Neural Network and Back Propagation

Nikunj Agarwal
Pranveer Singh Institute of Technology
Kanpur, India

ABSTRACT

Humans and computers are suited to different types of tasks. Like, finding the sine of any number is difficult for humans, but easy for the computers. On the other hand, looking and identifying the image of a bird is very easy for us, as compared to the machines [1]. But nowadays, Artificial neural networks or ANNs are most commonly used in image processing. In fact, the recent deep learning algorithms have surpassed the human ability of image recognition. It is proved and stated that BPNN (Backpropagation Neural Networks) have a very high accuracy. A large part of recent success of neural network is explained on the basis of very high data availability. Thus, in B-P the network is trained with a lot of training examples so that it doesn't compromise with the efficiency of the system. By the end of this century, it is expected that computers will be training neural networks with as many neurons as the human brain has [1][2]. Although it is difficult to predict what the true abilities of artificial intelligence will be by then, our experience with computer vision should prepare us to expect something unexpected. So, with this attempt paper tries to give the idea of how image is recognized and how back propagation works.

I. INTRODUCTION

Artificial neural networks are the popular machine learning techniques that function exactly same as the neural network of the living body. ANNs are the most significant fields in today's world. They take the input, perform some operation and give the output. Each input to a neuron is scaled with a weight, which affects the function calculated at that unit. Similar to the living body neurons, external stimulus is needed for learning, here we provide the ANN with the set of training examples containing input-output pairs. These training data pairs are fed into the neural network by using the input representations to make predictions about the output labels. If there is some error in the output produced, then the suitable algorithms such as backpropagation are used to reduce the errors. So, when we provide different data sets of the same thing/ image, the machine eventually starts recognizing it. This ability of the machine to find out functions of the unseen dataset is called as Model Generalization.

II. HOW DOES MACHINE PERFORMS IMAGE RECOGNITION?

The answer to this problem is Neural networks. Neural network is made from the combination of two words Neural and Network. It basically tells us how the neurons are connected together and how to form the network. For initial purpose the neurons can be understood as the things that hold number basically between 0 & 1 and really not more than that. The network starts with, $28 \times 28 = 784$ neurons, each representing the grayscale value of the corresponding pixels (0 for black and 1 for white) and the grayscale number represents the Activation Number of that pixel, i.e., how much bright it would light up.

The digits belong to a dataset called MNIST (Modified National Institute of Standards and Technology). It contains 70,000 examples of digits written by human hands. Each of these digits is a picture of 28×28 pixels. So, in total each image of a digit has $28 \times 28 = 784$ pixels.

III. LAYERS

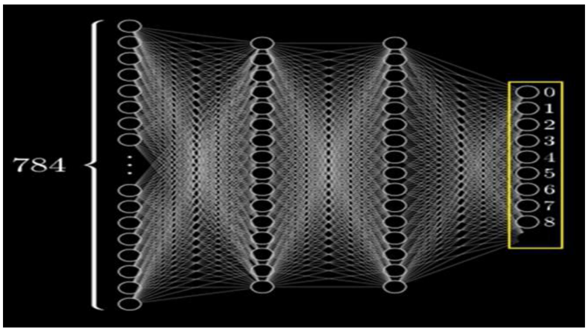

Fig 1. Showing the division of 784 pixels

All these 784 neurons make the first layer of the network, and the last layer contains 10 neurons represents the digits from 0 to 9, whereas the layers in the middle are called the hidden layers. There can be any number of hidden layers and they can have any numbers of neurons. The way the network operates, activation in one layer determines the activation of next layers.

Eg.- Since 9 can be broken down and can be written as follows.

Fig 2. Showing the writing pattern of 9

Thus, every neuron in 2nd to last layer corresponds to such features.

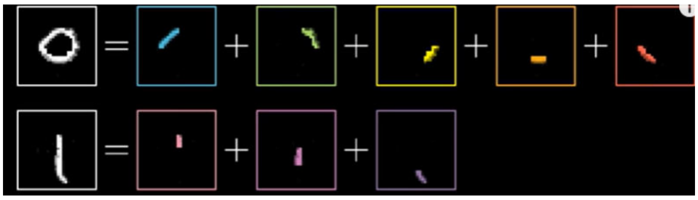

Fig 3. Showing neuron with corresponding features

The number ‘9’ can be broken down into two parts, a loop and a straight line. Also, the loop and the straight line can be further broken down into many parts as shown. These different parts form the layers of the neural network. Each part of the loop and the straight line is contained in a single neuron of the first hidden layer. Different neurons of the first layer merge together to form the loop and the straight line in the 2nd layer. And the loop and the straight-line merge together (in the 3rd layer) to form 9 and display in the output.

