Arabic Text Encryption Using Artificial Neural Networks

Oday Kamil Hamid
Computer Techniques Engineering Department, Dijlah University College
Email:qwera9222@gmail.com

Received on:23/8/2015 & Accepted on:4/2/2016

ABSTRACT

This research aims to build a cipher system using back propagation Algorithm with artificial neural network to encrypt any Arabic text and to prevent any data attack during the transition process. Encryption information holds four stages:

1) A neural network was trained by using back propagation algorithm to encrypt the whole input Arabic text and grasp final weights and consider these weights as a public key.
2) Training a second neural network by using back propagation algorithm to decrypt the input Arabic text of first stage and grasp weights and consider the weights as a private key.
3) Encrypt any Arabic text by using the weights obtained from first stage.
4) Decrypt the Arabic text from third stage by using the weights obtained from second stage.

The four stages are achieved prosperously for data encryption process and decryption.
This work is executed by using Matlab program version 7 and Notepad++ for writing text because it supports Arabic numbers under windows 7 as operating system.

Keywords: symmetric key, a symmetric key, cryptography, public, private, back propagation.

INTRODUCTION

Encryption is the way of talking to someone while other people are listening, but without understanding what you are saying. It can also be used to protect data in storages as well as to detect active attacks, such as message or file modification [1]. Encryption plaintext result in unreadable gibberish called cipher text, the process of reverting cipher text to its original plaintext is called decryption [2]. In order to recover the contents of an encrypted text easily, a correct decryption key is required. The more complex encryption algorithm is, the more difficult to monitor on the communications without access to the key [3]. Figures (1and 2) represent the encryption and decryption types [4].

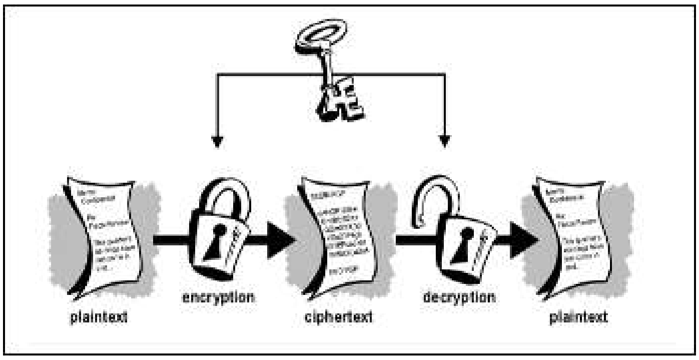

Figure (1): Secret Key Cryptography (SKC)

https://doi.org/10.30684/etj.34.5A.7
2412-0758/University of Technology-Iraq, Baghdad, Iraq

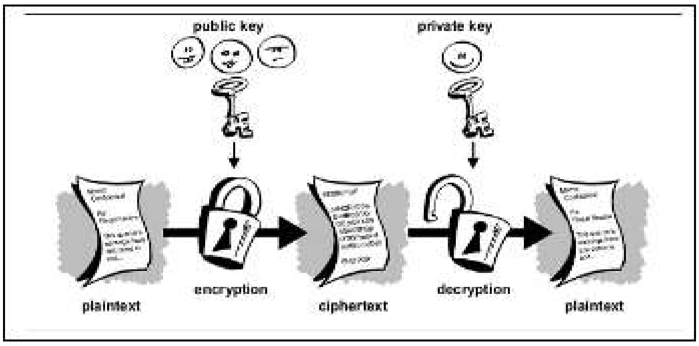

Figure (2): Public Key Cryptography (PKC)

Theory

Cryptography is a technique used to hide the meaning of a message and is derived from the Greek word krypton (hidden). Cryptography should ensure that message could not be read. Typically the sender and receiver agree upon a message scrambling protocol beforehand and agree upon methods for encrypting and decrypting messages [5].

Basics of neural network

Artificial Neural Networks are relatively crude electronic models based on the neural structure of the brain. The brain basically learns from experience. It is natural proof that some problems beyond the scope of current computers are indeed solvable by small energy efficient packages. This brain modeling also promises a less technical way to develop machine solutions. This new approach to computing also provides a more graceful degradation during system overload than its more traditional counterparts. These biologically inspired methods of computing are thought to be the next major advancement in the computing industry [6]. Figure (3) presents the model of an artificial neuron [7].

Figure (3): Model of an Artificial Neuron.

Neuron output signal is given by the following relationship:

$$O_k = f\left[\sum_{j=1}^{n} w_{jk}\, x_j + \theta_k\right] \qquad \text{.....(1)}$$

Where

$f\,(w, x, \theta\,)$ is the activation function

w: is the synaptic weight.

x: is the input signal .

