ANN-Based Model for Predicting RF Signal Received Power in Indoor Propagation Links

Juan Antonio ROMO ARGOTA and Ignacio ANITZINE
Department of Communication Engineering, University of The Basque Country, Bilbao, Spain

Keywords: Artificial neural networks, Neuron model, Learning rules, Backpropagation, Mobile communications, Ray-tracing, Dominant path.

Abstract. This chapter presents the use of Artificial Neural Networks (ANN) to predict the received power/path loss in indoor links. The prediction approach combines the use of ANNs and ray-tracing in order to identify and parameterize the so-called dominant path. A complete description of the process for creating and training an ANN-based model is presented, in which past values of one or more time series are used to predict future values. Special emphasis is placed on the training process. More specifically, we will be discussing various techniques to arrive at valid predictions focusing on an optimum selection of the training set. A quantitative analysis based on results from narrowband measurement campaigns is also presented.

Introduction and Prediction models

The need for connectivity anywhere, added to the increment in the number of users, has triggered the development of various generations of mobile communication standards in the last decades. Accurate and fast prediction models are needed for making received signal level/path loss predictions prior to actual network deployment. In this paper we analyze the performance achievable with an intermediate technique between purely empirical and purely deterministic, based on the use of Artificial Neural Networks, ANNs.

A great variety of methods has been proposed for predicting the expected received electric field level or, alternatively, the path loss. These calculations can be made using empirical or deterministic models. An intermediate alternative is using artificial neural network-based (ANN) models.

Empirical models are based on measurement campaigns carried out in specific, representative environments. Regression techniques are then used for obtaining mathematical expressions describing the propagation loss as a function of the path length. The computational efficiency of these models is satisfactory, while having a limited accuracy. A typical example is the well known Okumura-Hata model [1].

On the other hand, deterministic models apply accurate electromagnetic techniques or simplified versions of them. These require accurate input information of the propagation environment: buildings, etc. Their main advantage is their precision, despite their lack of computational efficiency. It is quite common to see high frequency approximations of the full wave solutions which make use of ray-tracing techniques for identifying all possible paths between the transmitter and the receiver including multiple reflections, diffractions and transmissions through walls. The contribution of each ray is then calculated by using Fresnel's transmission and reflection coefficients, and GTD/UTD for diffracted contributions.

ANN-base models try to combine the advantages of empirical and deterministic models. In the literature, the most common choice is using feedforward networks, commonly referred to as multilayer perceptrons (MLPs). An alternative is to use the so-called radial basis function networks (RBF) for their fast convergence, robustness and small size [2].

Most implementations for our application use ANNs with two hidden layers. In the first, a number of neurons greater than the number of inputs is usually found. However, other studies show that more complex networks do not necessarily increase the prediction accuracy. Moreover, it has been found
that the generalization properties of ANNs may be reduced, i.e., they may be more sensitive to the
training set data [3].

In the hidden layers, non-linear activation functions are normally used, e.g., sigmoid-type
functions. For the output level, linear functions are normally used. In the hidden layers, also wavelet
functions can be found in received field prediction applications [4]. However, even though they show
faster computation times, in contrast, they require much larger training data sets.

Different algorithms can be used for training an ANN. In [5] their efficiencies were analyzed
showing that the best results are obtained with Bayesian Regularization and Levenberg-Marquardt
techniques, the latter being the most used option. Another algorithm also used offers good
performances is the Resilient Propagation algorithm. ANNs can also be combined with other
techniques for characterizing the effects of RF propagation. When simulation time is critical, the
so-called "dominant path", selected by means of a ray-tracing tool, can be used to provide the
necessary inputs to the AAN. This leads to acceptable results both in terms of time and accuracy. The
dominant path is the propagation path between the transmitter and receiver showing the smallest loss.
Thus, instead of searching for all possible ray combinations, the problem is simplified while an
acceptable generalization performance may be achieved. The dominant path can be calculated using
two main techniques: the recursive neighboring model and the convex corners approach [6].

In last few years, many researchers have applied ANNs for predicting path loss in indoor
environments [7].

Data Collection and Preprocessing

A single transmitter is assumed while various receive locations can be defined as part of a route or a
meshed grid. The route option is very well suited for the training process. A continuous wave, CW,
transmitter was set up at a number of sites while the received power was measured at several points
along a number of routes. Measurements were repeated several times so as to average out the signal
cancellations and enhancements due to multipath. For each measurement point, information on its
coordinates and the received power level in dBm were recorded. All the measurement routes and the
transmit locations are shown in Figures 1. CW measurements were made at the 900 and 1800 MHz
bands, using a vertically polarized 4 dBi gain antenna and 35 dBm transmit power. Receiver was a
spectrum analyzer connected to a PC. Measurements were triggered every 350 cm along the route.
Receive antennas were also vertically polarized, with omnidirectional patterns and 0 dBi gains.

