Proposal Abstract

- Artificial insights (AI) is the recreation of human insights forms by machines, particularly computer frameworks. These form incorporate learning, thinking, and self-correction. Reinforcement learning depicts the set of learning issues where a specialist must take activities in an environment in arrange to maximize a few characterized compensate work. Hence in this thesis, we will be developing a reinforcement learning model to train it to walk by itself by understanding the performance of a task from its experience. It can be deployed into real robots and maybe helpful to humanity for assigning tasks that are complicated by a human being. Each state will be awarded positive and negative rewards for success and failure execution correspondingly. This model will be trained on real-time scenarios with different obstacle based environments, which will have the scope of performance improvement by learning using the rewards points. The main focus of this thesis is to create a different model which can self learn using ARS algorithm of reinforcement learning, and for testing algorithm might it can improve if the random noise with respect to weights is more. Further, various performance tuning and metrics will be evaluated with results and outcomes will be interpreted at each stage.

Section 1 - Introduction

- In today's world, life is complicated and hard to imagine, without the help of Artificial intelligence. AI plays a major role in shaping the future of humanity. Each tech giant wants to step into the AI world to make there task more comfortable and smoother. Reinforcement Learning (RL) is a part of AI which concept is to train in such a way with the trial, punishment, and reward mechanism. The main focus of this project is on two things Algorithm and Environment. The environment is hardcoded in that way that it gives reward based on how the action performs into the environment by an agent (humanoid, half-cheeta, Racecar, ant, Hopper, etc.) and the Algorithm gives direction to the agent and try to improve the performances based on trial and error strategy. The main focus of the Algorithm is to improve reward, which is given by an environment. ARS doesn't have any hidden layers, means that the input comes from the environment, and goes to perceptrons further, the output goes to the environment. While backpropagation, most of AI uses the method of gradient and decent, but ARS uses finite differeces that is to be seen in section 2. The Algorithm checks into the two directions and picks maximum episode reward from multiple episodes.

- Up till now, the progress made by me is in different areas. Firstly, I have selected tools and the environment used in the project. Secondly, I had a depth analysis of the Algorithm and working that discussed in section 2. I have tested the code into the different models, but it is tasking to long to compute the results. Then after I thought to use mean absolute deviations instead of standard deviations because of outliers. Further, the comparison of the results is going on with different models. In the future, I also planned to change the random noise system and the size of the episode performed in one iteration because if the more number of the episode is there, then the evaluation of the results may be faster. As we move in Section 2, an in-depth analysis of the Algorithm described with the future work to be proposed.

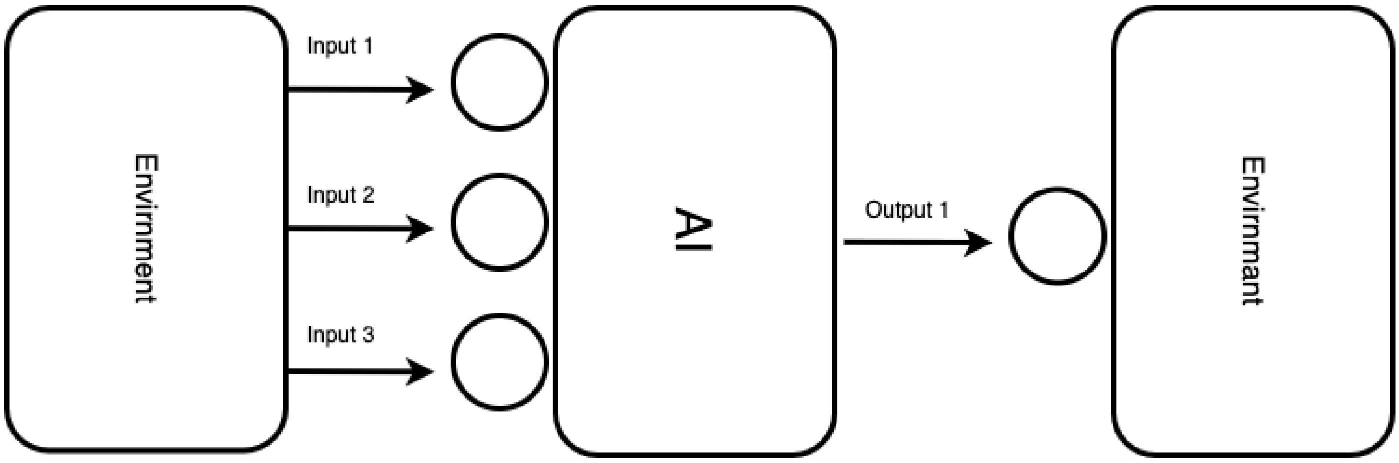
Section 2 - Latest Progress

- The algorithm detail model system described below from environment communication perceptrons to adjusting weights (also known as backpropagation). Let's have a closer look at every component and working of the algorithm.

