Southeast-Asian J. of Sciences: Vol. 5, No. 2 (2017) pp.101-110

EXPERIMENTS ON DEEP LEARNING FOR WEARABLE ACTIVITY RECOGNITION

Nguyen Thi Thanh Thuy and **Nguyen Ngoc Diep**

Faculty of Information Technology
Posts and Telecommunications Institute of Technology (PTIT)
Hanoi, Vietnam
e-mail: diepnguyenngoc@ptit.edu.vn

Abstract

Current research is beginning to adopt deep neural network models on human activity recognition to extract features automatically from sensor data rather than relying on carefully designing suitable feature representation. However, there is just a few of custom deep architectures are explored. In this paper, we presents experiments on three deep leaning models for human activity recognition using wearable sensor. The effectiveness of the three deep neural networks is validated on accelerometer data from two public datasets. The results show that with enough sensor input data, highway convolutional networks provide higher accuracy than the other deep learning models.

1 Introduction

Activity recognition is playing an important role in supporting people's daily life with a wide range of applications such as situated services [16], energy expenditure estimation [19], etc. In these applications, wearable sensors are used to capture the movements or behaviours of users or objects. In addition, signal processing and machine learning techniques can be applied to automatically recognize what and when an activity is being performed by a user in real time.

As activity recognition is a time series problem, the main task for analysing sensor data stream are often to extract useful features in segmented data/frame

Key words: Activity recognition, Wearable sensor, Deep learning, Neural networks.

using sliding windows and to classify those portions of data that cover activities of interest with the trained activity models. In this task, one of the keys to successful activity model is the appropriate feature representations of the sensor data. The predominant approach to feature representation is feature engineering obtained by heuristic processes. These features are investigated carefully in [4] and most of them are statistical metrics calculated directly on the raw sensor data within a frame. However, to discover suitable feature representations we need application-specific expert knowledge. This task is extremely difficult because human activities are too complex and temporal dynamics.

A better approach is using multilevel features which have shown good recognition performance in recent activity recognition works [9, 10, 24]. The features are extracted from sequential data based on feature learning using bag of features, which can automatically discover meaningful representation of data to be analysed. However, to generate higher level feature automatically, we still have to utilize simple local features which are extracted from small segments of each activity frame.

Current researches adopt deep learning techniques to extract features automatically from raw sensor data and the results are very promising [1, 6, 14, 23]. One of the advantage of this approach is it can completely substitute for manually handcrafted feature extraction. Moreover, by using many layers of non-linear information processing for feature extraction and classification, deep learning techniques allows for in-depth analysis of the underlying data since the new representation implicitly highlights the most informative portions of the analysed data. Also, in computer vision, audio and text processing [12, 13], deep learning techniques have outperformed many conventional methods.

In this paper, we provide a systematic exploration of the performance of state-of-the-art deep learning approaches on two public wearable activity recognition datasets. These are three models: convolution neural network (CNN), highway convolutional neural network (Highway CNN) and residual neural network (ResNet). The suitability of each models and hyper-parameters are investigated. The results show that with enough sensor input data, the highway CNN provides better results than the other deep learning models.

2 Related Work

Recent researches on wearable activity recognition using deep learning techniques such as [6, 14, 15, 17, 23] have achieved good results. The author of [17] used Deep Belief Networks which are RBMs consisting of 4-layers with 1024 units in each hidden layer and 30 units in the top one. The proposed deep models achieved good result compared to other models using conventional classifiers. Zeng et al. in [23] proposed CNN with partial weight sharing to recognize activities using accelerometer data. Compared to RBM (a fully connected DNN model), their proposed CNN model achieved higher accuracy. Similarly, Pham et al. in [15] achieved a promising results with the CNN-based models for their smart shoes system. The work of [1] shown a better recognition improvement with their proposed DBNs which consists of one Gaussian-binary RBM and some binary-binary RBMs with the input as spectrogram of windowed excerpts from acceleration data stream. The works of [6, 14] utilise the long-short term memory (LSTM) architecture for the recurrent neural network for the activity recognition and achieved reasonable results.

In other research fields like speech and image processing, ResNet [7] and Highway Network [18] have shown substantial improvements in accuracy but never been used in wearable activity recognition. In this paper, we explore these deep models and compare their performance with CNN models.

