Peer-reviewed version available at *Electronics* **2019**, *8*, 59; doi:10.3390/electronics8010059

Article

A Mobile Positioning Method Based on Deep Learning Techniques

Ling Wu [1,2], Chi-Hua Chen [2,*] and Qishan Zhang [1]

[1] School of Economics and Management, Fuzhou University, Fuzhou 350116, China; wuling1985@fzu.edu.cn; zhangqs@fzu.edu.cn
[2] College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350116, China; wuling1985@fzu.edu.cn; chihua0826@ fzu.edu.cn
* Correspondence: chihua0826@fzu.edu.cn; Tel.: +86-13859183858

Abstract: This study proposes a mobile positioning method which adopts recurrent neural network algorithms to analyze the received signal strength indications from heterogeneous networks (e.g., cellular networks and Wi-Fi networks) for estimating the locations of mobile stations. The recurrent neural networks with multiple consecutive timestamps can be applied to extract the features of time series data for the improvement of location estimation. In practical experimental environments, there are 4,525 records, 59 different base stations, and 582 different Wi-Fi access points detected in Fuzhou University in China. The lower location errors can be obtained by the recurrent neural networks with multiple consecutive timestamps (e.g., 2 timestamps and 3 timestamps); the experimental results can be observed that the average error of location estimation was 9.19 meters by the proposed mobile positioning method with 2 timestamps.

Keywords: deep learning; recurrent neural networks; mobile positioning method; fingerprinting positioning method; received signal strength

1. Introduction

With the development of wireless networks and mobile networks, the techniques of location-based services (LBS) can provide the corresponding services to the users according to users' current locations. LBS which have played an important role in many fields require the high accuracy of positioning technology [1-23].

For the LBS in outdoor environments, global positioning system (GPS) and assisted GPS (A-GPS) are popular techniques and meet most of the positioning requirements. However, these techniques may be no longer applicable if the problems of multi-path propagation of wireless signals exist [20]. Furthermore, higher power consumptions are required by these techniques [1]. Therefore, some studies proposed cellular-based positioning methods to analyze the signals of cellular networks for location estimation [1, 6, 8, 11, 13, 14]. Although cellular-based positioning methods can estimate the locations of mobile stations without GPS modules, big errors of estimated locations may be obtained.

For the LBS in indoor environments, Wi-Fi-based positioning methods are popular techniques to detect and analyze the received signal strength indications (RSSIs) from Wi-Fi access points (APs) [7, 12, 14, 18-22]. The fingerprinting positioning methods based on machine learning algorithms were proposed to learning the relationships among locations and RSSIs for the estimation of locations. Although these methods can estimate the locations of mobile stations without GPS modules, big errors of estimated locations may be obtained. Although higher precise estimated locations can be obtained by Wi-Fi-based positioning methods, these methods may be invalid in outdoor environments if the transmission coverage of Wi-Fi APs is not enough.

Some deep learning methods (e.g., neural networks, convolutional neural networks, recurrent neural networks, etc.) have been applied to improve the accuracies of estimation locations [12, 18, 19, 20, 22]. For instance, a modified probability neural network was used for indoor positioning, and the

Peer-reviewed version available at Electronics 2019, 8, 59; doi:10.3390/electronics8010059

accuracies of estimated locations by the method were higher than triangulation technique [18]. An improved neural network was trained with the correlation of the initial parameters to achieve the highest possible accuracy of the Wi-Fi-based positioning method in indoor environments [12].

Although cellular-based positioning methods can obtain estimated locations in outdoor environments, the errors of estimated locations may be larger. Furthermore, Wi-Fi-based positioning methods can obtain higher precise locations, but these methods may be not applicable in outdoor environments. Therefore, this study proposed a mobile positioning method to analyze the network signals from heterogeneous networks (e.g., cellular networks and Wi-Fi networks) for the LBS in outdoor environments. Furthermore, the recurrent neural networks [24] are applied into the proposed mobile positioning method for the analyses of consecutive locations and network signals (i.e., time series data).

The remainder of the paper is organized as follows. Section 2 provides the overview of mobile positioning methods and fingerprinting positioning methods. Section 3 presents the proposed mobile positioning system and method based on recurrent neural networks. The practical experimental results and discussions are illustrated in Section 4. Finally, conclusions and future work are given in Section 5.

2. Related Work

Mobile positioning methods and fingerprinting positioning methods includes two stages: training stage and performing stage (shown in Figure 1). In training stage, the RSSIs and locations measured by the mobile stations are matched and stored into a fingerprinting database for training. Machine learning methods can be performed to learn the relationships among RSSIs and locations for the establishments of mobile positioning models. In performing stage, mobile stations can detect the RSSIs of neighbor base stations and Wi-Fi APs which can be adopted into the trained models to estimate the locations of these mobile stations.

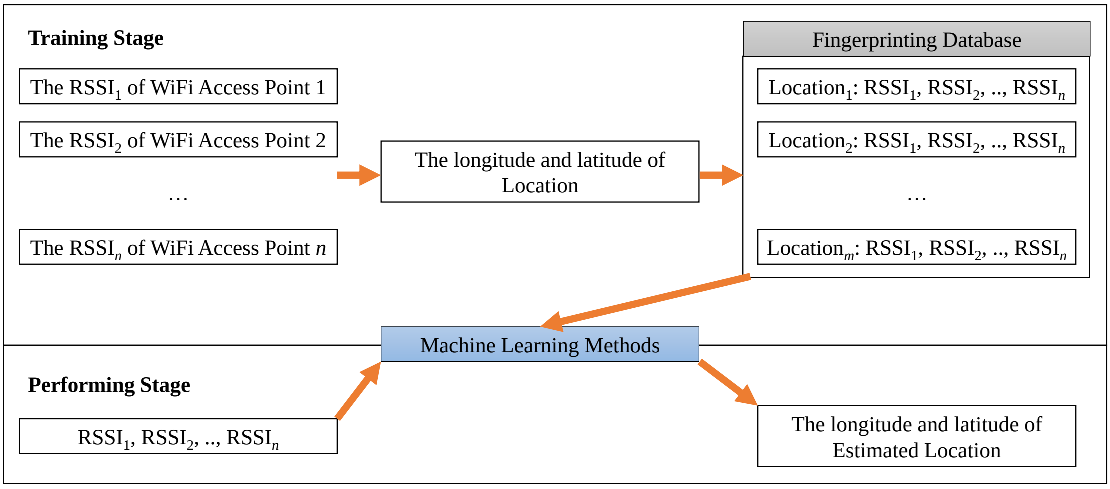


Figure 1. Fingerprinting positioning method

For training the mobile positioning models, some studies used k–nearest neighbors, Bayesian theory, support vector machine, neural networks, convolutional neural networks, or recurrent neural networks to estimate locations in accordance with RSSIs. For instance, a probabilistic positioning algorithm was proposed to store the probability distribution of RSSIs during a certain time in the fingerprinting database, and the probable locations of mobile stations were calculated by a Bayesian theory system [14]. However, the relationships among inputs were assumed as independent parameters, so big errors of estimated locations may be obtained if the inputs were not independent parameters. Some mobile positioning methods based on k–nearest neighbor algorithms can obtain higher accuracies of estimated locations, but these methods required more computation time in performing stage. Some neural networks have been proposed to analyze the interrelated influences of inputs for the improvement of location estimation [12, 18-20], and convolutional neural networks

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2018 doi:10.20944/preprints201810.0239.v1

Peer-reviewed version available at Electronics 2019, 8, 59; doi:10.3390/electronics8010059

were applied to extract the features of spatio metrics [21]. Although the spatio metrics may be analyzed by neural networks and convolutional neural networks, these methods cannot provide the solutions of temporal data analyses. Therefore, this study applies recurrent neural networks to analyze the temporal data for improving the accuracies of estimation locations.

3. Mobile Positioning System and Method

The architecture of the proposed mobile positioning system is presented in Subsection 3.1, and the concepts of the proposed mobile positioning method are illustrated in Subsection 3.2.

3.1. Mobile Positioning System

The proposed mobile positioning system includes (1) mobile stations, (2) a mobile positioning server, (3) a database server, and (4) a model server (shown in Figure 2). Each component in the proposed system is presented in the following subsections.

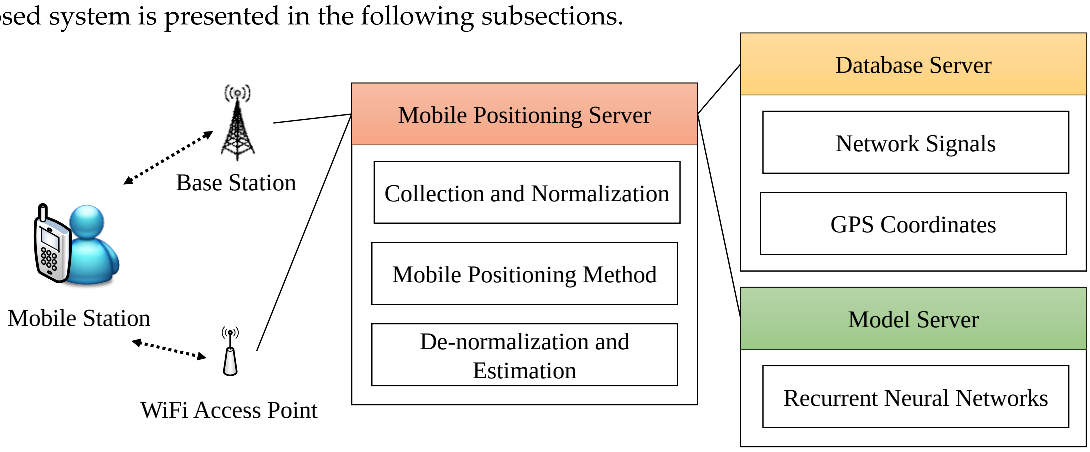


Figure 2. The proposed mobile positioning system

3.1.1. Mobile Stations

In training stage, mobile stations can detect and receive the RSSIs of neighbor base stations and Wi-Fi APs from heterogeneous networks. GPS modules can be equipped into the mobile stations and estimate the locations of mobile stations (i.e., coordinates). Then the mobile stations can send the vectors of GPS coordinates (i.e., longitudes and latitudes) and RSSIs to the mobile positioning server for the collection of network signals. In performing stage, mobile stations can send the detected RSSIs of neighbor base stations and Wi-Fi APs to the mobile positioning server for location estimation.

3.1.2. Mobile Positioning Server

In training stage, the mobile positioning server can receive GPS coordinates and network signals (i.e., the RSSIs of base stations and Wi-Fi APs) from mobile stations. These GPS coordinates and network signals can be sent to the database server for storing. The mobile positioning server can execute the proposed mobile positioning method to train RNN models. The network signals can be used as the input layer of the RNN models, and the GPS coordinates can be used as the output layer of the RNN models. Once the RNN models have been trained, these models can be sent to the model server for saving. In performing stage, the mobile positioning server can load the trained RNN models from the model server. When the mobile positioning server receives network signals from mobile stations, these network signals can be adopted into the trained RNN models for estimating the locations of mobile stations.

