

Modern Directions in Neural Networks with Optimization Techniques

Nishu Sethi¹, Neha Bhateja²

Assistant Professor, Department of Computer Engineering, Amity University Haryana, India¹

Assistant Professor, Department of Computer Engineering, Amity University Haryana, India²

ABSTRACT: In deep learning, artificial neural network(ANN) has shown vital growth in recent years. Artificial neural network takes inspiration from the brain to process the information. The core element of this paradigm is the structure of the information processing system. It is a network of interconnected processing elements commonly known as neurons work together to solve specific problems. This paper reviews the working of an artificial neural network, optimization techniques used for enhancing the performance of the neural networks, training methods, and real-world applications. Artificial neural network has several applications such as image processing, character recognition and forecasting. This paper briefly discusses optimization methods like ADAM, gradient descent and its variants.

KEYWORDS: Artificial Neural Network, Supervised Learning, Unsupervised Learning, CNN, RNN, Optimization Methods, Gradient Descent, ADAM, Framework, Image Classification

I. INTRODUCTION

Human brain is always a mystery to us no matter how much we know about it. Our brain efficiently analyzes and drive results from the given information. Brain can make sense out of surrounding environment and respond in real time. Interestingly, it needs very little space and consumes very less power. Brain's unique behavior and ability to deal with ambiguity made researchers to think differently and come up with a model to make computers work like the human brain in some aspect.

Neural networks have a distinct method to tackle a problem as compared to traditional computing systems. Traditional computing systems involves set of instructions to be performed to solve a problem. If a specific step is missing computer won't be able to perform tasks. Neural networks don't operate in such fashion. They learn from examples and figure out the way to solve a given problem itself and sometimes their operation can be unpredictable.

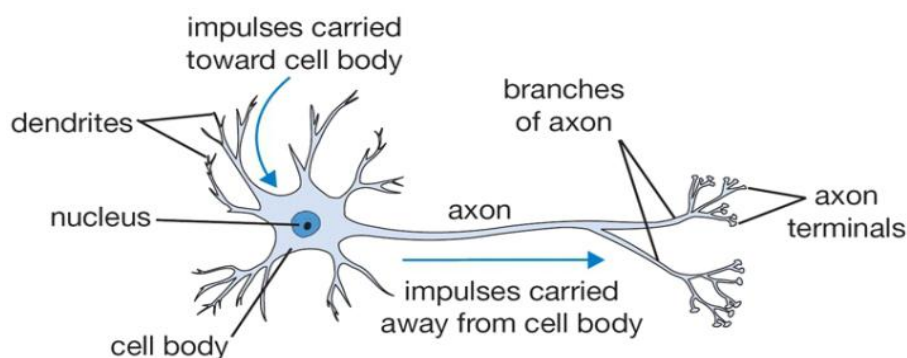


Fig 1: A model of biological neuron [7]

This paper is organized into 6 sections as follow: section 1 covers up the historical background of artificial neural network, section 2 explains important neural network models, section 3 deals with the learning methods to train a neural network, section 4 discusses the optimization methods and

the section 5 is presented with applications of artificial neural network. Last section summarizes the complete paper in tabular data.

II. HISTORY OF ANN

It was during the year 1943 when Warren McCulloch [3] a neurophysiologist, and a mathematician named Walter Pitts wrote first paper on how neurons might function by modelling a neural network with electric circuit. As an outcome they proposed the McCulloch-Pitts (MP) model. From the model it was noted that the neurons are in binary state and performed logic operations by altering the activation threshold value. It partially replicates the functioning of biological neurons and can find the way through ambiguity.

Later, Rosenblatt came up with a supervised learning model and it was called perceptron. Theoretically any function can be calculated by designing a multi-layered network of perceptron. Then the practical approach using back-propagation algorithm appeared. Multi-layered network and the back-propagation algorithm combined could be used practically so solve problems.

In next few years more, research get conducted and Widrow proposed a single layer artificial neural network based on MP model. Efficiency and application point of view was main concern among the researchers. Then Hinton coined a new concept “Deep Learning” in the journal Science in 2006. He utilized “pre-training” method to get sophisticated initial weights. Since then field of Artificial Neural Network attracted more research and rapid development is being carried out.

