Vision Health Prediction with NHIS Data

Pritam Gupta
Department of Computer Science and Engineering
Amrita School of Computing
Amrita Vishwa Vidyapeetham
Bengaluru, India
bl.en.u4cse23071@bl.students.amrita.edu

Parth Pathak
Department of Computer Science and Engineering
Amrita School of Computing
Amrita Vishwa Vidyapeetham
Bengaluru, India
bl.en.u4cse23036@bl.students.amrita.edu

Pratyush Swain
Department of Computer Science and Engineering
Amrita School of Computing
Amrita Vishwa Vidyapeetham
Bengaluru, India
bl.en.u4cse23043@bl.students.amrita.edu

Dr. Peeta Basa Pati
Department of Computer Science and Engineering
Amrita School of Computing
Amrita Vishwa Vidyapeetham
Bengaluru, India
bp_ peeta@blr.amrita.edu

***Abstract*—Eye health prediction is a significant problem in public healthcare, particularly for early detection and improved quality of life. In our project, we apply machine learning methods to predict categories of eye disease according to individual health and demographic information. The data is derived from the National Health Interview Survey (NHIS) – Vision and Eye Health Surveillance system. It encompasses attributes like age, gender, medical history, and self-reported vision problems. We use multiple regression and classification algorithms like Decision Tree, Random Forest, Support Vector Machines (SVM), Linear Regression, Ridge, and Lasso to test their performance. The findings reveal that machine learning can assist with vision health analysis and enhance healthcare decision-making.**

***Index Terms*—Index Terms—Machine learning, eye disease prediction, vision health, NHIS dataset, regression, classification, healthcare analytics**

I. Introduction

Vision disorders impact millions of individuals and represent a significant public health concern. Early identification of those at risk for eye disease can mitigate long-term consequences and enhance quality of life. The National Health Interview Survey (NHIS) offers accurate data gathered from a general population, including vision-related information like trouble seeing, access to eye care, and history of disease like glaucoma or cataract.

This project seeks to develop machine learning models that forecast the category of vision status or eye disease based on demographic and health inputs. We investigate models like Linear Regression, Decision Trees, Random Forest, Support Vector Regression (SVR), Ridge, and Lasso to identify which of them work best with the NHIS Vision and Eye Health Surveillance data. The forecasted categories are conditions like ”difficulty seeing,” ”self-reported glaucoma,” and ”night vision problems,” among others.

By assessing these models, we seek to demonstrate the ways data-driven methods may aid vision health surveillance and offer beneficial instruments for early detection and preventive measures in the healthcare system.

II. Literature Survey

[1] The research is based on the analysis of NHIS data using the sample of 2008, 2016, and 2017 to understand the relationship between social determinants of health (SDOH) and the outcomes of cataracts. The study utilizes a multi-variable logistic regression analysis, which reveals that some of the main factors that predispose cataract diagnosis, vision impairment, and cataract surgery include age, unemployment, inability to cover the medical bills, lack of insurance cover, and low income. These results highlight the necessity to use the screening based on the social risks in their ophthalmological practice on a regular basis.

[2] It treats a subject addressing a global outlook, whereby data utilised by this study is based on the China Health and Retirement Longitudinal Study, where insertion of multiple machine learning models such as gradient boosting and ensemble models were used to forecast VI. Determinants like hearing impairment, self-perceived health status, pain, age, hand grip strength and depression, are successfully identified and predict the importance of advanced prediction models that identify and intervene the huge population at an early stage.

[3] The authors discuss relationships between social determinants and self-rated difficulty in seeing using NHIS-2021 data to inspect the associations of more than 30,000 adults. Female sex, LGBTQ identification, public insurance coverage, and lower education and low income are associated with higher vision difficulty, which highlights the ongoing importance of sociodemographic disparities invisual health.