3.1.3. Database Server

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2018 doi:10.20944/preprints201810.0239.v1

Peer-reviewed version available at Electronics 2019, 8, 59; doi:10.3390/electronics8010059

The database server can store the vectors of coordinates (i.e., longitudes and latitudes) and RSSIs from mobile stations via the mobile positioning server. These vectors can be queried and used to train RNN models.

3.1.4. Model Server

The model server can save the trained RNN models from the mobile positioning server in training stage, and the saved RNN models can be loaded for location estimation by the mobile positioning server.

3.2. Mobile Positioning Method

The proposed mobile positioning method includes (1) collection and normalization, (2) the execution of mobile positioning method based on recurrent neural networks, (3) de-normalization and estimation. Each step in the proposed method is presented in the following subsections.

3.2.1. Collection and Normalization

For the collection of network signals and GPS coordinates, the RSSIs of base stations from cellular networks (i.e., $R_{c,i}$ in Equation (1)), the RSSIs of Wi-Fi APs from Wi-Fi networks (i.e., $R_{w,i}$ in Equation (2)), and the GPS coordinates (i.e., l_i in Equation (3)) can be detected and collected by the mobile station at time t_i (shown in Figure 3). The RSSI of the *j*-th base station from a cellular network at time t_i is defined as $r_{c,j,i}$, and the RSSI of the *k*-th Wi-Fi AP from a Wi-Fi network at time t_i is defined as $r_{w,k,i}$. The RSSI dataset of heterogeneous networks (i.e., cellular networks and Wi-Fi networks) at time t_i is defined as R_i (shown in Equation (4)). Furthermore, the location l_i (i.e., a GPS coordinate) includes a longitude $l_{x,i}$ and a latitude $l_{y,i}$. There are m locations, n_1 different base stations, and n_2 different Wi-Fi APs detected in the experiments. If the RSSIs of base stations or Wi-Fi APs cannot be detected, the values of these RSSIs can be encoded as null. For instance, the mobile station cannot detect the RSSI of Wi-Fi AP2 at time t_i in Figure 3, so the value of $r_{w,2,i}$ is encoded as null.

$$R_{c,i} = \left\{ r_{c,1,i}, r_{c,2,i}, \ldots, r_{c,n_1,i} \right\} \tag{1}$$

$$R_{w,i} = \left\{ r_{w,1,i}, r_{w,2,i}, \ldots, r_{w,n_2,i} \right\} \tag{2}$$

$$l_i = \left\{ l_{x,i}, l_{y,i} \right\} \tag{3}$$

$$\begin{aligned} R_i &= \left\{ R_{c,i}, R_{w,i} \right\} \\ &= \left\{ r_{c,1,i}, r_{c,2,i}, \ldots, r_{c,n_1,i}, r_{w,1,i}, r_{w,2,i}, \ldots, r_{w,n_2,i} \right\} \end{aligned} \tag{4}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2018 doi:10.20944/preprints201810.0239.v1

Peer-reviewed version available at Electronics 2019, 8, 59; doi:10.3390/electronics8010059

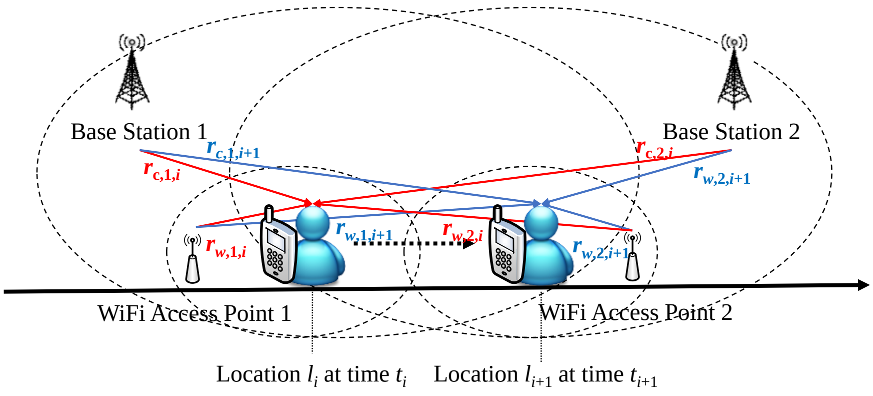


Figure 3. The scenario of network signal and GPS coordinate collection

For the normalization of network signals and GPS coordinates, the minimum values and maximum values of RSSIs and coordinates are considered and adopted into Equations (5), (6), (7), and (8). The normalized RSSI of the *j*-th base station from a cellular network at time t_i is defined as $c_{j,i}$ in accordance with the minimum value and maximum value of the RSSIs (i.e., $r_c^{(-)}$ and $r_c^{(+)}$ in Equation (5)) from cellular networks; the normalized RSSI of the *k*-th Wi-Fi APs from a cellular network at time t_i is defined as $w_{k,i}$ in accordance with the minimum value and maximum value of the RSSIs (i.e., $r_w^{(-)}$ and $r_w^{(+)}$ in Equation (6)) from Wi-Fi networks. Furthermore, the normalized longitude at time t_i is defined as x_i in accordance with the minimum value and maximum value of longitudes (i.e., $l_x^{(-)}$ and $l_x^{(+)}$ in Equation (7)) from GPS coordinates, and the normalized latitude at time t_i is defined as y_i in accordance with the minimum value and maximum value of latitudes (i.e., $l_y^{(-)}$ and $l_y^{(+)}$ in Equation (8)) from GPS coordinates.

$$c_{j,i} = \begin{cases} \dfrac{r_{c,j,i} - r_c^{(-)}}{r_c^{(+)} - r_c^{(-)}}, \text{if } r_{c,j,i} \neq null \\ 0, \text{otherwise} \end{cases}, \text{where } r_c^{(+)} = \max_{1 \le p \le n_1, 1 \le q \le m} r_{c,p,q},\ r_c^{(-)} = \min_{1 \le p \le n_1, 1 \le q \le m} r_{c,p,q} \tag{5}$$

$$w_{k,i} = \begin{cases} \dfrac{r_{w,k,i} - r_w^{(-)}}{r_w^{(+)} - r_w^{(-)}}, \text{if } r_{w,k,i} \neq null \\ 0, \text{otherwise} \end{cases}, \text{where } r_w^{(+)} = \max_{1 \le p \le n_2, 1 \le q \le m} r_{w,p,q},\ r_w^{(-)} = \min_{1 \le p \le n_2, 1 \le q \le m} r_{w,p,q} \tag{6}$$

$$x_i = \begin{cases} \dfrac{l_{x,i} - l_x^{(-)}}{l_x^{(+)} - l_x^{(-)}}, \text{if } l_{x,i} \neq null \\ 0, \text{otherwise} \end{cases}, \text{where } l_x^{(+)} = \max_{1 \le q \le m} l_{x,q},\ l_x^{(-)} = \min_{1 \le q \le m} l_{x,q} \tag{7}$$

$$y_i = \begin{cases} \dfrac{l_{y,i} - l_y^{(-)}}{l_y^{(+)} - l_y^{(-)}}, \text{if } l_{y,i} \neq null \\ 0, \text{otherwise} \end{cases}, \text{where } l_y^{(+)} = \max_{1 \le q \le m} l_{y,q},\ l_y^{(-)} = \min_{1 \le q \le m} l_{y,q} \tag{8}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2018 doi:10.20944/preprints201810.0239.v1

Peer-reviewed version available at Electronics 2019, 8, 59; doi:10.3390/electronics8010059

3.2.2. Mobile Positioning Method Based on Recurrent Neural Network

The proposed mobile positioning method adopts recurrent neural network algorithms to estimate the locations of mobile stations. The recurrent neural networks can be applied to extract the features of time series data, so this study considers and analyzes the normalized RSSIs with multiple consecutive timestamps. Subsection 3.2.2.1 presents recurrent neural networks with one timestamp, and Subsection 3.2.2.2 describes recurrent neural networks with multiple consecutive timestamps.

3.2.2.1. Recurrent Neural Networks with One Timestamp

This subsection shows the designs and optimization of recurrent neural networks with one timestamp. A simple case study of a recurrent neural network with one timestamp is illustrated in Figure 4. The recurrent neural network is constructed with an input layer, a recurrent hidden layer, and an output layer. The input layer includes the normalized RSSIs of two base stations and two Wi-Fi APs (i.e., $c_{1,i}$, $c_{2,i}$, $w_{1,i}$, and $w_{2,i}$), and the output layer includes the estimated normalized longitude and latitude (i.e., $\tilde{x}_i$ and $\tilde{y}_i$). The recurrent hidden layer includes a neuron, and the initial value of the neuron in the recurrent hidden layer is defined as h_0. The value of the neuron in the recurrent hidden layer can be updated as h_1 after calculating the RSSIs in the first timestamp. The weights of $c_{1,i}$, $c_{2,i}$, $w_{1,i}$, $w_{2,i}$, and h_0 are α_1, α_2, β_1, β_2, and v; the weights of h_1 for the outputs $\tilde{x}_i$ and $\tilde{y}_i$ are γ_1 and γ_2, respectively. The biases of neurons in the hidden layer and the output layer are defined as $b_{1,1}$, $b_{2,1}$, and $b_{3,1}$. The sigmoid function is elected as the activation function of each neuron, so the values of h_0, h_1, $\tilde{x}_i$, and $\tilde{y}_i$ can be calculated by Equations (9), (10), (11), and (12). Furthermore, the loss function is defined as Equation (13) in accordance with squared errors.

$$h_0 = 0 \tag{9}$$

$$h_1 = s\left(\sum_{j=1}^{2}\alpha_j \times c_{j,i} + \sum_{k=1}^{2}\beta_k \times w_{k,i} + v \times h_0 + b_{1,1}\right) = s\left(z_{1,1}\right), \text{where } s\left(z\right) = \frac{1}{1+e^{-z}} \tag{10}$$

$$\tilde{x}_i = s\left(\gamma_1 \times h_1 + b_{2,1}\right) = s\left(z_{2,1}\right), \text{where } s\left(z\right) = \frac{1}{1+e^{-z}} \tag{11}$$

$$\tilde{y}_i = s\left(\gamma_2 \times h_1 + b_{3,1}\right) = s\left(z_{3,1}\right), \text{where } s\left(z\right) = \frac{1}{1+e^{-z}} \tag{12}$$

$$E = \frac{1}{2}\left(\tilde{x}_i - x_i\right)^2 + \frac{1}{2}\left(\tilde{y}_i - y_i\right)^2 = \frac{1}{2}\sigma_1^{\,2} + \frac{1}{2}\sigma_2^{\,2} \tag{13}$$

