ATLANTIS PRESS

Advances in Intelligent Systems Research, volume 133

2nd International Conference on Artificial Intelligence and Industrial Engineering (AIIE2016)

An Empirical Study on Energy Disaggregation via Deep Learning

Wan He* and Ying Chai

State Grid Energy Research Institute, State Grid Corporation of China, Beijing, China

*Corresponding author

***Abstract*—Energy disaggregation is the task of estimating power consumption of each individual appliance from the whole-house electric signals. In this paper, we study this task based on deep learning methods which have achieved a lot of success in various domains recently. We introduce the feature extraction method that uses multiple parallel convolutional layers of varying filter sizes and present an LSTM (Long Short Term Memory) based recurrent network model as well as an auto-encoder model for energy disaggregation. Then we evaluate the proposed methods using the largest dataset available. And experimental results show the superiority of our feature extraction method and the LSTM based model.**

Keywords-energy disaggregation; neural networks; deep learning;NILM

I. INTRODUCTION

With the development of economy, every day we consume more and more electricity, most of which comes from fossil fuels. Energy saving and emission reduction are important topics in all countries, not only because fossil fuel reserves are finite, the consumption of them also causes environmental pollution. Energy disaggregation (also known as non-intrusive load monitoring, or NILM for short) is the task of estimating the power demand of each individual appliance given aggregate power demand signal recorded by a single electric meter which monitors multiple appliances. Disaggregated electricity consumptions can be used to produce itemized electricity bills to help customers to identify and improve their consumption behavior. Moreover, energy disaggregation can help operators to better manage power grid, and detect faulty or improperly used devices [1].

The research on energy disaggregation dates back to 1980s. The seminal work of Hart [2-3] considers each appliance as a finite state machine and extracts transients between steady states from real and reactive power signals. Note that Hart's method focuses on detecting electrical events rather than separating power signal of individual appliance. Subsequent approaches focusing on event detection usually incorporate more features from harmonics and sometimes use information of very high frequency harmonics. Recently, 160 teams joined in the Belkin Energy Disaggregation Competition [4] held on the Kaggle platform which requires participants to predict the status of each appliance at each time point.

As for methods that directly estimate power demand of each device from aggregate power signals, a series of Factorial Hidden Markov Model (FHMM) [5] based approaches have been studied. Kim et al. [6] compared the effectiveness of several unsupervised disaggregation methods on low frequency power measurements and proposed a model based on FHMM variant which can integrate additional features related to when and how appliances are used in the house. Kolter and Jaakkola [7] developed a convex programming based approximate algorithm to additive FHMM and achieved state-of-the-art performance. Supervised machine learning methods have also been introduced into the energy disaggregation domain. Kolter et al. [8] formulated the objective of maximizing disaggregation performance as a structured prediction problem and developed an effective algorithm to learn sparse representations of electrical signals discriminatively. Elhamifar and Sastry [9] defined dissimilarities between energy snippets of each device and used them in a subset selection scheme to find powerlets (representative power snippets), and then formulated the disaggregation problem as an optimization over the learned powerlet dictionary under various constraints of device usage patterns. In addition, from a broader perspective, energy disaggregation can be regarded as a single-channel source separation problem [10].

Currently, deep learning [11] receives more and more attention and has made significant improvements in a lot of fields such as computer vision, speech recognition and natural language processing. So researchers now start to adapt deep neural networks to energy disaggregation task. Kelly and Knottenbelt [1] applied three types of deep neural network structures, a recurrent neural network using Long Short Term Memory units (LSTM) [12], a denoising auto-encoder, and a regression model that predicts the start time, end time and average power demand of each appliance. However, they did not achieve state-of-the-art disaggregation performance with deep learning approaches. Nascimento [13] also experimented and analyzed various deep learning methods to improve the performance of NILM.

In this paper, we conduct empirical investigation of deep learning methods in energy disaggregation (NILM). We adapted two types of neural network architectures to NILM. The first can be viewed as regression problem which estimates the transient power demand of a single appliance given the whole series of the aggregate power. It can also be considered as a non-symmetric auto-encoder. The second type of network is a multi-layer RNN (recurrent neural network) using LSTM units, which is similar to the structure used in [1]. For both structures, we use multiple parallel convolutions with different filter sizes to transform the raw power signals. We borrow this

idea from GoogleLeNet in image processing [14] and research papers in natural language processing [15].

The rest of this paper is organized as follows. In Section 2, we introduce the formulation of NILM. After that, we detail our methods in Section 3. Then we provide experiments and evaluation results in Section 4. Finally, we conclude our work in Section 5.

II. ENERGY DISAGGREGATION

In this section we describe a simple formulation of energy disaggregation. The purpose of energy disaggregation is to separate the power demand of each individual appliance from the whole electric consumption signal. Assuming there are N different appliance in the building, let $y_t^{(i)}$ denote the power signal of appliance i at time t, where $t \in \{1,2,...,T\}$ and T is the length of the power series. So the aggregate signal recorded by a single smart meter can be represented as the mixture of signals of each device:

$$x_t = \sum_{i=1}^{N} y_t^{(i)} \quad . \tag{1}$$

So given the aggregate power consumption $\{x_t\}_{t=1}^{T}$, we need to estimate the power consumption $\{y_t^{(i)}\}_{t=1}^{T}$ of each appliance i, where $i = 1,2,...,N$.

In practice, the value of T varies for different type of devices, since each type of device has its own usage patterns. Similar to [1], we refer to the power over a complete cycle of an appliance as an appliance activation. For a short-duration appliance, for example a kettle, an activation usually lasts for several minutes, while for long-duration device such as washing machine, an activation may be as long as several hours. And in this paper, for every device we set T to be large enough to capture the majority of activations of that device. Figure I presents example activations of a kettle and a washing machine.

III. DEEP LEARNING BASED ENERGY DISAGGREGATION

In this section we first introduce deep learning and discuss the feature transform method which uses multiple parallel convolutional operations with different filter sizes. And then we present two different deep neural network architectures for energy disaggregation.

Deep learning and deep neural networks (DNN) have achieved a series of success in a number of domains such as computer vision, speech recognition and machine translation. In theory, deep neural networks have the ability to learn complex nonlinear relationship between input patterns and the target to predict. While in practice, they are quite flexible and enable building end-to-end solutions in a lot of tasks. So we attempt to apply deep neural networks in NILM to estimate the per device signals from the overall power signal.

Usually, a deep neural network can be considered as a directed acyclic graph (DAG), where each node represent a type of computation or transform, and edges correspond to data

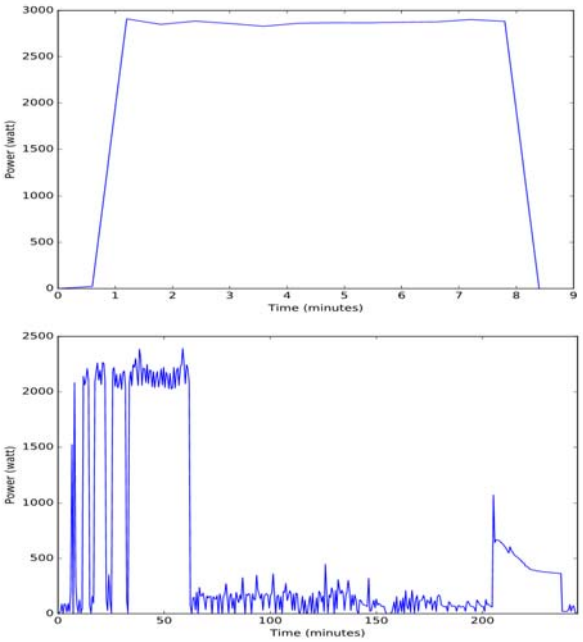

FIGURE I. ACTIVATIONS OF TWO DIFFERENT APPLIANCES. TOP: KETTLE, BOTTOM: WASHING MACHINE

flow between nodes. In general, a node in the DAG is modeled using a layer. So far, a variety of layers have been proposed, for example dense layer, convolutional layer, dropout layer, batch normalization layer and all kinds of activation layers, etc. So by combining these layers we have the flexibility to construct a wide variety of networks for different tasks. And in this paper, we present two types of network for energy disaggregation - a multi-layer feed forward network with convolutional layers and an LSTM based recurrent network. Generally DNNs are optimized using the back propagation algorithm, which originates from the chain rule of composite function derivation. And in the backpropagation approach, training data flow from input layer(s) to output layer(s) in the forward pass while error information flows in the opposite direction in the backward pass.

