Signal Processing 155 (2019) 268–280

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Target-Aware Recurrent Attentional Network for Radar HRRP Target Recognition

Bin Xu[a,b], Bo Chen[a,b,*], Jinwei Wan[a,b], Hongwei Liu[a,b], Lin Jin[a,b]

[a] National Laboratory of Radar Signal Processing, Xidian University, Xian 710071, PR China
[b] Collaborative Innovation Center of Information Sensing and Understanding at Xidian University, PR China

ARTICLE INFO

Article history:
Received 28 February 2018
Revised 25 July 2018
Accepted 28 September 2018
Available online 5 October 2018

Keywords:
Attention mechanism
Radar Automatic Target Recognition (RATR)
Recurrent Neural Network (RNN)
Spectrogram analysis

ABSTRACT

In this paper, we develop a Target-Aware Recurrent Attentional Network (TARAN) for Radar Automatic Target Recognition (RATR) based on High-Resolution Range Profile (HRRP) to make use of the temporal dependence and find the informative areas in HRRP, since it reflects the distribution of scatterers in target along the range dimension. Specifically, we utilize RNN to explore the sequential relationship between the range cells within a HRRP sample and employ the attention mechanism to weight up each timestep in the hidden state so as to discover the target area, which is more discriminative and informative. Effectiveness and efficiency are evaluated on the measured data. Compared with traditional methods, besides the competitive recognition performance, TARAN is also more robust to time-shift sensitivity thanks to the memory function of RNN and attention mechanism. Furthermore, detailed analysis of TARAN model are provided based on time domain and spectrogram features.

1. Introduction

A High-Resolution Range Profile (HRRP) is composed of the amplitude of the coherent summations of the complex returns from target scatterers in each range cell, which represents the projection of the complex returned echoes from the target scattering centers onto the radar Line-Of-Sight (LOS), as shown in Fig. 1. Since it contains abundant discriminative information, such as target size, scatterer distribution, radar HRRP target recognition has received intensive attention from the Radar Automatic Target Recognition (RATR) community [1–19].

For the RATR task, feature extraction is a critical step [20]. Liao et al. [21] extract invariant features based on the integration of the bispectra of range profiles. Du et al. [22] further investigate the recognition methods based on high-order spectra features. In [23], the feasibility of classifying aerial targets using the micro-Doppler signatures is studied, where the features are computed in the form of bicoherence estimates and cepstral coefficients. Although those algorithms based on engineered features are effective for recognition, the features are hand-crafted and rely on the personal experiences.

In recent years, deep learning algorithms have been explored for RATR. The autoencoder models have been improved to extract robust features for HRRP [13], which employ the average profile of each HRRP frame as the correction term to establish the cost function under the Mahalanobis distance criterion, and achieve better recognition performance compared with traditional Stacked Auto-Encoders (SAE) [24]. To model the aspect sectors with few HRRP data, Pan et al. [25] utilize the Deep Belief Network (DBN) to learn discriminative features by sharing latent information of HRRP data globally. However, those networks do not explore the temporal dependence between range cells, although they record the structure information of targets.

Several approaches have been proposed to describe the temporal dependence in HRRP. Pan et al. [15] transform the HRRP sample into a sequence and model the structure across range cells by the Hidden Markov Models (HMM) with a transition probability. To model the HRRP sequence, [4] describes the spatial structure across range by the hidden Markov structure and models the temporal dependence between HRRP samples by the time evolution of the transition probabilities. Instead of utilizing the HMM for RATR, it is possible to employ the Recurrent Neural Network (RNN) which exploits larger state-space and richer dynamics compared to HMM [26]. RNN is a class of artificial neural network with connections between hidden units, which allows the model to exhibit dynamic temporal behavior in the data [27]. It achieves state of the art performance on sequential data for different tasks, such as

* Corresponding author at: Department of Electronic Engineering, No.2 South TaiBai Road, XI'AN 710071, PR China.
E-mail address: bchen@mail.xidian.edu.cn (B. Chen).

https://doi.org/10.1016/j.sigpro.2018.09.041
0165-1684/

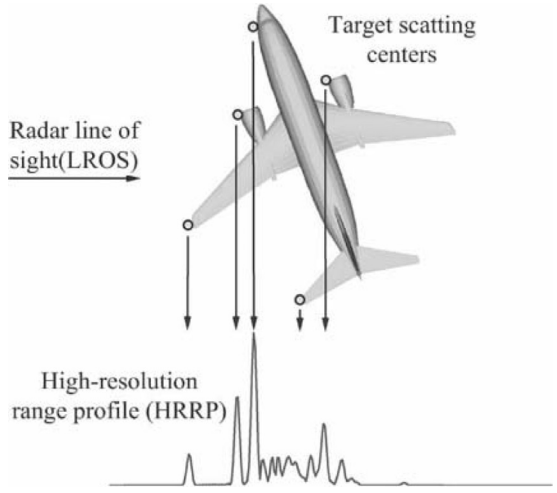

Fig. 1. Illustration of an HRRP sample from a plane target, where the circles on the plane represent some scatterers. This figure is cited from [30].

language modeling [28], speech recognition [26], machine translation [29] and so on.

In this paper, we take advantage of the temporal dependence within each HRRP sequence to use RNN with voting strategy to learn its latent representation for recognition task, where each HRRP is divided into multiple overlapping sequential feature as input. Furthermore, we integrate the basic formulation of RNN with an attention mechanism to reduce the influence of the noise areas and automatically catch the discriminative target areas in HRRP, called Target-Aware Recurrent Attentional Network (TARAN). The label for a HRRP sample is decided based on a weighted sum of the output from each hidden state computed by a multilayer perceptron. Therefore, different from the conventional feature selection methods [31,32], TARAN aims at evaluating the importance of input at each timestep for recognition rather than individual features and the learned corresponding weights are data-dependent rather than fixed for all of data, which may have different lengths for different HRRP samples.

Inspired by [33], besides the HRRP features in time domain, we also try spectrogram features as input, which is a two-dimensional feature providing the variation of frequency domain with time domain.

The remainder of the paper is organized as follows. We formulate the RATR task with RNNs in Section 2. The paper introduces the proposed architecture, i.e., TARAN for radar HRRP target recognition in Section 3. The parameter setting and detailed experimental results based on measured data are made in Section 4, followed by conclusions in Section 5.

2. Problem formulation

We formulate the problem in a generic manner, and consider classification of an object into one of K distinct classes based on a D-dimensional data measurement of that object. Specifically, given a set of N input signals $X = [x(1), \cdots, x(N)] \in \mathbb{R}^{D\times N}$, where X is the training data set containing N input signals, D is the dimensionality of each signal.

Denote $y = [y(1), \cdots, y(N)] \in \mathbb{R}^{K\times N}$, where $y(n)$ denotes the 1-of-K representation of the label of $x(n)$, i.e., $y(n) \in \{0, 1\}^K$, $y_k(n) = 1$ if the label of $x(n)$ is k, and $y_i(n) = 0, \forall i \neq k$. Let $L(x, y; w) : \mathbb{R}^D \times \mathbb{R}^K \mapsto \mathbb{R}$ be a cost function which encodes the problem that we are interested in solving. For the HRRP-based RATR, the classes are the types of targets and the data measurements are the radar echoes returned from the target.

Suppose there are radial displacement and little uniform rotation for a target. The transmitted signal is $s(t)e^{jw_c t}$ where $s(t)$ is the complex envelope and w_c is the carrier angular frequency. According to the scattering center model [34], the nth complex returned echo from the lth range cell ($l = 1, 2, \cdots, L$) in baseband is

$$x_l(t, n) = \sum_{i=1}^{V_l} \sigma_{li} s(t - \frac{2R_{li}(n)}{c}) e^{-j[\frac{4\pi}{\lambda} R_{li}(n) - \theta_{li0}]} \tag{1}$$

where V_l denotes the number of target scatterers in the lth range cell, λ represents the radar wavelength, σ_{li} and θ_{li0} denote the strength and the initial phase of the ith scatterer in the lth range cell, and $R_{li}(n)$ denotes the radial distance between the radar and the ith scatterer in the lth range cell of the nth returned echo. Thus, the nth complex HRRP sample is

$$x(n) = [x_1(n), x_2(n), \cdots, x_D(n)]^{\mathrm{T}} \tag{2}$$

For the RATR task, given an input signal $x(n)$, and its label y, we can write the likelihood as

$$L(X, Y, w) = \sum_{i=1}^{N} \log p(y(i)|x(i); \theta), \tag{3}$$

which is parameterized by θ. During the training procedure, the model would like to learn those parameters by maximizing the likelihood

$$\theta^* = \arg\max_{\theta} \sum L(X, Y, w). \tag{4}$$

Some statistical methods such as Adaptive Gaussian Classifier (AGC) [13] and Maximum Correlation Coefficient (MCC) [35] replace the label likelihood $P(y|x;\ \theta)$ by the class-conditional data likelihood $P(x|y;\ \theta)$ according to Bayes rule and build the model with the assumption that the echoes from different range cells are independent

$$L = \prod_{i=1}^{N} \prod_{j=1}^{D} p(x_j(i)|y(i); \theta) \tag{5}$$

In fact, there exists the temporal dependence across the range cells. According to the scattering center model, when the target rotates, the scatterers from different range cells rotate in the same way, causing the scatterers distance differences varying with different range resolution cells correspondingly. Moreover, the radial distance $R_{li}(n)$ changes and some scatterers run into other range cells, called the multiple reflection phenomenon, which also makes the echoes from different range cells dependent [36].

As mentioned above, HMM has been proposed to describe the temporal dependence across the range cells with the transition matrix

$$p(x, s|y; \theta) = p(s_1|\pi) \prod_{t=1}^{T-1} p(s_t|s_{t-1}; A) \prod_{t=1}^{T} p(x_t|s_t; \phi) \tag{6}$$

where s denotes the latent states, π denotes the initial state probability, A represents the state transition distribution and ϕ are a set of parameters governing the emission probability. However, the model assumes that the probability of each observation $x_t(n)$ depends only on the current state s_t, which makes contextual effects difficult to model. Besides, as the generative model, it has to build the model and train their parameters for each class $p(x|y;\ \theta)$ which does not fully use the label information and has heavy computational burden.

