American Journal of Information Science and Computer Engineering
Vol. 1, No. 2, 2015, pp. 53-58
http://www.aiscience.org/journal/ajisce

An Energy Optimizing Model of Mobile Terminal Software Based on BP Neural Network

Zhengying Cai*, Tengfei Zhan, Liwen Liu

College of Computer and Information Technology, China Three Gorges University, YiChang, China

Abstract

Software energy consumption of mobile terminal plays an important role in the continuous duty and stability of mobile business. Here an energy management model is built based on a multi-layer software structure and is described as a multi-variable function of energy consumption. The relation between software characteristics and energy consumption is discussed and analysed. Moreover the paper presents an optimal energy model by BP neural network for mobile software management, and lists the solving steps. The BP neural network can extract the component number from multi-layer software structure to optimize the energy consumption by multi-inputs. Lastly an experiment is designed for illustration and the results verify the proposed model, which can optimize the complex software energy by adaptive neurons and deduce the complexity of energy modelling.

Keywords

Software Energy, Energy Modelling, BP Neural Network, Mobile Terminal

Received: May 1, 2015 / Accepted: May 15, 2015 / Published online: June 14, 2015

1. Introduction

Along with society's progress and the rapid development of economy, energy problem has been more and more serious, and the energy crisis has become the focus of the whole social attention (Bai et al, 2014). Now the idea of energy saving has been deeply rooted in the hearts of the people. Ranging from study, work to life, nearly everything is closely related to our energy consumption, especially by all kinds of mobile terminal (Li et al, 2011) (Li et al, 2013). Although mobile application has made our lives easier, but the power consumption problem and too short running time of intelligent terminal still plague us today (Wang et al, 2014), which is often required to run light weight mobile service(Ruiz-Martinez, et al, 2014).

As the energy consumption of mobile terminal, it mainly composes of two kinds of energy consumers, namely hardware and software. Among all kinds of hardware in mobile terminal, the core and CPU consume most energy, and gain most attention. Bolla et al (2014) studied the energy adaptation in multi-core software routers, and Mishra et al (2014) designed an energy-efficient voltage scheduling for multi-core processors with software controlled dynamic voltage scaling.

In fact, all energy consumed in core or CPU comes from all kinds of software running in mobile terminal (Bolla et al, 2014), so some researches on energy consumption of software are appeared. Zhu et al (2013) proposed a high-performance and energy-efficient mobile web browsing on big/little systems, and Zhao et al (2015) studied a multi-domain control scheme for Diffserv QoS and energy saving consideration in software-defined flexible optical networks. Xing et al (2015) analysed the energy efficiency retrofit scheme for hotel buildings using eQuest software, and made a case study from Tianjin, China.

* Corresponding author
E-mail address: master_cai@163.com (Zhengying Cai)

In energy consumption control of software, different software consume energy differently, and the software architecture of mobile terminate is complex composing of many different software. Different researchers built different models from each other. For example, Karmellos et al (2015) built a multi-objective software model for optimal prioritization of energy efficiency measures in buildings, Li et al (2012) presented an application of mobile agent model in mobile payment, Li et al (2013) modelled a middleware architecture for price comparison service on mobile phones, and Peng et al (2014) put forward a layered context-aware agent for mobile applications based on user needs hierarchy.

Generally the main function of mobile e-commerce software is to keep the stable operation of commerce business and fit the need of users, and lightweight software is welcomed in mobile e-commerce research, such as Android development (Liu et al, 2012). Therefore function, stability, energy consumption of mobile software form a multi-objective function. For example, Song et al (2014) considered an environmental psychology perspective for application discoverability and user satisfaction in mobile application stores, and Sun et al (2014) researched the mobile business value network and model construction. Therefore, it is difficult to optimize the energy consumption of mobile software by traditional mathematical model or algorithm, and artificial intelligent algorithm maybe helpful. Bai et al (2014) built a predictive model of energy cost in steelmaking process based on BP neural network.

