

NEUROCOMPUTATION FOR MICROSTRIP ANTENNA

Sonia Sharma

ECE Department, University Institute of Engineering and Technology, MDU, Rohtak, India

Abstract: A Neural Network is a powerful computational tool that is able to represent complex input/output relationships. The motivation for the development of neural network technology to check from the desire to develop an artificial system that could perform intelligent tasks similar to those performed by the human brain. The paper provides a tutorial overview of neural network and how used neural networks for antennas, here we demonstrate with gives overview of calculate the resonant frequencies for an equilateral triangular microstrip antenna geometry using NNs. Neural networks resemble the human brain in the following two ways: A neural network acquires knowledge through learning from experience and neural network's knowledge is stored within inter-neuron connection strengths known as synaptic weights.

Keywords: Neural Networks, Equilateral Triangular Microstrip Antenna.

1. Introduction

The neural network structure has two types of basic components one is Processing elements and other is Interconnections between them. The processing elements are known as neurons and the connections between them are called as Links or synapses. Every link has a corresponding weight parameter associated with it. Each neuron receives input from other neurons connected to it, processes the information and produces an output. Neurons that receive input from other neurons and whose outputs are input for other neurons are known as hidden neurons in the network. The dendrites receive signals from other neurons. The axon can be considered as a long tube, which divides further into branches as shown in Fig.1. Depending upon the type of neuron, the number of synapses connections from other neurons may range from a few hundreds to 100,000. The cell body of a neuron sums the incoming signals from dendrites as well as the signals from several synapses on its surface. A neuron will send an impulse to its axon if sufficient signals are received to stimulate the neuron to its threshold level. The interest in neural networks comes from the network's ability to mimic human brain as well as its ability to learn and respond.

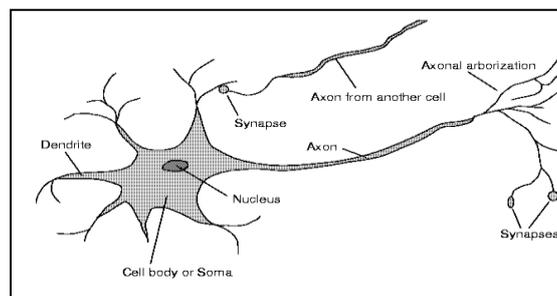


Fig.1. Neuron and its connection

2. How the Human Brain Learns?

In the human brain, a typical neuron collects signals from others through a host of fine structures called *dendrites*. The neuron sends out spikes of electrical activity through a long, thin strand known as an *axon*, which splits into thousands of branches. At the end of each branch, a structure called a *synapse* converts the activity from the axon into electrical effects that inhibit or excite activity from the axon into electrical effects that inhibit or excite activity in the connected neurons. When a neuron receives excitatory input that is sufficiently large compared with its inhibitory input, it sends a spike of electrical activity down its axon. Learning occurs by changing the effectiveness of the synapses so that the influence of one neuron on another changes.

3. LEARNING IN NEURAL NETWORKS

Learning is a process by which the free parameters of a neural network are adapted through a process of stimulation by the environment in which the network is embedded. The type of learning is determined by the

manner in which the parameter changes take place. All learning methods used for neural networks can be classified into two major categories:

1. SUPERVISED LEARNING which incorporates an external teacher, so that each output unit is told what its desired response to input signals taught to be. During the learning process global information may be required.
2. UNSUPERVISED LEARNING uses no external teacher and is based upon only local information. It is also referred to as self-organization, in the sense that it self-organizes data presented to the network and detects their emergent collective properties.

From Human Neurons to Artificial Neurons

We conduct these neural networks by first trying to deduce the essential features of neurons and their interconnections shown in fig.2. We then typically program a computer to simulate these features. However because our knowledge of neurons is incomplete and our computing power is limited, our models are necessarily gross idealizations of real networks of neurons.

