Journal of The Gujarat Research Society

ISSN: 0374-8588
Volume 22 Issue 1, January 2020

# A Review of Recent Research On Deep Learning in Robotics

Vijay Maheshwari, Mr. Kuldeep Chauhan
Shobhit Institute of Engineering and Technology (Deemed to be University), Meerut
Email Id- vijay@shobhituniversity.ac.in, kuldeep.chauhan@shobhituniversity.ac.in

***ABSTRACT: The science of training large artificial neural networks is known as deep learning. DNNs can have hundreds of millions of parameters, allowing them to model complex functions like nonlinear dynamics. They create compact state representations from raw, high-dimensional, multimodal sensor data found in robotic systems, and unlike many machine learning methods, they don't require a human expert to hand-engineer feature vectors from sensor data at design time. Deep learning advances have sparked a flurry of research in the application of deep artificial neural networks to robotic systems over the last decade, with at least 30 papers published on the topic between 2014 and now. Using current research as examples, this review discusses the applications, benefits, and limitations of deep learning in relation to physical robotic systems. Its goal is to inform the broader robotics community about recent advances and to pique interest in and application of deep learning in robotics.***

***KEYWORDS: Artificial Intelligence, Deep Learning, Neural Networks, Robotics, Simulation.***

## INTRODUCTION

However, in physical robotic systems, where generating training data is generally expensive and suboptimal training performance poses a danger in some applications, DNNs can present unique challenges. Despite these obstacles, roboticists are developing innovative solutions, such as using digital manipulation to leverage training data, automating training, and combining multiple DNNs to improve performance and reduce training time[1][2]. Deep learning for robotics is a hot topic in academia, with at least 30 papers published on the topic between 2014 and the time of this writing. This review summarizes the findings of recent research, with a focus on the benefits and challenges of robotics. Following a primer on deep learning, a discussion of how common DNN structures are used in robotics and examples from recent literature is presented. There are also practical considerations for roboticists who want to use DNNs. Finally, future trends and their limitations, as well as strategies to mitigate them, are discussed. The basic principles of linear regression, and many of those same principles still apply to what deep learning researchers study. Several significant advancements, however, have gradually transformed regression into what we now refer to as deep learning[3].

The addition of an activation function allowed regression methods to fit nonlinear functions for the first time. It also brought up some biological parallels with brain cells. Nonlinear models were then stacked in "layers" to create multi-layer perceptron's, which are powerful models. In the 1960s, a group of researchers independently figured out how to differentiate multi-layer perceptron's, and by the 1980s, back propagation had become a popular method for training them. Multi-layer perceptron's were quickly proven to be universal function approximates which meant they could fit any data, no matter how complex, with arbitrary precision using a finite number of regression units[4]. Backpropagation was the beginning of the deep learning revolution in many ways, but due to the problem of vanishing gradients, researchers still mostly limited their neural networks to a few layers. Deeper neural networks took an infinite amount of time to train. As early

Journal of The Gujarat Research Society
ISSN: 0374-8588
Volume 22 Issue 1, January 2020

as the 1980s, neural networks were successfully used to control robotics. Nonlinear regression was quickly recognized as having the functionality required for operating dynamical systems in continuous spaces, and fuzzy systems, which are closely related, appeared to be well suited for nominal logical control decisions[5]. Famously demonstrated that neural networks were effective for assisting vehicles in staying in their lanes as early as 1989. Neural networks, on the other hand, were still too slow to process entire images or perform the complex tasks required by many robotics applications. There are numerous robotics applications. Researchers began using graphical processing units (GPUs) to parallelize artificial neural network implementations. A matrix-vector multiplication step, which can be parallelized using GPUs, is the most significant bottleneck in training neural networks[6]. Hinton presented a training method for a multi-layered neural network, which he demonstrated to be effective.

The near-simultaneous appearance of these technologies sparked a surge in research interest, propelling deep learning forward at an unprecedented rate. As hardware improved and neural networks became more practical, they were discovered to be increasingly effective in real-world robotics applications. RNNPB demonstrated in 2004 that neural networks could self-organize high-level control schema that generalized well with a variety of robotics test problems. In 2008, neuroscientists made advances in recognizing how animals achieved locomotion, and were able to extend this knowledge all the way to neural networks for experimental control of robots. In 2011, TNLDR demonstrated that deep neural nets could effectively model both state and dynamics from strictly unsupervised training with raw images of a simulated robot. Another relevant work is surveying applications for neural networks in perception for robot guidance[7]. In hindsight, we see that chess was considered in the early years of artificial intelligence to be representative of human intelligence over machines. After machines beat world-class chess players, a new emblematic task was needed to represent the superior capabilities of human intelligence. Visual recognition was largely accepted to be something easy for humans but difficult for machines.

