Teaching Machine Learning and Deep Learning Introduction: An Innovative Tutorial-Based Practical Approach

M. SABRIGIRIRAJ[1], K. MANOHARAN[2]
[1]Department of Information Technology,
Hindusthan College of Engineering and Technology,
Coimbatore,
INDIA

[2]Department of ECE, SNS College of Technology,
Coimbatore,
INDIA

Abstract: - Machine learning and deep learning techniques have penetrated deep into the various domains of engineering, science, and technology. They are very powerful tools to solve a wide variety of complex problems in those domains. This paper presents an innovative tutorial with practical examples of teaching the introduction to machine learning and deep learning. Starting with the basic concepts, the tutorial takes the readers through the basics of linear regression, logistic regression, and deep neural networks. Then the fundamental association between linear regression, logistic regression, and deep neural network is revealed using the practical examples. This tutorial article provides a solid base for readers aspiring to learn machine learning and deep learning with a systematic and practical approach.

Key-Words: - Machine Learning, Linear Regression, Logistic Regression, Neural Networks, Deep Learning, Artificial intelligence.

Received: August 26, 2023. Revised: April 12, 2024. Accepted: May 9, 2024. Published: June 25, 2024.

1 Introduction

Artificial intelligence (AI) is an exciting field of computer science that finds wide applications in various domains. AI is being increasingly employed in all fields of science, technology, and management for better problem-solving. Machine learning (ML) and deep learning (DL), which are sub-fields of AI are commonly employed in most of the AI applications. ML and DL are widely used to construct expert systems for prediction and classification based on input data. Applications of ML and DL include image recognition, computer vision, speech recognition, stock market prediction, email filtering, and e-commerce website recommender systems among many others. However, researchers face challenges during the deployment of the ML model at every stage, [1] and this provides a wide opportunity for the research community to exploration. A few of the challenging applications include industrial end-systems, [2], medical images, [3], [4], autism disorder [5] and Image captioning [6].

A significant number of tutorials have been published in the area of ML and DL to facilitate the researchers for carrying out cutting-edge research and notable among them are [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17]. A tutorial about the usage of ML models for medical applications is given in [7]. Another tutorial on evaluating ML models for healthcare applications is dealt with in [8]. In [9], a tutorial is given for emotion recognition with ML techniques. A tutorial on automated ML imaging analysis is presented in [10]. In [11], a tutorial on ensemble methods is demonstrated with a practical perspective. In [12], a tutorial for studying fairness in the recommender system is given. An ML primer using Python with a well-edited dataset is described in [13]. A tutorial on reliable and safe ML analysis is covered in [14]. A tutorial on deep learning-based mammogram diagnosis dealt in [15] while an inclusive tutorial on fair and private deep learning is explored in [16]. An innovative tutorial-generating system based on generative AI is surveyed in [17].

Publication of books and book chapters, reviews and research articles focussing on specific applications using ML and DL in a variety of domains can be witnessed every day across multiple platforms. However, discovering the fundamental association between various ML and DL techniques is vital to identifying a suitable technique for efficiently solving complex problems. However, to the best of the authors' knowledge, a direct exhibition of relation connecting various techniques of ML and DL is missing in the literature. Hence, as a starting point, in this tutorial, the authors revisit linear regression, logistic regression, and deep neural networks through simple examples and reveal the connection between them.

The rest of this paper is organized as follows: Section 2 illustrates the concept of machine learning the weights (parameters) of a problem through an example of simple linear regression. Section 3 illustrates the concept of logistic regression through an example and shows the connection between linear regression and logistic regression. Section 4 illustrates the concept of deep neural networks with an example and exposes the fact a deep neural network is composed of multiple logistic regression units connected in series and/or parallel. Section 5 concludes the paper and points out future research directions.

2 Linear Regression

Linear Regression falls under the category of supervised machine learning. It is a very popular method employed for predictive analysis. It is used to find the relationship connecting a dependent variable with one or more independent variables. Interested readers may refer to [18] for more about linear regression techniques.

