Sensors-based Human Activity Recognition with Convolutional Neural Network and Attention Mechanism

Wenbo Zhang, Tao Zhu, Congmin Yang and Jiyi Xiao

School of Computer
University of South China
Hengyang, Hunan 421001, China
zwb1995418@gmail.com, tzhu@usc.edu.cn,
congminyang@163.com, 532735539@qq.com

Huansheng Ning
School of Computer and Communication Engineering
University of Science & Technology Beijing
Beijing, China
ninghuansheng@ustb.edu.cn

Abstract—Recently, Human Recognition Activity (HAR) has received more and more attention. At present, Recurrent neural networks (RNN), long short-term memory in particular, are main approaches in HAR. However, RNN suffers from the fact that it cannot process sequences in parallel and longer sequences cannot be remembered well. Therefore, this paper applies the attention mechanism to explore the relevant time context, and proposes a new model, named DeepConvAttn. DeepConvAttn is based on a well-known deep learning model, DeepConvLSTM. These two models are compared in experiments, and the results show DeepConvAttn is better than DeepConvLSTM on two popular HAR benchmark datasets.

Keywords—Human Activity Recognition, Attention Mechanism, Deep Learning

I. INTRODUCTION

In fields such as health, remote monitoring, game, automatic driving, video retrieval, gait analysis, human-computer interaction and so on, activity recognition [1-3] is a key supporting technology. In order to analyze and model the time series data, the time context of each sensor reading needs to be taken into account, which is usually implemented by sliding window method [4]. The time context of each sensor reading is usually simulated by a fixed size window. Sliding window segmentation plays an important role in many deep learning methods. For example, sliding window segmentation is used to map the time series data to fixed-length vectors, which are then the inputs of convolutional neural network. The length of the window is the key parameter, and is usually determined by prior knowledge. It is found that Recurrent Neural Networks (RNNs) [5, 6] including its variants of Long Short-Term Memory (LSTM) [7] and Gate Neural Networks (GRU) [8] can also get a very good recognition result in HAR.

DeepConvLSTM [9] is a recently proposed model which combines CNN and LSTM for activity recognition. DeepConvLSTM has obtained good performance in various other fields, such as image recognition or speech recognition. So DeepConvLSTM was later applied to activity recognition and got similar good results. However, LSTM has two limitations. First, the output of the next time point must be predicted based on the output of the previous time point, which cannot be processed in parallel [10, 11]; Second, the proposal of LSTM solves the gradient disappearance problem of RNN to some extent [7]. Nevertheless, due to the limited memory capacity of LSTM unit, it still forgets information from a long time ago.

In this paper, the attention mechanism is applied to the problem of HAR. Essentially, with the help of the attention mechanism, a set of weights from the input data sequence which represent the relative importance between each sensor reading was learned by the model. The performance of the model is explored by modifying the DeepConvLSTM and adding a layer of attention. The evaluation results of our model on the benchmark dataset show a significant improvement in performance compared with DeepConvLSTM.

Section 2 gives the background of human activity recognition; Section 3 introduces the proposed new model DeepConvAttn; Section 4 presents the experiments settings and results. The last section summarizes this paper.

Figure 1. Architecture of Baseline CNN, \( t \) denotes the time steps, \( d \) denotes the number of sensor channels, \( f \) denotes the number of filters, \( n \) denotes number of neural.
layers of the convolutional layer plus two fully connected layers, and finally classified by softmax (see Error! Reference source not found.). However, CNNs is only suitable for analyzing continuous data because it uses a sliding window segmentation method to segment fixed-size analysis frames from the input sequence of sensor data.

Not only has CNNs realized the powerful HAR system, but also RNNs has been applied in HAR. Most variants of RNNs [7, 8], such as Baseline LSTM, LSTM layer and fully connected layer interspersed, are finally classified by softmax (see Error! Reference source not found. 2). The LSTM merges specific gates into a single cell, and updates or forgets past memories according to the current input, the output of the feedback and the internal state.

One of the resent models is the combination of CNNs for feature extraction and LSTMs for sequence-to-sequence learning, known as DeepConvLSTM [9] (see Error! Reference source not found. 3). In this architecture, the input is a data window size containing 1 second (i.e., 24 frames), which is passed into a convolutional layer with four consecutive layers interspersed with the ReLU nonlinear activation function. The four convolutional layers act as automatic feature extraction. By adding dimensions, the time series input by the sensor is converted into data represented in two dimensions, in which the first dimension represents the time step length and the second dimension represents the number of features of each time step length.