The CNN network used in this paper consists of multiple convolutional layers and dense layers. Formally, the CNN network can be viewed as a mapping from the input aggregate power signal series $\{x_t\}_{t=1}^{T}$ to the power consumption $\{y_t^{(i)}\}_{t=1}^{T}$ of a specific appliance. An autoencoder (AE) is a network which tries to reconstruct the input. It first encodes its input into a compact representation, and then decodes to restore the input. Since energy disaggregation can be viewed as the process to reconstruct the clean power signal of each specific device from the corrupted power signal with noise from other devices, the CNN network can also be considered as a denoising autoencoder.

The detailed architecture of the CNN network is as follows:

1. Input (length T is appliance specific window size)
2. Parallel 1D convolution with filter size 3, 5, and 7 respectively, stride=1, number of filters=32, activation type=linear, border mode=same

3. Merge layer which concatenates the output of parallel 1D convolutions
4. Dense layer, output_dim=128, activation type=ReLU
5. Dense layer, output_dim=128, activation type=ReLU
6. Dense layer, output_dim= T , activation type=linear

Convolutional layers are typically used in the first layers of a deep neural network structure. And using a small number of filters with local receptive field, they have the ability to detect various features from the input data. Taking CNN for image recognition as an example, the lower layer convolutional filters identify features such as edges, corners and patches while higher convolutional layers detect even more abstract features like trees in a scene, nose in a human face, etc. CNN are also frequently used in 1 dimensional signal processing such as audio and Electroencephalography signals [11]. So in this paper we use CNN layers to extract representative features from power series.

In the above network architecture we employ multiple parallel convolutional layers with varying filter size to detect features from raw power signal. We borrow this idea from the GoogleLeNet [14] model for image recognition, which concatenates features extracted by 1x1, 3x3, 5x5 2D convolutions followed by different types of pooling. In contrast to the normal approaches which connect layers in a cascade manner, this introduces parallel structures into the network. And in this way we can learn richer features from the raw power in our task. This approach is also widely used in natural language processing where 1D convolution with different filter sizes are applied to sentences [14].

Compared to the feedforward network, one advantage of recurrent neural network (RNN) is the ability to remember the dynamics of previous inputs in its memory. And this makes them especially suitable for processing time series data like the power consumption data in our task. However, simple RNN suffers from the vanishing and exploding gradient problem in its training using back propagation through time [11], and this limits its ability in processing long term dependencies. In order to solve this problem, the gated recurrent unit called LSTM (Long Short Term Memory) is exploited in most practical applications. LSTM has a lot of advantages over the simple RNN model in addition to alleviating the vanishing gradient problem.

Formally, given the aggregate power series $\{x_t\}_{t=1}^{T}$ as input, the RNN predicts the target power series $\{y_t^{(i)}\}_{t=1}^{T}$ of each device i . This is a sequence to sequence generation process. And at each time step t RNN predicts the corresponding $y_t^{(i)}$. The following formula describes this process.

$$y_t^{(i)} = g(x_1, x_2, ..., x_T; y_1^{(i)}, y_2^{(i)}, y_{t-1}^{(i)}) \tag{2}$$

Given the input x_t , LSTM computes its output by combining the information in its memory cell and hidden state with the input. The process can be written as

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \tag{3}$$

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{4}$$

$$\tilde{C}_t = \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right) \tag{5}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{6}$$

$$o_t = \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_o\right) \tag{7}$$

$$h_t = o_t * \tanh\left(C_t\right), \tag{8}$$

where i_t , f_t , and o_t are the input gate, forget gate, and output gate respectively; C_t is cell state and h_t is hidden state; and $\tilde{C}_t$ represents candidate cell state. And the target $y_t^{(i)}$ is computed using h_t .

In practice, LSTM layers are usually used together with other types of layers to form a deep RNN structure. The detailed architecture used in this paper is as follows:

1. Input (length T is appliance specific window size)
2. Parallel 1D convolution with filter size 3, 5, and 7 respectively, stride=1, number of filters=32, activation type=linear, border mode=same
3. Merge layer which concatenates the output of parallel 1D convolutions
4. Bidirectional LSTM consists of a forward LSTM and a backward LSTM, output_dim=128
5. Bidirectional LSTM consists of a forward LSTM and a backward LSTM, output_dim=128
6. Dense layer, output_dim=128, activation type=ReLU
7. Dense layer, output_dim= T , activation type=linear

Here we use bidirectional RNN to utilize information from the whole sequence of $\{x_t\}_{t=1}^{T}$, including both history and future information for each time step t . Furthermore, we use multiple layers of LSTMs so that each of them learns features at a different time scale.

IV. EXPERIMENTAL RESULTS

In this section, we describe the evaluation of the proposed deep learning models for NILM. First we briefly introduce the data set and how we prepare training and testing data. Then we present our experimental results with quantitative evaluation of our methods on this dataset and some qualitative discussions.

We evaluated our methods on the UK-DALE [16] dataset, which is currently the largest dataset for NILM research. More

specifically, we used the latest May 2016 release of this dataset. The UK-DALE dataset records power consumption of 5 houses in England. And for each house, it has both the whole-house mains power demand and the power demand of each individual appliance. For the majority of the power demand data, the sampling rate is 1/6 Hz. However, the sample rate of active and reactive mains power in houses 1, 2, and 5 is 1Hz, and we have to downsample them to 1/6Hz.

We also used the five appliances as in [1], the kettle, dish washer, fridge, microwave oven and washing machine to perform experiments. These devices consume the majority of energy and each of them exists in at least three houses in the dataset.

TABLE I. SETTINGS FOR EXTRACTING ACTIVATIONS

Appliance	On power threshold (watts)	Min. on duration (seconds)	Min. off duration (seconds)
Kettle	2000	12	100
Dish washer	10	1800	1800
Fridge	50	60	12
Microwave	200	12	30
Washing mach.	20	1800	160

For each appliance, we reserve data from one house (house 5) for testing in the experiments. Furthermore, to train our models we use the last week of available data from each house for validation, and all other available data as training data. The NILMTK [17] toolkit is used to perform data preprocessing and activation extraction. Each device has its own usage patterns, and so the parameters for extracting activations differ between

devices. And the detail settings of parameters are shown in Table I. We referenced the settings in [1] and modified them according to our own dataset.

Using settings in Table I, we finally extracted activations of each appliance from data of each house. Table II and Table III show the activations for training and testing. As mentioned above, activations for validation comes from the same houses as those for training. More specifically, there are 154, 18, 651, 101 and 17 validation activations for the kettle, dish washer, fridge, microwave oven and washing machine, respectively.

TABLE II. NUMBER OF ACTIVATIONS FOR TRAINING PER HOUSE

Appliance\House	1	2	3	4
Kettle	2995	757	43	711
Dish washer	200	98	0	0
Fridge	17006	3506	0	4663
microwave	3483	422	0	0
Washing machine	547	53	0	0

TABLE III. NUMBER OF TESTING ACTIVATIONS

Appliance	#activations
Kettle	195
Dish washer	46
Fridge	2977
microwave	66
Washing machine	109

We trained one network per target appliance since the consumption patterns between appliances differ greatly. As mentioned in section 3, the input of each network is a window of mains power, while the target (i.e. the desired output of the network) is the power demand of the target appliance. In addition, we chose a different window size for each device so that it is large enough to capture the majority of the activations of that device. In more detail, the window sizes are 40, 1075, 465, 72 and 1246 for the kettle, dish washer, fridge, microwave oven and washing machine, respectively. The deep networks were trained using Keras [18] which is a well-known toolkit for building and tuning deep learning models.

There are a number of metrics to measure the performance of energy disaggregation approaches, and here we use the most widely used Mean Absolute Error (MAE). Table IV presents MAE scores of our auto-encoder model PCNN_AE and our RNN model PCNN_LSTM as well as scores of baselines, where NILM_AE and NILM_LSTM are deep learning models proposed in [1], CO represents Combinatorial Optimization [3] and FHMM [5] is short for Factorial Hidden Markov Model.

From Table IV, we can see that although our auto-encoder model does not get promising performance, our LSTM based recurrent model outperforms other approaches for every device. And this shows the effectiveness of our method that uses multiple CNN with varying filter size as feature extractors. The NILM_LSTM model proposed in [1] achieves good performance as well, and this suggests the advantages of recurrent networks for the NILM task.

