International Journal of Computer Sciences and Engineering Open Access
Review Paper Vol.-7, Special Issue-18, May 2019 E-ISSN: 2347-2693

Artificial Neural Networks in Compute: A Review

Soumya Dhol[1*], Itu Chakraborty[2]

[1,2]Dept. of Bachelor in Computer Application, Pailan College of Management & Technology, Kolkata, India

Corresponding Author: soumyadhol6406@gmail.com

Available online at: www.ijcseonline.org

***Abstract* :** Classification is one of the data mining problems receiving enormous attention in the database community. Although artificial neural networks (ANNs) have been successfully applied in a wide range of machine learning applications, they are however often regarded as black boxes, *i.e.*, their predictions cannot be explained. ANN methods have not been effectively utilized for data mining tasks because how the classifications were made is not explicitly stated as symbolic rules that are suitable for verification or interpretation by human experts. With the proposed approach, concise symbolic rules with high accuracy, that are easily explainable, can be extracted from the trained ANNs. Extracted rules are comparable with other methods in terms of number of rules, average number of conditions for a rule, and the accuracy.

***Keywords*—**Processing,Networks,neurons(keywords)

I. INTRODUCTION

The fundamental processing element of a neural network is a neuron. This building block of human awareness encompasses a few general capabilities. Basically, a biological neuron receives inputs from other sources, combines them in some way, performs a generally nonlinear operation on the result, and then outputs the final result.

Recent experimental data has provided further evidence that biological neurons are structurally more complex than the simplistic explanation above. They are significantly more complex than the existing artificial neurons that are built into today's artificial neural networks. As biology provides a better understanding of neurons, and as technology advances, network designers can continue to improve their systems by building upon man's understanding of the biological brain. But currently, the goal of artificial neural networks is not the grandiose recreation of the brain. On the contrary, neural network researchers are seeking an understanding of nature's capabilities for which people can engineer solutions to problems that have not been solved by traditional computing.

Some applications require "black and white," or binary, answers. These applications include the recognition of text, the identification of speech, and the image deciphering of scenes. These applications are required to turn real world inputs into discrete values. These potential values are limited to some known set, like the ASCII characters or the most common 50,000 English words. Because of this limitation of output options, these applications don't always utilize networks composed of neurons that simply sum up, and thereby smooth, inputs. These networks may utilize the binary properties of ORing and ANDing of inputs. These functions, and many others, can be built into the summation and transfer functions of a network.

Other applications might simply sum and compare to a threshold, thereby producing one of two possible outputs, a zero or a one. Other functions scale the outputs to match the application, such as the values minus one and one. Some functions even integrate the input data over time, creating time-dependent networks.

II. ARTIFICIAL NEURAL NETWORK AND IT'S CONTRIBUTION TO MACHINE LEARNING

Artificial Neural Networks (ANN) is a computational nonlinear model which is widely used in Machine Learning and is considered to be a prominent component of futuristic Artificial Intelligence. The neural part of the name aptly suggests that these are brain-inspired systems which are intended to replicate the way humans learn through their biological nervous system.

Working of Ann

In common **ANN** implementations, the signal at a connection between artificial neurons are a real number, and the output of each artificial neuron is computed by some non-linear function of the sum of its inputs. The original goal of the **ANN** approach was to solve problems in the same way that a human brain would.

Artificial Network Operations: The other part of the "art" of using neural networks revolve around the myriad of ways these individual neurons can be clustered together. This clustering occurs in the human mind in such a way that information can be processed in a dynamic, interactive, and self-organizing way. Biologically, neural networks are constructed in a three-dimensional world from microscopic components. These neurons seem capable of nearly unrestricted interconnections. That is not true of any proposed, or existing, man-made network. Integrated circuits, using current technology, are two-dimensional devices with a limited number of layers for interconnection. This physical reality restrains the types, and scope, of artificial neural networks that can be implemented in silicon.

Although there are useful networks which contain only one layer, or even one element, most applications require networks that contain at least the three normal types of layers - input, hidden, and output. The layer of input neurons receives the data either from input files or directly from electronic sensors in real-time applications. The output layer sends information directly to the outside world, to a secondary computer process, or to other devices such as a mechanical control system. Between these two layers can be many hidden layers. These internal layers contain many of the 9 neurons in various interconnected structures. The inputs and outputs of each of these hidden neurons simply go to other neurons.