After the 4-layer convolution, the 24-time steps become 8-time steps, and the dimension is \(8 \times d \times f\), where \(d\) denotes the number of features in each sample, namely the channel. \(f\) denotes the number of filters. For each convolutional layer, the number of filters is fixed at 64, and \(64 \times d \times f\) of 8 time steps are output. Then the output results were sent to the two-layer LSTM with 128 hidden units for sequential processing after one layer of dropout. Finally, the activity prediction was generated after one layer of softmax.

In the tasks of natural language processing and speech recognition [17], time series processing is required. However, the action of an activity is also time-related, so the activity is represented as a sequence and input into the model for recognition. RNN is usually used for time series processing, but it can output the value of the next time point only when seeing the output of the previous time point. Even with a bidirectional RNN, the output is time step dependent and therefore cannot be processed in parallel. This is also a flaw in the LSTM processing sequence in the DeepConvLSTM framework.

In order to solve the problem of sequential parallel processing, the CNN replacing the RNN to process sequence is studied, but it can only consider the limited information for that it collects information through a fixed size of the kernel. More information can be considered by superimposing the output of the upper layer. The advantage of the CNN is that it can be parallelized, however, the defects exist, that is to say, it needs many layers overlay to see the whole sequence information.

Therefore, this paper adopts Attention to deal with the sequence-to-sequence problem that can not only see the whole input sequence, but also perform parallel calculation to improve the operation efficiency.

III. HUMAN ACTIVITY RECOGNITION WITH CNN AND ATTENTION

A. Self-Attention Mechanism

Previous deep learning typically used a fixed-size time context to model readings from all sensors. This approach achieves good results, and such a model dominates the most challenging HAR benchmark data set (such as Opportunity dataset [18]).

The approach in this paper is to explore the attention model to automatically determine the time context associated with the sequence of modeling activities. This approach is used as a data-driven way to tune the analysis window. For natural language processing tasks in machine translation, sequence-to-sequence processing has successfully introduced the attention model [11]. The formal idea is to give the input sequence a query, key, and value through three sets of linear variations. Then, the similarity of each query for each key is calculated as a score. After softmax, the weight of the input sequence is obtained. Finally, the output sequence is obtained by the weighted sum of weight and value. See Error! Reference source not found. 4, the process of adding attention mechanism to the HAR application is illustrated. By modifying the DeepConvLSTM architecture [9], the LSTM layers are replaced by the attention layer to form the new DeepConvAttn framework.

The advantage of the Attention mechanism [11] is that it can be processed in parallel and can see the whole sequence of information directly, which completely solves the defect of LSTM processing from sequence to sequence.

The variant of self-attention is the multi-head self-attention mechanism [11], whose advantage is that different heads can pay
attention to different information. The specific process is to pass
the input sequence through the Linear Layer a few more times
to obtain groups of queries, keys and value, then the middle
process is the same as self-attention. After several groups of
output sequence are obtained, which at the same point in
time are connected, if the dimension of the connection is not what you
want, you can then convert the output into the desired
dimensions by a linear layer. And the number of groups in the
middle indicates how many heads there are.

\[
\begin{align*}
Q &= W_q^d \#(1) \\
K &= W_k^d \#(2) \\
V &= W_v^d \#(3) \\
S &= K^T Q / \sqrt{d} \#(4) \\
A &= \text{softmax}(S) \#(5) \\
O &= VA \#(6) \\
\end{align*}
\]

The DeepConvAttn architecture, due to the relatively small
dataset and the number of layers of the model with many
parameters, needs to avoid overfitting, which can be prevented
by dropout [19] and regularization [20].

IV. EXPERIMENTS

The proposed DeepConvAttn framework in this paper is
evaluated on two human activity recognition datasets and
compared with other frameworks for activity recognition. The
proposed framework provides performance reference value for
depth networks.