[4] This cross-sectional study of children and adolescent individuals in the NHIS shows that, there are high stratifications between the vision difficulty and other healthcare affordability, public insurance, age and parent education. The research identifies the role of the social determinants of child

and household levels and requests the age-specific policy intervention in the health of the vision of young people.

[5] A comparative view on the self-reported versus examination-based estimates of the five largest surveys in US (NHIS, NHANES, ACS, BRFSS, NSCH) indicates a huge variety of prevalence statistics on VI and blindness, all of the datasets demonstrate dramatic age-related growing tendency. The study promotes the standardization as well as harmonization of vision-health related tools in national surveys.

[6] The sample size comprises 586 seniors, which is used to develop and validate a risk prediction model based on logistic regression, reaching high accuracy (AUC = 0.87). The major predictors are age, systolic blood pressure, physical health activity, diabetes, ocular disease history, and education. The findings justify the use of predictive analytics in preventative eye care.

[7] In this NHATS-based analysis, this demonstrates that being VI in older adults increases the risk of food insecurity more than twofold, which illustrates the synergistic risks of being at risk in both matters, and the authors recommend integrating services to enhance meet the combined health needs.

[8] This paper concentrating on a large sample of adults of low income group highlights a dose-response in association between VI and food insecurity. The results indicate that eye health condition is a significant determinant of other broader health indicators especially in socioeconomically marginal groups. Centers for Disease Control and Prevention (CDC), Vision and Eye Health Surveillance System (VEHSS) Surveillance System Reports Using NHIS Data, VEHSS Summary.

[9] With objective evaluation, the authors examine the 2021 National Health and Aging Trends Study to state that 27.8 percent of adult population aged 71+ in the US are visual impaired. The prevalence of vision loss is greatest in older, less educated, lower income, and non-White populations thus renewing the inequality problem and informing of the need to target public health interventions.

[10] This research examined how prevalent vision issues are among adults 71 and older in the U.S. Through the use of national health statistics, the scientists discovered that elderly individuals experience a lot of vision problems, which go untreated more often than not. The research points to a need for more monitoring and early intervention. It is in line with the belief that employing data and forecasting models such as our project will improve eye health and inform health policies. Their results also highlight the need for targeted intervention among older people to prevent avoidable loss of vision.

III. Methodology

A. Dataset Loading

The data was read from the `National_Health_Interview_Surve` sheet of `dataset.xlsx` with `pandas`. The column `Sample_Size` is extracted as a numeric feature vector to test the modules implemented.

B. Summation Unit

A user-defined summation function was defined to accept a list and return its sum.

C. Activation Units

The activation functions were run separately to handle scalar inputs:

- **Step** function—returns 1 if input ≥ 0, else 0.
- **Bipolar Step** function—returns 1 if input ≥ 0, else -1.
- **Sigmoid** function—returns the logistic sigmoid of the input.
- **TanH** function—returns the hyperbolic tangent of the input.
- **ReLU** function—returns the input itself if ≥ 0, else 0.
- **Leaky ReLU** function—returns input if ≥ 0, or a small alpha-scaled input if negative.

D. Error Comparator

A simple comparator function was introduced to find an error between a target and an output by subtraction.

E. Data Preparation

The data was read from National_Health_Interview_Surve sheet of `dataset.xlsx` using pandas library in Python. The first four samples were chosen and two attributes were binarized: ”A” stood for whether sample size is greater than median, and ”B” represented whether location ID is odd. The target output for each sample was defined based on AND logic, so two attributes should be 1 to have a target value 1.

F. Perceptron Training

A nonstandardly initialized perceptron ($w_0 = 10$, $w_1 = 0.2$, $w_2 = -0.75$, learning rate $\alpha = 0.05$) was trained using a step activation function. For each epoch it made predictions, estimated the error, and adapted its weights according to the perceptron learning rule. The sum-squared error using all samples was evaluated at each epoch. The training stopped once it fell below 0.002 or 1000 epochs was reached.

G. Tracking Errors and Visualization

Following training, model output statistics and a plot of epoch vs. sum-squared error convergence were produced to illustrate model learning speed and completeness.