Working of the algorithms

2.1 Environment Working

- The below figure (1) describes the working of environments and algorithms. If this model used in a real-world robot instead of the virtual world, then the input data describes as a sensor like a particular task to accomplish by the set of object examples, like a dog robot doing movement of picking any object with a mouth. AI becomes a brain for a robot that gives instruction, and the output becomes the improvement or an achievement done by a robot.

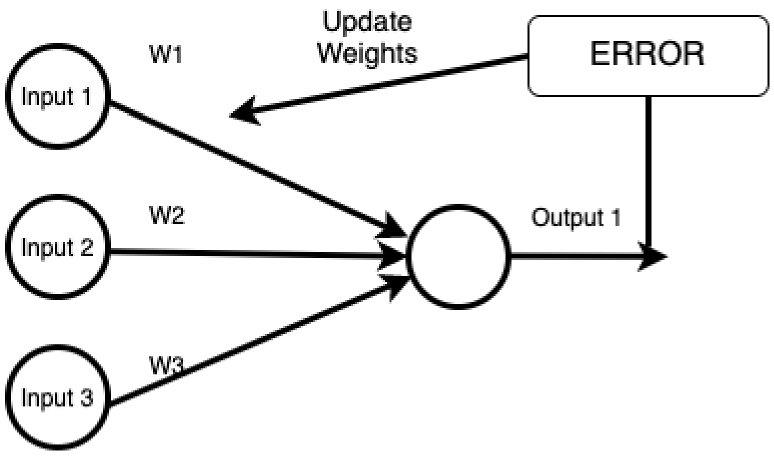


1. Working of Environment

- In figure 1, AI is nothing but the perception, and its tasks are to take input and weights and multiply them, as shown in the below equation. It also learns by itself adjusting weights. This process is also known as backpropagation and the working of this process done by taking results (Errors) and updating the weights.

