Journal of Computer Science

Original Research Paper

Logistic Regression with Multi-Connected Weights

[1]Avazjon Marakhimov Rakhimovich, [2]Kabul Khudaybergenov Kadirbergenovich, [3]Zahriddin Muminov Ishkobilovich and [4]Kudaybergenov Jabbarbergen Kadirbergenovich

[1]Department of Artificial Intelligence, Nurafshon Branch, Tashkent University of Information Technologies Named After Muhammad ibn Musa al-Khwarizmi, Nurafshon, Uzbekistan
[2]Department of Applied Informatics, Kimyo International University in Tashkent, Tashkent, Uzbekistan
[3]Department of Linear Algebra, V. I. Romanovsky Institute of Mathematics of the Academy of Sciences of Uzbekistan, Tashkent, Uzbekistan
[4]Department of Software Engineering, Nukus Branch, Tashkent University of Information Technologies Named After Muhammad ibn Musa al-Khwarizmi, Nukus, Uzbekistan

Article history
Received: 03-03-2024
Revised: 25-04-2024
Accepted: 16-05-2024

Corresponding Author:
Kabul Khudaybergenov Kadirbergenovich,
Department of Artificial Intelligence, Nurafshon Branch, Tashkent University of Information Technologies Named After Muhammad ibn Musa al-Khwarizmi,
Nurafshon, Uzbekistan
Email: kabul85@mai.ru

Abstract: Results of current research in biological and cognitive science reveal that neuronal interactions predominantly depend on neurotransmitters for signaling and sending information between neurons. Furthermore, a specific neuron that is connected to the next neuron transmits or sends multiple neurotransmitters in parallel ways, where each neurotransmitter has specific functional roles. Based on these results, a new type of logistic regression model is proposed that expands the dimensionality of connection weight coefficients from one to multiple coefficients, i.e., which means there are multiple connections between each input and hidden unit, rather than a single weight coefficient for every input unit. The number of dimensions of compound weights represents the number of various neurotransmitter categories and different weight components correspond to different neurotransmitter channels. According to recent biological studies, this new type of logistic regression model is promising to be much closer to a biological neuronal model. In terms of the new model structure in logistic regression with multidimensional weights, it is modeled on multiple filters and can enhance the interpretability of the sigmoid activation function of the learning algorithm. Results from computational experiments on CIFAR-10, CDC diabetes health indicators, and other benchmark datasets have shown that the performance of the existing logistic regression model can be enhanced by expanding the dimensionality of connected weights between each input unit and hidden unit and the approach of multiple weights will provide a new design architecture of models for artificial neural network architecture.

Keywords: Logistic Regression, Logistic Function, Neuron, Neural Networks, Weight Connection, Regularization, Learning

Introduction

Motivated by the hierarchical structure of the cerebral cortex in the brain, diverse types of artificial Neural Network (NN) models have been proposed with practical applications, such as the Logistic Regression (LR) model (Wang *et al.*, 2023, Yang and Loog, 2018), multilayer perceptron (Mahmoud *et al.*, 2023), feedforward neural networks deep learning (Lara-Benítez *et al.*, 2023; Sun *et al.*, 2023; Chaitra *et al.*, 2023; Rendón-Segador *et al.*, 2023), convolutional neural networks (Huang *et al.*, 2023; Lin and Wang, 2021), fuzzy neural networks (Lolaev *et al.*, 2024; Madrakhimov *et al.*, 2021; Rakhimovich *et al.*, 2022), etc., which turned out to be very good for solving many applied problems. These problems, which are closely related to artificial intelligence, include visual object recognition and detection (Wu *et al.*, 2022), speech recognition (Vanderreydt and Demuynck, 2024), and natural language processing (Zou *et al.*, 2024), etc. All these existing NN models have the same feature: Signal flow between two connected nodes is carried out on only one connected weight. In general, all input signals x_i of the module i are multiplied by the weight parameter $w_{i,j}$ and then processed by the activation function. Further, the obtained result is transferred into the input of the module j of the following layer.

DOI: 10.3844/jcssp.2024.1051.1058

Meanwhile, biological researchers, Lauder (1993); Brian *et al.* (2014) have shown that information transfer between two connected neurons occurs by the movement of chemical elements through a tiny tube called a neurotransmitter released by a connected neuron which is located at presynaptic terminals. Then these newly generated chemical elements are received by the following neuron at receptors. These types of neurotransmitters are classified into various tiny categories: Tiny neurotransmitters and neuropeptides. Tiny neurotransmitters are different tiny organic chemicals, which include elements such as glutamates, various types of neurotransmitters, biogenic dopamine, various types of purinergic neurotransmitters, etc. These neuropeptides are composed of different amino acids and they are much larger than tiny messenger molecules. Moreover, there are also other types of neuropeptides with different functionalities (Crawley *et al.*, 1985; McDonald and Pearson, 1989). In general, a neuron transfers data to another neuron via transferring multiple neurotransmitters in parallel flow and each neurotransmitter has a different behavior and role.