Types of Activation Signals Functions

1) Binary step.
2) Hard limiter function.
3) Threshold logic function.
4) Sigmoidal tangent function

The sigmoid function is by far the most common form of activation function used in the construction of artificial neural networks. As an example of the sigmoid function is the log sigmoid function, defined by:

$$f(x) = \frac{1}{1+\exp(-ax)} \qquad \text{....(2)}$$

Where (*a*) is the slop parameter of the sigmoid function [8].

Architecture of neural networks

To characterize a given ANN, it is necessary to specify the number of neurons, how they are interconnected and the processing that takes place throughout the network. The manner in which the neurons of neural network are structured is intimately linked with the learning algorithms used to train the network. In general, be identified: single layer feed forward networks, multi-layer feedback networks, and recurrent networks [9]

Forward networks

Feed-forward ANNs allow signals to travel one way only; from input to output. There is no feedback (loops) i.e. the output of any layer does not affect that same layer. Feed-forward ANNs tend to be straight forward networks that associate inputs with outputs. They are extensively used in pattern recognition. [10].

Feedback networks

Feedback networks can have signals travelling in both directions by introducing loops in the network. Feedback networks are very powerful and can get extremely complicated. Feedback networks are dynamic; their 'state' is changing continuously until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found [10].

The Back-Propagation Algorithm

The original algorithm used for training a MLP is the BP algorithm, which is an iterative gradient algorithm, designed to minimize the mean-squared error between the desired output and the actual output for a particular input to the network [11].

The learning rate (η) determines the portion of weight needed to be adjusted. However, the optimum value of η depends on the problem. The momentum (μ) determines the fraction of the previous weight adjustment that is added to current weight adjustment. It accelerates the network convergence process.

During the training process, the learning rate and the momentum are adjusted to bring the network out of its local minima and to accelerate the convergence of the network.

The algorithm of the error back-propagation training is given below:

BP Algorithm

This algorithm is to perform:

1- Initializes the values of network weight.
2- Sums weighted input and apply activation function to compute output of hidden layer.

$$h_j = f[\ \sum_i x_i\, w_{ij}\ ] \qquad \ldots\ldots (3)$$

h_j : The actual output of hidden neuron *j* for input signals *x*.

x_i : Input signal of input neuron (i*)*.

w_{ij} : Synaptic weight between input neuron *i* and hidden neuron *j*.

f : The activation function.

3- Sums weighted output of hidden layer and apply activation function to compute output of output layer.

$$y_k = f[\ \sum_j h_j\, w_{jk}\ ] \qquad \ldots\ldots\ldots (4)$$

Where

y_k : The actual output of output neuron *k*.

w_{jk} : Synaptic weight between hidden neuron *j* and output neuron *k*.

4- Computes back propagation error.

$$\delta_k = (d_k - y_k) f'[\ \sum_j h_j\, w_{jk}\ ] \qquad \ldots\ldots (5)$$

Where

f′ : The derivative of the activation function.

d_k : The desired of output neuron *k*.

5- Calculates weight correction term.

$$\Delta w_{jk}(n) = \eta \delta_k h_j + \mu \Delta w_{jk}(n-1) \qquad \ldots\ldots (6)$$

6- Sums delta input for each hidden unit and calculate error term.

$$\delta_j = \sum_k \delta_k w_{jk}\, f'\left(\sum_i x_i w_{ij}\right) \qquad \ldots\ldots (7)$$

7- Calculates weight correction term.

$$\Delta w_{ij} = \eta \delta_j x_i + \mu \Delta w_{ij}(n-1) \qquad \ldots\ldots (8)$$

8- Updates weights.

$$w_{jk}(new) = w_{jk}(old) + \Delta w_{jk} \qquad \ldots (9)$$

$$w_{ij}(new) = w_{ij}(old) + \Delta w_{ij} \qquad \ldots\ldots (10)$$

9- Repeats step (2) for a given number of error.

$$MSE = \frac{1}{2F}\left[\sum_F \sum_k \left(d_k^F - y_k^F\right)^2\right] \qquad \ldots\ldots (11)$$

Where

F: The number of patterns in the training set.

10- Ends [12].

Training a Network

The network weights will all start at random values and the training process starts as shown in figure (5) [13].