Example 1: Refer to Table 1 data for a sample regression problem. Find the relation connecting input X with output(s) Y, Y_1, and Y_2.

Table 1. Data for Sample Regression

Input	Output(s)		
X	Y	Y_1	Y_2
2	3	6	7
3	4	11	13
4	5	18	21
5	6	27	31
6	7	38	43

Solution: For this example, determining the relation between input and outputs is very trivial and it is given below:
$Y = 1\ X + 1$. It is a simple linear regression

$Y_1 = 1\ X_2 + 2$. It is a quadratic polynomial linear regression

$Y_2 = 1\ X_2 + X + 1$. It is also a quadratic polynomial linear regression.

Example 2: Refer to Table 1. Find the relation connecting input X with output Y using statistics.
Solution - A simple linear regression is of the form $Y = a\ X + b$ where a is the slope and b is Y-intercept of the line. For a simple linear regression, a and b may be calculated using the following formulas, [19] and the calculations are recorded in Table 2.

$$a = \frac{\sum y \sum x^2 - \sum x \sum xy}{\sum x^2 - (\sum x)(\sum x)}$$
$$b = \frac{n \sum xy - \sum x \sum y}{n \sum x^2 - (\sum x)(\sum x)}$$

Table 2. Regression Computation Data

X	Y	XY	X^2	Y^2
2	3	6	4	9
3	4	12	9	16
4	5	20	16	25
5	6	30	25	36
6	7	42	36	49

From this Table 2, $\sum X = 20$, $\sum Y$=25, $\sum XY = 110, \sum X^2 = 90$, $\sum Y^2 = 135$ and total number of samples n = 5.

Substituting the values, we get $a = 1$ and $b = 1$.
Hence, $Y = 1\ X + 1$.

It is very crucial to note that this statistics method may not be practicable in the case of a very large size dataset containing thousands of rows and columns. The alternative option to statistics is to employ machine learning to solve the problems. This concept is illustrated in the next Example 3.

Example 3: Illustrate how the machine learning algorithm finds the relation connecting input X with output Y of Example 2.
Solution: Machine learning uses the concept of gradient descent (or any other type of gradient descent) to find the relationship between two variables. Figure 1 shows a machine learning algorithm for simple linear regression based on gradient descent optimization, [20].

1. Input: $\{(x, y)\}$, a set of N pairs of (x, y)
2. Specify the desired model for the relation between x and y. Ex: $y = a\,x + b$ for a simple linear regression model.
3. Initialize a and b with random values. a and b are generally known as learnable parameters
4. Calculate Error (Cost) function J, where $J = y - \hat{y} = y - (a\,x + b) = y - a\,x - b$ for a set of N pairs. Also $\hat{y}$ is the predicted value and y is actual value
$$J = \frac{1}{N}\sum_{i=1}^{N}(y_i - ax_i - b)^2$$
5. Calculate partial derivative of J with respect to a and b.
$$\frac{\partial J}{\partial a} = \frac{1}{N} 2 \sum_{i=1}^{N}(y_i - ax_i - a)\ (-x_i\,)$$
$$\frac{\partial J}{\partial b} = \frac{1}{N} 2 \sum_{i=1}^{N}(y_i - ax_i - b)\ (-1\,)$$
6. Update the values of a and b which were randomly initialized in step 3 above
$$a_{new} = a_{old} - \alpha\frac{\partial J}{\partial a}$$
$$b_{new} = b_{old} - \alpha\frac{\partial J}{\partial b}$$
Assume the learning rate (is a hyperparameter) α =0.01 or any suitable small value.
7. Repeat (iterate) step 4 to 6 until J = 0 or J value falls below a threshold value (usually a very small or negligible) normally denoted by ε. Thus, the values of a and b are obtained in machine learning.

