International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552, Volume-8, Issue-3, May 2020
https://doi.org/10.21276/ijircst.2020.8.3.16
www.ijircst.org

A Review on the Concept of Deep Learning

Shalini Bhaskar Bajaj

ABSTRACT- Artificial Neural Networks (ANN) has a number of application areas ranging from economic analysis to image processing and recognition. ANN is used by many online stores in the form of recommendation systems to offer suitable products based on their liking. Not only this, artificial neural networks are used these days for routing and navigation systems as in case of unmanned vehicles, antivirus softwares, etc. In this work based on artificial neural network recurrent and convolutional network is proposed at the level of letters in order to classify and sort textual information with given classes.

KEYWORDS-Artificial neural networks, deep learning, fee forward network

I. INTRODUCTION

Neural networks come under the broad field of Artificial Intelligence. In neural networks parallel processing of information is executed by the links. It has a large number of inter-neural connections that helps in speeding up of the process of computation and thus processing of information. Due to large number of connections in the network, resistance to errors is provided by the network itself. In neural network the connection between the neurons is called synapses. With each synapse weight is associated which can be later adjusted based on inputs and outputs. There are three layers in a neural network model: input , output and hidden (refer Figure 1). Input layer receives the input, output layer displays the output and the hidden layers processes the information. Concept of "Deep learning" means that the model has many hidden layers. As we increase the number of hidden layers, the model moves from shallow to deep neural network and is capable of having significantly more complex behaviour.

Each layer of the neural network can use any function to process the information received from the previous layer and as we move deep into the neural network the function used may change from linear to non linear [7]. The weights are changed in the neural network during the training time in order to get the desired output from the given inputs. The activation functions used in neural networks are like complicated regression functions. There are three parameters in neural network:input data, output data and an activation function. When a network has only one node it behaves like regression. In the process of learning of a neural network, the variable parameters are weights of the synapse the keeps on changing during the training process. Apart from the weights, some parameters of the activation function also change during the training process. Some important properties of the activation function are: Non-linear, continuously differentiable, range, monotonic, approximates identify near the origin. The simplest neutron can have a binary activation function which can return either zero or one as output. In case of regular learning, the best activation function is S-shaped sigmoidal function.Other functions that belong to this category are: hyperbolic, rectified linear, hyperbolic tangent [20]. There are certain advantages and disadvantages associated with sigmoidal function. In sigmoidal, the derivatives can be calculated easily and activations are bounded and do not keep on increasing. Function outputs in sigmoidal are not zero centered and are computationally expensive when compared to other functions. Hyperbolic tangent is zero centered and can act as a good alternative to sigmoidal function. Rectified linear unit is a popular activation function as it returns zero in the negative area of the input parameters and input in the positive area of the input parameter. Not only this rectified linear unit is simple to compute and is faster in learning as the gradient of rectified linear unit can become a constant. Rest of the paper is divided into following section: Section 2 discusses feed forward network; section 3 focusses on the training algorithm; section 4 gives details on overfitting; section 5 explains convolution neural networks; section 6 discusses neural network optimisations and finally concluding remarks are given in the last section.

II. FEED FORWARD NETWORK

The information in feed forward neural networks flows in one direction from input to the output. The information flows in both the directions in recurrent topology [2]. Fully connected feed forward network [15] [16] [18] is the simplest type of neural network. There is no limit on the type of activation function chosen, the number of

Manuscript received May 18, 2020

Shalini Bhaskar Bajaj, Professor, Department of Computer Science and Engineering, Amity University Haryana, Gurugram, India, (e-mail: shalinivimal@gamil.com).

layers, number of neutrons in each layer in case of a feed forward neural network. Thus, the simplest form of the network consists of only one neutron. Problem complexity depends upon the number of layers [19][21]. Problems that divide the space into two zones with output zero and output one cannot be solved with one neutron network but can be easily solved with two layers network. As compared to single layer neural network, multilayer neural network has more representing power in case of non linearity. Two layer neural network always takes the form of a convex polygon, if the region is closed whereas, three layer network forms an arbitrary non convex polygonal area.