TABLE IV. MAE OF EACH METHOD (IN WATTS)

Method	Kettle	Dish washer	Fridge	Micro-wave	Washing machine
CO	70.23	75.91	71.51	42.37	70.30
FHMM	88.07	124.16	60.28	65.46	93.19
NILM_AE	34.47	77.51	3.34	16.93	115.26
NILM_LSTM	13.11	34.09	3.26	9.58	78.30
PCNN_AE	51.64	67.34	3.46	27.50	83.40
PCNN_LSTM	**10.21**	**29.62**	**3.22**	**9.42**	**73.16**

V. CONCLUSIONS AND FUTURE WORK

In this paper, we have empirically studied energy disaggregation using based on learning models. We first formalized the NILM task into a sequence to sequence prediction problem. Then we presented the feature extracting method that uses a parallel of multiple convolution layers and describe two deep neural networks for NILM. Finally, we evaluated the performance of our methods using the largest NILM dataset available. The experimental results showed the advantages of our recurrent neural based model.

Energy disaggregation (NILM) is a very challenging and can be improved in a number of ways. First, more data are required since deep learning models usually need a lot of training data. Second, the power consumption patterns of appliances differ greatly. And to achieve best performance, we may need to construct different models for each type of appliance. And finally, the LSTM model presented in this paper does not explicitly use the dynamic information of series in the prediction process. And the attention mechanisms which

have achieved great success in machine translation may also be promising in NILM.

ACKNOWLEDGMENT

We thank Mr Feng Jin for his kind help on solving issues in deep learning and Keras.

REFERENCES

[1] Kelly J, Knottenbelt W. Neural NILM: Deep Neural Networks Applied to Energy Disaggregation[C]//Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments. ACM, 2015: 55-64.

[2] Hart G W. Prototype Nonintrusive Appliance Load Monitor[M]//MIT Energy Laboratory Technical Report, and Electric Power Research Institute Technical Report. 1985.

[3] Hart G W. Nonintrusive Appliance Load Monitoring[J]. Proceedings of the IEEE, 1992, 80(12): 1870-1891.

[4] Kaggle. Belkin Energy Disaggregation Competition[EB/OL]. [2013]. https://www.kaggle.com/c/belkin-energy-disaggregation-competition.

[5] Ghahramani Z, Jordan M I. Factorial Hidden Markov Models[J]. Machine learning, 1997, 29(2-3): 245-273.

[6] Kim H, Marwah M, Arlitt M F, Lyon G., Han J. Unsupervised Disaggregation of Low Frequency Power Measurements[C]//SDM. 2011, 11: 747-758.

[7] Kolter J Z, Jaakkola T S. Approximate Inference in Additive Factorial HMMs with Application to Energy Disaggregation[C]//AISTATS. 2012, 22: 1472-1482.

[8] Kolter J Z, Batra S, Ng A Y. Energy Disaggregation via Discriminative Sparse Coding[C]//Advances in Neural Information Processing Systems. 2010: 1153-1161.

[9] Elhamifar E, Sastry S. Energy Disaggregation via Learning Powerlets and Sparse Coding[C]//AAAI. 2015: 629-635.

[10] Schmidt M N, Larsen J, Hsiao F T. Wind Noise Reduction Using Non-negative Sparse Coding[C]//2007 IEEE Workshop on Machine Learning for Signal Processing. IEEE, 2007: 431-436.

[11] Goodfellow I, Bengio Y, Courville A. Deep Learning[M]. MIT Press, 2016.

[12] Hochreiter S, Schmidhuber J. Long Short-term Memory[J]. Neural Computation, 1997, 9(8): 1735-1780.

[13] do Nascimento P P M. Applications of Deep Learning Techniques on NILM[D]. Universidade Federal do Rio de Janeiro, 2016.

[14] Szegedy C, Liu W, Jia Y, et al. Going Deeper with Convolutions[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 1-9.

[15] Johnson R, Zhang T. Effective Use of Word Order For Text Categorization with Convolutional Neural Networks[C]//Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2015: 103-112.

[16] Kelly J, Knottenbelt W. UK Domestic Appliance-Level Electricity (UK-DALE) dataset[OL]. [2016]. https://www.doc.ic.ac.uk/~dk3810/data/.

[17] Kelly J. NILMTK: Non-Intrusive Load Monitoring Toolkit[CP/OL]. https://github.com/nilmtk/nilmtk

[18] Chollet F. Keras: Deep Learning library for TensorFlow and Theano[CP/OL]. https://github.com/fchollet/keras.