2017 American Control Conference
Sheraton Seattle Hotel
May 24–26, 2017, Seattle, USA

A New Concept using LSTM Neural Networks for Dynamic System Identification

Yu Wang

Abstract— **Recently, Recurrent Neural Network becomes a very popular research topic in machine learning field. Many new ideas and RNN structures have been generated by different authors, including long short term memory (LSTM) RNN and Gated Recurrent United (GRU) RNN ([1],[2]), a number of applications have also been developed among various research labs or industrial companies ([3]-[5]). Most of these schemes, however, are only applicable to machine learning problems, or static systems in control field.**

In this paper, a new concept of applying one of the most popular RNN approach - LSTM to identify and control dynamic system is to be investigated. Both identification (or learning) dynamic system and design of controller based on identification are going to be discussed. Also, a new concept of using a convex-based LSTM networks for fast learning purpose will be explained in detail. Simulation studies will be presented to demonstrated the new LSTM structure performs much better than conventional RNN and even single LSTM network.

I. Introduction

Neural Network has a long history in scientific research. The earliest description about neural networks can be traced back to the early 1940s, psychologist Donald Hebb invent a learning scheme known as Hebbian Learning base on the neural plasticity mechanism ([6]). In 1958, Frank Rosenblatt created the perceptron, which is known as the principal components of neural network nowadays, and built a two layers neural network without any training procedure ([7]). The current most widely back-propagation algorithm was designed and published by Paul Werbos in his Ph.D thesis in 1975([8]). Since then, various neural network structures are proposed for different problems in control and machine learning fields, which includes feed-forward neural network, recurrent neural network, auto-encoder, time-delayed neural network and etc. The development of training large scale complex neural networks had become very slow or even stagnated. There are main two reasons:

1) The computational powers is not fast enough for training neural networks by computing their weights through back-propagation, especially when the networks has multiple layers and vast numbers of hidden nodes.

2) The vanishing gradient problem: the gradient of errors will vanish gradually through the back-propagation process. This issue was firstly addressed by Hochreiter in 1991 ([10]), which is also treated as the seed of deep learning. During more than one decade since early 2000s, more researchers started working on numerous deep neural network structures, most of which contain complex network structures with still reasonable training speeds.

Long-short term memory network, or abbreviating as LSTM, is one of most popular recurrent neural network structure in deep learning field. Invented by Schmidhuber in 1997 ([1]), LSTM avoids the vanishing gradient issue by adding three gated units: forget gate, input and output gates, through which the memory of past states can be efficiently controlled. LSTM is widely used in many areas, mostly in machine learning application field, including speech recognition, natural language processing and other pattern recognition applications. The use of LSTM for system identification in control field, however, has never been addressed by any existing literature. The reasons is mostly because:

1) Most of the system identifications using neural networks are nonlinear system, which requires multiple layers neural network and vanishing gradient is an issue during earlier years.

2) Unlike a typical machine learning problem, most of the systems to be controlled are in a dynamic and on-line manner, hence the speed requirement for designing a neural network structure for system identification is very high.

In the paper, both of these two problems will be addressed. The first problem can be mainly solved by the LSTM structure itself and the second one will be conquered by using the main technique to be introduced in this paper: a convex-based LSTM neural networks structure.

II. Mathematical Preliminaries

In this section, concepts related to dynamic systems, neural networks, long-short term memory neural network and system identification would be presented and explained. The major objective of this section is to provide readers some of the important prior knowledge before the main parts of the paper and also for easy reference purpose.

A. Dynamic System Representation

In conventional control theory of dynamic systems, there are mainly two types of system representations: state-space form and input-output based. The major difference between two forms is that, input-output approach mainly assume the inaccessibility of states while state-space form assume the full or partial accessibility of systems states. In this paper, due to the necessity of states information of system, the state-space approach is used as the major form. Follow are the differential equation representations of the system in a

†: Yu Wang is with the Department of Electrical Engineering, Yale University, New Haven, 06510

