**Paper Review: “Deep Recurrent Neural Network for Mobile Human Activity Recognition with High Throughput”**

* Paper proposes method of HAR with high throughput from raw accelerometer data by applying a deep RNN
  + Investigates various architectures and combination to find the best param values
* “High throughput” refers to short time at a time of recognition
* Uses dataset 432 trials with 6 activity classes from 7 people
  + Max recognition rate was 95% and 83% in test data of 108 segmented trials, each having a single activity class and 18 multiple sequential trials, respectively
  + Max recog rate was 72% and 55% by traditional methods
* Efficiency of found params also evaluated with additional dataset
* DRNN requiring 1.35ms throughput (recognition per unit time)
  + Previously been 11.031 by traditional method (11.027 of which for feature calculation)
* These advantages are caused by compact and small architecture of constructed real-time oriented DRNN
* HAR applicable in many domains
  + Healthcare
  + Preventative medicine
  + Elderly care
* Cost of sensing devices fallen significantly w/ rapid spread of devices w/ built-in sensors (e.g. smartphones)
  + Hence, researches on mobile activity have been actively conducted
* Traditional HAR schemes have used ML method (e.g. decision tree, KNN, SVM, RF, Naïve Bayes) to recognize activities from a feature vector extracted from signals in a time window by using statistical values of Fourier transform
* RNN suitable for time-series data (audio, video, and ML)
* In DL, original data can be directly inputted as opposed to traditional HAR methods which are input feature vectors
* Speed-up expected as calculation of feature vectors skipped at training and testing
* Paper proposes method of HAR from raw accelerometer data applying a RNN and investigates various architectures and its combination to find best param values
* Recog ability of RNN evaluated using human activity sensing consortium (HASC) open dataset
* Note that ‘trial’ with regards to ‘dataset of 432 trials’ means one sequential data sample such as segmented or sequence data
* Fast response advantage of RNN over previous best is caused by the number of weights less than 10% of traditional method
* Paper makes 3 overall points
  + RNN architecture used in order to construct fast response classifier
  + Various params were explored to improve accuracy to investigate the factors that affect accuracy (used 2 types of dataset)
  + Throughput for RNN shown to be faster than existing method
* In many recent mobile HAR approaches, ML model used to recognize the activity as a basic technique after feature vector has been extracted by stats or Fourier transform by taking time window
* Since HAR handles sequential data, techniques for sequential data like HMMs and CRF (used in speech recog and NLP) were proposed
  + However, previous studies using these focused mainly on how to reduce feature calculation by shifting feature vectors, as opposed to completing the required HAR in real time
* CNN used recently to achieve high accuracy recognition
  + However, requires time window to generate certain length segmentation of time series signal
  + Moreover, huge number of connections between layers
  + Thus, not suitable for RT execution of mobile devices
* Recently many methods using CNNs and RNNs aim to recognize by using the raw signal directly without extraction of feature vectors
* RNN = high throughput network architecture that can deal with raw sensor without feature extraction and can recognize by thorough fast sequential processing
* Note: a pair of elements of the input and output vectors is called a ‘unit’ (i.e. unit is one neuron at a layer with two parts, its input summation ‘z’ and output after activation ‘a’)
* Truncated BPTT, which sets the time to date back to an appropriate constant to perform BPTT, is used
  + This avoids enormous calculation of full BPTT when time is long
* LSTM utilized to replace some units of the RNN to solve problems of input-output weight conflict
  + Conflicts between input from previous layer and recurrent value and vanish/explode grad problem
* Memory cell is used to store the internal state that allows it to perform controls
  + E.g. whether to write info to cell, read info from cell, or delete info of cell
* Input and output gate are for eliminating input-output weight conflicts
* Input gate used to control how much of the input is able to pass
* With output gate, by providing a gate that performs such an operation, it is possible to determine whether to memorize a state or to read the memorized state in accordance with input value or output value to internal layer unit
* Forget gate determines whether to forget the memorized state
  + Operates to perform efficient learning even in cases such as that where pattern of time-series data is changed suddenly to a pattern having no correlation w/ the previous context
* With using LSTMs, weights to be updated will increase
