

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

YoungWoong KWON¹, Jooyoung Kim¹, and Kar-Ann Toh¹

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

1.2 Contribution

2 Related Works

2.1 ConvNets with triplet input

ConvNets with triplet input is widely used in face recognition [5] and person re-id [2, 1]. The distinction between classes may be ambiguous in the above tasks. This is because the targets may be looking in different directions or taking different positions. Triplet ConvNets works by extract feature vectors which expand margins between same class and different class, so it has good ability to distinguish between classes.

Loss function depends on the distance between each triplet samples, loss can be down to 0, when negative class distance is much more greater than positive distance. also, outliers affects a lot to training. so it is important to optimize the training process when traning triplet ConvNets.

There are many ways to optimize training. in [5], they selects appropriate samples from large mini-batch before each traning iteration. in [2], they adopt second margin to triplet loss and expand intra-class distance. in [1], they made quadruplet loss, which makes the distance between the same classes, regardless of class, is less than the distance between the different classes.

in this paper, we adopt multilayer feedforward networks, trained by the kernel and the range space learning to select good samples for training triplet ConvNets. it has advantage for reducing computing resources, speed up and optimize training.

2.2 the kernel and the range space learning

Multilayer feedforward neural networks has been widely used to feature extractor and classifier. For training the feedforward multilayer networks, non-gradient learning method by using linear matrix equation has invented recently. [8]

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 number of neurons in the network, no more parameters are needed 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 projection method [6, 7]. 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 training dataset.

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 [3] and the kernel and the range space projection learning (KAR space learning) [6, 7] is shown in Fig.1.

Essentially, the Wi-Fi signature signals are separated to create the training and validation data. Subsequently, KAR space projection learning generates triplet data by mining the hard positive and hard negative samples for given anchor signal from training dataset.

For training ConvNet structure, generated triplet data flows into ConvNet structure. ConvNet extracts feature vectors from given triplet. ConvNet training is achieved by triplet loss, which is formulated by the L-2 distance metrics of feature vectors. ConvNet network is trained by backpropagation. The following subsections detail the proposed method.

3.1 Triplet loss

Triplet loss is used to train ConvNet to extract feature vector better from triplet input. For the i_{th} anchor input signal $\mathbf{X}_{anc,i}$, triplet input is generated by grouping positive input signal $\mathbf{X}_{pos,i}$, which drawn from the same identity for anchor and negative input signal $\mathbf{X}_{neg,i}$, chosen from another identity for the anchor. Generated triplet input $\{\mathbf{X}_{anc,i}, \mathbf{X}_{pos,i}, \mathbf{X}_{neg,i}\}$ is fed into the ConvNet structure one by one and makes anchor, positive and negative feature vectors.

Let $\mathbf{v}_{anc,i} \in \mathbb{R}^{d \times 1}$, $\mathbf{v}_{pos,i} \in \mathbb{R}^{d \times 1}$ and $\mathbf{v}_{neg,i} \in \mathbb{R}^{d \times 1}$ be three feature vectors extracted from the ConvNet structure.

triplet loss is calculated by comparing positive distance(L-2 distance from anchor to positive feature vector)and negative distance(L-2 distance from anchor to negative feature vector). ConvNet is trained to maximize gap between positive distance to negative distance more than margin α .

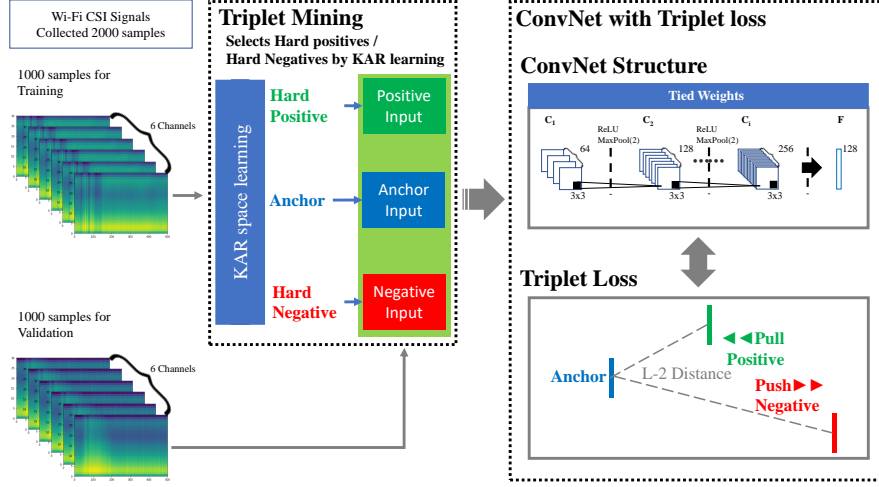


Fig. 1. Structure of the network

The triplet loss for the inputs can be calculated as follows:

$$loss = \sum_i^N \max \left(\left[\|\mathbf{v}_{anc,i} - \mathbf{v}_{pos,i}\|_2^2 - \|\mathbf{v}_{anc,i} - \mathbf{v}_{neg,i}\|_2^2 + \alpha \right], 0 \right) \quad (1)$$

where N is size of the mini-batch. note that triplet loss can down to 0 when negative distance is longer than positive distance more than margin α . if we select proper triplet for training, we can avoid this problem and make convergence of the loss function faster. This can be done by selecting hard triplet.

3.2 Triplet mining by the kernel and the range space learning

Regarding to [5], it is important to take hard positive and hard negative sample for faster convergence of the loss when training triplet networks. hard positive sample means distance between feature vectors from positive signal to anchor signal is the greatest, while hard negative sample means distance between feature vectors from negative signal to anchor signal is the smallest.

since we don't know which is the hard sample before training the network, we train multilayer feedforward neural network to mine the hard positive and the hard negative from training dataset. For training multilayer neural network, we adopted gradient-free the kernel and the range (KAR) space projection learning. [6, 7].