Training an Artificial Neural Network: Once a network has been structured for a particular application, that network is ready to be trained. To start this process the initial weights are chosen randomly. Then, the training, or learning, begins. There are two approaches to training - supervised and unsupervised. **Supervised training** involves a mechanism of providing the network with the desired output either by manually "grading" the network's performance or by providing the desired outputs with the inputs. **Unsupervised training** is where the network has to make sense of the inputs without outside help. The vast bulk of networks utilize supervised training. Unsupervised training is used to perform some initial characterization on inputs. However, in the full blown sense of being truly self learning, it is still just a shining promise that is not fully understood, does not completely work, and thus is relegated to the lab.

III. MAJOR COMPONENTS OF AN ARTIFICIAL NEURON

This section describes the seven major components which make up an artificial neuron. These components are valid whether the neuron is used for input, output, or is in one of the hidden layers. **Component 1. Weighting Factors:** A neuron usually receives many simultaneous inputs. Each input has its own relative weight which gives the input the impact that it needs on the processing element's summation function. These weights perform the same type of function as do the the varying synaptic strengths of biological neurons. In both cases, some inputs are made more important than others so that they have a greater effect on the processing element as they combine to produce a neural response. Weights are adaptive coefficients within the network that determine the intensity of the input signal as registered by the artificial neuron. They are a measure of an input's connection strength. These strengths can be modified in response to various training sets and according to a network's specific topology or through its learning rules. **Component 2. Summation Function**: The first step in a processing element's operation is to compute the weighted sum of all of the inputs. Mathematically, the inputs and the corresponding weights are vectors which can be represented as (i1, i2 . . . in) and (w1, w2 . . . wn). The total input signal is the dot, or inner, product of these two vectors. This simplistic summation function is found by multiplying each component of the i vector by the corresponding component of the w vector and then adding up all the products. Input1 = i1 * w1, input2 = i2 * w2, etc., are added as input1 + input2 + . . . + inputn. The result is a single number, not a multi-element vector. Geometrically, the inner product of two vectors can be considered a measure of their similarity. If the vectors point in the same direction, the inner product is maximum; if the vectors point in opposite direction (180 degrees out of phase), their inner product is minimum. The summation function can be more complex than just the simple input and weight sum of products. The input and weighting coefficients can be combined in many different ways before passing on to the transfer 23 function. In addition to a simple product summing, the summation function can select the minimum, maximum, majority, product, or several normalizing algorithms. The specific algorithm for combining neural inputs is determined by the chosen network architecture and paradigm. Some summation functions have an additional process applied to the result before it is passed on to the transfer function. This process is sometimes called the activation function. The purpose of utilizing an activation function is to allow the summation output to vary with respect to time. Activation functions currently are pretty much confined to research. Most of the current network implementations use an "identity" activation function, which is equivalent to not having one. Additionally, such a function is likely to be a component of the network as a whole rather than of each individual processing element component. **Component 3. Transfer Function**: The result of the summation function, almost always the weighted sum, is transformed to a working output through an algorithmic process known as the transfer function. In the transfer function the summation total can be compared with some threshold to determine the neural output. If the sum is greater than the threshold value, the processing element generates a signal. If the sum of the input and weight products is less than the threshold, no signal (or some inhibitory signal) is generated. Both types of response are significant. The threshold, or transfer function, is generally non-linear. Linear (straight-line) functions are limited because the output is simply proportional to the input. Linear functions are not very useful. That was the problem in the earliest network models as noted in Minsky and Papert's book Perceptrons. The transfer function could be something as simple as depending upon whether the result of the summation function is positive or negative. The network could output zero and one, one and minus one, or other numeric combinations. The transfer function would then be a "hard limiter" or step function.