In this paper, we focus on the family of Recurrent Neural Networks (RNNs) to produce the posterior probability $p(y|x;\ \theta)$. The models parameterize the posterior probability by a neural network and incorporate the context information including the historical states by the connection weight between hidden states to

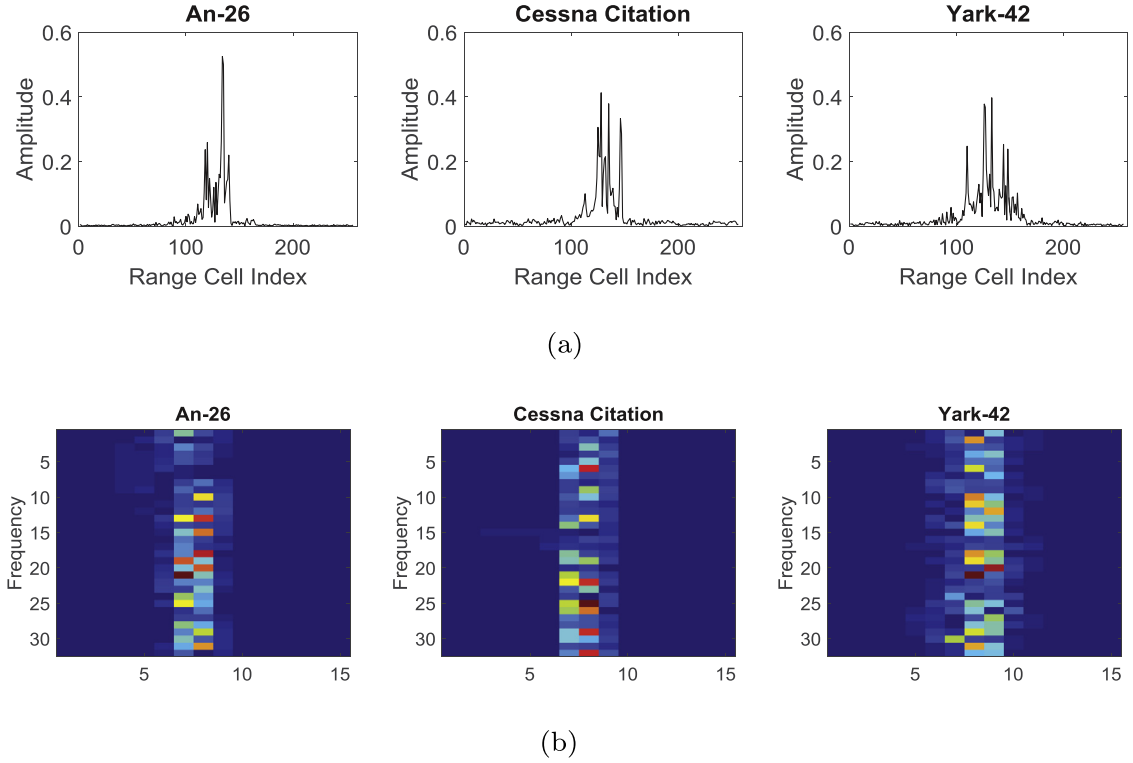

Fig. 2. Measured HRRP examples of time domain and corresponding spectrogram for three airplanes. (a) Time domain (b) Spectrogram.

describe the temporal dependence [37]. Meanwhile, as the supervised model, it generally gives better performance in classification tasks compared with the unsupervised methods.

3. Proposed model

To apply the RNN model, the input signals need to be first converted into sequential features. In this section, we consider two kinds of sequential features i.e. time sequential feature and spectrogram feature.

3.1. Input representation

3.1.1. Time sequential feature

According to [10], the nth real HRRP can be defined as

$$\hat{x}(n) = [|x_1(n)|, \cdots, |x_D(n)|]^{\mathrm{T}} \tag{7}$$

where T denotes the transpose operation and $x_l(n)$ denotes the lth range cell in nth sample. The time domain features of HRRP are shown in Fig. 2(a).

The time sequential feature can be transformed from the real HRRP sample as:

$$x_t = \hat{x}[1 + (d-b) * (t-1) : d + (d-b) * (t-1)] \tag{8}$$

where x_t denotes the input of sequence at timestep t, d represents the length of window, b denotes the length of overlap of the window.

3.1.2. Spectrogram feature

To further utilize the frequency information, we transform the original HRRP to the spectrogram representation. It is the view of frequency versus time domain and can be calculated by the Short-Time Fourier Transform (STFT) algorithm as

$$X(\tau, \omega) = \sum_{l=-\infty}^{\infty} x_l w(l-\tau) e^{-j\omega l} \tag{9}$$

where $w(l-\tau)$ is the window function that moves τ units to the right from the function $w(l)$ and x_l denotes the lth range cell in the complex signal sample in (2). Specifically, we use the Discrete Fourier Transformation (DFT) for each time signal obtained by the Hamming window function. Thus, the feature at timestep t of the spectrogram sequential feature is

$$\bar{x}_t(k) = \sum_{l=1}^{d} \tilde{x}_l e^{-j\frac{2\pi k}{d}l}; 0 \le k \le d-1 \tag{10}$$

where

$$\tilde{x}_l = x_l w(l-\tau); \tau = (d-b)(t-1) \tag{11}$$

$\bar{x}_t(k)$ is the kth frequency component in $\bar{x}_t$, d denotes the width of the window function, b is the overlap of the window. We utilize the power of the spectrogram as the input representation

$$x_t(k) = |\bar{x}_t(k)|^2 \tag{12}$$

Different from the time sequential feature, the spectrogram feature considers both of the amplitude of echoes and the phase information between different scatterers with Fourier transformation instead of only using the signal amplitude which may result in less information loss. Furthermore, the received echo is a strong function of the target aspect orientation, which means that a little variation of the target aspect will lead to the scatterers escaping from a range cell to another. The spectrogram takes several continuous range cells into consideration with a window function and is more robust to the variation of target aspect than the scatterers in a single range cell. Therefore, the sequential relationship across the timesteps in spectrogram can reflect the target physical composition more robustly [18].

Fig. 2 presents the real HRRPs and corresponding spectrogram representations of three airplane targets.

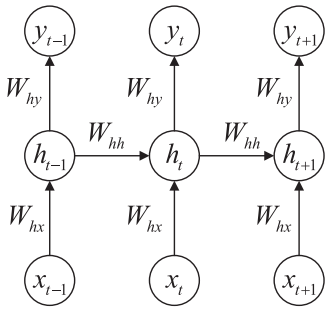

Fig. 3. A Recurrent Neural Network is a very deep feedforward neural network whose weights are shared across time. The nonlinear activation function used by the hidden units is the source of the RNNs rich dynamics.

3.2. Recurrent neural network with the voting strategy (RNNvot)

In order to consider the temporal correlation between the HRRP range cells, we adopt the RNN for the target recognition. However, distinct from the conventional sequence-to-sequence task addressed by RNN, in HRRP ATR, each sequential inputs from one HRRP sample share the single label. To deal with this issue, we integrate the RNN model with the voting strategy to combine the information at all timesteps and predict the label for each HRRP, called RNNvot. The RNN architecture is shown in Fig. 3, where it is unrolled across time to cover three consecutive inputs of HRRP sequence. The architecture includes an input layer on the bottom, a hidden layer in the middle, and an output layer at top. Given the HRRP sequence $x = [x_1, \cdots, x_T]$, the model computes the hidden vector sequence by iterating the following equation from $t = 1$ to T:

$$h_t = f(W_{hx}x_t + W_{hh}h_{t-1}) \tag{13}$$

where $W_{hx} \in \mathbb{R}^{m\times d}$ is the input-hidden weight matrix and $W_{hh} \in \mathbb{R}^{m\times\ m}$ denotes the matrix connects the hidden states, m is the dimensionality of hidden state. f refers to the sigmoid function which operates on each elements. The label vector sequence $\hat{y} = [\hat{y_1}, \cdots, \hat{y_T}]$ is calculated by soft-max function as follows:

$$p(y_{tk}|x_1, \ldots, x_t, \theta) = \frac{\exp(W_k h_t)}{\sum_{l=1}^{K} \exp(W_l h_t)} \tag{14}$$

where W_k denotes the kth row of the matrix $W_{hy} \in \mathbb{R}^{K\times m}$ and $\hat{y_t} = (y_{t1}, \cdots, y_{tK})^{\mathrm{T}}$. Note that the output y_{tk} is the probability of the current input x_t classified to label k and $\sum_k y_{tk} = 1$.

Given the label vector for different timesteps from the same HRRP sequence, we employ the voting strategy to combine them and predict the final label.

$$k^* = \arg\max_k (q_1, \cdots, q_K) \tag{15}$$

where k^* is the output label by RNN corresponding to the input sample x and q_k the count of the output sequence belonging to the label k

$$q_k = \sum_{t=1}^{T} f(r_t, k) \tag{16}$$

$$f(r_t, k) = \begin{cases} 1, & r_t = k. \\ 0, & r_t \neq k. \end{cases} \tag{17}$$

where r_t denotes the label of input at timestep t

$$r_t = \arg\max_k (y_{t1}, \cdots, y_{tK}) \tag{18}$$

We can also utilize the conventional RNN for the RATR task by outputting the label based on the hidden state at the last timestep, which is called Recurrent Neural Network encoder (RNNenc) [38]. By contrast, the RNNvot takes into account the labels from all timesteps to vote one label for the single HRRP sample, which also relaxes the requirement for memory capability of RNN.

3.3. Target-aware recurrent attentional network

As discussed above, we can modify RNN to process each HRRP sequential feature for the RATR task by voting strategy, while, this method assumes that the input at each timestep x_t takes the same contribution to the recognition task. As we know that the HRRP sequence, besides the target signal, also includes the noise area and moreover in target area the different range cells have different scatterers which may correspond to different parts of the target and has different discriminative information. To address those issues, we design a target-aware recurrent attentional network, which assigns the hidden feature at each timestep a weight automatically to figure out the useful and discriminative target areas and outputs the label with the weighted hidden features.

Fig. 4 is the overview of our proposed model. It consists of a latent representation encoder layer, which extracts the temporal discriminative features from the input sequences, and an attention mechanism responsible for measuring the contribution of the input at each timestep to the target recognition.