* Exploding gradient probability also solved via gradient chopping
  + Method of correctly L2 norm of gradient so that it doesn’t exceed a threshold value
  + Also serves as a regularization technique
* HASC corpus is a dataset for ML gathered and distributed by HASC
  + Distributed at a state with detailed label attached to the data measured by sensors mounted on a mobile device
* Paper uses part of acceleration signals of the HASC corpus as a dataset
  + Divided into segmented data and sequence data
  + Segmented data includes single activity in one trial
  + Sequence includes multiple consecutive activities
* Segmented data suitable for us as train data as able to label easily
* On the other hand, since sequence data are constructed by seamless measurement of human activities, these resemble actual human activities
* In evaluation, divided segment data into training data of 432 trials and test data of 108 trials
  + Number of samples in each activity class balances each other
* Test accuracy recognizes on segmented test data and sequence accuracy recognizes on sequence data
* “HAR using smartphones” dataset used to examine the generality of the method and applied with the best params found in the HASC dataset
  + Sensor data collected using smartphones equipped with 3-axis accelerometer and gyroscope
  + 6 types of activity class: standing, sitting, laying, walking, walking downstairs, and walking upstairs
  + Completed as sequential data
* To perform high throughput HAR for each time by using 3-axis accelerometer of smartphone as direct input, constructed DRNN such that the 3-axis accelerometer data of each time corresponded to the 3D input layer and 6 act classes to the 6D output layer
* Each internal layer is an LSTM unit
* Output activation function and error function defined by softmax and cross-entropy function, respectively
* Truncated BPTT under mini-batch SGD used to update weights at time of training
* Network outputs an activity class (the element having the largest value among elements of output vector obtained when an input vector is inputted)
* For each mini-batch and for each truncated time-range T, obtain error function from input and output to update weights via back-prop
* Chainer used to implement the DRNN as a framework for NN, where various NN models can be flexibly written in Python
* GPU used for training and CPU used for evaluating throughput of constructed DRNN
  + Strategy based on policy of RNN trained on large-scale computational architecture and execution done by standard-type mobile terminal
* Comparative methods include decision tree, SVM, and RF
  + Require time windows to calculate feature vectors (rather than raw sensor data)
  + Time windows of 5s extracted, shifting every 2.5s
  + Features include means of each axis, variance of each axis, mean sum of abs values of each axis, etc.
  + Reduced from 27 feature variables to 13 by applying stepwise feature selection using log recog
  + Over each ML model, grid search conducted over train data to choose the best model
* For evaluating throughput of recognition, the time required for the recognition of the entire sequence data was divided by the number of samples of the sequence data to derive the mean value of the recognition throughput per time unit
* For comparative method, calculated the compute time of feature vector in one time window, time taken to recognize the activity from the feature vector in one time window, and sum of these values
* Best model = model that showed the best recognition result for the sequence data
* Recog rate derived by ratio of the correct recog time against the total time for each trial (i.e. accuracy)
  + In best model, the test recognition rate = 95% and 83% for sequence data
* Needed ~80 epochs to obtain the best results
* DRNN shows between 22-35% better than DT/SVM/RF for segmented test data and 26-28% better on sequence data
* Recog rate highest in the case of 3 internal (hidden layers) for training, testing and sequence data
* Recog rate highest in case of 60 units in test + sequence data
* For truncated time, relatively good at T=30 or T=70, but noticeably worse (10.7%) between 70 and 10 for the sequence data
* No significant recognition rate difference due to variation between 3 and 9 in gradient clipping params
* P=0.3 for dropout rate showed highest recog rate for the test and sequence data
* Hence, a large difference appeared in recog rate for 5 params
* Regarding throughput time, when compared according to only recognition time calculation, existing method is faster (i.e. decision tree), but if compared according to the substantial time (including feature extraction), proposed method is 9.67ms faster
* Recog rate when carried over w/ methods and params to HAR dataset is at 95% recog rate at 45th epoch
* When tested with sequence data, some similar activities are erroneously recognized, thus impacting the recognition performance