Inspired by the behavior and principle of these information transfers between neurons, that neurons are connected to each other and transmit messages by sending multiple types of neurotransmitters via multiple channels, this study introduces a new model which is described further. In general, a new type of LR with Multi-Connected Weights (LRMCW) model is proposed by expanding the dimensions of connections between the input unit and hidden unit in the LR model from one connection to many. In LRMCW, various sizes of connection weights correspond to various types of neurotransmitters. In particular, weight coefficients between units in the existing LR model are considered as 1-dimensional, and weight coefficients between two units for the proposed model are considered as multi-dimensional.

The LR method plays a special role in the field of statistical data analysis and machine learning, which has the ability to predict categorical outcomes (Wichitaksorn *et al.*, 2023; Zhang *et al.*, 2019). The LR method was proposed by Cox (1958), which was originally considered a statistical approach that models the log odds of a simple event with a linear combination of independent variables of the algorithm. In machine learning, the LR method is a type of supervised learning (Martín-Baos *et al.*, 2020; Wang and Park, 2017), which learns from input and output data. This method plays a special role in binary classification problems, where it is necessary to estimate the probability of observations pertaining to a certain rank. This method is based on a linear function, similar to the linear regression method, and also with a combination of a non-linear function, offering a solution for modeling discrete outcomes (Yuan and Xu, 2023; Cheng *et al.*, 2022; Ming and Yang, 2024).

Zhang *et al.* (2020) proposed a NN model with multiple connection weights, which is slightly close to this paradigm. The results of computational experiments on training datasets have illustrated where the performance of existing NN models can be enhanced by expanding the dimension of weight connections within layers and this new approach provided a new architecture for NN models. Moreover, some applications for classification problems were demonstrated (Marakhimov and Khudaybergenov, 2022).

The significance of the proposed model compared to standard LR and other models, the proposed LRMCW model has some advantages:

- First, from a point of view of cognitive science, LRMCW considers the various types of neurotransmitters. With this assumption, the proposed LRMCW model can be considered as the much closer model to the structure of a biological NNs
- Secondly, the model structure and activation of the hidden module (sigmoid activation function) are defined by various filters (appropriate to several connected weights). For an input value, the probability of activation, the hidden unit gets maximum activation if the input values have completely all features belonging to those obtained filters appropriate to the hidden unit. Moreover, activation of the LRMCW model is rare, where the representation of every connection level of the model is more explainable

Logistic Regression Model

This section discusses the mathematical foundation of the traditional LR model. Before presenting the LRMCW formulation, let us consider the traditional LR. This requires the definition of the following functions: A logistic function or a logit function (Rigon and Aliverti, 2023) (Fig. 1). The logistic function, also called a sigmoid activation function is defined as:

$$S(x) = \frac{1}{1+e^{-x}}, \ x\hat{I}R$$

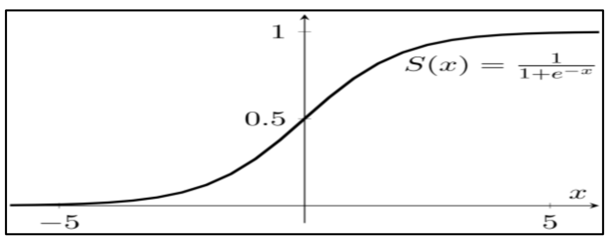


Fig. 1: Logistic function

The basis of LR is the logistic function, or sigmoid function, which converts a linear equation into a probability-a value between 0 and 1. In fact, it is a popular function in probability theory. This function is defined as:

$$P(Y=1)=\frac{1}{1+e^{-\sum_{i=1}^{n} w_i x_i + w_0}} \quad (1)$$

where, $P(Y=1)$ -probability which a dependent variable gets the value 1, $w_0, w_1, \ldots, w_n$ -model coefficients that determine the contribution of each predictor, $x_1, x_2 \ldots, x_n$ -independent variables (predictors), e -base of natural logarithm.

Further, for convenience, the following formula is used for both LR:

$$y=\sigma(z)=\frac{1}{1+e^{-z}} \quad (2)$$

where, $z=-\sum_{i=1}^{n} w_i x_i + w_0$.

The logistic function ensures that the estimated probabilities fall within the acceptable range from 0-1. LR with a single independent variable x and binary dependent variable $y \in \{0,1\}$ with a dataset (x_i, y_i) $(i=1,2,\ldots,n)$ tries to fit the logistic probability model (2).

Structure of the LRMCW Model

This section introduces the structure of the proposed LRMCW and its learning algorithm. First, let us consider the structure of the traditional LR model and the proposed LRMCW model for comparison, which are shown in Fig. 2 where for simplicity LRMCW model with 2-dimensional connection weights was modeled (Fig. 2b).