But now, with the emergence of deep learning, humans will not be able to claim that as an advantage for much longer. Deep learning has surged ahead of well-established image recognition techniques and has begun to dominate the benchmarks in handwriting recognition, video recognition, small-image identification, detection in biomedical imagine, and many others. It has even achieved super-human accuracy in several image recognition contests. Perhaps agility or dexterity will be a forthcoming achievement where machines will begin to demonstrate human-like proficiency. If so, it appears that DNNs may be the learning model that enables it. The idea of using machine learning in controlling robots requires humans to be willing to relinquish a degree of control[8]. This can seem counterintuitive at first, but the benefit for doing so is that the system can then begin to learn on its own. This makes the system capable of adapting and therefore has potential to ultimately make better use of the direction that comes from humans. DNNs are well suited for use with robots because they are flexible and can be used in structures that other machine learning models cannot support. Four common structures for using DNNs with robots. Structure. It is one common model for facilitating ‘unsupervised learning. It requires two DNNs, called an encoder’ and a ‘decoder. In this configuration, only x needs to be supplied by the user. s is a ‘latent’ or internal encoding that the DNN generates. For example, x might represent images observed by a robot’s camera, containing thousands or even millions of values. The encoder might use convolutional layers, which are known to be effective for digesting images. By learning to reduce x to s, the auto encoder essentially creates its own internal encoding of ‘state.’ It will not

ISSN: 0374-8588
Volume 22 Issue 1, January 2020

necessarily use an encoding that has meaning for humans, but it will be sufficient for the DNN to approximately reconstruct x. How are auto encoders useful in robotics? Sometimes, the robot designer may not know exactly what values are needed by the robot. Auto encoders enable the system to figure that out autonomously.

This becomes especially useful when a hybrid of supervised and unsupervised learning is used. For example, the user can impose certain values in perhaps, positional coordinates or joint angles and the DNNs will learn to work with those values, using the other free elements in s for its own encoding purposes. Auto encoders may also be used to initialize some parts of Structure. Generative models are closely related to auto encoders. They utilize just the decoder portion of the model to predict observations from an internal representation of state. Structure C is a type of 'recurrent neural network,' which is designed to model dynamic systems, including robots. It is often trained with an approach called 'backpropagation through time. Many advances, such as 'long short term memory units,' have made recurrent neural networks much stronger. In this configuration, u represents a control signal. u may also contain recent observations. An internal representation of future state, and x is a vector of anticipated future observations. The transition function approximates how the control signal will affect state over time. Just as with auto encoders, the representation of state can be entirely latent, or partially imposed by the user. If it were entirely imposed, the model would be prevented from learning.) If x includes an estimate of the utility, then this configuration is used in 'model- based reinforcement learning'. Each of the various types of deep learning models are made by stacking multiple layers of regression models[9].

Within these models, different types of layers have evolved for various purposes. One type of layer that warrants particular mention is convolutional layers. Unlike traditional fully connected layers, convolutional layers use the same weights to operate all across the input space. This significantly reduces the total number of weights in the neural network, which is especially important with images that typically have hundreds of thousands to millions of pixels that must be processed. Processing such images with fully connected layers would require more than (100 K)2 to (1 M)2 weights connecting each layer, which would be completely impractical. Convolutional layers were inspired by cortical neurons in the visual cortex, which respond only to stimuli with a receptive field. Ultimately, the underlying philosophy that prevails in the deep learning community is that every part of a complex system can be made to 'learn.' Thus, the real power of deep learning does not come from using just one of the structures described in the previous section as a component in a robotics system, but in connecting parts of all of these structures together to form a full system that learns throughout. This is where the 'deep' in deep learning begins to make its impact: When each part of a system is capable of learning, the system as a whole can adapt in sophisticated ways[10].

## DISCUSSION

Neuroscientists are even starting to recognize that many of the patterns evolving within the deep learning community and throughout artificial intelligence are starting to mirror some of those that have previously evolved in the brain. Doya identified that supervised learning methods (Structures A and C) mirror the function of the cerebellum, unsupervised methods Structure B learn in a manner comparable to that of the cerebral cortex, and reinforcement learning is analogous with the basal ganglia. Thus, the current trajectory of advancement strongly suggests that control of robots is leading toward full cognitive architectures that divide coordination tasks in a manner

