Design of "Deep Learning Controller"

Koksal Erenturk

Abstract— **Deep learning allows computational models of multiple processing layers to learn and represent data with multiple levels of abstraction mimicking how the brain perceives and understands multimodal information, thus implicitly capturing intricate structures of large- scale data. In the meantime, recent advances in deep learning, encompassing neural networks, hierarchical probabilistic models, and a variety of unsupervised and supervised feature learning algorithms, have brought about tremendous development to many areas of interest to the engineering community.**

In this work, an extended type of the current accomplishment of deep learning to chemical process control problem has been presented. As well-known, if one formulated the reward function properly, "deep learning" can be used for industrial process control purpose. The controller setup follows the typical reinforcement learning setup, whereby an agent (controller) interacts with an environment (process) through control actions and receives a reward in discrete time steps. Deep neural networks (DNN) serve as function approximators and are used to learn the control policies. Once the DNN trained, control actions can be achieved at the output of the learned network. Even though the policies are not explicitly specified for the DNN, the DNN has an ability to learn policies that are different from the traditional controllers. The designed "Deep Learning Controller" (DLC) for Single Input Single Output Systems (SISO) has been tested under various scenarios. Obtained results have been given in graphical illustrations for details and these results showed that DLC can be easily used for instead of any type of controller. Additionally, it can be concluded that DLC are very robust when compared with the other type of controllers in terms of noise and unknown disturbances.

Index Terms— **Deep learning, controller design, control engineering, big data.**

I. INTRODUCTION

Artificial neural networks trained by back-propagating error derivatives have the potential to learn much better models of data that lie on or near a nonlinear manifold. Over the last few years, advances in both machine learning algorithms and computer hardware have led to more efficient methods for training deep neural networks (DNNs) that contain many layers of non-linear hidden units and a very large output layer known as the deep learning algorithms [1].

Recently deep learning has been attracting a significant attention from the wide range of applications. Compare to the conventional neural networks, the key features of deep learning are to have more hidden layers and neurons, and to improve learning performance. Using these features, large and complex problems that could not be solved with conventional neural networks can be resolved by deep learning algorithms. Consequently, deep learning has been applied to various applications including pattern recognition and classification problems; for example, speech recognition [1-3], handwritten digit recognition [4], human action recognition [5], and so on. However, based on the literature searches of the authors, there are very limited papers in the automatic control field that have been published. For this reason, this paper aimed to show the utilizing possibility of deep learning in chemical control areas.

This study was designed to mimic the PID controller using a DNN algorithm. The simulation studies have been performed in MATLAB environments and the detailed comparison for the proposed deep learning controller was given to demonstrate the performance and effectiveness of the proposed algorithm.

This paper is organized as follows. The deep learning is described in Section 2. In Section 3, the design of deep learning controller is explained. The comparison details between the proposed deep learning controller and a classical PID controller are presented with the simulation results are shown in Section 4. Finally, a conclusion remarks are given in Section 5.

II. DEEP LEARNING

In this chapter, we will give some essential details for deep learning algorithms. Deep learning has many layers of hidden units and it also allows many more parameters to be used before over-fitting occurs. The generative pre-training creates many layers of feature detectors that become progressively more complex [6]. A subsequent phase of discriminative fine-tuning, using the standard backpropagation algorithm, then slightly adjusts the features in every layer to make them more useful for discrimination [6]. Thus, for deep learning, a deep architecture is used.

Deep learning is extended algorithm from conventional neural networks, where the number of hidden layers and the number of neurons are more than those of conventional neural networks [1]. In control system, conventional neural networks are well documented and used as a tool for controller design [7], system identification [8], auto-tuning [9], and compensator [10]. In contrary, the deep learning is not used yet, although it is more effective algorithm than conventional neural network, especially in big data. Further, the deep learning algorithm uses a deep architecture. There are several types of deep architectures, among them; the well-known architecture is a Deep Belief Network (DBN) or breifly DNN.

