Custom Application in Salesforce using Augmented Random Search

Pro. Kanchan Jadhav[1] Kalpraj Dakhore[2] Rushabh Gupta[3] Mayur Khandare[4] Sahil Mahale[5]

[1]Assistant Professor [2,3,4,5]U.G. Student

[1,2,3,4,5]Department of Computer Engineering

[1,2,3,4,5]Sinhgad Academy of Engineering, Kondhwa, Pune, Maharashtra, India

Abstract— India is a developing country and in various developing The increasing demands of Robots and Robotic Simulation all over the world has given a new dimension to the Automation and Robotics Industry. The physical training and repairing of a robot in real environment can adversely affect the costs, thereby reducing the efficiency. Reinforcement Learning has proved out to be most prominent way to overcome this issue with the help of Simulations and Agent based modelling. The Reinforcement learning has Basic Random Search algorithm which is used to perform such competitive tasks but it limitates due to high variance in random values generated. In this paper, we are implementing a model free reinforcement learning algorithm known as Augmented Random Search Algorithm which uses Shallow Learning Neural network and Method of Finite Differences. The simplified policies and derivative free methods make this algorithm simple to work. ARS is at least 15X faster than Basic Random Search Algorithm, but it can be made more efficient with the use of cloud rather than using local system hardware. That's where the use of Salesforce helps us to utilize the cloud environment.

Keywords: Augmented Random Search, Salesforce

I. Introduction

The humans still are making mistakes that costs billions of dollars sometimes and AI is a possible alternative that could be applied in robotics to reduce the number of accidents. Most deep learning algorithm work with Gradient Descent and Back Propagation, this is very effective but also incredibly expensive in terms of computation. Advance deep learning algorithm take forever to train unless you have a powerful expensive GPU. This paper uses one of the fastest cutting edge breakthrough algorithms of 2018 in the field of robotics for continuous motion tasks with dense reward structure. It outperforms cutting edge deep learning algorithm while training upto 15 times faster. Augmented Random Search is a Shallow learning algorithm which uses random noise and genetic evolution to get cutting edge performance on locomotion tasks. The training process demands high GPU requirement which can be overcome by the use of Cloud based technology.

The Cloud technology implemented in this paper is Salesforce. Salesforce is a Customer Relationship Management solution that brings together companies and customers. Salesforce uses Rest API to integrate with the python script.

Performance of Reinforcement Learning agents can be measured and visualized with the help of graphical representations. Salesforce CRM is used to provide analytics such as Histograms, HeatMaps, Scatter plots and much more. Salesforce uses Apex programming language for integration with python script. Simultaneously, Salesforce provides cloud storage for storing relevant data. We are using cloud to store step by step training videos generated as a result of successful execution of ARS.

II. Related Works

In the recent work, the Basic Random Search was implemented to train the agent [1]. The computations of rewards made at each and every step makes it bit complex to use which gives rise to Augmented Random Search [1]. Deep Mind suite is a set of continuous control tasks with a standardized structure and rewards. It also adds physics methods such as loading, rendering, running the simulation and the effective step of benchmarking [2].

LabView is a program development environment, similar to C and Basic Development environments. Labview provides simulation environment selection and Robot modelling [3]. Custom Application Development in Cloud Environment, Salesforce provides vital functionality of Cloud Computing. Salesforce use Salesforce.com, Force.com, Visualforce and Apex language for UI and backend accesses [4]. Research on Imitation Learning, which learns policy based on data on the behaviour of experts without explicit rewards signal [5], Imitation learning tries to optimize policies based on deep reinforcement learning.

III. Augmented Random Search Algorithm

1) Hyperparameters: step-size α, number of directions sampled per iteration N, standard deviation of the exploration noise ν, number of top-performing directions to use b (b < N is allowed only for V1-t and V2-t)
2) Initialize: $M_0 = 0 \in R^{p*n}$, $\mu_0 = 0 \in R^n$, and $\Sigma_0 = I_n \in R^{n*n}$, j = 0.
3) while ending condition not satisfied do
4) Sample $\delta_1, \delta_2, ..., \delta_n$ in R^{p*n} with i.i.d. standard normal entries.
5) Collect 2*N* rollouts of horizon *H* and their corresponding rewards using the 2*N* policies

V1: $\{\pi_{j,k,+}(x) = (M_j + \nu\delta_k)x$
$,\pi_{j,k,-}(x) = (M_j - \nu\delta_k)x\}$

V2: $\{\pi_{j,k,+}(x) = (M_j + \nu\delta_k)\, diag(\Sigma_j)^{-1/2}\, (x-\mu_j)$
$\pi_{j,k,-}(x) = (M_j - \nu\delta_k)\, diag(\Sigma_j)^{-1/2}\, (x-\mu_j)\ \}$

for $k \in \{1,2,...,N\}$.