$$\sum_{x=1} input1 * W1$$

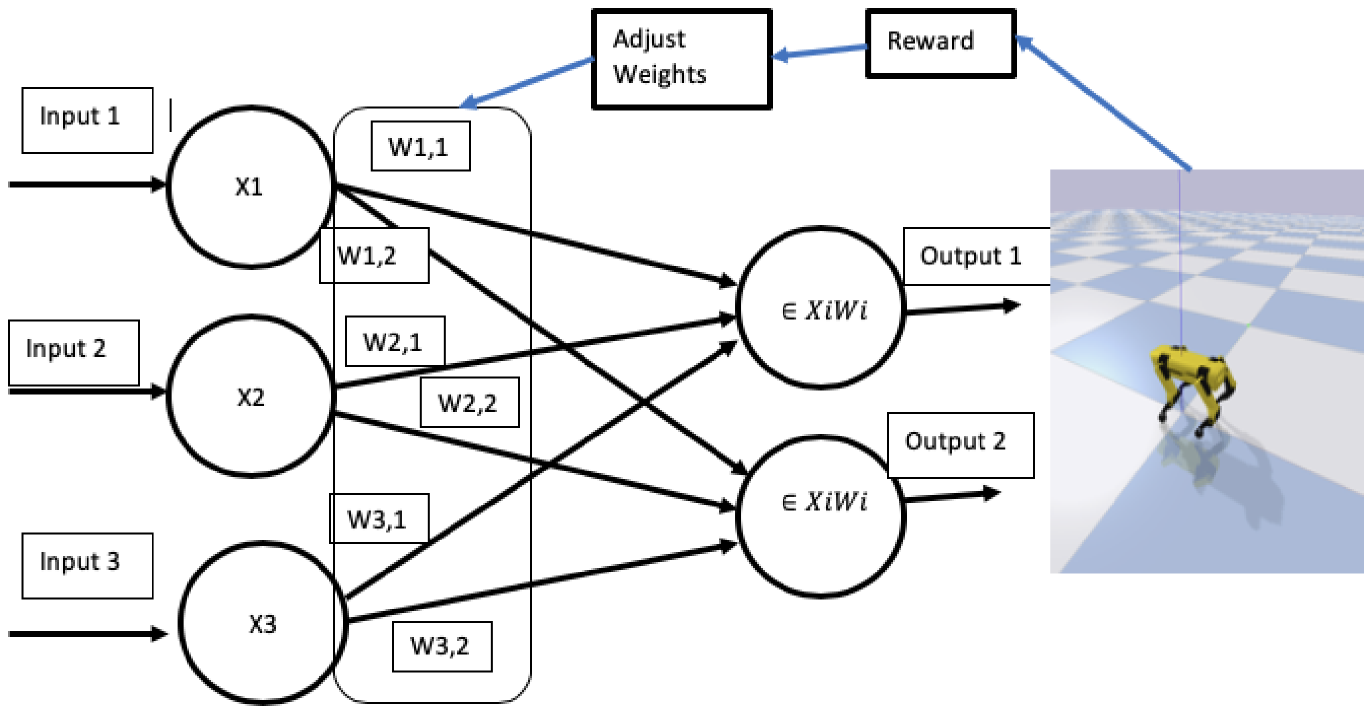
$$formulaofpercptron$$



2. Working of percptron

- The environment gives rewards after each episode plays. The pybullet environment coded in that form that it is capable of providing reward based on the moments of an object like some physic (object moments) not performing right action into the environment that may be the judge as a negative reward by a pybullet environment. In adjusted weights, the old value replaced with new rewards because of the updated weights goes to more positive reward. The adjusted weights described below.

weights goes to more positive reward. The adjusted weights described below.