Journal of The Gujarat Research Society
ISSN: 0374-8588
Volume 22 Issue 1, January 2020

increasingly analogous with the brain. The robotics community has identified numerous goals for robotics in the next 5 to 20 years. These include, but certainly are not limited to, human-like walking and running, teaching by demonstration, mobile navigation in pedestrian environments, collaborative automation, automated bin/shelf picking, automated combat recovery, automated aircraft inspection and maintenance, and robotic disaster mitigation and recovery. This paper identifies seven general challenges for robotics that are critical for reaching these goals and for which DNN technology has high potential for impact. learning complex, high-dimensional, and novel dynamics Analytic derivation of complex dynamics requires human experts, is time consuming, and poses a trade-off between state dimensionality and tractability. Making such models robust to uncertainty is difficult, and full state information is often unknown. Systems that can quickly and autonomously adapt to novel dynamics are needed to solve problems such as grasping new objects, traveling over surfaces with unknown or uncertain properties, managing interactions between a new tool and/or environment, or adapting to degradation and/or failure of robot subsystems. Also needed are methods to accomplish this for systems that possess hundreds (or even thousands) of degrees of freedom, exhibit high levels of uncertainty, and for which only partial state information is available. As with dynamics, control systems that accommodate high degrees of freedom for applications such as multi-arm mobile manipulators, anthropomorphic hands, and swarm robotics are needed. Such systems will be called upon to function reliably and safely in environments with high uncertainty and limited state information. Despite advances achieved over three decades of active research, robust and general solutions for tasks such as grasping deformable and/or complex geometries, using tools, and actuating systems in the environment (turn a valve handle, open a door, and so forth) remain elusive – especially in novel situations. This challenge includes the kinematics, kinetics, and grasp planning inherent in tasks such as these. advanced object recognition DNNs have already proven to be highly adept at recognizing and classifying objects.

Advanced application examples include recognizing deformable objects and estimating their state and pose for grasping, semantic task and path specification e.g. go around the table, to the car, and open the trunk, and recognizing the properties of objects and surfaces such as wet/slippery floors or sharp objects that could pose a danger to human collaborators. interpreting and anticipating human actions. This challenge is critical if robots are to work with or among people in applications such as collaborative robotics for manufacturing, eldercare, autonomous vehicles operating on public thoroughfares, or navigating pedestrian environments. It will enable teaching by demonstration, which will in turn facilitate task specification by individuals without expertise in robotics or programming. This challenge may also be extended to perceiving human needs and anticipating when robotic intervention is appropriate. The proliferation of low-cost sensing technologies has been a boon for robotics, providing a plethora of potentially rich, high-dimensional, and multimodal data. This challenge refers to methods for constructing meaningful and useful representations of state from such data. Robots will need to reliably execute high-level commands that fuse the previous six challenges to achieve a new level of utility, especially if they are to benefit the general public. For example, the command 'get the milk' must autonomously generate the lower-level tasks of navigating to/from the refrigerator, opening/closing the door, identifying the proper container milk containers may take many forms, and securely grasping the container. Recent robotics research that utilizes DNN technology according to these challenges, as well as the DNN structures discussed in the previous section. From this, several observations are made: First is that Structure is clearly the most popular DNN architecture in the recent robotics literature. This is likely explained by its intuitive nature, essentially learning to approximate the same function presented to it in the form of training samples. It also requires the least amount of domain knowledge in DNNs to implement. Robotics challenges, however, are not limited to the sort of classification and or regression problems to which this structure is best suited. Additional focus on applying Structures B, C, and D to robotics problems may very well catalyze significant advancement in many of the identified challenges. One of the purposes of this paper is to emphasize the potential of the other structures to the robotics community. Somewhat related is the fact that some cells in are empty. In the authors' opinion, this is due to a lack of research focus rather than any inherent incompatibilities between challenges and structures. In particular, the ability of Structure to learn compact representations of state would be particularly useful for estimating the pose, state, and properties of objects. Structure A involves using a deep learning model to approximate a function from sample input–output pairs.

This may be the most general-purpose deep learning structure, since there are many different functions in robotics that researchers and practitioners may want to approximate from sample observations. Some examples include mapping from actions to corresponding changes in state, mapping from changes in state to the actions that would cause it, or mapping from forces to motions. Whereas in some cases physical equations for these functions may already be known, there are many other cases where the environment is just too complex for these equations to yield acceptable accuracy. In such situations, learning to approximate the function from sample observations may yield significantly better accuracy. The functions that are approximated need not be continuous. Function approximating models also excel at classification tasks, such as determining what type of object lies before the robot, which grasping approach or general planning strategy is best suited for current conditions, or what the state of a certain complex object is with which the robot is interacting.