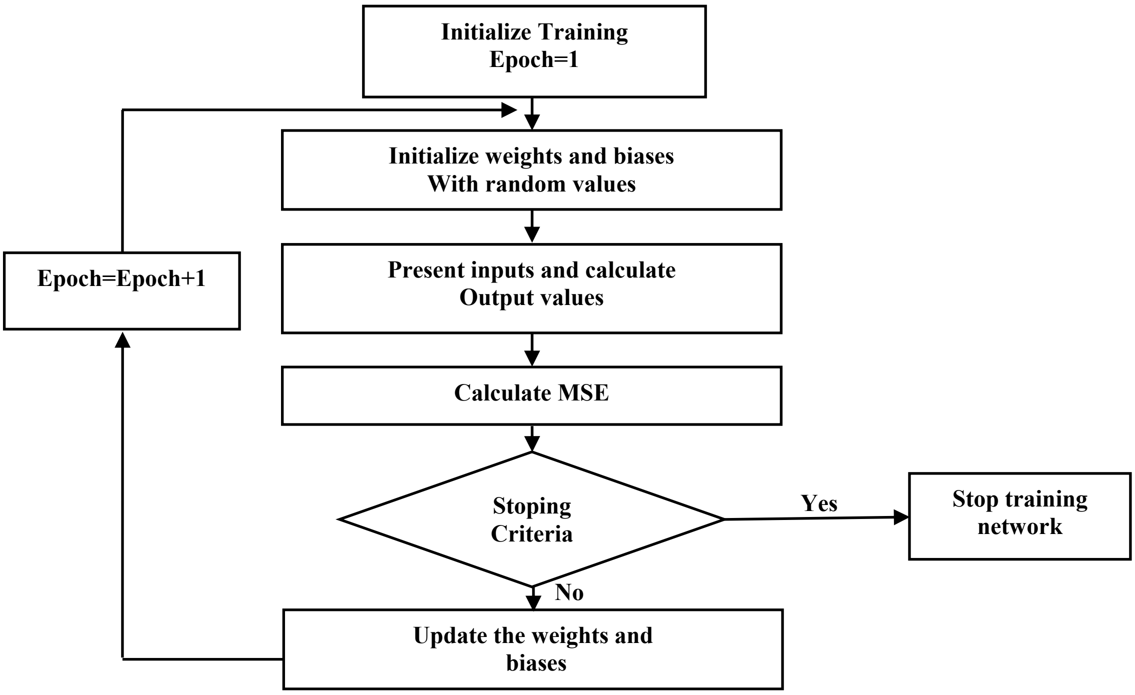

Figure (5): Basic training flow chart

Practical Work

A cryptosystem is a way of encoding and decoding messages so that only certain people are able to read them. Table (1) presents the Matlab Unicode conversion which includes numbers from (0 to 255) by using the instruction unicode2native and using NOTEPAD++ for writing the Arabic text because it supports Arabic numbers with script (windows 1256) similar to the script for Matlab program.

Table (1): Arabic Unicode Table

Dec	Char	Dec	Char	Dec	Char
59	;	211	س	247	÷
191	?	212	ش	176	٠
193	ء	213	ص	177	١
199	ا	214	ض	178	٢
195	أ	216	ط	179	٣
196	ؤ	217	ظ	180	٤
197	إ	218	ع	181	٥
198	ئ	219	غ	182	٦
199	ا	221	ف	183	٧
200	ب	222	ق	184	٨
201	ة	223	ك	185	٩
202	ت	225	ل		
203	ث	227	م		
204	ج	228	ن		
205	ح	229	ه		
206	خ	230	و		
207	د	237	ي		
208	ذ	43	+		

209	ر	45	-		
210	ز	215	×		

Training Procedure for Encryption

In this step the important computer keyboard symbols, mostly used in writing letters are taken. Their number is 51 then enters this text to the suggested program.

The algorithm of the program steps will be as follows:

The text is converted to Arabic Unicode then converted to binary with 8bit; each symbol in the text is represented by a code containing 51 digits for the first symbol. All digits are zeros except the first digit will be one and second symbol all digits will be zeros except second digit will be one and continue this procedure for all symbols so we create a diagonal matrix its dimensions are (51*51) as shown in figure (6).

In this procedure a neural network with three layers is used (input layer, Hidden layer, Output layer) with back propagation learning algorithm, for first layer we used 8 neurons equal to binary bits for the text and 20 neurons (by trial and error) for hidden layer and 51 neurons for output layer equal to the desired output that we want with an error rate of 10^-5. After running this program, the neural network will be trained and calculate eights between input layer and hidden layer and weights between hidden layer and output layers and stop training until the error approaches 10^-5.

```
100000000000000000000000000000000000000000000000000
010000000000000000000000000000000000000000000000000
001000000000000000000000000000000000000000000000000
000100000000000000000000000000000000000000000000000
000010000000000000000000000000000000000000000000000
000001000000000000000000000000000000000000000000000
000000100000000000000000000000000000000000000000000
000000010000000000000000000000000000000000000000000
.
.000000000000000000000000000000000000000000000000010
000000000000000000000000000000000000000000000000001
```

