Conference on
Information Technology and
Data Science

Inversion of Artificial Neural Networks for WiFi RSSI Propagation Modeling

Bence Bogdándy, Zsolt Tóth

Eszterházy Károly University, Faculty of Informatics, Eger, Hungary
bogdandy.bence@uni-eszterhazy.hu
zsolt.toth@uni-eszterhazy.hu

Proceedings of the 1st Conference on Information Technology and Data Science
Debrecen, Hungary, November 6–8, 2020
published at http://ceur-ws.org

Abstract

Wireless communication via access points has rapidly become widespread in almost all aspects of human life. There is an abundance of Wi-Fi access points in almost every building. Wi-Fi positioning systems take advantage of the widespread use of the access points. Wi-Fi based indoor positioning techniques use Wi-Fi fingerprinting to record the propagated signal of individual access points. Recording the data of the propagation models can be used to build a fingerprinting radio map. The built fingerprinting radio maps consist of a set of coordinates, and an access point radio signal strength indication. Artificial Neural networks have proven to be one of the most useful prediction methods, given a big data set. Inversion of Artificial Neural Network models is the process of creating a model that is capable of predicting a set of possible inputs from a given output. The inversion of the neural network which has been trained on the fingerprinting data set can create a novel positioning method. Received signal strength indication can be inverted into a set of coordinates. This paper includes a description, and evaluation of possible metrics for calculating the error of indoor positioning in an evolutionary artificial neural network inversion system.

Keywords: Indoor navigation, indoor positioning, machine learning, neural network inversion

1. Introduction

Indoor positioning has proven to be a challenge for the last few decades. Despite the popularity and the general use of global positioning systems, indoor positioning is yet to see widespread adoption. Many different theories and technologies were employed in order to create an indoor localization system which is capable of general adoption Wi-Fi positioning systems [11, 12] take advantage of the widespread use of the Wi-Fi access points. Wi-Fi based indoor positioning techniques use fingerprinting to record the propagated signal of individual access points. Recording the data of the propagation models can be used to build a fingerprinting radio map. The built fingerprinting radio maps consist of a set of coordinates, and an access point radio signal strength indication.

Methods of modern data science [3] can be used to process the fingerprinting data set in order to extract, and evaluate information. Furthermore, machine and deep learning [8] can be used to learn and predict radio signal strength indication values based on input coordinates. Artificial Neural networks have proven to be one of the best prediction methods, given a big data set. Inversion of Artificial Neural Network models [4–6, 10] is the process of creating a model that is capable of predicting a set of possible inputs from a given output.

The inversion of the neural network which has been trained on the fingerprinting data set can create a novel positioning method. Received signal strength indication can be inverted into a set of coordinates.

A number of different metrics are required to train the evolutionary algorithm behind the inversion method. Possible metrics include the mean squared error of the coordinates, the error of the predicted RSSI based on the inverted values, and the Jaccard index value of the set of original coordinates, and predicted coordinates. This paper includes a description, and evaluation of possible metrics for calculating the error of indoor positioning in a evolutionary artificial neural network inversion system.

2. Related Works

2.1. Indoor Positioning and Navigation

Indoor Positioning is an actively researched, which aims to find ways to automatically find the coordinates of users in indoor environments. Positioning and Navigation has been a solved issue outdoors for decades. The use of satellite navigation provides a communication which is stable and available in most conditions. Global Positioning Systems [7] have been part of everyday life, but not unlike other technologies of today, it originates from the military. One might assume that since GPS technologies have existed for so long and has enjoyed wide adoption in many industries, indoor positioning must also be a well developed field. Indoor environments provide a challenge as opposed to the outdoors; walls in buildings are built to keep heat inside. However, insulation also provides a dampening effect on electromagnetic radiation. Therefore, satellites cannot be used to accurately communicate with devices indoors.

For this reason, indoor positioning research has not enjoyed the attention like global positioning systems did. Nevertheless, many different systems have been developed for indoor positioning applications over the years. One of the most widely adopted technologies for indoor positioning is the use of WiFi RSSI with fingerprinting [1, 12]. WiFi-based Indoor Positioning and Localization techniques use pre-installed WiFi enabled Access Points. The systems record connections with devices, and measure the radio signal strength indication (RSSI) values. These values can be used to build radio maps. Radio map data can be later transformed into data sets, which provide the basis for indoor positioning applications.