The next section reviews some of the many applications for classifiers, regression models, and discriminative models that have appeared in the recent literature with robotics. A function approximating architecture with rectifiers to model the highly coupled dynamics of a radio-controlled helicopter, which is a challenging analytic derivation and difficult system identification problem. Training data was obtained as a human expert flew the helicopter through various aerobatic maneuvers, and the DNN outperformed three state of- the-art methods for obtaining helicopter dynamics by about 60 percent. The time between a driver's head movement and the occurrence of a maneuver varies with vehicle speed. The resulting system made predictions every 0.8 s based on the preceding 5 s of data and anticipated maneuvers about 3.5 s before they occurred, with 90.5 percent accuracy. A great many works have used function approximating models in the domains of (1) detection and perception, (2) grasping and object manipulation, and (3) scene understanding and sensor fusion. The following three subsections describe recent works in each of these domains. Detection and Perception. DNNs have surged ahead of other models in the domains of detection and perception. They are especially attractive models because they are capable of operating directly on high-dimensional input data instead of requiring feature vectors that are hand-engineered at design time by experts in machine learning and the particular application. This reduces dependence on human experts, and the additional training time may be partially offset by reducing initial engineering effort. The most challenging aspect of dealing with reinforcement learning models is the enormous amount of computing time required to train them. Although such models are extremely efficient after training, they often need a large number of training pattern presentations before they converge to reflect accurate control rules. Taking the time to discover an efficient GPU-optimized implementation may therefore make a significant impact. Another useful approach is to practice with a virtual robot before trying to train with a real one. This saves time and money by reducing the wear and tear on physical equipment. Even if just a rudimentary simulation is available, a model that has been pre-trained on a comparable task will converge considerably faster than one that has been trained from scratch to suit the actual issue. Traditional Q-learning isn't immediately relevant to robots since they often work in an area with continuous activities. Actor-critic models, on the other hand, do a good job of addressing this issue. They use the continuous Q-table to regress actions, resulting in a final model that directly computes the optimal action given the current observation, which is ideal for robotics applications.

## CONCLUSION

Deep learning has shown promise in significant sensing, cognition, and action problems, and even the potential to combine these normally separate functions into a single system. DNNs can operate on raw sensor data and deduce key features in that data without human assistance, potentially greatly reducing up-front engineering time. They are also adept at fusing high-dimensional, multimodal data. Improvement with experience has been demonstrated, facilitating adaptation in the dynamic, unstructured environments in which robots operate. Some remaining barriers to the adoption of deep learning in robotics include the necessity for large training data and long training times. Generating training data on physical systems can be relatively time consuming and expensive. One promising trend is crowdsourcing training data via cloud robotics. It is not even necessary that this data be from other robots, as shown by Yang's use of general-purpose cooking videos for object and grasp recognition. Regarding training time, local parallel processing and increases in raw processing speed have led to significant improvements. Distributed computing offers the potential to direct more computing resources to a given problem but can be limited by communication speeds. There may also be algorithmic ways yet to be discovered for making the training process more efficient. For example, deep learning researchers are actively working on directing the network's attention to the most relevant subspaces within the data and applying biologically inspired, sparse DNNs with fewer synaptic connections to train.

**REFERENCES**

[1] N. G. Polson and V. O. Sokolov, "Deep Learning - Nature Review," *Nature,* 2018.

[2] A. Hosny *et al.*, "Deep learning for lung cancer prognostication: A retrospective multi-cohort radiomics study," *PLoS Med.*, 2018, doi: 10.1371/journal.pmed.1002711.

[3] S. M. Stigler, "Gauss and the Invention of Least Squares," *Ann. Stat.*, 2007, doi: 10.1214/aos/1176345451.

[4] S. P. Martin and S. M. Stigler, "Statistics on the Table: The History of Statistical Concepts and Methods," *Contemp. Sociol.*, 2003, doi: 10.2307/1556614.

[5] "Gauss and the invention of least squares Stigler, Stephen M 1981. Annals of Statistics 9, 465–474," *Hist. Math.*, 1982, doi: 10.1016/0315-0860(82)90072-6.

[6] R. J. Howarth, "Fitting Geomagnetic Fields before the Invention of Least Squares: II. William Whiston's Isoclinic Maps of Southern England (1719 and 1721)," *Ann. Sci.*, 2003, doi: 10.1080/713801783.

[7] Jürgen Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks,* 2015.

[8] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their

applications," *Neurocomputing*, 2017, doi: 10.1016/j.neucom.2016.12.038.

[9] G. B. Goh, N. O. Hodas, and A. Vishnu, "Deep learning for computational chemistry," *Journal of Computational Chemistry*. 2017, doi: 10.1002/jcc.24764.

[10] L. Deng, G. Hinton, and B. Kingsbury, "New types of deep neural network learning for speech recognition and related applications: An overview," 2013, doi: 10.1109/ICASSP.2013.6639344.