Here is a description given for the structure of the proposed LRMCW method for a binary classification problem. Let us consider LR with x_i -input and y -predicted output. The structure of the proposed LR and its single neuron, as well as the architecture of traditional LR and its uniform connection weights, are shown in Fig. 2. LR with the two-dimensional vector of connection weights is taken as an example which is illustrated in Fig. 2b.

A standard LR consists of one connection between an input unit and hidden neuron and output, which is shown in Fig. 2a, where only one connection from the input node to the hidden neuron is allowed and input connections from more than one are prohibited. In standard LR, the connection between two nodes is represented by a real value. This means that each input has its own weight coefficient. The sensor signals enter the LR method through input layers' nodes and propagate from input layers' nodes to output to obtain a prediction result (classification). In machine learning, the traditional LR can be defined as below:

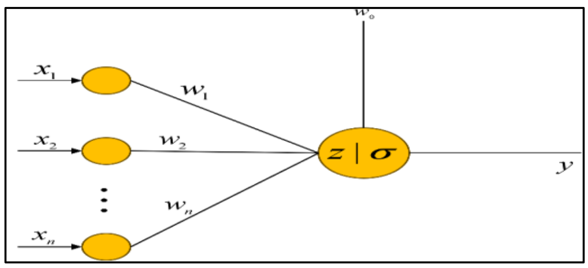
$$z=w_0+w_1x_1+\ldots+w_nx_n=\sum_{i=1}^{n} w_i x_i + w_0 \quad (3)$$

$$\sigma(z)=\frac{1}{1+e^{-z}} \quad (4)$$

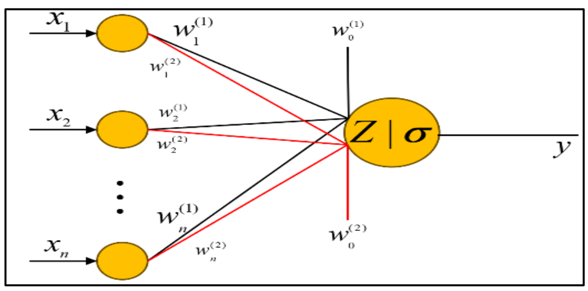
where, $x=(x_1, x_2, \ldots, x_n) \in R^n$ - input vector, $w_1, \ldots, w_n$ - weight coefficients, w_0 -threshold value, σ -nonlinear function **Error! Reference source not found.** In the standard LR, which is used in case of binary classification.

LR with Multi-Connected Weights

The LRMCW consists of multiple connections between each input unit and a hidden neuron and output (computing) block, as depicted in Fig. 2b, where multiple connection coefficients from the input unit to the hidden neuron are permitted. Then, each input unit has its own weight vector of coefficients, which allows multiple weight connections between each input unit and summation block. Similarly, sensory signals are input to the LRMCW method through input and propagated from the input layer to output to obtain a classification result. The following formulas are the formulation of the proposed LRMCW method.



(a)



(b)

Fig. 2: (a) Architecture of the traditional LR model; (b) Model of the proposed LRMCW method with 2-dimensional weights

DOI: 10.3844/jcssp.2024.1051.1058

Let us consider that there are H connections between each input and summing block. Then the following equations can be obtained:

$$z_1 = w_0^{(1)} + w_1^{(1)} x_1 + \ldots + w_n^{(1)} x_n = \sum_{i=1}^{n} w_i^{(1)} x_i + w_0^{(1)} \quad (5)$$

$$z_2 = w_0^{(2)} + w_1^{(2)} x_1 + \ldots + w_n^{(2)} x_n = \sum_{i=1}^{n} w_i^{(2)} x_i + w_0^{(2)} \quad (6)$$

$$z_H = w_0^{(H)} + w_1^{(H)} x_1 + \ldots + w_n^{(H)} x_n = \sum_{i=1}^{n} w_i^{(H)} x_i + w_0^{(H)} \quad (7)$$

Next, the input vector can be obtained as below:

$$Z = \left(z_1, \ldots, z_H\right) \quad (8)$$

Hence:

$$\sigma\left(z_1, \ldots, z_n\right) = \frac{1}{1 + \sum_{h=1}^{H} e^{-z_h}} \quad (9)$$

where, $\sigma(\cdot)$ -usually a nonlinear activation function, which is often used in NN and usually called a sigmoid activation function, $h \in \{1, \ldots, H\}$ -the number of weights between each input unit and the summation block. To illustrate the model below is a case where the number of weight coefficients between each input unit and summation block where two weight connections are taken, i.e., $H = 2$, (Fig. 3).

