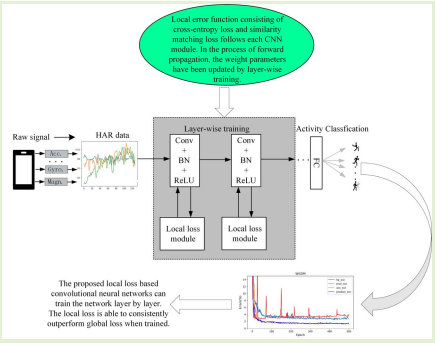
The Layer-Wise Training Convolutional Neural Networks Using Local Loss for Sensor-Based Human Activity Recognition

Qi Teng, Kun Wang, Lei Zhang, and Jun He, *Member, IEEE*

***Abstract*— Recently, deep learning, which are able to extract automatically features from data, has achieved state-of-the-art performance across a variety of sensor based human activity recognition (HAR) tasks. However, the existing deep neural networks are usually trained with a global loss, and all hidden layer weights have to be always kept in memory before the forward and backward pass has completed. The backward locking phenomenon prevents the reuse of memory, which is a crucial limitation for wearable activity recognition. In the paper, we proposed a layer-wise convolutional neural networks (CNN) with local loss for the use of HAR task. To our knowledge, this paper is the first that uses local loss based CNN for HAR in ubiquitous and wearable computing arena. We performed experiments on five public HAR datasets including UCI HAR dataset, OPPOTUNITY dataset, UniMib-SHAR dataset, PAMAP dataset, and WISDM dataset. The results show that local loss works better than global loss for tested baseline architectures. At no extra cost, the local loss can approach the state-of-the-arts on a variety of HAR datasets, even though the number of parameters was smaller. We believe that the layer-wise CNN with local loss can be used to update the existing deep HAR methods.**



***Index Terms*— Activity recognition, deep learning, convolutional neural networks, sensor, local loss.**

I. Introduction

THE development and popularity of smartphones or other various wearable devices embedded with sensors such as accelerometer, gyroscope, magnetometer have enabled researchers to collect human physiological signal for monitoring of activity of daily living (ADL). Human activity recognition (HAR) [1] based on wearable sensors has become a new research area with a wide range of real-world applications such as smart homes [2], health monitoring [3], sports tracking [4], and game console designing [5] to name but a few. The traditional machine learning approaches such as multilayer perceptions [6], support vector machine (SVM) [7], and decision trees [8] have made tremendous progress in inferring activity details. However, there is one obvious drawback to these traditional methods, which heavily rely on a heuristic or hand-crafted feature extraction. The hand-crafted features [9], which requires expert experience or domain knowledge, lead to a lower chance to build a successful recognition system for more general environments or tasks. Later, deep learning has been widely used in HAR field [5], which can automatically learn high-level or meaningful features. Even though deep learning based approaches to HAR widely outperform the state-of-the-art using non-deep learning methods, some of the challenges for HAR in ubiquitous and wearable computing still remain. In particular, the existing deep learning methods usually require a vast amount of memory resources to store the parameters of hidden layers when training deep networks, which is usually limited on smartphones or wearable devices.

For HAR, the state-of-the-art methods such as CNN [10] or DeepConvLSTM [11] tend to utilize global loss and gradient backpropagation to train neural networks, whereas it blocks the update of hidden layer weights before the forward and backward pass has completed. The phenomenon is termed as backward locking [12], [13], which impedes the

Manuscript received January 21, 2020; revised March 3, 2020; accepted March 4, 2020. Date of publication March 6, 2020; date of current version June 4, 2020. This work was supported in part by the National Science Foundation of China under Grant 61203237, in part by the Industry–Academia Cooperation Innovation Fund Projection of Jiangsu Province under Grant BY2016001-02, and in part by the Natural Science Foundation of Jiangsu Province under Grant BK20191371. The associate editor coordinating the review of this article and approving it for publication was Dr. Rosario Morello. *(Corresponding author: Lei Zhang.)*

Qi Teng, Kun Wang, and Lei Zhang are with the School of Electrical and Automation Engineering, Nanjing Normal University, Nanjing 210023, China (e-mail: leizhang@njnu.edu.cn).

Jun He is with the School of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: jhe@nuist.edu.cn).

Digital Object Identifier 10.1109/JSEN.2020.2978772

reuse of memory used to store hidden layer activation. That is to say, the backward locking prevents parallelization of the weight updates on different hidden layers, which causes the failure to release memory. The backward locking phenomenon is severe for sensors based HAR due to the limitation of memory resources on wearable devices. On the other hand, the gradient backpropagation of global loss for deep learning has been criticized by neuroscientists for its biological plausibility [14], [15]. With no use of direct backpropagation of global loss, several researches have been conducted to discover more biologically plausible deep learning approach Recently, Arild *et al.* proposed a novel approach to image classification that employs local loss and layer-wise training to avoid the backward locking problems [16]. For sensor based HAR tasks, replacing global loss with local loss could be one feasible step towards more biologically plausible or memory efficient deep learning model.

In the paper, we present a local loss based framework for deep learning based activity recognition using wearable sensing data, which can effectively alleviate memory requirement. Unlike the popular CNN or DeepConvLSTM trained with global loss, we combine local loss and layer-wise training mechanism into deep HAR classifiers that enable substantially improved memory efficiency for HAR applications. The global loss of baseline CNN is replaced with local loss throughout two different supervised loss functions. Instead of direct gradient back-backpropagation, global targets are projected into hidden layers and local loss is used to train each layer simultaneously, which make weights can be updated along with forward pass. In other words, activations do not have to be kept in memory for the backward process. To the best of our knowledge, this paper is the first that uses local error based deep neural networks for HAR in ubiquitous and wearable computing arena. We evaluate our method on five public benchmark datasets, namely UCI HAR dataset [7], Opportunity dataset [17], UniMib-SHAR dataset [18], PAMAP2 dataset [19] and WISDM dataset [20]. By outperforming the state-of-the-art methods on classification accuracy and memory efficiency without any extra cost, our results indicate the advantage of the local loss based model with regards to typical challenges of HAR in ubiquitous and wearable computing scenarios. The models' performance is analyzed in detail, which allows us to draw conclusions for related applications and future scenarios.

The paper is structured as follows. In section II, we summarize the related work of HAR. Section III presents the details of HAR with local loss. Section IV details the HAR dataset we used, experimental setup, and the experimental results. In section V, we summarize our conclusions.