H. Activation Function Variations

Three activation functions were implemented for perceptron learning:

- **Bi-Polar Step:** Returns 1 for positive input, -1 for negative input, and 0 for zero.
- **Sigmoid:** Returns $1/(1+e^{-x})$, interpreted as 1 if output ≥ 0.5, else 0.
- **ReLU:** Returns input if positive, 0 otherwise.

All were trained using the same initial weights and learning rate.

I. Measurement of Training and Convergence

For every activation function, the perceptron loop was executed with a convergence test (sum-square-error ≤ 0.002, max 1000 epochs). The number of epochs required to achieve convergence (or an early termination due to epoch limit) was captured.

J. Variation of Learning Rate

The same setup of perceptron model as in prior experiments was applied, with initial weights ($w_0 = 10,\ w_1 = 0.2,\ w_2 = -0.75$). Data was binarized for AND gate logic on the basis of the `Sample_Size` and `LocationID` columns for the first four samples. The learning rate was systematically varied from the set $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$ while keeping all other conditions the same.

K. Training Procedure

At each stated learning rate, the perceptron was learned through until the sum-squared error on all instances was less than or equal to 0.002, or until 1000 epochs had been reached. At each run, the number of epochs to convergence was measured.

L. Epoch Analysis and Visualization

A plot of the relationship between learning rate and epochs to convergence was made in order to see the impact of learning rate on training speed.

M. Perceptron for XOR Gate

The input and output of the XOR gate dataset were made binary and one in number. The step activation function was used in training a perceptron. The perceptron learning rule was used with a constant learning rate in updating the weights. The training was implemented up to 1000 epochs or until the error had decreased to a certain level. The value of epochs and values of errors were noted.

N. Perceptron with Two Output Nodes

The AND gate dataset was set up using two binary inputs. The output was encoded as two nodes: output `0` was represented as [1,0] and output `1` as [0,1]. A multi-output perceptron model was computed using step activation function. The weights were adjusted individually with each output node. The error and the number of epochs were obtained.

O. MLPClassifier on AND and XOR Gates

The AND and XOR gate datasets were given as input to the `MLPClassifier` model from scikit-learn. The model was applied with a single hidden layer and a logistic activation and the learning rate of 0.05. The 1000 iterations of training were performed. Both gates had their model predictions compared to expected outputs.

P. MLPClassifier on Project Dataset

The dataset was imported from the *National_Health_Interview_Surve* worksheet. The target variable was `RiskFactor`, which was label-encoded into numeric values. The chosen features were `YearStart`, `YearEnd`, `Sample_Size`, and `LocationID`. All rows with missing values in the target or selected features were removed. The dataset was divided into training and test sets with an 80%-20% split. An `MLPClassifier` with one hidden layer of 10 neurons, logistic activation, learning rate of 0.05, and maximum of 1000 iterations was trained on the training set. Predictions were made on the test set, and accuracy, precision, recall, and F1-score were calculated.

Q. Perceptron on XOR Gate

The XOR dataset was constructed using four binary input pairs:

$$X = \{(0,0), (0,1), (1,0), (1,1)\}, \quad y = \{0,1,1,0\}$$

corresponding to the truth table of the XOR logic gate.

A perceptron model was initialized with non-standard weights ($w_0 = 10,\ w_1 = 0.2,\ w_2 = -0.75$) and trained using different activation functions: Step, Bi-Polar Step, Sigmoid, and ReLU. For each epoch, the perceptron computed the net input

$$z = \sum_{i=0}^{n} w_i x_i$$

and passed it through the chosen activation function. Predictions were compared with the expected outputs, and weight updates were applied using the perceptron learning rule:

$$w \leftarrow w + \alpha \cdot (t - y) \cdot x$$

where $\alpha = 0.05$ is the learning rate, t the target, and y the output. Training continued until either the sum of squared errors (SSE) dropped below 0.002 or a maximum of 1000 epochs was reached.