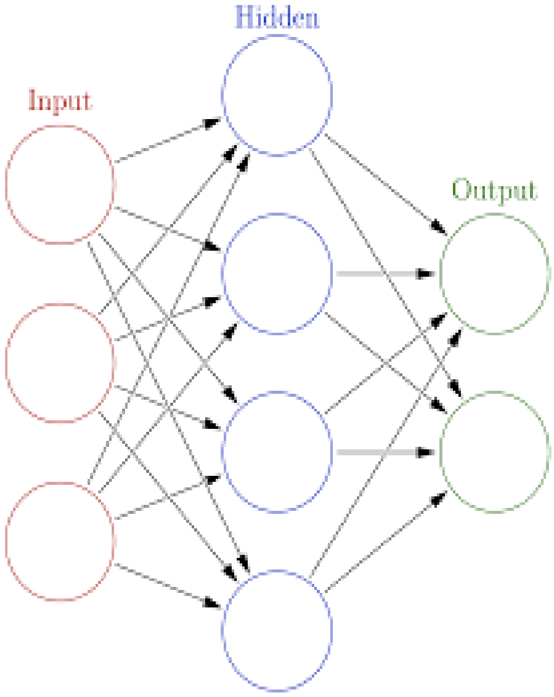

Fig 1: Feed Forward Neural Network [30]

III. TRAINING ALGORITHM

Though we can solve the problem by using neural network does not mean that the solution can be obtained from the initial state of the network. The network topology affects the result more as compared to the learning algorithm. The learning algorithms are broadly divided into two types: supervised and unsupervised [14]. In case of supervised learning algorithm, for each input the neural network has the correct answer in the form of output and accordingly the weights are adjusted based on the activation function used, such that they produce answer close to the correct answers. In case of unsupervised learning, the answers are not known. In case of unsupervised learning training can be done in three ways: batch method, stochastic method and mini-batch method. In batch method, weight change only once. It basically summarises the weight increment on the current epoch. In stochastic method, if the increment of the weight is found, immediately it is updated. In mini-batch method, advantages of both the batch method and the stochastic method are combined.

A. Supervised Learning

This learning method performs the following steps: determine the type of the training examples, determine the input feature representation of the learned function, determine the structure of the learned function and corresponding learning algorithm, complete the design and run the learning algorithm on the gathered training dataset and finally to evaluate the accuracy of the learned function.

a. Loss Functions

Some of the loss functions are given in [20] [22]. The objective of the loss function is to find average deviation in the obtained value and the desired value. Different neural networks uses different loss functions. In case of classification using SVM, the objective of the loss function is to find the margin between the correct structure and highest scoring incorrect structure.

b. Training Algorithm (Improvements)

Implementing the learning rate decay is very crucial. It can be done by adopting any of the mentioned ways [20]: Step decay, Exponential decay and 1/t decay. In step decay, the rate of learning is reduced by some predetermined factor after every few epochs. In case of exponential decay, some exponential formula such as $a=a_0e^{-kt}$, can be used for the decay where t is the epoch number and a_0 and k are hyper-parameters. In case of 1/t decay, mathematical formula such as $a = a_0/(1+kt)$ can be used where t is the iteration number and a_0 and k are hyper-parameters.

c. Hyper-parameters

Selection of the hyper-parameter values is done by trial and error method. Among these values are: mini-batch size, learning algorithm, regularisation, activation function, the size of the window and fold pooling in case of a train convolution neural network, number of layers and neutrons in each layer, parameters that are part of learning algorithm such as learning rate and moment,the ensemble size and ways of combining as in case of ensemble networks. Convergence of neural network is directly affected by the selection of hyper-parameters. There is one simple rule that is followed: more the number of neutrons and hidden layers more is the learning but at the same time the it increases the learning time exponentially when training the model. Though there is no rule on selection of the number of hidden layers and neutrons in each layer but limitations are there which help in deciding it. In case, the function is defined on a finite set of points or the function is defined and continuous on a compact area then three layer model is used for all other functions four layer neural network is used. In theory the number of hidden layers can be between two to four but we can solve the real world problems by increasing the number of layers. Selection of the number of neutrons in each layer is also very important. If the number of neurons are very less then the model will not be able to learn and if the number of neurons are more than the network will take more time to learn due to training on unreal values. It may also lead to overfitting problem. The general approach is to increase both the number of layers and the number of neutrons in each layer until the network becomes overfitting and then deal with the overfitting problem.