3.3.1. Latent representation encoder

The goal of the encoder in our model is to efficiently extract the discriminative feature and the temporal correlation between range cells in the HRRP sample. Given the input sample, our model dose not take the whole HRRP as input but instead convert it into a sequence feature and utilize an RNN as the encoder to learn the hidden states, which has been proven to be an effective technique for sequential data [39,40].

As shown in Fig. 4, the encoder is comprised of an input sequence and a hidden sequence. The hidden sequence is calculated according to Eq. (13). In the encoder, the hidden state h_t is updated not only with the input at the current timestep x_t, but also with the previous hidden state h_{t-1}. This allows the hidden states to deliver information, enabling the model to extract more discriminative features. Meanwhile, by considering the temporal correlation of HRRP, this dynamic model represents the data more accurately.

3.3.2. Attention mechanism

As mentioned above, the target areas reflect the structure information of the targets and useful for the recognition task. Consequently, the goal of the attention mechanism is to help the model focus on the discriminative target areas by assigning larger weights to their outputs in the final decision.

As shown in Fig. 4, the attention mechanism contains the weight coefficient vector a, the context vector C and the label vector y. Due to the tendency of RNNs to better represent recent inputs, the feature h_t focuses on the input around x_t. In order to allow the attention weight a to evaluate the contribution of hidden states at each timestep, we adopt a multilayer perceptron to calculate the attention weight based on the hidden states.

$$e_t = U_a^{\mathrm{T}} \tanh(W_a h_t) \tag{19}$$

$$a_t = \frac{\exp(e_t)}{\sum_{l=1}^{T} \exp(e_l)} \tag{20}$$

where $U_a \in \mathbb{R}^{1\times n}, W_a \in \mathbb{R}^{n\times m}$ are the matrices in the attention mechanism and e_t the coefficient that scores how well the input x_t matches with the output label y. It is worth to mention that, in contrast to the conventional feature selection methods [41], the weights, e_t, calculated by TARAN is a data-dependent function even for test data rather than fixed and the corresponding length of selected features may also vary with different data, which makes the

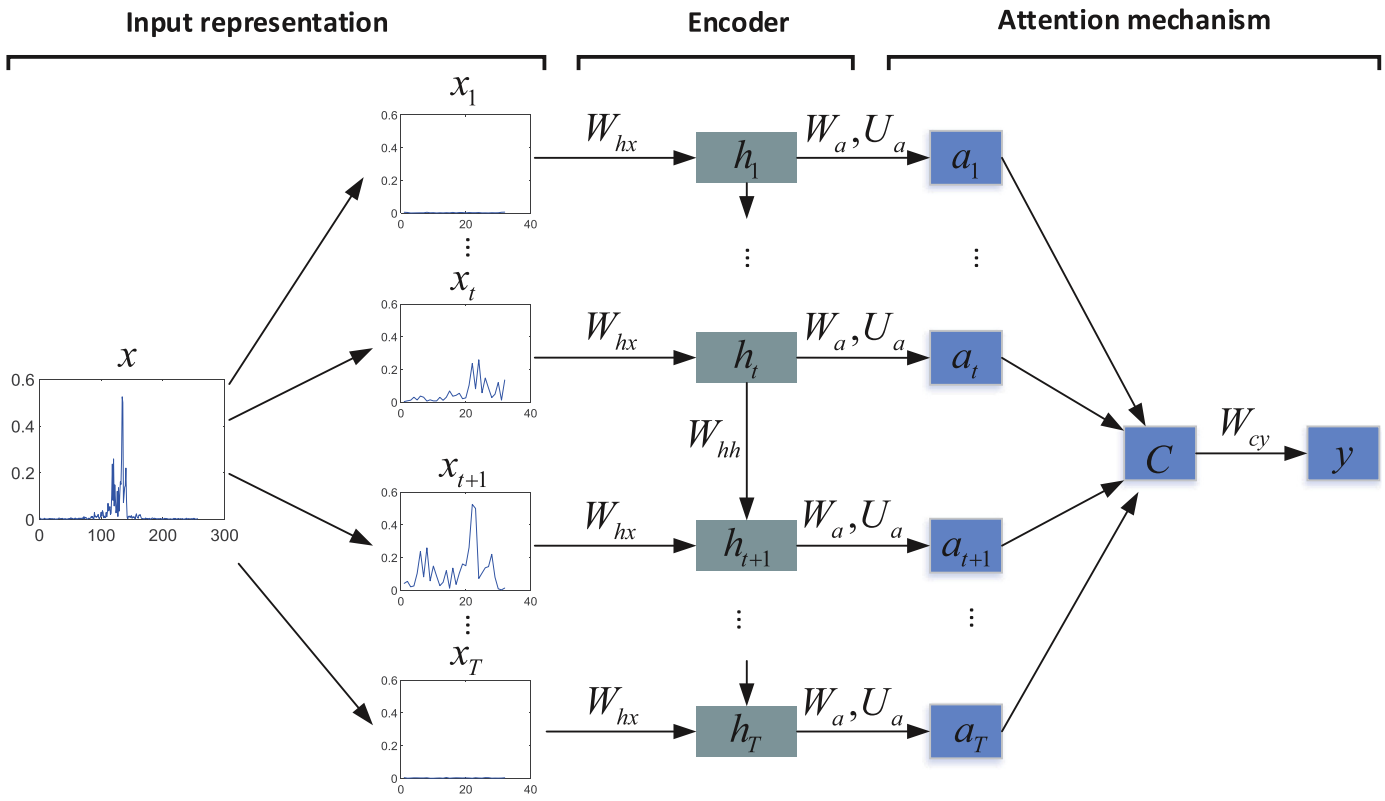

Fig. 4. The graphical illustration of the proposed model trying to generate the target label y given an HRRP sample x.

model more flexible and practical. The value of a_t means the importance of the input x_t to output label y. Intuitively, the coefficient a corresponding to the target areas should be large while those noisy areas should have small coefficients. The context vector C is calculated by summing the hidden states at each timestep with the corresponding weight coefficient.

$$C = \sum_{t=1}^{T} a_t h_t \tag{21}$$

The context vector integrates the information at all timesteps according to the discrimination of the hidden state at each timestep. Given the context vector, we adopt the conventional softmax function to predict the label vector of the input sample.

$$y_k = softmax(W_{cy}C) = \frac{\exp(W_k C)}{\sum_{l=1}^{K} \exp(W_l C)} \tag{22}$$

where W_k denotes the kth row of the matrix $W_{cy} \in \mathbb{R}^{K\times m}$ and the label vector $y = (y_1, \cdots, y_k, \cdots, y_K)^{\mathrm{T}}$.

The TARAN model is parameterized by $\theta = \{W_{hh} \in \mathbb{R}^{m\times m}, W_{hx} \in \mathbb{R}^{m\times d}, W_{cy} \in \mathbb{R}^{K\times m}, W_a \in \mathbb{R}^{1\times n}, U_a \in \mathbb{R}^{n\times m}\}$ and we jointly train the components of the model, the encoder and the attention mechanism in an end-to-end style. Through integrating the attention mechanism with encoder for HRRP ATR, our model, TARAN, is able to automatically locate the target areas, which is more principled than those using Signal-to-Noise Ratio (SNR) to cut the target by heuristic threshold strategy [42].

3.3.3. Training procedure

The process of HRRP target recognition with TARAN model contains two stages, i.e., training stage and test stage. In the training stage, we use the training data to learn the parameters by calculating the gradient of the cost function with respect to the parameters and fix the learned parameters when the model converges. In the test stage, the test dataset are directly mapped using Eq. (13) to determine their labels via Eq. (22).

Therefore, the cost function L is defined as the cross entropy [43].

$$L = -\sum_{n=1}^{N}\sum_{k=1}^{K} z_k(n) \log p(y_k(n)|x(n);\theta) \tag{23}$$

where $z_k(n)$ is the kth element in the real label vector $z(n) \in \mathbb{R}^{K\times 1}$ corresponding to the signal $x(n)$ and $p(y_k(n)|x(n);\theta)$ denotes the probability that the sample x belongs to class k. We utilize the stochastic gradient descent (SGD) method to optimize the proposed TARAN model [44].

$$\theta^{i+1} = \theta^i - \epsilon_i \sum_{j=1}^{N_1} \nabla_\theta L_j \tag{24}$$

where $\theta = \{W_a, U_a, W_{cy}, W_{hh}, W_{hx}\}$, ϵ_i is learning rate, N_1 denotes the number of batch samples and $\nabla_\theta L_j$ is the gradient of the jth cost function to the parameters.

The gradient of L with respect to the parameters in the attention mechanism $\theta_1 = \{W_a, U_a, W_{cy}\}$ can be calculated by first recursively computing the gradient of the cost function with respect to the attention weight e_t according to the following recursion

$$\frac{dL}{de_t} = \frac{da}{de_t}\frac{dC}{da}\frac{dL}{dC} = a_t\left\{\frac{dL}{da_t} - \sum_{j=1}^{T}\left\{a_j \frac{dL}{da_j}\right\}\right\} \tag{25}$$

$$\frac{dL}{da_t} = h_t\{W_{cy}^{\mathrm{T}}(y - z)\} \tag{26}$$

where $\frac{dL}{da} = [\frac{dL}{da_1}, \cdots, \frac{dL}{da_T}]$ and then apply the chain rule to obtain

$$\nabla_{\theta_1} L = \sum_{t=1}^{T} \frac{de_t}{d\theta_1}\frac{dL}{de_t} \tag{27}$$

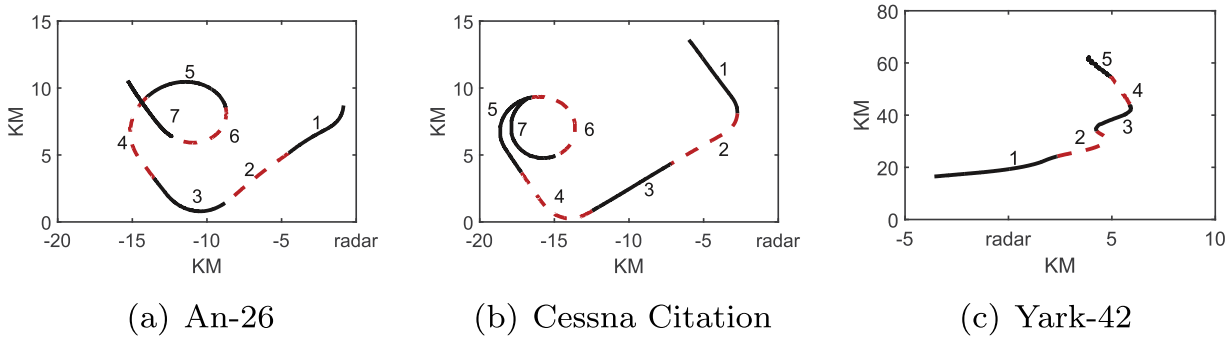

Fig. 5. Projections of target trajectories onto the ground plane: (a) An-26. (b) Cessna Citation. (c) Yark-42.