Figure (6): Representation of each symbol

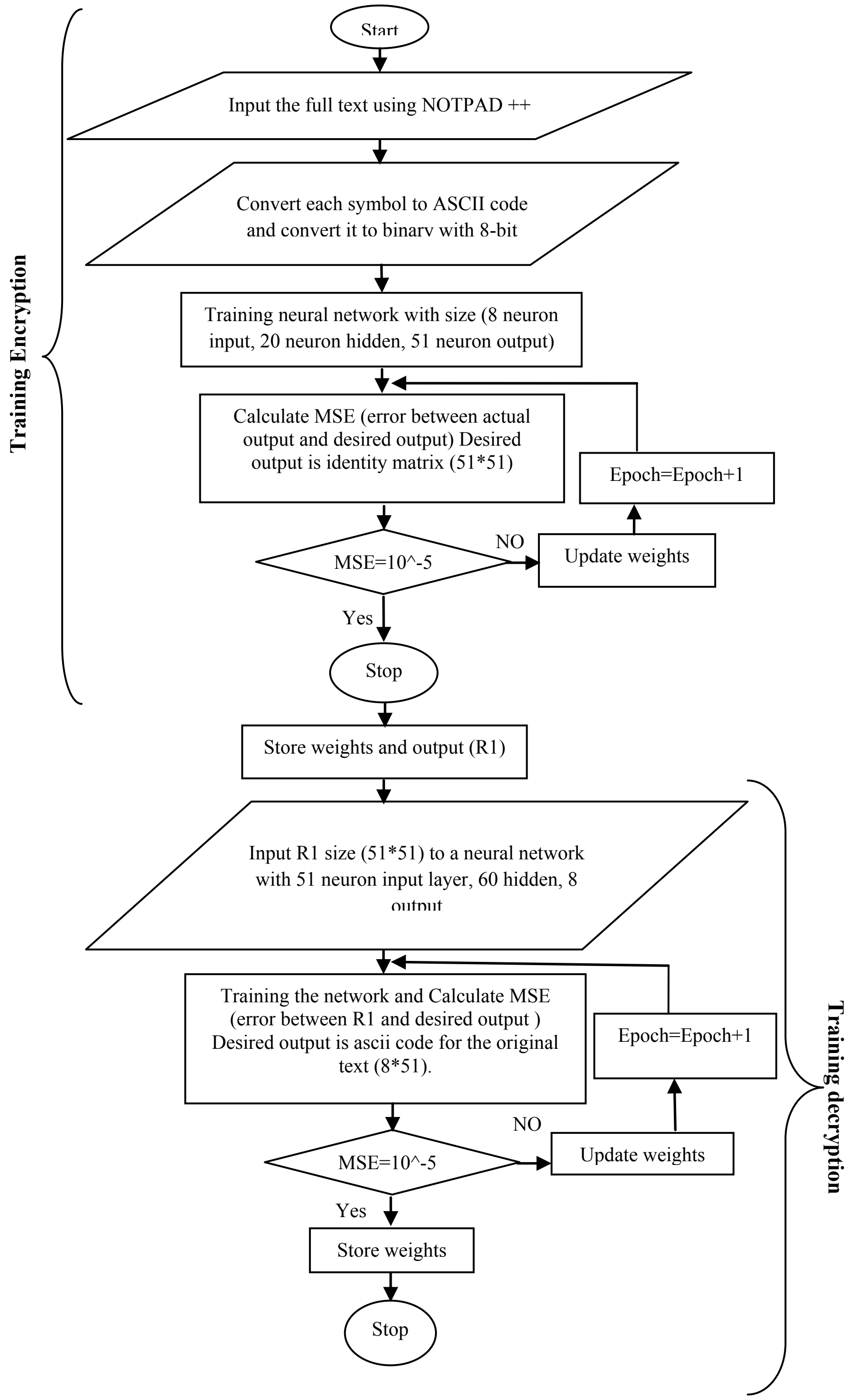
Training Encryption
Start
Input the full text using NOTPAD ++
Convert each symbol to ASCII code and convert it to binary with 8-bit
Training neural network with size (8 neuron input, 20 neuron hidden, 51 neuron output)
Calculate MSE (error between actual output and desired output) Desired output is identity matrix (51*51)
Epoch=Epoch+1
MSE=10^-5
NO
Update weights
Yes
Stop
Store weights and output (R1)
Input R1 size (51*51) to a neural network with 51 neuron input layer, 60 hidden, 8 output
Training the network and Calculate MSE (error between R1 and desired output) Desired output is ascii code for the original text (8*51).
Epoch=Epoch+1
MSE=10^-5
NO
Update weights
Yes
Store weights
Stop
Training decryption

Fig. (7): Algorithm for Training Encryption and Decryption

Training Procedure for Decryption

After executing the encryption training program we enter the output matrix that has dimensions (51*51) to the training decryption program. In this phase we use neural network with Back Propagation with three layers; input layer contains 51 neurons, hidden layer contains 60 neurons and output layer contains 8 neurons. In this phase the desired output must be a matrix with dimensions (8*51) equal to the input original text of computer keyboard entered in the first phase. Error rate in this phase is also equal to 10^-5. At the end the final weights between layers will be saved. Figure (7) presents the algorithm for training procedure for encryption and decryption phases.