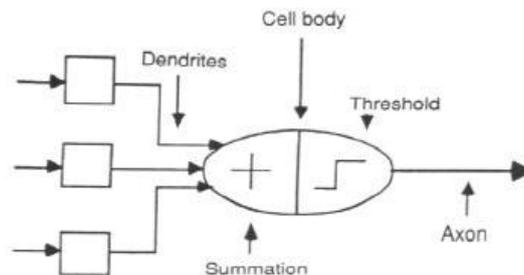


Fig.2. Neuron Model

An engineering approach

An artificial neuron is a device with many inputs and one output shown in fig.3. The neuron has two modes of operation; the training mode and the using mode. In the training mode, the neuron can be trained to fire (or not), for particular input patterns. In the using mode, when a taught input pattern is detected at the input, its associated output becomes the current output. If the input pattern does not belong in the taught list of input patterns, the firing rule is used to determine whether to fire or not.

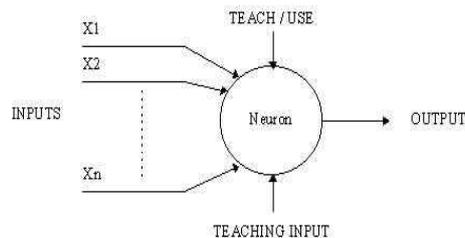


Fig.3. A simple neuron

Feed-forward networks: Feed-forward ANNs allow signals to travel one way only; from input to output. There is no feedback (loops) i.e. the output of any layer does not affect that same layer. Feed forward ANNs tend to be straight forward networks that associate inputs with outputs shown in fig.4. They are extensively used in pattern recognition. This type of organization is also referred to as bottom-up or top-down. [1]

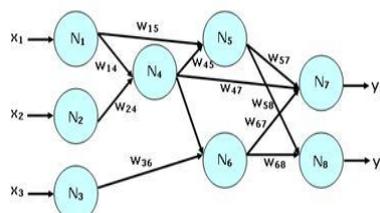


Fig.4. Feed-forward networks

Feedback networks: Feedback networks can have signals travelling in both directions by introducing loops in the network as shown in fig.5. Feedback networks are very powerful and can get extremely complicated.

Feedback networks are dynamic; their 'state' is changing continuously until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found. Feedback architectures are also referred to as interactive or recurrent, although the latter term is often used to denote feedback connections in single-layer organizations.[2],[3]

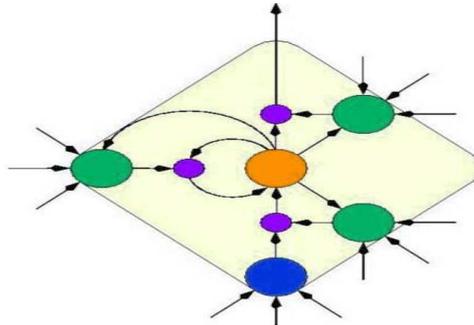


Fig.5.Feedback Network

Neural network model development for antennas

The NN does not represent any antenna unless we train it with antenna data. To develop an NN model, we need to identify input and output parameters of the component in order to generate and preprocess data, and then use this data to carry out NN training.

4. Problem Formulation and Data Processing

1. ANN Inputs and Outputs The first step toward developing an NN is the identification of inputs (\mathbf{x}) and outputs (\mathbf{y}). The output parameters are determined based on the purpose of the NN model. Other factors influencing the choice of outputs are (1) ease of data generation and (2) ease of incorporation of the neural model into circuit simulators. Neural model input parameters are those antenna parameters that affect the output parameter values.

2. Data Range and Sample Distribution The next step is to define the range of data to be used in ANN model development and the distribution of \mathbf{x} - \mathbf{y} samples within that range. Suppose the range of input space (i.e., \mathbf{x} -space) in which the neural model would be used after training (during design) is $[\mathbf{x}_{\min}, \mathbf{x}_{\max}]$. Training data is sampled slightly beyond the model utilization range that is, $[\mathbf{x}_{\min} - \Delta, \mathbf{x}_{\max} + \Delta]$, in order to ensure reliability of the neural model at the boundaries of model utilization range.

3. Data Generation In this step, \mathbf{x} - \mathbf{y} sample pairs are generated using either simulation data or measured data. The generated data could be used for training the NN and testing the resulting NN model. In practice, both simulations and measurements could suffer from errors that will affect the accuracy of the NN predictions. Considering this, we introduce a vector \mathbf{d} to represent the outputs from simulation/measurement corresponding to an input \mathbf{x} . Data generation is then defined as the use of simulation/measurement data to obtain sample pairs $(\mathbf{x}_k, \mathbf{d}_k)$, $k = 1, 2, \dots, P$. The total number of samples P is chosen such that the developed neural model best represents the given problem \mathbf{f} . A general guideline is to generate a larger number of samples for a nonlinear high-dimensional problem and fewer samples for a relatively smooth low-dimensional problem.