This article studies the energy consumption problem of mobile terminal software, and the whole paper is organized as follows. Section 1 reviews related researches. Section 2 puts forward a multi-level energy consumption model of mobile terminal software, and classifies different software according to their function and architecture level. Section 3 builds a nonlinear function relationship between software energy consumption and their characteristics, and solves the total energy consumption model of mobile terminal software architecture by BP neural network algorithm. Section 4 designs a numerical experiment to verify the proposed model. Section 5 draws some interesting conclusions and forecasts the future works.

2. Energy Consumption Model of Mobile Terminal Software

Mobile terminal software is often lightweight and is operating at all kinds of non-standard mobile terminals to meet different business requirements. The mobile terminal software can help terminal operators and service providers to dispose and defragment all kinds of business and recommendations. Mobile terminal software gathers and drives mobile terminal requirements by multi-level architecture, and processes and interchanges business data by their communications. Mobile terminal software is technology neutral, which means its data interactions and recommendations intended for terminal users across the multi-level technology platforms, operating systems and middleware layers.

The common mobile terminal software architecture is shown in Figure 1 and there are 4 levels play an important role in the mobile terminal system. The lowest bottom is platform layer with hardware driver, protocol stack and operating system, and is the basis of the whole mobile terminal software. GUI level runs on the Operating System Abstraction Layer (OSAL) and the Protocol Stack Abstraction Layer (PSAL). PSAL refers to the set of protocol in each layer of network, and reflects a network file transfer process. OSAL, PSAL, and Device Abstraction Layer (DAL) are all located at the interface layer between the operating system kernel and hardware circuit layer. The highest layer composes of all kinds of applications, such as phonebook, call, SMS, set and so on.

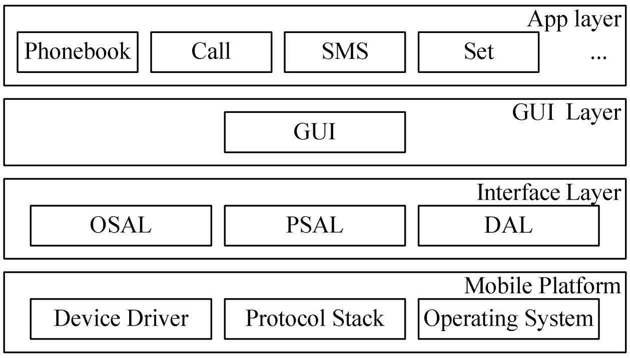

Figure 1. Software structure of mobile terminate.

The multi-layer software system is mainly to fit the need of a variety of users including business managers, operators and customers, but all these users have their own requirements with respect to the mobile system. To balance these concerns and demonstrate their requirements result in the complex system structure. This implies that multi-layer architecture involves a broad variety of concerns to be dealing with, and every layer has a unique nature.

Apparently different software in the structure consumes energy different from each other, and their importance is also different. Generally energy consumption modelling of multi-layer software structure can be described as a linear regression method, mainly considering the software characteristics in different layer. Assuming that there is n software with multi-layer structure, the total energy consumption E can be described as a multi-variable linear function relationship between different software and their energy consumption models as follows

$$E = a_1 + a_2 b_1 + a_3 b_2 + \ldots + a_{n+1} b_n \quad (1)$$

Where E is the software energy consumption, a_1 is the software initialization of the consumption of energy, b_n is the software characteristic value which is associated with energy consumption of mobile software, the value of a_n represents the parameter of the characteristics of various software.

More generally the device driver, protocol stack, operating system, OSAL, PSAL, and DAL come from different designers with different design characteristics, meaning the relationship between energy consumption and software structure are affected by many factors and characteristics. Hence the relationship between the energy consumption of software and software characteristics in different level can be modelled as a nonlinear function relation, and we can get a new structure model of energy consumption as equation(2) .

$$E_s = A \times T = f(B) \times T = f(L, C, R_c, R_w, DC) \times T \quad (2)$$

In the above formula, E_s is the consumption of software and A is the average power consumption. Also, T is the software running time and B is some software metrics associated with power consumption characteristics of software. f is a nonlinear function between the power consumption of software and software-related measurement features.

