International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 6 | ISSN : 2456-3307

Case Study- Machine Learning Based on Scenario Analysis

Prabadevi. B*[1], Vanmathi. C[1], Krithika. L. B[1], Sudha. S[2], Vivek Akabote[1]
*[1]School of Information Technology and Engineering, VIT University, Vellore, Tamilnadu, India
[2]Victoria University, Melbourne, Australia

ABSTRACT

Machine learning now-a-days becoming too obligatory, so that we use it in wide range of applications like predicting next words in keyboard to space related computing. In reinforcement machine learning which depends only on the rewards it gained previously, handle only interactive situations and never knew about the factors that influence in gaining rewards. The proposed deep learning system handles both interactive and non-interactive situations. In this paper, the main goal is to make the machine to learn whose factors in environment are affecting the reward, so to make this possible deep learning system is used. Here machine learns about the factors that affects the reward and machine can act on that factors to get best reward and even the knowledge which has gained in this system can be used with different applications. A car tuning machine is developed by using this deep learning system which is able to tune the car based up on selected track so it can get maximum performance. The system is implemented in python, a simulator is designed to simulate the environment of different tracks and which can be used to race by tuning the car.

Keywords : Deep Learning, Reinforcement Learning, Machine Learning, Reward, Agent, Environment

I. INTRODUCTION

Artificial intelligence has a subfield named Machine learning which deals with construction and study of algorithms which can learn from data and make predictions on data [1]. By building a model, this kind of algorithms are operated [2]. This can make predictions or choose a decisions by learning from data other than static instructions. Some of the concepts in Machine learning are closely related to computational statistics which are used in tasks like making-predictions.

Machine learning has a sub-branch called Reinforcement learning which is inspired by behaviorist psychology. Reinforcement learning is concerned with how software agents reacts when it gets a reward for choosing an action by observing the state of environment. This problem is studied in many other disciplines, like as simulation-based optimization, game theory, operations research, swarm intelligence, genetic algorithms, information theory, multi-agent systems, statistics, and control theory because of the generality of the problem. Reinforcement learning is also called approximate dynamic programming because of studying of methods in the operations research and control literature. [3]

Core of the machine learning is to generalize all its experiences and learn from those, so that it can handle a new unseen task. In the present trend the need of intelligent machine is more, there are lot of applications that are depending on this kind of system like filtering spam from mails, medical diagnosis, removing noise from signal, detecting unusual traffic in network, marketing predictions, weather prediction and intruder detection.

A tutorial that introduces current applications, techniques and systems with the aim of cross-fertilizing research between the database and machine learning communities. Recent results like tera-scale learning [4] and very large neural networks [5] suggest that scale is an important factor in quality modelling. This tutorial covers current large scale applications of Machine Learning, workflow behind building them and their computational model [6].

The organization of this document is as follows. In Section 2 provides methods and materials used, Section 3 discusses the result and section 4 concludes the paper.

II. METHODS AND MATERIAL

2.1 Deep Learning System Architecture:

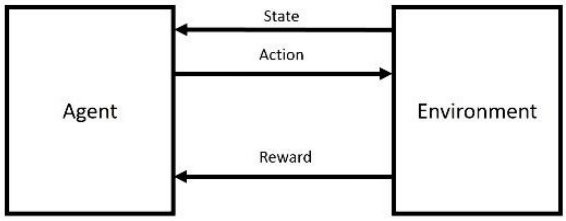

Figure 1. Deep Learning Machine architecture

An easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it.

2.1.1 Working

i. Get the state of the world, S.
ii. Choose an action A=(S,K) where K is knowledge of Machine
iii. Carry out action A in the world
iv. Get the reward R.
v. Update the knowledge K with gained reward R K'=update(K,R)
vi. Repeat above steps for every scenario.

The machine when called to perform an action for the given scenario will follow the above steps. First, it reads the state of the world, gets the analysis of the scenario very well and chooses an action for the given state of the world using knowledge K it obtained and then it carries out the action it chosen. After successfully completing the action, machine gains the reward R from the environment, knowledge is then updated by making use of obtained reward R from the environment. After updating the knowledge, machine can be used for choosing action for next move.

This kind of system is useful when responsive actions can't be made according to state change. In this system an agent will study the environment state briefly and perform set of actions, after several state transitions undergone in environment it gives a final reward. Agent has to make action by briefly analyzing the state of environment for desired reward since it has to make action only once in a course of time. In this model, State transition matrices which holds all state transitions P=f (S, A) is generated for action made to respective state and reward function=r (p) calculate final reward which will decide the loosing or gaining for the action made.