2.2. Data Science

Data science has provided many innovations in the past decade, and seems to be a very important part of computer science and life in general in the future.

Data recorded from WiFi fingerprinting can be parsed and transformed using the tools of data science, such as statistical analysis. The process of data acquisition was performed in the University of Miskolc's Department of Information Science. The data will be discussed in the Methods section of the paper. Once the data has been loaded, and transformed various models can trained. After the training, the trained models can be used for prediction of various target variables.

There are a number of machine learning models which have been used successfully to solve various complex tasks in the past decade. Models have been varied over the years, always changing attention to the most recent when a significant improvement in prediction accuracy has been reached with a particular model. One of the most versatile models has been the artificial neural network. Artificial Neural Networks (ANN) have gained a reputation as one of the most significant machine learning models over the years for solving problems which would otherwise seem unsolvable. Deep learning is the collective term used for neural networks with deep and complex structures.

2.3. ANN Optimization Methods

The goal of artificial neural networks is to use training data to modify the inner connections called neurons of the network. Using the known input, and output combinations of the data set, the neural network is able to iteratively modify its own connections. This mechanism is able to automatically focus on important connections in the network in order to increase prediction efficiency.

The most popular optimization method for this process is the gradient descent algorithm, a *backpropagation*. *Backpropagation* uses the partial derivatives to calculate the importance of certain neurons. The weight of the neurons are propagated back from the end of the neural network. There are optimization algorithms for neural networks other than gradient descent. These algorithms are usually Newton, or quasi-Newton methods.

After the network has been trained, new input data can be fed into the network in order to predict unknown outputs. A section of the data set is usually reserved for testing the model by comparing actual outputs with expected values. The difference between these values is usually a metric of general performance of models.

3. Methods

3.1. Indoor Positioning Method

The proposed indoor positioning system maintains an active connection with all users, while measuring and recording their WiFi RSSI values. The WiFi RSSI values are fed into a novel artificial neural network inversion method in order to determine possible locations of the users. The actual positions can be fine-tuned using tracking of previous positions, and other methods.

3.1.1. Data Set

A dataset was previously collected [9] entirely for future indoor position prediction tasks. The data set was recorded at the University of Miskolc in the Department of Information Science. The data set consists of individual measurements, which were measured using x, y, z coordinates, WiFi RSSI values, and Bluetooth signal measurements among other values. The data set consists of 67 features, and 1540 rows. The dataset consists of raw data, and had to be transformed into individual data sets of certain WiFi Access points.

The X, Y, Z coordinates were used for training, while the RSSI value was chosen as a target for prediction.

3.2. Artificial Neural Network Inversion

After a neural network has been trained and tested, it can be used to calculate predictions based on incoming data. The structure and the weight of the neurons, as well as the inputs are fixed, while the output remains a variable. Therefore, the trained network can be thought of as an approximated black box function f. The structure within the network does not describe the various transformations that result in an output, it only provides a functional approximation.

Inversion is the process of predicting the input values based on a fixed neural network and an output. The problem is that the approximated function of the neural network is only linear from input to output. The f^{-1} inverse function is a non-linear function, as certain inputs (originally the outputs) can be assigned to multiple outputs (originally the inputs). Therefore, no unique input values exist which can be calculated from the outputs. The possible values create manifold surfaces in the *n-dimensional* space, where n denotes the number of inputs. Since f^{-1} cannot be calculated using the original network's weights, another method must be used. This method is usually called *Inversion*. Two different methods of neural network inversion are usually distinguished [4].

On Figure 1 a figure of a two dimensional input space can be seen. Each contour line consists of points, which when passed though the trained artificial neural network, will produce the same output.