6) Sort the directions δ_k by $\max\{r(\pi_{j,k,+}), r(\pi_{j,k,-})\}$, denote by $\delta_{(k)}$ the k-th largest direction, and by $\pi_{j,(k),+}$ and $\pi_{j,(k),-}$ the corresponding policies.
7) Make the update step

$$M_{j+1} = M_j + \frac{\alpha}{b\sigma R} \sum_{k=1}^{b} [\, r(\pi_{j,(k),+}) - r(\pi_{j,(k),-})]\delta_{(k)},$$

where σ_R is the standard deviation of the 2b rewards used in the update step.

8) V2 : Set μ_{j+1}, Σ_{j+1} to be the mean and covariance of the 2N H(j + 1) states encountered from the start of training.
9) $j \leftarrow j + 1$
10) end while

ARS is a random search method for training linear policies for continuous control problems. Our ARS implementation relies on Python 3, OpenAI Gym version 0.9.3, mujoco-py 0.5.7, MuJoCo Pro version 1.31.

In simpler terms algorithm becomes like the following:

Let $\boldsymbol{\nu}$ a positive constant < 1 Let $\boldsymbol{\alpha}$ be the learning rate Let N the number of perturbations Let θ a (p x n) matrix representing the parameters of the policy π Let $\boldsymbol{\delta}$i a (p x n) matrix representing the ith perturbation

1) While end condition not satisfied do:
2) Generate N perturbations $\boldsymbol{\delta}$ from a normal distribution
3) Normalize πi+ = $(\theta+\boldsymbol{\nu\delta}i)^T x$ and πi- = $(\theta-\boldsymbol{\nu\delta}i)^T x$ for i = 1 to N
4) Generate 2N episodes and their 2N rewards using πi+ and πi- and collect the rewards ri+ and ri-
5) Sort all $\boldsymbol{\delta}$ by max(ri+, ri-)
6) Update $\theta = \theta + (\boldsymbol{\alpha}/(b*\boldsymbol{\sigma}_r))\ \Sigma(ri+ - ri-)\boldsymbol{\delta}i$ (where i = 1 to b)
7) End While.

The ARS is an improved version of BRS, it contains a three axis of enhancements that makes it more performant.

a) Dividing by the Standard Deviation $\boldsymbol{\sigma}_r$
b) Normalizing the states
c) Using top performing directions

IV. Methodology

The system is designed as per a developer's point of view. The first page implements a login implementation where developer has to login to existing credentials or sign up for new account. At successful login, the application provides a page for selection of MuJoCo and environment.

The developer can use provided agents and environments for training a similar kind of robot. The application implements the Augmented Random Search Algorithm for given specifications and number of parameters differing from agent to agent. Parameters include number of steps, episode length, noise, environment name and rewards.

The model is trained for agents and corresponding output videos are generated on how the model was trained with the use of algorithm. The statistical analytics is also developed for the same which gives an idea about efficiency of training with the help of pictorial representation such as Histogram or Heat Map. It is possible with the use of Salesforce CRM. Training videos can be stored on cloud and Histograms or Heatmaps can be generated with the use of Salesforce analytics.

Thus, the developer can conclude the training of agent and videos are stored on cloud for space saving efficiency.