Fig. 1: Machine learning Algorithm based on gradient descent, [20]

The core idea of the ML algorithm is to feed the data ({(x, y)} and specify the model expected to fit the data, which may require some domain knowledge. In this example, the model expected to fit the data is assumed to be of the form $Y = a*X + b$ where a and b correspond to the parameters (weights) to be learned. Also, a is termed as weight and b is termed as bias in machine learning terminology. In addition, the learnable parameters a and b have to be initialized with random values [20]. Then, the algorithm finds the difference (error or cost function) between the actual output and predicted output and iterates multiple times (epochs) to take small steps for minimizing the loss. As a result, it converges to the right value of the learnable parameters a and b. This process is known as the 'training phase' of machine learning, which helps to find the right values of the learnable parameters.

A simple linear regressor with only one independent variable may be sketched as in Figure 2. The summer unit performs the operation $a\,X + b$. Figure 3 shows a model of a simple linear regressor with two independent variables. The summer unit performs the operation $a_1x_1 + a_2x_2 + b$. In short, both Figure 2 and Figure 3 are simple linear units performing some linear operation.

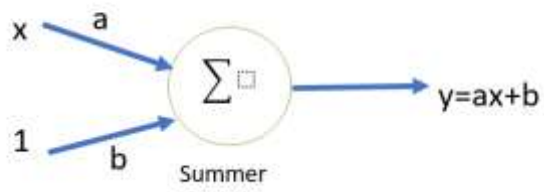

Fig. 2: Model of a linear regressor with one independent variable

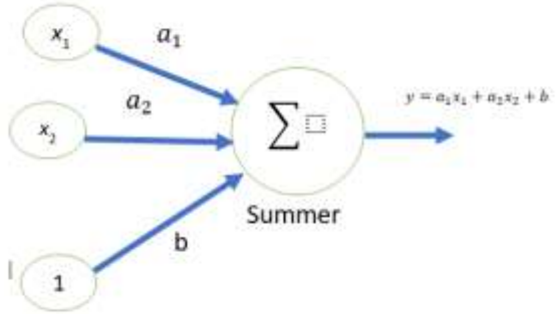

Fig. 3: Model of a linear regressor with two independent variables

In general, machine learning is all about learning the right values of the parameters/weights/bias. Based on the inputs, outputs, and models specified in the machine learning program, it comes up with appropriate values of learnable parameters. More about parameters and hyperparameters can be found in [21].

3 Logistic Regression

Linear Regression belongs to the category of supervised machine learning. It is a very popular method employed for binary classification tasks. It is used to analyze the relationship linking a dependent binary variable with one or more independent variables. Interested readers may refer to [22] for more about logistic regression techniques. Logistic regression can be interpreted as a linear regression unit followed by a threshold sensing or decision-making unit. The threshold sensing or decision-making unit is generally a nonlinear unit. Activation functions are available to perform the role of non-linearity.

Example 4: Draw the truth table of the 'OR' logic gate and 'AND' logic gate. Using the truth tables, illustrate logistic regression and sketch the decision boundary in a 2-dimensional space.
Solution: Figure 4 and Figure 5 show the truth table of the OR gate and AND gate along with their decision boundary. The input combinations are plotted in a two-dimensional space and it is very trivial to get a decision boundary separating output '0' and '1'. The decision boundary line mentioned alongside the figures, separating the two possible outputs, is just one of the many possible solutions.