Encryption Testing Phase

Any text written in Arabic and then entered in to the Special program for the testing phase, which consists of neural network of three layers; the same size of the network used in the training phase for encryption, and using weights extracted from the stage of the encryption training phase and after the implementation of this program the resulting matrix will be sent, which represents the cipher text and also send the size number of the original text (row and column) that also encrypted by changing its size.

Decryption Testing Phase

After receipt of the encrypted message and the rows and columns file, these files will be entered to the decryption program. Which consists of neural network with three layers equal to the size of the network used in the decryption training phase and after running the program, the original text will be extracted. Figure (8) presents the algorithm for testing encryption and decryption phase.

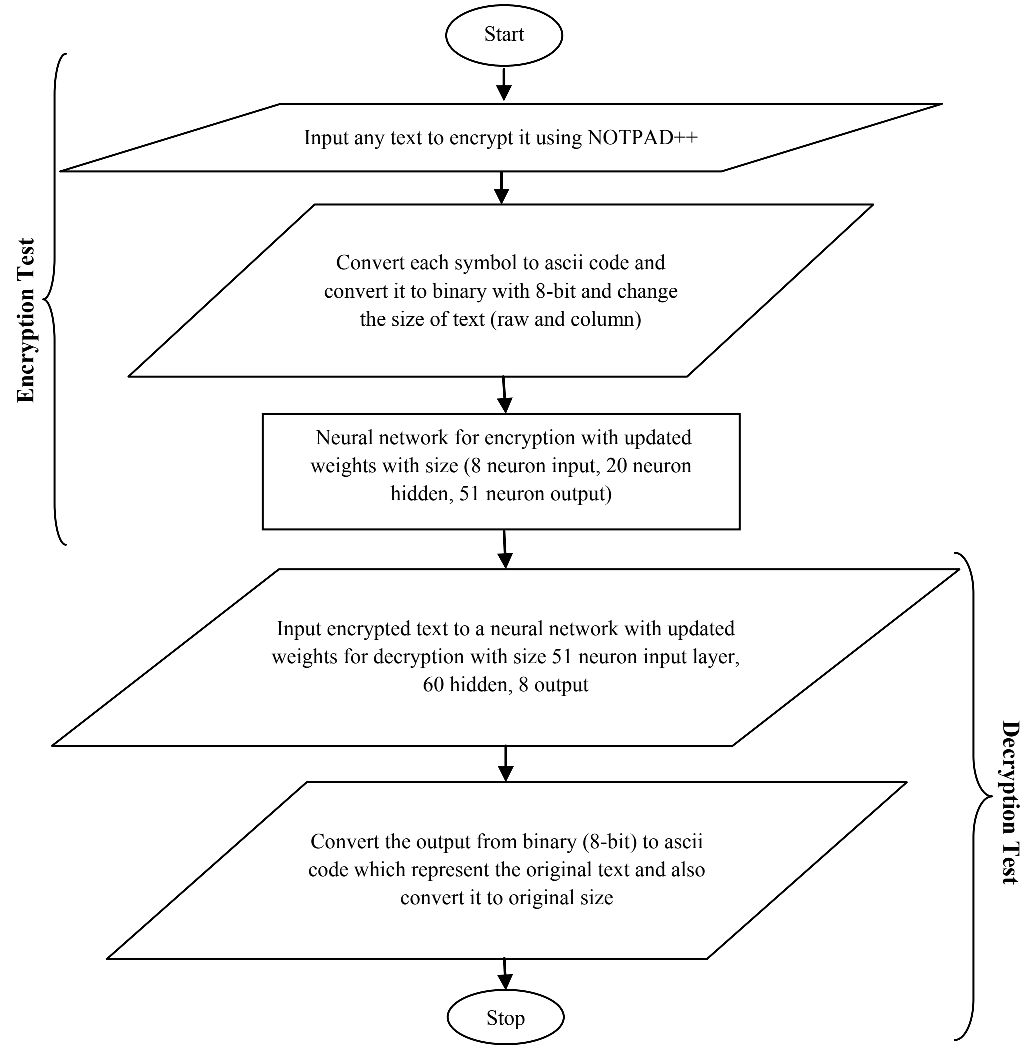
Start
Input any text to encrypt it using NOTPAD++
Convert each symbol to ascii code and convert it to binary with 8-bit and change the size of text (raw and column)
Neural network for encryption with updated weights with size (8 neuron input, 20 neuron hidden, 51 neuron output)
Encryption Test
Input encrypted text to a neural network with updated weights for decryption with size 51 neuron input layer, 60 hidden, 8 output
Convert the output from binary (8-bit) to ascii code which represent the original text and also convert it to original size
Decryption Test
Stop

Figure (8): Algorithm for Testing Encryption and Decryption

Results

The results of the programs as appear in the command window of matlab program will be explained.