II. Related Works

Initially, hand-crafted features methods were widely used in the field of HAR [21], [22], which utilized several statistical values such as variance, mean, or kurtosis to explore the time series sensor data. These features were popular due to their simplicity as well as their high performance across a variety of activity recognition problems. For example, Bao and Intille [23] proposed a novel approach that distill handcrafted features from accelerometer sensor data and then fed them to different classifiers including K-nearest neighbor (KNN), decision tree, and Nave Bayes. Anguita *et al.* [7] used a multiclass SVM with 561 hand-designed features to classify six different activities. There were several researchers using transform encoding techniques such as Fourier or wavelet methods [24] to extract features from human activity data. However, these methods using handcrafted features have to rely on laborious human intervention or domain knowledge, which fails to recognize very similar activities such as walking upstairs and walking downstairs.

In order to address the above limitation, more efforts are shifting to automatic feature extraction directly from raw sensor data. Recently, the HAR research about automatic feature extraction using deep learning has received much attention. For example, Zeng *et al.* [25] treated each dimension of the accelerometer as one channel like RGB of an image, then the convolution and pooling operation were performed separately. Jiang and Yin [10] converted the raw sensor signal into 2D signal image, and then a two layer 2D ConvNet was used to classify this signal image equaling to the desired activity recognition. Wang *et al.* [26] presented an attention based CNN, which can be used to recognize and locate the weakly labeled sensor data. However, unlike images only have spatial connection between pixels, sensor data is a time series as well. Ordóñez and Roggen [11] presented a generic deep framework composed of CNN and long-term short memory (LSTM) for activity recognition, which could fuse multimodal sensors to further improve the CNN on classification accuracy. However, all these models have to be trained with global loss, which leads to a low memory efficiency due to the backward-locking problem.

Actually, deep neural networks are usually trained with a global loss, which is not biologically plausible due to a number of reasons. Several reseaches have been devoted to train neural networks without the use of global loss. Jaderberg *et al.* [12] used synthetic gradients, instead of true backpropagated error gradients to train the hidden layers, which can avoid the backward-locking phenomenon. In the field of image processing, Belilovsky *et al.* proposed to use the layer-wise loss functions to train the network, which approaches the best results of global backprop on CIFAR-10. Gomez *et al.* [13] proposed reversible residual network (RevNet), a variant of ResNets where each layer's activations can be reconstructed exactly from the next layer's. Therefore, the activations for most layers need not be stored in memory during backpropagation. Nøkland and Eidnes [16] for the first time demonstrated that using supervised layer-wise loss functions can match state-of-the-arts on a variety of image datasets. Despite the advantage that local loss functions are more memory efficient and biologically plausible, the layer-wise deep HAR model has been poorly reported in the related literature.

III. Model

Without loss of generality, CNN is used as the basic model, because the local loss based methods used below are based on CNN. The baseline model is built as a typical deep

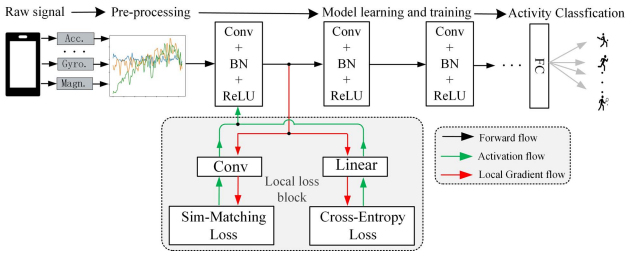


Fig. 1. Overview of the layer-wise training CNN for HAR. Local error function consisting of cross-entropy loss and similarity matching loss follows each CNN module. In the process of forward propagation, the weight parameters have been updated by layer-wise training.

CNN, which comprises convolutional layers, dense layers and softmax layers. For simplicity, the pooling layers are not used in the tested structure. Our research aims to realize layer-wise training CNN for the practical use of HAR. In order to train the network with local loss, we implement two different local learning signals to update and optimize each hidden layer. Referring to the Nøkland et at' s work [16], the learning signals directly from two separate single-layer sub-network are fed into distinct loss functions, and the framework of the model is shown in Fig.1. The standard CNN layer and two disparate loss functions compose one sub-network to train the weight parameters of each hidden layer. Along with the forward flow, all hidden layers are orderly trained one-at-a-time. Actually, here the final output layer is directly trained to minimize by conventional by the conventional loss, as shown in Algorithm 1. To avoid the use of global backpropagation across the whole network, all other layers are trained to match their associated targets, which strive to minimize the loss computed in the output layer according to the input already realized by feedforward propagation.

A. Similarity Matching Loss

In the paper, the similarity matching loss [16], [27], is formulated as follows:

$$l_s = \|S\left(C\left(X; w\right)\right) - S\left(Y\right)\|_2 \tag{1}$$

where the matrix X is the output of a forward-flow convolutional layer and the C represents a convolutional operation with kernel size 3*3, stride 1 and padding 1. In the process of minimizing the loss, the convolution parameters w can be gradually adjusted to approximate the similarity matrix of one-hot encoded label $Y = (y_1, y_2, \ldots y_n)$, which encourages neural network to learn optimal features from activity data. In particular, a standard deviation operation is enforced on the feature maps to reduce the dimension to 2D before the feature maps C(.) are fed into S(.) The S(H) denotes the adjusted cosine similarity matrix operation, which is symmetric and can be expressed as:

$$S_{ij} = S_{ji} = \frac{\left\langle \tilde{h}_i, \tilde{h}_j \right\rangle}{\left\| \tilde{h}_i \right\|_2 \left\| \tilde{h}_j \right\|_2} \tag{2}$$

where the H denotes a mini-batch of hidden layer activations, and the outputs of S(.) contain the pairwise similarities between all examples in the matrix C(.). In essence, the similarity matching loss can be considered as a supervised clustering loss [16]. The similarity matching loss is denoted as sim loss.

B. Prediction Loss

The other local loss signal is implemented by the cross entropy between a prediction of local linear classifier and the target, which is called prediction loss. The prediction loss abbreviated as l_p can be expressed as follows:

$$l_p = CrossEntropy\left(Y,\ W^T X\right) \tag{3}$$

where the W denotes a linear classifier, and the Y denotes the label matrix of one-hot encoded targets.The prediction loss is denoted as pred loss.

C. Local Loss

Finally, the local loss of each hidden layer is implement by a weighted combination between the similarity matching loss l_s and the prediction loss l_p.

$$l_{sp} = \alpha l_s + (1 - \alpha)\, l_p \tag{4}$$

where α is a positive real number and $\alpha <= 1$. The combination is denoted as predsim loss.