III. MODELS OF ANN

3.1 A Simple Neural Network with Back-Propagation

This is the classical example of neural network model, most of the neural networks are based on MP model. The neuron model is expressed mathematically as the following:

$$y=f(x_iw_i-\theta) \tag{1}$$

Where x_i is the input data and w_i is the corresponding weight, threshold value is represented by

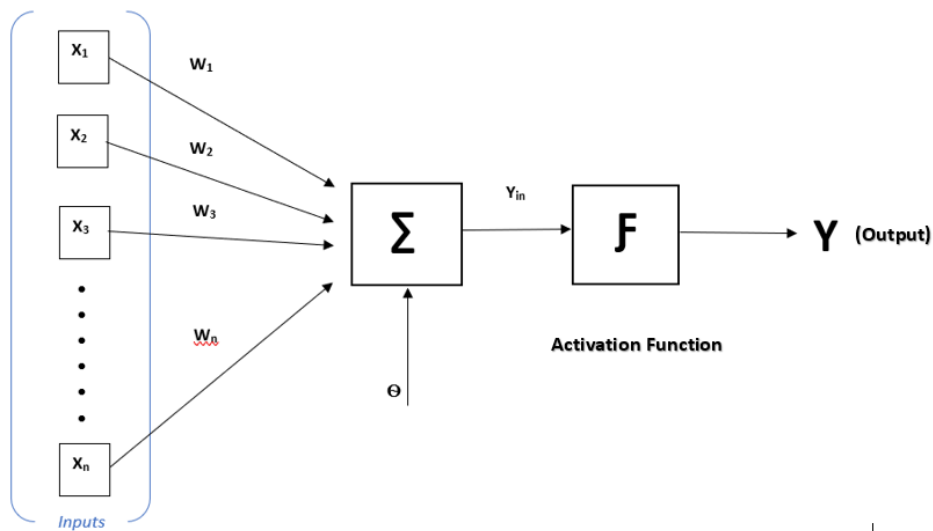


Fig 2: McCulloch and Pitts model of artificial neuron.

Multi-layered neural network with the back-propagation compatibility algorithm lead to the construction of a back propagation artificial neural network (BP-ANN). There are three components: inputs in the form of input layer, to

process the inputs there are several hidden layers and outputs in the final layer [4]. These layers are constructed from lots of neurons which act independently.

Three steps involved in learning process:

Step - 1 Forward propagation - To generate a set of outputs based on inputs

Step -2 Error Back-Propagation - Computes the error gap present between the ideal signal and output signal. Error moves backwards between layers.

Step -3 Updating the weight - Weight is adjusted for matrix of each layer depending upon the error.

Above mentioned three steps are imitated until output error levels up with the given condition otherwise it halts when iterations meet a set number. The above network can automatically adjust the weight. Convergence rate get moderated for bigger network and time for training is increased.

3.2 Convolution and Deep Neural Network

In DNN the hidden layers are branched into two types of layers: Convolution and Pooling. Normalization layers can be used in DNN if required. A classifier layer is present at the end. Due to many layers DNN can help in achieving complex functions.

The convolution neural network behaves no different than deep neural network except one thing that the CNN uses shared kernels and DNN uses separate kernels. CNN can extract parameters of examined and translated data, rotation invariance, scaling, hence degree of effectiveness is higher for image recognition and analysis of speech [4].

For extraction of high-level features, we use the convolution layer. The feature map is the result of outputs from the neuron. For the same we have expression as follow:

$$x_{lj} = f(\sum_{i \in M} x_{l-1i} * k_{li,j} + b_{lj}) \dots \dots \dots (2)$$

Where x_{lj} represents the j th feature map for i th layer, $k_{li,j}$ kernel which links the feature map of previous and the current layer, b_{lj} is the j th define the bias of the l th layer, and nonlinear activation function is denoted by f .

Sampling of non-overlapping areas for the equal size is done by pooling layer and it also consider maximum or average for the output. Dimensionality for the given data is reduced hence the partial transformation and the network are paralleled. Mathematical representation is as follow:

$$x_{lj} = f(\beta_{lj} \text{down}(x_{l-1i}) + b_{lj}) \dots \dots \dots (3)$$

here down acts as the sampling function, β_{lj} denotes coefficient of sampling, b_{lj} tells about j th bias present in l th layer, and f stands for nonlinear activation function.

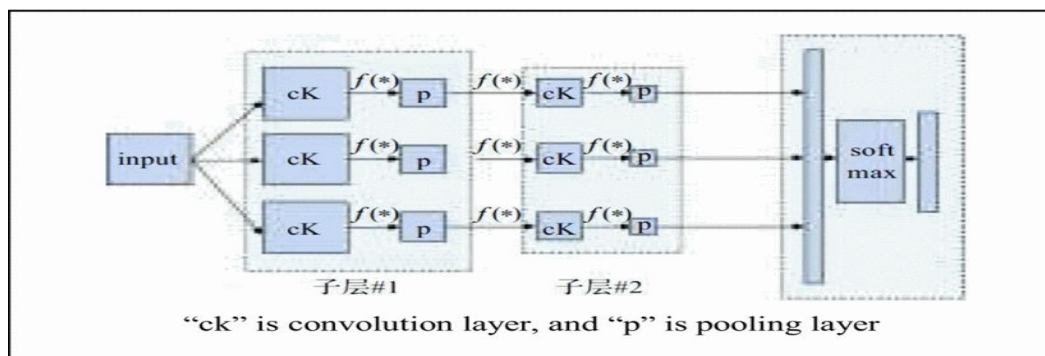


Fig3 : Layers in DNN and CNN [4]

