Deep Neural Network for RFID-Based Activity Recognition

Xinyu Li[1], Yanyi Zhang[1], Mengzhu Li[1], Ivan Marsic[1], JaeWon Yang[2], Randall S. Burd[2]
[1] Department of Electrical and Computer Engineering, Rutgers University, Piscataway, NJ, USA
{xinyu.li1118, yz593, mengzhu.li, marsic}@rutgers.edu
[2] Division of Trauma and Burn Surgery, Children's National Medical Center, Washington, D.C., USA
{jayang, rburd}@childrensnational.org

ABSTRACT

We propose a Deep Neural Network (DNN) structure for RFID-based activity recognition. RFID data collected from several reader antennas with overlapping coverage have potential spatiotemporal relationships that can be used for object tracking. We augmented the standard fully-connected DNN structure with additional pooling layers to extract the most representative features. For model training and testing, we used RFID data from 12 tagged objects collected during 25 actual trauma resuscitations. Our results showed 76% recognition micro-accuracy for 7 resuscitation activities and 85% average micro-accuracy for 5 resuscitation phases, which is similar to existing system that, however, require the user to wear an RFID antenna.

CCS Concepts

I.5.2 [Pattern Recognition]: Design Methodology–Classifier design & evaluation; C.3 [Special-Purpose and Application-Based Systems]: Real-time and embedded systems.

Keywords

Activity Recognition; RFID; Deep Neural Network; Max Pooling.

1. INTRODUCTION

Mobile/wearable sensors for human activity recognition have been studied for decades, due to their affordability and privacy-preserving property [1][2]. These sensors, however, may not be applicable in many scenarios where they may interfere with work, such as medical settings. Passive RFID has recently been used for activity recognition in medical settings due to its unique advantages that include being battery-free, small size, low-cost, and unobtrusive [3][4]. The general approach used is to place RFID tags on medical tools of interest, extract features from the collected RFID data, use these features to detect object use, and finally decide whether an activity is being performed. The problem with this approach is that environmental changes (such as people movement or changes in the room layout) highly increase the variance of the RFID data. This noise reduces RFID data representativeness and impacts the activity recognition results.

Similar problems exist in the fields of image classification and speech recognition—capturing the same object from a different angle or background produces noise that compromises recognition performance. Recent deep learning approaches have transformed image classification and speech recognition. Early research has shown that deep learning can be successfully applied to recognize activities based on simple physical movement data from mobile phone sensors and accelerometers [6], but established network structure for processing RFID data has not been developed.

S3'16, October 03-07, 2016, New York City, NY, USA
© 2016 ACM. ISBN 978-1-4503-4255-1/16/10…$15.00

DOI: http://dx.doi.org/10.1145/ 2987354.2987355

We propose a DNN structure for recognizing activities from RFID data. The standard DNN uses fully connected hidden layers and lowers the weights of "unimportant" neurons [7], we propose ignoring these neurons because they do not provide important information for classification. Similar to a pooling layer in the Convolutional Neural Network (ConvNet) [5] and the "maxout" network [7], we introduced and implemented max pooling into a DNN to suppress noise in RFID data by selecting the neurons with maximum activation and ignoring ("dropping") the adjacent neurons within each pooling-window. We trained and tested the model with RFID data collected in 25 actual trauma resuscitations. Our results show an average of 76% micro-accuracy for 7 common resuscitation activities and 85% micro-accuracy for 5 resuscitation phases. Our contributions are:

1. Design of a deep learning structure for RFID-based activity recognition from multiple reader antennas.
2. Using the "pooling" concept in DNNs for performance improvement on noisy data.
3. Evaluation of the model with real-world data.

2. METHEDOLOGY

2.1 Data Collection and Pre-Processing

We collected RFID data in a Level 1 trauma center with two Impinj R420 (4 port) readers and eight RFID antennas using *maxmiller* reading mode. As in previous research [3], 12 types of medical objects were tagged for our experiments. We recorded 25 trauma resuscitation cases with more than 50,000 seconds of RFID data and used these data for training and testing purposes.