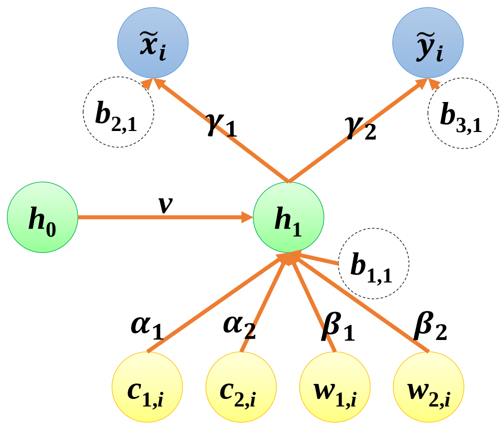


Figure 4. A recurrent neural network with one timestamp

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2018 doi:10.20944/preprints201810.0239.v1

Peer-reviewed version available at Electronics 2019, 8, 59; doi:10.3390/electronics8010059

For the optimization of recurrent neural network, the learning rate η and a gradient descent method is applied to update each weight and bias. The updates of γ_1, γ_2, $b_{2,1}$, $b_{3,1}$, α_1, α_2, β_1, β_2, v, and $b_{1,1}$ are proved and calculated by Equations (14), (15), (16), (17), (18), (19), (20), (21), (22), and (23), respectively.

$$\gamma_1 = \gamma_1 - \eta \times \frac{\partial E}{\partial \gamma_1}, \text{ where}$$

$$\frac{\partial E}{\partial \gamma_1} = \frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_i}\frac{\partial \tilde{x}_i}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial \gamma_1} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_i}\frac{\partial \tilde{y}_i}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial \gamma_1} = \sigma_1 \times \tilde{x}_i \times \left[1 - \tilde{x}_i\right] \times h_1 \tag{14}$$

$$\gamma_2 = \gamma_2 - \eta \times \frac{\partial E}{\partial \gamma_2}, \text{ where}$$

$$\frac{\partial E}{\partial \gamma_2} = \frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_{1,}}{\partial \tilde{x}_i}\frac{\partial \tilde{x}_i}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial \gamma_2} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_i}\frac{\partial \tilde{y}_i}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial \gamma_2} = \sigma_2 \times \tilde{y}_i \times \left[1 - \tilde{y}_i\right] \times h_1 \tag{15}$$

$$b_{2,1} = b_{2,1} - \eta \times \frac{\partial E}{\partial b_{2,1}}, \text{ where}$$

$$\frac{\partial E}{\partial b_{2,1}} = \frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_i}\frac{\partial \tilde{x}_i}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial b_{2,1}} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_i}\frac{\partial \tilde{y}_i}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial b_{2,1}} = \sigma_1 \times \tilde{x}_i \times \left[1 - \tilde{x}_i\right] \tag{16}$$

$$b_{3,1} = b_{3,1} - \eta \times \frac{\partial E}{\partial b_{3,1}}, \text{ where}$$

$$\frac{\partial E}{\partial b_{3,1}} = \frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_i}\frac{\partial \tilde{x}_i}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial b_{3,1}} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_i}\frac{\partial \tilde{y}_i}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial b_{3,1}} = \sigma_2 \times \tilde{y}_i \times \left[1 - \tilde{y}_i\right] \tag{17}$$

$$\alpha_1 = \alpha_1 - \eta \times \frac{\partial E}{\partial \alpha_1}, \text{ where}$$

$$\begin{aligned} \frac{\partial E}{\partial \alpha_1} &= \frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_i}\frac{\partial \tilde{x}_i}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial h_1}\frac{\partial h_1}{\partial z_{1,1}}\frac{\partial z_{1,1}}{\partial \alpha_1} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_i}\frac{\partial \tilde{y}_i}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial h_1}\frac{\partial h_1}{\partial z_{1,1}}\frac{\partial z_{1,1}}{\partial \alpha_1} \\ &= \left(\frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_i}\frac{\partial \tilde{x}_i}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial h_1} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_i}\frac{\partial \tilde{y}_i}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial h_1}\right)\frac{\partial h_1}{\partial z_{1,1}}\frac{\partial z_{1,1}}{\partial \alpha_1} \\ &= \left\{\sigma_1 \times \tilde{x}_i \times \left[1 - \tilde{x}_i\right] \times \gamma_1 + \sigma_2 \times \tilde{y}_i \times \left[1 - \tilde{y}_i\right] \times \gamma_2\right\} \times h_1 \times \left[1 - h_1\right] \times c_{1,i} \end{aligned} \tag{18}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2018 doi:10.20944/preprints201810.0239.v1

Peer-reviewed version available at Electronics 2019, 8, 59; doi:10.3390/electronics8010059

$$\alpha_2 = \alpha_2 - \eta \times \frac{\partial E}{\partial \alpha_2}, \text{ where}$$

$$\begin{aligned}
\frac{\partial E}{\partial \alpha_2} &= \frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_i}\frac{\partial \tilde{x}_i}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial h_1}\frac{\partial h_1}{\partial z_{1,1}}\frac{\partial z_{1,1}}{\partial \alpha_2} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_i}\frac{\partial \tilde{y}_i}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial h_1}\frac{\partial h_1}{\partial z_{1,1}}\frac{\partial z_{1,1}}{\partial \alpha_2} \\
&= \left(\frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_i}\frac{\partial \tilde{x}_i}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial h_1} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_i}\frac{\partial \tilde{y}_i}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial h_1}\right)\frac{\partial h_1}{\partial z_{1,1}}\frac{\partial z_{1,1}}{\partial \alpha_2} \\
&= \left\{\sigma_1 \times \tilde{x}_i \times [1-\tilde{x}_i] \times \gamma_1 + \sigma_2 \times \tilde{y}_i \times [1-\tilde{y}_i] \times \gamma_2\right\} \times h_1 \times [1-h_1] \times c_{2,i}
\end{aligned} \tag{19}$$

$$\beta_1 = \beta_1 - \eta \times \frac{\partial E}{\partial \beta_1}, \text{ where}$$

$$\begin{aligned}
\frac{\partial E}{\partial \beta_1} &= \frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_i}\frac{\partial \tilde{x}_i}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial h_1}\frac{\partial h_1}{\partial z_{1,1}}\frac{\partial z_{1,1}}{\partial \beta_1} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_i}\frac{\partial \tilde{y}_i}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial h_1}\frac{\partial h_1}{\partial z_{1,1}}\frac{\partial z_{1,1}}{\partial \beta_1} \\
&= \left(\frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_i}\frac{\partial \tilde{x}_i}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial h_1} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_i}\frac{\partial \tilde{y}_i}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial h_1}\right)\frac{\partial h_1}{\partial z_{1,1}}\frac{\partial z_{1,1}}{\partial \beta_1} \\
&= \left\{\sigma_1 \times \tilde{x}_i \times [1-\tilde{x}_i] \times \gamma_1 + \sigma_2 \times \tilde{y}_i \times [1-\tilde{y}_i] \times \gamma_2\right\} \times h_1 \times [1-h_1] \times w_{1,i}
\end{aligned} \tag{20}$$

$$\beta_2 = \beta_2 - \eta \times \frac{\partial E}{\partial \beta_2}, \text{ where}$$

$$\begin{aligned}
\frac{\partial E}{\partial \beta_2} &= \frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_i}\frac{\partial \tilde{x}_i}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial h_1}\frac{\partial h_1}{\partial z_{1,1}}\frac{\partial z_{1,1}}{\partial \alpha_2} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_i}\frac{\partial \tilde{y}_i}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial h_1}\frac{\partial h_1}{\partial z_{1,1}}\frac{\partial z_{1,1}}{\partial \beta_2} \\
&= \left(\frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_i}\frac{\partial \tilde{x}_i}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial h_1} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_i}\frac{\partial \tilde{y}_i}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial h_1}\right)\frac{\partial h_1}{\partial z_{1,1}}\frac{\partial z_{1,1}}{\partial \beta_2} \\
&= \left\{\sigma_1 \times \tilde{x}_i \times [1-\tilde{x}_i] \times \gamma_1 + \sigma_2 \times \tilde{y}_i \times [1-\tilde{y}_i] \times \gamma_2\right\} \times h_1 \times [1-h_1] \times w_{2,i}
\end{aligned} \tag{21}$$

$$v = v - \eta \times \frac{\partial E}{\partial v}, \text{ where}$$

$$\begin{aligned}
\frac{\partial E}{\partial v} &= \frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_i}\frac{\partial \tilde{x}_i}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial h_1}\frac{\partial h_1}{\partial z_{1,1}}\frac{\partial z_{1,1}}{\partial v} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_i}\frac{\partial \tilde{y}_i}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial h_1}\frac{\partial h_1}{\partial z_{1,1}}\frac{\partial z_{1,1}}{\partial v} \\
&= \left(\frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_i}\frac{\partial \tilde{x}_i}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial h_1} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_i}\frac{\partial \tilde{y}_i}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial h_1}\right)\frac{\partial h_1}{\partial z_{1,1}}\frac{\partial z_{1,1}}{\partial v} \\
&= \left\{\sigma_1 \times \tilde{x}_i \times [1-\tilde{x}_i] \times \gamma_1 + \sigma_2 \times \tilde{y}_i \times [1-\tilde{y}_i] \times \gamma_2\right\} \times h_1 \times [1-h_1] \times h_0
\end{aligned} \tag{22}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2018 doi:10.20944/preprints201810.0239.v1

Peer-reviewed version available at Electronics 2019, 8, 59; doi:10.3390/electronics8010059

$$
\begin{aligned}
&b_{1,1} = b_{1,1} - \eta \times \frac{\partial E}{\partial b_{1,1}}, \text{ where} \\
&\frac{\partial E}{\partial b_{1,1}} = \frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_i}\frac{\partial \tilde{x}_i}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial h_1}\frac{\partial h_1}{\partial z_{1,1}}\frac{\partial z_{1,1}}{\partial b_{1,1}} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_i}\frac{\partial \tilde{y}_i}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial h_1}\frac{\partial h_1}{\partial z_{1,1}}\frac{\partial z_{1,1}}{\partial b_{1,1}} \\
&= \left(\frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_i}\frac{\partial \tilde{x}_i}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial h_1} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_i}\frac{\partial \tilde{y}_i}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial h_1}\right)\frac{\partial h_1}{\partial z_{1,1}}\frac{\partial z_{1,1}}{\partial b_{1,1}} \\
&= \left\{\sigma_1 \times \tilde{x}_i \times [1-\tilde{x}_i] \times \gamma_1 + \sigma_2 \times \tilde{y}_i \times [1-\tilde{y}_i] \times \gamma_2\right\} \times h_1 \times [1-h_1]
\end{aligned} \tag{23}
$$

For the generalization of recurrent neural network, the number of base stations and the number of Wi-Fi APs can be extended as n_1 and n_2 in the input layer (shown in Figure 5). The value of h_1 can be revised and calculated by Equation (24); the updates of α_j and β_k are proved and calculated by Equations (25) and (26), respectively.