Scenario simulator takes input set 'I' which is set of individual input elements I = [i1, i2, i3... in](individual inputs are like individual tuning parameters of a car like engine tuning or transmission tuning which can be set independently). But Machine takes State transition matrix P, reward R (reward which finally conclude loosing or gaining on an event) and input set I (which used to generate P and R) as input. Machine generates an action A, which is input, set I to simulator, chooses appropriate to the current state based on the knowledge gained. Simulator generates a state transmission matrix P and final reward R.

Total number of turns required for gaining maximum knowledge on a single track :

$$V_{max} = \sum^{1}_{i=n} S(I_i) \qquad \text{eq (1)}$$

V_{max} : maximum knowledge

S(I):function to find no of states of input element

I: Input set
i: input Element

2.2 System Architecture

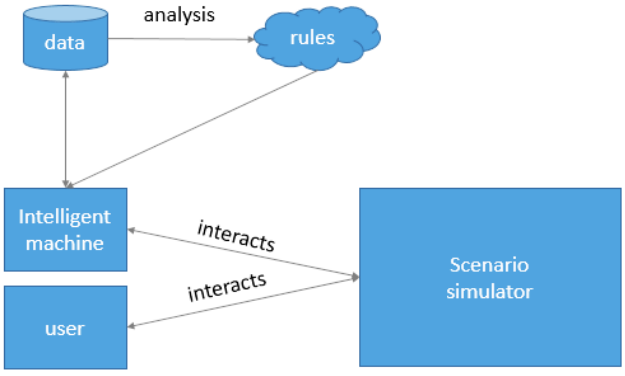

Figure 2. System Architecture

This system mainly consists of three following components:

User interface : User interface is used to interact with the simulator and compete with machine.

Simulator : Simulator takes the input values and computes the scenario and generates the output.
Machine : Machine here gives input to simulator based on knowledge it has gained and reads the complete simulation and result. It gets data from the simulation and processes that data for the gaining knowledge.

2.2.1 Working:

i. User initiates the process by selecting a track and choosing appropriate car tuning parameter to deal with given situation.
ii. The machine will choose its input set based on the knowledge it gained.
iii. After getting input from both user and machine, simulation is performed.
iv. Based on the simulation results, the machine starts the analysis process and frames rules

2.3 Proposed System Design

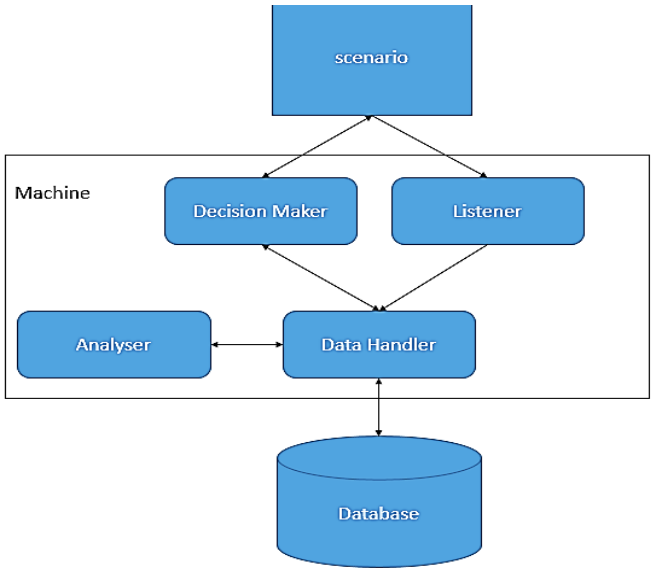

Figure 3. Intelligent machine workflow model

2.3.1 Workflow model components

Listener: Listener component's job is to listen to the scenario and collect the all possible data from the simulation.
Data Handler: All the other components that are in need of data or to save data must requests to this component and this takes the responsibility of handling the data like keeping data to storage device and retrieving the data that is requested from filtering other data that is present along with it.
Analyser: Analyser will get the data which is relevant to compare and computes the difference ratios that occurred by selected input, so that it can generate rules
Decision Maker: By using the rules generated by the analyser this will compute the best input for the given scenario.