Component 4. Scaling and Limiting: After the processing element's transfer function, the result can pass through additional processes which scale and limit. This scaling simply multiplies a scale factor times the transfer value, and then adds an offset. Limiting is the mechanism which insures that the scaled result does not exceed an upper or lower bound. This limiting is in addition to the hard limits that the original transfer function may have performed. This type of scaling and limiting is mainly used in topologies to test biological neuron models, such as James Anderson's brain-state-in-the-box. **Component 5. Output Function (Competition)**: Each processing element is allowed one output signal which it may output to hundreds of other neurons. This is just like the biological neuron, where there are many inputs and only one output action. Normally, the output is directly equivalent to the transfer function's result. Some network topologies, however, modify the transfer result to incorporate competition among neighboring processing elements. Neurons are allowed to compete with each other, inhibiting processing elements unless they have great strength. Competition can occur at one or both of two levels. First, competition determines which artificial neuron will be active, or provides an output. Second, competitive inputs help determine which processing element will participate in the learning or adaptation process. **Component 6. Error Function and Back-Propagated Value**: In most learning networks the difference between the current output and the desired output is calculated. This raw error is then transformed by the error function to match particular network architecture. The most basic architectures use this error directly, but some square the error while retaining its sign, some cube the error, other paradigms modify the raw error to fit their specific purposes. The artificial neuron's error is then typically propagated into the learning function of another processing element. This error term is sometimes called the current error. The current error is typically propagated backwards to a previous layer. Yet, this back-propagated value can be either the current error, the current 26 error scaled in some manner (often by the derivative of the transfer function), or some other desired output depending on the network type. Normally, this back-propagated value, after being scaled by the learning function, is multiplied against each of the incoming connection weights to modify them before the next learning cycle. **Component 7. Learning Function**: The purpose of the learning function is to modify the variable connection weights on the inputs of each processing element according to some neural based algorithm. This process of changing the weights of the input connections to achieve some desired result can also be called the adaptation function, as well as the learning mode. There are two types of learning: supervised and unsupervised. Supervised learning requires a teacher. The teacher may be a training set of data or an observer who grades the performance of the network results. Either way, having a teacher is learning by reinforcement. When there is no external teacher, the system must organize itself by some internal criteria designed into the network. This is learning by doing.

Networks for Prediction: The most common use for neural networks is to project what will most likely happen. There are many applications where prediction can help in setting priorities. For example, the emergency room at a hospital can be a hectic place. To know who needs the most time critical help can enable a more successful operation. Basically, all organizations must establish priorities which govern the allocation of their resources. This projection of the future is what drove the creation of networks of prediction.

IV. LIFE CHANGING AREAS OF ANN

Various model application neural networks :

The most developed area is work that results in “smarter medical devices,” says Bradley Greger, principal investigator at the Neural Engineering Lab at Arizona State University. From teamwork among experts in various fields is emerging thinking that neural inferences need to integrate sensory and motor components. “For decades they were talked about individually. But people are saying, ‘We can’t think about it that way,’” and work on simultaneous motor and sensory control automatic interfaces is underway. The result is a process that replicates how the brain works and will result in devices that will be able to read electrical and chemical signals from the nervous system and respond much more like the human body does.

Sensory and motor are tightly interwoven at the neural level, Greger says. “I can’t move if I don’t have good sensation. And I have to move to get good sensation. If I want to move my hand, I can’t do that without sensory input as a guide. It’s the neural interfaces, the physical connections to the brain that are going to let us do that.”

He says one straightforward example is the control of a robotic arm for someone paralysed. Current technology is guided by vision. The person has to look and pay careful attention to what they are doing because a sense of touch or arm position is not built into the device. The person thinks of what they want to do, but has to look and see to accomplish it. “That’s not how we really move,” he points out. “The arm has a massively complex sensory system that we unconsciously have access to that helps control our movements without thinking about it.”

There now are groups, including Greger’s own lab, working on neutrally controlled prosthesis that incorporate sensors of the arm leading into the brain providing a sense of touch and arm position.

Another project at his lab in this space involves vision restoration prosthesis for someone who is blind. They are hooked up to a camera that connects directly into the visual processing parts of the brain. “That seems like simple sensory access,” Greger says. “But it’s very complex. Your sense of vision is controlled by and intimately linked with how you move your eyes. The visual system has to know what your eyes are doing and how you move them in order for you to process that information.”

Another device being tested is an implant to control epileptic seizures. “It’s breakthrough in the sense that they are helping people right now. It looks very promising,” Greger says.

Finally, another potentially life-enhancing and promising technology is an implant for people with chronic pain not relieved by medications. The device delivers electrical impulses to the spinal cord to mask pain signals before they reach the brain.

Better Artificial Intelligence:

So much of the terminology being used in artificial intelligence comes from neural technology—neuronets or brainlike or cognitive, Greger says. Even so, the artificial intelligence work currently involves only one level of processing that goes on in the brain, individual groups of neurons, or cells that talk to each other through pulses of electricity.

There are other interactions that happen with electrical fields that are part of the computational process, and the complicated architecture of neural circuits are activities that also need to be considered, he says. He likened the work to building a model of a Ferrari and expecting it to behave like a real Ferrari. “That’s not going to happen. You are trying to interface with the brain without taking into account how the structure really functions, that it’s multidimensional, multi-scale and squishy, in terms of both its physical structure and function. We keep trying to put it in non-biological form. Once we start taking biological factors into account, it will give people a lot more knowledge about problem solving for intelligent control of machines,” Greger says.

