Feature Extraction and Learning for RSSI based Indoor Device Localization

Stavros Timotheatos, Grigorios Tsagkatakis, Panagiotis Tsakalides and Panos Trahanias

Institute of Computer Science
Foundation for Research and Technology - Hellas (FORTH)
Heraklion, Crete, Greece

Abstract. In this paper, we study and experimentally compare two state-of-the-art methods for low dimensional feature extraction, within the context of RSSI fingerprinting for localization. On one hand, we consider Stacked Autoencoders, a prominent example of a deep learning architecture, while on the other hand, we explore Random Projections, a universal feature extraction approach. Experimental results suggest that feature learning has a dramatic impact on the subsequent analysis like location based classification.

1 Introduction

To achieve high indoors localization accuracy, machine learning approaches are introduced for learning directly from data such as the Received Signal Strength Indicator (RSSI). Although RSSI values are easy to acquire, without a need for additional hardware, they are subject to shadowing, multipath fading and attenuation by static and dynamic objects. RSSI fingerprinting addresses this problem by creating an RSSI signature map of the physical space, where a training process involves acquiring measurements from all possible spatial locations and exploiting these measurements for localization during run-time. A critical issue that hinders the proliferation of RSSI fingerprinting methods is the need for storage, and at the same time processing of massive amounts of fingerprints is a costly process.

To address this limitation, features that both characterize the data and also reduce their size are thus needed. In this work, we consider two different paradigms for achieving both good localization and reduced storage, namely Deep Learning (DL) [1] and Random Projections (RPs) [2].

On the one hand, DL has led to state-of-the-art performance in a variety of applications, however, to achieve this level of performance, significant computational resources are needed during the training phase. On the other hand, RPs rely on randomly generated projection matrices which enjoy a universality property in that they achieve excellent performance diverse datasets.

In this work, we provide a detailed comparison between DL and RPs for the particular problem of RSSI based indoor location. The key novelties of this work include the application of the Stacked AutoEcoder (SAE) architecture for RSSI fingerprinting, and the comparison between SAE and RPs in terms of localization accuracy as a function of the feature space dimensionality. The key question we address in this paper is whether the intense training required for deep learning based features lead to better localization compared to the universal random projection method. Ideally, we would like to observe very high localization accuracy from very small output space since the dimensionality of the outputs space has a direct effect in both fingerprint storage and runtime matching. We note that very recently, Zhang et al. [3] considered a similar setup, however a different variant of autoencoders was considered, while no discussion of RPs not included.

2 Proposed framework

The localization framework considered in this work consists of two parts, feature extraction and classification. In the following section, we provide a quick overview of DL and RPs, two state-of-the-art methods for feature extraction, as well as the classification methods that are employed for localization.

2.1 Stacked AutoEncoders

An *AutoEncoder* (AE) is an unsupervised learning algorithm which is trained to reproduce the inputs at the output, *i.e.*, $\mathbf{y}^k = \mathbf{x}^k$, where $\mathbf{x}^k, \mathbf{y}^k \in \mathbb{R}^n$. An AE is composed of two parts, the *encoder*, which maps the inputs data $\mathbf{x}^k$ to a lower dimensional space $\mathbf{z}^k$, and the *decoder*, which reconstructs the inputs $\tilde{\mathbf{x}}^k$ for the lower dimensional space data $\mathbf{z}^k$. Formally, the objective of an AE is to estimate the proper weight vectors $\mathbf{W}_i$ and $\mathbf{W}_i^{'}$ which associate units between different layers following the process described by the following equations:

$$\begin{aligned} \mathbf{z}^k &= \mathcal{F}(\mathbf{W}_i \mathbf{x}^k + \mathbf{b}_1) \\ \tilde{\mathbf{x}}^k &= \mathcal{F}(\mathbf{W}_i^{'} \mathbf{z}^k + \mathbf{b}_2) \end{aligned} \tag{1}$$

The function $\mathcal{F}$ is an non-linear function that is introduced in order to increase the learning capability of the AE and in our case, a sigmoid function is used in that respect. Furthermore, for reasons of computational complexity, tied-weights are used, i.e., $\mathbf{W}_i = \mathbf{W}_i^{'}$. For problems of high complexity, a single layer AE does not exhibit sufficient learning capacity. To overcome this limitation, multiple AEs can be stacked together to form a *Stacked AutoEncoder* (SAE). Formally, a SAE corresponds to a cascade of single layer AE, such that the output of each one becomes the input for the subsequent, forming a deep multi-layer encoder and decoder architecture.