Algorithm 1 Local Error Model

Input:
X:The output of a forward flow on each concolutional layer;
Y:The label matrix of one-hot encoded target output;
L:The number of layers and l is certain layer
l_p, l_s, l_g, l_{sp}: prediction loss; similarity matching loss;global loss; local loss combined by l_p and l_s
$W^{(l)}_{forward\_c}$, $W^{(l)}_{forward\_f}$, $W^{(l)}_{local\_c}$, $W^{(l)}_{local\_f}$:The convolution parameters of forward flow, the full connected parameters of forward flow, the convolution parameters of local loss block, and the full connected parameter of local loss block separately.
Detach backward gradient flow from computation graph.
Initializing all weighting parameters W
while not Stop-Criterion **do**:
  for *each_layer* **in** range(L):
    computing l_p, l_s, l_{sp}
    update weight $W^{(l)}_{forward\_c}$
    if *each_layer* == L:
      computing l_g
      updating weight $W^{(l)}_{forward\_f}$
    update weights $W^{(l)}_{local\_c}$, $W^{(l)}_{local\_f}$

IV. Experiment

In this work, we performed experiments on several public HAR datasets including UCI HAR dataset, OPPOTUNITY dataset, UniMib-SHAR dataset, PAMAP2 dataset, and WISDM dataset. The CNN consisting of three convolutional layer and one fully connected layer was used as the baseline to evaluate whether the local loss can further

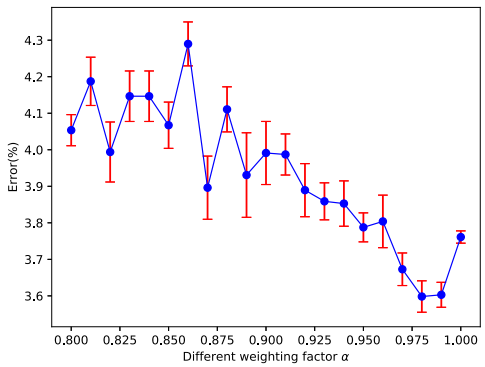


Fig. 2. The average error rate for the UCI HAR dataset on the last 50 epochs is plotted with an error bar of an standard deviation around the mean.

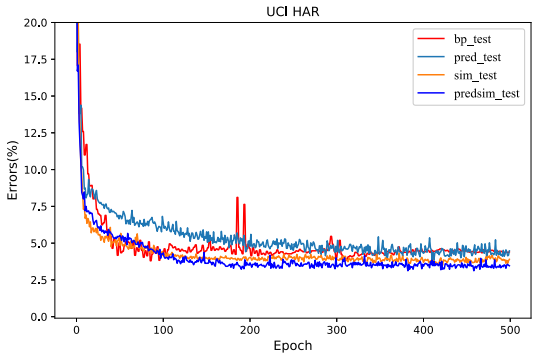


Fig. 3. Test errors on UCI HAR dataset with global loss and different local loss functions, on the tested CNN architecture.

improve performance. Batch normalization was applied before each activation function. The ReLU activation is used for networks trained with global loss or pred loss, and the Leakly-ReLU activation with a negative slope of 0.01 was used for networks trained with sim loss or predsim loss. Without loss of generality, the performance of the baseline CNN is compared when trained with global loss, the sim loss, pred loss and predsim loss. The max number of training epoch was set to 500 for all the datasets. The test error using different loss is reported, which is compared with other state-of-the-art methods on each HAR dataset.

The weighting factor α of predsim loss is determined by computing the performance of test errors on samples which were not chosen for training. The results on the UCI HAR dataset are shown in Fig.2. The average error rate on the last 50 epochs is plotted with an error bar of a standard deviation around the mean. On the whole, the average error rate displays a non-monotonic behavior on the weighting factor α, and attains a minimum. Without loss of generality, the weighting factor α for predsim loss is set to 0.99 for all the experiments.

For the HAR tasks, data segmentation is an important stage in the activity recognition process. The data stream must be segmented for preprocessing, and sliding windowing approaches are normally used for segmentation. The window size actually depends on the particular requirements for recognition system. To detect a specific activity, a particular window could be found to optimize the recognition quality. For applications which need to identify several activities, a sliding window that works well on average for the target activities is required. In intuition, small window size allows for a faster activity recognition. Large windows are normally used for the recognition of complex activities. Recently, for shallow learning, Banos *et al.* [28] investigated the effects of window for a wide range of window sizes and activities. However, no clear consensus exists on which window size should be preferably employed for deep learning. For comparison, we here use the same values used in previous cases of success.

Adam optimization method was used to train our model, and the initial learning rate was set according to different datasets. For the use of local loss, backward gradient flow is prevented by detaching the computation graph after each hidden layer. The final fully connected layer is trained normally via a cross-entropy loss function. The experiments were implemented. in Pytorch [Paszke *et al.* [29], 2017] deep learning framework on a machine with a NVIDIA GeForce RTX 2080 Ti GPU.

1) UCI HAR Dataset [7]: This dataset has been collected with a group of 30 volunteers within an age bracket of 19-48 years. Wearing a smartphone (Samsung Galaxy S II) on the waist, each person performed six activities including walking, walking-upstairs, walking-downstairs, laying, sitting and standing. The 3-axial linear acceleration and 3-axial angular velocity were captured at a constant rate of 50 Hz. The experiments have been video-recorded to label the data manually.

In the experiment, the whole dataset has been randomly partitioned into two sets, where 70% of the volunteers was selected for generating training data and 30% for testing data. The sensor signals including accelerometer and gyroscope were pre-processed by applying low pass filters and then sampled in fixed-width sliding windows of 2.56 sec and 50% overlap (window width is 128). For the UCI HAR dataset, the shorthand description of the baseline CNN is $C\,(64)\rightarrow C\,(128)\rightarrow C\,(256)\rightarrow$FC$\rightarrow$softmax, where C(Ls) denotes a convolutional layer s with Ls feature maps, FC depends on the dimension of the flatten feature vectors from stacked convolutional layers and softmax is a softmax classifier. The training time is 500 epochs, and batch size is 400. The learning rate was set to 0.004, 0.001, 0.0009, 0.0007 and 0.0005 for 10%, 14%, 16%, 30%, and 30% of the total training time.