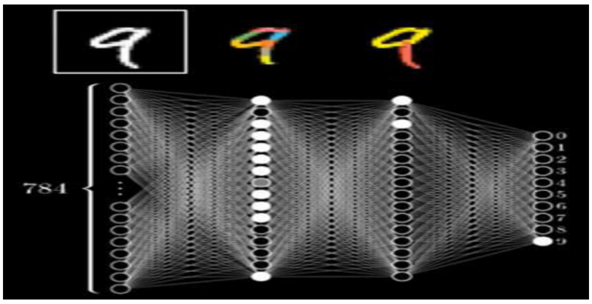

Fig 4. Showing the display of 9 in the output

IV. HOW DOES THE FIRST LAYER ACTIVATION ACTIVATE THE SECOND LAYER AND SO ON?

Each connection from one neuron to another is having its corresponding weight which may be either positive or negative and the 2nd layer neuron receives the value corresponding to the 1st layer neuron and its weight.

$$z = w_1 a_1 + w_2 a_2 + \cdots w_n a_n$$

This formula returns us the weighted sum and it might be any value. But for the activation purpose we need to compress it and bring it within 0 and 1. Thus we pass it through a sigmoid function.

$$y = \frac{1}{1 + e^{-z}}$$

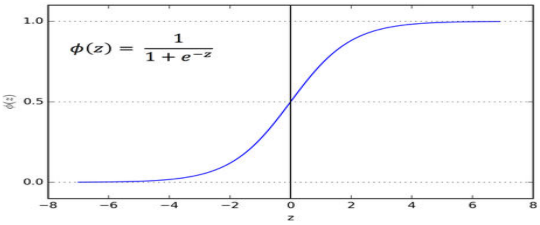

Fig 5. Showing the graph of sigmoid function

suppose we want the function be active when the value of the weighted sum is greater than “b”, i.e., we want some bias to be inactive, thus we subtract b from the weighted sum. Bias allows us to shift the activation function by adding a constant to the input.

$$z = w_1 a_1 + w_2 a_2 + \cdots w_n a_n - b$$

So, the weights tell what pixel pattern this neuron in the second layer is picking up and the bias tells how high the weighted sum needs to be before the neuron starts getting meaningfully active.

Every neuron in the 2nd layer (1st hidden layer) is connected to all 784 pixel neurons in 1st layer and each 784 connections has its weight associated with it.
Thus, total connections from the pixel layer to the 1st hidden layer

$$784 * 16 + 16 = 12{,}560$$

- ➔ Here 784 ∗ 16 are the total associated weights and 16 are the biases.

Similarly, total connections from 1st hidden layer to the 2nd hidden layer

$$16*16 \text{ (neurons) } +16 \text{ (bias) } = 272$$

- ➔ here there are 16 neurons in the first and second hidden layers.

Similarly, total connections from 2nd hidden layer to the output layer

$$16*10 \text{ (neurons) } +10 \text{ (bias) } = 170$$

Thus, total connections in this network

$$12{,}560 + 272 + 170 = 13{,}002$$

13,002 weights and biases that can be changed or adjusted to make the network behave in the different ways. Thus, LEARNING means finding the right weight and bias to give the output with very less error.

NOTATION:

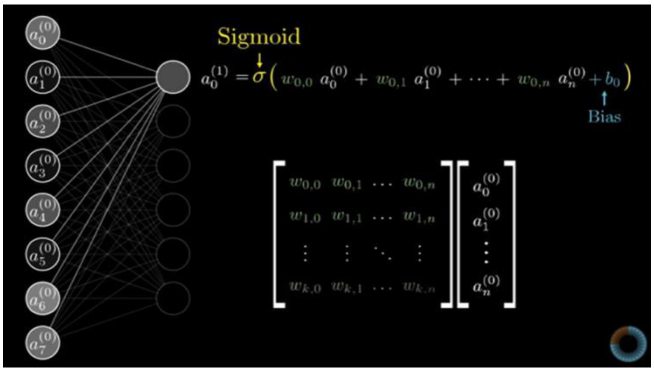

Fig 6. Showing the representation of sigmoid

a_q^p =it is the q th neuron of the p th layer. $w_{r,s}$ – weight of connection of r th neuron of the previous layer to the s th neuron of the next layer.