978-1-5090-5992-8/$31.00 ©2017 AACC

general form:

$$\begin{aligned} \dot{x}(t) &= f[x(t), u(t)] \quad t \in \mathbb{R}^+ \\ y(t) &= g[x(t)] \end{aligned} \tag{1}$$

where $x(t) = [x_1(t), x_2), ...x_m(t)]^T$ is the system states, $u(t) = [u_1(t), ...u_k(t)]$ is the input, and $y(t) = [y_1(t), ..., y_n(t)]$ is the output. f is a mapping from $\mathbb{R}^m \times \mathbb{R}^k$ to $\mathbb{R}^m$, and g is also a mapping from $\mathbb{R}^m$ to $\mathbb{R}^n$. Notice that both f and g can be linear or non-linear, but the approaches to deal under two different scenarios will be totally different.

Also, the general system can be considered in discrete space, which will put it into a form as :

$$\begin{aligned} x(k) &= f[x(k-1), \cdots, x(k-n), u(k-1), \cdots, u(k-m)] \\ &\quad (k \in \mathbb{Z}^+) \\ y(k) &= g[x(k)] \end{aligned} \tag{2}$$

where the system mapping for f and g are the same as in general case, instead that $k \in \mathbb{Z}^+$, and the inputs, internal states and outputs are discrete sequences.

The system can be further simplified to linear systems by mapping systems representation into a linear space. The illustration, identification and controller design of such systems are very mature and well developed, hence it will not be discussed in detail in this paper. In this paper, we will use the discrete form of general system representation for further explanation.

B. Some Theorems

In this subsection, two important theorems will be presented: the Stone-Weierstrass Theorem, and Universal-Approximation Theorem. Following is the well-known The Stone-Weierstrass Theorem

Theorem 1 *The Stone-Weierstrass Theorem:*

Suppose X is a compact Hausdorff space and A is a subalgebra of $C(X, \mathbb{R})$ which contains a non-zero constant function. Then A is dense in $C(X, \mathbb{R})$ if and only if it separates points.

By using this theorem, it can be shown that the a nonlinear equation under certain conditions can be represented by series like Wiener series. This will then lead to the discover of universal approximation theorem.

Theorem 2 *The Universal-Approximation Theorem*

Let $\varphi(t)$ be a nonconstant, bounded, and monotonically-increasing continuous function. Let I_m denotes the m-dimensional unit hypercube $[0, 1]^m$. The space of continuous functions on I_m is denoted by $C(I_m)$. Then, given any function $f \in C(I_m)$ and $\varepsilon > 0$, there exists an integer N, real constants v_i,$b_i \in \mathbb{R}$ and real vectors $w_i \in \mathbb{R}^m$, where $i = 1, \cdots, N$, such that we may define:

$$F(x) = \sum_{i=1}^{N} v_i \varphi\left(w_i^T x + b_i\right) \tag{3}$$

as an approximate realization of the function f where f is independent of φ ; that is,

$$|F(x) - f(x)| < \varepsilon \tag{4}$$

or all $x \in I_m$. In other words, functions of the form $F(x)$ are dense in $C(I_m)$.

III. LSTM Neural Networks

In this section, a brief overview of LSTM Neural Network will be introduced. Before that, some basic information on conventional Recurrent Neural Network, back-propagation and their common issue: vanishing gradient problem, will be presented. These together will give a solid reason for alternating from Simple Neural Network Structure in early work[11]-[12] to LSTM for the system identification purpose.

A. Conventional RNN

Where $w(h)$, $w(x)$ and $w(y)$ are the states weight, input weights and output layer weight correspondingly, all of them are functions of k. And g is a nonlinear activation function.

From both the graphical illustration and mathematical presentation, it is shown that the output of recurrent neural network $y(k)$ is dependent on two parameters, i.e. input $x(k)$ and the feed-back internal state $h(k-1)$.

Theoretically, due to the recurrent property, current output $y(k)$ should be affected by all the internal states $h(k)$ where $k = \{0, \cdots, m-1\}$, and m is the memory step. However, the stored information over extended time intervals is very limited in a short term memory manner due to the decaying error feedback. This effect is also normally called as Vanishing Gradient Issue, which will be explained in detail in next section.

