Deep triplet network adopting the kernel and the range space learning for Wi-Fi signature verification

YoungWoong KWON[1], JooYoung KIM[1], and Kar-Ann TOH[1]

[1] Yonsei University
[2] lncs@springer.com
http://www.springer.com/gp/computer-science/lncs
[3] ABC Institute, Rupert-Karls-University Heidelberg, Heidelberg, Germany
{abc,lncs}@uni-heidelberg.de

Abstract. The abstract should briefly summarize the contents of the paper in 15–250 words.

Keywords: First keyword · Second keyword · Another keyword.

1 Introduction

1.1 Motivation

i. Pros of In-Air WIFI CSI signature system 1) Cheap: Use commercial device 2) Easy: No additional devices is needed 3) Secure: Hard to forgery ii. Cons of In-Air WIFI CSI signature system 1) Setting direction problem a) Different direction -> Different feature is needed b) Hard to set exactly same direction as authentication before Size of signature can varies

1.2 Contribution

- Overcome cons of WIFI signature system - Robust to signal direction, size

2 Related Works

2.1 WIFI CSI

- An In-Air Signature Verification System Using Wi-Fi Signals

CSI captures signal strength and phase information for OFDM subcarriers and between each pair of transmit-receive antennas. It runs on a commodity 802.11n NIC, and records Channel State Information (CSI) based on the 802.11 standard. The CSI contains information about the channel between sender and receiver at the level of individual data subcarriers, for each pair of transmit and receive antennas.

In a frequency domain, the CSI of sub-carrier **c** between transmitter(Tx) and receiver(Rx) can be modeled as $R_c = \mathbf{H}_c T_c + N$ where the R_c and T_c denote the received and the transmitted signal vector of dimension r and t, respectively. The N is the additive channel noise and $\mathbf{H}_c$ is the $r \times t$ channel matrix. The CSI of sub-carrier c can be modeled as follows:

$$h_c = \mid h_c \mid e^{\angle \theta}, \tag{1}$$

where $\mid h_c \mid$ and θ represent the amplitude and the phase of the sub-carrier, respectively.

2.2 the kernel and the range space learning

Multilayer feedforward neural networks has been widely used to feature extractor and classifier. For traning the feedforward multilayer networks, non-gradient learning method by using linear matrix equation has invented recently. [6]

This model avoids the problems of the deep learning model such as vanishing gradient and local minima. And except for the number of layers and neurons used in the network, there are few parameters for the network configuration.

More recently, a new learning framework has been developed to train the weights of the deep networks by a series of kernel and range space manipulations [5, 4]. This method allows to quickly learn the weights of the deep network with little system resources. we adopted this method to mining negative samples for the anker signal.

3 Proposed System

In this section, we propose a direction-free identify verification system based on the Wi-Fi based in-air handwritten signature (will be called Wi-Fi signature signals hereafter). An overview of the proposed system utilizing the ConvNet [1] and the kernel and the range space projection learning (KAR space learning) is shown in Fig.1. Essentially, the Wi-Fi signature signals are preprocessed to create the input data for our network. Subsequently, the training dataset is moved to KAR space learning. it mines the hard positve and negative samples for given anchor signal from training dataset whtout using gradient. ConvNet structure extract feature vectors from given triplet using convnet filters. ConvNet training is achived by triplet loss, which is L2 distance of given feature vectors of the triplet. The following subsections detail the proposed method.

3.1 Triplet loss

Our proposed networks utilizes the L_2 distance to calculate the triplet loss. For the i_{th} signal as anchor input $\mathbf{I}_{i,anc}$, positive input $\mathbf{I}_{i,pos}$ is belong to the same class for anchor input while negative input $\mathbf{I}_{i,neg}$ is from another class. Let $\mathbf{x}_{i,anc} \in \mathrm{R}^{d\times 1}$, $\mathbf{x}_{i,pos} \in \mathrm{R}^{d\times 1}$ and $\mathbf{x}_{i,neg} \in \mathrm{R}^{d\times 1}$ be three feature vectors