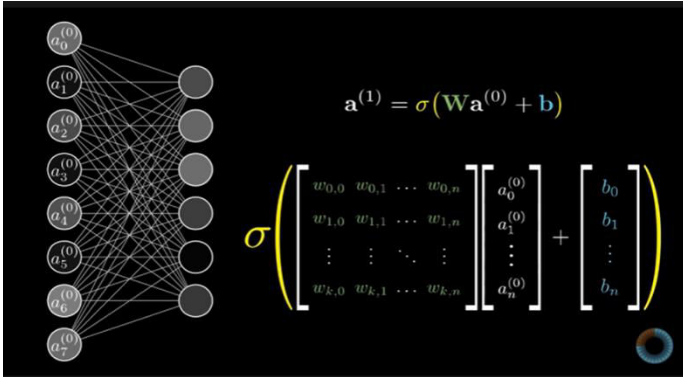

Fig 7. Showing the weighted representation

The a^1, a^0 represents the neurons of the 1st hidden layer, and that of the first layer of the network, W is the weight matrix and b is bias matrix.

Many modern networks do not use sigmoid function any more due to its complexity, thus a new function named ReLU (Rectified Linear Unit) is used. It shows whether the neurons are activated or not if it passes certain threshold value.

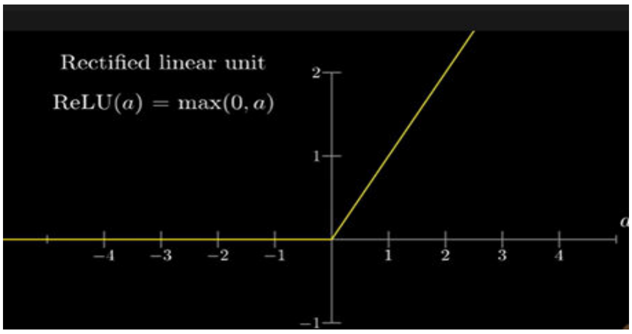

Fig 8. Showing threshold with rectified linear unit

V. FUNCTIONALITY

We consider all the neurons in the previous layer connected to the next neurons in the forward layers and the weights in the weighted sum defining its activation, and the bias define its state of activity / inactivity. If we give any value to the weights, then the network will give us the most unsignificant output. Thus, a cost function is generated, which is a way to tell the bad network that the output is 0 for all the neurons except for 1 for the desired answer containing neuron.

COST FUNCTION=$\sum(obtained\ o/p - desired\ o/p)^{2}$

Obtained O/P- it is the output of the network that is initialized randomly.

Desired O/P-it is the correct output expected by the network. It is 0 for all the neurons, except one corresponding to the right answer.

The summation is small when the network correctly classifies the image, else it would be a higher value. Also, the average of thousands of data is taken into consideration. Thus, we need to change these weights and biases in order to get the correct o/p i.e., to minimize the function. Using multivariable calculus, we can find the negative of the gradient of that function at that point which describes the how the function decreases so quickly

But, here calculus is difficult to be used for complicated functions with 13,002 inputs to be minimized. So, we start with lower most input and move in the direction of falling graph to attain the lowermost point (if slope of the function is positive at that point, then shift to the left else shift the right if negative).

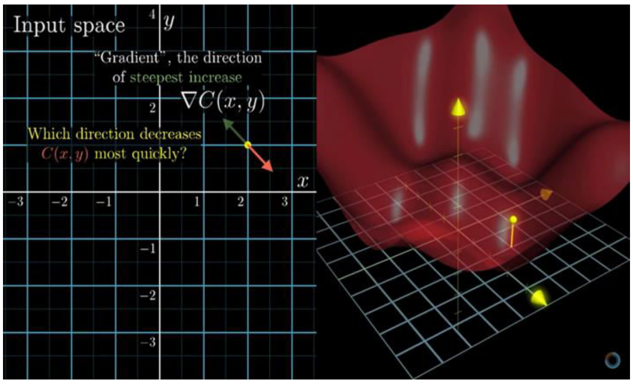

Fig 9. Showing the direction pattern

VI. BACKPROPAGATION ALGORITHM

It was introduced by Rumelhart and McClelland, in 1986 and is used in layered feed- forward Artificial Neural Networks (ANNs). It can be interpreted as the iteration of forward and backward propagation to obtain trained weights and biases [4].This implies that there are many neurons arranged in different layers which are used to find results in forward direction (FORWARD PROPAGATION) and if the errors are present, they are calculated backwardly (BACKWARD PROPAGATION). It is the algorithm that uses supervised learning, meaning, we provide the algorithm the data set and it is expected to compute the result and then the error is calculated. Its basic idea is to train the ANN to such an extent that it learns the training data (error ≈0), by changing the weights, which initially begins with any random value.