A. Datasets

Human activities can be classified into three categories. The
first category is periodic activities, such as walking, running and
cycling. The second category is discrete activities, such as
operation-oriented gestures, opening or closing a car door and
holding a water glass; The third category is static activities, such as
lying, standing and sitting still. Activities must be
benchmarked against datasets containing various types of
activities. In addition, manipulation-oriented activities are often
embedded in classes that contain a large number of Null.
Identifying activities embedded in Null is more challenging
because the recognition system must implicitly recognize the
starting and ending points containing the gesture data, and
finally classify them. In this paper, Opportunity dataset [18] and
Skoda dataset [21] are selected to evaluate the model Proposed
by me.

1) The Opportunity Dataset

Opportunity dataset [18] contains a complex set of natural
activities, recording the daily life scenarios of four subjects' morning activities, and incorporating the environment, objects and wearable sensors. At the time of recording, each subject underwent one practice and five daily living activities (ADL). In practice, the subjects performed 20 repetitions of the 17 predefined activities. During each ADL, subjects are required to give a loose description of the actions they perform in order to perform the activity without restriction. The dataset contains about six hours of records.

The Opportunity dataset consists of periodic, discrete, and
static activities. This data set is available in the UCIMachine
learning database and is used by many third-party publications (e.g. [14, 22, 23]).

In this paper, the same subset of the Opportunity dataset was
used to train and test the model we proposed. The practice of the
first subject and all ADL, ADL1, ADL2 and ADL3 of the second and third subjects were selected to train the model, and ADL4 and ADL5 of the second and third subjects were selected to verify the test model.

The Opportunity dataset consists of readings from motion sensors recorded when subjects perform typical daily activities. The environment sensor consists of 13 switches and 8th 3D acceleration sensors. The object sensor contains 12th 3D accelerometer and 2D gyroscope sensors. The wearable sensor contains 7th inertial measurement units, 12th 3D acceleration sensors, and 4th 3D positioning information. Each sensor reading is treated as a separate channel, and after feature selection, a 113-dimensional channel is generated. These sensors have a sampling rate of 30Hz.

Wireless sensors can lose readings due to signal connection problems, and in order to fill in the missing value, the wireless sensor data was preprocessed by linear interpolation and normalized for each channel.

Opportunity dataset contains multiple tasks. In this paper, two tasks are selected. Task A recognizes discrete gestures, which is an 18-classes problem. Task B is the recognition of movement and posture, which is a five-classes problem. Table I summarizes the activities contained by each task in the dataset, the number of activity repetitions, and the number of instances.

The entire program is trained using PyTorch deep learning framework [24]. All the experiments adopt the sliding window method to extract the processing frame of analysis. The length of the frame is set to the data window of 1 second, that is, 24 samples, and the window step is set to 12, that is, there is 50% overlap between successive frames. During the training process, the extracted frames are randomly scrambled to avoid deviation.

Crossentropy loss was used to train the model, the learning rate was set to 0.01, and momentum gradient descent [25] was used to optimize the parameters. The batch size of all experiments was set to 100 and regularized using the dropout layer to prevent overfitting of the model.

<table>
<thead>
<tr>
<th>Class Labors of the Opportunity and Skoda Datasets</th>
<th>Opportunity</th>
<th>Skoda</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gestures</td>
<td>Models of Locomotion</td>
<td></td>
</tr>
<tr>
<td>Label</td>
<td># of Instances</td>
<td># of Repetitions</td>
</tr>
<tr>
<td>Open Door1</td>
<td>1583</td>
<td>94</td>
</tr>
<tr>
<td>Open Door2</td>
<td>1685</td>
<td>92</td>
</tr>
<tr>
<td>Open Drawer1</td>
<td>897</td>
<td>96</td>
</tr>
<tr>
<td>Open Drawer2</td>
<td>861</td>
<td>91</td>
</tr>
<tr>
<td>Open Drawer3</td>
<td>1082</td>
<td>102</td>
</tr>
<tr>
<td>Open Fringe</td>
<td>196</td>
<td>157</td>
</tr>
<tr>
<td>Open Dishwasher</td>
<td>1314</td>
<td>102</td>
</tr>
<tr>
<td>Close Door1</td>
<td>1497</td>
<td>89</td>
</tr>
<tr>
<td>Close Door2</td>
<td>1588</td>
<td>90</td>
</tr>
<tr>
<td>Close Drawer 1</td>
<td>781</td>
<td>95</td>
</tr>
<tr>
<td>Close Drawer 2</td>
<td>754</td>
<td>90</td>
</tr>
<tr>
<td>Close Drawer 3</td>
<td>1070</td>
<td>103</td>
</tr>
<tr>
<td>Close Fringe</td>
<td>1728</td>
<td>159</td>
</tr>
<tr>
<td>Close Dishwasher</td>
<td>1214</td>
<td>99</td>
</tr>
<tr>
<td>Clean Table</td>
<td>1717</td>
<td>79</td>
</tr>
<tr>
<td>Drink from Cup</td>
<td>6115</td>
<td>213</td>
</tr>
<tr>
<td>Toggle Switch</td>
<td>1257</td>
<td>156</td>
</tr>
<tr>
<td>Null</td>
<td>69558</td>
<td>1605</td>
</tr>
</tbody>
</table>

