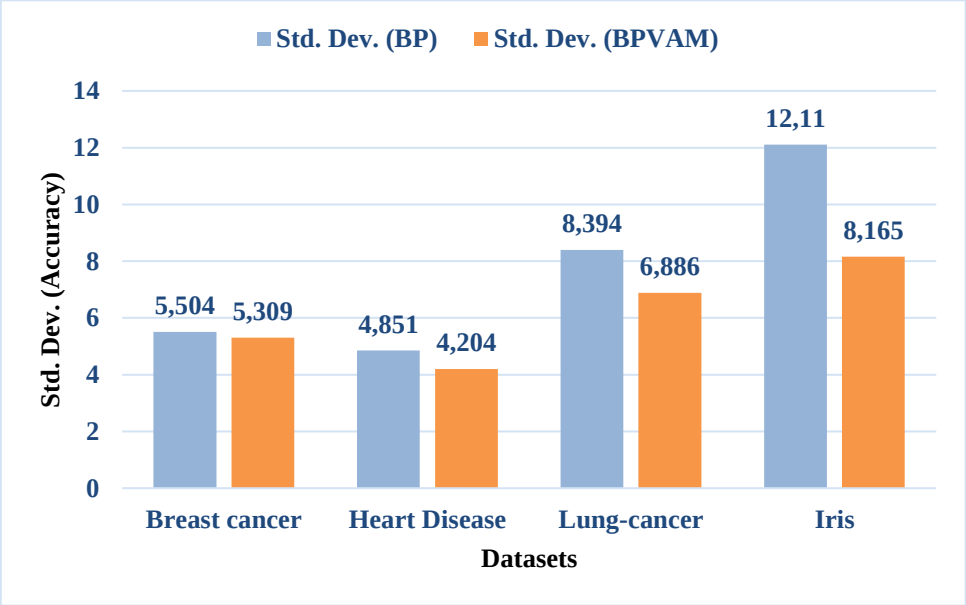
Case	Algorithm	α	η	λ	β	Training Cost	Accuracy Performance
1	BP	0.10	0.6	-	-	0.0275	30.00
	BPVAM	-	0.6	0.010	0.9994	0.0161	40.00
2	BP	0.09	0.5	-	-	0.0163	40.00
	BPVAM	-	0.5	0.009	0.9993	0.0081	50.00
3	BP	0.08	0.4	-	-	0.0120	40.00
	BPVAM	-	0.4	0.008	0.9992	0.0075	50.00
4	BP	0.07	0.3	-	-	0.0081	50.00
	BPVAM	-	0.3	0.007	0.9991	0.0066	60.00
5	BP	0.06	0.2	-	-	0.0072	60.00
	BPVAM	-	0.2	0.006	0.9990	0.0060	60.00
6	BP	0.05	0.1	-	-	0.0063	60.00
	BPVAM	-	0.1	0.005	0.9980	0.0054	60.00

Table 4. Performance comparison metrics of the tested algorithms for Iris dataset

Case	Algorithm	α	η	λ	β	Training Cost	Accuracy Performance
1	BP	0.006	0.10	-	-	0.991	77.78
	BPVAM	-	0.10	0.07	0.9994	0.853	84.44
2	BP	0.005	0.09	-	-	0.926	82.22
	BPVAM	-	0.09	0.06	0.9993	0.812	86.67
3	BP	0.004	0.08	-	-	0.756	88.89
	BPVAM	-	0.08	0.05	0.9992	0.618	91.11
4	BP	0.003	0.07	-	-	0.516	93.33
	BPVAM	-	0.07	0.04	0.9991	0.201	97.78
5	BP	0.002	0.06	-	-	0.334	95.56
	BPVAM	-	0.06	0.03	0.9990	0.182	100.00
6	BP	0.001	0.05	-	-	0.184	100.00
	BPVAM	-	0.05	0.02	0.9980	0.180	100.00