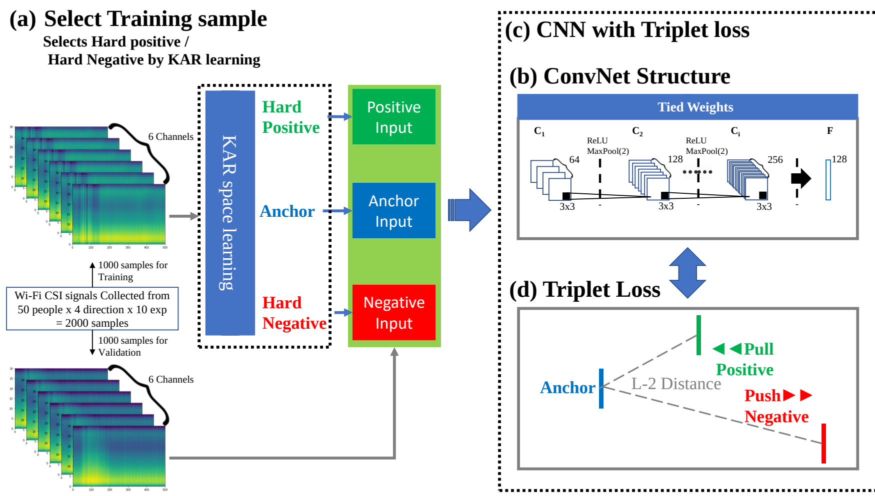

Fig. 1. Structure of the network

extracted from the ConvNet structure. Then the loss for the triplet inputs can be calculated as follows:

$$loss = \sum_{i}^{N} \left[\|\mathbf{x}_{i,anc} - \mathbf{x}_{i,pos}\|_2^2 - \|\mathbf{x}_{i,anc} - \mathbf{x}_{i,neg}\|_2^2 + \alpha \right]_+ , \tag{2}$$

where α is the margin to make distance between positive and negative pairs, N is size of the mini-batch. + sign means greater between zero or calculated loss.

3.2 Triplet mining by the kernel and the range space learning

in [3], when training triplet loss networks, it is important to take hard positive and hard negative sample for faster convergence of the loss function.

since we don't know which is the hard sample before training entire network, we make multilayer feedforward neural network to select the hard positive and the hard negative. we adopted gradient-free, the kernel and the range (KAR) space projection learning to train multilayer feedforward neural network. [4, 5].

first, multilayer neural network structure is shown below:

$$\mathbf{G} = \sigma\left(\left[\mathbf{1}, \sigma\left(\ldots\left[\mathbf{1}, \sigma\left(\left[\mathbf{1}, \sigma\left(\mathbf{X}\mathbf{W}_1\right)\right]\mathbf{W}_2\right)\right]\ldots\mathbf{W}_{(i-1)}\right)\right]\mathbf{W}_i\right), \tag{3}$$

let the training dataset $\mathbf{X} \in \mathrm{R}^{m\times(n+1)}$ and $\mathbf{G} \in \mathrm{R}^{m\times n}$ is network outputs.

network learning is archived using one-hot encoded target $\mathbf{Y} \in \mathrm{R}^{m\times n}$ instead of $\mathbf{G}$.

the weight matrix $\mathbf{W}_1 \ldots \mathbf{W}_i$ can be separated into weights and bias term, $\mathbf{W}_2 \ldots \mathbf{W}_i = \begin{bmatrix} \mathbf{w}_2^T \\ W_2 \end{bmatrix} \ldots \begin{bmatrix} \mathbf{w}_i^T \\ W_i \end{bmatrix}$.

where $\mathbf{W}_1 \in \mathrm{R}^{(n+1)\times h_1}$,$\mathbf{W}_2 \in \mathrm{R}^{(h_1+1)\times h_2}$,$\ldots$, $\mathbf{W}_i \in \mathrm{R}^{(h_{(i-1)}+1)\times n}$, $\mathbf{1} = [1,\ldots,1]^T \in \mathrm{R}^{m\times 1}$ and $\sigma(.)$ is activation function.

after assign random weights to $\mathbf{W}_1 \ldots \mathbf{W}_i$, we can get $\mathbf{W}_1$. the equation can be solved as follows:

$$\left[\sigma^{-1}\left(\mathbf{Y}\right) - \mathbf{1}\cdot\mathbf{w}_i^T\right] W_i^{\dagger} = \sigma\left(\ldots\left[\mathbf{1},\sigma\left(\left[\mathbf{1},\sigma\left(\mathbf{X}\mathbf{W}_1\right)\right]\mathbf{W}_2\right)\right]\ldots\mathbf{W}_{(i-1)}\right) \tag{4}$$

$$\begin{aligned}\Rightarrow \Bigg[\sigma^{-1}\Bigg(\ldots\Bigg[\sigma^{-1}\Bigg(\left[\sigma^{-1}\left(\mathbf{Y}\right) - \mathbf{1}\cdot\mathbf{w}_i^T\right] W_i^{\dagger}\Bigg) - \mathbf{1}\cdot\mathbf{w}_{(i-1)}^T\Bigg] W_{(i-1)}^{\dagger}\cdots\Bigg) \\ -\mathbf{1}\cdot\mathbf{w}_2^T\Bigg] W_2^{\dagger} = \sigma\left(\mathbf{X}\mathbf{W}_1\right)\end{aligned} \tag{5}$$