3. Gain more Reward

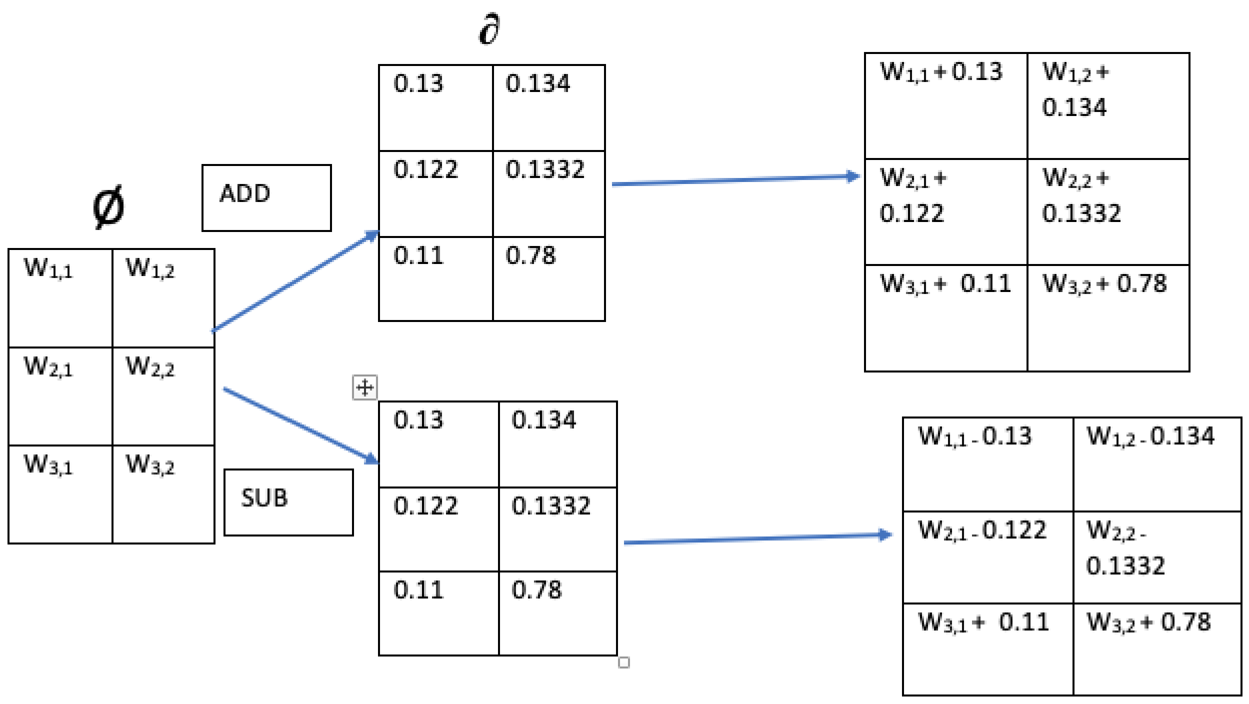
2.2 Deep Analysis of Algorithms

- Mostly AI model uses gradient and descent as backpropagation but in reinforcement learning uses finite differences for updating weights. Every model needs a mechanism like backpropagation or adjust weights for the performance of the model; it not only gives a proper reward, but it also provides accuracy that the model can perform batter. Now let's look deep into how the weights adjusted into the model.

- Now let's take :

```
Θ  = Weights
δ  = Random deltas or perturbations
v  = cofficent noise (just a tiny noise)
s  = 4
```

- Now we have matrixes of weights which received as an output from the environment also known as Θ and then after the random perturbation generated is known as δ.



4. Deltas assign

- As in figure 4, there are two weights, one for positive and another for negative like this; there are several directions. The direction always remains into 2^n means 2*4 = 8 examples; four directions for positive and another for negative. Each direction flows into the positive (Θ1,1 + R(δ1,1)) and negative direction (Θ1,1 - R(δ1,1)) . either positive or negative R(δ1,1) multiplied with a small chuck of noise knows as V, so the formula becomes for positive direction 1 episode (Θ1,1 - V R(δ1,1)). Here Θ1,1 means data in a matrix of the first episode row and column.