To training a SAE, two approaches have been investigated in the literature. In the unsupervised pre-training approach, each layer is independently trained while in back-propagation, a forward-backward optimization algorithm is employed. The back-propagation algorithm performs iterative passes over the network, updating the weights through a stochastic gradient descent algorithm, with the following objective function:

$$\mathbf{J}(\mathbf{W}_i, \mathbf{b}_i, \mathbf{b}_{i+1}) = \sum_{m}^{k=1} (\mathbf{W}_i(\mathbf{W}_i \mathbf{x}^k + \mathbf{b}_1) + \mathbf{b}_2 - \mathbf{x}^k)^2 \tag{2}$$

For classification problems, one can use a SAE as a dimensionality reduction technique, where the outputs of the last encoder are used as input features for the classification, where in our case, the entropy loss function was used. For the SoftMax classifier, the back-propagation algorithm was appropriately adapted in order to tune the SAE for the particular supervised learning setup.

2.2 Random Projections

While SAE require extensive training in order to achieve good performance, an alternative form of feature extraction, Random Projections, relies on *universal* dimensionality reduction for achieving the same goal, without requiring training data. The core concept of Random projections is the JohnsonLindenstrauss lemma [4], which states that given $0 \leqslant \epsilon \leqslant 1$, then for every point-set P of cardinality n in $\mathcal{R}^d$, there exists a Lipchitz function $\mathcal{F} : \mathcal{R}^d \rightarrow \mathcal{R}^k$, where $k \geqslant O(\epsilon^{-2}\log(n))$ such that:

$$(1-\epsilon)\|\mathbf{x}^i - \mathbf{x}^j\|^2 \leqslant \|\mathcal{L}(\mathbf{x}^i) - \mathcal{L}(\mathbf{x}^j)\|^2 \leqslant (1+\epsilon)\|\mathbf{x}^i - \mathbf{x}^j\|^2 \tag{3}$$

An example of such a random projections operator $\mathcal{L}$ is a linear projection through a projection matrix $\mathbf{M}$ whose elements are randomly drawn such that

$$m_{ij} = \begin{cases} +1 & \text{with probability } 50\% \\ -1 & \text{with probability } 50\% \end{cases} \tag{4}$$

To perform dimensionality reduction, the input data are linearly mapped to the features according to $\mathbf{z}^k = \mathbf{M}\mathbf{x}^k$.

2.3 Classification

Once appropriate low-dimensional features are extracted from the RSSI fingerprints, a classifier is applied to estimate the system pose during run-time. This is achieved with the comparison of the pose produced from training NN model and the ground truth pose labels. Two classifiers were considered in this work, the *K-Nearest Neighbour* (K-NN) and the *SoftMax* (SM).

The K-NN is applied to a training set of observations, which belong to different classes. To predict the label of a new observation, K-NN measures its similarity of the training examples, and select the label associated with the majority of the k-closest training example. In our work, we employ the 1-NN where the label of the closest neighbor is propagated to the new observation.

The second classifier is the SM classifier, a multi-label generalization of logistic regression. For the SM classification, a cost function is required, and the the cross-entropy is utilized here. Unlike K-NN, SM can be realized as an additional layer introduced at the output of the deepest SAE encoder. As such, back-propagation can be applied to the full feature extraction and classification pipeline, leading to greater accuracy as it will be demonstrated.