Let the training dataset $\mathbf{X} \in \mathbb{R}^{m \times (n+1)}$ and $\mathbf{G} \in \mathbb{R}^{m \times n}$ is network outputs. Multilayer neural network structure is shown below:

$$\mathbf{G} = \sigma \left([\mathbf{1}, \sigma (\dots [\mathbf{1}, \sigma ([\mathbf{1}, \sigma (\mathbf{X}\mathbf{W}_1)] \mathbf{W}_2)] \dots \mathbf{W}_{(i-1)})] \mathbf{W}_i \right), \quad (2)$$

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

when all weights have been learned, for given anker signal, hard negative samples can be mined among the distance of network output of desired signal \mathbf{G} is below the threshold.

Network learning is archived using one-hot encoded target $\mathbf{Y} \in \mathbb{R}^{m \times n}$ instead of network output \mathbf{G} . After that, we have the weight matrix $\mathbf{W}_1 \dots \mathbf{W}_i$ to train. the weight matrix can be separated into weights and bias term as $\mathbf{W}_2 \dots \mathbf{W}_i = \begin{bmatrix} \mathbf{w}_2^T \\ W_2 \end{bmatrix} \dots \begin{bmatrix} \mathbf{w}_i^T \\ W_i \end{bmatrix}$.

after assign random weights to $\mathbf{W}_1 \dots \mathbf{W}_i$, we get \mathbf{W}_1 . it is solved as follows:

$$[\sigma^{-1}(\mathbf{Y}) - \mathbf{1} \cdot \mathbf{w}_i^T] W_i^\dagger = \sigma (\dots [\mathbf{1}, \sigma ([\mathbf{1}, \sigma (\mathbf{X}\mathbf{W}_1)] \mathbf{W}_2)] \dots \mathbf{W}_{(i-1)}) \quad (3)$$

$$\Rightarrow \left[\sigma^{-1} \left(\dots \left[\sigma^{-1} \left([\sigma^{-1}(\mathbf{Y}) - \mathbf{1} \cdot \mathbf{w}_i^T] W_i^\dagger \right) - \mathbf{1} \cdot \mathbf{w}_{(i-1)}^T \right] W_{(i-1)}^\dagger \dots \right) - \mathbf{1} \cdot \mathbf{w}_2^T \right] W_2^\dagger = \sigma (\mathbf{X}\mathbf{W}_1) \quad (4)$$

$$\Rightarrow \mathbf{X}^\dagger \sigma^{-1} \left(\left[\sigma^{-1} \left(\dots \left[\sigma^{-1} \left([\sigma^{-1}(\mathbf{Y}) - \mathbf{1} \cdot \mathbf{w}_i^T] W_i^\dagger \right) - \mathbf{1} \cdot \mathbf{w}_{(i-1)}^T \right] W_{(i-1)}^\dagger \dots \right) - \mathbf{1} \cdot \mathbf{w}_2^T \right] W_2^\dagger \right) = \mathbf{W}_1 \quad (5)$$

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

$$\Rightarrow (\sigma (\mathbf{X}\mathbf{W}_1))^\dagger \left(\dots \left[\sigma^{-1} \left([\sigma^{-1}(\mathbf{Y}) - \mathbf{1} \cdot \mathbf{w}_i^T] W_i^\dagger \right) - \mathbf{1} \cdot \mathbf{w}_{(i-1)}^T \right] W_{(i-1)}^\dagger \dots \right) = \mathbf{W}_2 \quad (6)$$

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

$$\mathbf{W}_i = [\mathbf{1}, \sigma (\dots [\mathbf{1}, \sigma ([\mathbf{1}, \sigma (\mathbf{X}\mathbf{W}_1)] \mathbf{W}_2)] \dots \mathbf{W}_{(i-1)})]^\dagger \sigma^{-1}(\mathbf{Y}), \quad (7)$$

3.3 ConvNet Structures

To design the proposed networks, we firstly need to select the feature extracting networks which convert the triplet data into a feature vector. In this work, we utilize the ConvNet structure [3] 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 $\{64, 128, \dots, 2^{6+i}\}$, 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. we used sigmoid activation at ConvNet output and L-2 normalize output vectors to prevent negative distance when calculating triplet loss.

4 Experiments

Dataset To evaluate validation performance of proposed system, Wi-Fi CSI signature dataset from [4] 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 [4] 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 ($4 \text{ directions} \times 50 \text{ identities} \times 10 \text{ samples}$) which is dimension of ($500 \text{ packets} \times 30 \text{ subcarriers} \times 6 \text{ antennas}$).

ConvNet structure We impose a triplet loss objective on our classifier. This objective is combined with standard backpropagation algorithm. We initialized all network weights in the convolutional layers from a normal distribution with zero-mean and a standard deviation of 0.01. Biases were also initialized from a normal distribution, but with mean 0.5 and standard deviation 0.01. all layers are L-2 regularized using parameters of 0.0002. for output layer, regularization is 0.001

For KARnet trained multilayer neural networks, we set 2 layer and number of neurons are [1024,128]. initialized as uniform distribution over $[0, 1)$. activation and inverse activation functions are $[\arctan, \tan]$.

Results

Training Efficiency

5 Conclusion

References

1. Chen, W., Chen, X., Zhang, J., Huang, K.: Beyond triplet loss: a deep quadruplet network for person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 403–412 (2017)
2. Cheng, D., Gong, Y., Zhou, S., Wang, J., Zheng, N.: Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1335–1344 (2016)
3. 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)
4. 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)
5. 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)
6. 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)
7. 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)
8. 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)