The Result for Training Phase Encryption

After entering the text in the program the neural network will be trained for all symbols in the text which represent the most important symbols in the keyboard used to write any letter. The text is

؟ءاأؤإئ بةتثجحخدذرزسشصضطظعغفقكلمنهوي=+-÷×٠١٢٣٤٥٦٧٨٩

The network used is shown in figure (9):

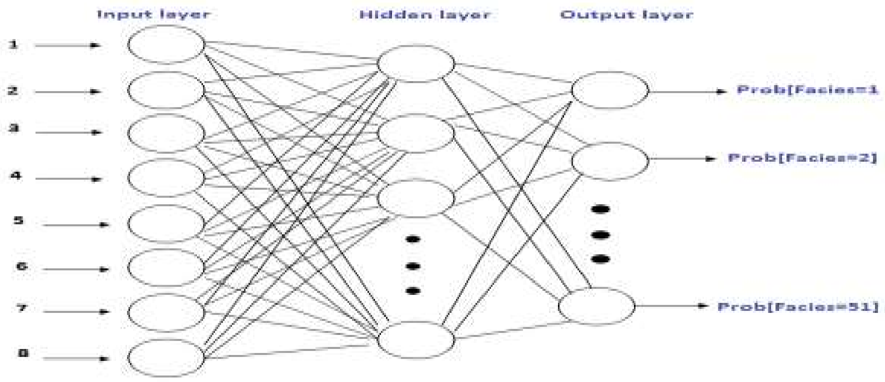

Figure (9): Encryption Training Phase network

The output of this program is shown in Appendix (A). The real output of this step is a matrix with size (51*51) because it is large only the first column was printed. This column represents the code for the first symbol in the text (?).And figure (10) shows numbers of epoch till the output reach error rate 10^-5(error between real output and desired output).

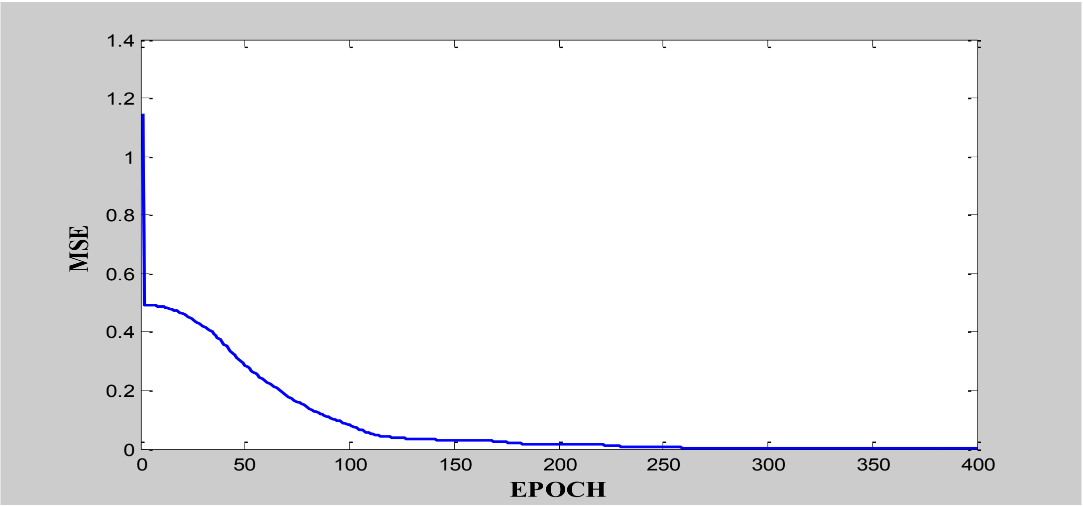

Figure (10): Encryption Training

The Result of Training Phase Decryption

After executing the encryption training program, the output matrix that contains dimensions (51*51) will be entered to the training decryption program. In this phase the desired output must be a matrix with dimensions (8*51) equals to the input original text of computer keyboard that entered in the first phase. The network used in this phase is shown in figure (11):