Table 1
Parameters of planes and radar in the ISAR experiment.

Radar parameters	Center Frequency	5520MHZ	
	bandwidth	400MHZ	
Planes	Length(m)	Width(m)	Height(m)
An-26	23.80	29.20	8.58
Cessna Citation	14.40	15.90	4.57
Yark-42	36.38	34.88	9.83

Different from the parameters in attention mechanism, the parameters in the encoder $\theta_2 = \{W_{hh}, W_{hx}\}$ compress the sequence. We compute the gradients of L with respect to the θ_2 by first computing the gradient of hidden states $dh = [dh_1, \cdots, dh_T]$

$$\frac{dL}{dh_t} = \frac{dh_{t+1}}{dh_t}\frac{dL}{dh_{t+1}} + \frac{dC}{dh_t}\frac{dL}{dC} \tag{28}$$

where $\frac{dL}{dh} = [\frac{dL}{dh_1}, \cdots, \frac{dL}{dh_T}]$. Thus, the derivative of θ_2 can be obtained by

$$\nabla_{\theta_2} L = \sum_{t=1}^{T} \frac{dh_t}{d\theta_2}\frac{dL}{dh_t} \tag{29}$$

This is an instance of Backpropagation Through Time (BPTT), which is a well known algorithm for training recurrent neural networks [45]. We train the encoder and attention mechanism jointly. According to [46], we can control the training procedure by adjusting the hyperparameters in the SGD method so that the TARAN model could converge quickly. The direction diagram for calculating the gradients of parameters and the variation diagram of the cost function versus the training epoch are shown in appendix.

In summary, the overall framework is illustrated in Algorithm 1.

Algorithm 1 Target-Aware Recurrent Attentional Network (TARAN).

Input: Dataset $\{X, Y\}$; cost function $L(f(x), y)$; initialize parameters $\theta = \{W_{hx}, W_{cy}, W_{hh}, W_a, U_a\}$ randomly; learning rate l_r;

repeat
  i = i+1;
  Sample $\{x, y\}$ uniform randomly from $\{X, Y\}$
  initial $h_0 = 0$;
  Calculate the attention weight $a = [a_1, \cdots, a_T]$
  Calculate the output label distribution $P(y|x; \theta^i)$ via (19)
  Computing the stochastic gradient of parameters: $\nabla_{\theta_1} L \leftarrow \sum_{t=1}^{T} \frac{de_t}{d\theta_1} a_t \{\frac{dL}{da_t} - \sum_{j=1}^{T}\{a_j \frac{dL}{da_j}\}\}$
  $\nabla_{\theta_2} L \leftarrow \sum_{t=1}^{T} \frac{dh_t}{d\theta_2}\{\frac{dh_{t+1}}{dh_t}\frac{dL}{dh_{t+1}} + \frac{dC}{dh_t}\frac{dL}{dC}\}$
  Update $\theta^{i+1} = \theta^i + l_r \nabla_\theta L$
until Converge

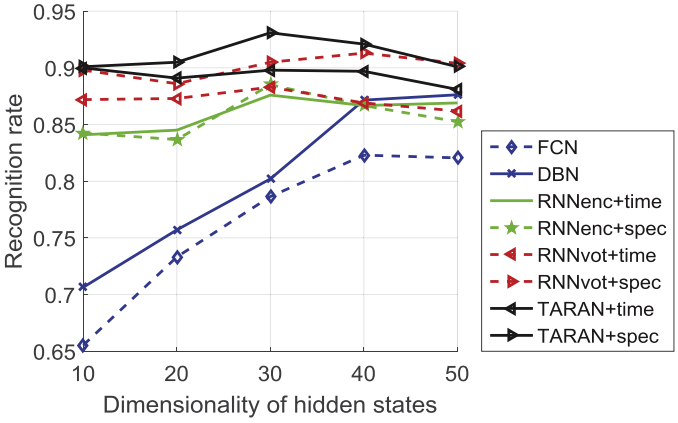

Fig. 6. Variation of the recognition performance with dimensionality of hidden states, via five neural network models: FCN, DBN, RNNenc, RNNvot, TARAN. We use the time sequential feature and spectrogram feature for RNNenc, RNNvot and TARAN models.

4. Experimental results

To show the effectiveness of the proposed TARAN model, we will compare our models to other counterparts on the measured data according to different criterions in the experiments. We will first briefly introduce our measured radar HRRP dataset and the detailed experiment settings. In the experiments, we will discuss the influence of different parameters on our model and give the detailed analysis based on the quantitative recognition performance. To provide qualitative analysis, we will look insight into the contents learned by our model, including the neurons and discriminative areas focused by attention mechanism.

4.1. Measured data

The results presented here are based on three measured airplane data, which are widely used in RATR [13,14]. An-26 is a medium-sized propeller aircraft, Cessna Citation is a small-sized jet aircraft and Yark-42 is a large-sized jet aircraft. The parameters of the targets and radar are shown in Table 1, and the projections of target trajectories onto ground plane are shown in Fig. 5, from which the aspect angle of airplane can be estimated according to its relative position to radar. The measured data are segmented into several parts. Training data and test data are from different data segments, among which the second and the fifth segments of Yark-42, the fifth and the sixth segments of An-26, the sixth and the seventh segments of Cessna Citation are taken as the training data, other data segments are taken as the test data. For the measured data, it requires that the training data cover almost all of the target-aspect angles of the test data. According to Fig. 5, this

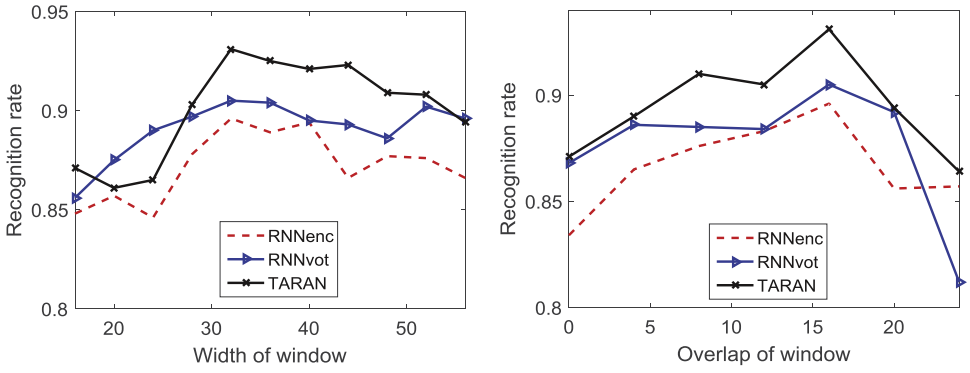

Fig. 7. Recognition performance of TARAN, RNNvot and RNNenc with different parameters in transformation window function. (a) The rate versus the width of window while fixed the overlap to the half of the width. (b) The rate versus the overlap while fixed the width as 32.

Table 2
Recognition performance of proposed models and counterpart methods.

Methods	Accuracy
MCC	0.590
AGC	0.852
HMM+time	0.870
HMM+spec	0.884
FCN	0.844
DBN	0.874
RNNenc+time	0.877
RNNenc+spec	0.896
RNNvot+time	0.883
RNNvot+spec	0.905
TARAN+time	0.901
TARAN+spec	**0.931**

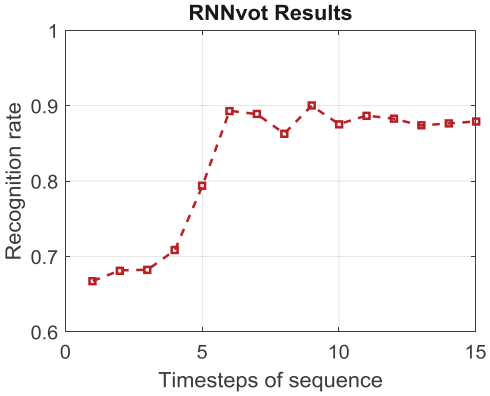

Fig. 8. Recognition rate versus the timesteps with RNNvot model.

can be satisfied. Moreover, the elevation angles of the test data are different from those of the training data, thus the generalization performance of the recognition methods can be tested. There are 7800 HRRP samples for training which correspond to 2600 profiles for each label. The number of test samples is 5200. Each sample is a 256-dimensional vector.

From the samples in Fig. 2 and Table 1, we could know that the size of target areas of Yark-42 is largest and the Cessna Citation is smallest. The measured data are manipulated to sequence by window function with window size $d = 32$ and overlap $b = 16$, thus the length of sequence is $T = 15$. It should be noticed that target areas of plane An-26 are approximately at timesteps 7 to 9 timesteps, those of Cessna Citation about 8 to 9 timesteps, and those of Yark-42 7 to 10 timesteps. The other areas of the HRRP sample are noisy areas which should take less effect in HRRP recognition.

4.2. Model settings

To investigate the role of modeling temporal dependence and attention mechanism, we evaluate the proposed models against several existing approaches for HRRP-based RATR, including the Maximum Correlation Coefficient (MCC) [35], Adaptive Gaussian Classifier (AGC) [36], and Hidden Markov Model (HMM) [18], the Fully Connected Network (FCN), Deep Belief Network (DBN) and Recurrent Neural Network encoder (RNNenc). DBN differs from the FCN is that DBN is pre-trained by Restricted Boltzmann Machine (RBM) [47]. The RNNenc is part of the RNNencoder-decoder [38], which has been used in machine translation. We replace the decoder with an soft-max function to output the label distribution with the representation at the final timestep.