B. Back-propagation and Vanishing Gradient Issue

In 1991, Hochreiter explained in his paper [10] about the vanishing gradient issue in detail when back propagation approach (BPTT) is applied to the error signal for generating network weights. It is claimed in the paper that the error trained by conventional BPTT approach will be decaying exponentially as time elapsed. A simple explanation is as following. Considering the mathematical equation for output error $e_k(k)$

$$e_k(k) = f_k^{'}(g_k(k)(y_k(k) - \hat{y}_k(k)) \tag{5}$$

where $\hat{y}_i(k) = f_i(g_i(k))$ the identification result or learning result obtain from the neural network by using the activation function $g_i(k)$. Here the $e_k(k)$ is the output error. Similarly, for any non-output error $e_i(k)$ within the hidden layers, the error can be represented as:

$$e_i(k) = f_i^{'}(g_i(k) \sum_i w_{ij} e_i(k+1) \tag{6}$$

Once one have obtained the error equations (5) and (6), it is very easy to further derive the scaling factor for the error occurred at iteration k propagated backed into m time steps.

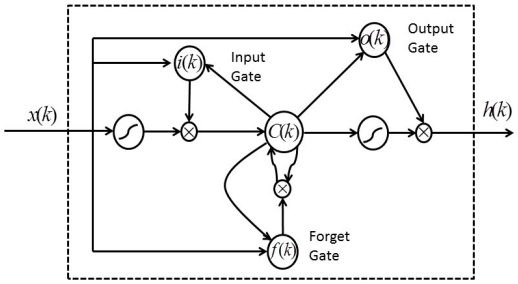

Fig. 1: LSTM Structure

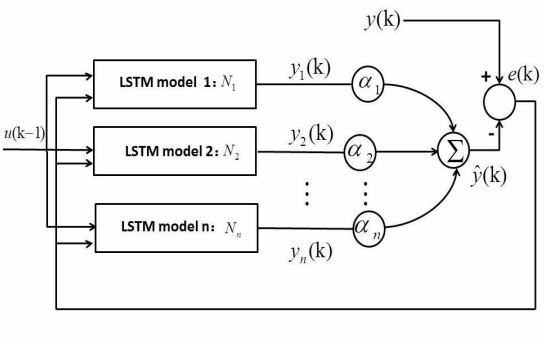

Fig. 2: Convex based LSTM Structure

The detail derivation will be omitted due to space limitation, only the result is shown as below:

$$\frac{e_a(k-m)}{e_b(k)} = \sum_{l_1=1}^{n} \cdots \sum_{l_{m-1}=1}^{n} \prod_{q=1}^{m} f'_{l_q}(g_{l_q}(k-q) w_{l_q l_{q-1}} \quad (7)$$

where q is from 1 to m, and the error flow from l_1 to l_m. It is stated in the paper that when the term in the product $-1 < f'_{l_q}(g_{l_q}(k-q)w_{l_q l_{q-1}} < 1$ for all q, then the largest product decrease exponentially with q, i.e. gradient of error vanishes.

C. LSTM with Gated Units

To overcome the issue of vanishing gradient, in 1997, Hochreiter proposed a RNN structure with Gated Units, named LSTM. A simple overview of the scheme will be discussed in this subsection.

To ensure a constant error overflow in LSTM, a memory cell, is added to the structure. Functioned as a sluice gate and controller of storing past states, the memory cell contained three gated units: input gate $i(k)$, output gate $o(k)$ and forget gate $f(k)$. The structure of LSTM is below:

The math formulation of LSTM is as below:

$$\begin{aligned} f(k) &= g(W_f(h(k-1), x(k)) + b_f) \\ i(k) &= g(W_i(h(k-1), x(k)) + b_i) \\ o(k) &= g(W_o(h(k-1), x(k)) + b_o) \\ \tilde{C}(k) &= \tilde{g} W_c(h(k-1), x(k)) + b_c \\ C(k) &= f(k)C(k) + i(k)\tilde{C}(k) \end{aligned} \quad (8)$$