$$
h_1 = s\left(\sum_{j=1}^{n_1}\alpha_j \times c_{j,i} + \sum_{k=1}^{n_2}\beta_k \times w_{k,i} + v \times h_0 + b_{1,1}\right) = s(z_{1,1}), \text{where } s(z) = \frac{1}{1+e^{-z}} \tag{24}
$$

$$
\begin{aligned}
&\alpha_j = \alpha_j - \eta \times \frac{\partial E}{\partial \alpha_j}, \text{ where} \\
&\frac{\partial E}{\partial \alpha_j} = \frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_i}\frac{\partial \tilde{x}_i}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial h_1}\frac{\partial h_1}{\partial z_{1,1}}\frac{\partial z_{1,1}}{\partial \alpha_1} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_i}\frac{\partial \tilde{y}_i}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial h_1}\frac{\partial h_1}{\partial z_{1,1}}\frac{\partial z_{1,1}}{\partial \alpha_j} \\
&= \left(\frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_i}\frac{\partial \tilde{x}_i}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial h_1} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_i}\frac{\partial \tilde{y}_i}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial h_1}\right)\frac{\partial h_1}{\partial z_{1,1}}\frac{\partial z_{1,1}}{\partial \alpha_j} \\
&= \left\{\sigma_1 \times \tilde{x}_i \times [1-\tilde{x}_i] \times \gamma_1 + \sigma_2 \times \tilde{y}_i \times [1-\tilde{y}_i] \times \gamma_2\right\} \times h_1 \times [1-h_1] \times c_{j,i}
\end{aligned} \tag{25}
$$

$$
\begin{aligned}
&\beta_k = \beta_k - \eta \times \frac{\partial E}{\partial \beta_k}, \text{ where} \\
&\frac{\partial E}{\partial \beta_k} = \frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_i}\frac{\partial \tilde{x}_i}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial h_1}\frac{\partial h_1}{\partial z_{1,1}}\frac{\partial z_{1,1}}{\partial \alpha_2} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_i}\frac{\partial \tilde{y}_i}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial h_1}\frac{\partial h_1}{\partial z_{1,1}}\frac{\partial z_{1,1}}{\partial \beta_k} \\
&= \left(\frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_i}\frac{\partial \tilde{x}_i}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial h_1} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_i}\frac{\partial \tilde{y}_i}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial h_1}\right)\frac{\partial h_1}{\partial z_{1,1}}\frac{\partial z_{1,1}}{\partial \beta_k} \\
&= \left\{\sigma_1 \times \tilde{x}_i \times [1-\tilde{x}_i] \times \gamma_1 + \sigma_2 \times \tilde{y}_i \times [1-\tilde{y}_i] \times \gamma_2\right\} \times h_1 \times [1-h_1] \times w_{k,i}
\end{aligned} \tag{26}
$$

 doi:10.20944/preprints201810.0239.v1

Peer-reviewed version available at *Electronics* **2019**, *8*, 59; doi:10.3390/electronics8010059

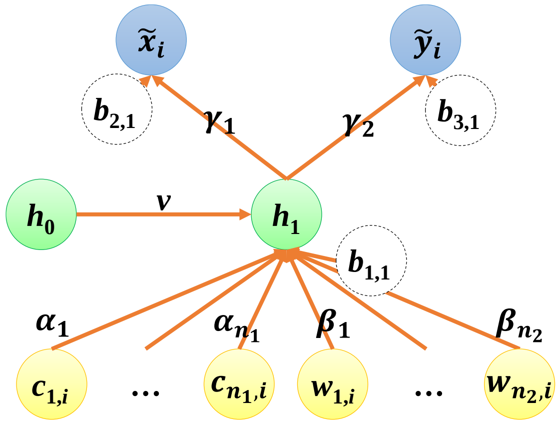


Figure 5. A generalized recurrent neural network with one timestamp

Furthermore, the number of neurons in the recurrent hidden layer can be extended for the extraction of time series data. The weight between each two neurons can be updated by the gradient descent method.

3.2.2.2. Two Timestamps for Recurrent Neural Network

This subsection illustrates the designs and optimization of recurrent neural networks with two consecutive timestamps. A simple case study of a recurrent neural network with two consecutive timestamps is showed in Figure 6. In the case, the recurrent neural network is constructed with an input layer, a recurrent hidden layer, and an output layer. The input layer includes four normalized RSSIs (i.e., $c_{1,i}$, $c_{2,i}$, $w_{1,i}$, and $w_{2,i}$) in the first timestamp and four normalized RSSIs (i.e., $c_{1,i+1}$, $c_{2,i+1}$, $w_{1,i+1}$, and $w_{2,i+1}$) in the second timestamp; the output layer includes the estimated normalized longitude and latitude (i.e., $\tilde{x}_{i+1}$ and $\tilde{y}_{i+1}$) in the second timestamp. The recurrent hidden layer includes a neuron, and the initial value of the neuron in the recurrent hidden layer is defined as h_0 (shown in Equations (9)). The value of the neuron in the recurrent hidden layer can be updated as h_1 in the first timestamp and be updated as h_2 in the second timestamp. The weights of Base Station 1, Base Station 2, Wi-Fi AP 1, and Wi-Fi AP in each timestamp are α_1, α_2, β_1, and β_2; the weights of h_2 for the outputs $\tilde{x}_{i+1}$ and $\tilde{y}_{i+1}$ are γ_1 and γ_2, respectively. Furthermore, the weight of the neurons in the recurrent hidden layer in least timestamp is defined as v. In the case, the biases of neurons in the hidden layer and the output layer are defined as $b_{1,1}$, $b_{2,1}$, and $b_{3,1}$. The sigmoid function is elected as the activation function of each neuron, so the values of h_1, h_2, $\tilde{x}_i$, and $\tilde{y}_i$ can be calculated by Equations (27), (28), (29), and (30). Furthermore, the loss function is defined as Equation (31) in accordance with squared errors.

$$h_1 = s\left(\sum_{j=1}^{2}\alpha_j \times c_{j,i} + \sum_{k=1}^{2}\beta_k \times w_{k,i} + v \times h_0 + b_{1,1}\right) = s\left(z_{1,1}\right), \text{where } s\left(z\right) = \frac{1}{1+e^{-z}} \tag{27}$$

$$h_2 = s\left(\sum_{j=1}^{2}\alpha_j \times c_{j,i+1} + \sum_{k=1}^{2}\beta_k \times w_{k,i+1} + v \times h_1 + b_{1,1}\right) = s\left(z_{1,2}\right), \text{where } s\left(z\right) = \frac{1}{1+e^{-z}} \tag{28}$$

$$\tilde{x}_{i+1} = s\left(\gamma_1 \times h_2 + b_{2,1}\right) = s\left(z_{2,1}\right), \text{where } s\left(z\right) = \frac{1}{1+e^{-z}} \tag{29}$$

$$\tilde{y}_{i+1} = s\left(\gamma_2 \times h_2 + b_{3,1}\right) = s\left(z_{3,1}\right), \text{where } s\left(z\right) = \frac{1}{1+e^{-z}} \tag{30}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2018 doi:10.20944/preprints201810.0239.v1

Peer-reviewed version available at *Electronics* **2019**, *8*, 59; doi:10.3390/electronics8010059

$$E=\frac{1}{2}\left(\tilde{x}_{i+1}-x_{i+1}\right)^2+\frac{1}{2}\left(\tilde{y}_{i+1}-y_{i+1}\right)^2=\frac{1}{2}\sigma_1{}^2+\frac{1}{2}\sigma_2{}^2 \tag{31}$$

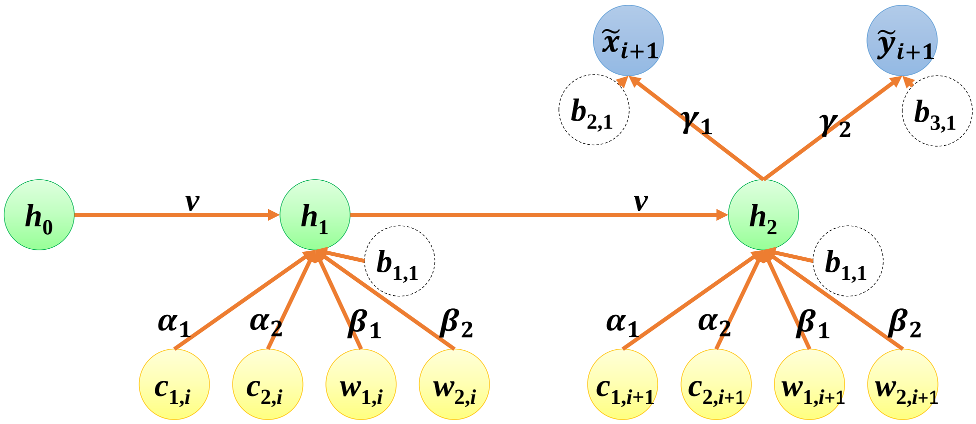


Figure 6. A recurrent neural network with two consecutive timestamps

For the optimization of recurrent neural network with two consecutive timestamps, the learning rate η and a gradient descent method is applied to update each weight and bias. The updates of γ_1, γ_2, $b_{2,1}$, $b_{3,1}$, α_1, α_2, β_1, β_2, v, and $b_{1,1}$ are proved and calculated by Equations (32), (33), (34), (35), (36), (37), (38), (39), (40), and (41), respectively.

$$\gamma_1=\gamma_1-\eta\times\frac{\partial E}{\partial\gamma_1}\text{, where}$$

$$\frac{\partial E}{\partial\gamma_1}=\frac{\partial E}{\partial\sigma_1}\frac{\partial\sigma_1}{\partial\tilde{x}_{i+1}}\frac{\partial\tilde{x}_{i+1}}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial\gamma_1}+\frac{\partial E}{\partial\sigma_2}\frac{\partial\sigma_2}{\partial\tilde{y}_{i+1}}\frac{\partial\tilde{y}_{i+1}}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial\gamma_1} \tag{32}$$

$$=\sigma_1\times\tilde{x}_{i+1}\times\left[1-\tilde{x}_{i+1}\right]\times h_2$$

$$\gamma_2=\gamma_2-\eta\times\frac{\partial E}{\partial\gamma_2}\text{, where}$$

$$\frac{\partial E}{\partial\gamma_2}=\frac{\partial E}{\partial\sigma_1}\frac{\partial\sigma_{1,}}{\partial\tilde{x}_{i+1}}\frac{\partial\tilde{x}_{i+1}}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial\gamma_2}+\frac{\partial E}{\partial\sigma_2}\frac{\partial\sigma_2}{\partial\tilde{y}_{i+1}}\frac{\partial\tilde{y}_{i+1}}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial\gamma_2} \tag{33}$$

$$=\sigma_2\times\tilde{y}_{i+1}\times\left[1-\tilde{y}_{i+1}\right]\times h_2$$

$$b_{2,1}=b_{2,1}-\eta\times\frac{\partial E}{\partial b_{2,1}}\text{, where}$$

$$\frac{\partial E}{\partial b_{2,1}}=\frac{\partial E}{\partial\sigma_1}\frac{\partial\sigma_1}{\partial\tilde{x}_{i+1}}\frac{\partial\tilde{x}_{i+1}}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial b_{2,1}}+\frac{\partial E}{\partial\sigma_2}\frac{\partial\sigma_2}{\partial\tilde{y}_{i+1}}\frac{\partial\tilde{y}_{i+1}}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial b_{2,1}} \tag{34}$$

$$=\sigma_1\times\tilde{x}_{i+1}\times\left[1-\tilde{x}_{i+1}\right]$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2018 doi:10.20944/preprints201810.0239.v1