The four main steps are

i) Feed forward computation.
ii) Back-propagation to the output layer
iii) Back-propagation to the hidden layer
iv) Weight updates [2].

BACKPROPAGATION- output layer → hidden layer → input layer

The activation of the neuron in the last layer be a_j^L, and that of the neuron in the previous layer be a_j^{L-1}. if the expected value in the last neuron is y, then the cost of this network is-

$$z_j^L = w_{jk}^L a_k^{L-1} + b^L$$

$$a_j^L = \sigma\left(z_j^L\right)$$

$$a_j^L = \frac{1}{1+e^{-z_j^L}}$$

$$C_0 = \frac{1}{2N}\sum_{j=1}^{N}\left(a_j^L - y_j\right)^2$$

→ $z_j^L$is the output of the j^{th} neuron in the L^{th} layer.

→ w_{jk}^L is the weight of the k^{th}neuron of the previous layer to the j^{th} neuron in the L^{th} layer.

→ a_k^{L-1} is the input value of the k^{th}neuron of the previous layer $(L-1)$.

→ b^L is the bias of the Layer L.

→ C_0 is the squared loss function and the number of neurons in the output layer be N.
σ is the sigmoid function which is very close to one for large positive numbers, 0.5 at zero, and very close to zero for large negative numbers. This allows the smooth movement between low and high output of the neuron.

Our main aim is to calculate how sensitive our cost functions is to small change in w_{jk}^{L} $(\frac{\partial C_0}{\partial \mathrm{w}_{\mathrm{jk}}^{\mathrm{L}}})$. So, the small change in $\mathrm{w}_{\mathrm{jk}}^{\mathrm{L}}$ causes a change in z_j^L and a small change in z_j^Lcauses a change in $a_j^L$and finally a small change in a_j^Lcauses a change in C_0.

$$\frac{\partial \mathrm{C}_0}{\partial \mathrm{w}_{\mathrm{jk}}^{\mathrm{L}}} = \frac{\partial \mathrm{C}_0}{\partial \mathrm{a}_{\mathrm{j}}^{\mathrm{L}}} * \frac{\partial \mathrm{a}_{\mathrm{j}}^{\mathrm{L}}}{\partial \mathrm{z}_{\mathrm{j}}^{\mathrm{L}}} * \frac{\partial \mathrm{z}_{\mathrm{j}}^{\mathrm{L}}}{\partial \mathrm{w}_{\mathrm{jk}}^{\mathrm{L}}} \quad [5][6]$$

$$\frac{\partial \mathrm{C}_0}{\partial \mathrm{a}_{\mathrm{j}}^{\mathrm{L}}} = \frac{\partial\left(\frac{1}{2N} * \sum_{j=1}^{N} \left(a_j^L - y_j\right)^2\right)}{\partial y_n}$$

$$= \frac{1}{N} * (a_n^L - y_n)$$

$$\frac{\partial \mathrm{a}_{\mathrm{j}}^{\mathrm{L}}}{\partial \mathrm{z}_{\mathrm{j}}^{\mathrm{L}}} = \frac{e^{\mathrm{z}_{\mathrm{j}}^{\mathrm{L}}}}{\left(1 + e^{-\mathrm{z}_{\mathrm{j}}^{\mathrm{L}}}\right)^2} = \sigma\left(z_j^L\right).\{1 - \sigma\left(z_j^L\right)\}$$

$$\frac{\partial z_j^L}{\partial w_{jk}^L} = \frac{\partial\left(w_{jk}^l a_k^{L-1} + b^L\right)}{\partial w_{jk}^L} = a_k^{L-1}$$