where $g(\cdot)$ is the activation function for input, output and forget gates, which is normally chosen as the sigmoid function. $\tilde{g}(\cdot)$ is the activation function for the memory cell state $\tilde{C}$, which can use $tanh$ for general cases. From the equation above, $f(k)$ is the forget gate which will select which part of memory is going to be passed to next step. The output of $f(k)$ is a number between 0 and 1 (when choosing sigmoid as $g(\cdot)$). And from the last formula in the equation list, memory cell will be fully passed to next state when $f(k)$ is equal to 1 and forgotten or thrown away when $f(k)$ is equal to 0. That is also why it is called Long-Short Term Memory. The main advantage of LSTM is that this structure can effectively avoid the vanishing gradient phenomenon and hence be selected as the RNN structure for system identification in this paper.

IV. A Convex-Based LSTM Cluster

Though the universal approximation theorem indicated that when appropriate activation functions and number of hidden nodes are chosen, the LSTM RNN structures can identify any nonlinear system structures described in equation (2), the training speed of a conventional BPTT as shown in equation (7) will be still very slow. In this section, a newly designed convex-based LSTM structure will be introduced, and the general idea is inspired by the early work in the field of adaptive control [13]-[16].

In adaptive control, one of the popular research direction is Multiple Models based adaptive structure. The principal idea is to use more than one models to make decisions for the identification or control purpose, similar idea has also been used in learning field [17]-[20]. The information can be obtained by a selected model among multiple models, or the collective information from all the models, which is also called as the second level adaptation [14]. The major advantage of using second level adaptation is that it dramatically increases the convergence speed during system identification by updating the convex coefficients of each model instead of models themselves. Here, a similar concept will be applied on identifying the discrete system using multiple LSTM neural networks. Following is a structure of the convex-based LSTM neural networks:

In the structure, n multiple LSTM Neural Networks $N_i(i = \{1 \cdots n\})$ with the same network structure (number of layers and hidden nodes)are used and connected by n convex coefficients α_i, which have values within the range [0,1]. The convex-based LSTM Neural Networks satisfy the following three properties:

1. $\sum_i \alpha_i$=1. $(i = \{1 \cdots n\})$
2. $\sum_i \alpha_i(0) N_i(0)$=$N(0)$
3. $\sum_i \alpha_i(\infty) N_i(\infty)$=$N(\infty)$

where $N(\cdot)$ is a virtual LSTM model satisfying the properties 2 and 3, and sharing the same structure as each single model N_i. The first property is the convex criteria need to be satisfied for α_i. The second property claims that the convex sum of the initial values for each LSTM model should be equal to that of the virtual model, and the third one indicate that the convergence of each single LSTM neural network

in the convex-based structure should be equal to that of the virtual model.

V. System Identification using LSTM

Recent years, LSTM has become a popular recurrent neural network (RNN) structure in the field of machine learning, and has been widely applied in many areas in industry [21]-[23]. Among these applications, most of them has inputs with long time lags, like: speech recognition or query classification in Natural Language Processing problems [24]-[25]. However, few literatures have ever been addressed on applying LSTM Neural Networks in system identification, though the network itself has been discussed extensively in the literature. In this section, a detail description on neural network identification of discrete dynamic system, and how to further extend the LSTM structure into the identification process will be discussed.

A. Single Input- Single Output (SISO) Discrete System Structure

In section II, a general form of discrete system has been discussed using the state-space representation. By limiting that only the inputs and outputs being accessible, the system representation can be further simplified as:

$$y(k) = f(y(k-1), ..., y(k-n); u(k-1), ...u(k-m)) \quad (9)$$

where $f(\cdot)$ is a nonlinear mapping: $\mathbb{R}^{m+n} \rightarrow \mathbb{R}$. Noticing that here the system structure is a SISO plant, which can be also extended to multi-variable case.

The system described in (2) is the most general case for nonlinear discrete system. There are also some of simpler forms widely accepted and applied in control applications. For instance:

$$y(k) = f_y(y(k-1), ..., y(k-n)] + f_u(u(k-1), ..., u(k-m)] \quad (10)$$

where the output $y(k)$ is assumed to be non-linearly related to its past and current input and output signals $u(k-i)$ where $i \in \{1, \cdots, m\}$ and $y(k-j)$ where $j \in \{1, \cdots, n\}$, which is particularly suited for control problems.