Peer-reviewed version available at *Electronics* **2019**, *8*, 59; doi:10.3390/electronics8010059

$$b_{3,1} = b_{3,1} - \eta \times \frac{\partial E}{\partial b_{3,1}}, \text{ where}$$

$$\frac{\partial E}{\partial b_{3,1}} = \frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_{i+1}}\frac{\partial \tilde{x}_{i+1}}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial b_{3,1}} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_{i+1}}\frac{\partial \tilde{y}_{i+1}}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial b_{3,1}} = \sigma_2 \times \tilde{y}_{i+1} \times [1 - \tilde{y}_{i+1}] \tag{35}$$

$$\alpha_1 = \alpha_1 - \eta \times \frac{\partial E}{\partial \alpha_1}, \text{ where}$$

$$\begin{aligned}\frac{\partial E}{\partial \alpha_1} &= \frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_{i+1}}\frac{\partial \tilde{x}_{i+1}}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial h_2}\frac{\partial h_2}{\partial z_{1,2}}\frac{\partial z_{1,2}}{\partial \alpha_1} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_{i+1}}\frac{\partial \tilde{y}_{i+1}}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial h_2}\frac{\partial h_2}{\partial z_{1,2}}\frac{\partial z_{1,2}}{\partial \alpha_1} \\ &= \left(\frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_{i+1}}\frac{\partial \tilde{x}_{i+1}}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial h_2} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_{i+1}}\frac{\partial \tilde{y}_{i+1}}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial h_2}\right)\frac{\partial h_2}{\partial z_{1,2}}\frac{\partial z_{1,2}}{\partial \alpha_1} \\ &= \{\sigma_1 \times \tilde{x}_{i+1} \times [1-\tilde{x}_{i+1}] \times \gamma_1 + \sigma_2 \times \tilde{y}_{i+1} \times [1-\tilde{y}_{i+1}] \times \gamma_2\} \times h_2 \times [1-h_2] \times \left(c_{1,i+1} + v\frac{\partial h_1}{\partial z_{1,1}}\frac{\partial z_{1,1}}{\partial \alpha_1}\right) \\ &= \{\sigma_1 \times \tilde{x}_{i+1} \times [1-\tilde{x}_{i+1}] \times \gamma_1 + \sigma_2 \times \tilde{y}_{i+1} \times [1-\tilde{y}_{i+1}] \times \gamma_2\} \times h_2 \times [1-h_2] \times \left(c_{1,i+1} + v \times h_1 \times [1-h_1] \times c_{1,i}\right)\end{aligned} \tag{36}$$

$$\alpha_2 = \alpha_2 - \eta \times \frac{\partial E}{\partial \alpha_2}, \text{ where}$$

$$\begin{aligned}\frac{\partial E}{\partial \alpha_2} &= \frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_{i+1}}\frac{\partial \tilde{x}_{i+1}}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial h_2}\frac{\partial h_2}{\partial z_{1,2}}\frac{\partial z_{1,2}}{\partial \alpha_2} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_{i+1}}\frac{\partial \tilde{y}_{i+1}}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial h_2}\frac{\partial h_2}{\partial z_{1,2}}\frac{\partial z_{1,2}}{\partial \alpha_2} \\ &= \left(\frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_{i+1}}\frac{\partial \tilde{x}_{i+1}}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial h_2} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_{i+1}}\frac{\partial \tilde{y}_{i+1}}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial h_2}\right)\frac{\partial h_2}{\partial z_{1,2}}\frac{\partial z_{1,2}}{\partial \alpha_2} \\ &= \{\sigma_1 \times \tilde{x}_{i+1} \times [1-\tilde{x}_{i+1}] \times \gamma_1 + \sigma_2 \times \tilde{y}_{i+1} \times [1-\tilde{y}_{i+1}] \times \gamma_2\} \times h_2 \times [1-h_2] \times \left(c_{2,i+1} + v\frac{\partial h_1}{\partial z_{1,1}}\frac{\partial z_{1,1}}{\partial \alpha_2}\right) \\ &= \{\sigma_1 \times \tilde{x}_{i+1} \times [1-\tilde{x}_{i+1}] \times \gamma_1 + \sigma_2 \times \tilde{y}_{i+1} \times [1-\tilde{y}_{i+1}] \times \gamma_2\} \times h_2 \times [1-h_2] \times \left(c_{2,i+1} + v \times h_1 \times [1-h_1] \times c_{2,i}\right)\end{aligned} \tag{37}$$

$$\beta_1 = \beta_1 - \eta \times \frac{\partial E}{\partial \beta_1}, \text{ where}$$

$$\begin{aligned}\frac{\partial E}{\partial \beta_1} &= \frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_{i+1}}\frac{\partial \tilde{x}_{i+1}}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial h_2}\frac{\partial h_2}{\partial z_{1,2}}\frac{\partial z_{1,2}}{\partial \beta_1} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_{i+1}}\frac{\partial \tilde{y}_{i+1}}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial h_2}\frac{\partial h_2}{\partial z_{1,2}}\frac{\partial z_{1,2}}{\partial \beta_1} \\ &= \left(\frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_{i+1}}\frac{\partial \tilde{x}_{i+1}}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial h_2} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_{i+1}}\frac{\partial \tilde{y}_{i+1}}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial h_2}\right)\frac{\partial h_2}{\partial z_{1,2}}\frac{\partial z_{1,2}}{\partial \beta_1} \\ &= \{\sigma_1 \times \tilde{x}_{i+1} \times [1-\tilde{x}_{i+1}] \times \gamma_1 + \sigma_2 \times \tilde{y}_{i+1} \times [1-\tilde{y}_{i+1}] \times \gamma_2\} \times h_2 \times [1-h_2] \times \left(w_{1,i+1} + v\frac{\partial h_1}{\partial z_{1,1}}\frac{\partial z_{1,1}}{\partial \beta_1}\right) \\ &= \{\sigma_1 \times \tilde{x}_{i+1} \times [1-\tilde{x}_{i+1}] \times \gamma_1 + \sigma_2 \times \tilde{y}_{i+1} \times [1-\tilde{y}_{i+1}] \times \gamma_2\} \times h_2 \times [1-h_2] \times \left(w_{1,i+1} + v \times h_1 \times [1-h_1] \times w_{1,i}\right)\end{aligned} \tag{38}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2018 doi:10.20944/preprints201810.0239.v1

Peer-reviewed version available at *Electronics* **2019**, *8*, 59; doi:10.3390/electronics8010059

$$\beta_2 = \beta_2 - \eta \times \frac{\partial E}{\partial \beta_2}, \text{ where}$$

$$\begin{aligned}
\frac{\partial E}{\partial \beta_2} &= \frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_{i+1}}\frac{\partial \tilde{x}_{i+1}}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial h_2}\frac{\partial h_2}{\partial z_{1,2}}\frac{\partial z_{1,2}}{\partial \alpha_2} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_{i+1}}\frac{\partial \tilde{y}_{i+1}}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial h_2}\frac{\partial h_2}{\partial z_{1,2}}\frac{\partial z_{1,2}}{\partial \beta_2} \\
&= \left(\frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_{i+1}}\frac{\partial \tilde{x}_{i+1}}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial h_2} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_{i+1}}\frac{\partial \tilde{y}_{i+1}}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial h_2}\right)\frac{\partial h_2}{\partial z_{1,2}}\frac{\partial z_{1,2}}{\partial \beta_2} \\
&= \left\{\sigma_1 \times \tilde{x}_{i+1} \times [1-\tilde{x}_{i+1}] \times \gamma_1 + \sigma_2 \times \tilde{y}_{i+1} \times [1-\tilde{y}_{i+1}] \times \gamma_2\right\} \times h_2 \times [1-h_2] \times \left(w_{2,i+1} + v\frac{\partial h_1}{\partial z_{1,1}}\frac{\partial z_{1,1}}{\partial \beta_2}\right) \\
&= \left\{\sigma_1 \times \tilde{x}_{i+1} \times [1-\tilde{x}_{i+1}] \times \gamma_1 + \sigma_2 \times \tilde{y}_{i+1} \times [1-\tilde{y}_{i+1}] \times \gamma_2\right\} \times h_2 \times [1-h_2] \times \left(w_{2,i+1} + v \times h_1 \times [1-h_1] \times w_{2,i}\right)
\end{aligned} \tag{39}$$

$$v = v - \eta \times \frac{\partial E}{\partial v}, \text{ where}$$

$$\begin{aligned}
\frac{\partial E}{\partial v} &= \frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_{i+1}}\frac{\partial \tilde{x}_{i+1}}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial h_2}\frac{\partial h_2}{\partial z_{1,2}}\frac{\partial z_{1,2}}{\partial v} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_{i+1}}\frac{\partial \tilde{y}_{i+1}}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial h_2}\frac{\partial h_2}{\partial z_{1,2}}\frac{\partial z_{1,2}}{\partial v} \\
&= \left(\frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_{i+1}}\frac{\partial \tilde{x}_{i+1}}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial h_2} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_{i+1}}\frac{\partial \tilde{y}_{i+1}}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial h_2}\right)\frac{\partial h_2}{\partial z_{1,2}}\frac{\partial z_{1,2}}{\partial v} \\
&= \left\{\sigma_1 \times \tilde{x}_{i+1} \times [1-\tilde{x}_{i+1}] \times \gamma_1 + \sigma_2 \times \tilde{y}_{i+1} \times [1-\tilde{y}_{i+1}] \times \gamma_2\right\} \times h_2 \times [1-h_2] \times \left(\frac{\partial v}{\partial v}h_1 + v \times \frac{\partial h_1}{\partial z_{1,1}}\frac{\partial z_{1,1}}{\partial v}\right) \\
&= \left\{\sigma_1 \times \tilde{x}_{i+1} \times [1-\tilde{x}_{i+1}] \times \gamma_1 + \sigma_2 \times \tilde{y}_{i+1} \times [1-\tilde{y}_{i+1}] \times \gamma_2\right\} \times h_2 \times [1-h_2] \times \left(h_1 + v \times h_1 \times [1-h_1] \times h_0\right)
\end{aligned} \tag{40}$$