$$\begin{aligned}\Rightarrow \mathbf{X}^{\dagger}\sigma^{-1}\Bigg(\Bigg[\sigma^{-1}\Bigg(\ldots\Bigg[\sigma^{-1}\Bigg(\left[\sigma^{-1}\left(\mathbf{Y}\right) - \mathbf{1}\cdot\mathbf{w}_i^T\right] W_i^{\dagger}\Bigg) - \mathbf{1}\cdot\mathbf{w}_{(i-1)}^T\Bigg] W_{(i-1)}^{\dagger}\cdots\Bigg) \\ -\mathbf{1}\cdot\mathbf{w}_2^T\Bigg] W_2^{\dagger} = \mathbf{W}_1\end{aligned} \tag{6}$$

after getting $\mathbf{W}_1$, $\mathbf{W}_2$ can also be optimized

$$\begin{aligned}\Rightarrow \left(\sigma\left(\mathbf{X}\mathbf{W}_1\right)\right)^{\dagger}\Bigg(\ldots\Bigg[\sigma^{-1}\Bigg(\left[\sigma^{-1}\left(\mathbf{Y}\right) - \mathbf{1}\cdot\mathbf{w}_i^T\right] W_i^{\dagger}\Bigg) - \mathbf{1}\cdot\mathbf{w}_{(i-1)}^T\Bigg] W_{(i-1)}^{\dagger}\cdots\Bigg) \\ = \mathbf{W}_2\end{aligned} \tag{7}$$

Repeat this process recursively until all weight maxtrix values are obtained. finally, $\mathbf{W}_i$ can be obtained as follows:

$$\mathbf{W}_i = \left[\mathbf{1},\sigma\left(\ldots\left[\mathbf{1},\sigma\left(\left[\mathbf{1},\sigma\left(\mathbf{X}\mathbf{W}_1\right)\right]\mathbf{W}_2\right)\right]\ldots\mathbf{W}_{(i-1)}\right)\right]^{\dagger}\sigma^{-1}\left(\mathbf{Y}\right), \tag{8}$$

when anker signal is set, negative samples can be mined if the distance of network output $\mathbf{G}$ is above the threshold.

3.3 ConvNet Structures

To design the proposed networks, we firstly need to select the feature extracting networks which convert the input data into a vector. In this work, we utilize the ConvNet structure [1] as a feature extractor since the three-dimensional data format of our preprocessed input signal can be regarded as an image data format with multiple channels.

Our ConvNet structure (See Fig 1 (b)) for the network consists of i convolutional layers $\mathbf{C}_i$ and one fully-connected layer $\mathbf{F}$. The number of convolutional filters to be trained in each layer is empirically chosen as $\{64, 128, ..., 2^{6+i}\}$, with fixed filter size of 3×3 and stride of 1. The Rectified Linear (ReLU) function as an activation function and the max-pooling layers are applied between each convolutional layers. The features from the last convolutional layer are directly flattened into a single vector without activation function and the Max-pooling layer followed by the fully-connected layer.

4 Experiments

Dataset To evaluate validation performance of proposed system, Wi-Fi CSI signature dataset from [2] was used. Since every Wi-Fi signature signal has different data size, we firstly adopted the gradient operation with respect to the time instance to measure the short time energy. Data points with the highest short-time energy within the time period are then manually selected as the starting and the ending points of the in-air signature action. Subsequently, the Fast Fourier Transform based re-sampling method [2] is implemented to unify the length of the signals. As a result, three-dimensional Wi-Fi signature signals with unified data size are obtained as the input of the ConvNet structure in the network. we utilized 2000 Wi-Fi CSI signature signals (4 directions $\times$ 50 identities $\times$ 10 samples) which is dimension of (500 packets $\times$ 30 subcarriers $\times$ 6 antennas).

ConvNet structure We impose a triplet loss objective on our classifier. This objective is combined with standard backpropagation algorithm, where the gradient is additive across the twin networks due to the tied weights. We initialized all network weights in the convolutional layers from a normal distribution with zero-mean and a standard deviation of $10{-}2$. Biases were also initialized from a normal distribution, but with mean 0.5 and standard deviation $10{-}2$.

5 Conclusion

References

1. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
2. Moon, H.C., Jang, S.I., Oh, K., Toh, K.A.: An in-air signature verification system using wi-fi signals. In: Proceedings of the 2017 4th International Conference on Biomedical and Bioinformatics Engineering. pp. 133–138. ACM (2017)
3. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 815–823 (2015)
4. Toh, K.A.: Learning from the kernel and the range space. In: 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS). pp. 1–6. IEEE (2018)
5. Toh, K.A., Lin, Z., Li, Z., Oh, B., Sun, L.: Gradient-free learning based on the kernel and the range space. arXiv preprint arXiv:1810.11581 (2018)
6. Wang, J., Guo, P., Xin, X.: Review of pseudoinverse learning algorithm for multilayer neural networks and applications. In: International Symposium on Neural Networks. pp. 99–106. Springer (2018)