B. Identification using LSTM Neural Networks

The identification process includes building (an) appropriate identification model(s) to estimate the real system, which is defined by equation (9) and (10). The basic target is to minimize the identification error between the constructed LSTM based model and real plant model. According to the Universal Approximation Theorem introduced in section II, by properly choosing the size and parameters of neural network, any nonlinear function f can be identified or learnt by NN under relatively weak pre-conditions. In some of the early literatures [9], two major types of identification structures are used, parallel and series-parallel identification model, where the later one is always recommended for stability reason. In this paper, the second approach is also used.

Series-Parallel Identification Model:

The series-parallel model takes advantage of both the output signal $y(\cdot)$ from the real plant and $\hat{y}(\cdot)$ from the estimator. The model has the form:

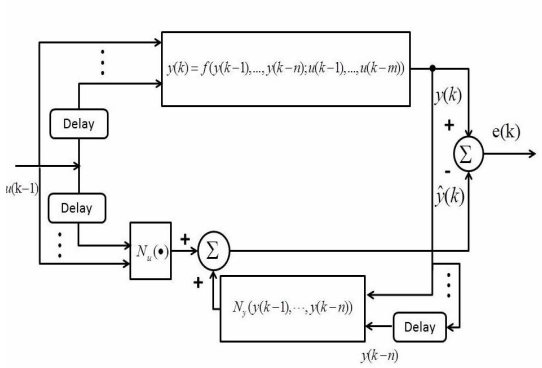

Fig. 3: Series-Parallel Identification Model

$$\begin{aligned} \hat{y}(k) = & N_y[y(k-1), \cdots, y(k-n)] \\ & + N_u[u(k-1), \cdots, u(k-m)] \end{aligned} \quad (11)$$

Noticing that on the right hand of the equation, $y(\cdot)$ is used to substitute $\hat{y}(\cdot)$ to ensure stability. The identification process, on the other hand, requires the accessibility of past plant system output, which is true for most of the time. Following is a graphical illustration for the series-parallel model (Figure (3)):

Comment 4: Noticing that the values of m and n are chosen before the identification process. n is the output memory indicating that how many past steps of output to be used in system identification. and m is generally called as the time-step in our LSTM structure, which is longest memory an LSTM can store. Simply speaking, the larger of the values m and n are chosen, the better identification result will be given by the design network system.

In this section, one discrete system representation and two corresponding identification structures are introduced. The identification structure, however, is taking the advantage of one LSTM neural network and its universal approximation property. Besides avoiding the commonly addressed vanishing gradient issue appeared in RNN based identification network, the speed of the identification (or on-line learning) process does not increase dramatically. (assuming the same gain factor is used in reducing the identification error $y(k) - \hat{y}(k)$ through the back propagation procedure). To overcome the slow convergence speed issue, a new design back-propagation scheme based on convex-based LSTM Neural Network is going to be discussed next section.

VI. A New Adaptive Learning Approach for Convex-based LSTM Neural Network

In section III, a convex-based LSTM structure is introduced. The corresponding adaptive learning approach for obtaining the convex coefficients α_i will be discussed in this section.

Figure (4) shows a detail structure of a convex-based LSTM neural network. The output of the identification model $y(k)$ is a convex sum of n LSTM models' outputs

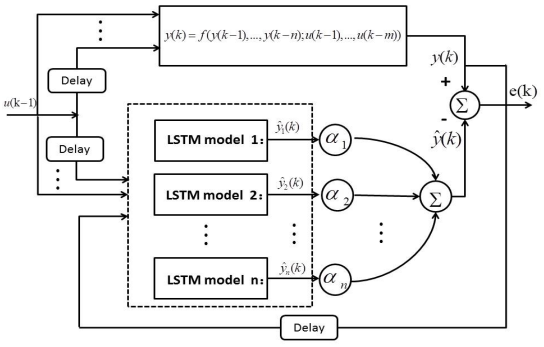

Fig. 4: Convex-based LSTM neural network for identification

$\hat{y}_1(k), \cdots, \hat{y}_n(k)$, as follows:

$$\hat{y}(k) = \alpha_1(k)\hat{y}_1(k) + \cdots + \alpha_n(k)\hat{y}_n(k) \tag{12}$$