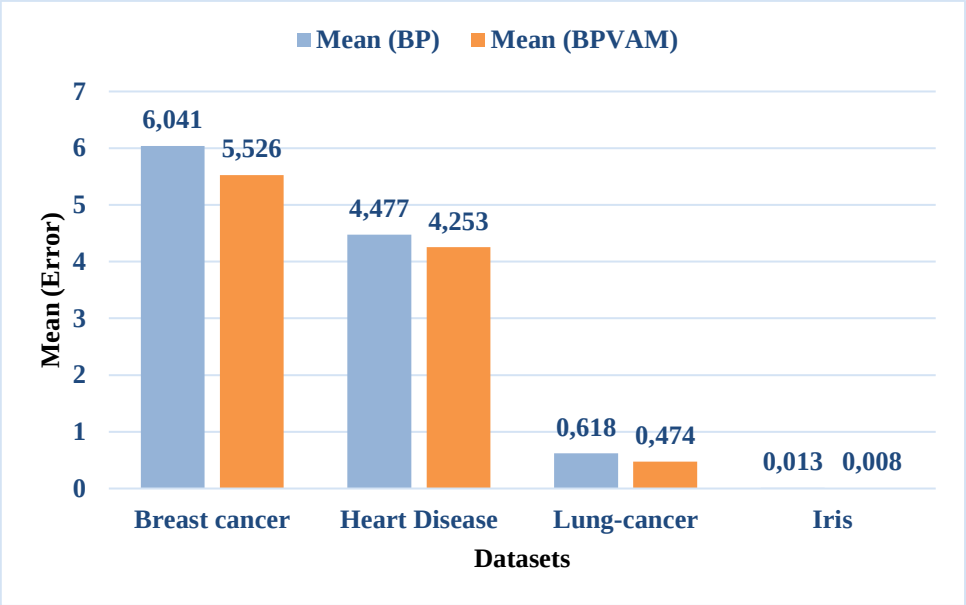


(a)

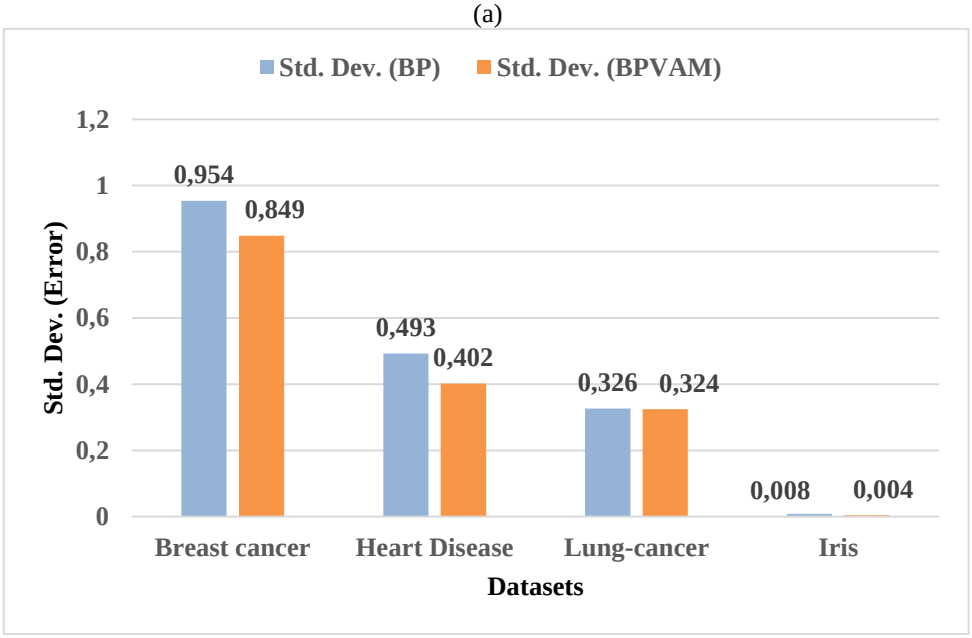


(b)

Figure 2. Performance evaluation metrics of BP and BPVAM for four benchmarks, a) mean accuracy, and b) standard deviation accuracy



(a)



(b)

Figure 3. Performance evaluation metrics of BP and BPVAM for four benchmarks, a) mean error, and b) standard deviation error

4. Conclusions

This study investigated the approach for obtaining an optimal set of hyperparameters for the machine learning model. Moreover, the model's weight matrix is updated using the adaptive momentum to help it overcome the local optima problem. The algorithm is controlled by different hyperparameters, which are fine-tuned using grid search. The results showed that the BPVAM algorithm obtains better convergence behavior than BP in the optimal steady-state models. The experiments investigated the compared methods from different aspects by considering the whole learning behavior in different training cases. The optimal results obtained on four benchmark datasets indicate that BPVAM improved the accuracy and robustness of the model. Moreover, this study suggests a significant improvement in accuracy, mean error, and standard deviation when the BPVAM is optimized with adaptive momentum. It can be observed that BPVAM exhibit features to guarantee its convergence and produce a much lower SSE against any valid data sets. In the future, we aim to apply this optimization algorithm to obtain an optimal set of parameters for a deep end-to-end neural network to overcome the issue of obtaining the optimal

hyperparameters, we also are plan to monitor the progress of the hyperparameter optimization in real-time. This will allow the extraction of highly discriminative features from input data that can improve the model's performance.