W1,1 + R(D1,1)	**W1,2+ R(D1,1)**
W2,1 + R(D1,1)	**W2,2 + R(D1,1)**
W3,1 + R(D1,1)	**W3,2 + R(D1,1)**

W1,1 + R(D1,1)	**W1,2+ R(D1,1)**
W2,1 + R(D1,1)	**W2,2 + R(D1,1)**
W3,1 + R(D1,1)	**W3,2 + R(D1,1)**

W1,1 + R(D1,1)	**W1,2+ R(D1,1)**
W2,1 + R(D1,1)	**W2,2 + R(D1,1)**
W3,1 + R(D1,1)	**W3,2 + R(D1,1)**

W1,1 + R(D1,1)	**W1,2+ R(D1,1)**
W2,1 + R(D1,1)	**W2,2 + R(D1,1)**
W3,1 + R(D1,1)	**W3,2 + R(D1,1)**

W1,1 - R(D1,1)	**W1,2 - R(D1,1)**
W2,1 - R(D1,1)	**W2,2 - R(D1,1)**
W3,1 - R(D1,1)	**W3,2 - R(D1,1)**

W1,1 - R(D1,1)	**W1,2 - R(D1,1)**
W2,1 - R(D1,1)	**W2,2 - R(D1,1)**
W3,1 - R(D1,1)	**W3,2 - R(D1,1)**

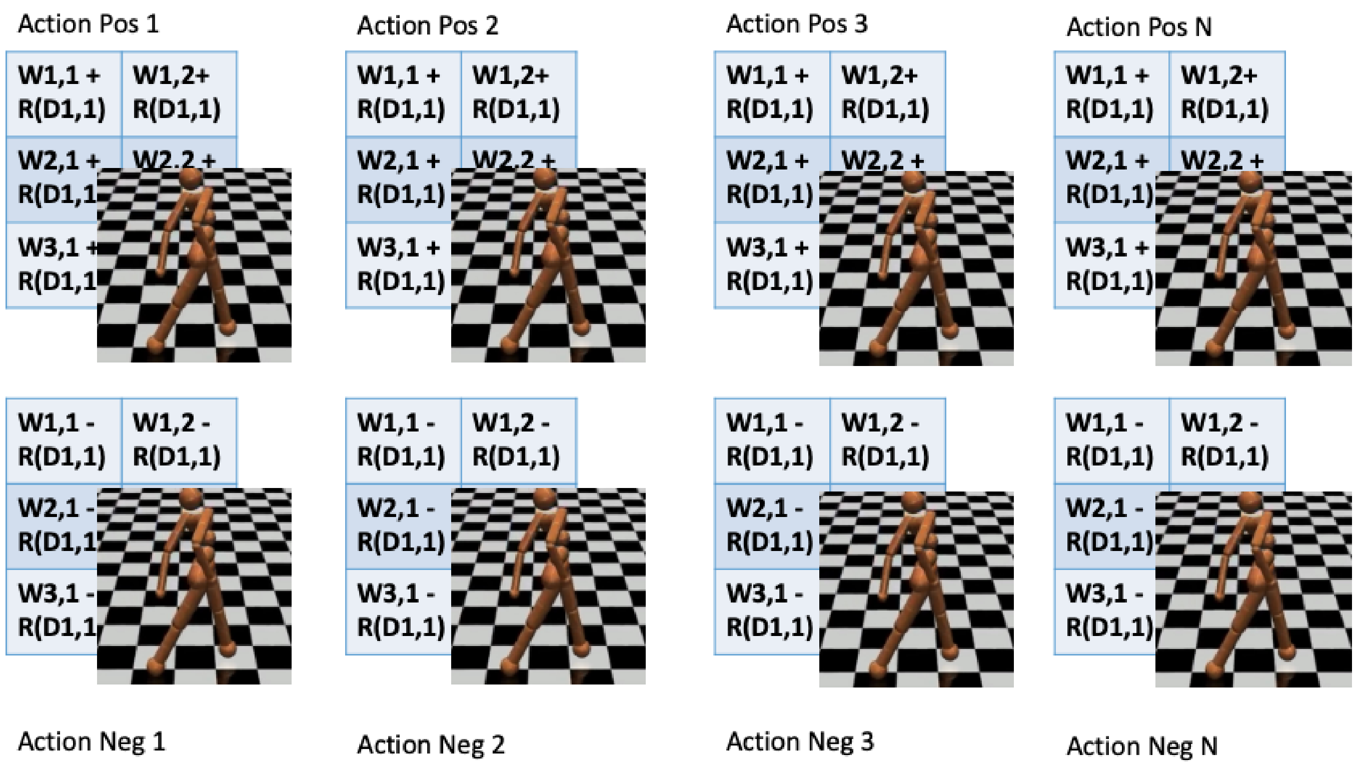
W1,1 - R(D1,1)	**W1,2 - R(D1,1)**
W2,1 - R(D1,1)	**W2,2 - R(D1,1)**
W3,1 - R(D1,1)	**W3,2 - R(D1,1)**