3 Deep Learning Models for Activity Recognition

Because the complexity of human activity, feature extraction for activity recognition using wearable sensors is challenging [3]. There are high-level activities, each consists of several basic activities and these basic activities are closely correlated to each other. Moreover, even activity signals of the same activity performed by same individual may vary depending on many factors [3]. This kind of distortions and local dependencies in activity signal can be effectively captured by deep learning models.

Besides that, the recent trend in designing neural network is deeper, which are from ten layers deep to even hundreds of layers deep [7, 8, 18]. For many applications, especially in image recognition, it shown that with the proper training method, the deeper neural networks, the better the performance. In this paper, beside CNN-based model, we use ResNet and Highway Convolutional Neural Network (Highway CNN) [7, 18] to recognize activity using accelerometers.

3.1 Data processing

Accelerometer data from the sensor are often ambiguity and noisy. The noise of sensor can either be from outside factors causing some samples dropped or the sensors themselves generating noisy readings (e.g. too large or small values). In such cases, a filter is applied to remove noises and to fill out the lost samples. The data filter performs both a low-pass filtering for removing abnormally too low sample values, and a high-pass filtering for removing abnormally too high sample values. Next step, samples are grouped into sliding windows or frames. If a frame contains less than 75% of its full complement, it is discarded on the grounds as there is insufficient information to classify activities. Otherwise, it is smoothed using a cubic spline interpolation method [20]. This step is often called pre-processing.

The next step of data processing is segmentation. The input signal stream is segmented into frames or windows, which are then classified as belonging to different types of activities. In this study, we use the sliding window length of 64 data points and 50% overlap between two consecutive windows. This is similar to the setting of ECDF method used in [17] which we will use to compare the performance with our proposed methods. In the next section, we will describe our proposed deep learning models: CNN, RestNet and Highway CNN. All of our neural networks take frames of 3D acceleration time series as input data. This means each frame comprises of three channels (corresponding to X, Y, Z axes).

3.2 Convolutional neural network for activity recognition

Our CNN model is stack of convolutional, max-pooling, fully-connected, dropout and softmax layers. A convolutional layer consists of a set of independent and learnable filters, which can learn when it detects some specific type of features at some spatial position in the input. In this layer, spatial local dependencies are exploited by enforcing a local connectivity constraint between units and adjacent layers. Each unit is connected to only a small region of input frame. Rectified linear unit (ReLU) is used as activation function for the convolution layer because it is better in most situations compared to other activation function. To reduce the spatial size of the feature maps, we use pooling layer. It also help to retain the most important information and can reduce number of parameters and computations of the neural networks, hence to control overfitting. The final layers are two fully-connected layers and a softmax layer. A fully-connected layer perform high-level reasoning in the neural networks by taking the features extracted from convolution and pooling layers and learning non-linear combinations of the features. The last layer sofmax is used to predict a single class of various mutually exclusive classes based on training set. In this CNN model, we use two blocks of convolution and pooling layers. A dropout layer is used after the second fully-connected layer for regularization. It can help to decrease overfitting by avoiding training nodes on all training data, then lead the network to learn more robust features [21].

For convolutional layers, the higher layers often use broader filters to process more complex parts of the input. Therefore, we use 64 filters for convolutional layer 1 and 128 filters for convolutional layer 2. Both convolutional layers apply filters with same width of 5 and stride of 1. The width of 5 is also used for all max-pooling layers in the experiment. The dimension of two fully-connected layers is set to 500. And for the dropout layer, the probability of selecting units is set to 0.5.

3.3 Residual networks for activity recognition

If we just simply stack more typical layers or blocks such as convolutional or fully-connected layers together, performance of the deep networks will decrease. The problem is when using backpropagation for training traditional neural networks, the gradient becomes slightly diminished as it passes through each layer of the network and may disappear for the very deep network [5]. This problem can be solved using a deep residual learning framework or ResNet [7]. At each layer in ResNet, they use a shortcut connection to send the gradient signal backward smoothly. Formally, in ResNet, a residual layer is defined as: y = F(x) + x. In which x, y are the input and output vector of the layers considered. The function F(x) represents the residual mapping to be learned, which can be convolution, matrix multiplication, or batch normalization, etc. The "+x" at the end is the shortcut. Based on this, the gradient can pass backward directly. Moreover, by stacking these layers, gradient can pass through the network without being diminished.