2) The Skoda Dataset

The Skoda Mini Checkpoint dataset [21] describes the activities of assembly line workers in the context of automobile production. These gestures represent the gestures performed by the production car factory for quality assurance inspection and are listed in Table I.

Twenty 3D acceleration sensors were worn on both arms of a subject, and the experiment was limited to 10 sensors on the subject's right arm. The original sampling rate of the sensor set in the Skoda data set is 98Hz. For comparison with the Opportunity data set, the sampling rate is reduced to the same 30Hz. The dataset contains 10 action gestures. The duration of the recording was about three hours, there are about 70 repetitions of each gesture, and the data set was also publicly available.

<table>
<thead>
<tr>
<th>Modeling</th>
<th>Datasets</th>
<th>Opportunity</th>
<th>Skoda</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gestures</td>
<td>Locomotion</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Baseline CNN</td>
<td>0.883</td>
<td>0.878</td>
<td>0.884</td>
</tr>
<tr>
<td>Baseline LSTM</td>
<td>0.909</td>
<td>0.884</td>
<td>0.923</td>
</tr>
<tr>
<td>DeepConvLSTM</td>
<td>0.915</td>
<td>0.895</td>
<td>0.958</td>
</tr>
<tr>
<td>DeepConvAttn</td>
<td><strong>0.928</strong></td>
<td><strong>0.906</strong></td>
<td><strong>0.965</strong></td>
</tr>
</tbody>
</table>


C. Results and discussion

Considering the unbalanced distribution of the two datasets, F1 score is used to judge the performance of the model respectively. F1 score is defined as the harmonic average of precision rate and recall rate, so it is also known as equilibrium F score. Table II shows the identification results of the benchmark dataset. Obviously, for the benchmark dataset, the combination of the convolutional neural network and the attention model significantly improves the performance of the model and the operation speed, as shown in Table III.

<table>
<thead>
<tr>
<th>Modeling</th>
<th>Opportunity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline CNN</td>
<td>103.794s</td>
</tr>
<tr>
<td>Baseline LSTM</td>
<td>72.565s</td>
</tr>
<tr>
<td>DeepConvLSTM</td>
<td>96.117s</td>
</tr>
<tr>
<td>M</td>
<td></td>
</tr>
<tr>
<td>DeepConvAttn</td>
<td>56.273s</td>
</tr>
</tbody>
</table>

The attention mechanism can deal with the sequence-to-sequence problem very effectively and solve the problems that LSTM cannot parallel and DNN cannot see all the information at once. It is unreasonable that the attention mechanism can see the entire input sequence completely, because activities in the distant past have little effect on current activities. Therefore, in the paper of "Attention" proposed by Google, a future solution technology named local Attention is mentioned, which is similar to sliding window and only focuses on local sequences. In this way, the influence of past activities on current activities can be avoided.

V. CONCLUSIONS

In this paper, the advantages of the deep frame based on the combination of CNN and Attention were demonstrated. The results of DeepConvAttn were on average 1% higher than DeepConvLSTM. In terms of running time, DeepConvAttn improved data processing speeds by nearly half as much as DeepConvLSTM. The data processing speed can be greatly improved because Attention can be processed in parallel. Effective artificial feature selection after sensor fusion can avoid the trouble caused by data loss of wireless sensor. As for my future work, a lifetime learning based on the DeepConvAttn framework for active recognition of large-scale data will be studied.

REFERENCES