The established way for software structure to reduce the structural complexity is to separate the nonlinear function because different software consumes power nonlinearly, and some software will cost power greatly in a certain environment. The nonlinear energy function shows that all concerns are addressed by different parameter modelling and describes the dynamics performance of energy consumption from separate points of view with the various concerns.

However, the nonlinear relationship between software system structure characteristics and software energy consumption is difficult to be solved by traditional algorithm. In order to solve the nonlinear relationship more accurately, choosing a reasonable and effective artificial intelligent method is very necessary. In machine learning and many sciences, artificial neural networks (ANNs) are very popular inspired from biological neural networks where there is a family of nervous systems for statistical learning as the brain. ANNs can be used to estimate or approximate the energy functions of mobile software depending on generally unknown multi-inputs. Hence, the BP neural network is introduced in the next section to model the energy consumption problem of mobile terminal software.

3. The BP Neural Network for Energy Optimization

3.1. The Fitting Process of Nonlinear Function

Neural networks are artificial intelligent algorithm similar to biological neural networks such as performing functions collectively and in parallel by the units, different from traditional algorithm where there is a clear definition of subtasks to be assigned with various units. The BP neural network is one of the most popular neural networks and refers to artificial intelligent models employed in statistics and cognitive psychology. Especially in modern software research and development, the artificial neural networks approach inspired by biology developed toward a more practical approach based on statistics and probability processing. In these software systems, neural networks or parts of neural networks can be taken as an artificial neurons or components in larger software systems to integrate both adaptive and non-adaptive metrics. Generally BP neural network approach is more suitable for these software modelling and real-world problem solving, and it structure is shown as figure.2.

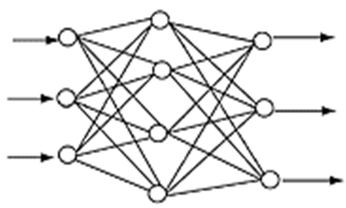

Figure 2. Structure of BP neural networks.

As we can see from it, BP neural networks are generally presented as systems of interconnected neurons, as circles in figure 1, to compute values from multi-inputs, and are capable of machine learning and pattern recognition due to their adaptive structure. There are three layers of nodes, including input layer, medial layer and output layer, and has little to do with the traditional algorithm models. As a BP neural network to carry on the fitting, L is the lines of effectively software code, C is component quantity, R_c is average complexity of component interface, the R_w is average path complexity, DC is the average component coupling. On the basis of the characteristics of software structure, modelling method of energy consumption level is divided into the following five steps:

(1) Assuming that there is a nonlinear function relationship between energy consumption and software characteristics of structure level.

(2) Describing the aspects of system structure and the energy consumption characteristics of software, and classifying

software by the following five characteristics: effective lines of code, number of components, the average complexity of component interface, the average path complexity, and the average component coupling.

(3) Choosing reasonable measurement methods to evaluate these indexes as number of code line, number of components, the average complexity of component interface, the average path complexity, and the average component coupling.

(4) Comparing the power consumption with the set of benchmark programs, which can be measured according to the formula (1) then it is concluded the average power consumption during the software running value.

$$P = E_s / T \tag{3}$$

(5) Training the BP neural network to fit the input values of the nonlinear correlation functions *f* for each software structure level and different characteristics measurements, and output the optimized average power for the whole mobile software system.

In the training process of BP neural network, it is needed to realize nonlinear fitting of the specific process.

For large sample program of 4 layers software structure, 5 kinds of characteristics (effective lines of code, member number, the average complexity of component interface, the average path complexity, and the average component coupling) were measured from these software. Resulted five values of software metric depict the pre-processing characteristics of energy, and are fed into the BP neural network as the input values after processing.

3.2. Solving Steps of the BP Neural Network

After getting effective structure characteristic measurements, it is necessary to further determine the structure of the BP neural network, and factors to be considered including the neural network learning rate, approximation error, the rate of convergence and memory footprint etc.

(1) The number of hidden layer. The number of hidden layer determines the error of the BP network, but too much number of hidden layer would make the network structure more complicate and increase the tendency of network training time and the fitting. A three layers BP network can be completed as any dimensional mapping from *n* to *m*. Therefore, according to the characteristics of mobile software, the number of hidden layer is determined as 1.