2.4 Gaining knowledge from scenario

In this system, machine learns from every simulation. A simulation can be of any kind of event like selecting best protocol for given network or selecting best car for a given race track. After the simulation, Listener stores the complete simulation data with the help of data handler. Analyser does its work after every simulation, it gets reward obtained from simulation, input selected for that simulation and complete simulation data. By classifying and clustering of data based on state changes that environment has undergone, the input analyser generates knowledge. Decision maker will make use of knowledge generated by analyser when it has to choose input by analysing the present state of environment.

Intelligent machine workflow model has two states which can be switched based on the application priority.

i. Learning state which is used when gaining knowledge is more important
ii. Trying state which is used to try winning with the knowledge it got.

Learning State algorithm: This is more concerned about learning than winning. If there is any possibility of gaining knowledge from encountered situation, this will choose to learn by experimenting the new possibility and records its outcome, which may lose that turn but gains knowledge about newly experimented possibility.
For each t in T and each i in I

- If K(t,i) == NULL
- Copy I_u to I_m
- Replace element i In I_m

T : total track
t : segment of track
I :All possible input set
i : input element
I_u : User's input set for simulator
I_m : Machine's input set for simulator
K(t,i) :function to checks knowledge on similar kind of track segment t with input element i

Trying State algorithm: This is more concerned about winning. This will uses all its knowledge for winning the encountered situation, this algorithm choose the best possible way of facing situation based on knowledge it gained.

- For each t in T
  - p_t = max_p(t)
- I = avg(P,T) where P=[p_1,p_2,p_3…..p_n]

p_t : best performance of input set for similar track segment t
P : this set contains best performances p_t of all segments t in the track T differently
max_p(t): returns best performance of the segment t
avg(P,T) : returns a final Input I for simulator by averaging performance of all segments in P while considering the length of each segment from T

2.5 Analysing and extracting knowledge from results

After running the simulation it collects all the data and start the process by steps as follows:

i. Classify the data collected in an experiment in to their category, to differentiate between the sequences of events that happened in the experiment.
ii. Cluster the whole data that has gathered by collecting from all the experiments by the classification done in the above step.
iii. Now based on the experiment compare the output of the data sets which are only varying with common input parameter, by this the impact of that input parameter in the experiment can be analysed.
iv. As data compared, by the splitting them the impact of input parameter in an individual event that had occurred in the experiment can be analysed.
v. By calculating the effect of an input parameter the impact of that input parameter can be generalised and can define rule for the kind of event to which the given input parameter gives best results
vi. Keep above steps repeating after every simulation or experiment.

2.6 Finding effect of input parameter on the result

This task is carried out by analyser component. To start analysing the effect of the input parameter on the result it needs to collect the data of an event with most similar set of input parameters so that the output of the simulations should only affected by the single input parameter. After collecting the data, it starts comparing the all possible valid combinations of attributes of output with varying results of given respective input parameter value and gets the ratio of change with different input parameter and writes the affecting factors along with its ratio of change as a rule. When it gets a turn to select the set of input parameters it calculates the best available set of input to the given condition. When a comparison is made between two sets of output, given conditions also considered in making of a rule.
In this system, machine is able to interact with the simulator and can give input to simulator and read output from simulator. In first run or when it don't have any rules framed it just act like a human i.e. it copies the input values from the user input and vary one input parameter which it doesn't have any kind of data about it and if it doesn't have any data about many input parameters it chooses randomly and runs simulation. It copies the output of simulator to its database and starts analysing, when it is able to extract any information from the data it gathered it frames rules to choose what kind of input parameter is chosen for the given experiment. So by this way it can make progress in every step.

III. RESULTS

Scenario is competition between two cars human tuned car and computer tuned car, the tuning is to be done by analysing the chosen track, different modified car will have advantage over different tracks, considering both cars are driven by the same driver one after another to measure exact performance difference. Here we choose a track and tune car according to the track, the input given here is performance tune of the car and machine will also chooses its performance tune and competes. The expected output is machine should able to learn how to use correct performance tune for given track and win over human tuned car.