The encoding process in standard LR and LRMCW is very different. For instance, Let us consider LR-3-1 and LRMCW-3-H2-1 as a model sample to show the discrepancy between expanding the size of weight connections, i.e. increasing the channels for weight connections, as shown in Fig. 2b. LR-3-1 is a traditional LR model, where 3 is the dimension of input vector, one hidden block for summation and activation and one output, which can be applied in the case of binary classification. LRMCW-3-H2-1 is the proposed model, where 3 is the dimension of the input vector, one hidden block for summation and activation, and one output block, which is used in the case of binary classification. In LRMCW-3-H2-1, the dimension of connection coefficients is equal to 2 and H2 will be deciphered as $H = 2$. The total number of weights is 3×2+6 = 12, where 6 means there exist 6 connections between the input and hidden unit. For LR-3-1, sensory inputs are passed through the weight connections of the input layer to obtain the hidden layer feature (which represents the activation probability of three input units), and the summation block feature is passed through the weight connections of the summation block to obtain the output. For LRMCW-3-H2-1, the sensory inputs are passed through input layer weight connections to obtain the temporal feature of the summation block and then separated into 2 parts according to the size of weights and then they are encoded in the final feature of the summation block. The last feature of the summation block checks the weights of input layer connections to produce output. Intuitively, LRMCW-3-H2-1 extracts more abstract latent features, which is more suitable for classification compared to LR-3-1.

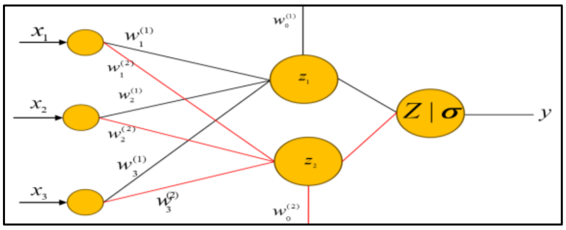


Fig. 3: Coding process LRMCW-3-H2-1

Learning Algorithm

This section describes the training algorithm of the proposed LRMCW. As an example and to simplify descriptions, the model of the above-mentioned LRMCW-3-H2-1 was chosen.

Backpropagation Algorithm for LRMCW

The proposed LRMCW with extended connectivity of weight coefficients up to $H = 2$, denoted as LRMCW-3-H2-1 can be trained using a standard learning algorithm similar to LR. Let $\{\mathrm{x}, d\}_{i=1}^{M}$ - training data sets, where $\mathrm{x} = \left(x_1, \ldots, x_n\right)^T$ - input data, n - dimension of input data and d - desired result (output), M -number of samples in the dataset. As a loss function to evaluate the accuracy, cross-entropy loss is used with the regularizing term, which is shown below:

$$E = -\sum_{j=1}^{M}\left(d_j \log\left(y_j\right) + \left(1 - d_j\right)\log\left(1 - y_j\right)\right) + \frac{\lambda}{2}\sum_{h=1}^{H}\sum_{i=1}^{n} L\left(w_i^h\right) \quad (10)$$

where:

$$L\left(w_i^h\right) = \begin{cases} \left(w_i^h\right)^2, & w_i^h < 0 \\ 0, & otherwise \end{cases} \quad (11)$$

And y -output value, λ -regularization parameter. The second part of the expression is a non-negativity constraint imposed on connection weights w_i^h, which were applied to train NN models to improve their model interpretability in some works (Chorowski and Zurada, 2015; Chen *et al.*, 1997). Typically, the concept of non-negativity values makes connection coefficients sparse, for instance, exclusively a portion of them keep non-zero and convert the activation in sigmoid activation function sparse.

By applying the backpropagation rule, the partial derivative of weights $\vartriangle w_i^h$ and biases $\vartriangle b_i^h$ can be estimated as follows:

$$\vartriangle w_i^h = \frac{\partial E}{\partial w_i^h} = y\left(y - d\right) \quad (12)$$

$$\triangle b_i^h = \frac{\partial E}{\partial b_i^h} = y - d \tag{13}$$

Further, the updating of the parameters for the LRMCW method can be carried out consequently to the gradient descent rule. The backpropagation training procedure for LRMCW is illustrated in Algorithm 1.

Algorithm 1. *Procedure for Training the backpropagation of the LRMCW model*

Input data: Training data set $\{x^{(i)}, d^{(i)}\}_{i=1}^{M}$. Parameters: Extended dimension of connection weights H, regularization parameter λ and learning rate η.

1: Randomly **initialize** weight and bias parameters: $\{w_i^h\}$, $\{b_i^h\}$, $h = 1, \ldots, H$, $i = 1, \ldots, n$.

2: For $t = 1$ to T (the number of epochs) **do**

• **For** each training sample $\mathbf{x}^{(j)}$ **do**

2.1 Forward Propagation:

Compute input sum and activation probability of output by using Eqs. **Error! Reference source not found.-Error! Reference source not found.**.

2.2 Backward Propagation:

Compute partial derivatives of multi-connected weights w_i^h, and biases b_i^h by using Eqs. **Error! Reference source not found.-Error! Reference source not found.**.