x_1	x_2	OR(x1,x2)
0	0	0
0	1	0
1	0	0
1	1	1

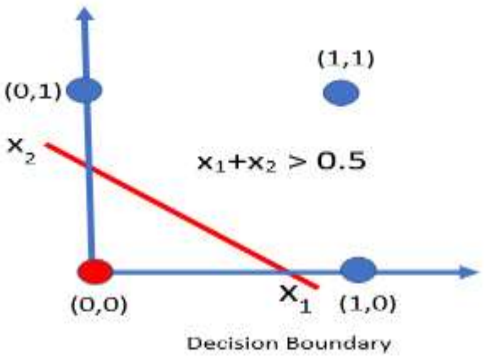

Fig. 4: Truth table of OR gate and its Decision Boundary

x_1	x_2	AND(x1, x2)
0	0	0
0	1	1
1	0	1
1	1	1

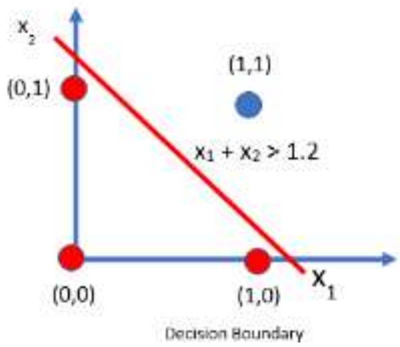

Fig. 5: Truth table of AND gate and its Decision Boundary

Figure 6 shows a model of a logistic regressor with two independent variables. The summer unit performs the operation $a_1x_1 + a_2x_2 + b$. If the output from the summer exceeds a pre-set threshold value, the final output is logical '1', else the final output is logical '0'.

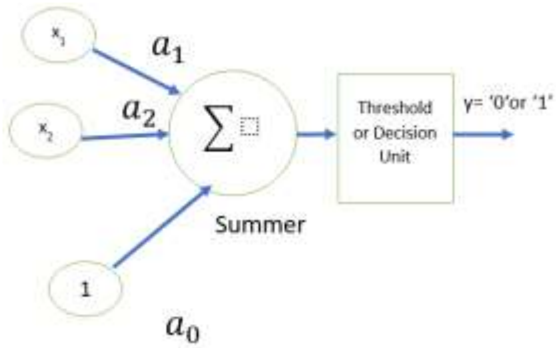

Fig. 6: Model of a logistic regressor with two independent variables

The parameters (weights) a_0, a_1, and a_2 need to be learned so as to satisfy the truth table. For example, for an AND gate, given a threshold value of 30, one of the many possible solutions for the values of a_0, a_1, and a_2 are {10, 10, 10}. Similarly for an OR gate, given a threshold value of 10, one of the many possible solutions for the values of a_0, a_1, and a_2 are {10, 0, 0}. A logistic regressor is thus a linear unit (linear regressor) followed by a non-linear unit (threshold or decision unit). Figure 7 depicts this relation. Interested readers can refer to [23], [24], for parameter learning rules (weight updation) in logistic regression problems. Also, standard non-linear activation functions are available and used implement threshold or decision units, [25].

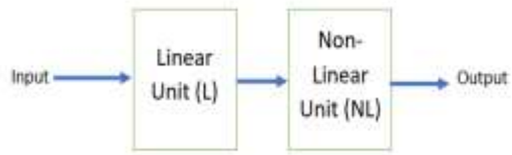

Fig. 7: Simplified block diagram of a logistic regressor

4 Deep Neural Networks

A deep neural network can be mathematically defined as a cascade of similar algorithmic execution units that work together to recognize underlying relationships in a set of data and is the basic requirement for a deep learning network. It has the ability to learn the relationship or the highly complex patterns that exist within the data. Interested readers may refer to [26] for more about neural networks.

Example 5: Draw the truth table of the 'XOR' (Exclusive OR) logic gate. Using the truth table, illustrate its decision boundary

x_1	x_2	XOR(x1,x2)
0	0	0
0	1	1
1	0	1
1	1	0

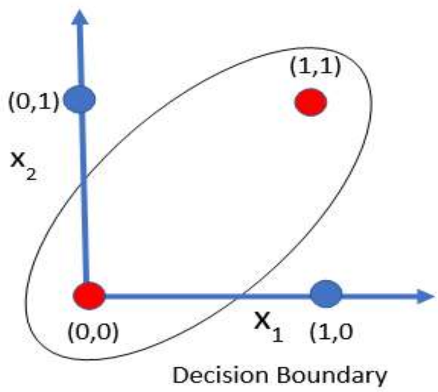

Fig. 8: Truth table of XOR gate and a decision boundary

Figure 8 shows the truth table of the XOR gate along with one of the many possible cases of decision boundary. The input combinations are plotted in a two-dimensional space and it can be easily observed that it is not possible to get a linear decision boundary separating outputs '0' and '1'. Hence, the XOR gate cannot be implemented as a simple logistic regressor with two inputs. Therefore, it is necessary to move to a higher dimensional space to get a boundary separating outputs '0' and '1'.