$$b_{1,1} = b_{1,1} - \eta \times \frac{\partial E}{\partial b_{1,1}}, \text{ where}$$

$$\begin{aligned}
\frac{\partial E}{\partial b_{1,1}} &= \frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_{i+1}}\frac{\partial \tilde{x}_{i+1}}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial h_2}\frac{\partial h_2}{\partial z_{1,2}}\frac{\partial z_{1,2}}{\partial b_{1,1}} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_{i+1}}\frac{\partial \tilde{y}_{i+1}}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial h_2}\frac{\partial h_2}{\partial z_{1,2}}\frac{\partial z_{1,2}}{\partial b_{1,1}} \\
&= \left(\frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_{i+1}}\frac{\partial \tilde{x}_{i+1}}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial h_2} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_{i+1}}\frac{\partial \tilde{y}_{i+1}}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial h_2}\right)\frac{\partial h_2}{\partial z_{1,2}}\frac{\partial z_{1,2}}{\partial b_{1,1}} \\
&= \left\{\sigma_1 \times \tilde{x}_{i+1} \times [1-\tilde{x}_{i+1}] \times \gamma_1 + \sigma_2 \times \tilde{y}_{i+1} \times [1-\tilde{y}_{i+1}] \times \gamma_2\right\} \times h_2 \times [1-h_2] \times \left(1 + v\frac{\partial h_1}{\partial z_{1,1}}\frac{\partial z_{1,1}}{\partial b_{1,1}}\right) \\
&= \left\{\sigma_1 \times \tilde{x}_{i+1} \times [1-\tilde{x}_{i+1}] \times \gamma_1 + \sigma_2 \times \tilde{y}_{i+1} \times [1-\tilde{y}_{i+1}] \times \gamma_2\right\} \times h_2 \times [1-h_2] \times \left(1 + v \times h_1 \times [1-h_1]\right)
\end{aligned} \tag{41}$$

For the generalization of recurrent neural network, the number of base stations and the number of Wi-Fi APs can be extended as n_1 and n_2 in the input layer (shown in Figure 7). The values of h_1 and h_2 can be revised and calculated by Equation (42) and (43); the updates of α_j and β_k are proved and calculated by Equations (44) and (45), respectively.

$$h_1 = s\left(\sum_{j=1}^{n_1}\alpha_j \times c_{j,i} + \sum_{k=1}^{n_2}\beta_k \times w_{k,i} + v \times h_0 + b_{1,1}\right) = s\left(z_{1,1}\right), \text{where } s(z) = \frac{1}{1+e^{-z}} \tag{42}$$

$$h_2 = s\left(\sum_{j=1}^{n_1}\alpha_j \times c_{j,i+1} + \sum_{k=1}^{n_2}\beta_k \times w_{k,i+1} + v \times h_1 + b_{1,1}\right) = s\left(z_{1,2}\right), \text{where } s(z) = \frac{1}{1+e^{-z}} \tag{43}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2018 doi:10.20944/preprints201810.0239.v1

Peer-reviewed version available at Electronics 2019, 8, 59; doi:10.3390/electronics8010059

$$
\begin{aligned}
&\alpha_j = \alpha_j - \eta \times \frac{\partial E}{\partial \alpha_j}, \text{ where} \\
&\frac{\partial E}{\partial \alpha_j} = \frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_{i+1}}\frac{\partial \tilde{x}_{i+1}}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial h_2}\frac{\partial h_2}{\partial z_{1,2}}\frac{\partial z_{1,2}}{\partial \alpha_1} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_{i+1}}\frac{\partial \tilde{y}_{i+1}}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial h_2}\frac{\partial h_2}{\partial z_{1,2}}\frac{\partial z_{1,2}}{\partial \alpha_j} \\
&= \left(\frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_{i+1}}\frac{\partial \tilde{x}_{i+1}}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial h_2} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_{i+1}}\frac{\partial \tilde{y}_{i+1}}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial h_2} \right)\frac{\partial h_2}{\partial z_{1,2}}\frac{\partial z_{1,2}}{\partial \alpha_j} \\
&= \left\{ \sigma_1 \times \tilde{x}_{i+1} \times [1-\tilde{x}_{i+1}] \times \gamma_1 + \sigma_2 \times \tilde{y}_{i+1} \times [1-\tilde{y}_{i+1}] \times \gamma_2 \right\} \times h_2 \times [1-h_2] \times \left(c_{j,i+1} + v \frac{\partial h_1}{\partial z_{1,1}}\frac{\partial z_{1,1}}{\partial \alpha_j} \right) \\
&= \left\{ \sigma_1 \times \tilde{x}_{i+1} \times [1-\tilde{x}_{i+1}] \times \gamma_1 + \sigma_2 \times \tilde{y}_{i+1} \times [1-\tilde{y}_{i+1}] \times \gamma_2 \right\} \times h_2 \times [1-h_2] \times \left(c_{j,i+1} + v \times h_1 \times [1-h_1] \times c_{j,i} \right)
\end{aligned} \tag{44}
$$

$$
\begin{aligned}
&\beta_k = \beta_k - \eta \times \frac{\partial E}{\partial \beta_k}, \text{ where} \\
&\frac{\partial E}{\partial \beta_k} = \frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_{i+1}}\frac{\partial \tilde{x}_{i+1}}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial h_2}\frac{\partial h_2}{\partial z_{1,2}}\frac{\partial z_{1,2}}{\partial \alpha_2} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_{i+1}}\frac{\partial \tilde{y}_{i+1}}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial h_2}\frac{\partial h_2}{\partial z_{1,2}}\frac{\partial z_{1,2}}{\partial \beta_k} \\
&= \left(\frac{\partial E}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial \tilde{x}_{i+1}}\frac{\partial \tilde{x}_{i+1}}{\partial z_{2,1}}\frac{\partial z_{2,1}}{\partial h_2} + \frac{\partial E}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \tilde{y}_{i+1}}\frac{\partial \tilde{y}_{i+1}}{\partial z_{3,1}}\frac{\partial z_{3,1}}{\partial h_2} \right)\frac{\partial h_2}{\partial z_{1,2}}\frac{\partial z_{1,2}}{\partial \beta_k} \\
&= \left\{ \sigma_1 \times \tilde{x}_{i+1} \times [1-\tilde{x}_{i+1}] \times \gamma_1 + \sigma_2 \times \tilde{y}_{i+1} \times [1-\tilde{y}_{i+1}] \times \gamma_2 \right\} \times h_2 \times [1-h_2] \times \left(w_{k,i+1} + v \frac{\partial h_1}{\partial z_{1,1}}\frac{\partial z_{1,1}}{\partial \beta_k} \right) \\
&= \left\{ \sigma_1 \times \tilde{x}_{i+1} \times [1-\tilde{x}_{i+1}] \times \gamma_1 + \sigma_2 \times \tilde{y}_{i+1} \times [1-\tilde{y}_{i+1}] \times \gamma_2 \right\} \times h_2 \times [1-h_2] \times \left(w_{k,i+1} + v \times h_1 \times [1-h_1] \times w_{k,i} \right)
\end{aligned} \tag{45}
$$

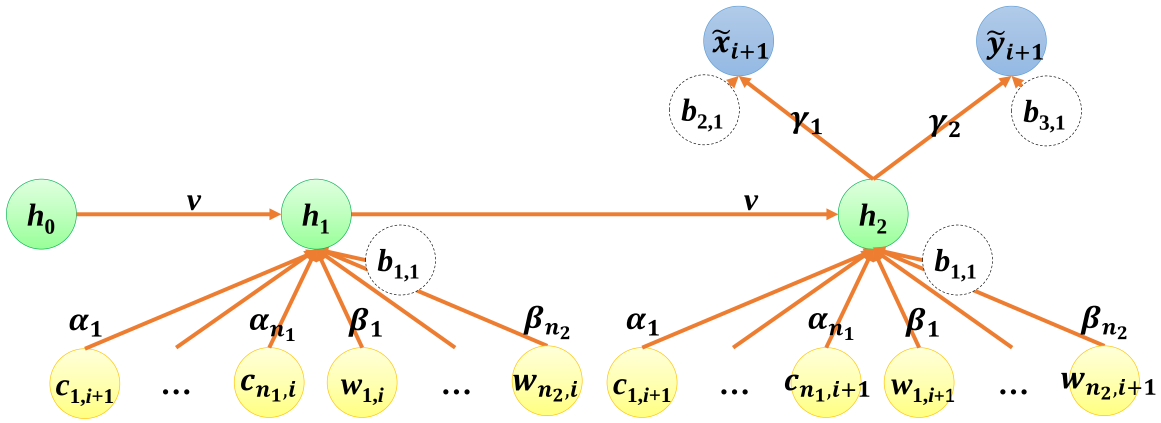


Figure 7. A generalized recurrent neural network with two consecutive timestamps

Furthermore, the recurrent neural network can analyze with more consecutive timestamps, and the number of neurons in the recurrent hidden layer of the recurrent neural network can be extended for the extraction of time series data. The weight between each two neurons can be updated by the gradient descent method.

3.2.3. De-normalization and Estimation

For de-normalization and estimation, the estimated normalized longitude and latitude (i.e., $\tilde{x}_i$ and $\tilde{y}_i$) can be adopted into Equations (46) and (47) to retrieve the estimated longitude and latitude (i.e., $\tilde{l}_{x,i}$ and $\tilde{l}_{y,i}$).

$$\tilde{l}_{x,i} = \tilde{x}_i \times \left(l_x^{(+)} - l_x^{(-)} \right) + l_x^{(-)} \tag{46}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2018 doi:10.20944/preprints201810.0239.v1

Peer-reviewed version available at Electronics 2019, 8, 59; doi:10.3390/electronics8010059

$$\tilde{l}_{y,i} = \tilde{y}_i \times \left(l_y^{(+)} - l_y^{(-)} \right) + l_y^{(-)} \tag{47}$$

4. Practical Experimental Results and Discussion

This section presents and discusses the practical experimental results. Practical experimental environments are illustrated in Subsection 4.1, and practical experimental results are showed in Subsection 4.2. Subsection 4.3 discussed the results of different recurrent neural networks.

4.1. Practical Experimental Environments

In the practical experimental environments, an Android application was implemented and installed into mobile stations (e.g., Redmi 5 running Android platform 7.1.2). The mobile stations were carried out on a 5.6 km long road segment in Fuzhou University in China (shown in Figure 8). The segment was traversed 8 times by mobile stations to collect GPS coordinates and network signals (i.e., the RSSIs of base stations and Wi-Fi APs). There are 4,525 records (i.e., m = 4,525), 59 different base stations (i.e., n_1 = 59), and 582 different Wi-Fi APs (i.e., n_2 = 582) detected in the experiments. This study selected 2,263 records including GPS coordinates and RSSIs as training data, and other 2,262 records were selected as testing data.