Also defining the system output errors as:

$$e(k) = y(k) - \hat{y}(k) \tag{13}$$

Substituting (13) into (12), and combining with the convex property that α_n=1-$\sum_{i=1}^{n-1}\alpha_i$, a rearranged form of $e(k)$ can be obtained

$$\begin{aligned} e(k) &= y(k) - (\alpha_1(k)\hat{y}_1(k) + \cdots + \alpha_n(k)\hat{y}_n(k)) \\ &= \sum_{i=1}^{n}\alpha_i(k)y(k) - (\alpha_1(k)\hat{y}_1(k) + \cdots + \alpha_n(k)\hat{y}_n(k)) \\ &= \sum_{i=1}^{n-1}\alpha_i(k)e_i(k) + \alpha_n e_n(k) \\ &= \sum_{i=1}^{n-1}\alpha_i(k)e_i(k) + (1 - \sum_{i=1}^{n-1}\alpha_i(k))e_n(k) \\ &= \sum_{i=1}^{n-1}\alpha_i(k)\tilde{e}_i(k) + e_n(k) \end{aligned} \tag{14}$$

where $e_i(k)$ is defined as the error between the i^{th} LSTM model and the plant output, i.e. $y(k) - \hat{y}_i(k)$, and $\tilde{e}_i(k)$ is the difference between $e_i(k)$ and $e_n(k)$, i.e. $\tilde{e}_i(k) \triangleq e_i(k) - e_n(k)$, which is the error differences between the i^{th} and n^{th} node.

The error equation obtained from (14) can be further simplified as:

$$\tilde{e}(k) = \tilde{\mathbf{E}}^{\mathbf{T}}(\mathbf{k})\tilde{\alpha}(\mathbf{k}) \tag{15}$$

where $\tilde{e}$ is a scalar value, which is equal to $e - e_n$, and $\tilde{\mathbf{E}} \in \mathbb{R}^{1\times(n-1)}$ is a vector defined as $[\tilde{e}_1, \cdots, \tilde{e}_{n-1}]$, as well as $\tilde{\alpha} = [\alpha_1, \cdots, \alpha_{n-1}] \in \mathbb{R}^{1\times(n-1)}$.

The update rule for $\tilde{\alpha}$ can also be derived by multiplying $\tilde{\mathbf{E}}$ on both sides of the equation, and move the left hand side of the equation to the right:

$$\begin{aligned} \tilde{\alpha}(\mathbf{k}) - \tilde{\alpha}(\mathbf{k-1}) &= -\tilde{\mathbf{E}}\tilde{\mathbf{E}}^{\mathbf{T}}\tilde{\alpha}(\mathbf{k-1}) + \tilde{\mathbf{E}}\tilde{e}(k-1) \\ \tilde{\alpha}(\mathbf{k}) &= \tilde{\alpha}(\mathbf{k-1}) - \tilde{\mathbf{E}}\tilde{\mathbf{E}}^{\mathbf{T}}\tilde{\alpha}(\mathbf{k-1}) + \tilde{\mathbf{E}}\tilde{e}(k-1) \end{aligned} \tag{16}$$

Hence, equation (16) has become the new back propagation law for updating the convex parameter $\tilde{\alpha}$(k), which will give us the first $n-1$ elements in the convex coefficient vectors $\alpha = [\alpha_1, \cdots, \alpha_n]$. By the convex property of α, the last element α_n can be obtained by $1 - \sum_1^{n-1}\alpha_i$ once $\tilde{\alpha}$ is obtained.

Also, for each single LSTM model, its weights will be updated by the standard back-propagation law using the model error e_i, which has been illustrated in detail in section III B. It should be noticed that the update procedure of $\tilde{\alpha}$ and networks' weights are both on-line and in a simultaneous manner.