For moving into a higher dimensional space, the operation of the XOR gate can be represented as
XOR (x_1, x_2) = NOR (NOR (x_1, x_2), AND (x_1, x_2)) and this can be diagrammatically represented as shown in Figure 9.

That is, XOR (x_1, x_2) can be expressed as:
XOR (x_1, x_2) = f_2 (f_2(x_1, x_2), f_1 (x_1, x_2)) where f_1(.) is an AND function and f_2(.) is an OR function. Hence, the XOR gate can be implemented as a neural network with a single hidden layer as shown in Figure 10. Figure 10 corresponds to 2 inputs at the input layer, 2 neurons at the hidden layer, and 1 neuron at the output layer

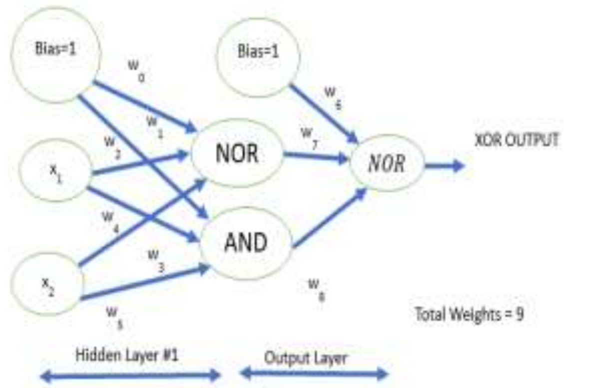

Fig. 9: Interpretation of XOR gate logic implementation

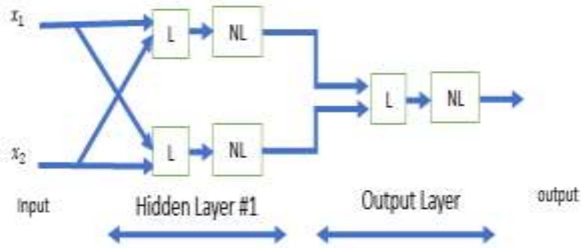

Fig. 10: Block diagram equivalent of XOR gate logic implementation

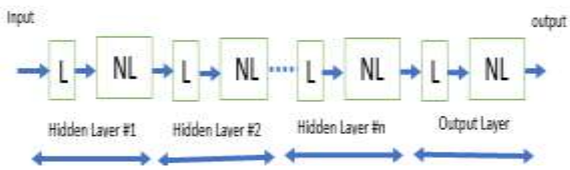

Fig. 11: General block diagram model of a neural network with n hidden layers

Each neuron performs the function of a logistic regressor (applying a nonlinearity function over a linear output). Since the XOR gate function is simple, it was theoretically possible to identify the functions f_1(.) and f_2(.) mentioned above and identify the number of layers and number of neurons required at each layer. However, for complex applications, a number of functions f_1(.), f_2(.), …, f_n(.) of the form f_1(f_2(f_3(…(f_n)…))) may be involved and it is highly complex and difficult to trace it theoretically. For analyzing such complex applications, deep neural networks come to the rescue of the researchers. The various neurons present in the deep neural network will catch these functions in terms of various learnable parameters during the training phase. It is not possible to view these functions mathematically at each of the hidden layers but only end-to-end performance of the network can be studied. Since the number of hidden layers (is a hyperparameter) and the number of neurons in each hidden layer (hyperparameter) cannot be predetermined beforehand for complex applications, these hyperparameters have to be learned only as a trial-and-error process during the training phase of a neural network architecture.

Figure 11 shows a basic block diagram of a deep neural network with *n* hidden layers. A deep neural network is just a cascade of logistic regressors. More complex deep neural networks can be constructed by stacking such cascaded networks one over the other. Interested readers can refer [27] to the parameter learning rule (weight update rule) in neural network architectures.

5 Conclusion

In this tutorial paper, linear regression, logistic regression, and deep neural networks are revisited through simple examples, and the relations between them are directly revealed. Logistic regression is a cascade connection of linear regression unit and nonlinearity, while deep neural networks are a cascade connection of multiple logistic regression units. Also, machine learning is all about learning the right values of learnable parameters, given inputs, and outputs along with the desired model for the machine learning algorithm. An interesting future work is to relate other machine learning and deep learning techniques with each other. Another challenging future work includes developing techniques to identify the optimum number of hidden layers and optimum number of neurons required in each layer of a deep neural network specific to a particular task and application.