IV. TRAINING METHODS FOR ARTIFICIAL NEURAL NETWORK

After construction of a neural network for a particular application our goal is to train the network. This process starts with the initialization of random weights. Then the real training or learning process begins. Generally, we have two techniques to train a neural network - supervised and unsupervised learning. Supervised learning method a supervisor (manually grading or providing desired output) helps in the learning process. Whereas, in unsupervised learning there is no help provided and neural network make sense out of inputs itself.

4.1 Supervised Learning

As the name suggests we require external supervision to train the neural network. In such type of learning neural network is provided with the set of inputs and then the resulting inputs are compared with the desired outputs. This way we calculate the error generated by the network and then the error is propagated back the network. By error backpropagation the network adjusts the weights which control the network. This process is repeated until we result close to our desired results [1].

4.2 Unsupervised Learning

In unsupervised learning we are not provided with any kind of supervision. A neural network is presented with inputs and there are n desired outputs available. In this learning technique neural network decides which features to use and how to group the data. This is often termed as adaption or self-organization. At the present scenario unsupervised learning is not much convenient for the use. Unsupervised learning is suitable to train machines for new situations and the machine learns itself from its environment. In real life we always don't have proper data regarding the situations. In future unsupervised learning may make robots able to adapt according to the surroundings and become capable of taking decision their own [1].

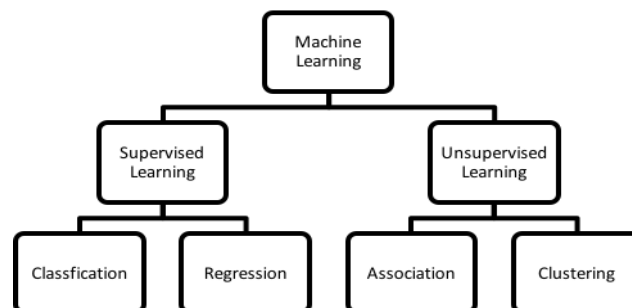


Fig 4: Different types of learning techniques

V. OPTIMIZATION METHODS USED IN NEURAL NETWORKS

Optimization algorithms helps to speed up the learning process and to ensure that we are heading in right direction. Optimization algorithms simply reduce the value of error function. Error function is based on the neural network's internal parameters which are used in computing the output from the predictors used in the model. Internal parameters have key role in effectively and more efficiently training a neural network model and generate accurate results.

5.1 Types of optimization algorithms:

1. First Order Optimization Algorithms - These algorithms simply use the Gradient values to minimize or maximize an error function. Gradient Descent is most popular among first order optimization algorithms. Using first order derivative we identify whether the function is increasing or decreasing at a selected point [8].

2. Second Order Optimization Algorithms - It uses second order derivatives (also referred as Hessian) to minimize or maximize the loss function. As it is costly in terms of computation it is not used much [8].

In this paper we will mainly discuss only about the following optimization algorithms:

- Gradient descent and its type
- ADAM

5.2 Gradient Descent

Gradient Descent algorithm utilize the concept of prediction on the training data and error function. To find out the minimum value of error function it updates the values of bias and weights of the neural network. It is efficient method to speed up the training process of a model. Further the gradient descent is divided into three parts depending upon the number of training patterns used to calculate the error function and update the neural network [5].

Types of Gradient Descent:

- Stochastic Gradient Descent
- Batch Gradient Descent[5]
- Mini-Batch Gradient Descent [5]

5.2.1 Stochastic Gradient Descent

Stochastic Gradient Descent [5] algorithm first evaluate all the data points of the training dataset and then computes the error for each data point. After the error computation it updates the model for each data point available in dataset. Stochastic Gradient Descent does not change the learning rate i.e. alpha learning.

5.2.2 Batch Gradient Descent

Works like the first optimization algorithm but instead of using single learning rate it uses the entire batch to evaluate the dataset and calculates the error function. During implementation of batch gradient, the entire dataset should be present in the memory. This algorithm performs poorly in terms of time complexity for larger datasets.

5.2.3 Mini-Batch Gradient Descent

Mini-Batch Gradient Descent [5] combines functionality and performance of the earlier two optimization algorithms. As the name suggests it breaks a large dataset into small sets called mini-batch training set. The error function is calculated for each mini-batch and updates are made according to the results. It combines the robustness of stochastic gradient descent with the efficiency of batch gradient descent.