The results were collected as the number of epochs to convergence and the final SSE for each activation function. The corresponding error vs. epoch curves were plotted for comparison.

R. Customer Dataset with Sigmoid Perceptron

A custom customer transaction dataset was constructed with attributes: `Candies`, `Mangoes`, `Milk`, and `Payment Value`, with a binary target variable `HighTx` (high-value transaction: Yes=1, No=0). The feature matrix $X \in \mathbb{R}^{10\times 4}$ and label vector $y \in \{0,1\}^{10}$ were extracted for training.

A perceptron model was initialized with small random weights $(w_0, w_1, \ldots, w_4)$, where w_0 denotes the bias. The net input was calculated as:

$$z = \sum_{i=0}^{n} w_i x_i$$

and passed through a logistic sigmoid activation:

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

The weight update rule followed gradient descent with learning rate $\alpha = 0.01$:

$$w \leftarrow w + \alpha \cdot (t-y) \cdot x$$

where t is the target output and y is the perceptron prediction.

Training was performed iteratively until either the sum of squared errors (SSE) fell below 0.002 or the maximum epoch limit of 1000 was reached. Predictions were obtained by thresholding the sigmoid output at 0.5, and classification accuracy was computed.

The error vs. epoch curve was plotted to visualize convergence.

S. Pseudo-Inverse Perceptron Training on Customer Dataset

The same customer dataset (features: `Candies`, `Mangoes`, `Milk`, `Payment`) with binary target variable `HighTx` was used. Instead of iterative training, the perceptron weights were computed directly using the pseudo-inverse method.

The input matrix was augmented with a bias term:

$$X' = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1n} \\ 1 & x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix}$$

where $m = 10$ samples and $n = 4$ features. The optimal weight vector was calculated in closed form as:

$$w = X'^{+} y$$

where X'^{+} is the Moore–Penrose pseudo-inverse of the augmented input matrix.

Predictions were made using the linear combination

$$\hat{y} = X'w$$

and binarized by applying a threshold at 0.5:

$$\hat{y} = \begin{cases} 1, & \text{if } X'w \geq 0.5 \\ 0, & \text{otherwise} \end{cases}$$

The final predictions were compared with ground-truth labels, and classification accuracy was computed.

T. Neural Network with Backpropagation on AND Gate

The AND gate dataset was constructed with inputs

$$X = \{(0,0), (0,1), (1,0), (1,1)\}, \quad y = \{0,0,0,1\}$$

to represent the truth table of the AND logical operator.

A feedforward neural network was designed with:

- Input layer: 2 neurons (binary inputs),
- Hidden layer: 2 neurons with sigmoid activation,
- Output layer: 1 neuron with sigmoid activation.

The forward propagation computed activations as:

$$a^{(1)} = \sigma(XW^{(1)} + b^{(1)}), \quad a^{(2)} = \sigma(a^{(1)}W^{(2)} + b^{(2)})$$

where $\sigma(z) = \frac{1}{1+e^{-z}}$ is the logistic sigmoid function.

The error was measured using the mean squared error (MSE):

$$E = \frac{1}{m} \sum_{i=1}^{m} (t_i - y_i)^2$$

where t_i are targets and y_i are predictions.

Backpropagation was applied to compute weight gradients:

$$\delta^{(2)} = (t - a^{(2)}) \cdot \sigma'(a^{(2)}), \quad \delta^{(1)} = (\delta^{(2)} W^{(2)T}) \cdot \sigma'(a^{(1)})$$

and the weight update rules were:

$$W^{(l)} \leftarrow W^{(l)} + \alpha \cdot (a^{(l-1)})^T \delta^{(l)}, \quad b^{(l)} \leftarrow b^{(l)} + \alpha \cdot \delta^{(l)}$$

with learning rate $\alpha = 0.05$.