III. DESIGN OF DEEP LEARNING CONTROLLER [3]

Reinforcement learning (RL) is an approach to automating goal-directed learning and decision-making. This approach is meant to solve problems in which an agent interacts with an environment and receives reward signal at each time step. RL algorithms aim to find a policy, which is a mapping from state to action, that maximizes the expected cumulative reward (value function) under that policy. The two main

Manuscript received October 09, 2018

Koksal Erenturk, Department of Computer Engineering, Ataturk University/ College of Engineering, Erzurum, Turkey, +90-542-255 1068, (erenturk@yahoo.com).

approaches used to achieve this goal are (1) Policy based approach: searches directly for the optimal policy which achieves maximum future reward; (2) Value based approach: estimates the optimal value function which is the maximum value achievable under any policy. Storing the value function (or) policy might not be possible especially if the state-action pairs are high dimensional. Hence, function approximators like linear regression, Neural networks are used with RL. Since it has been found that deep neural networks serve as effective functional approximators and have found great success in image, speech and language understanding, deep neural networks are used to approximate value function or policy resulting in Deep Reinforcement Learning (DeepRL). DeepRL aims at learning the policy and/or value function end-to-end with or without the plant model. However, using neural networks as such might not necessarily gaurantee convergence and stability. Recent success of Deep Q nework along with improvements in the field of deep learning like Batch Normalization have addressed the issue of using neural networks in reinforcement learning. With those advancements, this work aims at using DeepRL for process control applications. Process control applications involve continuous states and action space. Policy gradient algorithms are widely used RL techniques to address continuous control problems [11]. We have used actor-critic based deterministic policy gradient similar to [11]. The actor learns the policy with the help of critic which in turn learns the value function. At each time step as the agent interacts with the environment both critic and actor are updated and the actor policy continues to improve with respect to the number of iterations. The learnt actor is then used for control.

Formally, RL can be described as a Markov decision process (MDP), which consists of

- a set of states $\mathcal{S}$, plus a distribution of starting states $p(\mathbf{s}_0)$
- a set of actions $\mathcal{A}$
- transition dynamics $\mathcal{T}(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$ that map a state-action pair at time t onto a distribution of states at time $t + 1$
- an immediate/instantaneous reward function $\mathcal{R}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1})$
- a discount factor $\gamma \in [0,1]$, where lower values place more emphasis on immediate rewards.

In general, the policy π is a mapping from states to a probability distribution over actions $\pi: \mathcal{S} \rightarrow p(\mathcal{A}=\mathbf{a}|\mathbf{s})$. If the MDP is episodic, i.e., the state is reset after each episode of length $\mathcal{T}$, then the sequence of states, actions, and rewards in an episode constitutes a trajectory or rollout of the policy. Every rollout of a policy accumulates rewards from the environment, resulting in the return $R = \sum_{t=0}^{T-1} \gamma^t r_{t+1}$. The goal of RL is to find an optimal policy, π^* that achieves the maximum expected return from all states:

$$\pi^* = \mathrm{argmax}_\pi \, \mathbb{E}[R|\pi] \tag{1}$$

The reward function is provided as given below:

$$r(s,a,s') = \begin{cases} c, & \text{if } |y^i - y^i_{set}| \le \varepsilon, \forall i \\ -\sum_i^n |y^i - y^i_{set}|, & \text{otherwise} \end{cases} \tag{2}$$

The goal of learning is to find a control policy, π that maximizes the expected value of the cumulative reward, R. Here, R can be expressed as the time-discounted sum of all transition rewards, r_i, from the current action up to a horizon T (where T may be infinite) i.e.,

$$R(s_0) = r_0 + \gamma r_1 + \cdots + \gamma^T r_T \tag{3}$$

The discount factor ensures that the cumulative reward is bounded, and captures the fact that events occurring in the distant future are likely to be of less consequential than those occurring in the more immediate future. The goal of RL is to maximize the expected cumulative reward.

The algorithm for learning the control policy is given in Algorithm 1.