(2) The number of hidden layer nodes. For BP neural network, number of hidden layer nodes plays a great role on BP network performance. It is typical of us to determine the node number of hidden layer by the following empirical formula *n*:

$$n = \sqrt{a+b} + i \tag{4}$$

Among them, *a* is input layer node number, *b* is the output layer node number, and *i* is the constant between 1 and 10.

(3) The identification of functions of each layer. As the BP neural network for single hidden layer, broadly speaking, the transfer functions of input layer and hidden layer are *tansig* function, and the transfer function of the output layer is *purelin* function. As *trainlm* function, its convergence speed and the approximation error can achieve satisfactory results when the node number of hidden layer is 11 which can be found through many experiments. So, the training function of BP neural network is *trainglm*, and the hidden layer node number of training function is 11.

After determining the various parameters, the network algorithm can be solved and tested. Through a large number of sample data inputs, the weights of BP neural network w_1, w_2 and threshold b_1, b_2 can be gotten when error reaches a predefined precision.

4. Experiment and Analysis

In order to verify the effectiveness of the proposed model based on multi-layer characteristics of software energy consumption, this paper adopts several experimental programs containing common multi-media web file, MP3 file and decoding program, which is very popular in mobile terminal.

Table 1. The level table of the mobile software.

Software Level	Software	Description	Number of Software
1	hardware driver, protocol stack and operating system	Obligatory, low power	3
2	OSAL, PSAL, and DAL	Obligatory, but power-wasting	3
3	GUI	Optional, but power-wasting	3
4	phonebook, call, sms, set and so on	Optional, partly power-wasting	3

First, this work distributes the software into 4 levels with different characteristics of energy consumption as description in Table 1. The 4 levels of software are in accordance with the structure in Figure 1 as mentioned above. Then there are 3 software in 4 levels respectively used to calculate the energy performance in the last column of table 1. Running a sample programs on simulation experiment platform Matlab

7.0, the sample process energy consumption values of *E* can be gotten, and *E* will serve as the output value of the BP neural network.

Second, after determining the specific structure of the BP neural network, the experiment includes the number of hidden layer, hidden layer node number and transfer function of hidden layer and output layer transfer function, etc. After inputting sample applications of software, the different measured characteristics value and energy consumption values can be calculated to train the BP neural network and determine the weights of BP neural network and threshold of hidden layer. The expected energy output and optimized energy output are shown in Figure 3.

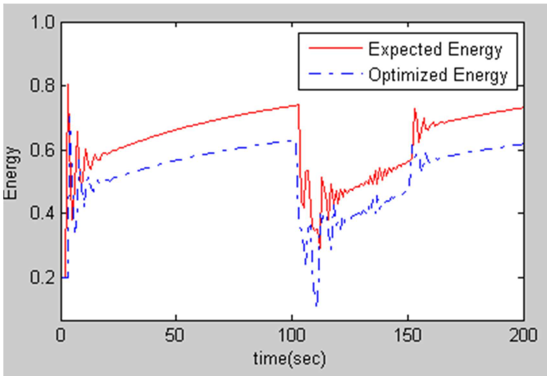

Figure 3. Total energy output.

As we can see, there is a boot-strap at the starting up of mobile terminal, then the whole energy keeps at a stable value about 0.6-0.8 of expected value, but the optimized energy is lower than it at about 0.4-0.6. Therefore inputting the characteristics of target program to the trained BP neural network, the energy consumption is optimized. When time=100, another mobile software begin to run, as shown the Software 2 in Figure 4. The BP neural network can get the predictive value of energy consumption, and then compare with the actual energy consumption of target program, to optimize the software running and lessen energy consumption.

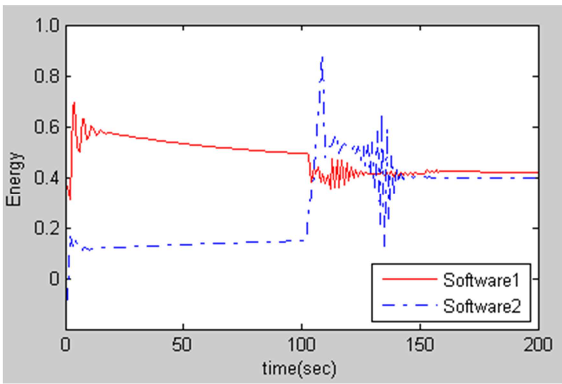

Figure 4. Energy of Software.