References

A. and Newman, D. J. (2007). UCI Machine Learning Repository, Department of Information and Computer Sciences, University of California, Irvine. Available at www.ics.uci.edu/~mlearn/MLRepository.html.

Bengio, Y., Simard, P., Frasconi, P., 1994. Learning Long-Term Dependencies with Gradient Descent is Difficult. IEEE Transactions on Neural Networks 5, 157–166. https://doi.org/10.1109/72.279181

Demircan Keskin, F., Çiçekli, U., İçli, D., 2022. Prediction of Failure Categories in Plastic Extrusion Process with Deep Learning. Journal of Intelligent Systems: Theory and Applications, 5(1), 27–34. https://doi.org/10.38016/jista.878854

Duchi, J., Hazan, E., Singer, Y., 2011. Adaptive subgradient methods for online learning and stochastic optimization. Journal of machine learning research, 12(7).

Erol, B.A., Majumdar, A., Lwowski, J., Benavidez, P., Rad, P., Jamshidi, M., 2018. Improved deep neural network object tracking system for applications in home robotics, in: Studies in Computational Intelligence. Springer Verlag, pp. 369–395. https://doi.org/10.1007/978-3-319-89629-8_14

Gemirter, C. B., Goularas, D., 2021. A Turkish Question Answering System Based on Deep Learning Neural Networks. Journal of Intelligent Systems: Theory and Applications, 4(2), 65–75. https://doi.org/10.38016/jista.815823

Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the thirteenth international conference on artificial intelligence and statistics, pp. 249-256.

Güney, E., Çakmak, O., Kocaman, Ç., 2022. Classification of Stockwell Transform Based Power Quality Disturbance with Support Vector Machine and Artificial Neural Networks. Journal of Intelligent Systems: Theory and Applications, 5(1), 75–84. https://doi.org/10.38016/jista.996541

Hameed, A.A., Karlik, B., Salman, M.S., 2016. Back-propagation algorithm with variable adaptive momentum. Knowledge-Based Systems 114, 79–87. https://doi.org/10.1016/j.knosys.2016.10.001

He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE international conference on computer vision, pp. 1026-1034.

Hertel, L., Collado, J., Sadowski, P., Ott, J., Baldi, P., 2020. Sherpa: Robust hyperparameter optimization for machine learning. SoftwareX, 12, 100591.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. R., 2012. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580.

Houssein, E.H., Emam, M.M., Ali, A.A., Suganthan, P.N., 2021. Deep and machine learning techniques for medical imaging-based breast cancer: A comprehensive review. Expert Systems with Applications, 167. https://doi.org/10.1016/j.eswa.2020.114161

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning, pp. 448-456.

Jagtap, A.D., Kawaguchi, K., Karniadakis, G.E., 2020. Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. Journal of Computational Physics 404. https://doi.org/10.1016/j.jcp.2019.109136

Jain, D.K., Shamsolmoali, P., Sehdev, P., 2019. Extended deep neural network for facial emotion recognition. Pattern Recognition Letters 120, 69–74. https://doi.org/10.1016/j.patrec.2019.01.008

Jain, N., Kumar, S., Kumar, A., Shamsolmoali, P., Zareapoor, M., 2018. Hybrid deep neural networks for face emotion recognition. Pattern Recognition Letters 115, 101–106. https://doi.org/10.1016/j.patrec.2018.04.010

Jin, J., Zhu, J., Gong, J., Chen, W., 2022. Novel activation functions-based ZNN models for fixed-time solving dynamirc Sylvester equation. Neural Computing and Applications, 1-19. https://doi.org/10.1007/s00521-022-06905-2

Klambauer, G., Unterthiner, T., Mayr, A., Hochreiter, S., 2017. Self-Normalizing Neural Networks, Advances in neural information processing systems, 30.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet Classification with Deep Convolutional Neural Networks. Advances in neural information processing systems, 25.

Li, Z., Arora, S., 2019. An Exponential Learning Rate Schedule for Deep Learning. arXiv preprint arXiv:1910.07454.

Liu, M., Chen, L., Du, X., Jin, L., Shang, M., 2021. Activated Gradients for Deep Neural Networks. IEEE Transactions on Neural Networks and Learning Systems 1–13. https://doi.org/10.1109/tnnls.2021.3106044

Mestres, A., Rodriguez-Natal, A., Carner, J., Barlet-Ros, P., Alarcón, E., Solé, M., Muntés-Mulero, V., Meyer, D., Barkai, S., Hibbett, M.J., Estrada, G., Ma'ruf, K., Coras, F., Ermagan, V., Latapie, H., Cassar, C., Evans, J., Maino, F., Walrand, J., Cabellos, A., 2017. Knowledge-defined networking. Computer Communication Review 47, 1–10. https://doi.org/10.1145/3138808.3138810

Nair, V., Hinton, G. E., 2010. Rectified linear units improve restricted boltzmann machines. In Appearing in Proceedings of the 27 th International Conference on Machine Learning (ICML).