Our ANN model works in combination with a simplified ray-tracing tool. This performs CAD tasks
as well as basic ray-tracing for finding the dominant propagation path for each Tx-Rx pair, then it
calculates this path's parameters. The dominant path can belong to any of four different types: (a)
direct ray paths, when the line-of-sight, LOS, path is not blocked, (b) wall-reflection paths, (c)
corner-diffraction paths and (d) propagation through-obstacle paths.
Selecting the Architecture

Starting from an earlier version of the tool [7], we have implemented a new one using the dominant path approach. Then, this implementation has been trained with measurements. Finally, comparisons between predictions and measurements for data sets different from those used for training were carried out.

ANN is a MLP network with pyramidal structure consisting of three main parts: an input layer with 8 neurons, each associated with one of the 8 selected input parameters, two hidden layers with 6 and 4 neurons, respectively, with sigmoid-type activation functions and, finally, an output layer with a single neuron with a linear function (Figure 2).

Input parameters must characterize the propagation path between transmitter and receiver:
- Screen effect, Po1, Po2. It occurs when there are walls near the transmitter or receiver blocking the direct ray.
- Local reflections, Po3, Po4. They exist when either the receiver or the transmitter are located close to a corner giving rise to multiple reflections.
- Waveguide effect, Po5. It appears in corridors.
- Change of direction, Po6. It occurs when diffraction takes place.
- Transmission loss, Po7. It is introduced when the signal must pass through an obstacle.
- Free space loss, Po8. It depends on the distance and the frequency.

The most critical step when designing an ANN-based model is the training process. The back-propagation technique was selected as learning method, where the predicted power is compared with the actual measurement and the difference (error) is fed back to the network for correcting the various network connection weights. The Levenberg-Marquardt algorithm was used for training the model. This method uses the evolution of the gradient changing the coefficient for each neuron connection in the direction that causes a larger error reduction. The chosen number of training cycles was one thousand. This is a tradeoff between error and time.

![Figure 2. Architecture of the neural network.](image-url)

![Figure 3. Example of prediction result and comparison with measurements.](image-url)
After the ANNs were trained, we analyzed the prediction errors by comparing the results of the ANN-based model and the received power levels measured at points different from those used in the training phase. Figure 3 illustrates a measurement route and the obtained prediction. For each route, the mean error, mean squared error and standard deviation were calculated. In the figure we can observe how the prediction curve is much smoother than that of the measurement. This is because the ANN input parameters, obtained from the ray-tracer, are very similar for neighboring points along the route. The user of such a prediction tool must be aware of this limitation. Still, as observed, the average error and its spread are very small.

Training the Network and Conclusions

A wise selection of real propagation paths from which the neural network will learn how to calculate the received power is the most critical factor in the training phase. Those real situations form the so-called "training set". To optimize the training set several routes with different characteristics must be selected so as to provide the ANN with all the propagation conditions (reflection paths, direct ray paths, etc.) likely to be encountered. In addition, the selected routes have to include received positions showing different ranges of input parameters. In this way, the network will learn to behave in many different situations and will be able to make correct generalizations when applied to new cases. After learning from a number of routes, the network must be tested with other data sets from different routes. Predictions for those test routes must show similar errors to those for the training routes. If this is the case, network will be correctly trained.

The first and essential step in the training process involves a suitable characterization of the measurements points in the training routes according to their dominant path type. The choice of training routes must be a planned process based supplying a sufficient and balanced number of measured points belonging to the various propagation conditions to be expected. Based on the dominant path concept, we have to be careful when training the ANN to provide an appropriate mix of the four path types identified.

A total of 50 measurement routes were recorded, each with a different number of receive positions depending on its length. Hence, the available measurements correspond to a total of 79 routes with 3420 sampling or receive points. As indicated earlier, each route was measured several times and, then, point-wise averages were calculated. The number of transmitter sites was 6 while. Table 1 presents a summary of all measurement locations according to their corresponding path types.