  + Also noted sometimes a delay in recognition; considered to have been caused by the fact that unlearned signals that cannot be classified into any activ during the transition of activity were input
* Reduction in accuracy from 3 to 4 hidden layers can be interpreted to have occurred as a result of increasing in the learning difficult by excessive # of layers (also due to overfitting when freedom of model becomes too high)
* Also, more layers = more computed time and memory usage
  + Would add 0.35ms recog time to go from 3 to 4
  + Could reduce to 1 hidden layer to improve throughput at the expense of accuracy
  + Similar considering w.r.t. # of units per layer
* No performance difference generated by grad clip params
  + Implies that gradient explosion did not occur to a great extent
* With a throughput of 1.347ms for DRNN, considering that the data is currently acquired at 100Hz, it is sufficient to allow real-time processing
* Considered that, as RNN only performs product-sum operation with ‘Wxh’ and ‘Whh’ at time of recognition, considered that implementation in low-power devices (e.g. smartphones) may be possible
  + However, training is 116.39s per epoch, requiring a high-speed computer in advance
* DRNN can achieve high speed w/ high recog rate using CPU due to compact size of the DRNN, total size of inner variables/architecture of layers is 74k pieces (<10% size of conventional CNN + LSTM model), and is shown to have a clear advantage for miniaturization of the devices of recognition
* Possible to apply HMM as a post-processing technique on sequence data (which is noticeably worse than test segmented data accuracy)
  + Alternatively apply method that takes context of label into account in RNN called ‘connectionist temporal classification’
* Could further develop a compact DRNN circuit to equip it into a mobile device

**Significant Points and Takeaways from Paper**

* Paper proposes method of HAR from raw accelerometer data by applying an RNN and investigates various architectures and its combination to find best param values
* Since HAR handles sequential data, techniques for sequential data like HMMs and CRF were proposed
  + However, previous studies using these focused mainly on how to reduce feature calculation by shifting feature vectors, as opposed to completing the required HAR in real time
* LSTM utilized to replace some units of the RNN to solve problems of input-output weight conflict
  + Conflicts between input from previous layer and recurrent value and vanish/explode grad problem
  + Input and output gate are for eliminating input-output weight conflicts
* Paper uses part of acceleration signals of the HASC corpus as a dataset, which is divided into segmented data and sequence data, w/ segment including single activity in 1 trial and sequen including multiple consecutive)
* Segmented data suitable for us as train data as able to label easily
  + On the other hand, since sequence data are constructed by seamless measurement of human activities, these resemble actual human activities
* ‘Chainer’ used to implement the DRNN as a framework for NN, where NN models can be written in Python
* GPU used for training and CPU used for evaluating throughput of constructed DRNN; strategy based on policy of RNN trained on large-scale compute archit and execution done by standard-type mobile terminal
* In best model (highest recog rate for sequence data), the test recog rate = 95% and 83% for sequence data
* DRNN shows between 22-35% better than DT/SVM/RF (require time windows to calc feature vectors) for segmen test data and 26-28% better on sequen data; recog rate highest for test/sequence data w/ params
  + 3 internal (hidden) layers, 60 units, truncated time between T=30 and T=70, and drop rate of P=0.3
* When compared according to only recognition time calculation, existing method is faster (i.e. decision tree), but if compared according to the substantial time (inc feature extraction), proposed method is 9.67ms faster
* Recog rate when carried over w/ methods and params to HAR dataset (separate set from HASC dataset to test generality) is at 95% recog rate at 45th epoch
* When tested with sequence data, some similar activities are erroneously recognized, thus impacting the recognition performance
* With a throughput of 1.347ms for DRNN, considering that the data is currently acquired at 100Hz, it is sufficient to allow real-time processing
* Considered that, as RNN only performs product-sum operation with ‘Wxh’ and ‘Whh’ at time of recognition, considered that implementation in low-power devices (e.g. smartphones) may be possible
  + However, training is 116.39s per epoch, requiring a high-speed computer in advance
* DRNN can achieve high speed w/ high recog rate using CPU due to compact size of the DRNN, total size of inner variables/architecture of layers is 74k pieces (<10% size of conventional CNN + LSTM model), and is shown to have a clear advantage for miniaturization of the devices of recognition