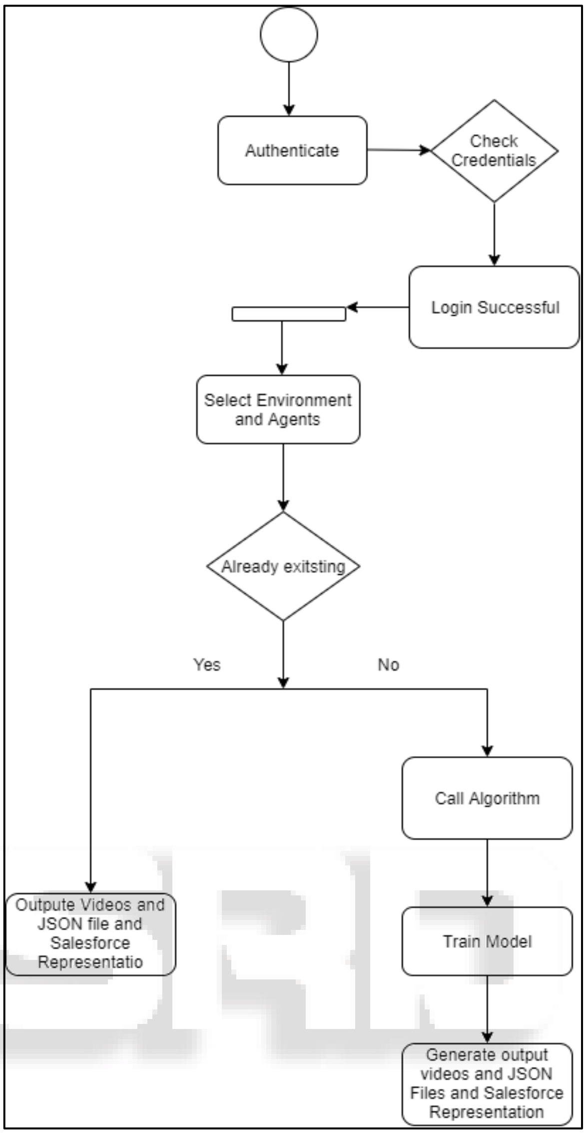

Fig. 1:

V. Advantages

– Our system implements Augmented Random Search algorithm which is 15x faster than the existing system.
– Rather than using the traditional storage, our system uses salesforce cloud to store and analyse the data.
– The Augmented Random Search implements integrated reward calculations which increases performance.

VI. Disadvantages

– The existing system was applying the tedious Basic Random Search algorithm.
– Also, the system was using the traditional way for storing the data such as HDDs.
– The Basic Random Search implements step by step calculations which leads to degradation of performance.

VII. Applications

A. Robotics in Industrial Automation:

Within industrial automation, robots are used as a flexible way to automate a physical task or process. Collaborative robots are designed to carry out the task in the same way a human would. More traditional industrial robots tend to carry out the task more efficiently than a human would.

B. Games:

Using this approach one can train an agent in a simulated environment to learn how to play games. RL is so well-known these days because it is the mainstream algorithm used to solve different games and sometimes achieve super-human performance.

VIII. Conclusion

ARS uses a perceptron instead of a deep neural network. ARS randomly adds tiny values to the weights along with the negative of that value to figure out if they help the agent get a bigger reward.

The bigger the reward from a specific weight configuration, the bigger its influence on the adjustment of the weights.

So at first, you can see that the agent barely lasts for a couple seconds before falling and the episode ends. A little bit later on, it's able to balance, but it doesn't understand quite how to walk using legs.

Finally nearing the end of the training, the agent understands how to balance and use legs to push itself forward. Although these were short peeks into what the agent was doing, you can clearly see the progress from not being able to balance, to not moving, to finally balancing and moving to walk.

Acknowledgement

This work was supported by Sinhgad Academy of Engineering, Pune, Maharashtra.

References

[1] Horia Mania Aurelia Guy Benjamin Recht, "Simple random search provides a competitive approach to reinforcement learning", Department of Electrical Engineering and Computer Science University of California, Berkeley March 20, 2018
[2] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, Martin Riedmiller "DeepMind Control Suite", January 3, 2018.
[3] Chang xiaowei, Sun chuanwei "The study of robot simulation applications based on the LabVIEW Robotics", 2017 4th International Conference on Information, Cybernetics and Computational Social Systems (ICCSS)
[4] Anuradha Manchar, Ankit Chouhan,Salesforce "CRM : A new way of managing customer relationship in cloud environment", 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)
[5] MyungJae Shin , Joongheon Kim, "Adversial Imitation Learning using Random Search", s 2019 International Joint Conference on Neural Networks (IJCNN)