4. Data Organization The generated (\mathbf{x}, \mathbf{d}) sample pairs could be divided into three sets, namely, training data, validation data, and test data. Let T_r , V , T_e , and D represent index sets of training data, validation data, test data, and generated (available) data, respectively. *Training* data is utilized to guide the training process, that is, to update the NN weight parameters during training. *Validation* data is used to monitor the quality of the NN model during the training and to determine termination criteria for the training process. Test data is used to independently examine the final quality of the trained neural model in terms of accuracy and generalization capability.

5. Data Preprocessing Binary data (0's and 1's), the orders of magnitude of various input (\mathbf{x}) and output (\mathbf{d}) parameter values in antenna applications can be very different from one another. As such, a systematic preprocessing of training data called scaling is desirable for efficient NN training. At the end of this step, the scaled data is ready to be used for training [5]. A flowchart summarizing major steps in NN training and testing is shown in Fig.6.

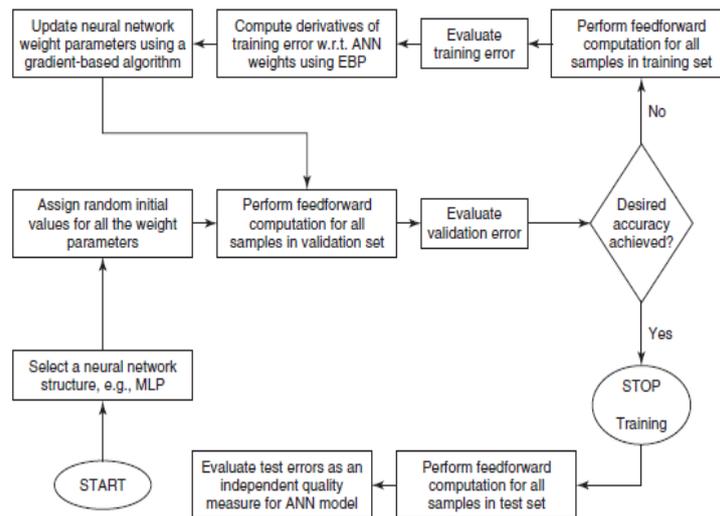


Fig.6. Flowchart demonstrating NN training, neural model testing, and use of training, validation and test data sets in ANN modeling approach.

5. ANN Technique for Antennas:

In order to apply ANN techniques for antenna problem, the first task is to bring the problem to a form that is suitable for an ANN application: that is, the problem has to be formulated into one of the categories of mapping, function approximation, or classification. Data has to be generated to create the training environment for the ANN. This can be done using experiments or simulation packages. Depending on the type of problem and looking to other factors, like number of data patterns available, number of input/output parameters, or complexity of the problem, a training algorithm has to be chosen. Standard training algorithms [6] are available. The aim of the training is to get a specified set of weights that can be used to determine the outputs of the ANN or the system as a whole. The steps involved to calculate the resonant frequencies for an equilateral triangular microstrip antenna geometry using NNs are shown in fig.7. below:

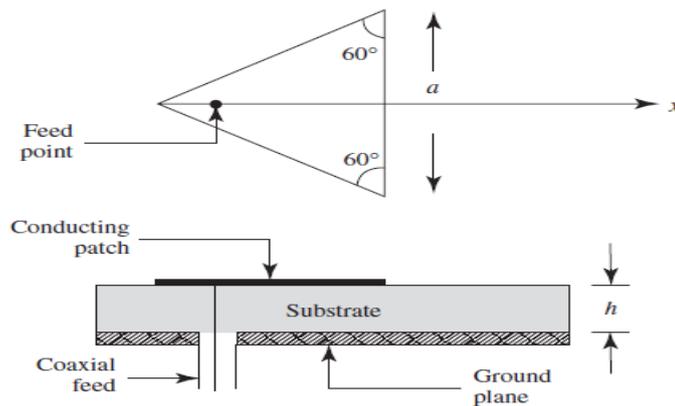


Fig.7. Geometry of Equilateral Triangular Microstrip Antenna

Step1.Generation of Data: Experimentally measured resonant frequency data available in the literature were used for training of the NN. Fifteen sets of such data values were collected for different dimensions and modes of the antenna.