3 Experimental results

3.1 Dataset and Setup

To evaluate the performance of the feature extraction and localization schemes, the UJIIndoorLoc dataset [4] is employed, where each location is mapped to a 520-dimensional RSSI fingerprint. In all cases, a set of 10501 fingerprints from four floors of buildings A and B are used for training and a set of 842 fingerprints are used for testing.

3.2 Performance of Stacked AutoEncoders

First, we investigate the capabilities of SAE as a low dimensional feature extraction method for localization. We consider SAE architecture with SM classification in terms of building and floor localization, for two and three hidden layers. Specifically, Tables 1 and 2 consider a two-layer SAE with 300 and 100 hidden units, without and with back-propagation respectively, while the results reported in Table 3 assume a three layer, $300-200-100$, structure of hidden units. For all three scenarios, we consider the SM classifier. The motivations for this experiment are (i) quantify the effects of back-propagation and (ii) quantify the value of introducing deeper architectures, when the same number of output features are extracted.

Predicted \ Ground Truth								
A0	78.0	12.9	1.2	0.0	0.0	0.0	0.0	0.0
A1	17.9	71.8	7.2	1.1	0.0	0.0	0.0	0.0
A2	2.5	14.9	69.8	23.0	0.0	0.0	0.0	0.0
A3	1.2	0.4	21.8	75.4	0.0	0.0	0.0	0.0
B0	0.0	0.0	0.0	0.0	70.1	9.7	0.0	0.0
B1	0.0	0.0	0.0	0.0	20.0	61.1	0.0	0.0
B2	0.0	0.0	0.0	0.0	3.3	18.8	66.7	10.6
B3	0.0	0.0	0.0	0.0	6.6	10.4	33.3	89.4

Table 1: Confusion matrix for the two-layers SAE feed-forward only architecture. Overall accuracy is **71.0**%.

Regarding the impact of back-propagation, it is clear from Tables 1 and 2 that there is a significant increase in performance. Furthremore, we also observe that miss-classifications are also most closely clustered in the case of back-propagation, leading to a miss-labeling in neighboring flours. As far as the depth of the SAE is concerned, comparing Tables 2 and 3, we observe that introducing additional hidden layers did not lead to better modeling of the RSSI data, while deep architectures induce additional computational complexity.

3.3 Comparative Analysis

Figure 1, illustrates the comparison results for both feed-forward SAE and back-propagation SAE method of all two and three layer architectures, as a function

Predicted								
A0	84.7	8.6	3.0	1.1	0.0	0.0	0.0	0.0
A1	12.8	77.5	3.6	0.0	0.0	0.0	0.0	0.0
A2	2.5	13.9	81.9	17.6	0.0	0.0	0.0	0.0
A3	0.0	0.0	11.5	81.3	0.0	0.0	0.0	0.0
B0	0.0	0.0	0.0	0.0	73.4	7.6	0.0	0.0
B1	0.0	0.0	0.0	0.0	13.3	63.2	2.2	0.0
B2	0.0	0.0	0.0	0.0	3.3	23.7	71.4	6.3
B3	0.0	0.0	0.0	0.0	10.0	5.5	26.4	93.7
	Ground Truth							

Table 2: Confusion matrix for the two-layer SAE architecture with back-propagation. Overall accuracy is **77.4**%.

Predicted								
A0	82.1	11.0.0	1.2	0.0	0.0	0.0	0.0	0.0
A1	14.1	77.5	4.2	0.0	0.0	0.0	0.0	0.0
A2	3.8	11.5	81.3	17.6	0.0	0.0	0.0	0.0
A3	0.0	0.0	13.3	82.4	0.0	0.0	0.0	0.0
B0	0.0	0.0	0.0	0.0	70.1	6.9	0.0	0.0
B1	0.0	0.0	0.0	0.0	20.0.0	63.9	0.0	0.0
B2	0.0	0.0	0.0	0.0	3.3	20.2	75.9	4.2
B3	0.0	0.0	0.0	0.0	6.6	9.0.0	24.1	95.8
	Ground Truth							

Table 3: Confusion matrix for the three-layer SAE architecture with back-propagation and SM classification. Overall accuracy is **77.0**%.