A. Performance Analysis of Convex Coefficients α:

In the convex-based LSTM Neural Networks structure, a new convex coefficient vector α is introduced. The question involved is that how this new parameter can change the performance of LSTM networks. Two properties will be claimed here as below:
1) The error between α and its true value α^* is decreasing exponentially with respect to the iteration round number k.
2) The identification system will converge when α converges, regardless of whether the standard back-propagation process converges or not. i.e. $\sum_1^n \alpha_i N_i$=$y(k)$, once α_i=α_i^* for all $i \in \{1 \cdots n\}$.

The first property can be easily obtained from equation (16). As it is shown, the change difference of α is proportional to its last iteration's value. Hence, an exponential property will be given by solving the difference equation. The interesting property is the second one, which indicates that the convergence speed of α dominates the identification speed , as that of conventional back-propagation part for the LSTM network is far inferior than the exponential convergence. The error e is defined as the convex combination of each single model's error:

$$\begin{aligned} e &= \sum_{i=1}^{n}\alpha_i N_i - y \\ &= \sum_{i=1}^{n}\alpha_i(N_i - y) \\ &= \sum_{i=1}^{n}\alpha_i e_i \end{aligned} \tag{17}$$

where the convex property of $\sum_{i=1}^n \alpha_i = 1$ is used. Equation (17) indicates that only the convex sum of error e_i, i.e. $\sum_{i=1}^n \alpha_i e_i$, needs to be zero for identification, instead of single model error $e_i \to 0$. It gives the theoretical foundation that why the convex-based approach is much faster than conventional LSTM Neural Networks.

Comment 5: The convex property of $\sum_{i=1}^n \alpha_i = 1$ gives us a possibility to design the adaptive update law as shown in equation (16), by ensuring the robustness of the system and speed of convergence at the same time. Also, noticing that α_i is within a range $[0, 1]$, which gives us a relatively short range for parameter to update. this is another potential reason that why it will converge much faster than the network itself

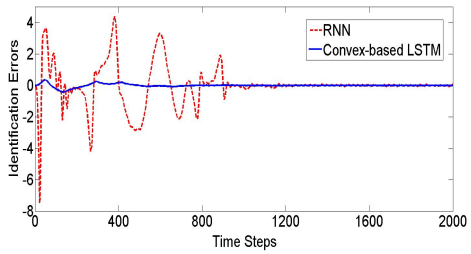

Fig. 5: Identification error for Simulation 1

using backpropagation. The choice of n, which is the number of models, depends on system complexity, and normally with larger value when the nonlinear system become more complex.

VII. Simulation Study

In this section, a simulation conducted for identifying nonlinear dynamic systems. A comparison between the results from convex-based LSTM method and conventional RNN approach is also given for each system.

A. Simulation 1:

Here we consider a system structure in a form:

$$y(k) = f_y(y(k-1), y(k-2)) + \sum_{i=1}^{2} u(k-i) \qquad (18)$$

The system equation is

$$\begin{aligned} y(k) &= 0.7y(k-1) - 0.8y(k-1)e^{y(k-1)} \\ &\quad - 0.6y(k-2) - 0.5y(k-2)e^{y^2(k-1)} \\ &\quad + u(k-1) + 0.3u(k-2) \end{aligned} \qquad (19)$$

The identification error $y(k) - \hat{y(k)}$ is plotted for both convex-based LSTM approach and Conventional RNN approach in Figure (5). The input is used as $u(k) = sin(2\pi k/125) + cos(2\pi k/50)$

From the simulation result, it is easy to figure out that the identification error obtained by convex-based LSTM neural network converges much faster and smoother than the conventional RNN approach.

VIII. Conclusion

In this paper, a new concept using LSTM neural networks for dynamic systems identification has been proposed. By taking the LSTM advantage over vanishing gradient issue, together with the convex multiple models for increasing the speed, the designed structure has shown far superior performance compared with conventional RNN and LSTM, as shown in section VIII. Theoretical explanations why the convex-based approach gives a faster convergence speed than the other RNN-based neural network methods are also given in section VI. A brief controller structure is shown graphically in chapter VII. From the theories and simulations discussed in this paper, it is confident to conclude that the newly proposed LSTM based identification scheme is well suited to identify the discrete dynamic systems, especially when there is a requirement for a high speed and accuracy during identification procedure.