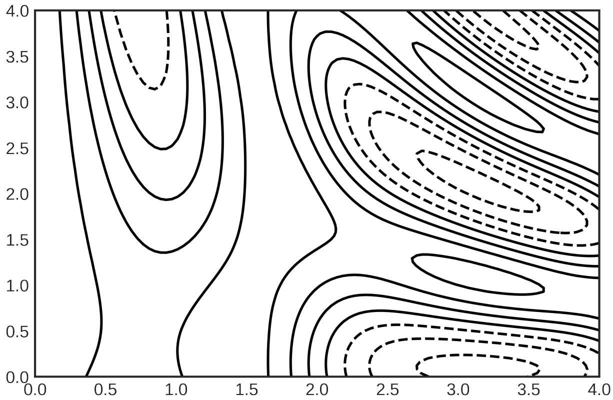

Figure 1. Contour Lines of Possible Input Combinations on a Two Dimensional Surface.

3.2.1. Single-element Search

Single-element search methods are capable of calculating one possible input combination for a given output. Because the value of only one input combination is required, the methods used for neural network training can also be used to train the network to be able to invert itself. However, different structures, and optimization methods might be required for the inverse function.

One implementation of the single-element search methods for artificial neural network inversion is the William-Linden-Kindermann(WLK) Algorithm [5, 6]. The algorithm proposes a structure of different set of neurons for a given trained neural network, which are trained by a modified backpropagation algorithm. The WLK algorithm has seen real-life use, as it was noted by Jensen et. al. The algorithm has been used for sonar performance analysis in submarines in order to determine the position, and direction the submarine is going in order to avoid collision.

3.2.2. Multi-element Search

A different approach to the problem of generating possible input values from a given output comes can be categorized as the multi-element search methods. These methods are usually stochastic methods which are able to generate multiple possible values using iterative fine-tuning to fit the given criteria.

The implementation of such method can be achieved with an evolutionary algorithm. A genetic algorithm was used during the implementation to simultaneously produce, and optimize a number of possible input combinations. The individuals are optimized using the standardized selection, crossover, and mutation genetic algorithm methods. A special repulsion method can be implemented between points in order to try to evenly distribute individual points along the input manifold of the space.

3.3. Error Metrics

The error values of the generated input combinations, and the original input values have to be calculated in order to validate the inversion algorithm.

3.3.1. Distance Between Points

One way to calculate error is taking individual points in the predicted input combinations, and measuring the distance to the closest input point from the original data set. Of course the comparison can only be drawn to input combinations with the same output as the inversion input. As there are many different input combinations for every output, this metric might not provide general error measurement.

In order to better see the difference between the generated points and the original inputs is to calculate a weighted central point for both. The distance between these weighted central points can be measured in order to calculate a more general error between elements.

3.3.2. Jaccard Index

A third proposed error measurement for the input values is to calculate the Jaccard Index of both value sets. The Jaccard Index calculates the similarity, and distance between two sets of coordinates. The Jaccard index is measured by $\frac{A \cap B}{A \cup B}$, where A and B are finite sets. The Jaccard Index is frequently used in image recognition tasks, where it is used to measure the similarity between the position of an object, and a predicted position. Similarly, the paper proposes that the jaccard index can be used to calculate similarity between a set of input values, and predicted set of input values. On Figure 2, a visual representation of the jaccard index can be seen.

4. Implementation

4.1. Artificial Neural Network Inversion

The implementation of the inversion is in the *Python* programming language. Python is already the language of choice for data science and machine learning, therefore it was a natural choice for the implementation.

The implementation uses the Sci-kit Learn package, which contains a number of modern data science and machine learning models. The *MLPRegressor* multi layer perceptron model. The MLPRegressor is an easy-to-use implementation of the artificial neural network. The network can be fine-tuned using the multitude of

