Artificial Neural Network based Learning in Cognitive Radio

Mithra Venkatesan1, Dr. Anju V. Kulkarni2 and Dr. Radhika Menon3

1Research Scholar, Department of Electronics and Telecommunication Engineering, JSPM's Rajarshi Shahu College of Engineering, Thawade, Pune-411033.

2,3Department of Electronics & Telecommunication, Padmashree Dr. D.Y. Patil Institute of Engg. & Technology, Pune-411018

mithrav@rediffmail.com, anju_k64@yahoo.co.in, radhika.tharoor@gmail.com

Abstract: Over the last decade the world of wireless communications has been undergoing some crucial changes, which have brought it at the forefront of international research and development interest, eventually resulting in the advent of a multitude of innovative technologies and associated products such as WiFi, WiMax, 802.20, 802.22, wireless mesh networks and software defined radio. Such a disparate and highly varying radio environment calls for intelligent management, allocation and usage of a scarce resource, namely the radio spectrum. One of the most prominent emerging technologies that promise to handle such situations is Cognitive Radio. A Cognitive Radio system has the ability to adjust its operating parameters, observe the results and, eventually take actions, that is to say, decide to operate in a specific radio configuration (i.e. radio access technology, carrier frequency, modulation type, etc.), expecting to move the radio toward some optimized operational state. In such a process, learning mechanisms that are capable of exploiting measurements sensed from the environment, gathered experience and stored knowledge, are judged as rather beneficial for guiding decisions and actions. Framed within this statement, this paper introduces learning schemes that are based on artificial neural networks and can be used for discovering the performance (e.g. data rate) that can be achieved by a specific radio configuration in a cognitive radio system. Interesting scenarios, which include both commercial off-the-shelf and simulation hardware/software products, are to be mobilized for the performance assessment work, conducted in order to design and use an appropriate neural network structure, while the benefits of incorporating such learning schemes into cognitive radio systems is discussed.

Keywords: Cognitive Radio, Artificial Neural Networks, Learning, Feed forward networks

I. INTRODUCTION

Cognitive Radio systems are based on Software Defined Radio technology and utilize intelligent software packages that enrich their transceivers with the highly attractive properties of self-awareness, adaptability and capability to learn. A Cognitive Radiosystem has the ability to adjust its operating parameters, observe the results and, eventually take actions, that is to say, decide to operate in a specific radio configuration (i.e. radio access technology, carrier frequency, modulation type, etc.), expecting to move the radio toward some
optimized operational state. In such a process, learning mechanisms that are capable of exploiting measurements sensed from the environment, gathered experience and stored knowledge, are judged as rather beneficial for guiding decisions and actions. The Cognitive Engine, the intelligent system behind the Cognitive Radio, combines sensing, learning, and optimization algorithms to control and adapt the radio system from the physical layer and up the communication stack. To this effect, many different learning techniques are available and can be used by a Cognitive Radio ranging from pure lookup tables to arbitrary combinations of soft Computing techniques, which include among others: Artificial Neural Networks, evolutionary/Genetic Algorithms, reinforcement learning, fuzzy systems, Hidden Markov Models, etc. The proposed work contributes in this direction, aiming to develop a learning scheme and work towards solving problems related to learning phase of Cognitive Radio systems. In the near future, such learning schemes are expected to assist a Cognitive Radio system to compare among the whole of available, candidate radio configurations and finally select the best one to operate in.

Different learning models are built toward spectrum behavior, spectrum sensing and Spectrum learning using approaches such as Collaborative filtering [1], self learning algorithms [2], and machine learning techniques [3]. Learning Models are also built towards Dynamic Channel Selection and Dynamic Spectrum Access using approaches such as Markov Model [4], Neural Networks [5], and Game Theory [6]. The learning engine is the intelligence behind the cognitive radio where the context awareness and the capacity to learn is implemented through methods like Support Vector Machine [7], Neural Networks[8], Genetic Algorithms[9], Reinforcement learning[10]. The decision maker of Cognitive Radio is built through a neural network based model [11]. Signal classification to detect the presence of unknown signal is implemented using self organizing maps [12]. Learning Models are also built towards finding parameters to decide which the best configuration to operate with is [13][14]. Transmission rate prediction is done through a learning model built using Neural Fuzzy Interference System [15]. Some learning models use supervised algorithms [13][14] while certain use unsupervised algorithms such as self organized maps [16].

This paper contributes in this direction, by a learning scheme that relies on artificial neural networks and aims at solving the problems related to the channel estimation and predictive modeling phase of cognitive radio systems. As a result, the proposed scheme can facilitate the cognitive terminal in making the best decision regarding the configuration in which it should operate. The performance assessment work that needs to be conducted in order to design and use an appropriate neural network structure is also described in the paper.