Seven experiments are conducted in a row when knowledge gained by machine at that movement is zero, on the Track2 and with input used to tune car is:

– Engine : Max Torque
– Transmission : Max Topspeed
– Tyres : Grip
Brakes : Balanced

Figure 4. Track 2

TABLE 1

MACHINE TUNED CAR'S PERFORMANCE (1-7)

Inputs given by Machine				Time	Top Speed	Average speed
Engine	Transmission	Tyres	Brakes			
Balanced	Top speed	Grip	Balanced	1:42.461	177.174	114.540
Torque	Balanced	Grip	Balanced	1:40.368	226.006	116.929
Torque	Top speed	Balanced	Balanced	1:42.581	201.062	114.406
Torque	Top speed	Grip	Front Biased	1:41.589	180.877	115.523

Horse power	Top speed	Drift	Front Biased	1:39.267	240.273	118.226
Torque	Top speed	Grip	Rare Biased	1:44.444	180.897	112.366
Horse power	Top speed	Drift	Front Biased	1:39.267	240.273	118.226

TABLE 2

USER TUNED CAR'S PERFORMANCE (1-7)

Inputs given by User				Time	Top Speed	Average speed
Engine	Transmission	Tyres	Brakes			
Torque	Top speed	Grip	Balanced	1:42.707	180.877	114.266
Torque	Top speed	Grip	Balanced	1:42.707	180.877	114.266
Torque	Top speed	Grip	Balanced	1:42.707	180.877	114.266
Torque	Top speed	Grip	Balanced	1:42.707	180.877	114.266
Torque	Top speed	Grip	Balanced	1:42.707	180.877	114.266
Torque	Top speed	Grip	Balanced	1:42.707	180.877	114.266
Torque	Top speed	Grip	Balanced	1:42.707	180.877	114.266

Now changing the strategy for next three experiments, following input is used to tune user's car:

- Engine : Horse power
- Transmission : Acceleration
- Tyres : Drift
- Brakes : Balanced

TABLE 3

MACHINE TUNED CAR'S PERFORMANCE (8-10)

Inputs given by Machine				Time	Top Speed	Average speed
Engine	Transmission	Tyres	Brakes			
Horse power	Balanced	Drift	Balanced	1:40.324	295.662	116.980
Horse power	Acceleration	Balanced	Balanced	1:39.797	284.749	117.598
Horse power	Top speed	Drift	Front Biased	1:39.267	240.273	118.226

TABLE 4

USER TUNED CAR'S PERFORMANCE (8-10)

Inputs given by User				Time	Top Speed	Average speed
Engine	Transmission	Tyres	Brakes			
Horse power	Acceleration	Drift	Balanced	1:41.350	293.000	115.796
Horse power	Acceleration	Drift	Balanced	1:41.350	293.000	115.796
Horse power	Acceleration	Drift	Balanced	1:41.350	293.000	115.796

By this stage it has gained maximum knowledge on this track using random input.

TABLE 5

MACHINE TUNED CAR'S PERFORMANCE (11-13)

Inputs given by Machine				Time	Top Speed	Average speed
Engine	Transmission	Tyres	Brakes			
Horse power	Top speed	Drift	Front Biased	1:39.267	240.273	118.226
Horse power	Top speed	Drift	Front Biased	1:39.267	240.273	118.226

TABLE 6

USER TUNED CAR'S PERFORMANCE

Inputs given by User				Time	Top Speed	Average speed
Engine	Transmission	Tyres	Brakes			
Horse power	Acceleration	Balanced	Balanced	1:39.797	284.749	117.598
Torque	Balanced	Grip	Balanced	1:40.368	226.006	116.929

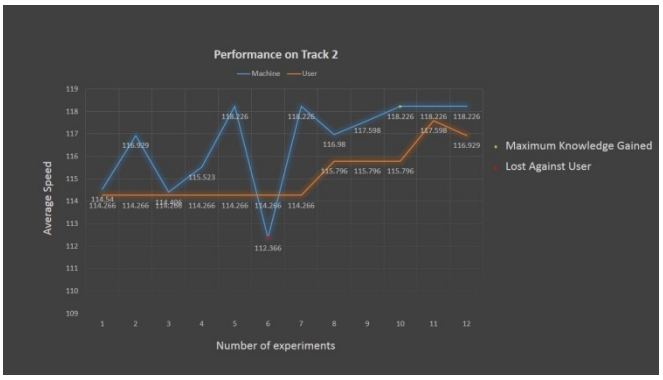

Figure 5. Performance graph on track 2

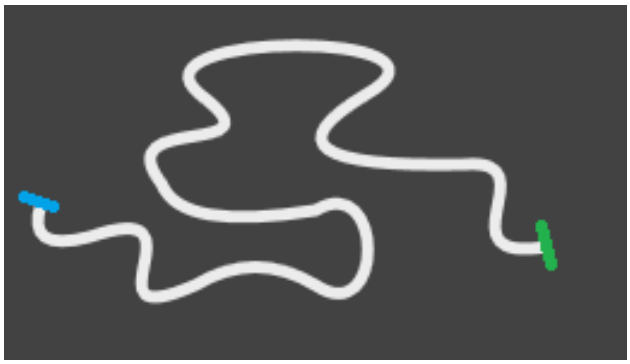

Figure 6. Track 5

Now trying other track Track 5

TABLE 7

MACHINE TUNED CAR'S PERFORMANCE (14-16)