In our proposed model for activity recognition, we use a convolutional layer, then 8 residual bottleneck blocks [7]. A bottleneck residual block makes residual networks more economical compared to the basic residual block. Similar to CNN model, we use ReLU activation for the convolution layer and residual layers. Next layer is pooling which can help to reduce the spatial size of the feature maps and the number of network parameters. Two fully-connected layers are used for high-level reasoning in the neural networks before the last layer softmax for classification. To control over-fitting, we also use two dropout layers, each on the outputs of the fully-connected layers.

In the convolution layer of the ResNet, we use 32 filters with size of 12 and stride of 1. Max pooling layer has width of 5 and the dimension of the first fully-connected layer is 1024 and the dimension of the second is 30. For the two dropout layers, the activation of randomly selected units during training to zero is set with probability of 0.8.

3.4 Highway networks for activity recognition

Highway network [18] is another architecture which can solve the problem of gradient vanishing because it uses the shortcuts as introduced in ResNet. The difference is the shortcuts are modified with a learnable parameter. This parameter can serve as gating unit which learn to regulate the flow of information through a network. Layers in Highway Network are defined as:

$$\mathrm{y} = \mathrm{H}(\mathrm{x}, \mathrm{W_H}) \cdot \mathrm{T}(\mathrm{x}, \mathrm{W_T}) + \mathrm{x} \cdot (1 - \mathrm{T}(\mathrm{x}, \mathrm{W_T})) \tag{1}$$

Here, T is called transform gate. Notice that, if $T(x, W\mathrm{T}) = 0$, then $y = x$ and if $T(x, W\mathrm{T}) = 1 then y = H(x, W\mathrm{H})$. Therefore, a highway layer can act as a plain layer or a layer which simply passes its input through depending on the output of transform gate.

Our Highway CNN model follows this architecture with stack of highway convolution layer, pooling layer, batch normalization layer, activation layer, fully-connected layers and softmax layer. Each block of highway convolution contains three highway convolutional layers, then a max-pooling layer and a batch normalization layer. Max-pooling layer is used to reduce number of parameters for the networks and the batch normalization can help to speed up the training process as well as to reduce the sensitivity to network initialization. In this model, we use five blocks of highway convolution with different filter sizes. All filters have same stride of 1 and width of 4, 8, 12, 8, 4 respectively. Similar to the two above models, we use ReLU as activation functions for highway convolution layers and fully-connected layers. The size of two fully-connected layers for high-level reasoning is 1024 and 256 respectively. For the last layer, the softmax layer is used for classifying different activities. Also for regularization, a dropout layer is used after each fully-connected layer. The probability value of 0.8 is used to control overfitting.

4 Experiment Evaluation

This section presents two datasets used for the three proposed deep learning models for wearable activity recognition. Furthermore, this part also shows the configuration parameters applied in the experiment of individual models and the evaluation metrics.

4.1 Dataset

The experiments are conducted on two public datasets widely used in activity recognition research. Both of them contain data streams from tri-axial accelerometers worn on subjects, which performed various activities in different contexts. A sliding window with a size of 64 sample points and 50% overlap is used to segment the data streams into frames.

The first dataset is Activity Prediction [11] which contains accelerometer data for six daily locomotor activities performed by 36 participants under laboratory settings. These are walking, jogging, ascending stairs, descending stairs, sitting, and standing, which are regularly performed by many people in their daily routines. Most of these activities involve repetitive motions and there are no background activities included. Each participant carry a cell phone in the pocket and the frequency of accelerometers in the phones is set at 20Hz. Frame size of 3.2 seconds is used. Totally, the dataset consists of 29,000 frames, which is the biggest dataset used in the experiment.

Second dataset is Skoda Mini Checkpoint [22]. The acceleration data was collected from assembly-line using multiple worn accelerometers in a car maintenance environment. We restrict our experiments to a single sensor worn on the right arm and create a subset from the original dataset. The new dataset Skoda contains 10 fine-grained activities plus unknown activities. With sampling rate of 48Hz, the Skoda dataset consists of 7,500 frames.