All neural networks use a softmax output and are trained to minimize the categorical cross entropy error. For RNNs i.e. RNNenc, RNNvot and TARAN, all the weight matrices except the hidden-hidden matrix W_{hh} are initialized from a normal Gaussian distribution with its standard deviation set to 0.1. We initialize the W_{hh} as a diagonal matrix with its diagonal line set to 0.01. We use a mini-batch SGD algorithm to train each model with 10 samples in each minibatch. The learning rate is $0.1/(1+i)$ where i is the number of cycles. For DBN and FCN, we utilize the real HRRP vectors in (7).

4.3. The influence of model parameters

In this section, we analyze the effect of model parameters on the model performance. There are three parameters need to be discussed in the model, i.e. the dimension of hidden layers m, the window size d and the overlap b.

Fig. 6 shows the average recognition performance of methods versus the dimensionality of hidden states. As we can see, the recognition results of DBN and FCN are gradually raised with the increase of hidden layer dimension while the performance of RNNs is basically the peak value when the hidden layer dimension is 30. We attribute this to fact that the RNNs consider the data as temporal sequences, whereas DBN and FCN deal with the real HRRP vector in (7) with a relatively higher dimension.

We analyze the parameters in the window function i.e. the width d and the overlap b with fixing the hidden dimension as 30.

Table 3
Confusion matrices and average recognition rates of the RNNenc, RNNvot and TARAN, for time sequential feature.

Methods	RNNenc+time			RNNvot+time			TARAN+time		
	An-26	C.C	Yark-42	An-26	C.C	Yark-42	An-26	C.C	Yark-42
An-26	0.875	0.194	0.019	0.925	0.246	0.016	0.914	0.143	0.042
C.C	0.043	0.801	0.027	0.026	0.746	0.015	0.027	0.840	0.010
Yark-42	0.082	0.055	0.954	0.048	0.008	0.977	0.058	0.017	0.948
Ave Rate		0.877			0.883			**0.901**	

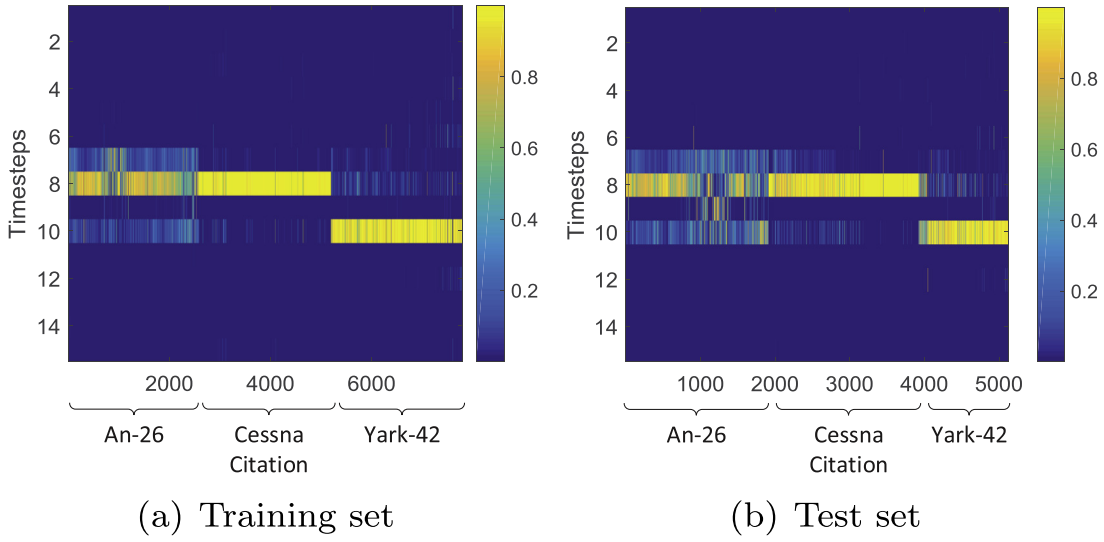

Fig. 9. Attention Coefficients in TARAN of training data and test data. (a) Training data. (b) Test data. Each column of the matrix corresponds to the attention weight a of each input sequence.

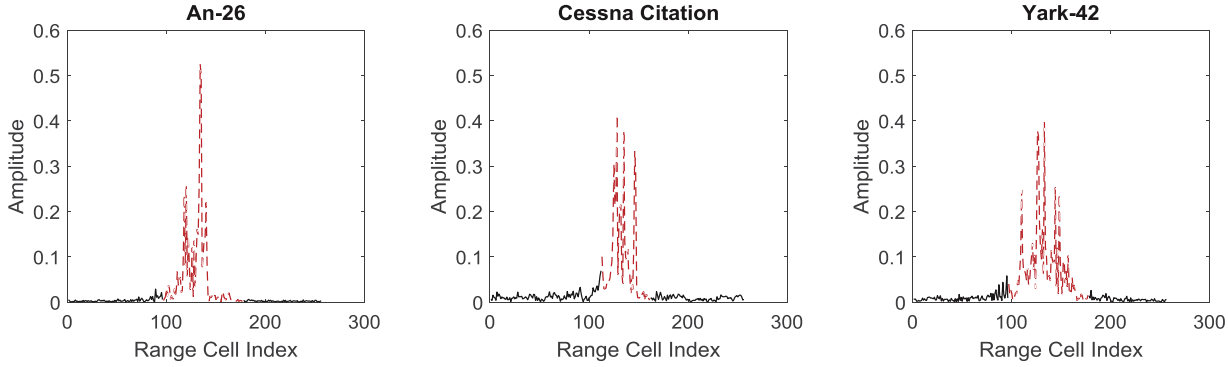

Fig. 10. Discriminative areas in red found by the attention mechanism.

For the spectrogram feature, a wide window function provides better frequency resolution, but deteriorates the time resolution, and vice versa. Meanwhile, the overlap across the window determines the degree of correlation between adjacent timesteps.

Fig. 7(a) shows the recognition performance of RNNs with spectrogram feature when the width d is varied from 16 to 56 and overlap $b = d/2$. As we can see, small values of d result in larger length of sequence, which requires a higher memory capacity of the model, and reduce the performance for all methods. On the other hand, the model does not work well when the width is too large and models basically perform best at width of 32. In Fig. 7(b), we fix the width as 32 and illustrate the dependence of these RNNs on the overlap b. Note that the performance gradually increases when overlap ranges from 0 to 16, while from 16 to 24, it drops sharply. The reason is that the correlation between adjacent input decrease when overlap is too small, which will weaken the temporal dependence between timesteps. On the contrary, excessive overlap grows the length of sequence rapidly, which increases the memory burden of RNN methods and decreases the recognition rate.

4.4. Recognition performance

In this section, to fairly compare all the different methods, according to the analysis of the results in the previous subsection, we set the hidden dimension of RNN methods as 30, while that of the DBN and FCN is set to 50 in the following experiments. For the transformation window function, we set the window size as $d = 32$ and the overlap $b = 16$. We follow the same setup as the prior works to train the MCC, AGC and HMM [13,18].

Table 2 compares the results of our models with traditional methods on both time sequential feature and spectrogram feature. As shown in Table 2, in conventional HRRP classification methods, the HMM performs better than the AGC and MCC, while, in neural

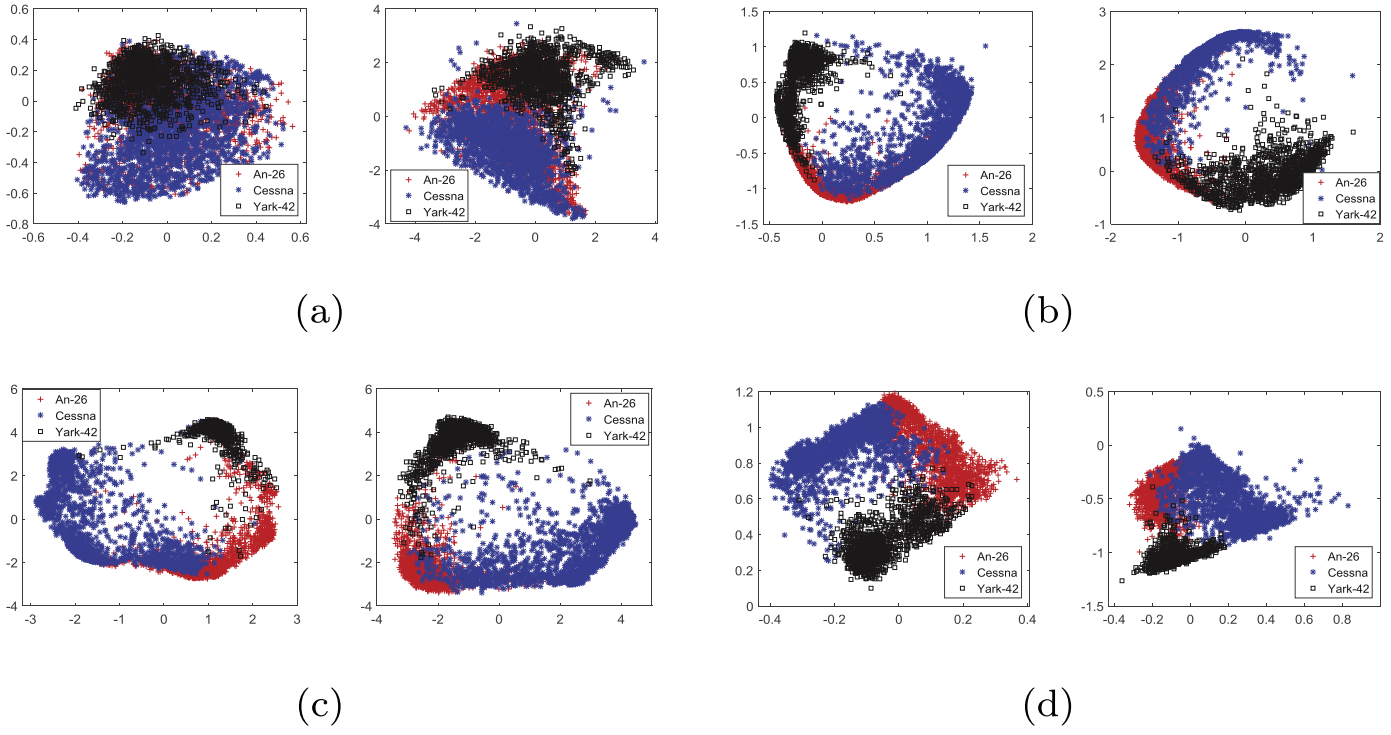

Fig. 11. Data distribution of the test HRRP samples after feature extraction by RAN, RNNvot and RNNenc via two-dimensional PCA. (a) measured testdata. (b) RNNenc. (c) RNNvot. (d) TARAN. The first and third column: The time domain. The second and forth column: The spectrogram.

networks, the RNNenc performs slightly better than the DBN and FCN. Our RNNvot outperforms the RNNenc and HMM by approximately 2% on both features. Since the HRRP sequence at different timestep holds different discriminative information, after combining with the attention mechanism, our best model TARAN outperforms all the other models. Notably, all RNN-based models with spectrogram feature perform better than that with the time sequential feature, which demonstrates the effectiveness of introducing the spectrogram in RATR task.