Building Protoplasmic Circuits :

The third big area—Neuro technology that is truly biological—is “where it gets really crazy,” Greger says. “We are seeing the beginning of this with stem cells and artificial tissues. As the fields of cellular neurobiology and systems biology mature, we can start building complex cellular structures and cellular-like tissues, replications of areas of the brain for replacement or for controlling circuits.”

Researchers have started building very simple protoplasmic circuits and programming them to perform certain functions, leading toward autonomous cars being driven by protoplasm, a little brain instead of electrical circuit. “It would be much more powerful in terms of its computational ability than a digital circuit,” he says. “We are decades away from seeing your Amazon delivery drone controlled by protoplasm, but it’s not total science fiction.”

Summary: In summary, artificial neural networks are one of the promises for the future in computing. They offer an ability to perform tasks outside the scope of traditional processors. They can recognize patterns within vast data sets and then generalize those patterns into recommended courses of action. Neural networks learn they are not programmed. Yet, even though they are not traditionally programmed, the designing of neural networks does require a skill. It requires an "art." This art involves the understanding of the various network topologies, current hardware, current software tools, the application to be solved, and a strategy to acquire the necessary data to train the network. This art further involves the selection of learning rules, transfer functions, summation functions, and how to connect the neurons within the network. Then, the art of neural networking requires a lot of hard work as data is fed into the system, performances are monitored, processes tweaked, connections added, rules modified, and on and on until the network achieves the desired results. These desired results are statistical in nature. The network is not always right. It is for that reason that neural networks are finding themselves in applications where humans are also unable to always be right. Neural networks can now pick stocks, cull marketing prospects, approve loans, deny credit cards, tweak control systems, grade coins, and inspect work. Yet, the future holds even more promises. Neural networks need faster hardware. They need to become part of hybrid systems which also utilize fuzzy logic and expert systems. It is then that these systems will be able to hear speech, read handwriting, and formulate actions. They will be able to become the intelligence behind robots that never tire nor become distracted. It is then that they will become the leading edge in an age of "intelligent" machines.

V. CONCLUSION

Neural networks are suitable for predicting time series mainly because of learning only from examples , without any need to add additional information that can bring nore confusion than prediction effect. Neural networks are able to generalize and are resistant to noise. On the other hand, it is generally not possible to determine exactly what a neural network learned and it is also hard to estimate possible prediction error. However, Neural networks were often successfully used for predicting time series.

VI. FUTURE SCOPE

The scope of future work can deal with Incremental learning, which stores the existing model and processes the new incoming data more efficiently. More specifically, the models with incremental learning can be used in categorization process to improve the following aspects in each type of problems.

1. The optimization of the high suitability index terms in BBO is essential for classification. Each one of the high suitability index terms can be seen as an objective to be optimized in a multi-objective approach and this could reveal further scope for improvement in the classification.
2. In future, the hybrid networks must be designed and implemented to classify a huge set of documents to achieve improved performance to prove its effectiveness.
3. This model can be enhanced in future, by making it suitable for colour images and real-time images, in an image restoration system. With the help of a digital camera, real-time images can be easily tracked, which is further fed to this system making it a reliable and flexible architecture.
4. The image classification model can be enhanced in future, by including more low-level features such as shape and spatial location features apart from optimizing the weights and learning rate of the neural network.

REFERENCES

[1]. DAWSON, CHRISTIAN W (1998). "An **artificial neural network** approach to rainfall-runoff modeling". Hydrological Sciences Journal.

Author's Profile

Mr. Soumya Dhol has completed secondary education from Muchisha Haridas Krishi Silpa Vidyapith High School under WBSE in 2015. He has completed higher secondary education from Muchisha Haridas Krishi Silpa Vidyapith High School under WBHSE in 2017. He is currently pursuing Bachelor in Computer Application from Pailan College of Management & Technology under MAKAUT.

Miss Itu Chakraborty has completed secondary education from Aligunj Rishi Rajnarayan Balika Vidyalaya under WBSE in 2014. She has completed higher secondary education from Midnapore Collegiate GIRLS High School under WBHSE in 2017. She is currently pursuing Bachelor in Computer Application from Pailan College of Management & Technology under MAKAUT.