Building/Floor Classification and Location Estimation using WIFI signals

Yupeng Cao

EEE413 Coursework

yupeng.cao13@student.xjtlu.edu.cn

December 14, 2017

Overview

1. Introduction
   - Background
   - Related Work
   - Motivation
2. Methodologies
   - Stacked Autoencoder (SAE)
   - Training Processing
3. Results and Discussions
4. Conclusions
   - Future Work
   - Learning Points

Background

Indoor Localization is a challenging task and there exist no universal solution for all possible applications.

External infrastructure is efficient for localization in limited areas. In large buildings the most precise agent's pose estimates are obtained with laser scanners, however laser scabbers are expensive.

WiFi information can be exploited to provide rough, global position estimates, without additional costs of exteroceptive sensors.

Related Work

WiFi Ranging

the properties of WiFi signal wave are exploited to directly estimate the distance to the access points.

WiFi Fingerprinting

Focus on efficiently comparing achieved WiFi scans to the prerecorded database of scans inside building and thus are more robust to local signal disturbances.

k nearest neighbors

k scans that are the most similar to the analyzed WiFi scan are queried from the database and their positions are averaged to achieve the position estimate.

Motivations

Those solutions are difficult to tune in a case of a larger building and if a large amount of data is available.

However, the growing amount of available date is caused by the popularity if mobile devices equipped with WiFi adapters.Therefore, machine learning approaches are a promising solution due to less parameter tuning and better scalability in larger environments.GPU processing capabilities improving also can rise the performance.

This project propose DNN approach for floor and building classification and location.

- We propose a classification DNN with autoencoders to predict floor and building base on a single scan.

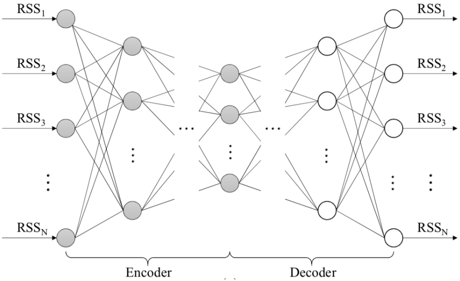

Figure: SAE Model

Methodologies

- The already pre-trained encoder part is connected to classifier. The number in parentheses represent the number of neurons in the layer.

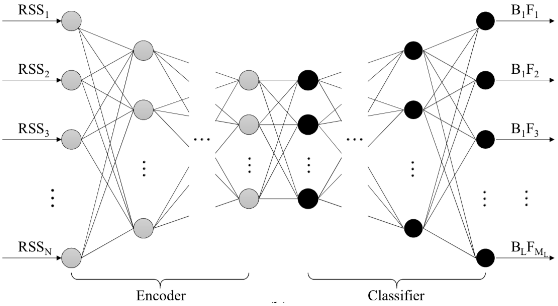

Figure: Training Process

Methodologies

- Pre-Processing Data
- SAE: reduce the redundancy data.
- Training
- Classifier
- Testing

Testing Steps

Key parameters

1. Branch Size and Epoch times
2. SAE layer and Classifier layer
3. Classifier Activation

Before we modify the Training Model, we want to try to adjust some key parameters to improve the accuracy first. Based on coding, we choose those parameters from three aspects to test.

Results and Discussions

Batch Size	Epoch	Accuracy
10	20	0.5984
200	200	0.7137
256	200	0.7459
256	1000	0.0004

Table: Change the parameter of Branch Size and Epoch

Results and Discussions

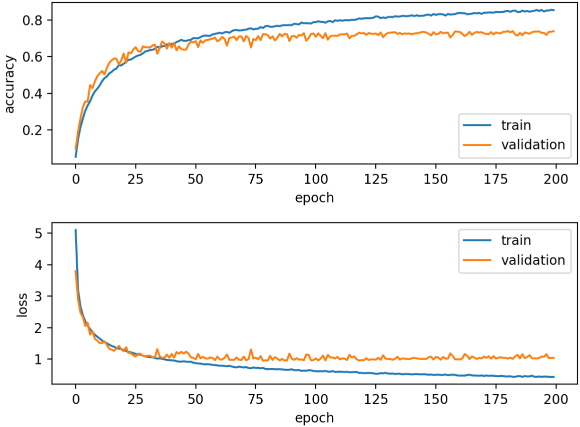

Figure: The accuracy carve with epoch times increasing

Results and Discussions

Based on the Branch Size and Epoch result, the next step is trying to modify the number of the layer of classifier.