References:

[1] Paleyes, Andrei, Raoul-Gabriel Urma, and Neil D. Lawrence, "Challenges in deploying machine learning: a survey of case studies", *ACM Computing Surveys*, Vol.55, No.6, 2022, pp.1-29, https://doi.org/10.1145/3533378.

[2] Nunes, Carlos, EJ Solteiro Pires, and Arsenio Reis, "Machine Learning and Deep Learning applied to End-of-Line Systems: A review", *WSEAS Transactions on Systems*, Vol.21, 2022, pp.147-156, https://doi.org/10.37394/23202.2022.21.16.

[3] Shwetha, V., and CH Renu Madhavi, "MR Image Based Brain Tumor Classification with Deep Learning Neural Networks", *WSEAS Transactions on Systems and Control*, Vol.17 2022, pp.193-200, http://dx.doi.org/10.37394/23203.2022.17.22.

[4] Gancheva, Veska, Ivaylo Georgiev, and Violeta Todorova, "X-Ray Images Analytics Algorithm based on Machine Learning. *WSEAS Transactions on Information Science and Applications*, Vol.20, 2023, pp.136-145, https://doi.org/10.37394/23209.2023.20.16.

[5] Mukherjee, Prasenjit, Sourav Sadhukhan, Manish Godse, and Vodafone Intelligent Solutions, "A Review of Machine Learning Models to Detect Autism Spectrum Disorders (ASD)", *WSEAS Transactions on Computers,* Vol.22, 2023, pp.177-189, https://doi.org/10.37394/23205.2023.22.21.

[6] Nivedita, M, "A survey on different deep learning architectures for image captioning", *WSEAS Transactions on Systems and Control*, Vol.15, 2020, pp.635-646, https://doi.org/10.37394/23203.2020.15.63.

[7] Vercio, L.L., Amador, K., Bannister, J.J., Crites, S., Gutierrez, A., MacDonald, M.E., Moore, J., Mouches, P., Rajashekar, D., Schimert, S. and Subbanna, N., "Supervised machine learning tools: a tutorial for clinicians", *Journal of Neural Engineering*, Vol.17, No.6, 2020, p.062001, DOI: 10.1088/1741-2552/abbff2.

[8] Tohka, Jussi, and Mark Van Gils, "Evaluation of machine learning algorithms for health and wellness applications: A tutorial", *Computers in Biology and Medicine*, Vol.132, 2021, p.104324, DOI: 10.1016/j.compbiomed.2021.104324.

[9] Zhang, Jianhua, Zhong Yin, Peng Chen, and Stefano Nichele, "Emotion recognition using multi-modal data and machine learning techniques: A tutorial and review", *Information Fusion*, Vol.59, 2020, pp.103-126, DOI: 10.1016/j.inffus.2020.01.011.

[10] Thirunavukarasu, Arun James, Kabilan Elangovan, Laura Gutierrez, Yong Li, Iris Tan, Pearse A. Keane, Edward Korot, and Daniel Shu Wei Ting, "Democratizing Artificial Intelligence Imaging Analysis with Automated Machine Learning: Tutorial", *Journal of Medical Internet Research*, Vol.25, 2023, e49949. DOI: 10.2196/49949.

[11] González, Sergio, Salvador García, Javier Del Ser, Lior Rokach, and Francisco Herrera, "A practical tutorial on bagging and boosting based ensembles for machine learning: Algorithms, software tools, performance study, practical perspectives and opportunities", *Information Fusion*, Vol.64, 2020, pp.205-237, DOI: 10.1016/j.inffus.2020.07.007.

[12] Li, Yunqi, Yingqiang Ge, and Yongfeng Zhang, "Tutorial on fairness of machine learning in recommender systems", *Proceedings of the 44th international ACM SIGIR Conference on Research and Development in Information Retrieval*, Online-only event, 2021, https://doi.org/10.1145/3404835.3462814.