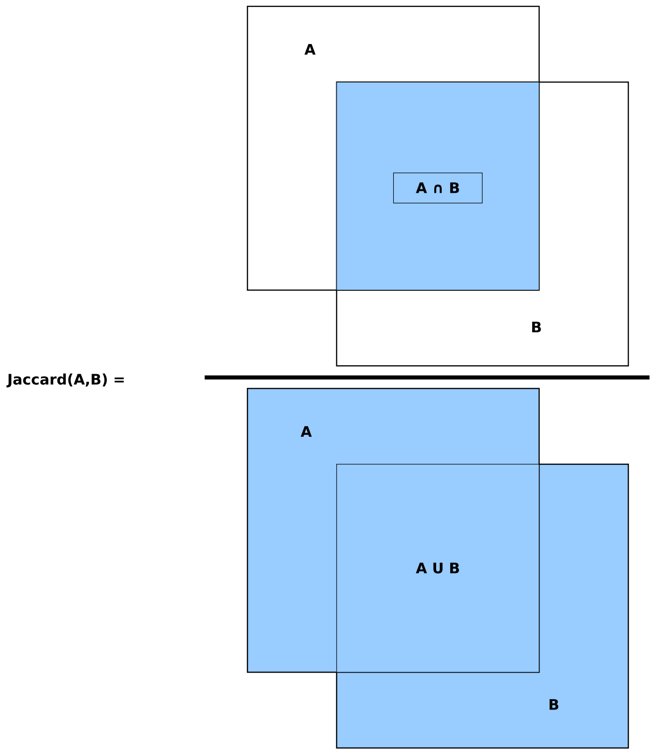

Figure 2. Jaccard index.

optional parameters which can be passed during the initialization of the network. The neural network has been fine-tuned using hyperparameter tuning [2].

The chosen network is capable of predicting the output RSSI values based on the coordinates, with the performance of 75% with a variance of 15%. The high variance can be attributed to the usage of a single network structure for every WiFi RSSI Access Point. Running a hyperparamter tuning algorithm for every access point would undoubtedly yield better, unique network structures.

4.2. Evolutionary Inversion Implementation

A general genetic algorithm-based inversion method was implemented. The inversion method is capable of the inversion of a single *MLPRegressor*. The inputs of the inverter consists of the regressor, and optional genetic algoritm parameters for fine-tuning. These parameters include the numberic bounds in which the parameters are generated, population size, number of elites, and strategies for crossover, and selection. On Figure 3, a flowchart of the genetic algorithm can be seen.

The *invert* method returns the inverted population of the given regressor. The python code of the invert method can be inspected on Code Listing 1.

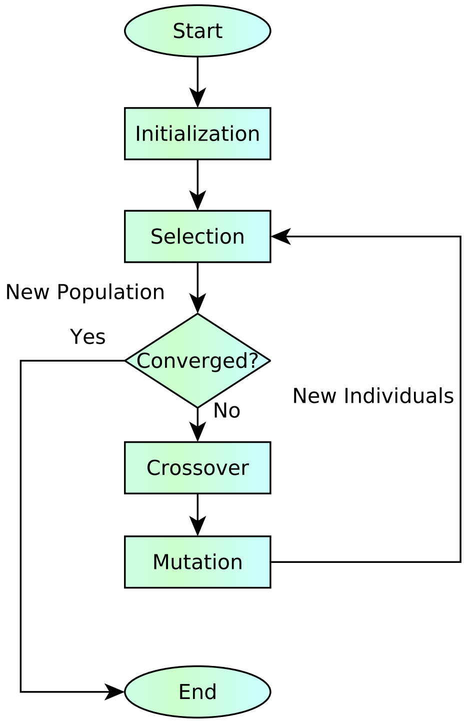

Figure 3. Flowchart of the Genetic Algorithm.

```
def invert(self,
           desired_output: np.ndarray) -> List[np.ndarray]:
    self.logger.info("GAMLPInverter.invert started")
    population = self._init_ga_population()
    for _ in range(self.max_generations):
        fitness_values = [self.__fitness(individual, desired_output)
                          for individual in population]
        sorted_fitnesses, sorted_offsprings = self.__sort_by_fitness(fitness_values, population)
        elites = sorted_offsprings[0:self.elite_count]
        crossed_mutated_offsprings = []
        for _ in range(self.population_size - self.elite_count):
            parents = self.__selection(sorted_fitnesses, sorted_offsprings)
            crossed_mutated_offsprings.append(self.__mutate(
                self.__crossover(parents[0], parents[1])))
        population = [*elites, *crossed_mutated_offsprings]
    fitness_values, population = self.__sort_by_fitness(fitness_values, population)
    self.logger.debug("population: ", population)
    self.logger.info("GAMLPInverter.invert stopped")
    return population
```

Listing 1.