In the experiment, our model was trained by using local loss, and the baseline was trained by using global loss. It can be seen from Fig.3 that pred loss when trained does not surpass the baseline. Sim loss can achieve test error close to that of the baseline. In particular, both of its combination, predsim loss, can consistently outperforms the baseline on the dataset. On the other hand, presim loss is able to make model converge faster at the same learning rate. Table.I demonstrates the performance of our model compared with state-of-the-arts in this experiment. Our method achieves 1.2% and 1.8% improvements over the results previously published by Jiang and Yin [10] and Ronao and Cho [30] using CNN. In particular, the model trained with predsim loss approaches the results obtained by Ignatov [31] using CNN with local features and statistical features, which to our knowledge is the best results on the dataset.Actually, the number of parameters does not decrease if one trains all the layers at the

TABLE I
ACCURACY AND F1 SCORE PERFORMANCE ON UCI DATASET

model	par	accuracy	F1
Baseline	1.29M	0.9620	0.9619
pred	0.35M	0.9542	0.9541
sim		0.9616	0.9615
predsim		0.9698	0.9697
Jiang et al.[10]	-	0.9518	-
Ignatovet al.[31]	-	**0.9763**	**0.9762**
Ronao et al.[30]	-	0.9575	-

same time. For the reuse of the memory, the local loss enables greedily training layers one-at-a-time. In this case, the number of parameters can be estimated according to the largest one of all training layers. Thus the number of parameters can be much smaller than that of the whole network. Compared with baseline, the number of parameters for our model decreases from 1.29M to 0.35M due to the use of layer-wise training.

2) OPPOTUNITY Dataset [17]: The OPPOTUNITY dataset contains naturalistic human activities collected in a very rich sensor environment where subjects performed daily morning activities. The data, sampled at a frequency of 30Hz, is composed of the recordings of 4 subjects including only on-body sensors consisting of 7 inertial measurement units (IMU) and 12 triaxial acceleration. Each subject was asked to perform 17 different ADLs with 20 repetitions. Data was manually labelled during the recording and later reviewed by at least two different persons based on the video recording.

In the experiment, the ADL1 ADL2 and ADL3 from Subject 1, 2 and 3 were used as training data and ADL4, ADL5 from Subject 4 and 5 were used as testing data. The sliding windows size is set to 64 and the sliding step is 8. For the dataset, the shorthand description of the baseline CNN is C (128)→C (256)→C (384)→FC→softmax. The training time is 300 epochs, and batch size is 200. The learning rate is set to 0.001 during total training time.

As there is a notable imbalance in the OPPOTUNITY dataset where the NULL class represents 72.28%, an evaluation metric mean F1 score independent of class repartition was used. In the case of the OPPORTUNITY dataset, we evaluatue performance including the Null class, which may lead to an overestimation of the performance given its large prevalence. Fig.4 demonstrates the performance of our model compared with other loss functions in this experiment. It can be seen that predsim loss, yielding an improvement of 4.1% on baseline, worked better than the other loss for the tested CNN architecture. Our method is also compared with other state-of-the-arts in Table.III. Even though our result is lower than the previously published result 82.3%(Zhang *et al.* [32], 2015) using deep belief networks, which needs larger computation, predsim loss is able to achieve the same 4.1% improvement on Zeng *et al's* method [25] using CNN alone. In particular, without extra cost, the number of parameters for our model decreased from 5.13M to 0.35M compared with the baseline.

3) UniMib-SHAR Dataset [18]: UniMiB SHAR, is a new dataset collected for the use of HAR and fall detection, which includes 11,771 samples of both human activities

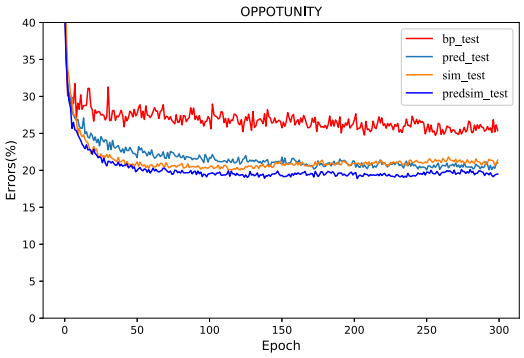


Fig. 4. Test errors on OPPOTUNITY dataset with global loss and different local loss functions, on the tested CNN architecture.

TABLE II
ACCURACY AND F1 SCORE PERFORMANCE ON OPPOTUNITY DATASET

model	par	accuracy	F1
Baseline	5.13M	0.7685	0.7620
pred	0.35M	0.7994	0.7932
sim		0.7997	7932
predsim		0.8100	**0.8058**
Zeng et al.[25]	-	0.7683	-
Zhang et al.[32]	-	**0.823**	-
Hammerla et al.[33]	-	-	0.7450
Ordóñez et al.[11]	-	0.7574	0.75

and falls performed by 30 subjects whose ages ranging from 18 to 60 years. The samples were recorded by an Samsung Nexus I9250 smartphone equipped with a Bosh BMA220 acceleration sensor. The data was resampled at a constant rate of 50 Hz, which is commonly used in the literature for HAR. The whole dataset consists of 17 fine grained classes, which were grouped into two coarse grained classes: one containing samples of 9 types of activities of ADLs and the other containing samples of 8 types of falls. In addition, the related criteria, such as activity type, age and gender, used to select samples, has been included in the dataset.

In the experiment, the dataset is randomly divided into two sets, where 70% was selected to generate training data and 30% testing data. The data is segmented with a sliding window of fixed length. The length of the sliding window is 151, with a step size of 3. For the UniMiB SHAR dataset, the shorthand description of the baseline CNN is C (128)→C (256)→C (384)→FC→softmax. The training time is 400 epochs, and batch size is 100. The learning rate was set to 0.003, 0.0015, 0.0009 for 7.5%, 5%, and 87.5% of total training time.

Fig.5 demonstrates the performance of our model compared with the baseline on the dataset. A surprising observation is that sim loss alone is still able to achieve significant improvement on the baseline. When combined, predsim loss achieve almost 3.7% improvement on the baseline. According to Table.III, to our knowledge, the best reported result on the dataset is 77.27% (Yang *et al.* [34], 2018) using dynamic fusion method. The second best result is 74.66% using CNN on raw signal data (Li *et al.* [35], 2018). Our result with

TABLE III
ACCURACY AND F1 SCORE PERFORMANCE ON UNIMIB-SHAR DATASET

model	par	accuracy	F1
Baseline	5.81M	0.7431	0.7396
pred	0.55M	0.7406	0.7347
sim		0.7706	0.7670
predsim		**0.7807**	**0.7782**
Yang et al.[34]	-	-	0.7727
Li et al.[35]	-	0.7466	0.7416

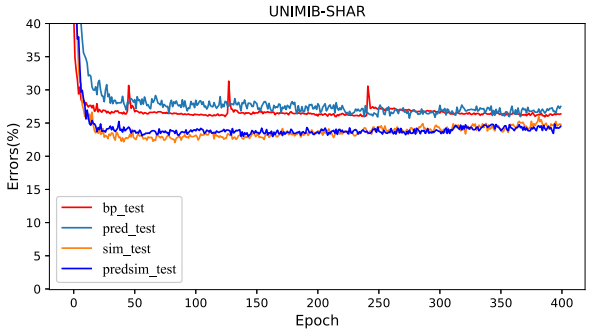


Fig. 5. Test errors on UNIMIB-SHAR dataset with global loss and different local loss functions, on the tested CNN architecture.

predsim loss surpasses Li *et al's* result by a large margin, and approach the result obtained by Yang *et al.* At no extra cost, the number of parameters for our method decreases from 5.81M to 0.55M by the use of local loss.