The sensitivity of the bias is almost same as that of weight. We just need to replace the ∂w by ∂b

$$\frac{\partial \mathrm{z}_{\mathrm{j}}^{\mathrm{L}}}{\partial b^L} = \frac{\partial}{\partial b^L}[\, w_{jk}^L a_k^{L-1} + b^L] = 1 \;;$$

The derivative of the cost with respect to one of the activation layer $\mathrm{a}_{\mathrm{k}}^{\mathrm{L-1}}$

$$\frac{\partial \mathrm{C}_0}{\partial \mathrm{a}_{\mathrm{k}}^{\mathrm{L}-1}} = \sum_{\mathrm{j}=0}^{\mathrm{n_L}-1} \frac{\partial \mathrm{z}_{\mathrm{j}}^{\mathrm{L}}}{\partial \mathrm{a}_{\mathrm{k}}^{\mathrm{L}-1}} \frac{\partial \mathrm{a}_{\mathrm{j}}^{\mathrm{L}}}{\partial \mathrm{z}_{\mathrm{j}}^{\mathrm{L}}} \frac{\partial \mathrm{C}_0}{\partial \mathrm{a}_{\mathrm{j}}^{\mathrm{L}}}$$

Sum of all layers as the neuron influences the cost function through multiple paths.

VII. CONCLUSION

"Data Is the New Oil", is forming the base of Artificial Neural Network. The emergence of huge data flow in modern internet era is also increasing the importance of cryptography and allied areas [7-15] specifically in wireless communication scenarios [16-21] and related network security applications [22-26]. There is a very wide scope of AI and ANN's in the future. Neural network is among the central technologies of artificial intelligence. Neural networks have already changed the world, but we've only touched the surface**.** It has been around us for years (e.g., facial recognition on our smart phones) and in many other applications like finance, business analytics, and enterprise training. But they're growing ever more prevalent [3]. Mathematically the major challenge is to take action so as to minimize the cost function, or maximize the utility function over time. In the future, neural networks will improve processing, internet speed, and sensor monitoring, thus making our lives easier and adding more fun. There are four main trends driving the AI industry – responsible AI, small and wide data, operationalization of AI platforms and efficient use of resources.

REFERENCES

[1] Neural Networks and Deep Learning, Springer, A Textbook by Charu C. Aggarwal, IBM T. J. Watson Research Center, International Business Machines Yorktown Heights, NY, USA

[2] Back Propogation Algorithm: An Artificial Neural Network Approach International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 Published by, www.ijert.org ICCCS - 2017 Conference Proceedings

[3] A Review on Back-Propagation Neural Networks in the Application of Remote Sensing Image Classification Alaeldin Suliman and Yun Zhang Department of Geodesy and Geomatics Engineering, University of New Brunswick, Fredericton E3B-5A3, Canada

[4] Comparison of backpropagation and ernn methods in predicting corn production Sigit Susanto Putro , Muhammad Ali Syakur, Eka Mala Sari Rochman, Husni, Musfirotummamlu'ah, Aeri Rachmad Department of Informatics Engineering, Faculty of Engineering, University of Trunojoyo Madura, Bangkalan, Indonesia

[5] Mathematics for machine learning by Marc Peter Deisenroth, A. Aldo Faisal , Cheng Soon Ong.

[6] Implementation of Software for Machine Learning and Deep Learning by Egor Filonov, Vaasan Ammattikorkeakoulu University Of Applied Sciences Information technology.

[7] V. Shukla, A. Chaturvedi, N. Srivastava, Nanotechnology and cryptographic protocols: issues and possible solutions, Nanomaterials and energy, volume 8, issue 1, 2019, 1-6,
DOI: 10.1680/jnaen.18.00006

[8] M.K. Misra, A. Chaturvedi, S.P. Tripathi, V. Shukla, A unique key sharing protocol among three users using non-commutative group for electronic health record system, Journal of discrete mathematical sciences and cryptography, volume 22, issue 8, 2019, 1435–1451,
DOI: 10.1080/09720529.2019.1692450

[9] A. Chaturvedi, N. Srivastava, V. Shukla, A secure wireless communication protocol using Diffie-Hellman key exchange, International journal of computer applications, volume 126, number 5, 2015, 35-38,
DOI: 10.5120/ijca2015906060

[10] V. Shukla, A. Chaturvedi, M.K. Misra, On authentication schemes using polynomials over non commutative rings, Wireless personal communications, volume 118, issue 1, 2021, 1-9,
DOI: https://doi.org/10.1007/s11277-020-08008-4