II. NEURAL NETWORK REVIEW

Artificial neural networks (ANN or simply NN) are made up of artificial neurons interconnected with each other to form a programming structure that mimics the behavior and neural processing (organization and learning) of biological neurons. A neural network consists of a pool of simple processing units, the ‘neurons’. Within NNs three types of neurons are distinguished: input neurons which receive data from outside the NN and are organized in the so called input layer, output neurons which send data out of the NN and generally comprise the output layer, and hidden neurons whose input and output signals remain within the NN and form the so called hidden layer (or layers). The topology of a NN plays an important role for its achievable performance. Depending on the pattern of connections that a NN uses to propagate data among the neurons, it can be classified into one over two basic (non exhaustive) categories. (a) Feed-forward NNs, where data enters at the inputs and passes through the network, layer by layer, until it arrives at the outputs, with classical examples being the Perceptron and Adaline. (b) Recurrent NNs that contain feedback connections, which are connections extending from outputs of neurons to inputs of neurons within the same or previous layers. Examples of recurrent networks have been presented by Elman and Hopfield. In the analysis within this paper, we treat both feed-forward and recurrent networks.

In any manifestation, a NN has to be configured such that the application of a set of inputs produces the desired set of outputs. This can be achieved by properly adjusting the weights $w_{jk}$ of the existing connections among all $(j,k)$ neuron pairs. This process is called learning or training. Learning can be generally distinguished between supervised and unsupervised learning (with reinforcement learning being also an option). In supervised learning, the NN is fed with teaching patterns and trained by letting it change its weights according to some learning rule, the so called back-propagation rule. The NN learns the input–output mapping by a stepwise change of the weights with the objective to minimize the difference between the actual and desired output. In the next step the actual output vector is compared with the desired output. Error values are assigned to each neuron in the output layer. The error values are back-propagated from the output
layer to the hidden layers. The weights are changed so that there is a lower error for a new presentation of the same pattern. As a result of this procedure, the weights on the connections between neurons are adjusted so as to encode the actual knowledge of the NN. At that time, the NN can be used for the purpose that it was initially set up for. On the contrary, in unsupervised learning the NN discovers features of the input data in a statistical manner by developing its own ways of classifying the input irritants. In this paper, we only deal with the supervised learning mode.

III. PROPOSED WORK: COGNITIVE ENGINE BASED NEURAL NETWORK

The following are the steps which are to be followed for proposed methodology:

Step 1: Deciding the requirements of database.
Step 2: Selection of suitable platform for database collection.
Step 3: Setup for database collection.
Step 4: Database filtering.
Step 5: Designing neural network based on
   i. Input output parameters.
   ii. Network type.
      iii. Network parameters
      iv. Database length
Step 6: Analyze the results.
Step 7: Redesign the network.

The data used for the test cases have been obtained from real measurements that took place in a real working environment within our college premises. Specifically, a laptop equipped with an Intel PROset/Wireless card has been used for measuring, among others, the maximum achievable transmission data rate, the signal strength in user predefined time intervals (with the default value being 3 s). The laptop has been setup with a Windows OS and using the ipw3945 driver for the wireless card. Another laptop has been setup with same vision of windows OS but uses Dell Wireless 1702 wireless card. The wireless access point (AP) used was a D-Link broadband router //model WRT54GS// able to operate in both IEEE 802.11 b/g standard modes [23,24]. This comprises the radio configuration (it can be seen as one single configuration given that the operating carrier frequency is the same, i.e. 2.4 GHz in both modes), the capabilities of which need to be discovered.

Intel Wireless Card supports the IEEE 802.11a/b/g/n,802.11. It comes with Intel PROset/Wireless tool software. We can manage wireless connection and setup new connection. There is provision for monitoring advanced statistics like RSSI, No of packets transmitted and received at different data rates, transmitted bytes, received bytes, transmission retries, reception errors etc. The statistics can be logged in form of html file. Intel PROset/Wireless card has been used for measuring, among others, the maximum achievable transmission data rate, the signal strength in user predefined time intervals (with the default value being 3 sec).

All simulation analyses and/or results are to be presented, assuming that the NN-based scheme is tuned in an arbitrary radio configuration, e.g. IEEE WLAN 802.11g. This is actually the radio configuration, the capabilities of which (i.e., data rate) need to be discovered–evaluated. Initially a database is to be formed which would form the input for the neural network. The input would consist of various parameters of the arbitrary radio configuration, e.g. IEEE WLAN 802.11g for cognitive radio. The different data rate is to be collected and will form the target value for the neural network.