W1,1 - R(D1,1)	**W1,2 - R(D1,1)**
W2,1 - R(D1,1)	**W2,2 - R(D1,1)**
W3,1 - R(D1,1)	**W3,2 - R(D1,1)**

5. After Add and SUB

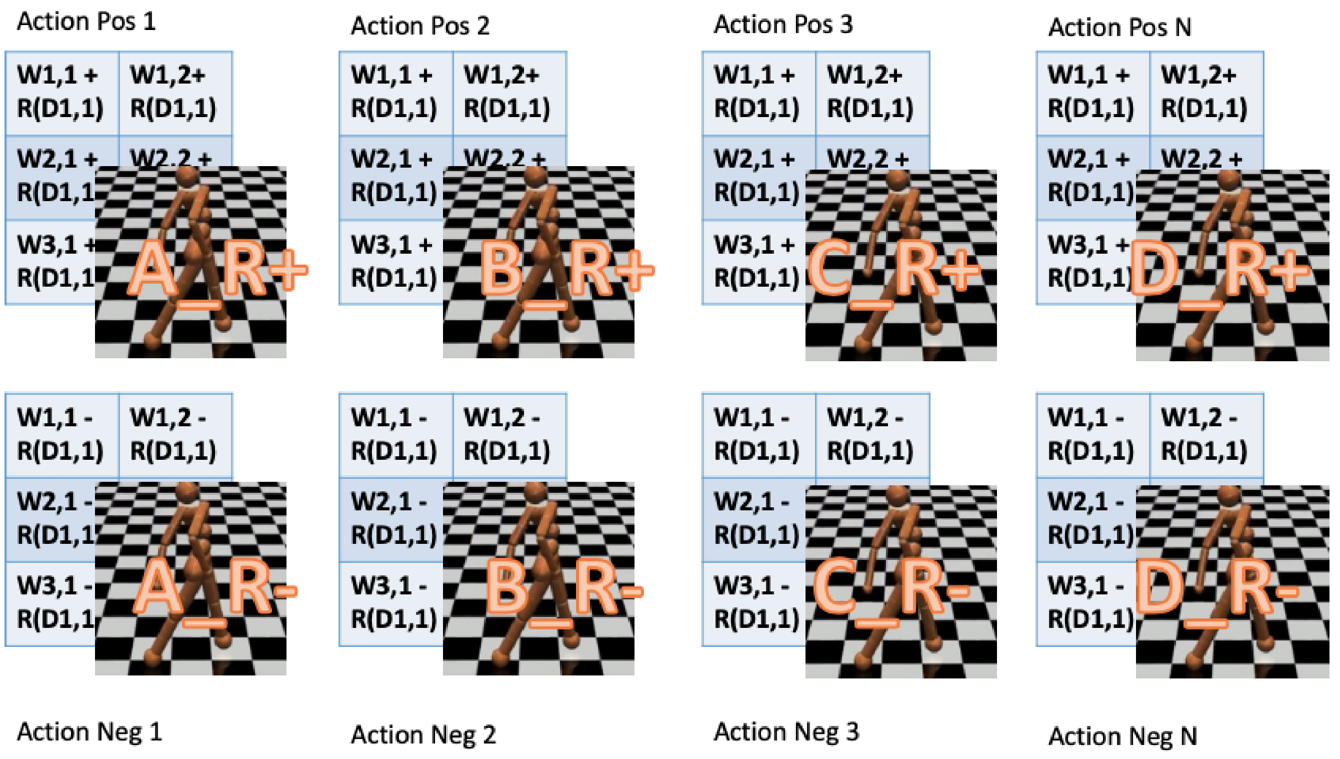
- For each weight, the random noise assigned with a positive direction or negative direction like similarly, it does with our 2^s let's say we assume s = 4 then it will generate eight total frames. The example in figure

5 is a small scale while running the algorithm; it may be more frames like 120 or even more than that.



