Hamming Code Detecting and Correcting Errors in Credit Cards Using Neural Networks

Dr R.Dhanapal[1] , Gayathiri.P[2]

[1] *Professor and Director*
Research and Development
Procademia
E-mail:drdhanapal@gmail.com,
drdhanapal@procademia.com

[2]*Asst.. Professor*
Research Scholar in MSU University
Department of Computer Science,
Kanchi Sri Krishna College ,Kanchipuram ,
Email:gai3msc@gmail.com

Abstract-In this Paper we Discuss about Hamming Code Detecting and Correcting errors in Credit Card Information such as Card Number,Expiry Date and CVV Number .card can store the card holder information in chip .chip transmit the card holders information in binary form.the detection and correction of error two types ,the type detects the errors during the transmission and the second type corrects the errors during the transmission.one common type of error detecting code is the parity check code used in credit card. Hamming developed technique for detecting and correcting single bit errors in transmitted data. The technique requires that three parity bits (or check bits) be transmitted with every four data bits. The algorithm is called a (7,4) code, because it requires seven bits to encoded four bits of data. implementation of Hamming codes.Parity bit is used to detect the errors in codeword.Parity bit check whether the error occur in the card holder information or not and how to correct the error in the bit and it can implemented in Hamming Code Genarator Matrix using Neural Networks.

Keywords: Hamming Code,Neural Networks,Credit Card,Genarator Matrix,Parity Bit.

I..Introduction

A swipeless or contactless credit card to store the information within a smart chip. The chip is embedded within the credit card itself. When exposed to a contactless credit card reader transmits the data through a parity bit, or check bit, is a bit added to the end of a card holder data of binary code. Hamming Code Parity bits are used as the form of Error Detecting Code .There are two variants of parity bits: even parity bit and odd parity bit. for example credit card data information such as card number,expiry date and CVV given in Table 1. If an odd number of bits (including the parity bit) areTransmitted incorrectly, the parity bit will be incorrect, thus indicating that a parity error occurred in the transmission otherwise successful Transmission. Hamming code Detecting errors in Credit Card Information such as Card No,Expiry Date and CVV Number.[13].we have to check whether the error occur in the code word or not Card Holder information is given in table1 If an odd number of bits (including the parity bit) areTransmitted incorrectly the parity bit will be incorrect, thus indicating that a parity error occurred in the transmission. The parity bit is only suitable for Detecting errors. 24 digits encoded in to two binary numbers such as 2 contains the binary value 0010 and 4 contains the 0100 value.Hamming code (7,4) matrix encoded data in binary form and implement in to neural network layer1 and layer2 produce the output[2].

Table 1 for cardholder information

Field	Length
Credit Card No	16
Expiry Date	4
Cvv	4
Total Bits	24

II. Hamming Code

Hamming Code is an Error Correction Code .hamming code introducing number of parity bits in the code.it can detect the position of the error in the bit.Determine error correcting code for BCD Number 0010 and 0100 using even and odd parity.let p=3 2^3 =8 since $2^p \geq m+p+1=8$. Total code bits=7. Binary data 0010 and 0100 code because 7 bits to encoded four bits of data.Hamming code is implemented with the matrix.in this paper Network model of correcting and detecting errors in Hamming Codes.Neural networks detects an error in the bit and Display the Correct Code word[1].Hamming developed technique for detecting and correcting single bit errors in transmitted data. His technique requires that three parity bits (or check bits) be transmitted with every four data bits. The algorithm is called a (7, 4) code[1] because it requires seven bits to encoded four bits of data. implementation of Hamming codes.In hamming Mtrix 4 bit code 0010 and 0100 convert to error correction code produce the result 7 bit codeword contains 3 parity bits and 4 data bits. In the below table P_1, P_2 and P_4 are parity bits and d_3 , d_5, d_6 and d_7 are data bits[2] .

Table 2 for 7 bit hamming code for word 0010

Data Bits			d_3		d_5	d_6	d_7
Parity Bits	P_1	P_2	0	P_4	0	1	0
Bit Position	1	2	3	4	5	6	7

The error can be corrected and detected using the following procedure

P_1 check the bit 3,5 and 7 =0+0+0=0

P_2 check the bit 3,6 and 7=0+1+0=1

P_4 check the bit 5,6 and 7=0+1+0=1

Hamming code check matrix of X_1

$$X_{1=}\begin{bmatrix}000\\010\\010\end{bmatrix}\begin{bmatrix}0\\1\\1\end{bmatrix}$$

7 bit Hamming code[3] was transmitted and it was received as 0101010.