4) PAMAP2 Dataset [19]: The PAMAP2 Physical Activity Monitoring dataset is composed of 18 different physical activities (such as walking, cycling, playing soccer, etc.), which was collected on 9 subjects wearing 3 IMUs and a heart rate monitor. The 3 IMUs were placed over the wrist, chest and ankle on the dominant. Following a protocol, each of the subjects performed 12 different activities, and some of the subjects performed 6 optional activities. The dataset contains 54 columns: each line consists of a timestamp, an activity label (the ground truth) and 52 attributes of raw sensory data. For the PAMAP2 dataset, the IMU and heart rate were sampled at a frequency of 100Hz and 9HZ respectively. Khan *et al.* [36] shows that activity recognition with wearable sensing typically uses non-optimal sampling rates. A frequency of 100Hz is too high for HAR tasks, which leads to waste of resources. For comparisons, we subsampled the sensor signals in order to match the frequency used in other related literatures [33], [34], [36], [37] on the dataset, which is still sufficient for sensing body motion.

In the experiment, for the whole dataset we randomly split 80% for training and 20% for testing. Time series signals were down sampled to 30 Hz following suggestions in the related literature. The sensor signals were sampled in fixed-width sliding windows of 2.56 sec and 50% overlap (window width is 128). For the PAMAP2 dataset, the shorthand description of the baseline CNN is C (128)→C (256)→C (384)→FC→softmax. The training time is 500 epochs, and batch size is 200. The learning rate is set to 0.0005 during total training time.

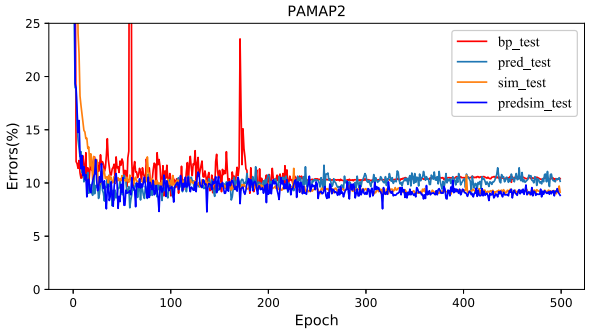


Fig. 6. Test errors on PAMAP2 dataset with global loss and different local loss functions, on the tested CNN architecture.

TABLE IV
ACCURACY AND F1 SCORE PERFORMANCE ON PAMAP2 DATASET

model	par	accuracy	F1
Baseline	8.62M	0.9116	0.91083
pred	2.60M	0.9224	0.9221
sim		0.9258	0.9244
predsim		**0.9297**	0.9303
CNN_Hammerla[33]	-	-	**0.937**
Yang et al.[34]	-	-	0.901
Zeng et al.[37]	-	-	0.8996
Khan et al.[36]	-	-	0.8600

The results of the local loss methods on the PAMAP2 dataset are shown in Fig.6. It can be seen that among several local loss functions, predsim loss systematically performs better than the baseline. Table.IV demonstrates the performance of our model compared with state-of-the-arts on the dataset. When compared to the best submissions of the PAMAP2 challenge, our model trained with predsim loss approaches the result reported using the CNN with 3 extra pooling layers and fully connceted layers (Hammerla *et al.* [33], 2016). For the PAMAP2 dataset, the number of parameters decreases from 8.62M to 2.60M compared with the baseline.

5) WISDM Dataset [20]: This experiment uses a standard HAR dataset which is publicly available from the WISDM group. A triaxial accelerometer sensor embedded on Android smartphone was used to generate data. In supervised condition, each subject wearing the smartphone in a front leg pocket performed five different activities including walking, jogging, walking upstairs, walking downstairs, sitting, and standing. While performing these activities, the data was recorded and the sampling rate was kept at 20Hz.

In the experiment, the dataset was randomly divided into two parts, where 70% was selected to generate training data and the rest testing data. The sliding windows size was set to 200 and the sliding step length was 20. For the WISDM dataset, the shorthand description of the baseline CNN is C (128)→C (256)→C (384)→FC→softmax. The training time is 500 epochs, and batch size is 200. The learning rate was set to 0.004,0.001,0.0009,0.0008,0.0007 and 0.0005 for 10%, 10%, 10%, 10%, 20% and 40% of total training time.

Results for local loss and global loss are reported in Fig.7. It can be seen that both predsim loss and sim loss consistently outperform other loss functions. In particular, sim loss almost

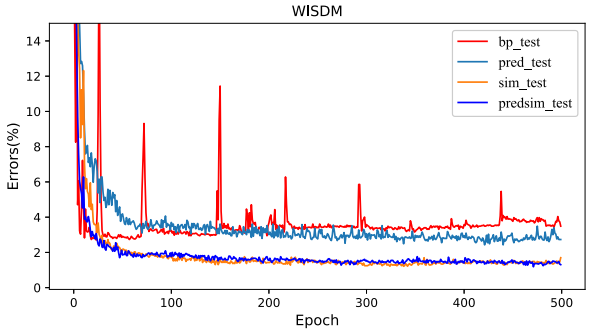


Fig. 7. Test errors on WISDM HAR dataset with global loss and different local loss functions, on the tested CNN architecture.

TABLE V
ACCURACY AND F1 SCORE PERFORMANCE ON WISDM DATASET

model	par	accuracy	F1
Baseline	8.62M	0.9724	0.9723
pred		0.9751	0.9750
sim	2.60M	0.9873	0.9872
predsim		**0.9882**	**0.9881**
Ravi et al.[20]	-	-	0.9820
Alsheikh et al.[38]	-	-	0.9823
Ignatov et al[31]	-	-	0.9332
Ravi et al.[20]	-	-	0.9860

approaches the test error of predsim loss. The predsim loss achieves 1.6% improvement on baseline. Table.V demonstrates the performance of our model compared with state-of-the-arts on the WISDM dataset. The best reported results for this task is to our knowledge 98.6% combining automatically learnt features and shallow features (Ravi *et al.* [20], 2018). The second best result is 98.2% using deep belief networks with greedy layer-wise training (Alsheikh *et al.* [38], 2016). Our result with predsim loss surpasses these results, even though the number of parameters were smaller.