Two types of NNs have been selected for the first set of test cases. The first network that is to be used is a feed-forward back-propagation network. The second network type used is the Elman network which is a two-layer back-propagation, recurrent network. All networks used during the investigation procedure are to be setup with hidden and output layer. It is proposed that the NNs use the tanS function for the neurons in their hidden (recurrent) layer, and purelin function for the neuron in their output layer, respectively. For the training session, the input and target values have been properly normalized in the range of [1, 1], in a pre-processing phase. During training, for the networks, the weights and bias values are to be updated according to Levenberg–Marquardt optimization (a.k.a. trainlm) while for the Elman networks they are to be updated according to a gradient descent momentum and an adaptive learning rate method (a.k.a. traingdx). Finally, the Mean Squared Error (MSE) is to be used as a metric for measuring the neural network’s performance.
Initially, two different data sets are to be used, which is to be extracted from the input sequence and served as the target values for teaching and validating the NN: (a) a “training” set (seen data) which is to be used to build the model, i.e. determine its parameters, during the so called training session, (b) a “validation” set (unseen data) which is to be used to measure the performance of the network by holding the parameters found during training constant. With the term “unseen” here we characterize data that have never been used to update the weights of the network. In general, performance on the training only tells us that the model learns what it’s supposed to learn, but it is not a good indicator of performance on unseen data, i.e. whether the NN is able to generalize well or not.[13]

The following figure shows the results obtained:

**Fig 1 Intel PROset/Wireless tool software**

**Fig 2: Plot fit window (comparing nn outputs and targets)**
The output tracks the targets very well for training, testing, and validation, and the R-value is over 0.95 for the total response. If even more accurate results were required, you could try any of these approaches:

- Reset the initial network weights and biases to new values with init and train again.
- Increase the number of hidden neurons.
- Increase the number of training vectors.
- Increase the number of input values, if more relevant information is available.
- Try a different training algorithm (see Speed and Memory Comparison).

The importance of testing the network with both datasets, when searching for the best structure, is significant, since a small error in the training set can be misleading. If the network has not been trained well, it may not learn the basic structure of the data, but rather learn irrelevant details of the individual cases, a.k.a. overfitting. This would lead to a small error during testing with the training set, but in a large error during testing with the validation dataset. Moreover, the number of hidden layers and/or neurons plays a critical role in the learning process, and strongly influences the performance of the network. The use of too few hidden neurons would result in a NN that is unable to learn what we want it to learn. On the other hand, the use of too many hidden neurons would dramatically increase the time needed to learn, without yielding any significant improvement in the performance of the network. There exist some valid rules to set the number of hidden nodes, but in general, it is better to start with a big net, train, and then carefully follow a pruning strategy for gradually reducing the size of the network.[13]

For the test cases of the next scenario, the focus will be on the achievable transmission data rate. Though, the target in this scenario is to build a NN that would be able to predict the achievable bit rate, taken as input the quality of the link and the signal strength of the wireless transceiver. A number of different test cases are to be investigated. Again, all networks are to use the tansig function for the neurons in their hidden (recurrent) layer(s) and the purelin function for the single neuron in their output layer. The bias and weight values are to be updated according to trainlm optimization, during training sessions. Finally, once again, the MSE is to be used for measuring the performance of the neural networks.[14]

IV. CONCLUSION

The implementation of learning schemes will assist the cognitive radios in the derivation and enforcement of decisions regarding the selection of the best radio configuration. This paper proposes learning schemes which are based on Artifical Neural Networks (ANNs) motivated by the fact that ANNs are widely different from conventional information processing as they have the ability to learn from given examples, thus being also able to perform better in cognitive tasks. Scenarios and test cases that are to be used for the derivation of the appropriate NN structures are analytically described.

Our intention for future research on the topic is manifold. First, the exploration upon other crucial, context information that can be used to feed the NN input layer, e.g., location, user preferences or even weather conditions, etc., will be continued in order to achieve even better results. Furthermore, new types and enhanced structures of NNs that have been found to improve both short-term and long-term time-series prediction capabilities will be investigated for application to our scheme, including also NNs that are geared towards unsupervised learning such as Self-Organizing Maps.[16] Last but not least, as long as evidence on the performance capabilities of each candidate radio configuration of the cognitive terminal can be drawn, the optimization process/algorithm for selecting the optimum one also needs to be thoroughly studied as part of our future work.

REFERENCES

[3] LiangYin, SiXing Yin, Weijun Hong, ShuFang Li, Senior Member IEEE, 2011 Military Communications Conference, Spectrum Behaviour in Cognitive Radio Network based on Artificial Neural Networks.


[12] Qiao Cai, Sheng Chen, Xiaochen Li, Nansai Hu, Hibo He, Yu-Dong Yao, Mitola J, International Joint Conference on Neural Networks (IJCNN), An integrated incremental self-organizing map and hierarchical neural network approach for cognitive radio learning, 18-23 July 2010