In order to investigate that different areas may provide different discriminative information, we plot the recognition rate of test dataset over the timesteps in Fig. 8. The result at timestep t is the average recognition rate of test data calculated by the y_t in (14). As can be seen from Fig. 8, the recognition rate from 1 to 4 timesteps corresponding to the noisy areas is poor while that of the following timesteps is basically stable in a relatively high level, which indicates the target areas are more discriminative than the noisy areas. Due to the dependency and transmission of the hidden states in RNN, the input from 11 to 15 timesteps, even if they are noise, also keep the stable performance as the already seen target areas.

Table 3 shows the confusion matrices and average recognition rates of RNN methods with time sequential feature. We see that the recognition performance of TARAN raising mainly come from the improvement in Cessna Citation. As the Cessna Citation is a small-sized jet aircraft whose waveform has larger fluctuation relative to An-26 and Yark-42, TARAN, which considers the temporal dependence between range cells and focuses on the relevant areas with attention mechanism, is more robust to the fluctuation and extract discriminative features to increase the classification performance.

We show the discriminative areas figured out by the attention mechanism by visualizing the attention weights a from (22) in Fig. 9, where each column of the matrix indicates the weights associated with each HRRP sequence and the bright parts of the graph are the detected regions most relevant to the RATR task. The attention mechanism focuses on the middle areas and puts small weight on the others. Specifically, the weights of Cessna Citation mainly focus on the timestep 8 while An-26 and Yark-42 concentrate on larger regions. This coincides with the characteristic of each target, as discussed in measured data, that each target may have different target area sizes equivalent to different input timesteps. In Fig. 10, we present an example for each target, where the dashed line parts denote the regions the model concentrates on. Across the examples, we see that the attention regions correspond to the target areas well, which coincides with our intuition.

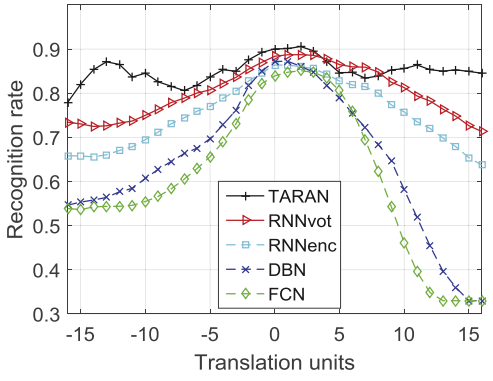

Fig. 12. Variation of the recognition rate with the different extents of misalignment, via TARAN, RNNvot, RNNenc, DBN and FCN. The training data are time-shift compensated and we translate the test data to evaluatethe ability of different methods' robustness to misalignment. Negative value of the horizontal coordinate means the left shift and positive the right shift.

Table 4 shows the confusion matrices and average recognition rates of RNNs with spectrogram feature. We see that all methods utilizing spectrogram feature in Table 4 improve the performance by approximately 2 percent and nearly all targets increase the rates compared with Table 3, since the spectrogram considers the phase information and is more robust to the variation of the target-aspect.

Table 4
Confusion matrices and average recognition rates of the RNNenc, RNNvot and TARAN, for spectrogram feature.

Methods	RNNenc+spec			RNNvot+spec			TARAN+spec		
	An-26	C.C	Yark-42	An-26	C.C	Yark-42	An-26	C.C	Yark-42
An-26	0.885	0.160	0.026	0.973	0.202	0.028	0.931	0.108	0.017
C.C	0.042	0.838	0.010	0.012	0.779	0.009	0.031	0.882	0.002
Yark-42	0.073	0.002	0.964	0.015	0.019	0.963	0.038	0.010	0.982
Ave Rate	0.896			0.905			**0.931**		

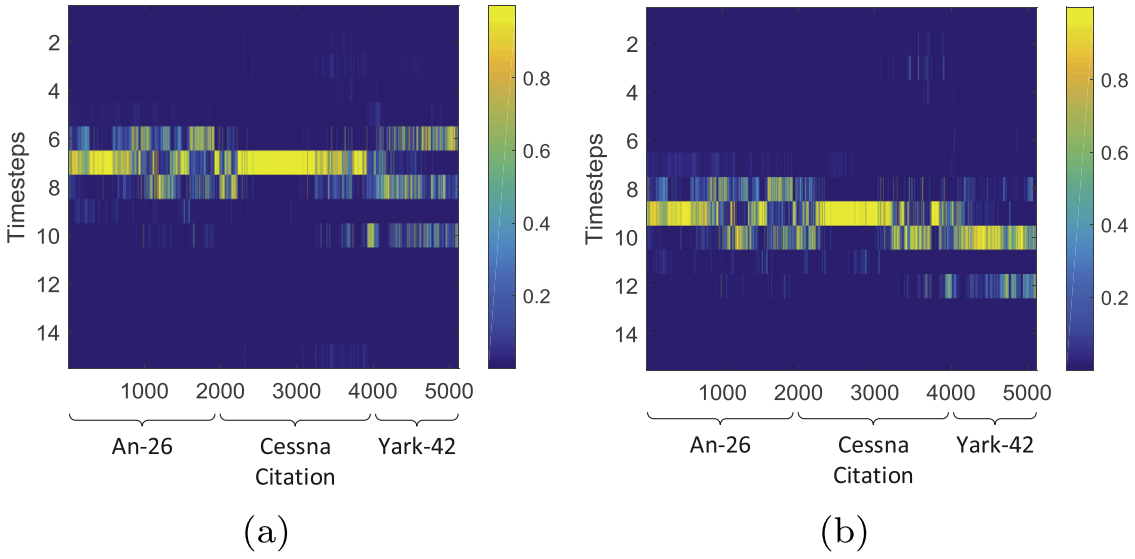

Fig. 13. The learned attention coefficients of test data with translation. (a) Translate the input to the left by 16 units. (b) Translate the input to the right by 16 units.

Fig. 11 shows the mapping of the features extracted by TARAN, RNNvot and RNNenc and the original test dataset to the first two principal components, where TARAN produces more discriminative feature space than the other methods and the spectrogram feature are more separable compared with the time domain feature.

4.5. Generalization performance

According to [14], HRRP has the characteristic of time-shift sensitivity, which means that the position of target areas in HRRP may change during measuring. In this experiment, we align each training sample with the centroid of sample and translate the test data set to evaluate the robustness of the proposed model to the time-shift sensitivity. We compare the TARAN with RNNvot, RNNenc, DBN and FCN methods which are trained in the same way as aforementioned in Section 4.2.

In Fig. 12, we depict the recognition performance of methods varying with the translation units of test samples. Note that the performance of DBN and FCN decreases significantly as the misalignment increases while that of the RNNenc model dose not change a lot thanks to the ability of the model delivering discriminative information. By voting strategy, the RNNvot frees the RNNenc from having to squash all the information of an HRRP sequence into a fixed-length vector and achieves better performance than the RNNenc. Our TARAN model has the least performance degradation compared with other methods, which can be attributed to the attention mechanism adaptively focusing on the discriminative target regions even with test samples having different extents of misalignment.

We look insight to the learned attention weights based on the translated test dataset, as shown in Fig. 13. The left diagram represents the attention weights that the dataset is shifted to the left by 16 units, which corresponds to moving one timestep on the sequence and the right figure denotes the weights of translating the data 16 units to the right. As can be seen from Fig. 13, the attention position of the left graph is basically shifted up by one timestep compared to Fig. 9, while the right one moves down a timestep. Based on the above phenomenon, we can find that the proposed model is able to still focus on the target areas even in the presence of translation during the test stage with the help of the data-dependent attention mechanism.

5. Conclusions

In this article, RNNs are analyzed and utilized for HRRP-based RATR to further take advantage of the temporal dependence between range cells. In order to find the discriminative areas in HRRP, we further propose Target-Aware Recurrent Attentional Network (TARAN), which employs the RNN to encode the HRRP sequence into hidden states and weight the hidden state at each timestep to focus on the discriminative regions data-dependently with the help of attention strategy. All the pieces of the TARAN model, including attention mechanism and RNN, are jointly trained by optimizing the log-probability of producing correct labels. Experimental results on measured data demonstrate that the proposed model not only achieves superior recognition performance than traditional methods, but more robust to the time-shift sensitivity thanks to the good properties of RNN and attention mechanism.

Acknowledgment

This work is partially supported by the Thousand Young Talent Program of China, NSFC (61771361), 111 Project (B18039), and the National Science Fund for Distinguished Young Scholars of China (61525105).

Appendix A

In this appendix, we show the procedure that calculates the gradients of the cost function to the parameters in the TARAN model in Fig. 14 and the value of cost function versus epoch in Fig. 15. We begin with the feedforward of the model.

$$v_t = W_{hx}x_t + W_{hh}h_{t-1} \tag{30}$$

$$h_t = f(W_{hx}x_t + W_{hh}h_{t-1}) \tag{31}$$

$$e_t = U_a^{\mathrm{T}} \tanh(W_a h_t) \tag{32}$$

$$a_t = \frac{\exp(e_t)}{\sum_{l=1}^{T}\exp(e_l)} \tag{33}$$

$$C = \sum_{t=1}^{T} a_t h_t \tag{34}$$

$$O_t = W_{cy}C_t \tag{35}$$

$$y = softmax(W_{cy}C) = \frac{\exp(W_k C)}{\sum_{l=1}^{K}\exp(W_l C)} \tag{36}$$

where $y \in \mathbb{R}^{K\times 1}$, and the parameters $W_{hx} \in \mathbb{R}^{m\times d}, W_{hh} \in \mathbb{R}^{m\times m}, W_{cy} \in \mathbb{R}^{K\times m}, U_a \in \mathbb{R}^{1\times n}, W_a \in \mathbb{R}^{n\times m}$.

For easier understanding, we rewrite the cost function from (24) as:

$$L = -\sum_{n=1}^{N}\sum_{k=1}^{K} z_k \log(y_k) \tag{37}$$

where z_k is the kth value of the real label vector z and y_k denotes the kth value of the label vector y the model predicts. Thus, the gradient of the cost function to the variable O can be calculated as:

$$\begin{aligned}\frac{dL}{dO_i} &= \sum_{k=1}^{K}\frac{dL}{dy_k}\frac{dy_k}{dO_i}\\ &= \left(\sum_{k\neq i,k=1}^{K}\frac{dL}{dy_k}\frac{dy_k}{dO_i}\right) + \frac{dL}{dy_i}\frac{dy_i}{dO_i}\\ &= \left(\sum_{k\neq i,k=1}^{K}\left(-\frac{z_k}{y_k}\right)\frac{dy_k}{do_i}\right) + \left(-\frac{y_i}{y_i}\frac{dy_i}{dO_i}\right)\\ &= \left(\sum_{k\neq i,k=1}^{K}\left(-\frac{y_k}{y_k}\right)(-y_i y_k)\right) + \left(-\frac{z_i}{y_i}y_i\right)\\ &= y_i - z_i\end{aligned} \tag{38}$$

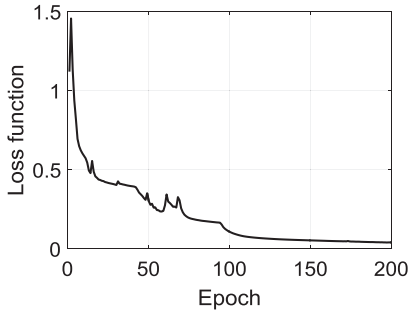

Fig. 15. The cost function value versus the training epoch. The value of the cost function decreases gradually with the training epoch and becomes stable when the epoch reaches about 150, which indicates the model has tended to be converge.

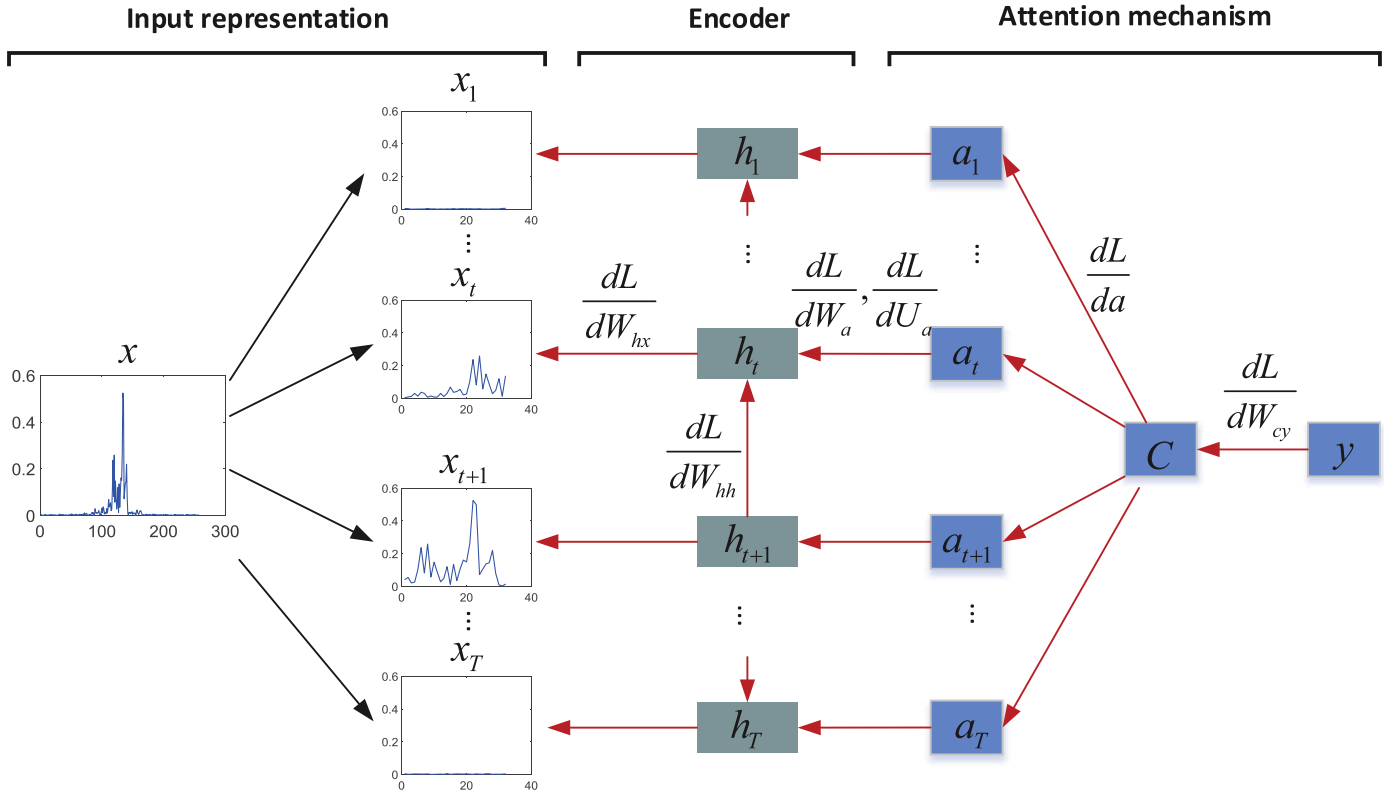

Fig. 14. The detailed message passing between different states and parameters. The red line represents the direction from the cost function to the parameters and the latent variables. Since the parameters W_{hx}, W_{hh}, W_a, U_a are shared at all timesteps, the gradients of them are calculated by summing up the gradients at all timesteps. On the other hand, as the hidden states connected by W_{hh}, it should consider the gradient of h_{t+1} to h_t besides the gradient of a to h_t when calculating the gradient of h_t. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article).

where O_i denotes the ith element of the variable $O \in \mathbb{R}^{K\times 1}$ and

$$\frac{dL}{dO} = z - y \tag{39}$$

Suppose $W_{cy} = [W_1, \cdots, W_i, \cdots, W_m]$, then

$$O = \sum_{i=1}^{m} C_i * W_i \tag{40}$$

where C_i is the ith value of the vector C. The gradient of the cost function to the W_{cy} is

$$\begin{aligned}\frac{dL}{dW_i} &= \frac{dO^{\mathrm{T}}}{dW_i}\frac{dL}{dO} \\ &= \frac{d(C_i W_i^{\mathrm{T}})}{dW_i}\frac{dL}{d_O} \\ &= C_i(y-z)\end{aligned} \tag{41}$$

$$\frac{dL}{dW_{cy}} = (y-z)C^{\mathrm{T}} \tag{42}$$

For the parameters $\theta = \{W_a, U_a\}$, we first calculate the gradient of cost function to the latent variable $\frac{dL}{de}$.

$$\begin{aligned}\frac{dL}{da_i} &= \frac{dC^{\mathrm{T}}}{da_i}\frac{dL}{dC} \\ &= \frac{d(\sum_{i=1}^{T} a_i h_i^{\mathrm{T}})}{da_i}\frac{dL}{dC} \\ &= h_i^{\mathrm{T}}[W_{cy}^{\mathrm{T}}(y-z)]\end{aligned} \tag{43}$$

$$\begin{aligned}\frac{dL}{de_i} &= \sum_{j=1}^{T}\frac{dL}{da_i}\frac{da_i}{de_i} \\ &= \sum_{j=1,j\neq i}^{T}\frac{dL}{da_j}\frac{da_j}{de_i} + \frac{dL}{da_i}\frac{da_i}{de_i}\end{aligned} \tag{44}$$

where

$$\frac{da_j}{de_i} = \frac{-exp(e_i)exp(e_j)}{\sum_{t=1}^{T} exp(e_t)^2} = -a_i a_j \tag{45}$$

$$\frac{da_i}{de_i} = \frac{exp(e_i)\sum_{t=1}^{T} exp(e_t) - exp(e_i)^2}{\sum_{t=1}^{T} exp(e_t)^2} = a_i - a_i^2 \tag{46}$$

Therefore,

$$\begin{aligned}\frac{dL}{de_i} &= \sum_{j=1,j\neq i}^{T}\frac{dL}{da_j}(-a_i a_j) + \frac{dL}{da_i}(a_i - a_i^2) \\ &= a_i\frac{dL}{da_i} - \sum_{j=1}^{T}\frac{dL}{da_j}(a_i a_j)\end{aligned} \tag{47}$$

According to the chain rule, the gradient of the cost function to the parameter U_a, W_a can be calculated as:

$$\begin{aligned}\frac{dL}{dU_a} &= \sum_{i=1}^{T}\frac{dL}{de_i}\frac{de_i}{dU_a} \\ &= \sum_{i=1}^{T}\frac{dL}{de_i}tanh(W_a h_i)\end{aligned} \tag{48}$$

$$\begin{aligned}\frac{dL}{dW_a} &= \sum_{i=1}^{T}\frac{dL}{de_i}\frac{de_i}{dW_a} \\ &= \sum_{i=1}^{T}\frac{dL}{de_i}diag\{1 - tanh^2(W_a h_i)\}U_a h_i^{\mathrm{T}}\end{aligned} \tag{49}$$

For the parameters in the encoder $\theta_2 = \{W_{hx}, W_{hh}\}$, we first calculate the gradient of cost function to the hidden states $h = [h_1, \cdots, h_t, \cdots, h_T]$

$$\begin{aligned}\frac{dL}{dh_i} &= \frac{dC^{\mathrm{T}}}{dh_i}\frac{dL}{dC} \\ &= \frac{dC^{\mathrm{T}}}{de_i}\frac{de_i}{h_i}\frac{dL}{dC} \\ &= \sum_{j=1}^{T}(-a_i a_j)\frac{de_i}{dh_i}h_j^{\mathrm{T}}\frac{dL}{dC} + a_i\frac{de_i}{dh_i}h_i^{\mathrm{T}}\frac{dL}{dC} + a_i\frac{dL}{dC}\end{aligned} \tag{50}$$

According to the chain rule, we can get the gradient of cost function to the variable v_t.

$$\begin{aligned}\frac{dL}{dv_t} &= \frac{dh_t^{\mathrm{T}}}{dv_t}\frac{dL}{dh_t} \\ &= \frac{dh_t^{\mathrm{T}}}{dv_t}\left(\frac{dL}{dh_t} + \frac{dv_{t+1}^{\mathrm{T}}}{dh_t}\frac{dL}{dh_t}\right) \\ &= diag(h_t(1-h_t))\left[\frac{dL}{dh_t} + W_{hh}^{\mathrm{T}}\frac{dL}{dv_{t+1}}\right]\end{aligned} \tag{51}$$

Therefore, the gradient of the cost function to the parameters W_{hh}, W_{hx} can be calculated as:

$$\begin{aligned}\frac{dL}{dW_{hx}} &= \sum_{t=1}^{T}\frac{dL}{dv_t}\frac{dv_t}{dW_h x} \\ &= \sum_{t=1}^{T}\frac{dL}{dv_t}x_i^{\mathrm{T}}\end{aligned} \tag{52}$$

$$\begin{aligned}\frac{dL}{dW_{hh}} &= \sum_{t=2}^{T}\frac{dL}{dv_t}\frac{dv_t}{dW_{hh}} \\ &= \sum_{t=2}^{T}\frac{dL}{dv_t}h_{t-1}^{\mathrm{T}}\end{aligned} \tag{53}$$

The calculation of the gradients of the cost function to the parameters $\theta = W_{cy}, W_a, U_a, W_{hh}, W_{hx}$ is complete.