Step2. Choosing the Training Algorithm and the Network Structure: A back propagation training algorithm was used in this case because this is the most widely used training algorithm for function implementation and at the same time it is easy to use. The task of choosing the number of appropriate neurons in the input and output layers is problem dependent. In this particular problem the intention is to find the resonant frequency, which is a function of the antenna parameters: side length (a), height of the substrate (h), substrate permittivity (ϵ_r), and the mode numbers (m and n). Therefore five neurons in the input layer and one neuron in the output layer are required. There is no specific rule to determine the number of neurons in the hidden layer(s).

Step3. Choosing the Training Parameters: After fixing the network structure, the efficiency of training depends on many training parameters such as learning rate, momentum, and training tolerance. The optimum values of these parameters are problem dependent and found by trial and error or experience. This is more of an art than science. The values of the different training parameters for this problem are given in table.1. below.

Table.1: Training Parameters values

Parameter	Value
Number of input layer neuron(s)	5
Number of output layer neuron(s)	1
Number of hidden layers	2
Number of first hidden layer neuron(s)	5
Number of second hidden layer neurons	3
Learning rate	0.3
Momentum	0.5
Number of epochs	25,000

Step4. Testing of the Developed Network: In order to ascertain that the developed network is properly trained, the trained NN has to be tested for validation with the test data set. If the network is providing proper results within the desired tolerance, then the developed network will be treated as a properly trained one. A typical set of results from the tested NN, in this problem, is given in table.2.

Table.2: Comparison of Measured and Calculated Resonant Frequencies of the First Five Modes of an Equilateral Triangular Microstrip Antenna

Mode	f_{me} (MHz)	f_{mn} Present Method (MHz)
TM ₁₀	1519	1526
TM ₁₁	2637	2637
TM ₂₀	2995	2995
TM ₂₁	3973	3973
TM ₃₀	4439	4439
TM ₁₀	1489	1478
TM ₁₁	2596	2596
TM ₂₀	2969	2969
TM ₂₁	3968	3968
TM ₃₀	4443	4443
TM ₁₀	1280	1280
TM ₁₁	2242	2242
TM ₂₀	2550	2550
TM ₂₁	3400	3400
TM ₃₀	3824	3829

REFERENCES

- [1] Simon Haykin, "Neural Networks", Second edition by, Prentice Hall of India, 2005.
- [2] Christos Stergiou and Dimitrios Siganos, "Neural Networks", Computer Science Deptt. University of U.K., Journal, Vol. 4, 1996
- [3] Jordan B.Pollack, "Connectionism: Past, Present and Future", Computer and Information Science Department, The Ohio State University, 1998.
- [4] Q. J. Zhang, K. C. Gupta, and V. Devabhaktuni, Artificial neural networks for RF and microwave design—from theory to practice, IEEE Trans. Microwave Theory Tech., Vol. 51, No. 4, pp. 1339–1350, April 2003.
- [5] Christos G. Christodoulou And Amalendu Patnaik, "Neural Networks For Antennas" In Modern Antenna Handbook By Constantine A. Balanis.
- [6] S. Haykins, Neural Networks: A Comprehensive Foundation, Ieee Press/Ieee Computer Society Press, New York, 1994.
- [5] Sonia Sharma and C. C. Tripathi Rahul Rishi, "Impedance Matching Techniques for Microstrip Patch Antenna, Indian Journal of Science and Technology, Vol. 10(28), pp. 1-16.
- [6] Sonia Sharma and C.C. Tripathi, "An Integrated Antenna for Cognitive Radio Application" Radioengineering, vol. 26, no. 3, pp. 1-9, September 2017
- [7] Sonia Sharma, C.C.Tripathi, "Design of Frequency Reconfigurable Antenna For Multi Standard Mobile Communication", Institute of Research in Engineering and Technology (IRET), International Journal of Emerging Trends in Electrical and Electronics Vol. 6, Issue-2, pp 8-14, Aug-2013