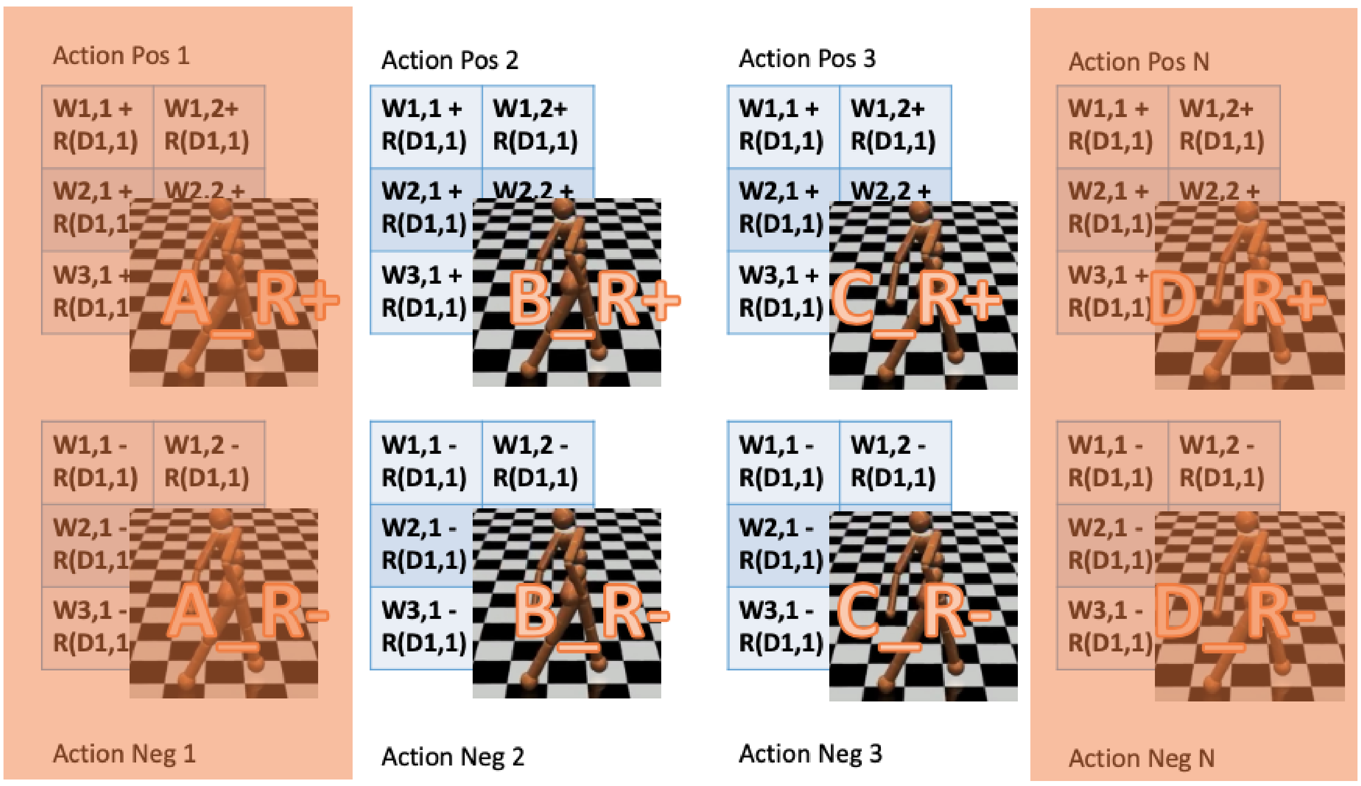
6. After Add and SUB

- Then in each frame, the specific action is performed by an environment. Here the noise is random, so in every frame, there will be a different action. For example, if the w11 = 2 and R(δ1,1) = 0.02 then in positive direction action be performed by humanoid is keeping 1 step further while in the negative direction the humanoid will keep 1 step backwards.