Raw RFID data were first preprocessed, then fed into the DNN for training and testing. The preprocessing steps are:

1. *Object name lookup:* We maintained a lookup table mapping the passive RFID tag IDs to associated medical objects. Before further processing, each tag ID in the raw RFID data was replaced with the associated object name from the lookup table.

2. *Regularization:* Because successful RFID readouts do not occur at a constant rate, the recorded data must be regularized to a constant rate. We averaged the received signal Strength (RSS) over each one-second interval and inserted zeros where no data were received, resulting in a 1-second sampling time.

3. *Format:* The output of regularization is a matrix with the dimensions of time, object name, and antenna number. We used a time-window to select and format the data. The size of the time window should be a power of two for use in the pooling operation (we selected a 16-second time windows). We sliced and reshaped the time series received by each reader antenna into 16-second vectors, similar to the time window used in sports video classification[8]. The data for each window were formatted as:

[$O_1A_1T_{1:16}$, …, $O_1A_8T_{1:16}$, $O_2A_1T_{1:16}$, …, $O_2A_8T_{1:16}$, …, $O_8A_8T_{1:16}$]

where O_i denotes the object i, A_j denotes the antenna j, and T_k denotes the index of the k^{th} second in the time window.

2.2 Neural Network Structure

Different deep learning structures have been applied to different problems: convolutional neural networks (ConvNets or CNNs) to image classification [5] and DNNs to ambient sound classification [9]. ConvNets do not fit our setting because they use a partially connected network and full weight sharing (using the same kernel over the entire dataset) for feature extraction. RFID data, however, consist of spatiotemporally related points from antennas with overlapping coverage areas and, unlike images, all data points are potentially related with each other. Instead, we chose to use a DNN capable of establishing full connectivity between two adjacent layers. Not all neurons contribute positively to the final classification result, so we did not connect all the neurons in the network's hidden layers. Instead, based on ConvNet's max pooling [5], we introduced pooling layers into the standard DNN to drop neurons with low activations. We used the DNN structure with five fully-connected layers, four pooling layers, and one softmax layer (Figure 1). We designed the network size to be adjustable based on the number of input data points and the number of output labels.

2.3 Pooling the Hidden Layers

Data from mobile sensors or RFID tags are subject to noise, so not all input and hidden layer neurons will contribute positively to final decision-making. We designed our DNN similar to max pooling in ConvNet [5] and the idea of "maxout" [7], to selectively feed-forward the neurons of each layer. We chose to use non-overlapping one-dimensional 2-point windows to perform pooling of RFID data, similar to the 2-by-2 window pooling in ConvNets. Our 2-point windows are pairs of adjacent neurons in a layer. For each 2-point window, the pooling operation selects the neuron with the maximum activation as input for the next layer.

Using the 2-point window method, each pooling layer halves the size of the previous hidden layer. Our pooling operation has the same purpose as the pooing layer in ConvNet: for extracting the most significant features by moving from specific to general, and to filter out data points not important for the final classification. The influence of pooling window size to the system performance is part of our future work.

2.4 Model Training

2.4.1 *Pre-Training*

We pre-trained the connection weights of our DNN with a Restricted Boltzmann Machine (RBM) [10], to initialize the weights and better generalize the DNN. We trained the DNN using a greedy approach [10], one layer at a time, where the output of the previous layer (visible) was used as the input of the next layer (hidden). The weights of each neuron were updated as:

$$\Delta w_{i,j} = \eta \frac{\partial log P(v)}{\partial w_{i,j}} \quad (1)$$

where $w_{i,j}$ denotes the weight between the i^{th} neuron in a visible layer and j^{th} neuron in a hidden layer, η denotes the learning rate, v is a vector of all neurons in the visible layer, and $P(v)$ is a vector of activation probabilities of neurons in the visible layer [11].