<table>
<thead>
<tr>
<th>Indoor receivers</th>
<th>% of total</th>
<th>Set A</th>
<th>Set B</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOS direct ray</td>
<td>572</td>
<td>16.7</td>
<td>399</td>
</tr>
<tr>
<td>NLOS reflection</td>
<td>-</td>
<td>-</td>
<td>30</td>
</tr>
<tr>
<td>NLOS diffraction</td>
<td>635</td>
<td>18.6</td>
<td>219</td>
</tr>
<tr>
<td>NLOS obstacles</td>
<td>2213</td>
<td>64.7</td>
<td>648 (17.5% of 3420)</td>
</tr>
<tr>
<td>TOTAL</td>
<td>3420</td>
<td>100</td>
<td>Set A</td>
</tr>
</tbody>
</table>

Table 1. Classification of receive locations and Distribution of measurement points in Sets-A and –B.

Two strategies have been analyzed in the selection of the training set. In the first we selected entire routes while the second focused on selecting specific receive points according to the dominant path category to which they belonged.

We now analyze the first, i.e., route-wise strategy. From the available measurements, a subset of the routes was used for training while the rest was used for testing. To illustrate the effect of the number of routes considered in the training process in relation to the achieved prediction accuracy, several training sets were used as discussed below.

To train the indoor network, two different sets were used. Set-A consisted of data from two transmitters and four different routes. In all, 648 measurements were used: 399 points corresponded to direct-ray paths, 219 to through-obstacle paths and 30 to diffraction paths. Set-B consisted of eight routes corresponding to four transmitters. Now, 897 training points were used (26.3% of a total of
The distribution of path types is as follows: 406 were direct ray paths, 101 diffraction paths and 390 through-obstacle paths, Table 1. For the test set four complete routes were used, Table 2.

Table 2. Numerical results of simulations, with two and four transmitters.

<table>
<thead>
<tr>
<th>Test routes</th>
<th>Route 6</th>
<th>Route 7</th>
<th>Route 8</th>
<th>Route 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network trained with Set-A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean error</td>
<td>1.23</td>
<td>2.58</td>
<td>2.19</td>
<td>2.37</td>
</tr>
<tr>
<td>RMS error</td>
<td>1.72</td>
<td>3.46</td>
<td>3.14</td>
<td>3.27</td>
</tr>
<tr>
<td>std</td>
<td>1.20</td>
<td>2.31</td>
<td>2.25</td>
<td>2.25</td>
</tr>
<tr>
<td>Network trained with Set-B</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean error</td>
<td>0.96</td>
<td>1.99</td>
<td>1.36</td>
<td>1.47</td>
</tr>
<tr>
<td>RMS error</td>
<td>1.48</td>
<td>2.84</td>
<td>1.99</td>
<td>2.60</td>
</tr>
<tr>
<td>std</td>
<td>1.13</td>
<td>2.03</td>
<td>1.45</td>
<td>2.15</td>
</tr>
</tbody>
</table>

In the case of Set-B, the errors were much smaller than for the Set-A. With the new training, the same routes were simulated. Due to the path type mix in Set-A, routes with diffraction paths were badly predicted: the network so trained cannot properly simulate those measurement points where the dominating conditions are not sufficiently well represented in the training set. Training Set-B introduces more measurements and also covers a more balanced mix of propagation path types. Thus, the selected routes in Set-B encompass an appropriate assortment of paths from all types.

Now we analyze the second strategy to selecting the training set, i.e., a path-type oriented selection. In this case, the training process was separately carried out for each type of propagation path. Training the ANN with separate receiver locations according to their propagation path types could, in principle, allow achieving a much better prediction accuracy. According to this approach, several routes were split into subsets, as a function of their dominant path, so that all receive points with a direct-ray predominant path were placed into the same subset. Then, some of those points were used to train the ANN and others for testing it.

As shown in Table 3, results show a similar error parameter range. The error parameter range in through-obstacle and diffraction paths is in the order of 2-3 dB, whereas for direct ray paths it again shows a lower value. In any case, the general performance is quite good, it does not seem to be much better than the one achieved in the previous analyses.

Table 3. Errors for the path-type oriented analysis for the indoor case.

<table>
<thead>
<tr>
<th>Test routes</th>
<th>LOS direct ray</th>
<th>NLOS diffraction</th>
<th>NLOS obstacles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean error</td>
<td>1.77</td>
<td>3.26</td>
<td>2.15</td>
</tr>
<tr>
<td>RMS error</td>
<td>2.52</td>
<td>4.74</td>
<td>3.15</td>
</tr>
<tr>
<td>std</td>
<td>1.79</td>
<td>3.44</td>
<td>2.30</td>
</tr>
</tbody>
</table>

References