SAE Layer	Classifier Layer	Accuracy
256-128-64-128-256	64-128-256-512	0.7359
256-128-64-32-64-128-256	64-128-256-512	0.62322
256-128-256	128-256-512	0.7489
256-128-256	256-512	0.7554

Table: Comparing the performance in different layer

Finally, we didn't change the layers.

Results and Discussions

Classifier Activation is also an important parameter, therefore we try to optimize it.

Classifier Activation	Results
relu	-
elu	+0.02
softmax	-0.18
selu	+0.02
softplus	-0.20
softsign	-0.15
linear	-
sigmod	-0.40

Table: Comparing the performance in different Classifier Activation

Results and Discussions

What's more, we have some interesting result when we try to improve the performance by changing classifier loss.

Classifier Loss	**Results**
categorical crossentropy	-
binary crossentropy	0.998

Table: Comparing the performance in different Classifier Loss

Results and Discussions

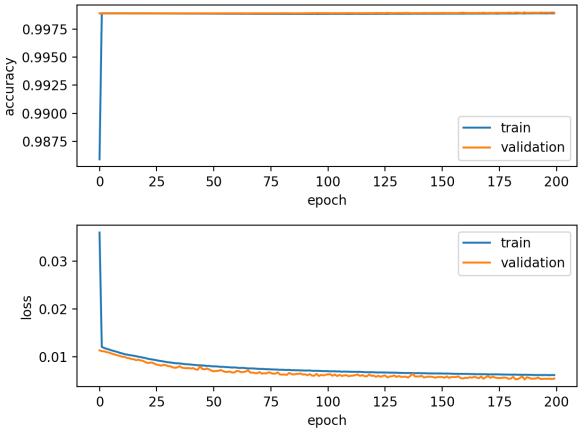

Figure: The accuracy carve with binary crossentropy

Settle down the final parameter setting

Parameters	**Value**
Ratio of training data	0.70
Ratio of validation data	0.20
Ratio of test data	0.10
Epochs	200
Batch Size	256
SAE hidden layers	256-128-64-128-256
Classifier layers	128-256-512
Classifier activation	elu

Table: The parameter setting

Results and Discussions

However, improving the performance by modifying the parameters is limitation.

Understanding the model and training process is more importantly. Although we try to do some works to modify the model and training process, we failed it.

```
model.add(Dense(units, activation=CLASSIFIER_ACTIVATION,
history = model.fit(x_train, y_train, validation_data=(x_
```

Figure: We focus on those coding

Future Work

- Totally understanding the model and code
- Try to modify the model
- Adding kNN into whole process
- Increasing the number of the data

Learning Points

- Good starting point in Artificial Intelligence
- Learning some basic knowledge about Neural Network and Deep Learning
- Combine the telecommunication and AI.
- Time Management!

References

Michal Nowicki and Jan Wietrzykowski (Apr,2017)
Low-effort place recognition with WiFi fingerprints using deep learning
arXiv:1611.02049v2 [Online].Available: https://arxiv.org/abs/1611.02049v2

Kyeong Soo Kim, Ruihao Wang,et al. (Oct,2017)
Large-Scale Location-Aware Services in Access: Hierarchical Building/Floor Classification and Location Estimation using Wi-Fi Fingerprinting Based on Deep Neural Networks.
arXiv:1710.00951v3 [Online].Available: https://arxiv.org/abs/1710.00951

The End