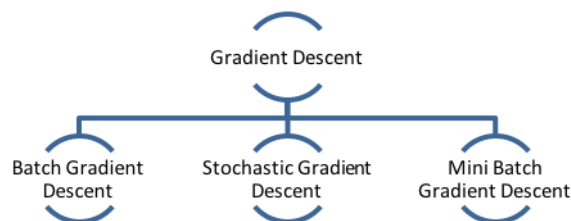


Fig 5: Types of Gradient Descent

5.3 ADAM

ADAM is acronym for Adaptive Moment Estimation. It is enhanced version of Stochastic Gradient Descent. It utilizes the adaptive learning rate to repeatedly update the weights in the model. It combines AdaGrad and RMS prop [6]. For optimization process. Here AdaGrad provides great performance with sparse gradients and RMS prop works fine in non-stationary settings.



VI. APPLICATIONS OF NEURAL NETWORK

S.No.	Creators	Project name	Description
1.	Satoshi Iizuka, Edgar Simo-Serra Hiroshi Ishikawa	Let there be color!	A framework build on the CNN helps you to automatically colorize black & white images. They even experimented it with hundred years old grayscale pictures.
2.	Google Brain Researchers	Pixel Recursive Super Resolution	A deep neural network trained to enhance the resolution of photos significantly.
3.	Andrej Karpathy & Li Fei-Fei	NeuraTalk 2	These researchers created a deep neural network which identify various interesting areas in a picture and write a sentence to describe it.
4.	Isola	Pix2Pix	Performs multiple tasks: creating real street scenes from colored blobs, map from a real aerial photo, fill out the colors between edges and turn day scenes into night.
5.	Google	Google Sunroof	Takes aerial photos from Google Earth to make a 3-D model of your roof. Uses weather conditions and sun's trajectory to predict how much energy can be generated by installing solar panels on the roof.
6.	Oxford Researchers	Read text in the wild	It helps to read text from images and videos to extend Google so we can search for text in BBC News videos.
7.	Google Researchers	Google Translate	Google translate helps you to translate text written on images into various languages.
8.	Yong Zhao	DeepGlint	Real time analysis of various people poses. This model recognizes and describe the photos.
9.	Deep Mind researchers	WaveNet	WaveNet learn to mimic various human voices and improve with time. It is pretty difficult to differentiate between the auto generated voice and human voice.
10.	Pasal van Koten	Neural Complete	Neural Complete is based on LSTM neural network, helps in writing neural network code. Using this model, you can generate code for neural networks based on python.

VII. CONCLUSION

Serial Number	Section Name	Description	References
1	Introduction	A brief overview of correlation between human brain and artificial neural network. How ANN tries to replicate the behavior of biological neurons.	-
2	History of ANN	History behind the development of neural network and how things have changed so far.	3
3	Models of ANN	Describes a few important models in artificial neural networks and how they work.	4
4	Training	The methods used to train the neural networks are	1



	Methods	discussed in this section. Mainly supervised and unsupervised learning techniques are preferred.	
5	Optimization Methods	Covers up the details of some important optimization algorithms used in the field of artificial neural network.	5, 7
6	Applications of ANN	This section mainly discuss recent developments and live applications of ANN. Various projects in image classification and computer vision are based on neural networks.	-

REFERENCES

- [1] L. Deng and D. Yu, “Deep learning: methods and applications,” Foundations and Trends in Signal Processing, vol. 7, no. 3-4, pp. 197–387, 2013
- [2] Adrian Carrio, Carlos Sampedro, Alejandro Rodriguez Ramos and Pascual Campoy “A Review of Deep Learning Methods and Applications for unmanned Aerial Vehicles ” Hindawi Journal of Sensors 2017.
- [3] Xuedam Du, Yinghao Cai, Wang, and Leijie Zhang “ Overview of Deep Learning” 31st Youth Academic Annual Conference of Chinese Association of Automation Wuhan China November 11-13-2016.
- [4] Tao-ran Cheng, Peng-cheng Wen, Yang Li, ” Research Status of Artificial Neural Network and Its Application Assumption in Aviation” International Conference on Computational Intelligence and Security, 2016.
- [5] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient based learning applied to document recognition,” Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.
- [6] www.deeplearningbook.org/contents/optimization.html
- [7] <http://www.wpclipart.com/medical/anatomy/cells/neuron/neuron.png.html>
- [8] <https://towardsdatascience.com/types-of-optimization-algorithms-used-in-neural-networks-and-ways-to-optimize-gradient-95ae5d39529f>