[11] V. Shukla, A. Mishra, S. Agarwal, A new one time password generation method for financial transactions with randomness analysis, Innovations in electrical and electronic engineering (Part of the lecture notes in electrical engineering book series (LNEE, volume 661)), 2020, 713-720,
DOI: https://doi.org/10.1007/978-981-15-4692-1_54

[12] V. Shukla, A. Mishra, A. Yadav, An authenticated and secure electronic health record system, IEEE international conference on information and communication technology, 2019, 1-5,
DOI: 10.1109/CICT48419.2019.9066168

[13] A. Chaturvedi, V. Shukla, M.K. Misra, Three party key sharing protocol using polynomial rings, 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), 2018, 1-5,
DOI: 10.1109/UPCON.2018.8596905

[14] V. Shukla, A. Chaturvedi, N. Srivastava, Secure wireless communication protocol: to avoid vulnerabilities in shared authentication, Communications on applied electronics, volume 7, number 6, 2017, 4-7,
DOI: 10.5120/cae2017652680

[15] V. Shukla, M.K. Misra, A. Chaturvedi, Journey of cryptocurrency in India in view of financial budget 2022-23, Cornell university arxiv, 2022, 1-6,
DOI: https://doi.org/10.48550/arXiv.2203.12606

[16] V. Shukla, N. Srivastava, A. Chaturvedi, A bit commitment signcryption protocol for wireless transport layer security (wtls), IEEE international conference on electrical, computer and electronics engineering , 2016, 83-86,
DOI: 10.1109/UPCON.2016.7894629

[17] V. Shukla, A. Chaturvedi, N. Srivastava, A new secure authenticated key agreement scheme for wireless (mobile) communication in an EHR system using cryptography, Communications on applied electronics, volume 3, number 3, 2015, 16-21,
DOI: 10.5120/cae2015651903

[18] A. Chaturvedi, N. Srivastava, V. Shukla, S.P. Tripathi, M.K. Misra, A secure zero knowledge authentication protocol for wireless (mobile) ad-hoc networks, International journal of computer applications, volume 128, number 2, 2015, 36-39,
DOI: 10.5120/ijca2015906437

[19] V. Shukla, A. Chaturvedi, N. Srivastava, Authentication aspects of dynamic routing protocols: associated problem & proposed solution, International journal of recent technology and engineering, volume 8, issue 2, 2019, 412-419,
DOI: 10.35940/ijrte.B1503.078219

[20] V. Shukla, A. Kushwaha, S.S. Parihar, S. Srivastava, V. P. Singh, Authenticated wireless information display system using GSM module, Communications on applied electronics, volume 5, number 3, 2016, 7-11,
DOI: 10.5120/cae2016652251

[21] V. Shukla, A. Chaturvedi, N. Srivastava, Double layer cryptographic protocol for mobile ad-hoc networks (MANETs) by commitment scheme, Communications on applied electronics, volume 7, number 9, 2017, 32-36,
DOI: 10.5120/cae2017652716

[22] V. Shukla, A. Chaturvedi, N. Srivastava, A secure stop and wait communication protocol for disturbed networks, Wireless personal communications, volume 110, 861-872, 2019,
DOI: https://doi.org/10.1007/s11277-019-06760-w

[23] V. Shukla, A. Mishra, A new sequential coding method for secure data communication, IEEE international conference on computing, power and communication technologies, 2020, 529-533,
DOI: 10.1109/GUCON48875.2020.9231252

[24] A. Chaturvedi, V. Shukla, M.K. Misra, A random encoding method for secure data communication: an extension of sequential coding, Journal of discrete mathematical sciences and cryptography, volume 24, issue 5, 2021, 1189-1204,
DOI: https://doi.org/10.1080/09720529.2021.1932902

[25] V. Shukla, A. Chaturvedi, N. Srivastava, A new one time password mechanism for client-server applications, Journal of discrete mathematical sciences and cryptography, volume 22, 2019, 1393-1406,
DOI: 10.1080/09720529.2019.1692447

[26] V. Shukla, M.K. Misra, A. Chaturvedi, A new authentication procedure for client-server applications using HMAC, Journal of discrete mathematical sciences and cryptography, volume 24, issue 5, 2021, 1241-1256,
DOI: https://doi.org/10.1080/09720529.2021.1932908