Table 3 for for 7 bit hamming code for word 0100

Data Bits			d_3		d_5	d_6	d_7
Parity Bits	P_1	P_2	0	P_4	1	0	0
Bit Position	1	2	3	4	5	6	7

Hamming code check matrix of X_2

$$X_2=\begin{bmatrix}010\\000\\100\end{bmatrix}\begin{bmatrix}1\\0\\1\end{bmatrix}$$

The error can be corrected and detected using the following procedure

P_1 check the bit 3,5 and 7 =0+1+0=1

P_2 check the bit 3,6 and 7=0+0+0=0

P_4 check the bit 5,6 and 7=1+0+0=1

Hence the code is 1001100.7 bit Hamming code was transmitted and it was received as 1001100.

III. NEURAL NETWORKS

A Neural Network has a parallel distributed architecture with a large number of nodes and connections.each connection points from one node and another and its associated with a weight.Figure 1 determine the Basic Structure of the Graph.Topology of Neural Network refers[7] to its Framework and its interconnection scheme.the Framework specified by the number of layers.the types of layes include.

Input layer:the input node encode the instance presented to the Network for Processing.for example each input designated by an attribute value possessed by the instance.

Hidden layer: The hidden layer which are not directly observable and hence hidden.they provide nonlinearities for the network.

Output layer:output node which encode possible concepts(or values) to be assigned to the instance under consideration

In this example there is one input,hidden and output layer. Input layer functions x_1, $x_{x....}$ x_n simply take the corresponding attribute value in the replicate it as output on each of the arcs coming out of the node.the function of the hidden layer perform more complicated functions.the arcs are called labeled with weights, where w_{ij} is the weight between nodes and j,the functions at each node are applied too the input data produce the output[8].

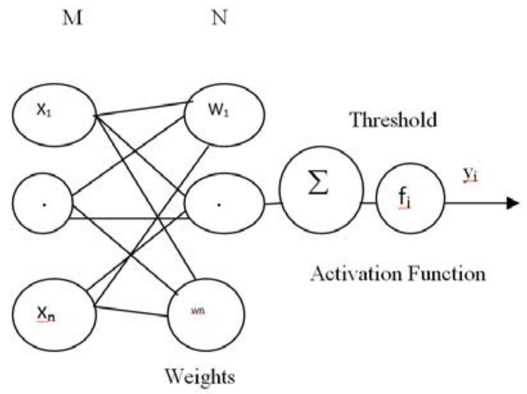

Figure1 Neural Network structure

A. Algorithm

A Neural Network is a Directed Graph F=<V,W>

Inputs: V=M*N Matrix

M=[W_1,$W_{2...}W_m$]

N=[W_1,$W_{2...}N_n$]

M represent the Check Matrix and N represent the Codeword

W=(i,j)

1: V is Partitioned in to set of input Nodes $V_{I,}$ hidden nodes V_H and output node V_O

2: Vertices are Partitioned in to layer{1…N} with all input nodes in layer1 and output nodes layer1. all hidden nodes are in layer2 to N-1 nare called hidden layer.

3:Any arc<i,j> must have node i in integer h-1 and node j in layer h

4:W<i,j> is labeled with a numeric value w_{ij}

5:Node I is labeled with a function f_i

output:Hamming Matrix M*N

S=H*R

H=Hamming Code

R=Received Codeword

IV.NEURAL NETWORKK THRESHOLD

Neural networks are models of biological neural structures. The starting point for most neural networks is a model neuron, as in Figure 2. This neuron consists of multiple inputs and a single output. Each input is modified by a *weight*, which multiplies with the input value. The neuron will combine these weighted inputs and, with reference to a threshold value and activation function, use these to determine its outpu[6]. This behavior follows closely our understanding of how real neurons work.