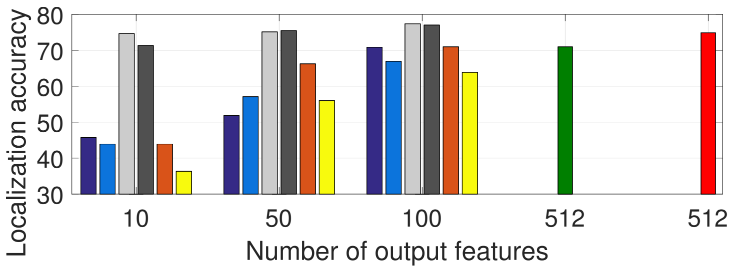

Fig. 1: Comparison of localization accuracy of SAE and Random Projections with 10, 50, and 100 output features, as well the localization of the 1-NN and the SM applied on the raw 512-dimensional input measurements.

of the output layer size. We consider 10, 50 and 100 output sizes and report the performance of two layers feed-forward SAE (dark blue), three layers feed-forward SAE (cyan), two layer SAE with back-prorogation (light grey) and three layer SAE with back-propagation (dark grey). In addition to the SAE, Figure 1 also presents the performance achieved by RPs and 1-NN classification (orange bars) and RPs with SM (yellow bars). As a benchmark, we also report the performance of 1-NN (green) and SM (red) using the original inputs as features.

The average accuracy achieved for the two layer SAE with feed-forward are 45.7%, 52%, and 71%, while for the three layer are 44%, 57.1% and 67% for the different output sizes. For the same architectures, introducing the back-propagation lead to 74.6%, 75.1%, and 77.4% for the two layers and 71.3%, 75.5% and 77% for three layers respectively. The experimental results on SAE suggest that for the feed-forward approach, increasing the deep of the SAE network can actually lead to worse performance compared to more shallow ones. Introducing back-propagation immediately increases the performance of the localization, especially when the output sizes are extremely restricted, e.g 40% increase in performance for 10 outputs units. It is safe then to argue that back-propagation, and the fine-tuning it offers, is critical for high performing systems.

For the RPs reported in Figure 1, the average achieved accuracy is 43.9%, 66.2% and 70.9% for the 1-NN classifier and 36.3%, 56% and 63.8% for the SM. Results suggest that increasing the dimensionality of the feature space has a positive impact, however, the performance is inferior compared to learning based SAE architectures. In 1-NN and SM cases the accuracy is 71% and 74.7% respectively. While in the case of RPs feature extraction has a negative impact on accuracy, for the case of SAE, much higher performance is achieved compared to the raw input.

4 Conclusions

In this work, we compared Stacked Autoencoders and Random Projection for extracting low dimensional features from RSSI fingerprints, which were subsequently considered for indoor localization, posed as a supervised classification problem. Experimental results demonstrate that the Stacked Autoeconders can achieve similar or higher performance compared to raw feature based localization schemes, using significant lower dimensional features.

References

[1] Yoshua Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127, 2009.

[2] Dimitris Achlioptas. Database-friendly random projections: Johnson-lindenstrauss with binary coins. *J. of computer and System Sciences*, 66(4):671–687, 2003.

[3] Wei Zhang, Kan Liu, Weidong Zhang, Youmei Zhang, and Jason Gu. Deep neural networks for wireless localization in indoor and outdoor environments. *Neurocomputing*, 194:279–287, 2016.

[4] Joaquín Torres-Sospedra, Raúl Montoliu, Adolfo Martínez-Usó, Joan P Avariento, Tomás J Arnau, Mauri Benedito-Bordonau, and Joaquín Huerta. Ujiindoorloc: A new multi-building and multi-floor database for wlan fingerprint-based indoor localization problems. *IPIN*, pages 261–270, 2014.