[13] Palkovits, Stefan., "A primer about machine learning in catalysis–a tutorial with code", *ChemCatChem*, Vol.12, No.16, 2020, pp.3995-4008, https://doi.org/10.1002/cctc.202000234.

[14] Saria, Suchi, and Adarsh Subbaswamy, "Tutorial: safe and reliable machine learning." *ArXiv preprint arXiv*:1904.07204, 2019, DOI: 10.48550/arXiv.1904.07204,

[15] Naeem, Osama Bin, Yasir Saleem, M. Khan, Amjad Rehman Khan, Tanzila Saba, Saeed Ali Bahaj, and Noor Ayesha, "Breast Mammograms Diagnosis Using Deep Learning: State of Art Tutorial Review", Archives *of Computational Methods in Engineering,* 2024, pp.1-19, DOI: 10.1007/s11831-023-10052-9.

[16] Padala, Manisha, Sankarshan Damle, and Sujit Gujar, "Tutorial on Fair and Private Deep Learning", In *Proceedings of the 7th Joint International Conference on Data Science & Management of Data* (11th ACM IKDD CODS and 29th COMAD), Bangalore, India, 2024, pp.510-513, DOI: 10.1145/3632410.3633294.

[17] Wu, Xiang, HuanHuan Wang, YongTing Zhang, BaoWen Zou, and HuaQing Hong, "A Tutorial-Generating Method for Autonomous Online Learning", *IEEE Transactions on Learning Technologies*, 2024, pp.1558-1567 https://doi.org/10.1109/TLT.2024.3390593.

[18] Gupta, Mohit. "ML, Linear Regression - GeeksforGeeks." GeeksforGeeks, 13 Sept. 2018, [Online]. www.geeksforgeeks.org/ml-linear-regression/ (Accessed Date: June 4, 2024).

[19] I.Goodfellow, Y.Benjio and A.Courville, "Deep Learning", *MIT Press*, 2006.

[20] Menon, Adarsh. "Linear Regression Using Gradient Descent." Medium, 19 Sept. 2018, [Online]. https://towardsdatascience.com/linear-regression-using-gradient-descent-97a6c8700931 (Accessed Date: June 4, 2024).

[21] Nyuytiymbiy, Kizito. "Parameters and Hyperparameters in Machine Learning and Deep Learning." Medium, 5 Apr. 2021, [Online]. https://towardsdatascience.com/parameters-and-hyperparameters-aa609601a9ac (Accessed Date: June 4, 2024).

[22] "Understanding Logistic Regression." GeeksforGeeks, 9 May 2017, [Online]. www.geeksforgeeks.org/understanding-logistic-regression/ (Accessed Date: June 04, 2024).

[23] Tokuç, A. Aylin. "Gradient Descent Equation in Logistic Regression, Baeldung on Computer Science." www.baeldung.com, 30 Jan. 2021, [Online]. www.baeldung.com/cs/gradient-descent-logistic-regression (Accessed Date: June 4, 2024).

[24] Dhalla, adam, "Gradient Descent Update Rule for Multiclass Logistic Regression." Medium, 29 Nov. 2020, [Online]. https://ai.plainenglish.io/gradient-descent-update-rule-for-multiclass-logistic-regression-4bf3033cac10 (Accessed Date: June 4, 2024).

[25] Baheti, Pragati, "12 Types of Neural Networks Activation Functions: How to Choose?" www.v7labs.com, 8 Mar. 2022, [Online]. www.v7labs.com/blog/neural-networks-activation-functions (Accessed Date: June 4, 2024).

[26] "Introduction to Deep Learning - GeeksforGeeks." GeeksforGeeks, 15 Apr. 2019, [Online]. www.geeksforgeeks.org/introduction-deep-learning/ (Accessed Date: June 4, 2024).

[27] Mazur, "A Step-by-Step Backpropagation Example." Matt Mazur, Matt Mazur, 17 Mar. 2015, [Online]. https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/ (Accessed Date: June 4, 2024).

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare.