RAA 2019 Vol. **19** No. **9**, 135(12pp) doi: 10.1088/1674–4527/19/9/135

http://www.raa-journal.org *http://iopscience.iop.org/raa*

Research in Astronomy and Astrophysics

LSTM neural network for solar radio spectrum classification

Long Xu[1,2], Yi-Hua Yan[1], Xue-Xin Yu[1], Wei-Qiang Zhang[2], Jie Chen[3] and Ling-Yu Duan[3]

[1] Key Laboratory of Solar Activity, National Astronomical Observatories, Chinese Academy of Sciences, Beijing 100101, China; *lxu@nao.cas.cn*

[2] College of Mathematics and Statistics, Shenzhen University, Shenzhen 518060, China

[3] National Engineering Lab for Video Technology, Peking University, Beijing 100871, China

Received 2019 February 13; accepted 2019 April 27

Abstract A solar radio spectrometer records solar radio radiation in the radio waveband. Such solar radio radiation spanning multiple frequency channels and over a short time period could provide a solar radio spectrum which is a two dimensional image. The vertical axis of a spectrum represents frequency channel and the horizontal axis signifies time. Intrinsically, time dependence exists between neighboring columns of a spectrum since solar radio radiation varies continuously over time. Thus, a spectrum can be treated as a time series consisting of all columns of a spectrum, while treating it as a general image would lose its time series property. A recurrent neural network (RNN) is designed for time series analysis. It can explore the correlation and interaction between neighboring inputs of a time series by augmenting a loop in a network. This paper makes the first attempt to utilize an RNN, specifically long short-term memory (LSTM), for solar radio spectrum classification. LSTM can mine well the context of a time series to acquire more information beyond a non-time series model. As such, as demonstrated by our experimental results, LSTM can learn a better representation of a spectrum, and thus contribute better classification.

Key words: deep learning — long short-term memory (LSTM) — classification — solar radio spectrum — solar burst detection

1 INTRODUCTION

Since solar radio radiation can go through Earth's atmosphere without being absorbed, the radio waveband has been becoming a more and more important window to explore the Universe, not only targeting the Sun, but also in nighttime astronomy for probing more distant stars. Solar observation in the radio waveband provides us a window to study the Sun, especially the solar atmosphere. Recently, with the development of telescopes with higher precision in terms of spatio-temporal resolution, solar observations have entered the big data era. Traditional data analysis and processing, usually performed manually, cannot fulfill the mission of daily solar observation. We need automatic and intelligent data analysis and processing in the big data era. In this paper, we investigate classification of a solar radio spectrum by introducing deep learning. The spectrums are provided by the Solar Broadband Radio Spectrometer (SBRS) in China (Fu et al. 2004). The SBRS, consisting of five "component spectrometers" and covering a wide frequency range from $0.7-7.6$ GHz, monitors solar radio activity every day, therefore accumulating massive amounts of data. The classification of data is the first step in big data analysis. However, manual classification of data is boring and exhausting to researchers. In addition, people cannot easily and correctly classify spectrums under the condition of strong noise interference. Therefore, automatic spectral classification is very meaningful for both our research and daily observation. Since a solar burst event is very sparse among all data (less than 5% of all recorded data), only binary classification is of great value. Through binary classification, we can firstly identify bursts from the extensive data, saving more than 95% of human labor in the following data analysis. In this work, we are concerned with only three types of spectrums, "burst," "non-burst" and "calibration."

Nowadays, with the availability of massive amounts of data, deep learning (Bengio 2009) has been extensively explored to perform many traditional tasks of recognition, classification, regression and clustering. The methods of deep learning, such as convolutional neural networks (CNNs) (LeCun et al. 1989; Simonyan & Zisserman 2014), auto-encoders (AEs) (Vincent et al. 2008) and deep

belief networks (DBNs) (Hinton & Salakhutdinov 2006; Hinton et al. 2006; Hinton 2012), have demonstrated state-of-the-art performances in a wide variety of tasks, including visual recognition (Sohn et al. 2011), audio recognition (Mohamed et al. 2012) and natural language processing (Collobert et al. 2011). These methods can directly learn useful features from unlabeled or labeled data, with no need for hand-crafted features. This advantage of deep learning is very useful, especially when applied to tasks for which we do not have strong domain knowledge, or where their physical processes are so complex that we cannot rationalize or model them precisely.

A solar radio spectrometer records solar radio flux over time. In addition, it may record at several frequency channels. Then, the image of solar radio flux can be drawn in the two-dimensional space of time and frequency. This kind of image is named a solar radio spectrum (or spectrum for short). Since each spectrum represents a specific type of solar activity, it demonstrates a distinctive image pattern. There are many kinds of spectrums, such as "pulse," "drifting," "zebra" and "fiber." They are named according to their morphology, i.e., from their external shapes. The pattern of a spectrum gives a special solar activity caused by different radio radiation. In addition, solar bursts are very sparse among all recorded data, especially in this period, namely Solar Cycle 24, which is a minimum solar cycle. It is meaningful to identify solar bursts from the extensive data before the following data processing and analysis are applied, which only concern solar burst data. Up till now, there have been little works on automatic classification/detection of solar radio bursts. Also, there is no prior knowledge about hand-crafted features of spectrums. So, we employed deep learning to extract features of a spectrum and train classifiers. The feature extractor and classifier form an end-to-end framework of spectrum classification, which directly learns image features from an input image without the need for hand-crafted features. In Chen et al. (2016), we established a model of spectrum classification using a DBN. It has been shown that the proposed model can learn better representation of a solar radio spectrum, and thus achieve higher accuracy of classification beyond the traditional support vector machine (SVM) (Suykens & Vandewalle 1999) coupled with principal component analysis (PCA) (Jolliffe 2011; Wold et al. 1987). In Chen et al. (2015); Ma et al. (2017), an AE (Vincent et al. 2008, 2010) was explored for spectrum classification. In addition, the multiple modality concept (Guillaumin et al. 2010; Ngiam et al. 2011) was introduced to exploit the correlations between adjacent frequency channels, where each channel was regarded as one modality. In Chen et al. (2017b), a CNN was employed for spectrum classification, where each spectrum was treated as a general image. In this CNN model, four convolutional layers and a softmax layer were stacked together to realize both feature extraction and classification.

The previous models (Chen et al. 2016, 2017b) are all implemented at each time input. The context information of a spectrum which is a time series was not utilized. In these models, each spectrum is regarded as a general image. In fact, a spectrum is a special type of image, describing a process of solar radio radiation over a certain time. In addition, it extends over multiple frequency channels. In Ma et al. (2017), we made the first attempt to establish a multimodal learning model, specifically a model composed of AE and structured regularization (SR), to learn the representation of a solar radio spectrum. We regarded a spectrum as a set of multiple modalities, where the multiple modalities correspond to multiple frequency channels in a spectrum. This multimodal method took multiple frequency channels of a spectrum as the inputs at each time. In lower layers, each modality was trained independently using an AE network. Meanwhile, all modalities interacted through an SR. Then, two fully-connected layers were stacked on top of these AE layers. Finally, a softmax layer was stacked on the top of all hidden layers.

Taking each column as one input, a spectrum can then be treated as a time series. To process a time series, recurrent neural networks (RNNs) (Suykens & Vandewalle 1999; Graves et al. 2013; Sundermeyer et al. 2012) were developed. They contain a loop which allows post inputs to persist until an output is produced. There have been lots of variants of RNNs now in the literature. A long short-term memory (LSTM) network (Hochreiter & Schmidhuber 1997; Gers et al. 1999) is a typical example of an RNN. A common LSTM unit consists of a cell, a forget gate, an input gate and an output gate. The cell keeps the values over arbitrary time intervals for the following LSTM units. The flow of information into and out of the cell is regulated by three gates.

In this paper, all the models we have developed for spectrum classification in Chen et al. (2016, 2015); Ma et al. (2017); Chen et al. (2017b); Yu et al. (2017) are summarized and compared. In particular, the LSTM model is extended and further enriched. Among all models, the LSTM achieves the best performance by exploring more information in a time series. The rest of the paper is organized as follows. In Section 2, the LSTM network is introduced. Section 3 discusses the pre-processing of solar radio spectrums to improve visual quality and reduce computational complexity for the subsequent classification task. Section 4 presents the proposed model based on the LSTM network. Section 5 gives the experimental results and analysis of spectrum classification. The final section concludes the paper.

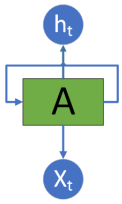


Fig. 1 RNNs with loops.

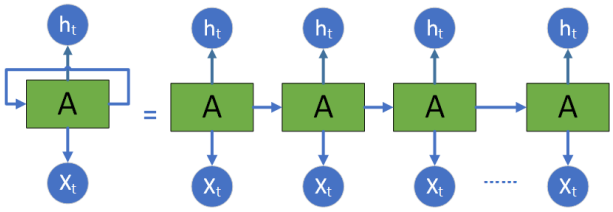


Fig. 2 Unrolled RNN.

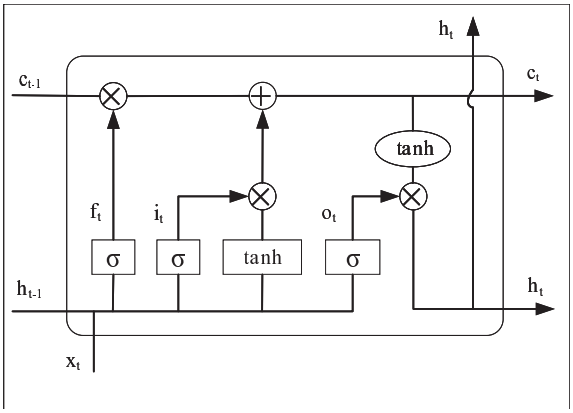


Fig. 3 LSTM neural network.

2 LONG SHORT-TERM MEMORY NETWORK

In the field of deep learning, most methods, such as fully-connected and CNN, produce output only from the current input. That means no context of a time series is utilized. However, for the tasks of video processing, language modeling, translation, speech recognition and reading comprehension, the input context is of great importance in recognition and understanding. From the perspective of human understanding, we do not start our thinking from scratch every second as we read a paper, and we understand each word based on our understanding of previous words. We do not throw everything away and start thinking from scratch again, but traditional neural networks cannot do this. They process the inputs separately, with output at every input. They have no mechanism for telling how the history of inputs informs the current status and output.

RNNs can address this issue. They are networks with loops in them, allowing information to persist. In Figure 1, a diagram of an RNN is illustrated, where a chunk of neural network, A, receives an input x_t and outputs h_t. A loop allows information to be passed from one step of the network to the next. An RNN can be thought of as multiple copies of the same network, each passing a message to a successor. It is not all that different from a traditional neural network as shown in Figure 2 if we unroll the loop. This chain-like nature reveals that RNNs are intimately related to sequences and lists. They are the natural architecture for neural networks to use for such data to explore correlations and interactions among adjacent time points in a time series. In the last few years, a variety of applications of RNNs, including speech recognition (Graves et al. 2013), language modeling (Sundermeyer et al. 2012; Mikolov et al. 2010), machine translation (Cho et al. 2014) and image captioning (Chen et al. 2017a; Ren et al. 2017), has achieved incredible success.

RNN keeps historical inputs until the current output to better enhance understanding of the current status, e.g., reading comprehension. In the case of short-term dependency, only very recent historical inputs are needed, which can be done well by RNN. However, for long-term dependency RNN would fail in practice, although it should be good for any case in theory. LSTM is a special kind of RNN. It can process both long-term and short-term dependencies well. All RNNs have a repeating basic unit, which conforms to a chain-structure. For a traditional RNN, the repeating basic unit has only one nonlinear layer, while the basic unit for LSTM is more complex. As shown in Figure 3, an LSTM unit has four nonlinear layers. In addition, they interact with each other through a special strategy, namely gate. The key of LSTM is a cell status, which is the horizontal axis in Figure 3. LSTM keeps or inhibits cell status by gates. The gate is a technique to let signals pass selectively. In Figure 3, it realizes information persistence and inhibition through three gates. Intuitively, the forget gate controls the extent to which a value remains in the cell, the input gate controls the extent to which a new value flows into the cell and the output gate controls the extent to which the value in the cell is used to compute the output activation of the LSTM unit.

In an LSTM layer, the mapping function from an input sequence $x = (x_1, x_2, \cdots, x_T)$ to an output sequence $h = (h_1, h_2, \cdots, h_T)$ is precisely specified by

$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f), \tag{1}$$

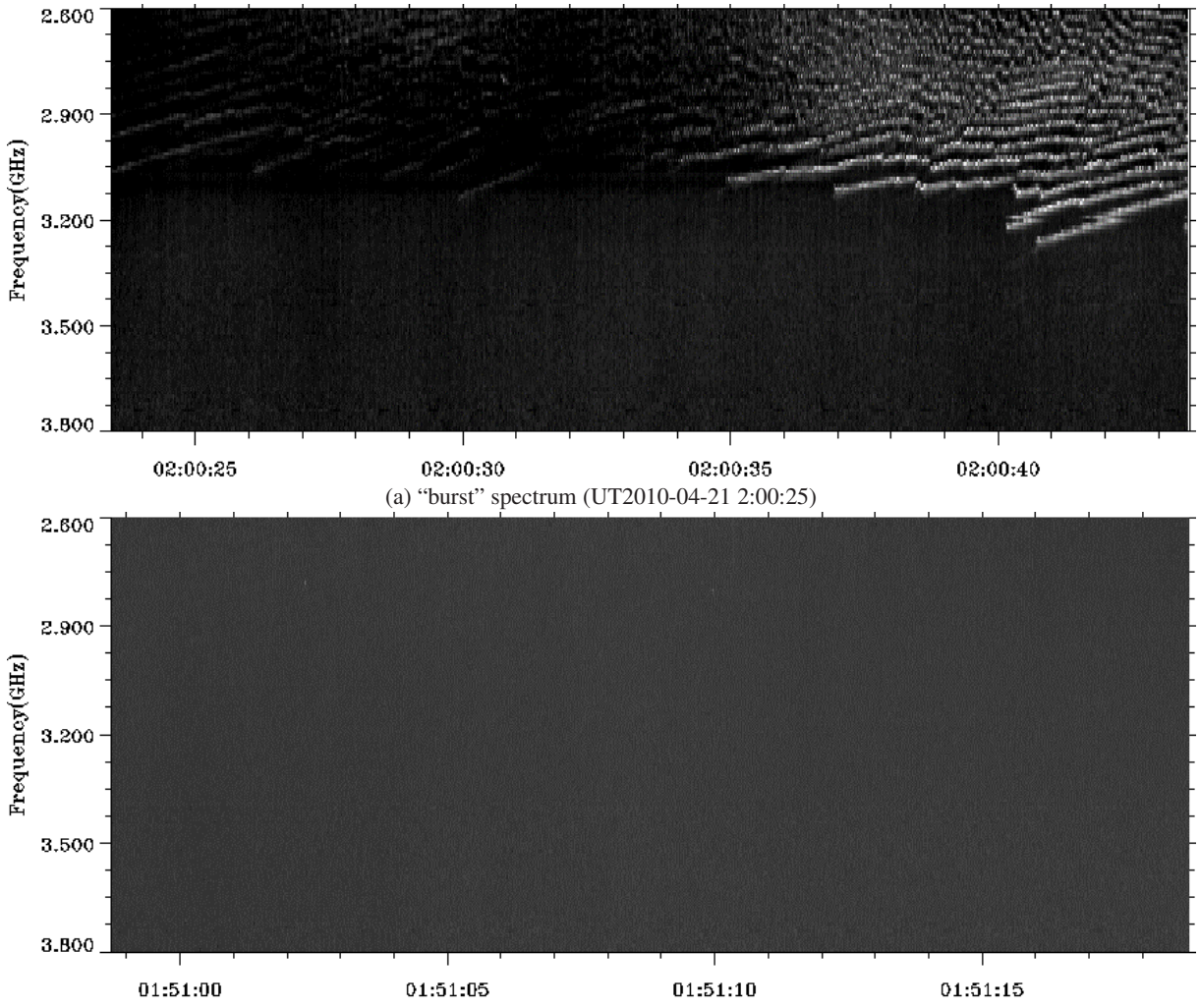
$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i), \tag{2}$$

$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o), \tag{3}$$

$$\begin{aligned} c_t &= f_t * c_{t-1} + i_t * \tilde{c}_t, \\ \tilde{c}_t &= \tanh(W_{cx}x_t + W_{ch}h_{t-1} + b_c), \end{aligned} \tag{4}$$

$$h_t = o_t * \tanh(c_t), \tag{5}$$

where σ is the logistic sigmoid function, and i, f, o, c and $\tilde{c}$ are the input gate, forget gate, output gate, cell and cell



(a) "burst" spectrum (UT2010-04-21 2:00:25)

(b) "non-burst" spectrum (UT2010-04-21 1:51:00)

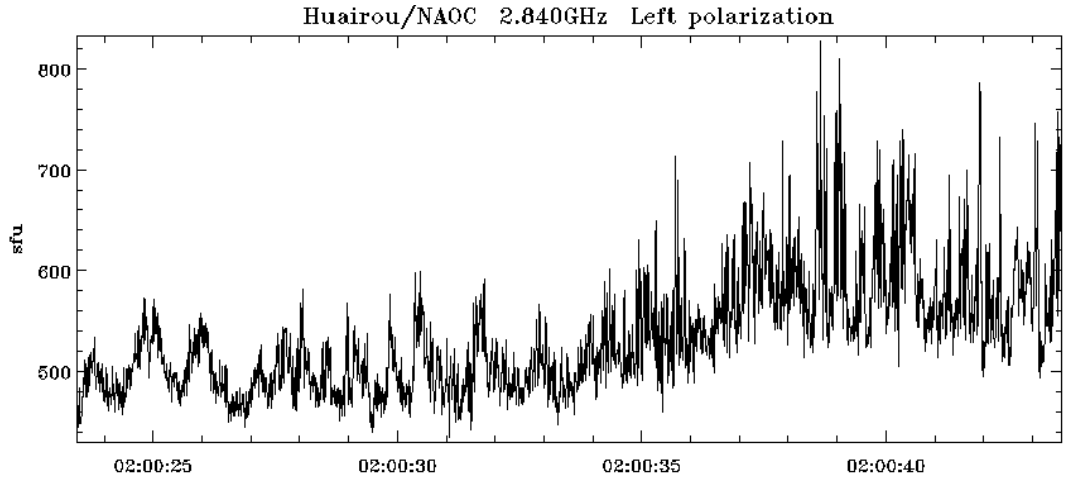
Fig. 4 Solar radio spectrums from the SBRS.

input activation vectors, respectively, all of which are the same size as the cell output activation vector h. W represents weight matrices and b refers to bias vectors. The operator "$*$" indicates the element-wise vector product.

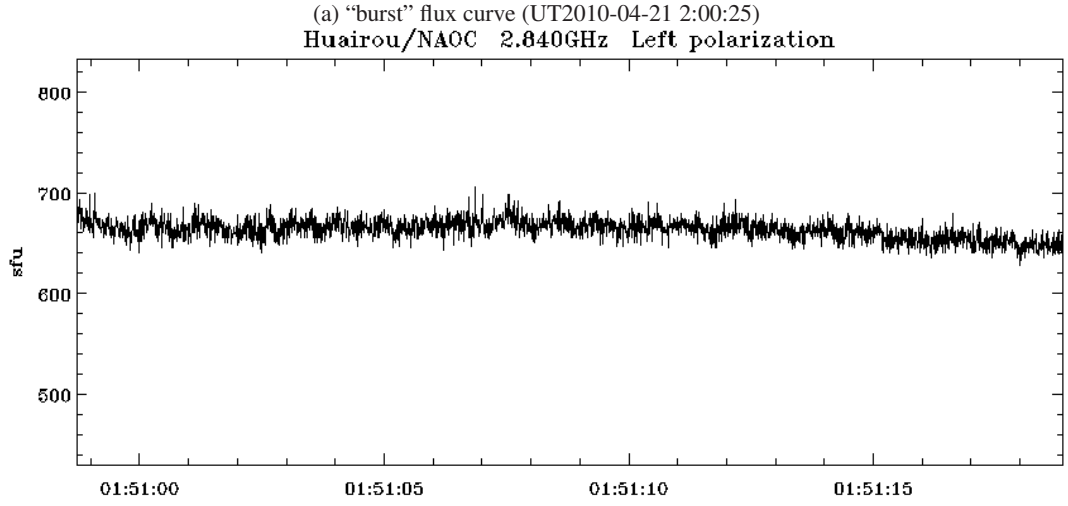
The first step in LSTM is to decide what information we are going to throw away from the cell state. This decision is made by a sigmoid layer called the "forget gate." As described in Equation (1), it looks at h_{t-1} and x_t, and outputs a number between 0 and 1 for each element in the cell state C_{t-1}. "1" means that we completely keep the history of that element while "0" signifies that we completely get rid of the history of that element. The second step of LSTM is to decide what information we will update into the cell state. This decision is made by a sigmoid layer called the "input gate," as given in Equation (2). It looks at h_{t-1} and x_t, and outputs a number between 0 and 1. Next, it will be sent to a tanh layer which creates a vector of new candidate values, (c_t), that could be added to the new state c_t as given in Equation (4). Therefore, the input gate decides how much each element of the state (candidate state) is updated.

After the two steps above, the cell state c_t is updated by both the old state c_{t-1} and a new candidate state ($\tilde{c_t}$) via Equation (4). In Equation (4), old state c_{t-1} is scaled by forget gate f_t, indicating how much we forget the history; candidate state ($\tilde{c_t}$) is scaled by input gate i_t, controlling how much each state value of ($\tilde{c_t}$) is updated into the new state.

Finally, we decide what we are going to output. This output will be based on the current cell state, but will be a filtered version as described in Equation (5). First, we run a sigmoid layer which decides which parts of the cell state we are going to output. The sigmoid layer is named "output gate," o_t given in Equation (3). Then, we put the cell state c_t through tanh and multiply it by output gate o_t, so that we only output the parts we decide to.



(a) "burst" flux curve (UT2010-04-21 2:00:25)



(b) "non-burst" flux curve (UT2010-04-21 1:51:00)

Fig. 5 Solar radio flux curves.

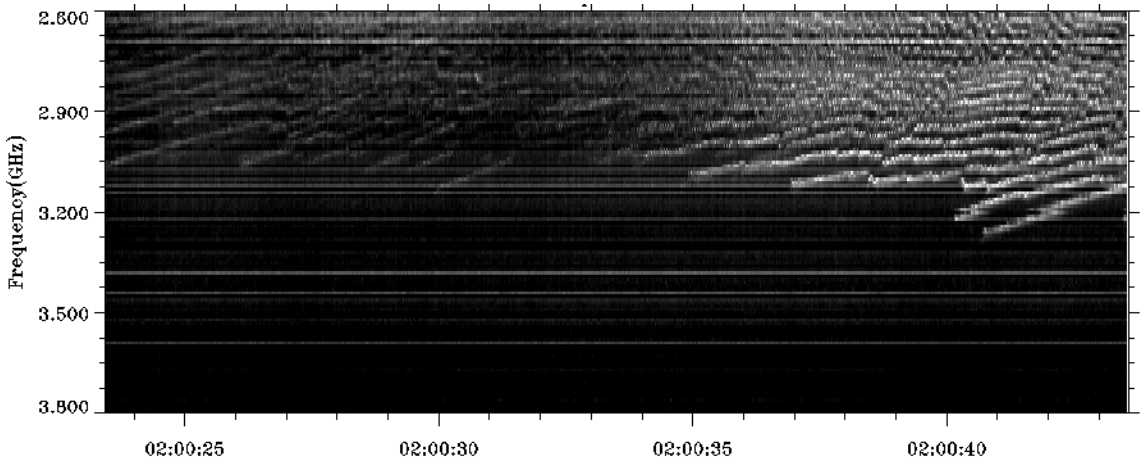
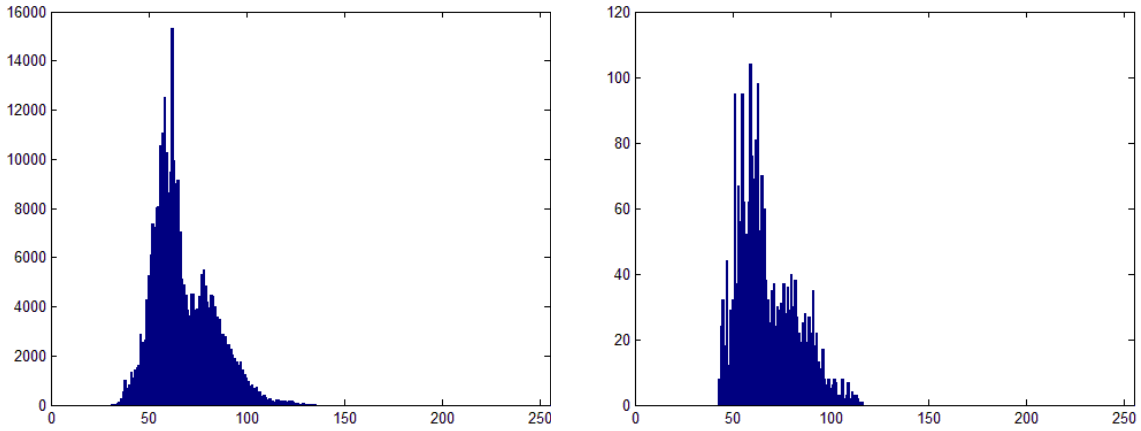


Fig. 6 Original spectrum without channel normalization (UT2010-04-21 2:00:25).

3 PRE-PROCESSING ALGORITHM FOR A SOLAR RADIO SPECTRUM

The SBRS has 120 frequency channels that can monitor a solar burst simultaneously. The digital signal representing solar radiation flux is recorded. There is a variety of timescales of recorded data files, such as 0.2 s and 8 ms. They have the same data length. Each data file can produce two spectrums, left polarization and right polarization



(a) Original (b) Down-sampling

Fig. 7 Histograms of an original spectrum and a down-sampled one.

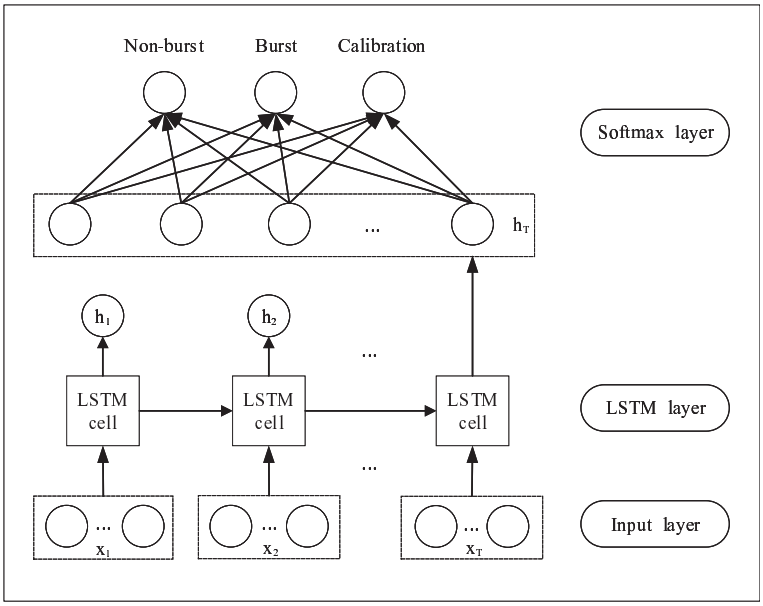


Fig. 8 The proposed network for solar radio spectrum classification.

spectrums, each of which has size 120×2520. A solar radio spectrum represents solar radio flux variation during a certain time interval and across multiple frequency channels as illustrated in Figure 4. The horizontal axis and vertical axis are time and frequency, respectively. For a single frequency channel, we can draw a profile of solar radio flux as shown in Figure 5, where the vertical axis gives the intensity of solar radio flux. From Figure 4, a solar radio burst spectrum is illustrated by intensity changes in an image, representing radio flux variation, while a static one has no obvious variation of intensity. The spectrums given in Figure 4 have undergone a series of post-processing procedures, including channel normalization, wavelet-based denoising and enhancement (Yan et al. 2002). However, the original spectrum contains various noises which may diminish valuable information in a spectrum. As shown in Figure 6, the original spectrum of Figure 4(a) is contaminated by strong noises, where the most obvious one is the stripe-like noise which is caused by the nonuniform channel effect of the receiver. Usually, the signal is covered by strong noise, so a series of image processing procedures is performed on the original spectrum. In our previous work on channel normalization, wavelet-based denoising was proposed to enhance solar burst signals and compress noise, which achieved satisfactory results (Yan et al. 2002). In this work, regarding the task of classification, the preprocessing, including normalization and down-sampling, is performed on the collected spectrums before the training model.

3.1 Channel Normalization

To reduce the channel effect, we introduce a channel normalization method, which is formulated as

$$g = f - f_{\mathrm{LM}} + f_{\mathrm{GM}}\,, \tag{6}$$

where f gives the data matrix of a spectrum, g is the processed result of f after channel normalization, and f_{LM} and f_{GM} denote the local mean and global mean, respectively. The local mean f_{LM} gives the mean of each channel. f_{GM} is the mean of the whole matrix computed from all channels. f_{LM} reduces the effect of uneven channel gain (the channel effect as mentioned above), while f_{GM} is provided to add a global background for all channels. The main purpose of channel normalization is to reduce the stripe-like noise, so that a burst hiding behind stripe-like noise becomes more identifiable. For example, an original spectrum containing annoying stripe-like noise as depicted in Figure 6 becomes more identifiable after performing the channel normalization, as illustrated in Figure 4(a).

3.2 Down-sampling

A spectrum used in this study originally has high resolution. However, the image pattern in a spectrum is very simple. There is high redundancy in a spectrum. As examined from the point of view of a stochastic process, a spectrum is an observation of a discrete stochastic process. Each column of a spectrum is a random variable. The correlation of a stochastic process can be measured by a correlation coefficient (Szegedy et al. 2016; Simonyan & Zisserman 2014), which is computed by

$$\gamma(n, n+\tau) = \frac{\varphi[(x(n)-\mu_1)\cdot(x(n+\tau)-\mu_2)]}{(\sigma_1\cdot\sigma_2)}\,, \tag{7}$$

where $x(n)$ and $x(n+\tau)$ represent two random variables from the same stochastic process, μ_1 and μ_2 give the means of $x(n)$ and $x(n+\tau)$ respectively, σ_1 and σ_2 are standard deviations of $x(n)$ and $x(n+\tau)$ respectively, and φ is the ensemble average operator. Here, $\gamma(n, n+\tau)$ depends not only on τ but also on n. In other words, a spectrum is not a collection of samples from a wide-sense stationary stochastic process. Consequently, for a given τ, $\gamma(\tau)$ is a random variable instead of a constant. Moreover, we can treat each row of a spectrum as a random variable, and the correlation coefficient can be computed in the same way. We calculate $\gamma(\tau)$ with different values of τ. The average correlation coefficients for all collected spectrums are listed in Table 1.

From Table 1, the correlation coefficient is close to 1 for (a), indicating high correlation exists in a spectrum along the horizontal axis, i.e., the time axis. In addition, it does not vary a lot as τ changes. Therefore, the spectrum can be further down-sampled to reduce data volume while its statistical properties do not change very much.

Table 1 Correlation Coefficients ($\gamma(\tau)$) of Solar Radio Spectrums

	(a) $\gamma(\tau)$ Along Horizontal Axis	(b) $\gamma(\tau)$ Along Vertical Axis
$\tau = 1$	0.9978	0.8343
$\tau = 10$	0.9829	0.7738
$\tau = 20$	0.9771	0.7660
$\tau = 30$	0.9753	0.7096
$\tau = 40$	0.9750	0.6650

The original spectrum is down-sampled into a 120×120 image using the nearest neighbor sampling method. In Figure 7, histograms of the original spectrum and the down-sampled one are illustrated. It can be observed that image characteristics do not change much after down-sampling.

After the pre-processing above, we have a collection of spectrums for establishing a database for training a classifier. The database will be presented in detail in the experimental results section.

4 PROPOSED MODEL FOR SOLAR RADIO SPECTRUM CLASSIFICATION

The LSTM network consists of LSTM cells, as depicted in Figure 3. Each cell controls how much the current input comes in, how much and for how long the historical data remain, and how the current status outputs through the three gates, input, forget and output.

The architecture of the proposed LSTM model for spectrum classification is illustrated in Figure 8, where $(x_1, x_2, \cdots, x_T)$ is a time series consisting of the columns of a spectrum. As mentioned above, a spectrum is a time series representing solar flux variation over time. For this reason, we can treat a spectrum as a time series. Thus, the LSTM is employed to learn the representation of a spectrum for classification. As shown in Figure 8, the proposed LSTM model consists of input layer, LSTM layer and softmax layer.

In the input layer, the vectors $(x_1, x_2, \cdots, x_T)$ are given by the columns of a spectrum. They are fed into the LSTM cells one by one in order. LSTM cells receive a time series x_i, and process it according to Equation (3) to Equation (7). The recurrent concept in an LSTM network means that each LSTM cell does not output immediately responding to current input, but keeps silent until its time slot comes. The time slot is scheduled by a time step which is manually tuned by several rounds of experiments. Then, a softmax layer is stacked upon the LSTM layer. It takes the LSTM output as the input, and outputs the spectrum type. In our case, the input image has size 120×120. The

time step is 120 so that each input image has an output to the softmax layer.

4.1 LSTM Layer

As already mentioned, LSTM has many variants, such as LSTM with added peephole connection (Cho et al. 2014) and the Gated Recurrent Unit (GRU) (Graves & Jaitly 2014). In Gers et al. (2002); Greff et al. (2017), readers can find more about these LSTM variants. In this work, the initial one in Jozefowicz et al. (2015) is used. The structure of a basic LSTM cell is illustrated in Figure 2.

In an LSTM cell, a memory block is provided to store the current status of the network so that this status can be kept for the next time step. In this way, the LSTM layer can explore interactions and correlations of a sequential input $x_1, x_2, \cdots, x_T$. Assuming $f_{LR}(\cdot)$ is the mapping function of an LSTM module and h_T represents the hidden state of an LSTM cell at time T, the output of an LSTM layer is expressed as

$$h_T = f_{LR}(x_1, x_2, \cdots, x_T), \tag{8}$$

where $\{x_i, i = 1, \ldots, T\}$ is a time sequence corresponding to a spectrum, with the subscript i giving the time index. The LSTM unit f_{LR} is applied to $\{x_i, i = 1, \ldots, T\}$ to produce an output h_T. Through Equation (8), the highly compressed representation of a spectrum can be learned. Afterwards, this representation is fed into a softmax layer, where a classifier is trained by supervised learning for classification. Assuming $\hat{L}$ is the output of the classifier and $\varphi[\cdot]$ is the function that performs classification, then the whole process of spectrum classification can be described by

$$\hat{L} = \varphi[f_{LR}(x_1, x_2, \cdots, x_T)]\,. \tag{9}$$

4.2 Softmax Layer

For the task of spectrum classification, a softmax layer is stacked on top of the LSTM layer, which is defined as

$$\mathrm{softmax}(z_i) = \frac{\exp(z_i)}{\sum_j \exp(z_i)}\,. \tag{10}$$

The output of the LSTM layer h_T firstly goes through a fully-connected network with three outputs as demonstrated in Figure 8. This process can be described by

$$z = W_s h_T + b_s\,, \tag{11}$$

where W_s and b_s are the weights and bias respectively of the fully-connected layer connecting LSTM output and the softmax function. Then, z is fed into the softmax expression, Equation (10), to output the probabilities of z belonging to the given three classes. This process is demonstrated in Figure 9, where the input is a vector, the dimension of which is dependent on the number of classes. The output must be less than 1 so that it can be interpreted as a probability value.

5 EXPERIMENTAL RESULTS AND ANALYSES

To evaluate the proposed LSTM model for spectrum classification, we implement it on the TensorFlow platform. It is worth pointing out that a database of solar radio spectrums is established for validating our model, which can be accessed via `http://www.deepsolar.org.cn`.

5.1 Solar Radio Spectrum Database

In this work, we use the spectrums provided by the SBRS in the Huairou Solar Observing Station (HSOS), National Astronomical Observatories, Chinese Academy of Sciences. The SBRS is designed to acquire dynamic spectrograms of solar microwave bursts across the frequency range 0.7−7.6 GHz. The SBRS is comprised of five "component spectrometers," working at five wavebands, 0.7−1.5, 1.0−2.0, 2.6−3.8, 4.5−7.5 and 5.2−7.6 GHz, respectively. These five spectrometers work simultaneously to give a full view of solar radio bursts. Please refer to Fu et al. (2004) for more details.

A lot of fine structures of microwave bursts were recorded by the SBRS from 1995 to 2001. Some of them are exhibited in Figure 10. The burst types were basically named after their morphologies. The statistics indicate that the percentage of solar radio bursts is tiny among all data. We have recorded millions of microwaves in total from 1995 to the end of 2001. However, only hundreds of them are "bursts" as shown in Table 2.

With the assistance of experts in solar astronomy, we finally established a dataset, containing 8816 spectrums which are divided into six categories (0: no burst or hard to identify, 1: weak burst, 2: moderate burst, 3: large burst, 4: data with interference, 5: calibration). Table 3 tells the number of spectrums for each type in the database. Since our task is to distinguish the bursts from the others, we combine labels for burst types (1, 2, 3), and relabel all spectrums into three categories, "burst," "non-burst," and "calibration." Table 4 provides the number of spectrums in each category. A "burst" spectrum at least contains a detectable solar radio burst during its lifetime, while a "non-burst" spectrum has no identifiable burst during its lifetime. The "calibration" spectrum usually contains a step calibration signal. It is much easier to identify.

An original spectrum has size 120×2520. The horizontal axis is time and the vertical axis is frequency. After pre-processing as described in Section 3, the new spectrum has size 120×120.

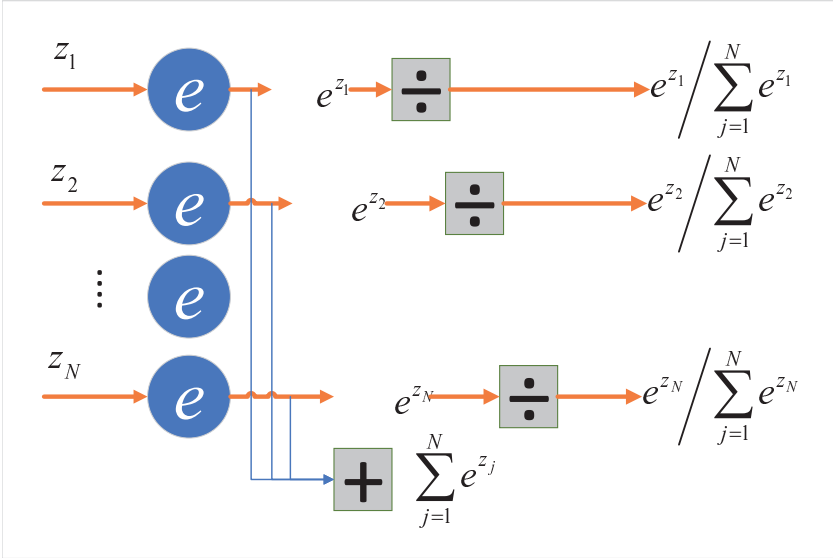


Fig. 9 A schematic diagram of the softmax layer for classification.

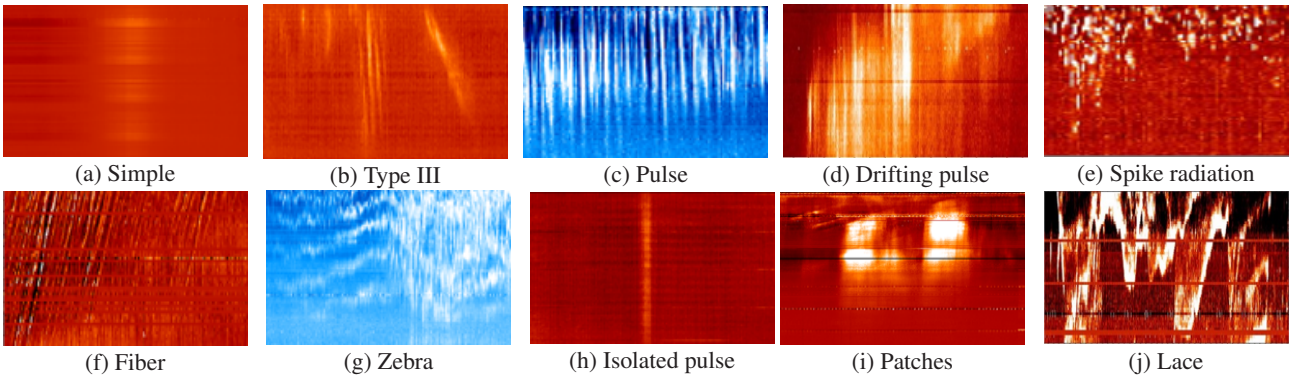


Fig. 10 Examples of solar radio spectrums.

Table 2 Statistics on Solar Radio Bursts from the SBRS

Freq. range (GHz)	0.5−1.5	1.0−2.0	2.6−3.8	4.5−7.5	5.2−7.6
Number of bursts	108	526	921	233	550

Table 3 Statistics on Spectrum Types

Categories	0	1	2	3	4	total
Number of spectrums	6670	618	268	272	570	8816

0: no burst or hard to identify; 1: weak burst; 2: moderate burst; 3: large burst; 4: data with interference; 5: calibration.

Table 4 Number of Entries in Database with Three Classes

Spectrum types	non-burst	burst	calibration	total
Number of spectrums	6670	1158	988	8816

5.2 Evaluation Metrics

To evaluate a classifier, many evaluation indexes/metrics have been provided, e.g., accuracy, error rate, precision and recall. They are defined on the basis of the following four basic terms, listed in Table 5 for comparison.

Table 5 Definitions of TP, FP, TN and FN

Positive	TP	FP
Negative	FN	TN

i) True positive (TP): if a positive instance is successfully predicted to be in a positive class;
ii) False negative (FN): if a positive instance is wrongly predicted to be in a negative class;
iii) False positive (FP): if a negative instance is predicted to be in a positive class;
iv) True negative (TN): if a negative instance is successfully predicted to be in a negative class.

From these four terms, the percentage/ratio of TP among all positive instances (TP+FN) is named the true positive rate (TPR), i.e.,

$$\mathrm{TPR} = \mathrm{TP}/(\mathrm{TP}+\mathrm{FN}), \tag{12}$$

representing the percentage of positive instances which are successfully retrieved (correctly classified into positive class) within all positive instances. Similarly, the false positive rate (FPR) is defined as

$$\text{FPR} = \text{FP}/(\text{FP}+\text{TN}), \tag{13}$$

indicating the percentage of negative instances which are wrongly decided as positive instances within the negative instance category. Let P and N represent the number of positive instances and negative instances, respectively. The accuracy index is defined as

$$\text{Accuracy} = (\text{TP}+\text{TN})/(\text{P}+\text{N}), \tag{14}$$

the precision or precision ratio is defined as

$$\text{Precision} = \text{TP}/(\text{TP}+\text{FP}), \tag{15}$$

and the recall or recall ratio is defined as

$$\text{Recall} = \text{TP}/(\text{TP}+\text{FN}) = \text{TP}/\text{P} = \text{sensitive}. \tag{16}$$

From the definitions above, accuracy gives the percentage of instances successfully classified in all instances (P+N is the total number of instances). The numerator is composed of all instances classified correctly, no matter if positive instance or negative instance, while the denominator consists of all instances. Generally, the higher the accuracy, the better the classifier. Precision (precision ratio) is only associated with a certain class, where the numerator is the number of instances which are classified correctly, and the denominator is the total number of instances which are classified into this class. Recall ratio measures the percentage of positive instances that are retrieved, different from precision, and the denominator is the number of positive instances.

5.3 Model Evaluation

To evaluate a model, the dataset is split into two parts, a training set and a testing set. For training, 800 "burst," 800 "non-burst" and 800 "calibration" cases are randomly selected each time from the dataset to form a training set. A classifier is trained on this training set with the input being spectrums and their labels. The rest of the dataset forms the testing set. For comparison, we compare the proposed LSTM model with the previous DBN model. In order to efficiently assess the proposed LSTM model, we compare it with the previous DBN model (Chen et al. 2016), CNN model (Chen et al. 2017b) and multimodal model (Chen et al. 2015; Ma et al. 2017). It is also compared with a traditional model of PCA+SVM (Chen et al. 2016). For fair comparison, we follow the evaluation indexes in the benchmarks (Chen et al. 2016, 2015; Ma et al. 2017; Chen et al. 2017b), TPR and FPR indexes as defined in Equation (12) and Equation (13) respectively, to evaluate the proposed LSTM model. TPR indicates the percentage of positive instances that are successfully classified. Thus, a larger TPR value represents a more accurate classifier. FPR describes the percentage of negative samples which are wrongly assigned to be in the positive category among all negative samples. Thus, a smaller FPR value gives a better classifier.

In our case, we have three types of spectrum, so TPR and FPR are computed independently for each type. In addition, here an interclass imbalance is an issue, where there is only a small percentage of "burst" samples. For an interclass imbalance problem, the TRP and FPR indexes are better than the accuracy index for describing the performance of classification. There are two reasons for the shortage of "burst" samples. The first is that solar burst events are sparse among all recorded data. The other is that labeling is time consuming. Especially for spectrum labeling, an expertise in solar astronomy is required.

In our case, only one LSTM layer is used since only a small dataset is available. Too many layers would result in an over-fitting problem for lack of training samples. After several rounds of training and testing, the statistics for TPR and FPR are listed in Table 6. From analyzing Table 6, we can draw the following conclusions:

1) All of the five models from deep learning perform well on spectrum classification, while the method of PCA+SVM which was workable for a general image fails on a solar radio spectrum.
2) For the "calibration" type, all deep learning models achieve good performance. The TPR of all models is around 95%, which means 95% of data are classified correctly. The FPR of all models is very small, at less than 3.2%, which means only 3.2% of other types of data are wrongly classified as "calibration."
3) For the "burst" type, the proposed LSTM model achieves the highest TPR, up to 85.4%, meaning 85.4% of "burst" data are correctly identified by the LSTM model. This means we can accurately distinguish "burst" from others. This is most important in our daily task of archiving and reporting "burst" from massive datasets. For the "burst" type, the FPR of the LSMT model is 6.7%. This indicates that 6.7% of other types of spectrum are wrongly classified as "burst." From the definition of FPR, the smaller the FPR is, the better the performance that is achieved. Thus, the LSTM model is the best among all compared models regarding FPR.
4) For the "non-burst" type, LSTM is the best among all compared models with respect to TPR.
5) Comparing "burst" and "non-burst," both TPR and FPR (larger TPR and smaller FPR) are better for "non-

Table 6 Performance Comparisons

	TPR(%)	FPR(%)	TPR(%)	FPR(%)	TPR(%)	FPR(%)	TPR(%)	FPR(%)	TPR(%)	FPR(%)	TPR(%)	FPR(%)
Burst	85.4	6.7	83.8	9.4	82.2	22.5	70.9	15.6	67.4	13.2	52.7	26.6
Non-burst	92.3	8.2	89.7	8.7	83.3	9.6	80.9	13.9	86.4	14.1	0.1	16.6
Calibration	96.2	0.9	100	0.7	92.5	1.7	96.8	3.2	95.7	0.4	38.3	72.2

burst." The reason is that "non-burst" has a much simpler image pattern than "burst" as shown in Figure 4. Thus, it is easier to identify than "burst."

6) For all deep learning models, the achievement on "calibration" is the best compared to the other two types, "burst" and "non-burst." The reason is that "calibration" is the simplest compared to the other two types. In fact, a "calibration" spectrum is almost a step signal spread over all frequency channels. It is very special relative to the other two classes.
7) For the proposed LSTM model, we can observe that the TPR of "burst" is smaller than that of "non-burst" and "calibration." The reason is twofold. First, "burst" is more complex than the two others regarding image structure, so it is more difficult to identify. Second, compared to "non-burst" and "calibration," "burst" is much more plentiful and variable, which makes the learned model unable to handle variational situations.

The gain of the proposed LSTM model may come from three aspects. First, a spectrum is reorganized into a time series so that its inner structure can be exploited during classification. Second, the LSTM module can efficiently learn the relations and interactions of a time series to generate more representative features of a spectrum. Third, pre-processing further enhances the distinguishable characteristics of a spectrum, making it easier to identify.

6 CONCLUSIONS

This paper makes efforts towards the classification of solar radio spectrums by employing an LSTM network. A spectrum is organized into a time series to be fed into the LSTM module to explore the inner relations and interactions of a time series. Experimental results demonstrated the superiority of the proposed LSTM model on spectrum classification beyond the benchmarks, which are all static models related to instant input and instant output without the context of the input. It can be explained in that this proposed time series model can generate more representative and recognizable features of a spectrum compared to the benchmarks.

Acknowledgements This work was partially supported by the National Natural Science Foundation of China (Grant Nos. 61572461, 11790305, 61811530282, 61872429, 61661146005 and U1611461) and CAS 100-Talents (Dr. Xu Long).

References

Bengio, Y. 2009, Foundations and Trends® in Machine Learning, 2, 1

Chen, L., Zhang, H., Xiao, J., et al. 2017a, in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 6298

Chen, S., Xu, L., Ma, L., et al. 2017b, in Multimedia & Expo Workshops (ICMEW), 2017 IEEE International Conference on, IEEE, 198

Chen, Z., Ma, L., Xu, L., Tan, C., & Yan, Y. 2016, Multimedia Tools and Applications, 75, 2859

Chen, Z., Ma, L., Xu, L., Weng, Y., & Yan, Y. 2015, in Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on, IEEE, 1035

Cho, K., van Merrienboer, B., Gulcehre, C., et al. 2014, arXiv:1406.1078

Collobert, R., Weston, J., Bottou, L., et al. 2011, Journal of Machine Learning Research, 12, 2493

Fu, Q., Ji, H., Qin, Z., et al. 2004, Sol. Phys., 222, 167

Gers, F. A., Schmidhuber, J., & Cummins, F. 1999, Learning to Forget: Continual Prediction with LSTM (IET)

Gers, F. A., Schraudolph, N. N., & Schmidhuber, J. 2002, Journal of Machine Learning Research, 3, 115

Graves, A., & Jaitly, N. 2014, in International Conference on Machine Learning, 1764

Graves, A., Mohamed, A.-r., & Hinton, G. 2013, in Acoustics, Speech and Signal Processing (icassp), 2013 IEEE International Conference on, IEEE, 6645

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. 2017, IEEE Transactions on Neural Networks and Learning Systems, 28, 2222

Guillaumin, M., Verbeek, J., & Schmid, C. 2010, in Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, 902

Hinton, G. E. 2012, in Neural Networks: Tricks of the Trade (Springer), 599

Hinton, G. E., Osindero, S., & Teh, Y.-W. 2006, Neural Computation, 18, 1527

Hinton, G. E., & Salakhutdinov, R. R. 2006, Science, 313, 504

Hochreiter, S., & Schmidhuber, J. 1997, Neural Computation, 9, 1735

Jolliffe, I. 2011, in International Encyclopedia of Statistical Science (Springer), 1094

Jozefowicz, R., Zaremba, W., & Sutskever, I. 2015, in International Conference on Machine Learning, 2342

LeCun, Y., Boser, B., Denker, J. S., et al. 1989, Neural Computation, 1, 541

Ma, L., Chen, Z., Xu, L., & Yan, Y. 2017, Pattern Recognition, 61, 573

Mikolov, T., Karafiát, M., Burget, L., Černockỳ, J., & Khudanpur, S. 2010, in Eleventh Annual Conference of the International Speech Communication Association

Mohamed, A.-r., Dahl, G. E., Hinton, G., et al. 2012, IEEE Trans. Audio, Speech & Language Processing, 20, 14

Ngiam, J., Khosla, A., Kim, M., et al. 2011, in Proceedings of the 28th International Conference on Machine Learning (ICML-11), 689

Ren, Z., Wang, X., Zhang, N., Lv, X., & Li, L.-J. 2017, arXiv:1704.03899

Simonyan, K., & Zisserman, A. 2014, arXiv:1409.1556

Sohn, K., Jung, D. Y., Lee, H., & Hero, A. O. 2011, in Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2643

Sundermeyer, M., Schlüter, R., & Ney, H. 2012, in Thirteenth Annual Conference of the International Speech Communication Association

Suykens, J. A., & Vandewalle, J. 1999, Neural Processing Letters, 9, 293

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. 2016, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2818

Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P.-A. 2008, in Machine Learning, 25th International Conference on (New York, NY, USA: ACM), 1096

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P.-A. 2010, Journal of Machine Learning Research, 11, 3371

Wold, S., Esbensen, K., & Geladi, P. 1987, Chemometrics and Intelligent Laboratory Systems, 2, 37

Yan, Y., Tan, C., Xu, L., et al. 2002, Science in China A: Mathematics, 45, 89

Yu, X., Xu, L., Ma, L., Chen, Z., & Yan, Y. 2017, in Multimedia & Expo Workshops (ICMEW), 2017 IEEE International Conference on, IEEE, 519