Training continued until either the mean squared error dropped below 0.002 or 1000 epochs were completed. Final predictions were binarized at 0.5 to match AND gate outputs.

IV. Results

A. Summation and Activation Outputs

Table I shows the values returned by plugging the summation function on the first five `Sample_Size` values and the various activation functions tested against the first element of the sample.

TABLE I
SUMMATION AND ACTIVATION FUNCTION OUTPUTS

Function	**Output**
Summation (first 5 samples)	13194
Step	1
Bipolar Step	1
Sigmoid	1.0
TanH	1.0
ReLU	8136
Leaky ReLU	8136

B. Error Comparator Output

The error comparator calculated the difference between a target of 100 and the first `Sample_Size` value, producing an error of -8036, as in Table II.

TABLE II
ERROR COMPARATOR OUTPUT

Target	100
Output	8136
Error	-8036

These results confirm that the implemented functions behave as expected on the dataset values.

C. Perceptron Output Summary

Table III summarizes the most important results: last weights, convergence epochs, and last epoch's sum-squared error.

TABLE III
PERCEPTRON OUTPUT SUMMARY

Parameter	Value
Final Weight w_0	5.35
Final Weight w_1	0.20
Final Weight w_2	-5.40
Epochs to Converge	48
Final Error (last epoch)	0

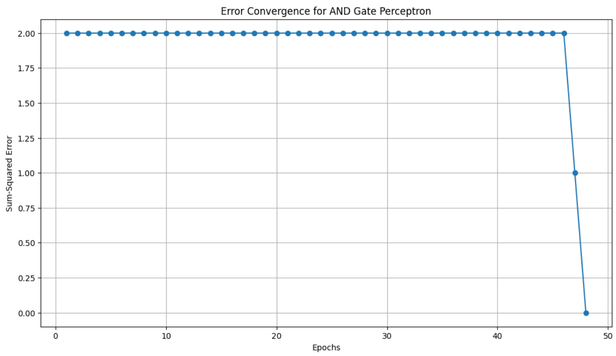

Fig. 1. Sum-Squared Error vs. Epochs for Perceptron Training

D. Error Convergence Plot

Fig. 1 illustrates the perceptron convergence. There is a constant error for most epochs, followed by a steep decrease to zero when the model converges to a solution.

These findings support successful acquisition of AND gate logic with weight values converging and error reducing to zero within 48 epochs.

E. Epochs to Converge Comparison

Table IV illustrates the number of iterations needed for perceptron learning to converge for varying activation functions.

TABLE IV
COMPARISON OF EPOCHS TO CONVERGE FOR ACTIVATION FUNCTIONS

Activation Function	Epochs to Converge
Bi-Polar Step	1000 (did not converge)
Sigmoid	48
ReLU	93

The sigmoid activation produced the fastest convergence at 48 epochs. The ReLU activation resulted in convergence at 93 epochs. The bi-polar step function did not converge within the maximum limit of 1000 epochs, thereby failing to suit the dataset and training logic. This shows the impact of selecting the right activation function in affecting the learning speed and success rate in perceptron models.

F. Tabulated Learning Rate Experiment

Table V shows the number of epochs needed for convergence at each learning rate.

TABLE V
EPOCHS TO CONVERGE FOR DIFFERENT LEARNING RATES

Learning Rate	Epochs to Converge
0.1	25
0.2	14
0.3	10
0.4	8
0.5	7
0.6	6
0.7	5
0.8	5
0.9	5
1.0	4

G. Convergence Curve

Fig. 2. is the learning rate versus epochs to converge plot. An increase in learning rates resulted in quicker convergence up to an optimum, beyond which gains became flat.