2.3 Parameter Update:

$w_i^h(t) = w_i^h(t-1) - \eta \triangle w_i^h(t-1)$

$b_i^h(t) = b_i^h(t-1) - \eta \triangle b_i^h(t-1)$

• **Endfor**

3: Endfor

Computational Experiments

The proposed method was tested on popular datasets that are publicly available at (Asuncion and Newman, 2007). Numerical experiments were obtained and learning characteristics as well as the classification characteristics for different configurations of LRMCW were displayed. Comparisons were carried out with the relevant state-of-the-art algorithms such as C4.5, CART, XgBoost, etc.

The regularization parameter λ was used from the set {1×10-3, 1×10-4, 5×10-5,..., 1×10-9} for smaller data sets and {1×10-3, 1×10-4,1×10-6} for data sets such as CIFAR-10 and CDC diabetes health indicators, based on test set performance. The learning rate parameter was used from set {0.1, 0.01, 0.005} according to the validation set performance across all datasets. For the traditional LR and proposed LRMCW, the activation function was employed in the hidden layer, and the output layer was a sigmoid function $\sigma(\cdot)$, which was chosen according to the model. For the NN model, the activation functions that were used in the hidden layer were also a sigmoid function $\sigma(\cdot)$, and in the output layer, a SoftMax function was used. The objective function for learning the NN model was the cross-entropy loss function.

First, numerical experiments are conducted to compare the classification performance of different LRMCW models with different numbers of weight connections on CIFAR-10 and CDC Diabetes Health Indicators datasets. In experiments, the number of weight connections was set to 1 and 2 for LRMCW and LR, which means that both models have a relatively equal number of trainable parameters (weights). Also, models were built and tested with a dimension of weight connections which was set equal to 2,..., 8 for LRMCW. For LRMCW, different numbers of weight connections were used. The name of the model, for example, LRMCW-1024-H2-1 means that the LRMCW model has 1024 inputs (32×32 pixel image in CIFAR-10 dataset), 2 weight connections between each input and hidden unit (neuron), respectively, their 1024 bias (threshold) coefficients and 1 output and the total number of its training parameters equals 1024×2+1024 = 3072. And LRMCW-3-H8-1 model means that the LRMCW model has 3 inputs and 8 weight connections between each input and hidden unit, 8 biases for each connection and a total number of parameters is 3x8+8=36. In the computational experiment, for comparison, the average classification results and training time were obtained for LRMCW and LR models for 5 attempts, where the optimal ones were also given for each attempt. The results of computational experiments are shown in Tables 1-2. From Table 1 it can be seen that expanding the number of connections to more than 10 for the CIFAR-10 dataset and more than 7 for the Abalone dataset in the LRMCW model makes higher classification errors and can lead to overfitting problems.

From the output results depicted in Table 2, it is evident LRMCW consistently surpassed LR on most training datasets and the proposed LRMCW model achieved better classification results on most datasets, except for CIFAR-10 and Abalone data, on which C4.5 achieved better results. However, on the Cirrhosis Patient Survival Prediction and Secondary Mushroom Dataset datasets, NN and C4.5 methods performed only slightly better than LRMCW. Meanwhile, the LRMCW model can outperform these two methods if there are much higher dimensions of weights.

In Fig. 4(b) were shown performance characteristics of LRMCW-5-H2-1, LRMCW-5-H4-1, LRMCW-5-H10-1, LRMCW-5-H15-1, LRMCW-5-H20-1 models for various parameters regularization on CIFAR-10 dataset. It can be observed when the regularization value λ was less, LRMCW obtained much higher performance on the CIFAR-10 dataset.

Table 1: The average results of test classification errors, training time, and best result for 5 attempts with various LRMCW models on the CIFAR-10 training dataset. The lowest classification errors for methods are illustrated in bold

	Model	Regularization parameter λ	Classification error (%)	Time (s)
CIFAR-10	LRMCW-1024-H2-10	1×10^{-5}, 1×10^{-6}, 1×10^{-8}	1.77	188
	LRMCW-1024-H4-10	1×10^{-3}, 1×10^{-4}, 1×10^{-6}	1.79	201
	LRMCW-1024-H5-10	1×10^{-3}, 1×10^{-4}, 1×10^{-8}	1.65	210
	LRMCW-1024-H8-10	1×10^{-3}, 1×10^{-5}, 1×10^{-6}	1.69	223
	LRMCW-1024-H10-10	1×10^{-3}, 1×10^{-6}, 1×10^{-7}	1.63	235
	LRMCW-1024-H15-10	1×10^{-4}, 1×10^{-5}, 1×10^{-6}	1.64	241
	LRMCW-1024-H20-10	1×10^{-3}, 1×10^{-4}, 1×10^{-6}	1.69	255
Abalone	LRMCW-1024-H3-3	1×10^{-5}, 1×10^{-4}, 1×10^{-7}	1.83	152
	LRMCW-1024-H4-3	1×10^{-5}, 1×10^{-4}, 1×10^{-8}	1.71	162
	LRMCW-1024-H5-3	1×10^{-5}, 1×10^{-4}, 1×10^{-6}	1.73	169
	LRMCW-1024-H6-3	1×10^{-3}, 1×10^{-4}, 1×10^{-6}	1.52	180
	LRMCW-1024-H7-3	1×10^{-3}, 1×10^{-4}, 1×10^{-6}	1.41	195
	LRMCW-1024-H10-3	1×10^{-3}, 1×10^{-4}, 1×10^{-6}	1.45	206
	LRMCW-1024-H11-3	1×10^{-3}, 1×10^{-4}, 1×10^{-6}	1.43	217