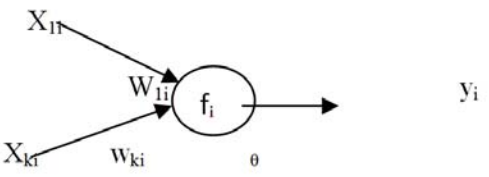

Figure 2 for model neuron

there are k input arcs coming from nodes 1,2..k with weights of $w_{1i....}w_{ki}$ and output values of $x_{1i....}x_{ki}$ there is one output value y_i produced.This value is output on all output arcs of the node.the activation function f is applied to the inputs,which are scaled by applying the corresponding weights.for each node output is given by[$x_{1i} \dots x_{ki}$]^T,[$w_{1i\dots}w_{ki}$].F_i is the activation function.X_i represents input from other nodes and w_i is weight of each connection.The structure of the Neural Network may also be viewed from the perspective of matrices.input and weight on the arcs in to node i are [$x_{1i,}x_{2i\dots}x_{ki}$] , [$w_{1i,}w_{2i\dots}w_{ki}$] .There is one output value from node i ,$y_{i,}$ which is propagated to all output arcs during the propagation process.using summation to combine the inputs then the output of a node is

$$Y_i = f_i(\textstyle\sum_{j=1}^{k} \quad w_{ji}\, x_{ji})$$

$$Y_i \;= f_i\, (w_{1i\dots}w_{ki})(\, x_{1i\dots}x_{ki}\,)$$

the output of each node i in the Neural Network is based on the function $f_{i.}$activation function associated with it .an activation function may also called a firing rule. The input to a neuron fires axon. The firing rule deal with binary output,the output[8] is either 0 or 1.the functions associated with each node in threshold function.activation function f_i is applied to the input values$\{x_{1i,}x_{2i\dots}x_{ki}\}$, $\{w_{1i,}w_{2i\dots}w_{ki}\}$.these inputs are usally combined in a sum of products form s=($wh_i\, xh_i$).if a bias input exists s=w_{0i} + ($wh_i\, xh_i$).activation functions f_i(s) at node i .Activation function may be unipolar with values[0,1].the output value is either a 0 or 1 depending on the sum of the products of the input values and their associated weights.The 1 value may be replaced by using constant.A Variation of this tansig threshold function is a linear threshold function with the linear threshold function,shown in figure 3 the value of activation function increases gradually from the low value to the high value function is

$$S_i = \textstyle\sum_{j=1}^{k} \quad w_{ji}\, x_{ji} - \theta$$

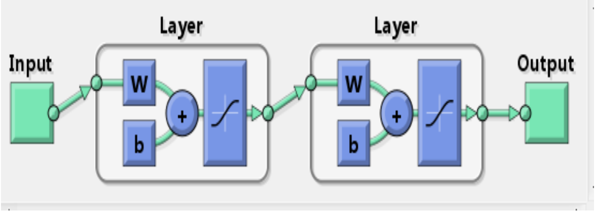

Figure 3 for tansig function

As with the threshold function[7] the value may be between -1 and 1 or 0 and 1.nodes in Neural Networks often have an extra input called a Bias.the Bias value of 1 input on an arc with a weight of θ.The effect of the bias input is move to the activation function on the x axis by a value of θ.Thus a weight of theta becomes a threshold of θ.

V.Error Detection

Neural Network Hamming code N represents the Length of the codeword,M represents the number of rows in the check matrix H.Input is connected with ach Neuron of layer1 and layer2 represent the position of the error of the Received word.Elements in the given check matrix are used as weight at the connections of the n input [2]with the nodes of the first layer.let H be the check matrix of ith row and jth Column of H.The weight w_{ij} $x_{ij} = h_{ij}$ where $1 \leq n$ and $1 \leq j \leq m$.x_{ij} the ith neuron of the input layer with the jth neuron of layer1 and w_{ij} denotes the weight of the connection of the ith neuron layer2 and weights assigned by the elements of the check matrix H bipolar form

$$fi\ (\,S\,) \;=\; \begin{cases} 1 & \text{if}\quad S\;>\;T \\ 0 & \text{if}\quad S\;<\;T \end{cases}$$

The bit positions in Hamming code word can be numbered from 2^{i-1}. Any number whose number is a power of 2 is a check bit and the remaining positions are data bits.check bit and data bits are specified by a parity check matrix in figure. First check all the parity bits.if all the bits have even parity then the received word is assumed to be correct otherwise error occurred in the bit .Hamming codes generalize the concept of a parity check bit. We encode the message $x_1,x_2,\dots x_n$ as $x_1,x_2,\dots x_{k+1}$ where x_{k+1}=($x_1,x_2,\dots x_{k+1}$) mod 2. Where the last n-k bits $x_{k+1}\dots x_n$ are parity check bits.let G be a k*n matrix that begins with the k*k identity matrix I_k.that is G=(I_k|H) where A is a k*(n-k) matrix .we encode x as E(X)=X(G). A Hamming code of order r where r is a positive integer, is a code generated when we take as parity check matrix H an r × (2^{r-1}) matrix with columns that are all the 2^{r-1} nonzero bit strings of length r in any order such that the last r columns form the identity matrix.So G=H*R.This produces the codeword 0010 and 0100