The method consists of standardized mechanisms of genetic algorithms. Implementation contains different implementations of selection, and crossover methods. Selection methods include random, rank, tournament and roulette selection methods.

Crossover methods include one point, multi point, uniform, and arithmetic crossover methods. These methods can be changed by passing a string to their respective parameters.

Individual creation is implemented using numpy's *np.uniform* method which randomly draws samples from a uniform distribution within the passed parameters.

5. Conclusion

In this paper, a general outline of an artificial neural network inversion based indoor positioning was presented. Inversion as an artificial neural network operation was shown. Inversion could become the third natural operation of any neural network. Artificial neural network implementations already contain the training, and predict methods. Training freezes the input and output values in order to modify the underlying structure of the network in order to increase performance. Prediction on the other hand freezes the input, and weights of the network in order to predict new values. Inversion has not seen widespread adoption as a possible third operation, which freezes the weights and the output, to predict the possible inputs. This method could be used to increase the explainability, and performance of neural networks by exploring the different input values of a given output.

The two different categories of neural network inversion methods were outlined. Single search methods use well known algorithms used in neural networks already. However, these methods are only capable of producing one input combination at a time. Multi element search methods use stochastic search methods, particularly evolutionary algorithms. These methods provide a less stable search, but they are capable of predicting all possible input values given a well defined evolutionary algorithm.

A general implementation of a genetic algorithm was described in the python programming language.

References

[1] M. Abbas, M. Elhamshary, H. Rizk, M. Torki, M. Youssef: *WiDeep: WiFi-based accurate and robust indoor localization system using deep learning*, in: 2019 IEEE International Conference on Pervasive Computing and Communications (PerCom, IEEE, 2019, pp. 1–10.

[2] B. Bogdándy, Zs. Tóth: *Analysis of Training Parameters of Feed Forward Neural Networks for WiFi RSSI Modeling*, in: 2019 IEEE 15th International Scientific Conference on Informatics, IEEE, 2019, pp. 000273–000278.

[3] J. Han, J. Pei, M. Kamber: *Data Mining: Concepts and Techniques*, The Morgan Kaufmann Series in Data Management Systems, Elsevier Science, 2011, isbn: 9780123814807, url: `https://books.google.hu/books?id=pQws07tdpjoC`.

[4] C. A. Jensen, R. D. Reed, R. J. Marks, M. A. El-Sharkawi, J.-B. Jung, R. T. Miyamoto, G. M. Anderson, C. J. Eggen: *Inversion of feedforward neural networks: Algorithms and applications*, Proceedings of the IEEE 87.9 (1999), pp. 1536–1549.

[5] J. Kindermann, A. Linden: *Inversion of neural networks by gradient descent*, Parallel Computing 14.3 (1990), pp. 277–286, issn: 0167-8191,
doi: https://doi.org/10.1016/0167-8191(90)90081-J,
url: http://www.sciencedirect.com/science/article/pii/016781919090081J.

[6] A. Linden, J. Kindermann: *Inversion of multilayer nets*, in: Proc. Int. Joint Conf. Neural Networks, vol. 2, 1989, pp. 425–430.

[7] Y. Masumoto: *Global positioning system*, US Patent 5,210,540, May 1993.

[8] T. Mitchell: *Machine Learning*, McGraw-Hill International Editions, McGraw-Hill, 1997, isbn: 9780071154673,
url: https://books.google.hu/books?id=EoYBngEACAAJ.

[9] Zs. Tóth, J. Tamás: *Miskolc IIS Hybrid IPS: Dataset for Hybrid Indoor Positioning*, in: 26st International Conference on Radioelektronika, IEEE, 2016, pp. 408–412.

[10] R. J. Williams: *Inverting a connectionist network mapping by backpropagation of error*, in: 8th Annual Conf. Cognitive Sci. Soc. 1986.

[11] C. Yang, H.-R. Shao: *WiFi-based indoor positioning*, IEEE Communications Magazine 53.3 (2015), pp. 150–157.

[12] M. Youssef, A. Agrawala: *The Horus WLAN location determination system*, in: Proceedings of the 3rd international conference on Mobile systems, applications, and services, 2005, pp. 205–218.