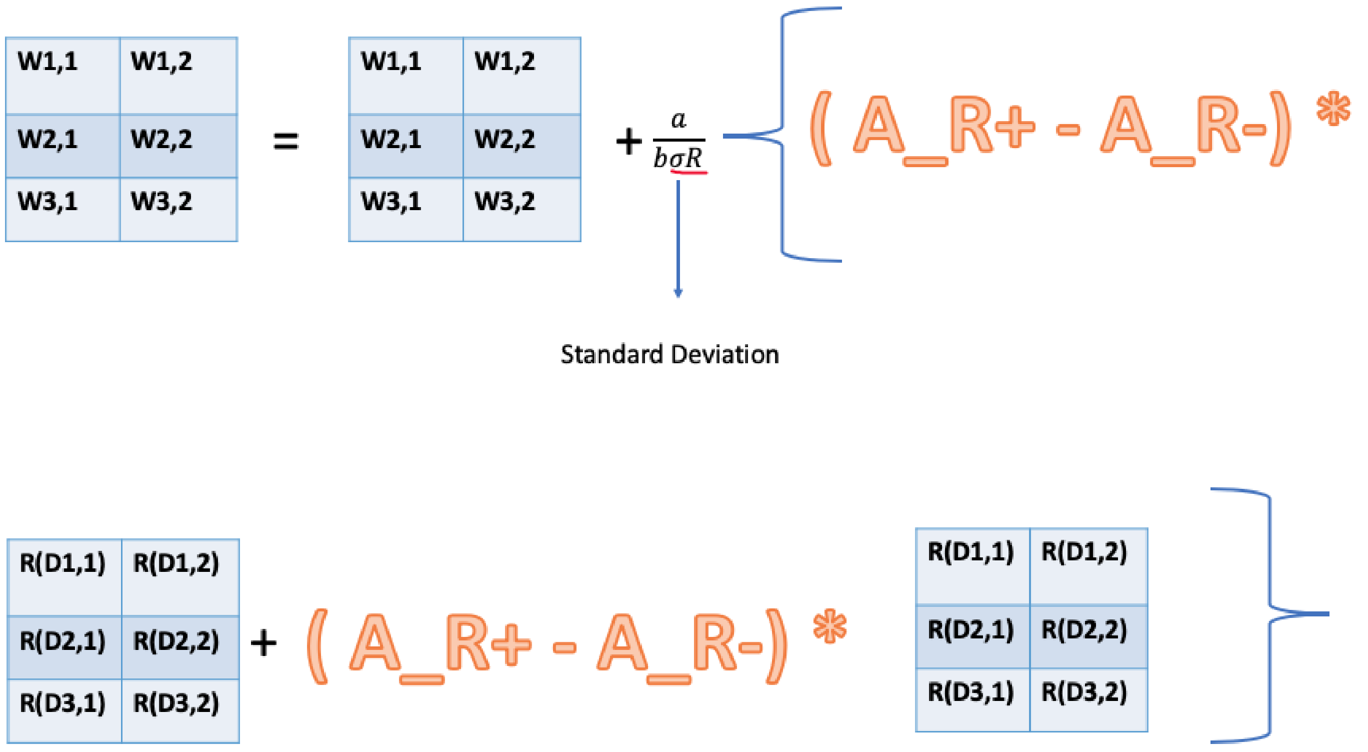
7. Gain Reward for each action

- The environment will give reward on the action performed by an model. Here the environment name pybullet is used, and it is hardcoded in that form that it can provide rewards based on moments done by physics. Each frame gets one particular reward based on the action performed in pybullet environment.



8. Select Maximum Reward

- The algorithm identifies the maximum reward and picks both directions of that reward positive as well as negative. If the positive of frame 1 has a maximum reward, then it will pick negative as well.



9. Inner Loop

- Now the weights are then added with a = step size divide by b = total number of direction in our case it is four and standard deviation of reward which use to update weights with the sum of all noise concerning maximum positive minus negative noise. Figure 9 shows the inner loop of the algorithm.

2.3 changes to the proposal and motivation for the changes

- From the proposed algorithm now, I will demonstrate all the changes I would like to make for better performance of an algorithm.

2.3.1 In place of standard deviation to mean absolute deviation

```
* The data which represents the outliers for fixing outlier issue standard
deviation used but in many cases mean absolute deviation can be used. For example,
if the outliers are high at that time mean absolute deviation outraced standard
deviation, so there may be a chabce which might affect the output.

* For example, we have a dataset like 219,11,14,20,24,32,21  the best method that
can be used here is mean absolute deviation because here 219 is an outlier.
```

2.3.2 More direction

- Instead of exploring positive and negative direction, it can also explore the direction into more ways like multiplication or natural ways. I believe that if the direction is explored in many ways, the high reward getting chance may me more.
- If the random search explores the more direction so the more maximum number may pick and after the process, more episode can generate, and it may obtain a good reward distributing straighty.

2.3.3 In place of Z-score Normalization to Some other normalization technique.

- For data normalization technique, some technique may be suitable then Z-score normalization. For this, I have prefered below research paper in that paper it shows various technique like Decimal Scaling Normalization, Sigmoid Normalization, Median Normalization, Tanh estimators, and Min-Max normalization.

- http://www.mirlabs.org/ijcisim/regular_papers_2014/IJCISIM_24.pdf (http://www.mirlabs.org/ijcisim/regular_papers_2014/IJCISIM_24.pdf)

2.4 Implementation done

- Environment selection
- In-depth analysis of algorit`hm
- Code analysis
- Mathematics behind the algorithm
- Mean absolute deviation in place of standard deviation but testing is remaining

2.5 Tools Used

- python 3.7
- anaconda navigator
- Numpy
- pybullet for environment and physics
- matplotlib or seaborn

2.6 Formal Specification

- GPU or TPU to evaluate rewards

Section 3 - Future work

- For my future work, I have thought to divide the project into three main sections. This work I am going to propose every week. Firstly, I have thought to reimplement an algorithm based on the changes I have prefered.Like changing standard deviation to mean absolute deviation, explore the direction rather than positive and negative. The main advantages of this algorithm may be it gives output for more number of frame and that may outrace current version of algorithm, and by changing the normalization technique in place of using Z-score normalization technique it is good and beneficial to use Tanh estimators and Min-Max for reward maximizing mechanism. Secondly, the testing phase I have thought to use explore my model into multiple environments like a humanoid, ant walk, half cheeta, Walker2D , and HopperBullet. After testing on this model time I will also test my model on games environment which is provided by open ai gyms like Doom, super Mario, and breakout game by the testing model to game environment I may get what my model is capable of doing in real-world scenarios. Finally, I will focus on my research report which is an important task. I will note each output and further, I will study the previous out and may get the hint that has to be done in future. I will also explore the mathematics behind the algorithm in my documentation. Then the ppt is also an essential thing it not only improves the presentation but with it people may get background information easily.