B. Unsupervised Learning

When the output is not known the learning is called as unsupervised learning and is used in clustering techniques. Kohonen neural network is used for unsupervised learning, the data loss is minimised in kohonen neural network by reducing the dimensionality of data. In kohonen neural network model the number of neurons are equal to the number of clusters and input variables are normalised. The structure of the model has a single layer of neutrons without biases. The number of features used for characterisation of the object in the model equals the number of input variables. Unsupervised networks have a fixed structure whereas self organisation networks does not have fixed structure. In self organising kohonen algorithm for neural network, the distance used is euclidean distance which is set as the critical distance which corresponds to the maximum allowable euclidian distance. When the first sample from the training set is fed to the network it creates the first neutron weight. When the next test sample from the training set is applied to the network, the euclidian distance is calculated between the new input and each cluster centre and thus the lowest distance is recorded as *R-minimum*. If the value of *R-minimum* is smaller than the critical distance value then the correction weight coefficients corresponds to the neuron winner. This procedure is repeated and if till the last epoch any cluster is not involved in the network structure then finally it is excluded from the network.

IV. OVER-FITTING

Overfitting is a serious problem in training of the neural network. It could be because the neural network is exposed to a big set of training examples or training dataset is very complex or insufficient training examples are there and thus it looses its ability to generalise. While training the neural network with the training set, the main objective is to give neural network the ability to generalise the result to the new observations. After training the network on a number of training samples and testing on a number of samples, the prediction error id reduced on both the test and the train sets. After some time the error on the test dataset begins to increase and on the train dataset the error begins to decrease. Thus, we can see that the accuracy on the test dataset falls. The test and train dataset should not be overlapping. To handle noisy data both the linear and the polynomial functions work properly. We can observe that the polynomial function fits perfectly on the noisy data whereas the linear function generalise better for the model [23]. In order to solve the problem of overfitting in neural networks, the methods used are dropout, adding noise to the data [12], regularisation, batch normalisation and thinning of the network. In case of regularisation [3] [9][24] of the model, fine is imposed on the objective function. In dropout regularisation method [4][25], neurons are dropped from the existing model and at every step a new network architecture is generated. In batch normalisation [17], input data is changed in such a way to obtain zero expectations and a unit variance. Before entering each layer, the process of normalisation is performed.

V. CONVOLUTION NEURAL NETWORK

Convolution neural network is a special artificial neural network (ANN) architecture [8] [10] [20] [26] [27] [28] [29]. It was proposed for effective image recognition [1] by Yann LeCun and is a type of deep learning algorithm. Convolution neural networks uses a filter which is a mathematical convolution operator and finds its application in the extraction of features from video, audio and text data [5][6][11][13]. There are three layers in convolution neural network: C-layers (convolution layers), S-layers (sub-sampling layers) and F-layers (fully connected feed forward) at the output. The three main paradigms of this architecture are: local invariance as input to the neuron feed only a part of the image and not the complete image; shared weight means that a very small set of weights is shared by the neuron with large number of links shared. Sub-sampling in the S-layer reduces the spatial dimensions of the image. Convolution neural networks are very fast and are implemented on General Processing Units (GPUs).

VI. NEURAL NETWORK (NN) OPTIMISATIONS

Ensembles of the neural network model helps improve the accuracy of the model. Advantages of ensembles are:statistical, representatives and computing. Ensembles solve either the problem of overfitting which is solved by begging or to not fit enough which is solved by boosting. Begging also called as bootstrap aggregating is the result of aggregating the results of different loadings whereas boosting or busting is a series of algorithms where the next algorithm in the row will try to overcome the shortcomings of the previous algorithm. In case of different classifiers, begging is useful. It is also useful when small changes in the input leads to significant changes in the classification. Busting on the other hand is a greedy algorithm which is used for constructing composition of algorithms.

VII. CONCLUSIONS

The study based on understanding of the neural networks. Details are provided on Feed forward network, training algorithms and their types. Difference between the supervised and unsupervised learning has been presented in this paper. Discussion is done on overfitting problem and its solution. Apart from this, convolution neural networks and neural network optimisations have also been discussed. Overall the paper presents a good overview on the above mentioned topics and gives a precise knowledge on the neural networks.