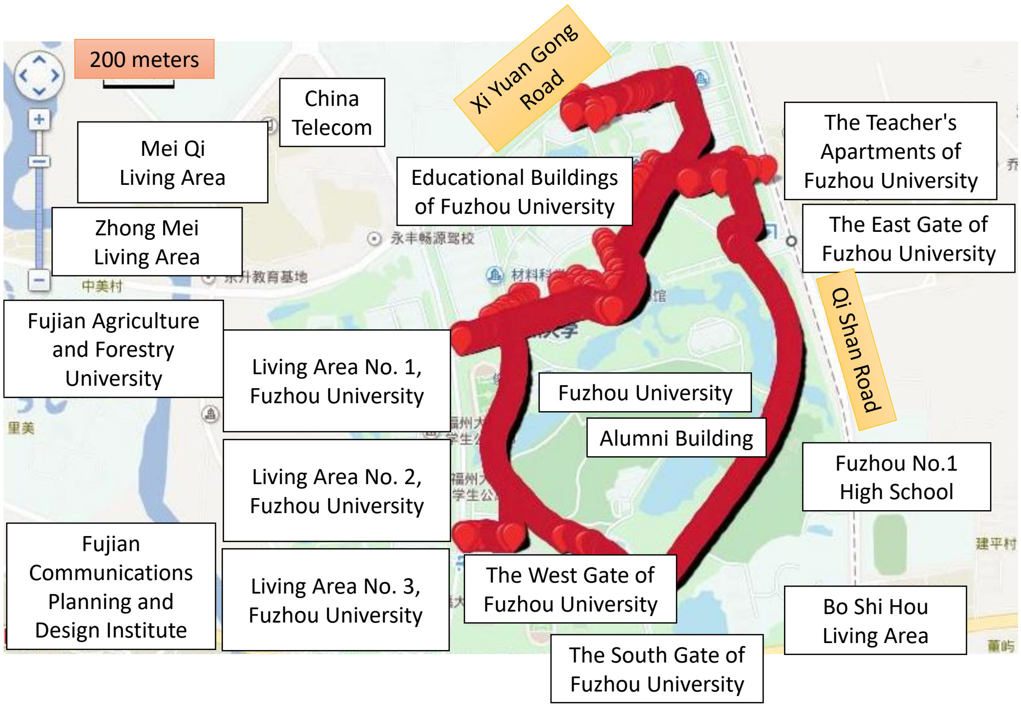


Figure 8. Practical experimental environments

4.2. Practical Experimental Results

For the evaluation of the proposed mobile positioning method, 9 experimental cases with different timestamp numbers (i.e., 1 timestamp, 2 timestamps, and 3 timestamps) and with different mobile networks (i.e., only cellular networks, only Wi-Fi networks, and cellular and Wi-Fi networks) were designed and performed. There were 30 neurons in the recurrent hidden layer of the recurrent neural network for each experimental case. The practical experimental results are showed in Table 1, Figure 9, Figure 10, Figure 11, and Figure 12. Table 1 and Figure 9 illustrated that the more precise location can be estimated by the proposed method with heterogeneous networks (i.e., cellular and Wi-Fi networks). The higher location errors may be obtained by the recurrent neural networks with one timestamp (i.e., traditional neural networks) which cannot extract the feature of time series data (shown in Table 1 and Figure 10). The lower location errors can be obtained by the recurrent neural

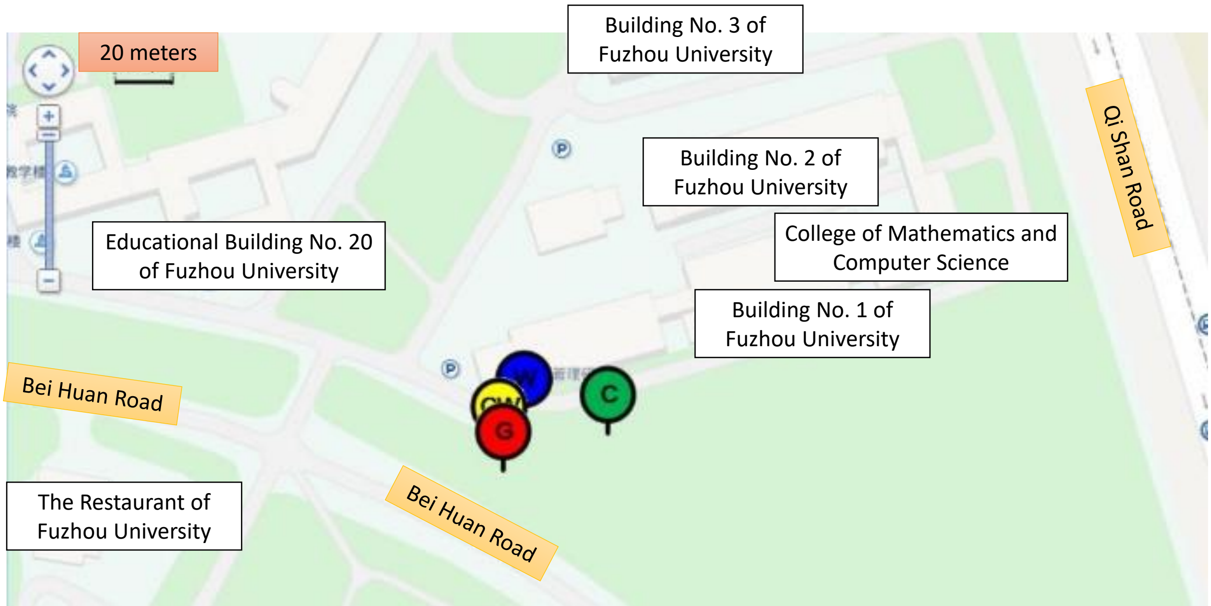
Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2018 doi:10.20944/preprints201810.0239.v1

Peer-reviewed version available at Electronics 2019, 8, 59; doi:10.3390/electronics8010059

networks with multiple consecutive timestamps (e.g., 2 timestamps and 3 timestamps); the experimental results can be observed that the average error of location estimation was 9.19 meters by the proposed mobile positioning method with 2 timestamps.

Table 1. The average errors of estimated locations by the proposed mobile positioning method (Unit: meters)

Number of timestamps	Only cellular networks	Only Wi-Fi networks	cellular and Wi-Fi networks
1 timestamp	39.88	18.88	16.21
2 timestamps	36.51	18.69	9.19
3 timestamps	34.57	17.83	9.26



G: GPS (a red point); C: cellular networks (a green point); W: Wi-Fi networks (a blue point); CW: cellular and Wi-Fi networks (a yellow point)

Figure 9. The estimated locations by the proposed mobile positioning method with different mobile networks

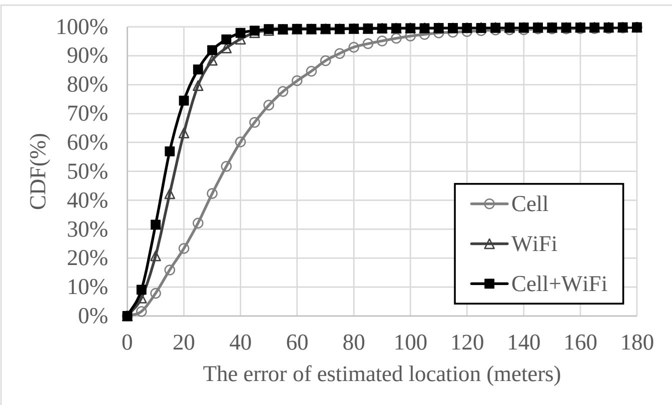


Figure 10. The cumulative distribution function of location errors by the proposed mobile positioning method with 1 timestamp

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2018 doi:10.20944/preprints201810.0239.v1

Peer-reviewed version available at Electronics 2019, 8, 59; doi:10.3390/electronics8010059

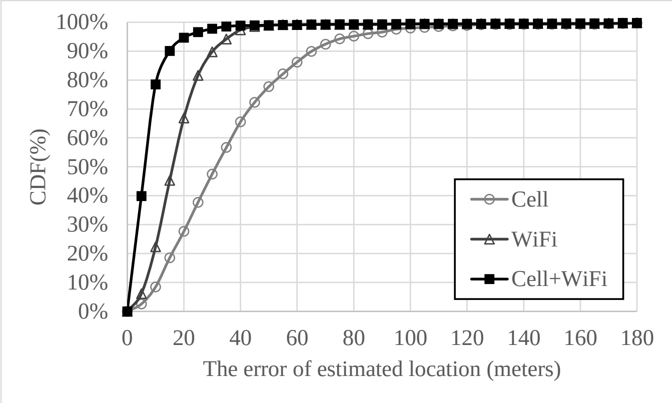


Figure 11. The cumulative distribution function of location errors by the proposed mobile positioning method with 2 timestamps

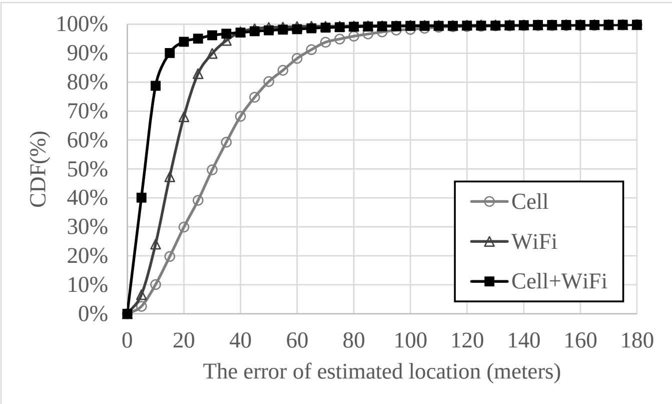


Figure 12. The cumulative distribution function of location errors by the proposed mobile positioning method with 3 timestamps

4.3. Discussions

The proposed mobile positioning method used a trained recurrent neural network to simultaneously estimate longitudes and latitudes; in the recurrent neural network, the estimated longitudes and latitudes were determined in accordance with the same weights in the input layer and hidden layers. In addition, this study also considered to separately train two recurrent neural networks for estimating longitudes and latitudes (shown in Figures 13 and 14); the estimated longitudes and latitudes were determined in accordance with different weights in these recurrent neural networks. The practical experimental results indicated that higher precise location may be obtained by the recurrent neural networks with one timestamp (i.e., traditional neural network)(shown in Table 2). However, big errors of estimated locations may be obtained by the recurrent neural networks with multiple consecutive timestamps. The overfitting problems may exist if longitudes and latitudes are estimated by different recurrent neural networks with multiple consecutive timestamps. Therefore, the interaction effects of longitudes and latitudes should be analyzed, so they should be estimated by the same recurrent neural network for determining higher precise locations.