V. DISCUSSION

The local loss model can alleviate memory requirements when training neural networks. For simplicity and without loss of generality, we compare the training errors between local loss and global loss on the UCI HAR dataset in Fig.8. Our results show that full backprop has a faster drop in training error, which is in agreement with previous investigation in several imagery datasets [16]. The result is not surprising considering that full backprop has direct access to the true gradient on each layer. Moreover, it can be seen that full backprop can achieve final lower training error, compared than local losses in the experiment. Actually, the local loss plays a regulating effect, and this is also why the local loss can efficiently improve test error results on several benchmark HAR datasets.

As previously indicated, for sensor based HAR tasks, other classification methods may heavily rely on heuristic hand-crafted feature extraction, which is usually limited by human domain knowledge. Many researchers have witnessed that CNN is competent to extract features automatically from signals and it has achieved promising results on HAR tasks [1]. For example, Zeng *et al.* showed that CNN-based deep learning approach outperforms these traditional classification

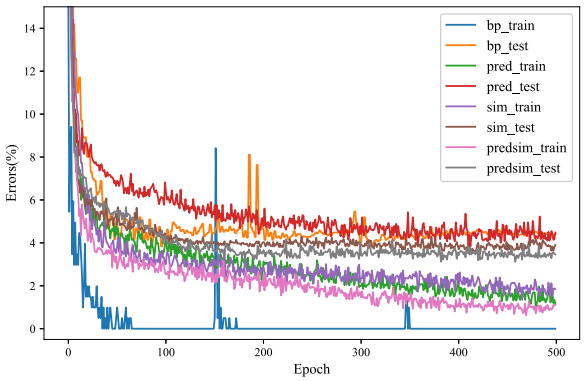
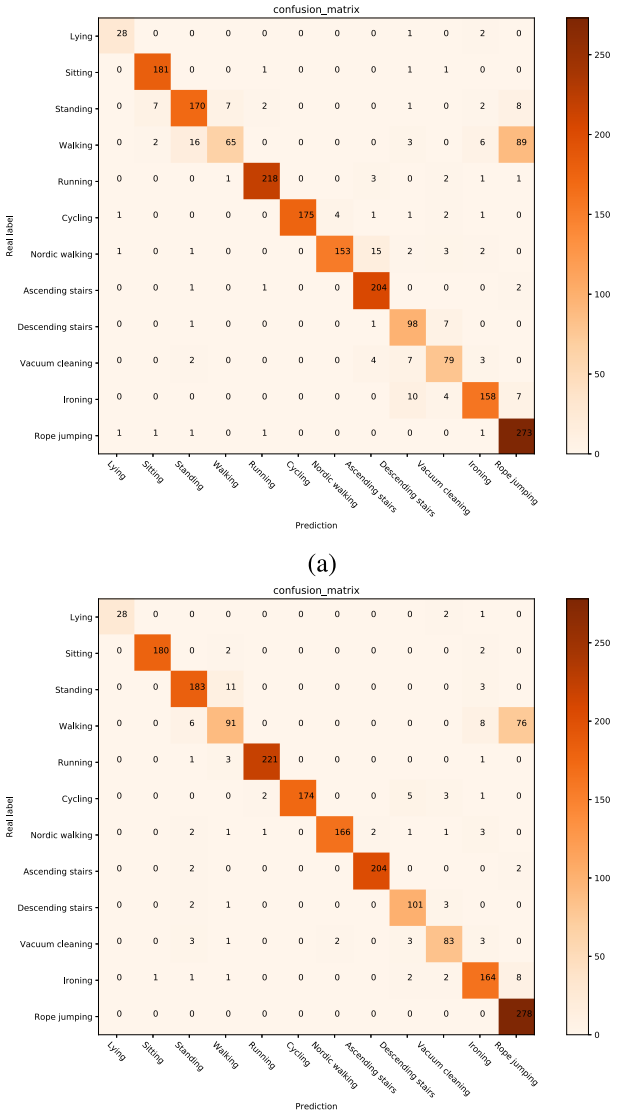


Fig. 8. Training and test errors on UCI HAR dataset with global loss and different local loss functions, on the tested CNN architecture.



(a)

(b)

Fig. 9. Confusion matrix for PAMAP2 data between the baseline and the predsim local error CNN. The Fig.9(a) denotes the confusion of baseline's CNN and the Fig.9(b) represents that of the predsim loss approaches.

methods [25]. Ronao *et al.* compared the performance of CNN and other state-of-the-art classification techniques, and witnessed that the CNN easily outperforms the latter methods,

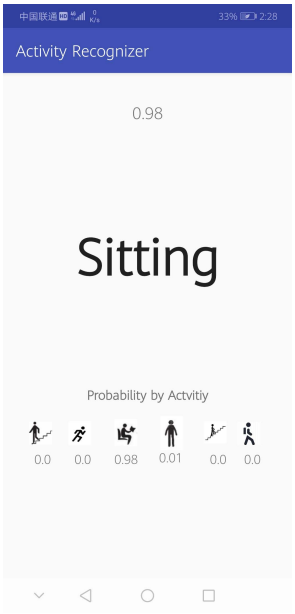


Fig. 10. Screenshot of the app's main window.

achieving higher accuracy [30]. Actually, the proposed model can be seen as a variant of CNN, which has better performance than CNN or other state-of-the-art classification methods according to our results. On the other hand, unlike image or video data, the discriminant features learned by CNN is hard to visualize for HAR. Referring to other HAR researches [11], [30], we use confusion matrix to associate explicit feature representation visually to demonstrate the superiority of classification of the CNN.

The confusion matrices on the PAMAP2 dataset for the HAR task are illustrated for the proposed model and the baseline CNN in Fig.9(a) and Fig.9(b). The confusion matrices contain visual information about actual and predicted activity classifications done by the system, to identify the nature of the classification errors, as well as their quantities. Each cell in the confusion matrix represents the number of times that the activity in the row is classified as the activity in the column. For two similar activities "walking" and "rope jumping", it can be seen that the local loss method has smaller number of misclassifications, compared than the baseline CNN.

During the learning the test errors do show high and small peaks. We suspect that such oscillation occurs due to label noise. Although sensor data is strictly labeled according to the kind of activity, unlike the clean imagery datasets such as CIFAR-10 or MINIST, the sensor data stream has to be segmented by sliding window for the use of the CNN. In particular, during transition between two activities of interest, some sliding windows sometimes inevitably consist of data from two activities. However, these windows have to be labeled as one of two activities according to their proportion, which lead to the occurrence of label noise. The HAR datasets usually are randomly partitioned into training and test set. During the training or test process, such oscillation inevitably occurs when one batch includes too many such polluted windows. For the HAR tasks, it can also be seen that the local loss is able to decrease the oscillations and make the training and test process more stable, compared with global loss.