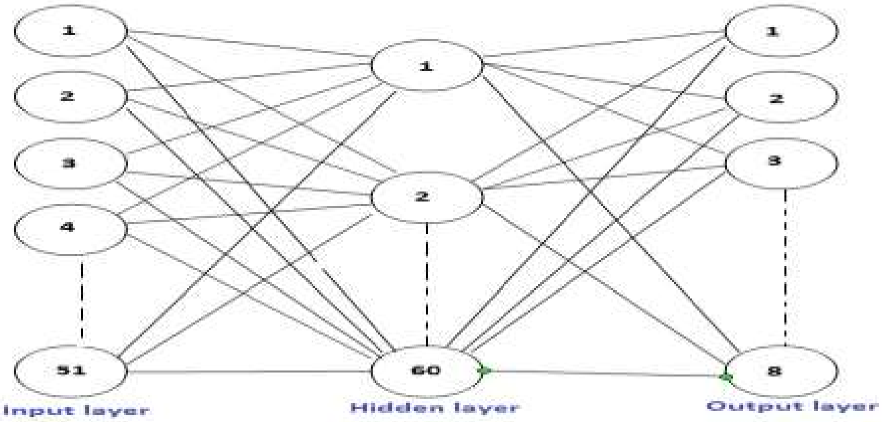

Figure (11): Decryption Training Phase network

The output of this program is shown in Appendix (B). The real output of the program is a matrix with size (8*51) and also only the first column was printed which represents the decryption of symbol (?) before converting it to binary number (ascii code).

Figure (12) shows numbers of epoch till the output reach error rate 10^-5(error between real output and desired output).

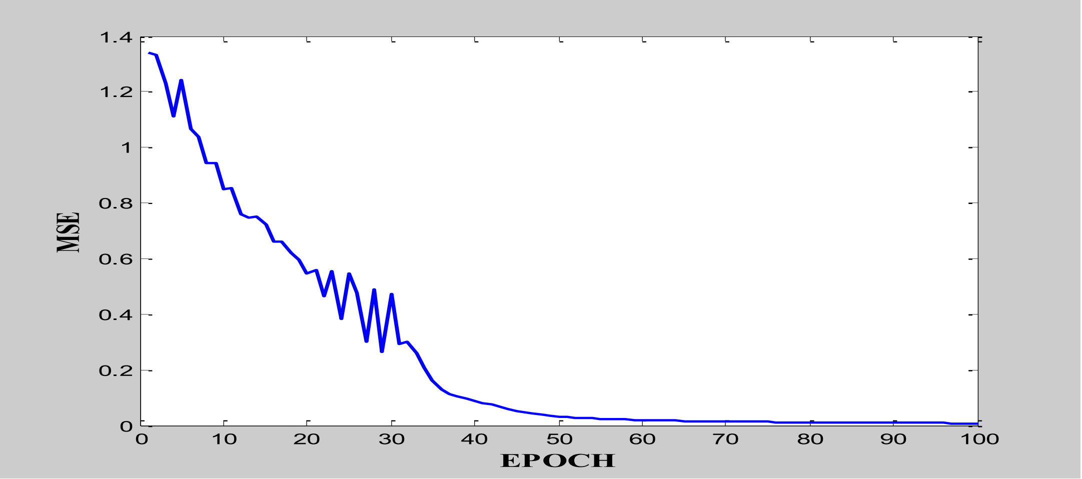

Figure (12): Decryption Training

The Result of Encryption Testing Phase

The testing phase, which consists of neural network with three layers, the same size of the network, is used in the training phase for encryption, which comprise 8 neurons input layer and 60 neurons hidden layer and 51 neurons output layer, while using weights extracted from the stage of the training phase encryption. After the implementation of this program, the resulting matrix will be send, which represents the cipher text and also send the number of rows and columns of the original text after changing it. The input text is as shown below:

السلام عليكم

هذه الرسالة سرية للغاية
شكرا جزيلا

The output of this program is shown in Appendix (C).The real output of this step is a matrix with size (51*138) which represents the encryption code of input text with 138 symbols. Also only the second column was printed which represents the encryption code of Arabic symbol(ل).In this step the output matrix represents the code of each symbol in the input text with one column (the letters and symbols of input text will be arranged in one column).

The Result of Decryption Testing Phase

After receiving the encrypted message, the rows and columns files and executing the decryption program, the output of neural network will be binary numbers 8-bit which represents every symbol in the text and convert it to decimal to get Unicode then convert it to original text as shown below (this text token from **work space** of matlab program).

السلام عليكم

هذه الرسالة سرية للغاية
شكرا جزيلا

The binary numbers and Unicode matrices are shown in Appendix (D). The actual output of this step is a matrix with size (8*138) which represents the decryption (ASCII code) of the encrypted text. It is arranged as one column (each column represent a symbol in the original text), here it shows the importance of knowing the original size of input text that is also encrypted (number of columns and rows of original text before encrypt it).only a second column was printed which represents decryption (ASCII code) of encrypted symbol (ل) as shown in appendix(D).