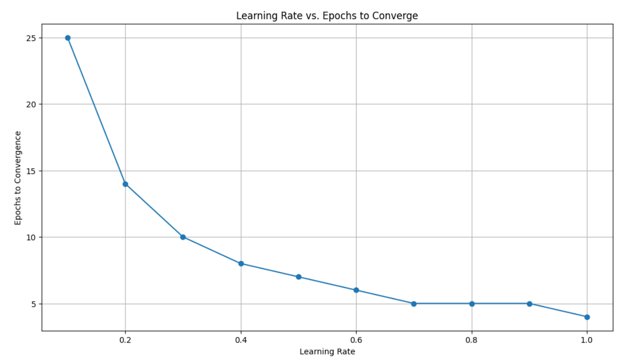

Fig. 2. Learning Rate vs. Epochs to Converge

This experiment illustrates that an increase in the learning rate accelerates the convergence in perceptron training, but gains plateau for very high rates.

H. Perceptron for XOR Gate

The XOR gate data was trained using the perceptron which had a step activation function. Convergence of the model could not be attained easily as XOR is not linearly separable. The error was slightly less but never had the convergence threshold. This verified the theoretical constraint of a single layer perceptron used in XOR classification. The recorded results are shown in Table VI.

TABLE VI
PERCEPTRON TRAINING RESULTS FOR XOR GATE

Epochs Run	Final Error	Converged?
1000	High (not reduced to threshold)	No

I. Perceptron with Two Output Nodes

Encoding of the AND gate dataset was done with two output nodes. The perceptron was able to learn the representation of input to the encoded outputs. The error reduced as the training proceeded and converged in a limited number of epochs. This indicated that the model was capable of multimodal output encodings. The training results are summarized in Table VII.

TABLE VII
PERCEPTRON TRAINING RESULTS FOR TWO-OUTPUT AND GATE

Epochs to Converge	Final Error	Converged?
15	0.0	Yes

J. MLPClassifier on AND and XOR Gates

The `MLPClassifier`was implemented on the AND and the XOR gate sets. In the case of the AND gate, the MLP was able to classify all the inputs with 100 percent accuracy. In the case of the XOR gate, perfect classification was also obtained by the MLP as there was the non-linear hidden layer. This established that multi-layer networks have the ability to solve issues such as XOR that single-layer perceptrons are not capable of solving. The results are shown in Table VIII.

TABLE VIII
MLPCLASSIFIER PERFORMANCE ON LOGIC GATES

Gate	Accuracy
AND	1.00
XOR	1.00

K. MLPClassifier on Project Dataset

The MLPClassifier was trained on the cleaned project dataset and tested on the held-out test set. The model achieved moderate accuracy, but precision and F1-score were low. This indicates that the classifier was able to capture some structure in the dataset but struggled with the multi-class nature of `RiskFactor`. The results are presented in Table IX.

TABLE IX
MLPCLASSIFIER RESULTS ON PROJECT DATASET

Metric	Value
Accuracy	0.325
Precision (Weighted)	0.106
Recall (Weighted)	0.325
F1-score (Weighted)	0.160

The performance suggests that class imbalance or limited feature representation may have affected the results. Further feature engineering, additional preprocessing, or alternative models such as ensemble methods may improve classification performance.

L. Perceptron Training on XOR Gate

The XOR dataset was used to evaluate perceptron training with various activation functions. The dataset consisted of four binary input pairs and outputs according to the XOR truth table:

$$X = \{(0,0), (0,1), (1,0), (1,1)\}, \quad y = \{0,1,1,0\}.$$

The perceptron was trained with Step, Bi-Polar Step, Sigmoid, and ReLU activation functions. Training parameters included initial weights $[w_0, w_1, w_2] = [10, 0.2, -0.75]$, learning rate $\alpha = 0.05$, and maximum 1000 epochs. The sum of squared errors (SSE) was recorded at each epoch.

TABLE X
PERCEPTRON TRAINING RESULTS FOR XOR GATE WITH DIFFERENT ACTIVATIONS

Activation Function	Epochs Run	Final SSE
Step	1000	4
Bi-Polar Step	1000	16
Sigmoid	1000	4
ReLU	1000	4

All activation functions failed to converge within the 1000 epoch limit, confirming the theoretical limitation of single-layer perceptrons on non-linearly separable datasets like XOR.