4.2 Experiment Settings

4.2.1 Evaluation metrics

In this paper, F-measure (F1 or harmonic mean) are used as measurement metric because it can measure the correct of classification of each class equally important. This metric also can be used to compare performance of different methods more easily. F1 score combines two measures which are precision and recall. The precision presents the rate of correct classification/prediction of a class, while the recall illustrates the rate of actual correct classification/prediction. F1 score presents the average of both precision and recall, therefore, it facilitates the performance comparison among classification models.

These measurements are defined as follows:

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

where TP (true positives): correct classifications of positive cases; FP (false positives): incorrect classifications of positive cases into negative class; FN (false negatives): incorrect classifications of negative cases into positive class.

In order to evaluate the performance of classification models, 10 folds cross-validation is used in the experiment. This validation technique divides the dataset into 10 parts such that 9 parts is employed for training and the last one is for testing. This technique is repeatedly applied until all of 10 parts are passed through the model. The measurements are computed by applying average functions.

4.2.2 Parameters of classification models

All of the proposed deep neural networks are built and trained by TFLearn library [2], a lightweight library featuring a higher-level API for TensorFlow to build and train neural networks. The model training and classification are run on a GPU with 1920 cores, 1506 MHz clock speed and 8 GB RAM.

Table 1: F1 score of the three proposed deep neural network models

Method	**Mean of F1 score (%)**	
	Activity Prediction	***Skoda***
CNN	97.68	87.09
ResNet	97.87	79.28
Highway CNN	**98.48**	85.56
PCA+ECDF	95.34	**88.25**

For training three proposed neural networks, we minimize the negative log likelihood using Adam optimizer, with learning rate of 0.001. All the networks are trained using mini-batches, where each mini-batch contains 64 frames and is stratified with respect to the class distribution in the training set and number of epochs used is set to 100.

4.3 Results

In the experiment, we evaluate the activity recognition result of four methods as shown in the first column in Table 1 using F1-score where this factor combines both precision and recall. These four methods including our three proposed deep neural network methods and a non-deep learning based state-of-the-art method called PCA+ECDF. This is a feature learning method proposed by Plotz et al. [17], which is based on PCA and ECDF combined with 1-NN classifier. We re-implemented the methods in Python and kept the best parameter values as reported in their paper [17].

As can be seen from the Table 1, all methods achieved high F1-scores on Activity Prediction dataset (more than 95%) while Skoda dataset proves to be more difficult in classification. The Skoda dataset consisting of 10 fined-grained activities with unknown activities are more challenging than Activity Prediction dataset containing only 6 simple daily activities without any background activities.

Precisely, on Activity Prediction dataset, the proposed Highway CNN model ranks the top by having F1-score at 98.48%; closely followed by the models of ResNet and CNN at 97.87% and 97.68% respectively. On this dataset, the improvements of the three deep learning models over PCA+ECDF is noticeable with almost 2.4% difference compared to the third best method ResNet. This result is not only showing that the three proposed deep learning models are more effective than PCA+ECDF in activity recognition but highway CNN with better way of using shortcuts is more effective than ResNet and CNN (without using any shortcut as ResNet or Highway CNN).

However, on Skoda dataset, the results are very different as PCA+ECDF ranks the top with F1-score at 88.25%. In addition, result of Highway CNN is even worse than CNN. This can be explained that the Skoda dataset has only 7,500 frames, which is not enough for training a good deep neural network model. Especially for models with big size like ResNet and Highway CNN, the amount of data required is much larger. Due to the lack of training data, these very deep learning neural networks could not be optimized and therefore result in degradation in recognition performance.

5 Conclusion

This paper conducts an investigation on three deep leaning models for human activity recognition using wearable sensor. Furthermore, the paper examines the performance of these deep neural networks against one state-of-the-art conventional classification model using PCA and ECDF [17]. The 10 times cross-validation method is applied in order to evaluate the raw performance of interested models on activity recognition using accelerometer data over two widely used datasets. The performance of all proposed deep neural networks is better than that of PCA+ECDF model. The experiments also show that, with enough training data, deeper neural network architectures like ResNet and Highway CNN are better than CNN, a normal deep architecture. Otherwise, the recognition performance will get degradation. In addition, with appropriate way of using shortcuts in network architecture, Highway CNN is more effective than ResNet.