References

[1] B. Chen, H. Liu, J. Chai, Z. Bao, Large margin feature weighting method via linear programming, IEEE Trans. Knowl. Data Eng. 21 (10) (2009) 1475–1488.

[2] L. Du, P. Wang, L. Zhang, H. He, H. Liu, Robust statistical recognition and reconstruction scheme based on hierarchical bayesian learning of HRR radar target signal, Expert Syst. Appl. 42 (14) (2015) 5860–5873.

[3] X.D. Zhang, Y. Shi, Z. Bao, A new feature vector using selected bispectra for signal classification with application in radar target recognition, Signal Process. IEEE Trans. 49 (9) (2001) 1875–1885.

[4] L. Du, P. Wang, H. Liu, M. Pan, F. Chen, Z. Bao, Bayesian spatiotemporal multi-task learning for radar HRRP target recognition, IEEE Trans. Signal Process. 59 (7) (2011) 3182–3196.

[5] J. Chai, H. Liu, Z. Bao, Combinatorial discriminant analysis: supervised feature extraction that integrates global and local criteria 45 (18) (2009) 934–935.

[6] X. Liao, P. Runkle, L. Carin, Identification of ground targets from sequential high-range-resolution radar signatures, IEEE Trans. Aerosp. Electron. Syst. 38 (4) (2002) 1230–1242.

[7] L. Du, L. Li, Y. Ma, B. Wang, A noise-robust radar target classification method based on complex probabilistic principal component analysis, in: General Assembly and Scientific Symposium, 2014, pp. 1–4.

[8] H. Liu, B. Feng, B. Chen, L. Du, Radar high-resolution range profiles target recognition based on stable dictionary learning, Iet Radar Sonar Navig. 10 (2) (2016) 228–237.

[9] L. Shi, P. Wang, H. Liu, L. Xu, Z. Bao, Radar HRRP statistical recognition with local factor analysis by automatic bayesian ying-yang harmony learning, IEEE Trans. Signal Process. 59 (2) (2011) 610–617.

[10] L. Du, P. Wang, H. Liu, M. Pan, Z. Bao, Radar HRRP target recognition based on dynamic multi-task hidden markov model, in: Proceeding of Empirical Methods in Natural Language Processing (EMNLP), 2011, pp. 253–255.

[11] B. Feng, L. Du, H.W. Liu, F. Li, Radar HRRP target recognition based on k-SVD algorithm, in: IEEE Cie International Conference on Radar, 2012, pp. 642–645.

[12] B. Feng, L. Du, C. Shao, P. Wang, H. Liu, Radar HRRP target recognition based on robust dictionary learning with small training data size (2013) 1–4.
[13] B. Feng, B. Chen, H. Liu, Radar HRRP target recognition with deep networks, Pattern Recognit. 61 (2017) 379–393.
[14] L. Du, H. Liu, Z. Bao, J. Zhang, A two-distribution compounded statistical model for radar HRRP target recognition, IEEE Trans. Signal Process. 54 (6) (2006) 2226–2238.
[15] M. Pan, L. Du, P. Wang, H. Liu, Z. Bao, Multi-task hidden markov model for radar automatic target recognition, in: IEEE Cie International Conference on Radar, 2011, pp. 650–653.
[16] L. Du, H. Liu, Z. Bao, Radar automatic target recognition based on complex high-resolution range profiles, in: International Conference on Radar, 2006, pp. 1–5.
[17] C. Wang, H. Zhang, Z. Bo, X. Wen, F. Wu, C. Zhang, Automatic target recognition in high resolution sar image based on backscattering model, Dianbo Kexue Xuebao/Chin. J. Radio Sci. 19 (4) (2004) 422–426.
[18] M. Pan, L. Du, P. Wang, H. Liu, Z. Bao, Multi-task hidden markov modeling of spectrogram feature from radar high-resolution range profiles, EURASIP J. Adv. Signal Process. 2012 (1) (2012) 86.
[19] R.A. Mitchell, J.J. Westerkamp, Robust statistical feature based aircraft identification, IEEE Trans. Aerosp. Electron. Syst. 35 (3) (1999) 1077–1094.
[20] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, Int. Conf. Learn. Represent. (2015).
[21] X. Liao, Z. Bao, Circularly integrated bispectra: novel shift invariant features for high-resolution radar target recognition, Electron Lett. 34 (19) (1998) 1879–1880.
[22] L. Du, H. Liu, Z. Bao, M. Xing, Radar HRRP target recognition based on higher order spectra, IEEE Trans. Signal Process. 53 (7) (2005) 2359–2368.
[23] P. Molchanov, K.O. Egiazarian, J. Astola, A.V. Totsky, S. Leshchenko, M. Jaraboamores, Classification of aircraft using micro-doppler bicoherence-based features, IEEE Trans.Aerosp. Electron. Syst. 50 (2) (2014) 1455–1467.
[24] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.A. Manzagol, Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion, J. Mach. Learn. Res. 11 (12) (2010) 3371–3408.
[25] M. Pan, J. Jiang, Q. Kong, J. Shi, Q Sheng, T. Zhou, Radar HRRP target recognition based on t-SNE segmentation and discriminant deep belief network, IEEE Geosci, Remote Sens. Lett. PP (99) (2017) 1–5.
[26] A. Graves, A. Mohamed, G.E. Hinton, Speech recognition with deep recurrent neural networks, Int. Conf. Acoust. Speech Signal Process. (2013) 6645–6649.
[27] J.L. Elman, Finding structure in time, Cogn. Sci. 14 (2) (1990) 179–211.
[28] M. Sundermeyer, R. Schlter, H. Ney, Lstm neural networks for language modeling, in: Interspeech, 2012, pp. 601–608.
[29] N. Kalchbrenner, P. Blunsom, Recurrent continuous translation models, in: IEEE Radar Conference, 2013, pp. 1700–1709.
[30] J.P. Zwart, R.V. Der Heiden, S. Gelsema, F.C.A. Groen, Fast translation invariant classification of HRR range profiles in a zero phase representation, IEEE Proc.-Radar Sonar Navig. 150 (6) (2003) 411–418.
[31] H.W. Liu, B. Zheng, Radar HRRP recognition based on hybrid features and multi-stage feature selection scheme, Syst. Eng. Electron. (2005).
[32] X. Fan, H.U. Shengliang, H.E. Jingbo, Feature extraction and selection of full polarization HRRP in target recognition process of maritime surveillance radar, J. Electron. Inf. Technol. (2016).
[33] B. Kingsbury, N. Morgan, S. Greenberg, Robust speech recognition using the modulation spectrogram, Speech Commun. 25 (1998) 117–132.
[34] L. Du, H. He, L. Zhao, P. Wang, Noise robust radar HRRP target recognition based on scatterer matching algorithm, IEEE Sens. J. 16 (6) (2016) 1743–1753.
[35] L. Du, H. Liu, Z. Bao, M. Xing, Radar HRRP target recognition based on higher order spectra, IEEE Trans. Signal Process. 53 (7) (2005) 2359–2368.
[36] L. Du, H. Liu, Z. Bao, Radar HRRP statistical recognition: parametric model and model selection, IEEE Trans. Signal Process. 56 (5) (2008) 1931–1944.
[37] B. Su, S. Lu, Accurate Scene Text Recognition Based on Recurrent Neural Network, Springer International Publishing, 2014.
[38] K. Cho, B.V. Merrienboer, D. Bahdanau, Y. Bengio, On the properties of neural machine translation: encoder-decoder approaches, Comput. Sci. (2014).
[39] J. Kuen, Z. Wang, G. Wang, Recurrent attentional networks for saliency detection, in: Computer Vision and Pattern Recognition, 2016, pp. 3668–3677.
[40] Z. Yang, X. He, J. Gao, L. Deng, A. Smola, Stacked attention networks for image question answering, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 21–29.
[41] I. Guyon, A. Elisseeff, An introduction to variable feature selection, J. Mach. Learn. Res. 3 (2003) 1157–1182.
[42] W. Penghui, Study of Radar High Resolution Range Profile Target Recognition based on Statistical Modeling, Xi'Dian University, 2012 Ph.D. thesis.
[43] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber, Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, in: J.F. Kolen, S. Kremer (Eds.), A Field Guide to Dynamical Recurrent Networks, IEEE Press, New York, 2001, pp. 237–243.
[44] S. Amari, Backpropagation and stochastic gradient descent method, Neurocomputing 5 (1993) 185–196.
[45] R. Pascanu, T. Mikolov, Y. Bengio, On the difficulty of training recurrent neural networks, Int. Conf. Mach. Learn. (2013) 1310–1318.
[46] L. Bottou, Stochastic gradient learning in neural networks, Proc. Neuro Nimes (1991).
[47] G.E. Hinton, R. Salakhutdinov, Reducing the dimensionality of data with neural networks., Science 313 (5786) (2006) 504–507.