Table 2: The average and smallest result errors of tests with various LR and LRMCW models for 5 attempts on small data sets. The obtained best output result for every training dataset is illustrated in bold

Training dataset	Errors	LR	LRMCW	CART	NN	C4.5	XgBoost
CIFAR-10	average	1.521±3.321	1.401±3.225	1.496±3.455	1.306±2.921	1.433±2.111	1.344±1.134
	smallest	0	0	0	0	0	0
WINE	average	1.582±0.421	1.622±0.729	1.696±0.850	1.151±0.987	1.406±0.749	1.306±0.757
	smallest	0	0	0	0	0	0
CDC diabetes	average	2.726±0.819	2.612±0.770	2.785±0.821	2.784±0.961	2.650±0.644	2.685±0.878
health indicators	smallest	0.541	0.257	0	0	0	0
Abalone	average	1.250±0.685	1.188±0.790	1.405±0.520	1.499±0.882	1.251±0.560	1.306±0.832
	smallest	0.158	0	0	0	0	0
MetroPT-3	average	3.211±1.415	2.956±1.210	3.578±1.350	3.496±2.742	3.201±1.336	3.176±1.453
dataset	smallest	0	0	0	0	0	0
Cirrhosis patient	average	1.632±2.229	1.322±2.156	1.389±1.888	1.120±1.144	1.557±1.250	1.433±0.957
survival prediction	smallest	0.255	0	0	0	0	0
Secondary	average	3.561±1.402	3.204±1.127	3.456±1.250	3.250±1.185	2.966±1.860	3.101±1.475
mushroom dataset	smallest	0.240	0.120	0	0	0	0

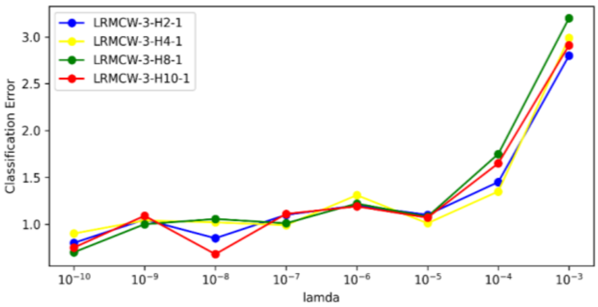


Fig. 4: Classification errors between LRMCWs in the CIFAR 10 dataset

From the experiment results depicted in Table 2, it can be also observed where LRMCW method with a lesser number of weights as well outperforms traditional LR with a large number of weights, which are LRMCW-5-H10-1, LRMCW-5-H15-1, LRMCW-5-H20-1. The results show that LRMCW was superior to LR not because LRMCW had more parameters compared to LR since each LRMCW unit is built coordinating multiple filters, whereas every LR unit is built with only a single filter. In Table 1 it is obvious that the training time for LRMCW is higher, and larger its size, which is the result of introducing a larger number of parameters.

Conclusion

The main idea of the article was inspired by the biological aspects of neurons where a stream of neurotransmitters is released simultaneously and in parallel by one neuron to transmit information to the next neuron. The LRMCW method was the expansion of the dimension of connection weights between the input unit and neural LR from one connection to a multidimensional connection, where each weight corresponds to one type of neurotransmitter. Using computational experiments, the effectiveness of the proposed model was demonstrated on training datasets that are located in a public repository (machine learning repository).

Expanding the dimensions of weights can be considered to significantly improve the performance of the LR model and provide new insights into improving models. However, it should be noted that the design of the encoding mechanism proposed by the model affects the performance of the LRMCW model. That is why increasing the dimension of weight coefficients significantly increases model parameters, which accordingly leads to an increase in the computational complexity of the algorithm. The problem of over-dimensionality becomes more severe when dimensional expansion is applied to large-scale models. However, this may give a model that can be considered close to the biological neurons of the brain. To avoid over-dimensionalization of model parameters and fitting, pruning techniques should be explored to throw away some of the weight connections to reduce the computational complexity of the model, which can be considered as further promising research challenges. Moreover, expanding the number of connections can lead to overfitting problems in some cases. In future research, further work will be carried out on these problems to solve.