Park, J., Yi, D., Ji, S., 2020. A novel learning rate schedule in optimization for neural networks and it's convergence. Symmetry (Basel) 12. https://doi.org/10.3390/SYM12040660

Patel, K., Rambach, K., Visentin, T., Rusev, D., Pfeiffer, M., Yang, B., 2019. Deep learning-based object classification on automotive radar spectra, in: 2019 IEEE Radar Conference, RadarConf 2019. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/RADAR.2019.8835775

Rahman, M.M., Tan, Y., Xue, J., Lu, K., 2020. Notice of Removal: Recent Advances in 3D Object Detection in the Era of Deep Neural Networks: A Survey. IEEE Transactions on Image Processing. https://doi.org/10.1109/TIP.2019.2955239

Reddi, S.J., Kale, S., Kumar, S., 2019. On the Convergence of Adam and Beyond. arXiv preprint arXiv:1904.09237.

Rumelhart, D. E., Hinton, G. E., Williams, R. J., 1986. Learning representations by back-propagating errors. nature, 323(6088), 533-536.

Sandha, S. S., Aggarwal, M., Fedorov, I., Srivastava, M. 2020. Mango: A python library for parallel hyperparameter tuning. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3987-3991.

Sarvamangala, D.R., Kulkarni, R. v., 2022. Convolutional neural networks in medical image understanding: a survey. Evolutionary Intelligence, 1-22. https://doi.org/10.1007/s12065-020-00540-3

Seong, S., Lee, Y., Kee, Y., Han, D., Kim, J., 2018. Towards Flatter Loss Surface via Nonmonotonic Learning Rate Scheduling, In UAI.

Sharma, N., Jain, V., Mishra, A., 2018. An Analysis of Convolutional Neural Networks for Image Classification, in: Procedia Computer Science. Elsevier B.V., pp. 377–384. https://doi.org/10.1016/j.procs.2018.05.198

Smith, L.N., Topin, N., 2019. Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates. In Artificial intelligence and machine learning for multi-domain operations applications, Vol. 11006, pp. 369-386.

Sohl-Dickstein, J., Poole, B., Ganguli, S., 2014. Fast large-scale optimization by unifying stochastic gradient and quasi-Newton methods. In International Conference on Machine Learning, pp. 604-612.

Srivastava, N., Hinton, G., Krizhevsky, A., Salakhutdinov, R., 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Journal of Machine Learning Research. 15(1), 1929-1958.

Sun, J., Yang, Y., Xun, G., Zhang, A., 2022. Scheduling Hyperparameters to Improve Generalization: From Centralized SGD to Asynchronous SGD. ACM Transactions on Knowledge Discovery from Data (TKDD). https://dl.acm.org/doi/pdf/10.1145/3544782.

Sutskever, I., Martens, J., Dahl, G., Hinton, G., 2013. On the importance of initialization and momentum in deep learning. In International conference on machine learning, pp. 1139-1147.

Tong, Q., Liang, G., Bi, J., 2022. Calibrating the adaptive learning rate to improve convergence of ADAM. Neurocomputing 481, 333–356. https://doi.org/10.1016/j.neucom.2022.01.014

Xue, Y., Tong, Y., Neri, F., 2022. An ensemble of differential evolution and Adam for training feed-forward neural networks. Information Sciences. Information Sciences, 608, 453-471. https://doi.org/10.1016/j.ins.2022.06.036

Yan, Z., Chen, J., Hu, R., Huang, T., Chen, Y., Wen, S., 2020. Training memristor-based multilayer neuromorphic networks with SGD, momentum and adaptive learning rates. Neural Networks 128, 142–149. https://doi.org/10.1016/j.neunet.2020.04.025

Yang, X., 2021. Kalman optimizer for consistent gradient descent, in: ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. Institute of Electrical and Electronics Engineers Inc., pp. 3900–3904. https://doi.org/10.1109/ICASSP39728.2021.9414588

YD., Ahn, J., Ji, S., 2020. An effective optimization method for machine learning based on ADAM. Applied Sciences (Switzerland) 10. https://doi.org/10.3390/app10031073

YH., Yang, L.T., Zhang, Q., Armstrong, D., Deen, M.J., 2021. Convolutional neural networks for medical image analysis: State-of-the-art, comparisons, improvement and perspectives. Neurocomputing 444, 92–110. https://doi.org/10.1016/j.neucom.2020.04.157