In Figure3 and figure 4, target application software can be applied to predict the energy consumption and optimize the actual energy consumption at about 18% (0.5/0.6 in Figure 2). The BP neural network has good numerical approximation, and it can approximate any nonlinear function in math with higher fitting degree.

When new software (software 3 in figure 4) begins to work, the energy requirement is also risen, and the neural network can also optimize it. Though the boot-strap energy of the software 2 is about 30% (0.9/0.6 in Figure 3), the energy still is lower than the boot stage and reaches a stable value at about 0.4. At the same time the software 1 remains a totally down tendency as shown in Figure 4.

To observe the learning process of neural network, a specific environment experiment is made in this section to show the training results in figure 5 and 6. It is not hard to see that there is a certain correlation between each lever of energy consumption and training of neurons, and the most obvious characteristics is the layer 3 change greatly which is on behalf of the effective energy adjusting of GUI.

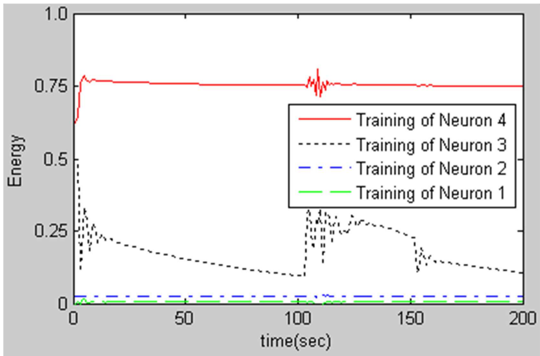

Figure 5. Training of Neuron.

The software layer introduced by this paper shows different characteristics of energy consumption according to different software layer, for example software 4 cost the maximum energy as 0.75. The software structure of a mobile terminal system represents an overall energy vision for neural optimization which can justify what it should do and how it should do it.

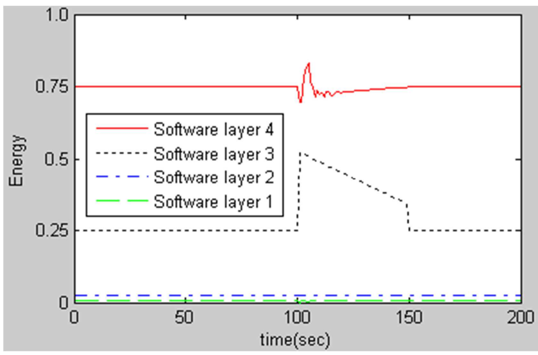

Figure 6. Energy of Software layer.

In figure 6, this division with 4 software layers should be separated from its implementation, so the layer 1 and layer 2 keep stable in the whole process which is the operational basic of the mobile terminal. The 4-layers of structure assumes the role of keeper to the vision, then the upper layer such as layer 3 and layer 4 in figure 6 can make sure that additional software needed by the user are in line with the software structure, preventing higher energy consumption at the same time.

Form the experiment above, apparently the level-based energy optimization approach proposed in this study can effectively to measure the energy design of mobile app and service performance. It can also help us to evaluate and choose appropriate software projects in the design stage. Different mobile terminals and other devices can also take advantage of the energy information in early research and development to understand and optimize the energy mechanism to improve mobile application.

5. Conclusions

This paper describes the energy problem of mobile terminal as a function fitting method, and establishes a multi-layer model of energy consumption. Then the simulation experiments are designed to verify the nonlinear function relationship between software characteristic and energy consumption. The future work is to further research these software characteristics and optimization problem of energy consumption for more kinds of mobile software.