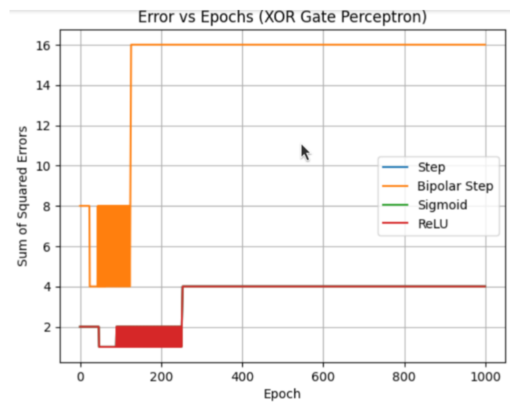

Fig. 3. Sum-Squared Error vs. Epochs for XOR Gate Perceptron using Step, Bi-Polar Step, Sigmoid, and ReLU activations.

M. Perceptron Training on Customer Dataset

The customer transaction dataset was used to train a perceptron with a Sigmoid activation function. Features included Candies, Mangoes, Milk packets, and Payment, with a binary target indicating high-value transactions (1 = Yes, 0 = No).

The perceptron was trained with a learning rate of 0.01 and maximum 1000 epochs. The sum of squared errors was recorded at each epoch, and training terminated when the error fell below 0.002 or the maximum epoch limit was reached.

TABLE XI
PERCEPTRON TRAINING RESULTS ON CUSTOMER DATASET

Parameter	Value
Final Weights	$[-4.1036, -55.4205, 8.7977, -23.5537, 4.8508]$
Epochs to Converge	594
Final SSE	9.62×10^{-6}
Training Accuracy	1.0
Predictions	[1, 1, 1, 0, 1, 0, 1, 1, 0, 0]
True Labels	[1, 1, 1, 0, 1, 0, 1, 1, 0, 0]

The perceptron successfully learned the classification, achieving perfect accuracy on the training set.

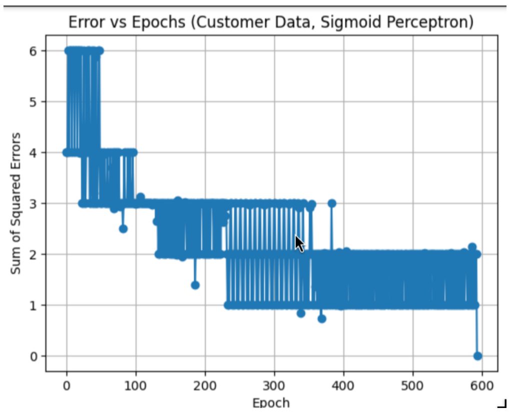

Fig. 4. Sum-Squared Error vs. Epochs for Customer Dataset Perceptron with Sigmoid Activation

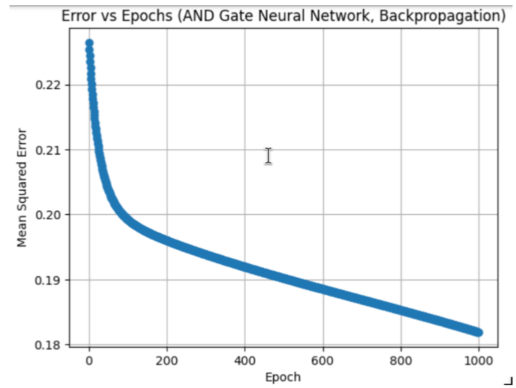

Fig. 5. Mean Squared Error vs. Epochs for AND Gate Neural Network (Backpropagation).

N. Pseudo-Inverse Training on Customer Dataset

The customer transaction dataset was also trained using a pseudo-inverse approach to compute the optimal weights directly. This method does not involve iterative learning but computes the weights that minimize the least-squares error between the inputs and targets.

