Supplementary Material of the Manuscript "Wave-RVFL: A Randomized Neural Network Based on Wave Loss Function"

M. Sajid, A. Quadir, and M. Tanveer

Indian Institute of Technology Indore, Simrol, Indore, India
{phd2101241003,mscphd2207141002,mtanveer}@iiti.ac.in

S.I Square Loss Function

The square loss function [3], widely utilized in machine learning, calculates the error by squaring the difference between the actual value y_i and the predicted value $f(x_i)$ for a given sample x_i, where $i = 1, 2, \dots, n$. The square loss function is depicted in Figure 1 and is defined as follows:

$$\mathcal{L}(y_i, f(x_i)) = \frac{1}{n}\sum_{i=1}^{n}\xi_i^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2, \tag{1}$$

where $\xi_i = y_i - f(x_i)$.

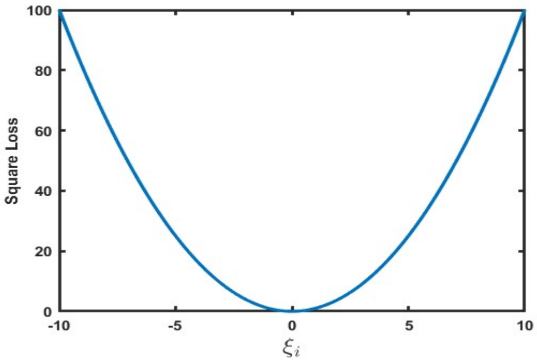

Fig. S.1: Visual depiction of squared loss function

The square loss function has the following drawbacks [3]:

a) The square error loss function gives higher weight to larger errors due to squaring. This means that outliers (data points that are far from the model's prediction) can disproportionately influence the loss, leading to suboptimal models.

b) The square error loss function is scale-dependent, meaning it is sensitive to the magnitude of the errors. If the errors are large, the squared errors can become disproportionately large, impacting the training process and model performance.
c) During optimization, the gradient of the squared error loss function can become very large for predictions that are far from the actual values. This can lead to unstable updates during gradient descent, causing the optimization process to become erratic or to diverge.

S.II Computational Complexity

The time complexity of the proposed Wave-RVFL model is influenced by two main components: (i) hidden layer generation and (ii) the application of the Adam algorithm for calculating β. According to [2], the complexity of generating the hidden layer is $\mathcal{O}(nmN)$, where n is the number of training samples, m is the number of features, and N represents the number of hidden layer nodes. In line with [1], the Adam algorithm's computational complexity primarily arises from computing the moving averages of the gradients' first and second moments, along with bias correction for these averages. This step requires a complexity of $\mathcal{O}(s)$, where s is the number of samples selected per iteration. Therefore, the total time complexity of the proposed Wave-RVFL model is $\mathcal{O}(Itr(nmN+s))$, where Itr represents the number of iterations used in the Adam algorithm.

S.III Experimental Setup

The experimental hardware configuration includes a personal computer featuring an Intel(R) Xeon(R) Gold 6226R CPU with a clock speed of 2.90 GHz and 128 GB of RAM. The system runs on Windows 11 and utilizes Matlab2023a to run all the experiments. Following the experimental setup of [2], we find the best hyperparamter setting and testing accuracy by employing grid search and five fold cross validation technique. Furthermore, all the hyperparameters of the baseline models are tune using the experimental setup followed in [2]. For all the models (baseline and proposed), following [2], we tune 6 activation functions: 1 denotes *Sigmoid*, 2 denotes *Sine*, 3 denotes *Tribas*, 4 denotes *Radbas*, 5 denotes *Tansig* and 6 denotes *Relu*. The regularization parameter $\mathcal{C}$ is taken from $\{10^{-5}, 10^{-4}, \ldots, 10^{5}\}$. The number of hidden nodes (N) is selected from a range spanning from 3 to 203, with a step size of 20. wave loss parameters are selected from the following range $\eta = [0.1 : 0.25 : 2]$ and $\gamma = [-2 : 0.5 : 5]$. The Adam algorithm is initialized with the following parameters: starting weights $\beta_0 = 0.01$, initial learning rate α selected from the set $\{0.0001, 0.001, 0.01\}$, initial first moment $g_0 = 0.01$, initial second moment $u_0 = 0.01$, first-order exponential decay rate $\lambda_1 = 0.9$, second-order exponential decay rate $\lambda_2 = 0.999$, error tolerance $\delta = 10^{-5}$, division constant $\epsilon = 10^{-8}$, maximum number of iteration $Itr = 1000$, and mini-batch size $s = 2^5$ if number of samples in the dataset is less than 500, otherwise 2^8.

Bibliography

[1] M. Akhtar, M. Tanveer, M. Arshad, and Alzheimer's Disease Neuroimaging Initiative. Advancing supervised learning with the wave loss function: A robust and smooth approach. *Pattern Recognition*, 155:110637, 2024.

[2] M. Sajid, A. K. Malik, M. Tanveer, and P. N. Suganthan. Neuro-fuzzy random vector functional link neural network for classification and regression problems. *IEEE Transactions on Fuzzy Systems*, 32(5):2738–2749, 2024.

[3] Q. Wang, Y. Ma, K. Zhao, and Y. Tian. A comprehensive survey of loss functions in machine learning. *Annals of Data Science*, 9:187–212, 2020.