Table 2. The average errors of estimated locations by the proposed mobile positioning method (Unit: meters)

Number of timestamps	Only cellular networks	Only Wi-Fi networks	cellular and Wi-Fi networks
1	34.61	16.46	14.39
2	259.44	255.87	252.53
3	254.85	256.61	253.85

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2018 doi:10.20944/preprints201810.0239.v1

Peer-reviewed version available at *Electronics* **2019**, *8*, 59; doi:10.3390/electronics8010059

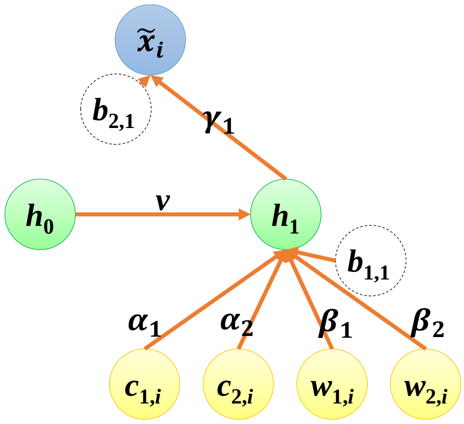


Figure 13. A recurrent neural network with one timestamp for estimating longitudes

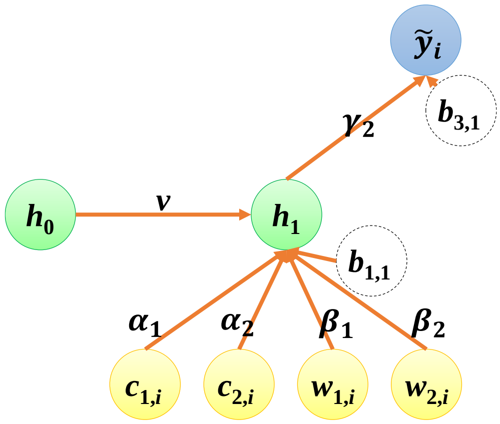


Figure 14. A recurrent neural network with one timestamp for estimating latitudes

5. Conclusions and Future Work

This section summarizes and describes the contributions of this study in Subsection 5.1. The limitations of the proposed method and future work are presented in Subsection 5.2.

5.1. Conclusions

In previous studies, cellular-based positioning methods can estimate locations of mobile stations in outdoor environments, but the accuracies of estimated locations may be lower. Moreover, Wi-Fi-based positioning methods can precisely estimate the locations mobile stations, but the transmission coverage of Wi-Fi APs is not enough in outdoor environments. Therefore, a mobile positioning system and a mobile positioning method based on recurrent neural networks are proposed to analyze the RSSIs from heterogeneous networks which include cellular networks and Wi-Fi networks. The network signals from heterogeneous networks can be analyzed to improve the accuracies of estimation locations. Furthermore, the RSSIs in multiple consecutive timestamps can be adopted into recurrent neural networks for the analyses of time series data and locations estimation. In practical experimental environments, the results showed that the average error of location estimation was 9.19 meters by the proposed mobile positioning method with 2 timestamps. Therefore, the proposed system and method can be applied to obtain LBS in outdoor environments.

5.2. Future Work

Although the higher accuracies of estimation locations can be obtained by recurrent neural networks with multiple consecutive timestamps, some overfitting problems may exist. For instance, the higher errors of estimated locations were obtained by recurrent neural networks with 3

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2018 doi:10.20944/preprints201810.0239.v1

Peer-reviewed version available at Electronics 2019, 8, 59; doi:10.3390/electronics8010059

timestamps. Therefore, overfitting solutions of time series data [25] can be investigated to improve the accuracies of estimated locations in the future.

Author Contributions: Wu and Chen proposed and implemented the methodology. Chen analyzed and discussed the practical experimental results. Wu, Chen, and Zhang wrote the manuscript.

Funding: The research was funded by the Natural Science Foundation of China under the project of 61300104, the Fujian Industry-Academy Cooperation Project under Grant No. 2017H6008, and the Natural Science Foundation of Fujian Province of China under the project of 2018J01791 and 2017J01752.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chen, C.H.; Lin, B.Y.; Lin, C.H.; Liu, Y.S.; Lo, C.C. A green positioning algorithm for Campus Guidance System. *International Journal of Mobile Communications* **2012**, *10(2)*, 119-131, DOI: 10.1504/IJMC.2012.045669
2. Wu, C.; Yang, Z.; Xu, Y.; Zhao, Y.; Liu, Y. Human mobility enhances global positioning accuracy for mobile phone localization. *IEEE Transactions on Parallel and Distributed Systems* **2015**, *26(1)*, 131-141, DOI: 10.1109/TPDS.2014.2308225
3. Thejaswini, M.; Rajalakshmi, P.; Desai, U.B. Novel sampling algorithm for human mobility-based mobile phone sensing. *IEEE Internet of Things Journal* **2015**, *2(3)*, 210-220, DOI: 10.1109/JIOT.2014.2388074
4. Chen, C.H. An arrival time prediction method for bus system. *IEEE Internet Things Journal* **2018**, *Early Access*, DOI: 10.1109/JIOT.2018.2863555
5. Molina, B.; Olivares, E.; Palau, C.E.; Esteve, M. A multimodal fingerprint-based indoor positioning system for airports. *IEEE Access* **2018**, *6*, 10092-10106, DOI: 10.1109/ACCESS.2018.2798918
6. Chen, C.H.; Lee, C.A.; Lo, C.C. Vehicle localization and velocity estimation based on mobile phone sensing. *IEEE Access* **2016**, *4*, pp. 803-817, DOI: 10.1109/ACCESS.2016.2530806
7. Chen, K.; Wang, C.; Yin, Z.; Jiang, H.; Tan, G. Slide: towards fast and accurate mobile fingerprinting for Wi-Fi indoor positioning systems. *IEEE Sensors Journal* **2018**, *18(3)*, 1213-1223, DOI: 10.1109/JSEN.2017.2778082
8. Lai, W.K.; Kuo, T.H.; Chen, C.H. Vehicle Speed Estimation and Forecasting Methods Based on Cellular Floating Vehicle Data. *Applied Sciences* **2016**, *6*, 47, DOI: 10.3390/app6020047
9. Liu, D.; Sheng, B.; Hou, F.; Rao, W.; Liu, H. From wireless positioning to mobile positioning: an overview of recent advances. *IEEE Systems Journal* **2014**, *8(4)*, 1249-1259, DOI: 10.1109/JSYST.2013.2295136
10. Taniuchi, D.; Liu, X.; Nakai, D.; Maekawa, T. Spring model based collaborative indoor position estimation with neighbor mobile devices. *IEEE Journal of Selected Topics in Signal Processing* **2015**, *9(2)*, 268-277, DOI: 10.1109/JSTSP.2014.2382478
11. Cheng, D.Y.; Chen, C.H.; Hsiang, C.H.; Lo, C.C.; Lin, H.F.; Lin, B.Y. The optimal sampling period of a fingerprint positioning algorithm for vehicle speed estimation. *Mathematical Problems in Engineering* **2013**, *2013*, 306783, DOI: 10.1155/2013/306783.
12. Mok, E.; Cheung, B.K.S. An improved neural network training algorithm for Wi-Fi fingerprinting positioning. *ISPRS International Journal of Geo-Information* **2013**, *2(3)*, 854-868, DOI: 10.3390/ijgi2030854
13. Chen, C.H.; Lin, J.H.; Kuan, T.S.; Lo, K.R. A high-efficiency method of mobile positioning based on commercial vehicle operation data. *ISPRS International Journal of Geo-Information* **2016**, *5*, 82, DOI: 10.3390/ijgi5060082
14. Xia, S.; Liu, Y.; Yuan, G.; Zhu, M.; Wang, Z. Indoor fingerprint positioning based on Wi-Fi: an overview. *ISPRS International Journal of Geo-Information* **2017**, *6(5)*, 135, DOI: 10.3390/ijgi6050135
15. Chen, C.H.; Lo, K.R. Applications of Internet of Things. *ISPRS International Journal of Geo-Information* **2018**, *7*, 334, DOI: 10.3390/ijgi7090334
16. Lo, C.L.; Chen, C.H.; Kuan, T.S.; Lo, K.R.; Cho, H.J. Fuel consumption estimation system and method with lower cost. *Symmetry* **2017**, *9*, 105, DOI: 10.3390/sym9070105
17. Chen, C.H.; Al-Masri, E.; Hwang, F.J.; Ktoridou, D.; Lo, K.R. Introduction to the special issue: applications of Internet of things. *Symmetry* **2018**, 10, 374, DOI: 10.3390/sym10090374
18. Chen, C.Y.; Yin, L.P.; Chen, Y.J.; Hwang, R.C. A modified probability neural network indoor positioning technique. Proceedings of 2012 IEEE International Conference on Information Security and Intelligent Control, Yunlin, Taiwan, 14-16 Aug. 2012, DOI: 10.1109/ISIC.2012.6449770

19. Xu, Y.; Sun, Y. Neural network-based accuracy enhancement method for WLAN indoor positioning. Proceedings of 2012 IEEE Vehicular Technology Conference, Quebec City, QC, Canada, 3-6 Sept. 2012, DOI: 10.1109/VTCFall.2012.6399107
20. Zhang, T.; Man, Y. The enhancement of WiFi fingerprint positioning using convolutional neural network. Proceedings of 2018 International Conference on Computer, Communication and Network Technology, Wuzhen, China, 29-30 June 2018, DOI: 10.12783/dtcse/CCNT2018/24745
21. Zhu, J.Y.; Xu, J.; Zheng, A.X.; He, J.; Wu, C.; Li, V.O.K. WIFI fingerprinting indoor localization system based on spatio-temporal (S-T) metrics. Proceedings of 2014 IEEE International Conference on Indoor Positioning and Indoor Navigation, Busan, South Korea, 27-30 Oct. 2014, DOI: 10.1109/IPIN.2014.7275534
22. Lukito, Y.; Chrismanto, A.R. Recurrent neural networks model for WiFi-based indoor positioning system. Proceedings of 2017 IEEE International Conference on Smart Cities, Automation & Intelligent Computing Systems, Yogyakarta, Indonesia, 8-10 Nov. 2017, DOI: 10.1109/ICON-SONICS.2017.8267833
23. Wang, X.; Gao, L.; Mao, S.; Pandey, S. DeepFi: Deep learning for indoor fingerprinting using channel state information. Proceedings of 2015 IEEE Wireless Communications and Networking Conference, New Orleans, LA, USA, 9-12 March 2015, DOI: 10.1109/WCNC.2015.7127718
24. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436-444, DOI: https://doi.org/10.1038/nature14539
25. Chen, C.H. Reducing the dimensionality of time-series data with deep learning techniques. *Science* **2018**, eLetter. Available online: http://science.sciencemag.org/content/313/5786/504/tab-e-letters (accessed on 10 Ocotober 2018).