Inputs given by Machine				Time	Top Speed	Average speed
Engine	Transmission	Tyres	Brakes			
Torque	Top speed	Drift	Rare Biased	2:24.277	146.785	84.536
Balanced	Top speed	Balanced	Balanced	2:15.944	153.925	89.719
Balanced	Balanced	Drift	Balanced	2:13.601	150.760	91.592

TABLE 8

USER TUNED CAR'S PERFORMANCE (14-16)

Inputs given by User				Time	Top Speed	Average speed
Engine	Transmission	Tyres	Brakes			
Torque	Balanced	Drift	Rare Biased	2:25.288	144.342	83.948
Torque	Top speed	Balanced	Balanced	2:18.235	151.357	88.232
Balanced	Acceleration	Grip	Balanced	2:13.675	149.575	90.441

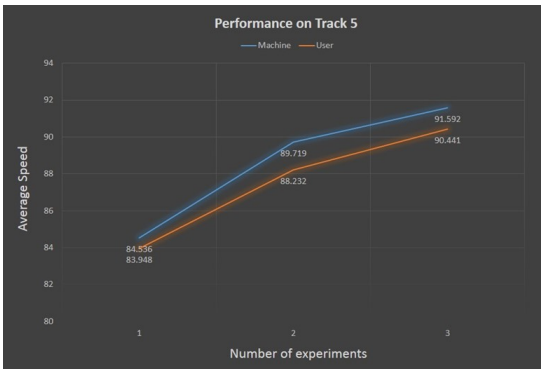

Figure 7. Performance graph on track 5

In the Track 2 machine doesn't have any knowledge, but it has gained its knowledge after each experiment, as we can notice in the processes of gaining knowledge it had tried all its possible ways of giving input to the simulator that is why the performance graph of machine has lot of flickering, in this process it lost against in the sixth attempt for finding the impact of selected input in that scenario, at last it has gained maximum knowledge on that track from tenth experiment and from this point it made itself very difficult to defeat in that track.

As it had gained some knowledge from the previous competition it can be shifted to trying mode from learning mode, which make use of present knowledge and try to win, in track 5 three experiments were conducted, it won in all three attempts although it doesn't have any experience in track 5. It was successfully able to learn from the environment and win even in new scenarios.

IV.CONCLUSION

Machine learning by scenario analysis is implemented successfully. The system is able to learn the knowledge from scenario by analyzing the simulation and can use it for right situations. Unlike commonly used methods such as choosing input for success ratios, this tries to learn by deriving the effect of individual input parameter on the output in different circumstances and use correct input for given situation.

This kind of machine learning is useful in real time scenario where no training is given before using it. This can also be used in studying new things or when

there are no solutions and need to deal with it. Also a car tuning machine is designed, which is able to tune the performance of car based on the given track. This machine is able to learn, by selecting proper input for given track to get best from car with no training and machine tuned car was able to win over human tuned car. So these techniques can be used where machine learning is needed without pre required training.

V. REFERENCES

[1] C. M. Bishop. “Pattern Recognition and Machine Learning” Springer. ISBN 0387310738,2007

[2] Wernick, Yang, Brankov, Yourganov and Strother, “Machine Learning in Medical Imaging”, IEEE Signal Processing Magazine, vol. 27, no. 4, July 2010, pp. 2538

[3] Kaelbling, L.P., Littman, M.L., and Moore, A.W,"Reinforcement Learning: A Survey", Journal of artificial intelligence Research 4, 1996, pages 237-285.

[4] A. Agarwal, O. Chapelle, M. Dudik and J. Langford, "A Reliable Effective Terascale Linear Learning System," arXiv.org, 2012.

[5] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, Q. Le, M. Mao, A. Senior, P. Tucker, K. Yang and A. Ng, "Large Scale Distributed Deep Networks," in Advances in Neural Information Processing Systems, 2013.

[6] Tyson Condie, Paul Mineiro, Neoklis Polyzotis and Markus Weimer, “Machine Learning on Big Data” , IEEE International Conference on Big Data, 2013, 978-1-4673-4910-9.