On the other hand, referring to Ghio *et al's* works [39], for the transition-aware HAR mentioned above, we can consider two implementations of the CNN architecture, which differ in the way they deal with transitions that occur in between the activities of interest. In the first case, transitions are treated as unknown activities. Therefore, they are not learned by the machine learning algorithm. The known relationships between two activities of interest can be learned by recurrent networks without explicitly learning transitions. Instead, in the second case, transitions are learned by the algorithm as an extra class, which yet require some extra label work.

The proposed model can be applicable in real time HAR system. As previously stated, the proposed model can alleviate memory requirements during training process. At inference stage, the model behaves as a standard CNN with global back-propagation. A smartphone-based application for real time HAR is implemented to test the proposed layer-wise training CNN using the WISDM dataset [40]. The application is deployed on a Huawei Honor 10 smartphone with the android system 9.0. The app is mainly composed of two parallel thread functions: $ReceiveSensorSignals()$ communicates with motion sensors such as accelerometer, and periodically (20Hz) processes these received sensor signals according to the conditioning of the WISDM dataset. $OnlineInference()$ is triggered by scheduled interruptions every 1s corresponding to the duration of sliding window length. The proposed local loss model is trained using the WISDM dataset. We saved the computation graph with weights and created a checkpoint folder. The computation graph is frozen and saved as a single protobuf file, which is exported to the android application using android studio. The OnlineInference() can perform forward inference and do predictions on a smartphone using real time accelerometer sensor as inputs to the model. One carried the android phone in his front pants leg pocket and performed several activities such as walking, standing, ascending stairs, descending stairs, and jogging in a controlled environment. A screenshot of the app's main window is shown in Fig.10, which represents the current performed activity. The app interface also can use text-to-speech to output the recognition results.

VI. Conclusion

In the paper, we for the first time proposed a layer-wise training deep neural networks for the use of HAR tasks. The model can be trained with local loss combined by pred loss and sim loss. We performed experiments on five public benchmark HAR dataset including UCI HAR dataset, OPPOTUNITY dataset, UniMib-SHAR dataset, PAMAP dataset, and WISDM dataset. For each dataset, we built the baseline CNN which approach the results previously reported using CNN in the related literature. As previously stated, the local loss can play a regulating effect, which leads to final higher training errors. Thus, the proposed method may achieve stronger test error results, due to a better generalization ability. In particular, for the HAR tasks, the local loss is able to make the training process more stable, compared with global loss. According to experiment results, the predsim loss is able to consistently outperform the baseline on test error. When compared to the

existing approaches, the layer-wise model is able to achieve a better accuracy that match these state-of-the-art results, even though the number of parameters was smaller.

In summary, the layer-wise training model can improve further the performance across a variety of HAR tasks, without any extra cost such as convergence rate, accuracy and memory efficiency and so on. Due to the use of local loss, the backward locking problem can be resolved, and the activation parameters do not need to be always kept in memory. That is to say, memory can be released or reused, which is very useful for the HAR tasks on wearable sensors. On the other hand, local loss has a better biological plausibility than global loss. We believe that the new layer-wise training model with local loss could also achieve promising performance for other deep HAR tasks.

Here, we stress that the main purpose of this paper is to propose a layer-wise training CNN with local loss for the use of HAR. Our results also demonstrate its superiority on benchmark HAR datasets. In real time environment [39], transition-aware HAR with traditional shallow classification method is investigated, and the results show that the training of transition data can be used to improve further the performance of the system. CNN and its variant can be used as an automatic feature extractor to replace hand-crafted features and update the transition-aware HAR system. However, this is not our main research motivates in this paper. Actually, combining CNN and traditional shallow transition-aware HAR still requires much research work, which is another line of research. The new deep transition-aware HAR system need to be designed. In addition, training of transition data, with short duration time, also need a large amount of strictly labeling work. We would like to put the novel deep transition-aware HAR research as our further work.

REFERENCES

[1] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognit. Lett.*, vol. 119, pp. 3–11, Mar. 2019.

[2] P. Rashidi and D. J. Cook, "Keeping the resident in the loop: Adapting the smart home to the user," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 39, no. 5, pp. 949–959, Sep. 2009.

[3] Y.-J. Hong, I.-J. Kim, S. C. Ahn, and H.-G. Kim, "Mobile health monitoring system based on activity recognition using accelerometer," *Simul. Model. Pract. Theory*, vol. 18, no. 4, pp. 446–455, Apr. 2010.

[4] S.-R. Ke, H. Thuc, Y.-J. Lee, J.-N. Hwang, J.-H. Yoo, and K.-H. Choi, "A review on video-based human activity recognition," *Computers*, vol. 2, no. 2, pp. 88–131, 2013.

[5] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1192–1209, 3rd Quart., 2013.

[6] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, "Sensor-based activity recognition," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 6, pp. 790–808, Nov. 2012.

[7] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *Proc. Int. Workshop Ambient Assist. Living*. Cham, Switzerland: Springer, 2012, pp. 216–223.

[8] P. Casale, O. Pujol, and P. Radeva, "Human activity recognition from accelerometer data using a wearable device," in *Proc. Iberian Conf. Pattern Recognit. Image Anal.* Berlin, Germany: Springer, 2011, pp. 289–296.

[9] M. Berchtold, M. Budde, D. Gordon, H. R. Schmidtke, and M. Beigl, "ActiServ: Activity recognition service for mobile phones," in *Proc. Int. Symp. Wearable Comput. (ISWC)*, Oct. 2010, pp. 1–8.

[10] W. Jiang and Z. Yin, "Human activity recognition using wearable sensors by deep convolutional neural networks," in *Proc. 23rd ACM Int. Conf. Multimedia (MM)*, 2015, pp. 1307–1310.

[11] F. Ordóñez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[12] M. Jaderberg *et al.*, "Decoupled neural interfaces using synthetic gradients," in *Proc. 34th Int. Conf. Mach. Learn. (JMLR)*, vol. 17, 2017, pp. 1627–1635.

[13] A. N. Gomez, M. Ren, R. Urtasun, and R. B. Grosse, "The reversible residual network: Backpropagation without storing activations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 2214–2224.

[14] Y. Bengio, D.-H. Lee, J. Bornschein, T. Mesnard, and Z. Lin, "Towards biologically plausible deep learning," 2015, *arXiv:1502.04156*. [Online]. Available: http://arxiv.org/abs/1502.04156

[15] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 315–323.

[16] A. Nøkland and L. H. Eidnes, "Training neural networks with local error signals," 2019, *arXiv:1901.06656*. [Online]. Available: http://arxiv.org/abs/1901.06656

[17] D. Roggen *et al.*, "Collecting complex activity datasets in highly rich networked sensor environments," in *Proc. 7th Int. Conf. Netw. Sens. Syst. (INSS)*, Jun. 2010, pp. 233–240.