Acknowledgment

The authors would like to thank the editors and the anonymous reviewers for their valuable comments and suggestions, which led to a substantial improvement in this study. The authors also thank for the support of the grant from the Republic of Uzbekistan for basic research (grant No. IL-5421101773).

Funding Information

The authors have not received any financial support or funding to report.

Author's Contributions

Kabul Khudaybergenov Kadirbergenovich: Main theoretical idea, model development, Contributed to the written of the manuscript.

Avazjon Marakhimov Rakhimovich: Designed the research plan and organized the study, Contributed to the written of the manuscript.

Zahriddin Muminov Ishkobilovich: This author participated in all experiments, coordinated the data analysis, and contributed to the written of the manuscript.

Kudaybergenov Jabbarbergen Kadirbergenovich: This author participated in all experiments and coordinated the data analysis.

Ethics

The presented research study represents an original contribution and has not been previously submitted or published. The authors have thoroughly reviewed and complied with the submission requirements for this original paper, as well as adhered to the research ethics code.

References

Asuncion, A., & Newman, D. (2007). *UCI machine learning repository*. UC Irvine Machine Learning Repository. https://ergodicity.net/2013/07/

Chaitra, B, C., & Bhaskar Reddy, P. V. (2023). An approach for copy-move image multiple forgery detection based on an optimized pre-trained deep learning model. *Knowledge-Based Systems*, *269*, 110508. https://doi.org/10.1016/j.knosys.2023.110508

Brian, K., Ian Q, W., & Teskey, G. C. (2014). *An introduction to brain and behavior* (1st Ed.). Ormerod, Spring 2018. ISBN: 10- 1-4641-0601-0.

Chen, Y. L., Rinks, D., & Tang, K. (1997). Critical path in an activity network with time constraints. *European Journal of Operational Research*, *100*(1), 122–133. https://doi.org/10.1016/S0377-2217(96)00140-3

Cheng, C., Feng, X., Huang, J., Jiao, Y., & Zhang, S. (2022). ℓ0-Regularized high-dimensional accelerated failure time model. *Computational Statistics & Data Analysis*, *170*, 107430. https://doi.org/10.1016/j.csda.2022.107430

Chorowski, J., & Zurada, J. M. (2015). Learning Understandable Neural Networks With Nonnegative Weight Constraints. *IEEE Transactions on Neural Networks and Learning Systems*, *26*(1), 62–69. https://doi.org/10.1109/tnnls.2014.2310059

Cox, D. R. (1958). The Regression Analysis of Binary Sequences. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *20*(2), 215–232. https://doi.org/10.1111/j.2517-6161.1958.tb00292.x

Crawley, J. N., Stivers, J. A., Blumstein, L. K., & Paul, S. M. (1985). Cholecystokinin potentiates dopamine-mediated behaviors: evidence for modulation specific to a site of coexistence. *The Journal of Neuroscience*, *5*(8), 1972–1983. https://doi.org/10.1523/jneurosci.05-08-01972.1985

Huang, J., Wang, M., Ju, H., Shi, Z., Ding, W., & Zhang, D. (2023). SD-CNN: A static-dynamic convolutional neural network for functional brain networks. *Medical Image Analysis*, *83*, 102679. https://doi.org/10.1016/j.media.2022.102679

Lara-Benítez, P., Carranza-García, M., Luna-Romera, J. M., & Riquelme, J. C. (2023). Short-term solar irradiance forecasting in streaming with deep learning. *Neurocomputing*, *546*, 126312. https://doi.org/10.1016/j.neucom.2023.126312

Lauder, J. M. (1993). Neurotransmitters as growth regulatory signals: Role of receptors and second messengers. *Trends in Neurosciences*, *16*(6), 233–240. https://doi.org/10.1016/0166-2236(93)90162-f

Lin, C.-H., & Wang, T.-Y. (2021). A novel convolutional neural network architecture of multispectral remote sensing images for automatic material classification. *Signal Processing: Image Communication*, *97*, 116329. https://doi.org/10.1016/j.image.2021.116329

Lolaev, M., Madrakhimov, S., Makharov, K., & Saidov, D. (2024). Choosing allowability boundaries for describing objects in subject areas. *IAES International Journal of Artificial Intelligence IJ-AI*, *13*(1), 329. https://doi.org/10.11591/ijai.v13.i1.pp329-336

Madrakhimov, Sh. F., Rozikhodjaeva, G. A., & Makharov, K. T. (2021). Construction of fuzzy inference rules for medicine diagnostics problems. *Journal of Physics: Conference Series*, *2032*(1), 012032. https://doi.org/10.1088/1742-6596/2032/1/012032