Discussion

The most important discussions that have been reached are:-

1-The proposed algorithm of NNT gives us 100% results for encryption and decryption of the text.

2-The users of BP suffer from many successful demonstrations of its power. Among which is the slow rate of convergence (long training times) and getting stuck in local minima.

3-BP requires the specification of a number of parameters such as learning rate momentum rate, and number of neurons in the hidden layer and weights. The success and speed of training depend on these parameters. They are chosen by intuition and time consuming trial and error.

4-The employment of the programming language (Matlab) becomes wealthy in the shorthand on the program volume and the ease of its correction.

Appendix (A)

0.9971
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0007
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0001
0.0001
0.0002
0.0011
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0012
0.0015
0.0006
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0003
0.0004
0.0006
0.0001

Encryption of Character (?)

Appendix (B)

	ASCII CODE
0.9958	1
0.0039	0
0.9975	1
0.9982	1
0.9980	1
0.9986	1
0.9981	1
0.9987	1

191 (Decimal)

Decryption of character (?)

Appendix (C)

0.0000
0.0010
0.0000
0.0001
0.0000
0.0000
0.0000
0.0014
0.0004
0.0012
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0014
0.0004
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.9972
0.0014
0.0002
0.0015
0.0003
0.0011
0.0000
0.0000
0.0000
0.0002
0.0000
0.0000
0.0011
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000

Encryption of Character (ل)

Appendix (D)

	ASCII CODE
0.9983	1
0.9980	1
0.9975	1
0.0014	0
0.0015	0
0.0018	0
0.0017	0
0.9985	1

225 (Decimal)

Decryption of character (ل)

REFRENSS

[1].Marshall D.H., Sushil Jajodia, Harold J. P "Information Security: An Integrated Collection of Essays", Essay 15 Cryptography, IEEE Computer Society Press Los Alamitos, California, USA.
http://www.acsac.org/secshelf/book001/book001.html#toc

[2].Kumar V.P., Premchand P.,Raghu S.,"Cryptographic Analysis and Security Issues On Cloud Computing". ACEC 2014, ISBN: 978-1-63248-029-3 doi: 10.15224/ 978-1-63248-029-3-176

[3].Sudha R.K, Sarma T.C., Prasad K.S.,"Text File Encryption Using FFT
in Lab View 8.6", IJRET: International Journal of Research in Engineering and Technology ISSN: 2319-1163, Sep-2012

[4].Indrayani I. Patle (M.Tech. 4th sem) , Prof.Jitendra Zalke (Assistant Professor) "Speed optimization of Cryptographic Algorithm Using Hardware-Software Co-design''.International Journal of Advancements in Research & Technology, Volume 2,Issue 5, M ay-2013 364 ISSN 2278-7763

[5]. Jacob M., "History of Computer Cryptography and Secrecy Systems"
http://www.dsm.fordham.edu/~mathai/crypto.html

[6].Jatin C., Hiteshi C., Abhimanyu V.,"Research Paper On Neural Networks"
2014 IJIRT , Volume1 ,Issue5,ISSN: 2349-6002

[7].Wikimedia Commons, "Artificial neural network". 2 December 2014.
http://commons.wikimedia.org/wiki/Artificial_neural_network

[8].Sami M.,Jabber M.,Faleh H.," Position Control For Flexible Joint Manipulator Using Artificial Neural Network". Diyala Journal of Engineering Sciences, Vol. 01, No. 01, December 2008.

[9].Sawsan S. A.,"Cryptography Using Artificial Neural Network". Department of Chemistry, College of Education/Ibn-Al-Haithem,Aldananer magazine,First assue

[10].Yashika Birdi , Tanjyot Aurora, Parul Arora ,"Study of Artificial Neural Networks and Neural Implants", International Journal on Recent and Innovation Trends in Computingand Communication, ISSN 2321 – 8169, Volume: 1 Issue: 4, 258 – 262

[11].ChinH.C., Ming C., L., "Classification of Underwater Signals Using Neural Networks".Tamkang Journal of Science and Engineering, Vol. 3, No. 1, pp. 31-48 (2000)

[12].Adil. A.H., Firas N.H., "Prediction o f the Point Efficiency of Sieve Tray Using Artificial Neural Network". Iraqi Journal of Chemical and Petroleum Engineering Vol.10 No.4 (December 2009) 57-62 ISSN: 1997-4884.

[13].ALI S., BILAL Z., "A Computationally Intelligent Expert System for Design Parameterization of Sub-Orbital Carrier Vehicles". Department Of Aerospace Engineering National University of Sciences & Technology (NUST), Publication Date 12/11/2014.