Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.017

IJCSMC, Vol. 7, Issue. 11, November 2018, pg.74 – 76

STUDY OF NEURAL NETWORK WHAT, HOW AND WHY?

Miss. Bharati Ambali[1]
[1]Bharatesh College of Computer Applications, Belagavi, India
[1] bharatia@rediffmail.com

Abstract— Neural Network is the subfield of Machine Learning. It is one of the most powerful and widely used algorithm. So we can explore what is Neural Network, how it works and why?

Keywords: Perceptron, Deep Learning, bias, cost function, threshold, feedforwardNN

I. INTRODUCTION

Neural Network is also called artificial neural networks, as the name suggests that, based on what science knows about the human brain's structure and function. The term 'neural network' is a reference to neurobiology, but although some of the central concepts in deep learning were developed in part by drawing inspiration from our understanding of the brain, deep-learning models are not models of the brain. The modern deep learning models doesn't have evidence to implement that there is anything learning mechanism of brain. The 1st successful neural application is In 1989 from Bell Labs, Yann LeCun combine ideas of convolutional neural networks and back propagation, applied the problem of classifying handwritten digits. In 1990s LeNet dubbed the resulting network, and was used by United States Postal Service to automate the reading ZIP codes on mail envelopes. Neural network makes solving problem much easier, the primary reason is that learning took off quickly and completely automates most crucial step in workflow of machine learning as well as in feature engineering

Anatomy of a neural network: The following are neural network objects:

- Layers, are network (or model)
- The input data
- The loss function, which defines the feedback signal used for learning
- The optimizer, which determines how learning proceeds

Data structure, layer is a fundamental of neural networks. A layer contain data-processing module that takes as one or more tensors as input as well as one or more tensors as output. Some layer's are stateless. The layers weights, one or several tensors learned with stochastic gradient descent.

II. HOW DOES A NEURAL NETWORK WORK?

The neural network can be defined briefly as system that consist of several number of simple but highly interconnected elements or nodes. Nodes are called 'neurons'. The neural network structure introduced by input layer that has one neuron for each component present in input data, communicated to one or more *hidden layers*. Commonly the input is received by weights and biases. Then neuron calculate a weighted sum by **activation function** (most common one is sigmoid, *σ)*. It decides whether it should be 'fired' or activated. Afterwards, the process called '*forward pass*' transmits the information downstream to connect other neurons. At the end process, the last hidden layer is linked to the output layer for each possible desired output of neuron.

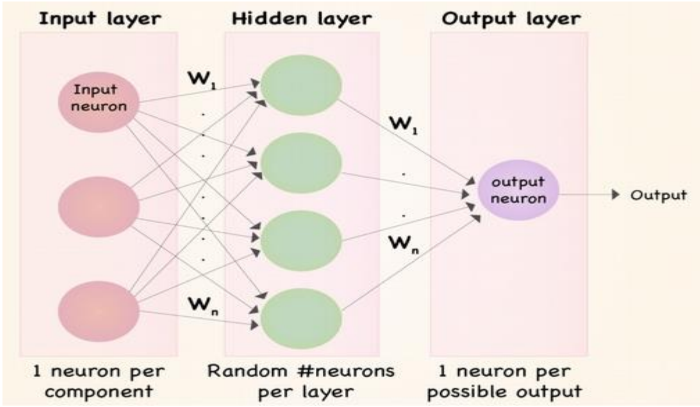

Fig 1: Neural Network Structure

The first type of neuron is perceptron. A perceptron uses a function to learn a binary classifier by mapping a vector of binary variables to a single binary output and it can also be used in supervised learning. In this context, the perceptron follows these steps:

1. Multiply all the inputs by their weights *w*, real numbers that express how important the corresponding inputs are to the output,
1. Add them together referred as *weighted sum:* $\sum wj\ xj$,
2. Apply the *activation function*, in other words, determine whether the weighted sum is greater than a *threshold value*, where -threshold is equivalent to *bias*, and assign 1 or less and assign 0 as an output.

In this algorithm, strongest point is that user can vary the weights, bias to obtain distinct models of decision-making. It allows the user to assign more weight to those inputs so that if they are positive, to desire output. Positive value gives output 1, as negative gives output 0.Also possible, to create more complex networks including more layers of perceptron's where every layer takes the output of previous and weights it and make a more and more complex decisions. Small changes in weights or bias in one perceptron, severely change output from 0 to 1 or vice versa is disadvantage of Perceptron. By introducing small modifications in the weights or bias be able to gradually change the behaviour of network. *Sigmoid neurons* is a Modern type of neuron. Main difference between perceptron and sigmoid neuron is the input and the output can be any continuous value between 0 and 1.