Algorithm 1 Learning Algorithm

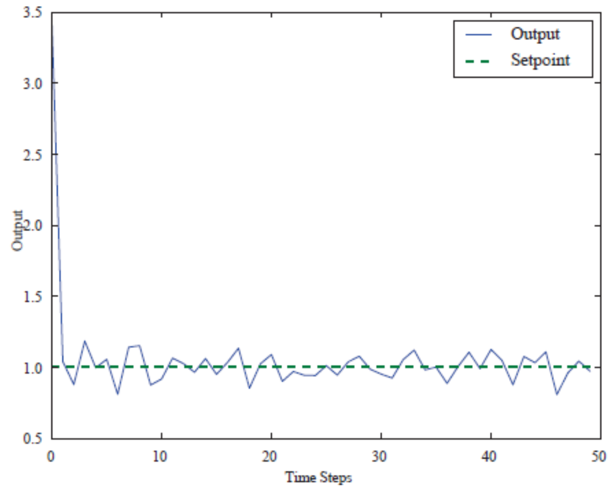
$W_a, W_c \leftarrow$ initialize random weights
initialize Replay memory, RM with random policies
for episode = 1 to E **do**
  Reset the OU process noise $\mathcal{N}$
  Specify setpoint, y_{set} at random
  for step = 1 to T **do**
    $s \leftarrow < y_t, y_{set} >$
    $a \leftarrow$ action, $u_t = \pi(s, W_a) + \mathcal{N}_t$
    Execute action u_t on the plant
    $s' \leftarrow < y_{t+1}, y_{set} >$, observe state at next instant
    $r \leftarrow$ reward
    Store the tuple $< s, a, s', r >$ in RM
    Sample a minibatch of n tuples from RM
    Compute $y^i = r^i + \gamma Q^i_t \;\; \forall i \in$ minibatch
    Update Critic:
    $W_c \leftarrow W_c + \alpha(\frac{1}{n}\sum_i^n (y^i - Q(s^i, a^i, W_c) \frac{\partial Q(s^i,a^i,W_c)}{\partial W_c})$
    Compute $\nabla^i_p = \frac{\partial Q(s^i,a^i,W_c)}{\partial a^i}$
    Invert ∇^i_p by (5)
    Update Actor:
    $W_a \leftarrow W_a + \alpha \frac{1}{n} \sum_1^n (\nabla^i_p \frac{\partial \pi(s^i, W_a)}{\partial W_a})$
    Update Target Critic:
    $W^t_a \leftarrow \tau W_a + (1-\tau) W^t_a$
    Update Target Actor:
    $W^t_c \leftarrow \tau W_c + (1-\tau) W^t_c$
  end for
end for

IV. Simulation Results

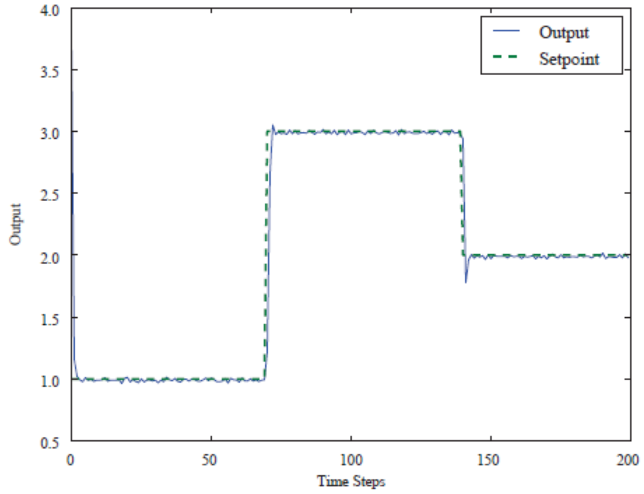
A 1×1 process is used to study the effect of this approach of learning the policy. The industrial system we choose to study is the control of manufacture of paper in a paper machine. The target output, y_{set}, is the desired moisture content of the paper sheet. The control action, u, is the steam flow rate, and the system output, y, is the current moisture content. The transfer function of the system is given below:

$$G(s) = \frac{0.0285s}{1-0.936s} \tag{4}$$

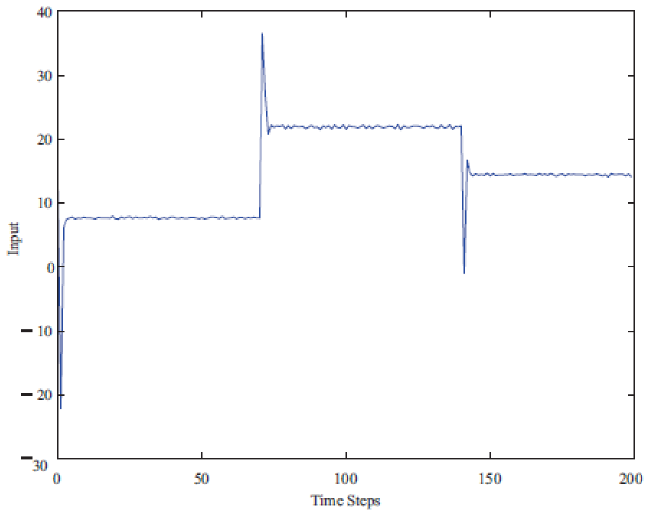
The learned controller is also tested on various other cases, such as the response to setpoint change and the effect of output and input noise. The results [11] are shown in Fig. 1.



(a) Output of the system with output noise



(b) Output of the system versus setpoint changes



(c) Input response for output noise

Figure 1. Output and input responses for the system

V. CONCLUSION

In this paper, a deep learning controller based on DBN algorithm was designed to explore the ability of applying the deep learning algorithm to the control problems. A comparison study for the proposed deep learning controller was performed to verify the feasibility of the use of deep learning in control theory. The simulation results demonstrate the effectiveness of the proposed deep learning controller to be used as a control tool.

REFERENCES

[1] K Cheon, J Kim, M Hamadache, D Lee, "On Replacing PID Controller with Deep Learning Controller for DC Motor System," Journal of Automation and Control Engineering Vol. 3, No. 6, December 2015.

[2] G Hinton, L Deng, D Yu, GE Dahl, "Deep Neural Networks for Acoustic Modeling in Speech Recognition," IEEE Signal Processing Magazine, Nov 2012.

[3] Yao, Kaisheng, et al. "Recurrent neural networks for language understanding." INTERSPEECH. 2013.

[4] I Arel, DC Rose, TP Karnowski, "Deep Machine Learning—A New Frontier in Artificial Intelligence Research," IEEE Computational Intelligence Magazine, Nov 2010.

[5] S Ji, W Xu, M Yang, K Yu, "3D Convolutional Neural Networks for Human Action Recognition," IEEE Transon Pattern Analysis and Machine Intelligence, vol. 35, no. 1, Jan 2013.

[6] A Mohamed, G E Dahl, and G Hinton, "Acoustic Modeling using Deep Belief Networks," IEEE Transactions on Audio, Speech & Language Processing, vol. 20 no. 1, pp. 14-22, 2012.

[7] M Lee, HS Choi, "A Robust Neural Controller for Underwater Robot Manipulators", IEEE Transactions on Neural Networks, Nov, 2000.

[8] V Prasad, BW Bequette, "Nonlinear System Identification and Model Reduction Using Artificial Neural Networks," Computers & Chemical Engineering, December, 2003.

[9] J Chen, TC Huang, "Applying Neural Networks to On-line Updated PID Controllers for Nonlinear Process Control," Journal of Process Control, 2004.

[10] B Xu, SR Pandian, N Sakagami, F Petry, "Neuro-fuzzy control of underwater vehicle-manipulator systems," Journal of the Franklin Institute, vol. 349, issue. 3, pp. 1125-1138, April 2012.

[11] S.P.K. Spielberg, R.B. Gopaluni, P.D. Loewen, "Deep Reinforcement Learning Approaches for Process Control," 2017 6th International Symposium on Advanced Control of Industrial Processes (AdCONIP) May 28-31, 2017. Taipei, Taiwan.

Koksal ERENTURK was born in Erzurum, Turkey, in 1973. He received his B.S. degree (with honors) from Yildiz Technical University, Istanbul, Turkey, in 1994. He also received his M.S. degree from Istanbul University, Istanbul, Turkey, in 1997, and Ph.D. degree from Karadeniz Technical University (Trabzon, Turkey), in 2002, all in electrical engineering. Currently, he is a Full Professor in the Department of Electrical and Electronics Engineering at Ataturk University, Erzurum, Turkey. His works have focused on the development and application of control theory to a variety of mechatronic systems, with a focus on observation and estimation based control. Much of his recent works have focused on the analysis and control of dynamical systems that arise in engineering applications. His areas of interest also include smart grids, fuzzy logic and fuzzy control, neural networks, genetic algorithms, and their applications to dynamic systems and electrical engineering.