References

[1] Mohammad Abu Alsheikh, Ahmed Selim, Dusit Niyato, Linda Doyle, Shaowei Lin, and Hwee-Pink Tan. Deep Activity Recognition Models with Triaxial Accelerometers. pages 8–13, 2015.

[2] Damien Aymeric. TFLearn: Deep learning library featuring a higher-level API for TensorFlow. *http://tflearn.org*, 2017.

[3] Andreas Bulling, Ulf Blanke, and Bernt Schiele. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys (CSUR)*, 46(3):33, 2014.

[4] Davide Figo, Pedro C. Diniz, Diogo R. Ferreira, and João M.P. Cardoso. Preprocessing techniques for context recognition from accelerometer data. *Personal and Ubiquitous Computing*, 14(7):645–662, 2010.

[5] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 9:249–256, 2010.

[6] Nils Y Hammerla, Shane Halloran, and Thomas Ploetz. Deep, Convolutional, and Recurrent Models for Human Activity Recognition using Wearables. *Ijcai*, pages 1533–1540, 2016.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. ResNet. *arXiv preprint arXiv:1512.03385v1*, 7(3):171–180, 2015.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[9] Tâm Hunh, Ulf Blanke, and Bernt Schiele. Scalable recognition of daily activities with wearable sensors. In *Location-and context-awareness*, pages 50–67. Springer, 2007.

[10] Tâm Huynh, Mario Fritz, and Bernt Schiele. Discovery of activity patterns using topic models. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 10–19. ACM, 2008.

[11] Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011.

[12] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*, pages 1–8, 2009.

[13] Honglak Lee, Y. Largman, Peter Pham, and a. Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. *Advances in Neural Information Processing Systems*, 22:1096–1104, 2009.

[14] Francisco Javier Ordóñez and Daniel Roggen. Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. *Sensors (Switzerland)*, 16(1), 2016.

[15] Cuong Pham, Nguyen Ngoc Diep, and Tu Minh Phuong. e-Shoes: Smart Shoes for Unobtrusive Human Activity. In *Knowledge and Systems Engineering (KSE), 2017 Ninth International Conference on*, 2017.

[16] Cuong Pham, Clare Hooper, Stephen Lindsay, Dan Jackson, John Shearer, Jurgen Wagner, Cassim Ladha, Karim Ladha, Thomas Ploetz, and Patrick Olivier. The ambient kitchen: A pervasive sensing environment for situated services. In *Paper presented at the ACM Conference on Designing Interactive Systems*, pages 2–3, 2012.

[17] Thomas Plötz, Nils Y Hammerla, and Patrick Olivier. Feature learning for activity recognition in ubiquitous computing. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 1729, 2011.

[18] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

[19] Emmanuel Munguia Tapia, Stephen S Intille, and Kent Larson. *Activity recognition in the home using simple and ubiquitous sensors*. Springer, 2004.

[20] Grace Wahba. Spline interpolation and smoothing on the sphere. *SIAM Journal on Scientific and Statistical Computing*, 2(1):5–16, 1981.

[21] Sida I Wang and Christopher D Manning. Fast dropout training. *Proceedings of the 30th International Conference on Machine Learning*, 28:118–126, 2013.

[22] Piero Zappi, Clemens Lombriser, Thomas Stiefmeier, Elisabetta Farella, Daniel Roggen, Luca Benini, and Gerhard Tröster. Activity recognition from on-body sensors: accuracy-power trade-off by dynamic sensor selection. In *Wireless sensor networks*, pages 17–33. Springer, 2008.

[23] Ming Zeng, Le T. Nguyen, Bo Yu, Ole J. Mengshoel, Jiang Zhu, Pang Wu, and Joy Zhang. Convolutional Neural Networks for Human Activity Recognition using Mobile Sensors. In *Proceedings of the 6th International Conference on Mobile Computing, Applications and Services*, 2014.

[24] Mi Zhang and Alexander A Sawchuk. Motion primitive-based human activity recognition using a bag-of-features approach. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, pages 631–640. ACM, 2012.