[18] D. Micucci, M. Mobilio, and P. Napoletano, "UniMiB SHAR: A dataset for human activity recognition using acceleration data from smartphones," *Appl. Sci.*, vol. 7, no. 10, p. 1101, 2017.

[19] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *Proc. 16th Int. Symp. Wearable Comput.*, Jun. 2012, pp. 108–109.

[20] D. Ravi, C. Wong, B. Lo, and G.-Z. Yang, "Deep learning for human activity recognition: A resource efficient implementation on low-power devices," in *Proc. IEEE 13th Int. Conf. Wearable Implant. Body Sensor Netw. (BSN)*, Jun. 2016, pp. 71–76.

[21] M. Janidarmian, A. R. Fekr, K. Radecka, and Z. Zilic, "A comprehensive analysis on wearable acceleration sensors in human activity recognition," *Sensors*, vol. 17, no. 3, p. 529, 2017.

[22] T. Plötz, N. Y. Hammerla, and P. L. Olivier, "Feature learning for activity recognition in ubiquitous computing," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, 2011, pp. 1–6.

[23] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *Proc. Int. Conf. Pervasive Comput.* Berlin, Germany: Springer, 2004, pp. 1–17.

[24] T. Huynh and B. Schiele, "Analyzing features for activity recognition," in *Proc. Joint Conf. Smart Objects Ambient Intell. Innov. Context-Aware Services, Usages Technol. (SOC-EUSAI)*, 2005, pp. 159–163.

[25] M. Zeng *et al.*, "Convolutional neural networks for human activity recognition using mobile sensors," in *Proc. 6th Int. Conf. Mobile Comput., Appl. Services*, 2014, pp. 197–205.

[26] K. Wang, J. He, and L. Zhang, "Attention-based convolutional neural network for weakly labeled human activities' recognition with wearable sensors," *IEEE Sensors J.*, vol. 19, no. 17, pp. 7598–7604, Sep. 2019.

[27] D. Kuang, C. Ding, and H. Park, "Symmetric nonnegative matrix factorization for graph clustering," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2012, pp. 106–117.

[28] O. Banos, J.-M. Galvez, M. Damas, H. Pomares, and I. Rojas, "Window size impact in human activity recognition," *Sensors*, vol. 14, no. 4, pp. 6474–6499, 2014.

[29] A. Paszke *et al.*, "Automatic differentiation in PyTorch," in *Proc. NIPS Workshop Autodiff*, 2017.

[30] C. A. Ronao and S.-B. Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," *Expert Syst. Appl.*, vol. 59, pp. 235–244, Oct. 2016.

[31] A. Ignatov, "Real-time human activity recognition from accelerometer data using convolutional neural networks," *Appl. Soft Comput.*, vol. 62, pp. 915–922, Jan. 2018.

[32] L. Zhang, X. Wu, and D. Luo, "Recognizing human activities from raw accelerometer data using deep neural networks," in *Proc. IEEE 14th Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2015, pp. 865–870.

[33] N. Y. Hammerla, S. Halloran, and T. Ploetz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," 2016, *arXiv:1604.08880*. [Online]. Available: http://arxiv.org/abs/1604.08880

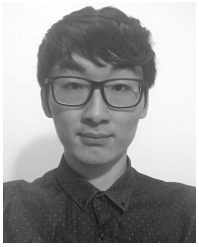
[34] Z. Yang, O. I. Raymond, C. Zhang, Y. Wan, and J. Long, "DFTerNet: Towards 2-bit dynamic fusion networks for accurate human activity recognition," *IEEE Access*, vol. 6, pp. 56750–56764, 2018.

[35] F. Li, K. Shirahama, M. Nisar, L. Köping, and M. Grzegorzek, "Comparison of feature learning methods for human activity recognition using wearable sensors," *Sensors*, vol. 18, no. 3, p. 679, 2018.

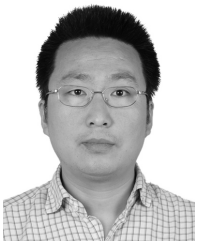
[36] A. Khan, N. Hammerla, S. Mellor, and T. Plötz, "Optimising sampling rates for accelerometer-based human activity recognition," *Pattern Recognit. Lett.*, vol. 73, pp. 33–40, Apr. 2016.

[37] M. Zeng *et al.*, "Understanding and improving recurrent networks for human activity recognition by continuous attention," in *Proc. ACM Int. Symp. Wearable Comput. (ISWC)*, 2018, pp. 56–63.
[38] M. A. Alsheikh, A. Selim, D. Niyato, L. Doyle, S. Lin, and H.-P. Tan, "Deep activity recognition models with triaxial accelerometers," in *Proc. 13th Workshops AAAI Conf. Artif. Intell.*, 2016, pp. 1–6.
[39] J.-L. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra, and D. Anguita, "Transition-aware human activity recognition using smartphones," *Neurocomputing*, vol. 171, pp. 754–767, Jan. 2016.
[40] Girish. *Human Activity Recognition Using Recurrent Neural Nets RNN LSTM and Tensorflow on Smartphones*. Accessed: Feb. 26, 2020. [Online]. Available: https://github.com/girishp92/Human-activity-recognition-using-Recurrent-Neural-Nets-RNN-LSTM-and-Tensorflow-on-Smartphones

Qi Teng received the B.S. degree from the Henan University of Engineering, Zhengzhou, China, in 2018. He is currently pursuing the M.S. degree with Nanjing Normal University. His research interests include activity recognition, computer vision, and machine learning.



Kun Wang received the B.S. degree from Southeast University, Nanjing, China, in 2017. He is currently pursuing the M.S. degree with Nanjing Normal University. His research interests include activity recognition, computer vision, and machine learning.



Lei Zhang received the B.Sc. degree in computer science from Zhengzhou University, China, the M.S. degree in pattern recognition and intelligent system from the Chinese Academy of Sciences, China, and the Ph.D. degree from Southeast University, China, in 2011. He was a Research Fellow with IPAM, UCLA, in 2008. He is currently an Associate Professor with the School of Electrical and Automation Engineering, Nanjing Normal University. His research interests include machine learning, human activity recognition, and computer vision.

Jun He (Member, IEEE) received the Ph.D. degree from Southeast University, Nanjing, China, in 2009. He was a Research Fellow with IPAM, UCLA, in 2008. From 2010 to 2011, he was a Postdoctoral Research Associate with The Chinese University of Hong Kong. He is currently an Associate Professor with the School of Electronic and Information Engineering, Nanjing University of Information Science and Technology. His main research are in the areas of machine learning, computer vision, and optimization methods. In particular, he is interested in the applications of weakly supervised learning via reinforcement learning methods.