2.4.2 *Fine-Tuning*

We divided the collected data into three parts, 60% for fine-tuning, 20% for cross-validation, and 20% for testing. When training the model, we used a sigmoid activation function, because previous research has shown that sigmoid works better for models pre-trained with RBM [11]. We initialized (pre-trained) all the parameters with an RBM before further fine-tuning. We updated the weights using back-propagation: the weights of neurons not selected during pooling were assigned the same weight changes as the selected neuron with maximum activation. To avoid converging the model to a local minimum, we used the momentum method [12] and stochastic gradient descent instead of the usual gradient decent. We set the initial momentum to 0.5. After the error rate stabilized, we increased the momentum to 0.9 [11]. We set 0.01 as the initial learning rate and then adjusted manually throughout the training.

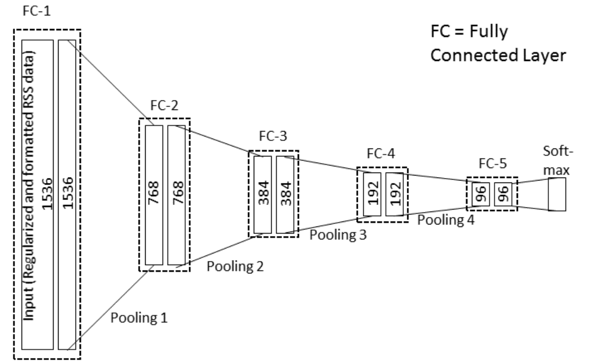

Figure 1. The layers in our DBM structure.

2.4.3 *Preventing Overfitting*

Overfitting is always a problem when training a deep model. To prevent overfitting, we applied "dropout" in all the fully connected layers [13]. In addition, we selectively updated the weights during fine-tuning: when the training error decreased but cross-validation error increased, the weights were not updated.

3. PRELIMINARY-RESULTS

We tagged 12 types of objects associated with seven medical activities in an actual trauma room (Table 1). We collected the RFID data during 25 actual trauma resuscitations. With the help of medical domain experts, we obtained ground truth coding for activities in 16 cases and for process phases in 25 cases. Because training the DNN requires a large amount of data and we had a limited amount of coded data for training, we semi-randomly [3] selected 80% of the data for training (includes fine-tuning and cross-validation), and used the rest for testing. Although we used dropout [13] and cross-validation, overfitting is still possible due to the limited amount of data for training and testing. These problems will be solved when more coded data become available.

Our system was able to achieve an average of 76% micro-accuracy for 7 medical activities and 85% for 5 process phases (Tables 2 and 3). The performance varied widely for different activities. For activities like EC (Table 1), accuracy greater than 80% was achieved, but for others, like BA, the performance was not better than chance. The variable performance can be explained

Table 1. The codes of medical activities used in our analysis.

Activity	Code	Activity	Code
Pulse Oximeter Placement	BA	Ear Exam	EAR
Oxygen Mask Preparation	BC	Warm Sheet Placement	EC
Blood Pressure Measurement	BP	Cardiac Lead Placement	CA
Temperature Measurement	EA		

Table 2. The confusion matrix for 7 medical activities.

	BA	BC	BP	CA	EA	EAR	EC
BA	0.14	0.29	0.29	0.00	0.29	0.00	0.00
BC	0.00	0.85	0.03	0.00	0.00	0.00	0.11
BP	0.00	0.25	0.41	0.00	0.00	0.08	0.25
CA	0.00	0.20	0.00	0.40	0.00	0.10	0.30
EA	0.00	0.31	0.23	0.00	0.31	0.00	0.15
EAR	0.00	0.00	0.20	0.00	0.10	0.30	0.40
EC	0.00	0.14	0.01	0.00	0.01	0.00	0.84

Table 3. The confusion matrix for five phases of the resuscitation process. 1: Pre-arrival, 2: Patient Arrival, 3: Primary Survey, 4: Secondary Survey, and 5: Post-Secondary.

	1	2	3	4	5
1	0.77	0.07	0.02	0.05	0.02
2	0.97	0.03	0.00	0.00	0.00
3	0.05	0.00	0.83	0.07	0.05
4	0.02	0.00	0.07	0.77	0.14
5	0.01	0.00	0.01	0.06	0.86

by the amount of training data. The typical duration for different medical activities and process phases varied: the shortest activity (BA) may take only a few seconds, while a longer activity (EC) may take several minutes. This large range of durations results in very different amounts of training data for different activities obtainable from each case. Longer activities provide more training data and lead to better generalizations in the model. Our results also show that the activities and process phases with longer durations have greater recognition accuracy. Activities and phases such as BA (29 times fewer instances than EC) and the second phase of the process (50 times fewer instances than the fourth or fifth phase) have poor recognition rates.