Feedforward neural network is to calculate the predicted output; in other words, we need to build the different layers involved in the network:

- layer0 : input layer; our training set read as a matrix (We can called it X)
- layer1: by apply the activation function a' = σ(w.x+b), in our case, performing the dot multiplication between input layer0 and the synapsis matrix syn0
- layer2: output layer obtained by the dot multiplication between layer1 and its synapsis syn1

III.WHY NEURAL NETWORKS?

The ability to learn and improve every time in predicting an output is the main strength of Machine learning algorithms. What does it mean that they can learn? In the context of neural networks, neurons implies that weights and biases define the connection between neurons more precise; this is, eventually, the weights and biases are selected such as the output from the network approximates the real value y(x) for all the training inputs. So how quantify our prediction for this, we need to calculate an error or in other words, define a **cost function**, is that difference between the expected and the predicted output. Most commonly used function in neural networks is quadratic cost function, it's also called mean squared error. It's defined by the formula:

$$\text{Cost function: } C(w, b) = \frac{1}{2n} \sum_x \|y(x) - a\|^2$$

w, b : referred to all the weights and biases in the network, respectively.
n : total number of training inputs.
a : the outputs when x is the input. ∑ is the sum of over all training inputs.

The main goal of algorithm is to minimize the cost function by finding a set of weights and biases to make it as small as possible. And to achieve this goal is a Gradient Descent algorithm. The computation of gradient of the cost function by *Backpropagation*. The goal of this is to compute the partial derivatives of the cost function with respect to any weight *w* and any bias *b; th*is means calculating the error. back propagating is to update the weight and biases from final layer start from vectors. The reason to go back is that the cost is a function of the output of our network. There are several backpropagation algorithm formula to calculate errors that we need to compute.
1) Output error (δL) related to the element wise ($\odot$) product of the gradient (∇C) by the derivative of activation function ($\sigma'(z)$), 2) error of one layer (□l) in terms of the error in the next layer related to the transpose matrix of the weights (Wl+1) multiplied by the error of the next layer (□l+1) and the element wise multiplication of the derivative of activation function, 3) rate of cost change with respect bias in network : it means that partial derivative of C with respect to any bias ($\partial C/\partial bj$) is equal to the error □l, 4) rate of cost change with respect to weight in the network means that partial derivative of C with respect to any weight ($\partial C/\partial wj$) is equal to the error (□l) multiplied by activation of the neuron input.

All terms of algorithm to implement:

- **INPUT**: It's a input set of training data set examples and set the activation *a* that correspond for the input layer.
- **FEEDFORWARD**: is each layer to compute. the compute function $z = w.a + b$, being $a = \sigma(z)$
- **OUTPUT ERROR**: is the output error by using the formula
- **BACKPROPAGATION**: we can back propagate the error; for each layer, by using the formula
- **OUTPUT**: We calculate the gradient descent with respect to any weight and bias by using the formulas

IV. CONCLUSIONS

Deep learning isn't synonymous with AI or even with machine learning. Artificial intelligence is an ancient, broad field that can generally be defined as "all attempts to automate cognitive processes"—in other words, the automation of thought. This can range from the very basic, such as an Excel spreadsheet, to the very advanced, like a humanoid robot that can walk and talk. Machine learning is a specific subfield of AI that aims at automatically developing programs (called models) purely from exposure to training data. This process of turning data into a program is called learning. Although machine learning has been around for a long time, it only started to take off in the 1990s. Deep learning is one of many branches of machine learning, where the models are long chains of geometric functions, applied one after the other. These operations are structured into modules called layers: deep-learning models are typically stacks of layers or, more generally, graphs of layers. These layers are parameterized by weights, which are the parameters learned during training. The knowledge of a model is stored in its weights, and the process of learning consists of finding good values for these weights. Even though deep learning is just one among many approaches to machine learning, it isn't on an equal footing with the others. Deep learning is a breakout success. In the span of only a few years, deep learning has achieved tremendous break throughs across a wide range of tasks that have been historically perceived as extremely difficult for computers, especially in the area of machine perception: extracting useful information from images, videos, sound, and more.
Given sufficient training data (in particular, training data appropriately labelled by humans), it's possible human can extract almost any data from perceptual. Hence, it's sometimes said that deep learning has solved perception, although that's true only for a fairly narrow definition of perception. As Feynman once said about the universe, "It's not complicated, it's just a lot of it." [2]

REFERENCES

[1] François Chollet : Deep Learning ©2018 by Manning Publications Co ; p 312-315
[2] Richard Feynman, interview, The World from another Point of View, Yorkshire Television, 1972.
[3] An Introduction to neural networks By Kevin Gurney University of Sheffield 2004 p 16-18, p 12-15
[4] https://en.wikipedia.org/wiki/Neural_network
[5] Montanari, U.: 1974, "Networks of Constraints: Fundamental Properties and Applications to Picture Processing", Information Sciences 7, 95-132