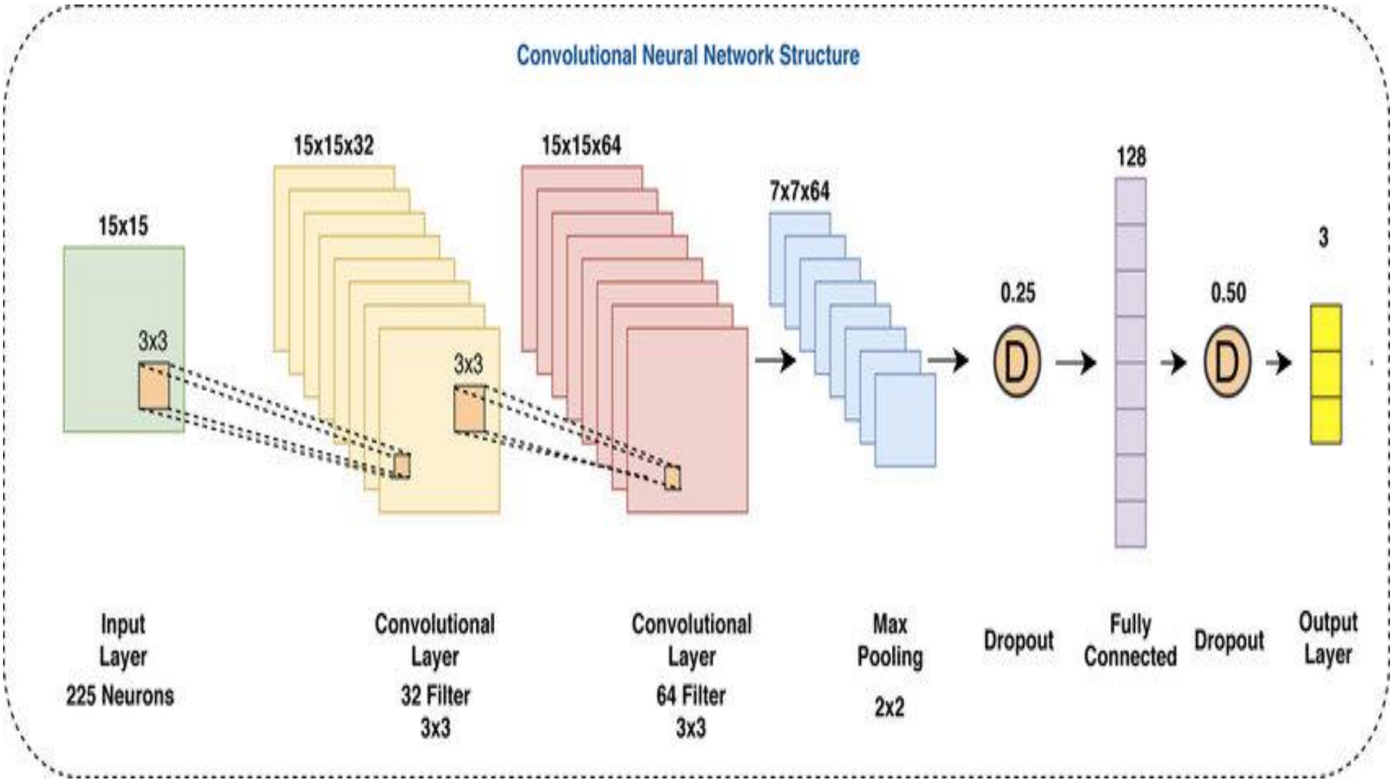

Fig 2: Convolution Neural Network [31]

REFERENCES

[1] Kaiming He ; Xiangyu Zhang ; Shaoqing Ren ; Jian Sun, Deep Residual Learning for Image Recognition, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, 2016.

[2] Andrej Krenker, Janez Bešter and Andrej Kos, Introduction to the Artificial Neural Networks , Artificial Neural Networks - Methodological Advances and Biomedical Applications

[3] Zhongqiang Zhang, Stochastic Processes, Springer, 2016, MA 529

[4] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutkever, R. Salakhutdinov, Droupout: A simple way to prevent neural networks from overfitting, Journal of Machine Learning Research, vol. 15, (2014), pp. 1929-1958

[5] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean, Efficient Estimation of Word Representations in Vector Space, Computation and Language, Cornell University, arXiv:1301.3781

[6] Yoon Kim, Convolutional Neural Networks for Sentence Classification, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar

[7] Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, Yonghui Wu, Exploring the Limits of Language Modeling, Computation and Language, Cornell University, arXiv:1602.02410

[8] Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R. Steunebrink, Jürgen Schmidhuber, LSTM: A Search Space Odyssey, Neural and Evolutionary Computing, Cornell University, arXiv:1503.04069

[9] Siwei Lai, Liheng Xu, Kang Liu, Jun Zhao, Recurrent Convolutional Neural Networks for Text Classification, Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, pp. 2267-2273

[10] Jiaming Xu, Peng Wang, Guanhua Tian, Bo Xu, Jun Zhao, Fangyuan Wang, Hongwei Hao, Short Text Clustering via Convolutional Neural Networks, Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, June 2015, pp. 62-69

[11] Yin Zhang, Rong Jin, Zhi-Hua Zhou, Understanding bag-of-words model: a statistical framework, International Journal of Machine Learning and Cybernetics volume 1, pages43–52(2010)

[12] Yann LeCun Leon Bottou Yoshua Bengio and Patrick Haner, GradientBased Learning Applied to Document Recognition, Proceedings of the IEEE, November 1998

[13] X. Chen ; X. Liu ; M. J. F. Gales ; P. C. Woodland, Recurrent neural network language model training with noise contrastive estimation for speech recognition, 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, QLD, Australia

[14] Matthew D. Zeiler, ADADELTA: An Adaptive Learning Rate Method, Machine Learning (2012), Cornell University, arXiv:1212.5701

[15] Dumitru Erhan, Pierre-Antoine Manzagol, Yoshua Bengio, Samy Bengio and Pascal Vincent, The Difficulty of Training Deep Architectures and the Ef-

fect of Unsupervised Pre-Training, Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS) 2009, Clearwater Beach, Florida, USA. Volume 5 of JMLR: W&CP 5

[16] Pierre Baldi, Kurt Hornik, Neural Networks and Principal Component Analysis: Learning from Examples Without Local Minima, Neural Networks, Vol. 2, pp. 53-58, 1989

[17] Sergey Ioffe, Christian Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, Proceedings of the 32 nd International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37

[18] Yadav, Neha, Yadav, Anupam, Kumar, Manoj, An Introduction to Neural Network Methods for Differential Equations, SpringerBriefs in Computational Intelligence, Springer, 2015

[19] I. V. Zaentsev, Neural Network: Basic Models

[20] https://cs231n.github.io/

[21] http://www.aiportal.ru/articles/neural-networks/decision- xor.html

[22] https://malaikannan.wordpress.com/2016/09/13/cross-entropy/

[23] https://en.wikipedia.org/wiki/Overfitting

[24] https://habr.com/en/post/175819/

[25] http://www.nanonewsnet.ru/articles/2016/kak-obuchae tsya-ii

[26] https://geektimes.ru/post/74326/

[27] http://deeplearning.net/tutorial/lenet.html

[28] http://www.360doc.com/content/16/0303/19/2459_53 9162206.shtml

[29] http://colah.github.io/posts/2014-07-Understanding-C onvolutions/

[30] https://www.google.com/url?sa=i&url=https%3A%2F %2 Fautomaticaddison.com%2Fartificial-feedforward-neural-network-with-backpropagation-from-scratch%2 F&psig=AOvVaw2jvJzDQ-fHIedy5qd-IgRz&us t=1 590387947314000& source=images &cd=vfe& ved=0 CAQQtaYDahcKEwjYhorr9MvpAhUAAAAAHQAA A A AQEA

[31] https://www.google.com/url?sa=i&url=https%3A%2F %2Fwww.researchgate.net%2Ffigure%2F15x15-Pixel -Labelled-Sample-Images-After-Image-Creation-Phase _fig4_324802031&psig=AOvVaw3_PZaA1ZZ7sII1cOssimWn&ust=1590388321832000&source=images&cd=vfe&ved=0CAMQjB1qFwoTCMDSldDwy-kCFQ AAAAAdAAAAABAR

ABOUT THE AUTHOR

Shalini Bhaskar Bajaj is working as Professor and HoD(Computer Science and Engineering). She completed her Ph.D. from IIT Delhi and is working in the field of Data Mining and Analytics. She is having 20 years of experience as an academician. She has more than 50 publications in reputed journals/conferences.