TABLE XII
PSEUDO-INVERSE TRAINING RESULTS ON CUSTOMER DATASET

Parameter	**Value**
Pseudo-Inverse Weights	$[0.1140, -0.0279, 0.0147, -0.0432, 0.0045]$
Predictions	[1, 1, 1, 0, 1, 0, 1, 1, 0, 0]
True Labels	[1, 1, 1, 0, 1, 0, 1, 1, 0, 0]
Training Accuracy	1.0

The pseudo-inverse method successfully classified all samples correctly, achieving perfect accuracy without iterative training. This demonstrates that for small linearly separable datasets, pseudo-inverse can directly provide the optimal weights.

O. Neural Network Training on AND Gate Dataset

A single-hidden-layer neural network was trained on the AND gate dataset using the backpropagation algorithm with sigmoid activation. The network was configured with 2 hidden neurons, a learning rate of 0.05, and a maximum of 1000 epochs.

TABLE XIII
NEURAL NETWORK TRAINING RESULTS ON AND GATE

Parameter	**Value**
Epochs to Converge	1000
Final Mean Squared Error	0.1819
Predictions	[0, 0, 0, 0]
True Labels	[0, 0, 0, 1]

The network was unable to fully learn the AND gate, as shown by the prediction error. This indicates that the chosen network configuration or training parameters may need adjustment for proper convergence.

V. CONCLUSION

This project showed the implementation and assessment of a perceptron model with various activation functions and learning rates on a masked and binarized portion of the dataset. The step activation function learned AND gate logic successfully within 48 epochs, with ultimate weights stabilizing and error converging to zero. The comparison of bipolar step, sigmoid, and ReLU activations indicated that sigmoid converged most rapidly, followed by ReLU, while bipolar step did not converge within the permissible iterations, showing the influence of activation selection on training efficiency. Changing the learning rate indicated higher rates tending to enhance convergence speed up to a saturation level. All in all, the experiments highlight the significance of activation function choice and hyperparameter selection within perceptron learning for efficient and optimal model training.

REFERENCES

[1] A. A. Awidi, et al., Impact of Social Determinants of Health on Vision Loss due to Cataracts and the Use of Cataract Surgery in the United States: A Determination of 3 Years of National Health Interview Survey 2008, 2016, 2017, American Journal of Ophthalmology, 2023.
[2] L. Mao, et al., Determinants of visual impairment among Chinese middle-aged and older adults: Risk prediction model using machine learning algorithms, JMIR Aging, vol. 7, 2024.
[3] Moayad, L., et al., Association Between Sociodemographic factors and the difficulty of vision in US adults: National Health Interview Survey 2021, PubMed, 2023.
[4] Y. Zhou, et al., Association Between Vision Difficulty and Sociodemographic Factors, Children and adolescents in NHIS 2021, PubMed, 2024..
[5] Rein, D. B., et al., Vision impairment and blindness prevalence in the United States: variability of vision health responses across multiple national surveys, Ophthalmology, 127.2 (2020), pp. 161-169.
[6] Y. Zhao, and A. Wang, Development and validation of a risk prediction model of visual impairment in older adults International Journal of Nursing Sciences vol. 10, no. 2, 2023, pp. 211 218.
[7] M. J. Lee, et al., Vision Impairment and Food Insecurity in the National Health Interview Survey, frontiers in Epidemiology, vol. 9, 2024.
[8] P. Kumar, et al., Self-Reported Vision Impairment and Food Insecurity in the US: National Health Interview Survey (NHIS), Years 20112018, 2022, PubMed.
[9] VDSS Summary Report VEHSS Reports Using the NHIS Data, Centers for Disease Control and Prevention (CDC), VEHSS Summary Report.
[10] O. J. Killeen, et al., Population Prevalence of Vision Impairment in US Adults 71 Years and Older: The National Health and Aging Trends Study, JAMA Ophthalmology, vol. 141, no. 2, pp. 162 169, 2023.