References

[1] Bai, Zhenchao; Wei, Gang; Liu, Xianhui; et al. Predictive model of energy cost in steelmaking process based on BP neural network [C]. 2nd International Conference on Software Engineering, Knowledge Engineering and Information Engineering (SEKEIE), Singapore, AUG 05-06, 2014, Proceedings Of The 2nd International Conference On Software Engineering, Knowledge Engineering And Information Engineering (SEKEIE 2014): Advances in Intelligent Systems Research, 2014 (114): 77-80.

[2] Bolla, Raffaele; Bruschi, Roberto; Lago, Paolo. Energy adaptation in multi-core software routers [J]. Computer Networks, 2014, 65(6): 111-128.

[3] Karmellos, M.; Kiprakis, A.; Mavrotas, G. A multi-objective approach for optimal prioritization of energy efficiency measures in buildings: Model, software and case studies [J]. Applied Energy, 2015, 139(2): 131-150.

[4] Li, Gangmin; Shen, Zhun. A middleware architecture for price comparison service on mobile phones [C]. 1st International Conference on Information Technology and Quantitative Management (ITQM). (Suzhou, Peoples R China, May 16-18, 2013), 2013, 17: 545-553.

[5] Li, Yan; Hu, Xiaoqiang; Zeng, Liang. The application of mobile agent in mobile payment[C]. International Conference on Computer Science and Network Technology (ICCSNT). (Harbin Normal Univ, Harbin, Peoples R China, Dec 24-26, 2011), 2012, 1-4:1612-1616.

[6] Liu, Yan-jun; Liu, Zu-peng. Android development and mobile e-commerce research[C]. 2nd International Conference on Mechanical Engineering and Green Manufacturing (MEGM 2012 (Hunan Inst Engn, Chongqing, Mar 16-18, 2012), 2012, 155-156:430-434.

[7] Mishra, Abhishek; Tripathi, Anil Kumar. Energy efficient voltage scheduling for multi-core processors with software controlled dynamic voltage scaling [J]. Applied Mathematical Modelling, 2014, 38 (14): 3456-3466.

[8] Peng Min-jing; Li Bo; Liu Meng. A layered context-aware agent for mobile applications based on users needs hierarchy [C]. International Conference on Mechatronics and Semiconductor Materials (ICMSCM 2013). (Xian, PEOPLES R CHINA, SEP 28-29, 2013), 2014, 846-847: 1689-1692.

[9] Ruiz-Martinez, A.; Inmaculada Marin-Lopez, C; Sanchez-Martinez, D; SIPmsign: a lightweight mobile signature service based on the Session Initiation Protocol[J].Software-Practice & Experience,2014,44(5):511-535.

[10] Song, Jaeki; Kim, Junghwan; Jones, Donald R. Application discoverability and user satisfaction in mobile application stores: An environmental psychology perspective [J]. Decision Support Systems, 2014, 59:37-51.

[11] Sun, Jie; Ren, Wei'an. Research on mobile business value network and model construction [C]. International Conference on Mechatronics and Information Technology (ICMIT 2013). (Guilin, Peoples R China, Oct 19-20, 2013), 2014, 462-463: 849-855.

[12] Wang, Jiangjing; Jiang, Jiulei; He, Feng. Problem and countermeasure on the development of mobile electronic commerce in China[C]. International Conference on Instrumentation, Measurement, Circuits and Systems (ICIMCS 2011). (Hong Kong, Peoples R China, Dec 12-13, 2011), 2012, 127: 861-866.

[13] Xing, Jincheng; Ren, Peng; Ling, Jihong. Analysis of energy efficiency retrofit scheme for hotel buildings using eQuest software: A case study from Tianjin, China [J]. Energy and Buildings, 2015, 87(1): 14-24.

[14] Zhao, Jijun; Yao, Qiuyan; Ren, Danping; et al. A multi-domain control scheme for diffserv QoS and energy saving consideration in software-defined flexible optical networks [J]. Optics Communications, 2015, 341(4): 178-187.

[15] Zhu, Yuhao; Reddi, Vijay Janap. High-performance and energy-efficient mobile web browsing on big/little systems[C]. 19th IEEE International Symposium on High Performance Computer Architecture (HPCA). (Shenzhen, Peoples R China. Feb 23-27, 2013), 2013: 13-24.