References

[1] Hochreiter, Sepp, and Jrgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.
[2] Chung, Junyoung, et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling." arXiv preprint arXiv:1412.3555 (2014).
[3] Rao, Kanishka, et al. "Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks." 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2015.
[4] Vinyals, Oriol, et al. "Grammar as a foreign language." Advances in Neural Information Processing Systems. 2015.
[5] Sak, Haim, Andrew Senior, and Franoise Beaufays. "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition." arXiv preprint arXiv:1402.1128 (2014).
[6] Hebb, D. O. "The organization of behavior; a neuropsychological theory." (1949).
[7] Rosenblatt, Frank. "The perceptron: a probabilistic model for information storage and organization in the brain." Psychological review 65.6 (1958): 386.
[8] Werbos, Paul. "Beyond regression: New tools for prediction and analysis in the behavioral sciences." (1974).
[9] Narendra, Kumpati S., and Kannan Parthasarathy. "Identification and control of dynamic systems using neural networks." IEEE Transactions on neural networks 1.1 (1990): 4-27.
[10] Hochreiter, Sepp. "Untersuchungen zu dynamischen neuronalen Netzen." Diploma, Technische Universitt Mnchen (1991): 91.
[11] Wang, Yu, and Xiaoxi Zhu. "A Supervised Adaptive Learning-based Fuzzy Controller for a non-linear vehicle system using Neural Network Identification." American Control Conference (ACC), Boston, 2016.
[12] Wang, Yu. "Design of Triple-Level Multiple Models Fuzzy Logic Controller for Adaptive Speed Control with Unknown External Disturbances." IFAC Proceedings Volumes 47.3 (2014): 6326-6331.
[13] Han, Zhuo, and Kumpati S. Narendra. "New concepts in adaptive control using multiple models." IEEE Transactions on Automatic Control 57.1 (2012): 78-89.
[14] Narendra, Kumpati S., Yu Wang, and Wei Chen. "The Rationale for Second Level Adaptation." arXiv preprint arXiv:1510.04989 (2015).
[15] Narendra, Kumpati S., Yu Wang, and Wei Chen. "Stability, robustness, and performance issues in second level adaptation." 2014 American Control Conference. IEEE, 2014.
[16] Narendra, Kumpati S., Yu Wang, and Wei Chen. "Extension of second level adaptation using multiple models to SISO systems." American Control Conference (ACC), 2015. IEEE, 2015.
[17] Narendra, Kumpati S., Yu Wang, and Snehasis Mukhopadhay. "Fast Reinforcement Learning using multiple models." Control and Decision Conference(CDC), Las Vegas, 2016.
[18] Narendra, Kumpati S., and Yu Wang. "Simulation Studies of Feedforward Learning Schemes with Multiple Models." Technical Report 1603 (2016).
[19] Narendra, Kumpati S., Snehasis Mukhopadyhay, and Yu Wang. "Improving the Speed of Response of Learning Algorithms Using Multiple Models." arXiv preprint arXiv:1510.05034 (2015).
[20] Narendra, Kumpati S., Yu Wang and Snehasis Mukhopadyhay, . "Multiple Estimation Models for Faster Reinforcement Learning." Technical Report 1604 (2016).
[21] Zen, Heiga. "Statistical parametric speech synthesis: from HMM to LSTM-RNN." (2015).
[22] Breuel, Thomas M., et al. "High-performance OCR for printed English and Fraktur using LSTM networks." 2013 12th International Conference on Document Analysis and Recognition. IEEE, 2013.
[23] Dai, Andrew M., and Quoc V. Le. "Semi-supervised sequence learning." Advances in Neural Information Processing Systems. 2015.
[24] Levada, Alexandre LM, et al. "Novel approaches for face recognition: template-matching using dynamic time warping and LSTM Neural Network Supervised Classification." 2008 15th International Conference on Systems, Signals and Image Processing. IEEE, 2008.
[25] Chiu, Jason PC, and Eric Nichols. "Named entity recognition with bidirectional lstm-cnns." arXiv preprint arXiv:1511.08308 (2015).