We also compared our system using two-point pooling window with the ordinary DNN structure. Our results show an average 5% accuracy gain with our DNN using pooling. This result compares well with a result on speech recognition where the use of the "maxout" structure yielded a 2 % gain [7].

4. CONCLUSION AND FUTURE WORK

We present a novel DNN structure for RFID-based activity recognition and presented the preliminary results of implementing the proposed system on data collected in a medical setting. The system structure can be applied in general mobile sensing by replacing the RFID data with data streams from mobile sensors.

The RFID data can be pre-processed and formatted in different ways, and different formats may require different deep learning structures. We preprocessed the RSS into data vectors and directly fed these into a DNN for activity recognition. Another possible approach (inspired by ConvNet speech recognition [14]) would be activity classification based on each RFID tag's time-frequency feature map. If we formatted the RFID data into three dimensions (object-ID, time, antenna-ID), we could also use the 3D ConvNet [15] for activity recognition. The study and comparison of various approaches will be part of our future work.

5. ACKNOWLEDGMENTS

The authors are grateful to Shuhong Chen, Sen Yang, and Moliang Zhou for comments on the paper draft. This research was supported by the National Institutes of Health under Award Number R01LM011834.

6. REFERENCES

[1] Choudhury, Tonmoy, et al. "The mobile sensing platform: An embedded activity recognition system." Pervasive Computing, IEEE 7.2 (2008): 32-41.

[2] Győrbíró, Norbert, Ákos Fábián, and Gergely Hományi. "An activity recognition system for mobile phones." Mobile Networks and Applications 14.1 (2009): 82-91.

[3] Xinyu Li, Dongyang Yao, Xuechao Pan, et.al. "Activity Recognition for Medical Teamwork Based on Passive RFID." RFID (RFID), 2016 IEEE Int'l Conference on, 2016.

[4] Parlak, Siddika, et al. "Design and evaluation of RFID deployments in a trauma resuscitation bay." Biomedical and Health Informatics, IEEE Journal of 18.3 (2014): 1091-1097.

[5] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).

[6] Alsheikh, Mohammad Abu, et al. "Mobile big data analytics using deep learning and Apache Spark." arXiv preprint arXiv:1602.07031 (2016).

[7] Miao, Yajie, Florian Metze, and Shourabh Rawat. "Deep maxout networks for low-resource speech recognition." Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on. IEEE, 2011.

[8] Karpathy, Andrej, et al. "Large-scale video classification with convolutional neural networks." Proceedings of the IEEE Conf. Computer Vision and Pattern Recognition. 2014.

[9] Lane, Nicholas D., Petko Georgiev, and Lorena Qendro. "DeepEar: robust smartphone audio sensing in unconstrained acoustic environments using deep learning." Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing. ACM, 2015.

[10] Goeffrey E. Hinton, Simon Osindero, and Yee Whye Teh. A fast learning algorithm for deep belief net. Neural Computation, 18:1527-1554 2006.

[11] Hinton, Geoffrey E. "A practical guide to training restricted boltzmann machines." Neural Networks: Tricks of the Trade. Springer Berlin Heidelberg, 2012. 599-61.

[12] Wiegerinck, W., Komoda, A., and Heskes, T. Stochastic dynamics of learning with momentum in neural networks. Journal of Physics A: Mathematical and General.

[13] Srivastava, Nitish, et al. "Dropout: A simple way to prevent neural networks from overfitting." The Journal of Machine Learning Research 15.1 (2014): 1929-1958.

[14] Abdel-Hamid, Ossama, et al. "Convolutional neural networks for speech recognition." Audio, Speech, and Language Processing, IEEE/ACM Transact. 22.10 (2014): 1533-1545.

[15] Maturana, Daniel, and Sebastian Scherer. "VoxNet: A 3D convolutional neural network for real-time object recognition." Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on. IEEE, 2015.