A.Example

Hamming code 4 bits X_1= 0010 and X_2= 0010 calculated as m*n matrix The codeword in a binary code generated by a Generator Matrix G by taking all possible linear combinations of the rows of G .Linear codeword y1,y2 generated by the Generator Matrix g.then there are bit strings x1 and x2 such that E(x1)=y1 and E(x2)=y2 where E(x)=XG. The weight of a codeword X denoted by w(x) in a Binary Code.The weight of a codeword x, denoted by w(x), in a binary code is the number of 1s in this codeword[3]. G1 and G2 results are shown in table 4.

X_1= 0010

$$G_1 = \begin{bmatrix} 0001000 \\ 0100100 \\ 0100010 \\ 0110001 \end{bmatrix} * \begin{bmatrix} 0101010 \end{bmatrix}$$

VI. Analysis Result

Hamming code algorithm of (7,4) matrix results are given in the tables.table shows inputs of check matrix and recived code values produce the output of 7 bit code word.theta can check the error in the position of the bit using parity bit.

Table 4 shows Hamming code check matrix

Data	2	4		2	4		2	4		2	4
w11	**0**	**0**	w12	**0**	**0**	w13	**0**	**1**	w14	**0**	**1**
w21	**0**	**1**	w22	**1**	**0**	w23	**1**	**0**	w24	**1**	**0**
w31	**0**	**0**	w32	**0**	**0**	w33	**0**	**0**	w34	**1**	**1**
w41	**1**	**1**	w42	**0**	**0**	w43	**0**	**0**	w44	**0**	**0**
w51	**0**	**0**	w52	**1**	**1**	w53	**0**	**0**	w54	**0**	**0**
w61	**0**	**0**	w62	**0**	**0**	w63	**1**	**1**	W64	**0**	**0**
w71	**0**	**0**	w72	**0**	**0**	w73	**0**	**0**	w74	**1**	**1**

Table 5 shows Received Codeword of 24 bits data 0010 and 0100

	V1	V2	V3	V4	V5	V6	V7
2	**0**	**1**	**0**	**1**	**0**	**1**	**0**
4	**1**	**0**	**0**	**1**	**1**	**0**	**0**

Sum of weighted inputs of layer1 N(0010) bits codeword as follows

$n_{1=}0.0+0.1+0.0+1.1+0.0+0.1+0.0=1$
$n_{2=}0.0+1.1+0.0+0.1+1.0+0.1+0.0=1$
$n_{3=}0.0+1.1+0.0+0.1+0.0+1.1+0.0=0$
$n_{4=}0.0+1.1+1.0+0.1+0.0+0.1+1.0=1$

Sum of weighted inputs layer2 M(0010) bits codeword as follows

m_1=-1.1+-1.1+-1.1+-1.1=-4
m_2=-1.1+1.1+1.-1+1.1=0
m_3=-1.1+-1.1+-1.0+1.1=1
m_4=1.1+-1.1+-1.-1+-1.1=0
m_5=-1.1+1.1+-1.-1+-1.1=0
m_6=-1.1+-1.1+1.-1+-1.1=-4
m_7=-1.1+-1.1+-1.-1+1.1=0

Activation function threshold θ=4 output of the neurons are

$f(m_1)=0$
$f(m_2)=0$
$f(m_3)=1$
$f(m_4)=0$
$f(m_5)=0$
$f(m_6)=0$
$f(m_7)=0$

Hamming code 4 bit 0010 detected error occur in third position of the bit.

X2=0100

$$G_{2=}\begin{bmatrix}0101000\\0000100\\1000010\\1010001\end{bmatrix}*\begin{bmatrix}1001100\end{bmatrix}$$

Sum of weighted inputs of layer1 N(0100) bits codeword as follows

$n_{1=}0.1+1.0+0.0+1.1+0.1+0.0+0.0=1$
$n_{2=}0.1+0.0+0.0+0.1+1.1+0.0+0.0=1$
$n_{3=}1.1+0.0+0.0+0.1+1.1+1.0+0.0=0$
$n_{4=}1.1+0.0+1.0+0.1+0.1+0.0+0.1=1$

Sum of weighted inputs layer2 M(0010) bits codeword as follows

m_1=-1.1+-1.1+1.1+1.1=0
m_2=1.-1+-1.1+-1.1+-1.1=-4
m_3=-1.-1+-1.1+-1.1+1.1=0
m_4=-1.-1+-1.1+-1.1+-1.1=-4
m_5=-1.1+1.1+-1.1+-1.1=0
m_6=-1.1+-1.1+1.1+-1.1=0
m_7=-1.-1+-1.1+1.1+-1.1=-1

Activation function threshold θ=4 output of the neurons are

$f(m_1)=0$
$f(m_2)=0$
$f(m_3)=0$
$f(m_4)=0$
$f(m_5)=0$
$f(m_6)=0$
$f(m_7)=1$

Hamming code 4 bit 0100 detected error occur in third position of the bit.