Mahmoud, A., Hu, T., Zeng, X., Jing, P., Li, X., & Da Costa Ribeiro, E. (2023). Hydraulic informed multi-layer perceptron for estimating discharge coefficient of labyrinth weir. *Engineering Applications of Artificial Intelligence*, *123*, 106435. https://doi.org/10.1016/j.engappai.2023.106435

Marakhimov, A. R., & Khudaybergenov, K. K. (2022). A Fuzzy MLP Approach for Nonlinear System Identification. *Journal of Mathematical Sciences*, *265*(1), 43–51. https://doi.org/10.1007/s10958-022-06043-z

Martín-Baos, J. Á., García-Ródenas, R., López-García, M. L., & Rodriguez-Benitez, L. (2020). Discrete choice modeling using Kernel Logistic Regression. *Transportation Research Procedia*, *47*, 457–464. https://doi.org/10.1016/j.trpro.2020.03.121

McDonald, A. J., & Pearson, J. C. (1989). Coexistence of GABA and peptide immunoreactivity in non-pyramidal neurons of the basolateral amygdala. *Neuroscience Letters*, *100*(1–3), 53–58. https://doi.org/10.1016/0304-3940(89)90659-9

Ming, H., & Yang, H. (2024). L0 regularized logistic regression for large-scale data. *Pattern Recognition*, *146*, 110024. https://doi.org/10.1016/j.patcog.2023.110024

Rakhimovich, M. A., Kadirbergenovich, K. K., Rakhmovich, O. U., & Rustem, J. (2022). A New Type of Architecture for Neural Networks with Multi-connected Weights in Classification Problems. *World Conference Intelligent System for Industrial Automation*, 105–112. https://doi.org/10.1007/978-3-031-51521-7_15

Rendón-Segador, F. J., Álvarez-García, J. A., Salazar-González, J. L., & Tommasi, T. (2023). CrimeNet: Neural Structured Learning using Vision Transformer for violence detection. *Neural Networks*, *161*, 318–329. https://doi.org/10.1016/j.neunet.2023.01.048

Rigon, T., & Aliverti, E. (2023). Conjugate priors and bias reduction for logistic regression models. *Statistics & Probability Letters*, *202*, 109901. https://doi.org/10.1016/j.spl.2023.109901

Sun, Z., Zhang, M., Sun, H., Li, J., Liu, T., & Gao, X. (2023). Multi-modal deep convolutional dictionary learning for image denoising. *Neurocomputing*, *562*, 126918. https://doi.org/10.1016/j.neucom.2023.126918

Vanderreydt, G., & Demuynck, K. (2024). A novel channel estimate for noise robust speech recognition. *Computer Speech & Language*, *86*, 101598. https://doi.org/10.1016/j.csl.2023.101598

Wang, J., & Park, E. (2017). Active learning for penalized logistic regression via sequential experimental design. *Neurocomputing*, *222*, 183–190. https://doi.org/10.1016/j.neucom.2016.10.013

Wang, K., He, F., He, M., & Huang, X. (2023). Learning non-parametric kernel via matrix decomposition for logistic regression. *Pattern Recognition Letters*, *171*, 177–183. https://doi.org/10.1016/j.patrec.2023.05.018

Wichitaksorn, N., Kang, Y., & Zhang, F. (2023). Random feature selection using random subspace logistic regression. *Expert Systems with Applications*, *217*, 119535. https://doi.org/10.1016/j.eswa.2023.119535

Wu, Y., Zhang, D., Yin, F., & Zhang, Y. (2022). Salient object detection based on global to local visual search guidance. *Signal Processing: Image Communication*, *102*, 116618. https://doi.org/10.1016/j.image.2021.116618

Yang, Y., & Loog, M. (2018). A benchmark and comparison of active learning for logistic regression. *Pattern Recognition*, *83*, 401–415. https://doi.org/10.1016/j.patcog.2018.06.004

Yuan, M., & Xu, Y. (2023). Feature screening strategy for non-convex sparse logistic regression with log sum penalty. *Information Sciences*, *624*, 732–747. https://doi.org/10.1016/j.ins.2022.12.105

Zhang, C.-X., Xu, S., & Zhang, J.-S. (2019). A novel variational Bayesian method for variable selection in logistic regression models. *Computational Statistics & Data Analysis*, *133*, 1–19. https://doi.org/10.1016/j.csda.2018.08.025

Zhang, J., Hu, J., & Liu, J. (2020). Neural network with multiple connection weights. *Pattern Recognition*, *107*, 107481. https://doi.org/10.1016/j.patcog.2020.107481

Zou, W., Tang, L., Zhou, M., & Zhang, X. (2024). Self-disclosure and received social support among women experiencing infertility on reddit: A natural language processing approach. *Computers in Human Behavior*, *154*, 108159. https://doi.org/10.1016/j.chb.2024.108159