Table 6 shows activation function of the codeword N.

F(N)	N	
	2	4
n1	**1**	**0**
n2	**1**	**1**
n3	**0**	**1**
n4	**1**	**1**

Table 7 shows M activation function and error position

	M		F(M)		θ		Error in bit position	
	2	4	2	4	2	4	2	4
m1	**-4**	**0**	**0**	**0**	**0**	**0**		
m2	**0**	**-4**	**0**	**0**	**0**	**0**		
m3	**1**	**0**	**1**	**0**	**1**	**1**	**3**	
m4	**0**	**-4**	**0**	**0**	**0**	**0**		
m5	**0**	**0**	**0**	**0**	**0**	**0**		
m6	**-4**	**0**	**0**	**0**	**0**	**0**		
m7	**0**	**-1**	**0**	**1**	**0**	**0**		**7**

VII.ERROR CORRECTION

Let H be the Hamming Code defined by the Check Matrix. If the word is received code word(0101010) r1 with one error was corrected by changing the 3rd bit based on the value of(0111010); the parity bit was also changed to one and the received word(1001100) r2 with one error was corrected by changing the 7th bit based on the value of(1000100) the error was detected and corrected.

VIII.CONCLUSION

In this paper we have discussed about Hamming Code Error Detection in credir card information using Neural Network Techniques .Neural Network Framework for error detection and examples are shown for illustrative Purposes. Hamming code corrects the errors during the transmission. one common type of error detecting code is the parity check code used in credit card. Hamming developed technique for detecting and correcting single bit errors in transmitted data. in this article describes hamming code detect the error occur in which position of the bit and correct the error.

REFERENCES

[1] E. Berlekamp, ed., Key Papers in the Development of Coding Theory, IEEE Press, 1974.

[2] R. Hamming, Coding and Information Theory, Prentice-Hall, 1980.

[3] R. Hill, A First Course in Coding Theory, Oxford University Press, 1986.

[4] V. Pless, Introduction to the Theory of Error-Correcting Codes, M.I.T. Press, 1972.

[5] S. Vanstone and P. van Oorschot, An Introduction to Error Correcting Codes with Applications, Kluwer, 1989.

[6] James A. Freeman and David M. Skapura."Neural Networks Algorithms, Applications and Programming Techniques", pp 115-116, 1991.

[7] Jacek M. Zurada, "Introduction to Artificial Neural Systems", pp 32-36, 2006.

[8] Rumelhart, D., G. Hinton, and R. Williams. 1988. Learning internal representations by error prop agation. In Neurocomputing, edited by J. Anderson and E. Rosenfeld, 675-695. Camb ridge, MA: MIT Press.

[9] Russell, I. 1991. Self-organization and adaptive resonan ce theory networks. In Proceedings of the Fourth Annual Neural Networks and Paral lel Processing Systems Conference, edited by Samir I. Sayegh, 227-234. Indianapol is, IN: Indiana University-Purdue University.

[10] Shea, P., and V. Lin. 1989. Detec tion of explosives in checked airline baggage using an artificial neural system. International Journal of Neural Networks 1 (4) (October 1989): 249-253.

[11] Stubbs, D. 1990. Three applications of neurocomputing in biomedical research. Neurocompu ting 2: 61-66.Wasserman, P. 1989. Neural Network Computing. New York: Van Nos trand Reinhold.

[12] Neural Networks A Comprehensive Foundation, Simon Haykin, Prentice-Hall, 1999

[13] Networks for Pattern Recognition , C.M. Bishop, Oxford University Press, 1996

[14] R.Dhanapal ,P.Gayathiri Credit Card Fraud Detection Using Decision Tree For Tracing Email and IP IJCSI International Journal of Computer Science Issues,Vol.9,Issue 5,No 2,ISSN 2012 ISSN(Online):1694-0814