***Mental Causation via Neuroprosthetics Project Documentation – with image dataset***

**KEYWORDS**

Mental State: The state of the mind thinking about an Intention.

Neural State: Pattern of Neural activity.

Neuroprosthetics: study of Establishing interface between brain and Computer.

Multiple Realisation: Same mental state can be achieved through different events

Neuroprosthesis: Device that is embedded in the robotic arm to encode and decode the brain signals. Brain signals are tracked by electrodes implanted at motor cortex region of brain.

1. ***ABSTRACT***

How it will be if an amputee can get a new arm and could use that arm as other normal people do? If it happens, then definitely it can lead to a new life style. It can be possible with mental causation**.** So, we explored many researches and come to know that one way of achieving mental causation is through Neuroprosthetics. An electrode will be implanted on brain which can help in trapping the brain signal. Neuroprosthetic device is used to decode the signal from the electrode or encode the signal from the arm. Our experiment is mainly on these signals in which, identifying the correct meaning of the signals from the brain and outputting the best possible action for the present situation with consideration of the brain signal.

***2. INTRODUCTION***

Mental Causation with the help Neuroprosthetic devices is the idea of this project. Neuroprosthetic devices can work as barrier between brain signals and the robotic arm. This device can encode or decode the signals, so that with the help of this device robotic arm can get the meaning of the signal and performs actions accordingly. To implement the learning model that helps the robotic arm to do actions, we select the Reinforcement Learning with feedback protocol to enhance the efficiency. We can extend it up-to normalized feedback protocol where feedback is stored after normalization, so redundancy experience or overlaps get filtered with normalized feedback protocol resulting best possible efficiency.

***3. PROJECT MANAGEMENT***

***3.1 Problem statement***

To implement a model that can get the neural signal as input and output the best action for that instance. Neural signal is passed in vector form as input to the model and model will output the best action that the robotic arm needs to be followed.

***3.2. project justification***

In this project, feedback protocol is used while building the model which can output action that arm needs to doas output by taking neural signal as input. Feedback protocol deals with the memory which is used to store all experiences that happened till now and uses this memory while making decisions. This can lead to efficient model as feedback is the key to learn fast in linear fashion.

***3.3 strategy***

Reinforcement Learning and Kmeans are the two machine learning techniques used in this project. Reinforcement Learning co-ordinates to include feedback protocol to our model where as Kmeans is to get the quality of the action taken by the arm, this quality of action is the measure of the goodness of action and based on this value our model learns or improves. Another phenomenon called "expansion of cluster boundaries” is also taken into consideration while calculating the quality of the action. As action space is continuous, Deep Deterministic Policy Gradient (DDPG): [1**]** is used to implement RL with feedback protocol.

***4. PROCESS***

***4.1 How it works?***

K-means has used to find the way for learning. Initially there will be some clusters formed based on training data (note: data points in each cluster are representing the actions). If new signal arrives then RL model sends the corresponding action that is needed to take by the robotic arm to the k-means. Then we mix this new action with the previous clusters, say that whole dataset of actions as D. Then we apply K-means on D which leads to formation of new clusters. Later we will collect all the actions which comes into the same cluster into which new action also comes. Then we do the improvised action of those actions. If again a new action arrives, we repeat the same process to find the best doable action for current situation. Thus, upon taking many actions number of clusters also increases or decreases non-linearly because of boundary expansion (4.2).

In DDPG (Deep Deterministic Policy Gradient) we have the Actor-Critic Network (input is neural signal and output is the action that robotic arm needs to take at that instance) to get best suitable action for the current situation. After getting the output, we store all the information of present situation in memory as experience which includes the current neural signal, action taken, next signal, reward, is\_done. After taking fixed steps (commonly said as batch size), we update the policy of the agent i.e., the weights of the Actor-Critic Network are updated based on Gradient policy which can be treated as feedback because these updated weights can help the network to output best action for next time. Therefore, learning through feedback is achieved. Feedback is needed as it can correct the past errors and also improvises the past output. we can use normalized feedback to improve the efficiency a bit more.

***4.2 Boundary Expansion***

* Boundary Expansion can be done in two situations
  + When Number of clusters decreases
  + When Number of clusters increases

**4.2.1 when number of clusters decreases**

No of clusters decreases if new action is similar to actions in two clusters, those the new action causes to merge the two cluster as a single one.



**Fig - 1**

**4.2.2 when number of clusters increases**

 No of clusters increases if the boundary points of cluster are similar to new action. Then the new action itself forms new cluster with boundary points.

**Fig- 2**

* 1. ***Working***

**4.3.1 Reinforcement Learning**

With feedback protocol means model will have the memory of past experiences. Thus, reinforcement Learning concept can help in implementing feedback protocol. First, we have to create clusters based on training data using Kmeans algorithm. Next start DDPG Agent who can update his policy based on feedback. An episode consists of taking some constant number of steps. In each step of **episode**, we pass current neural signal to agent and gets the corresponding action in return from the agent. Then, implement that action on environment and get reward.

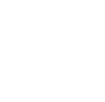
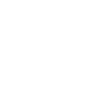
* + 1. **Reward Criteria**
* Reward is distributed for each action based on following criteria
* After applying Kmeans on the D (dataset of all actions including new action), we are giving reward based on output of the Kmeans.
* First of all, apply Kmeans on D, then find the cluster which has the new data point and get the similarity (**Euclidean Distance**) among the data points of that cluster and call that similarity as S.
  + If S<=0.005 then reward = 1 (which means there is improvisation among the similar actions)
  + Else If 0.005<S<=0.05 then reward = S\*5 (less improvisation among actions, so less reward)
  + Else reward = 0

***5. Feedback Protocol***

***5.1 Feedback Protocol Criteria***

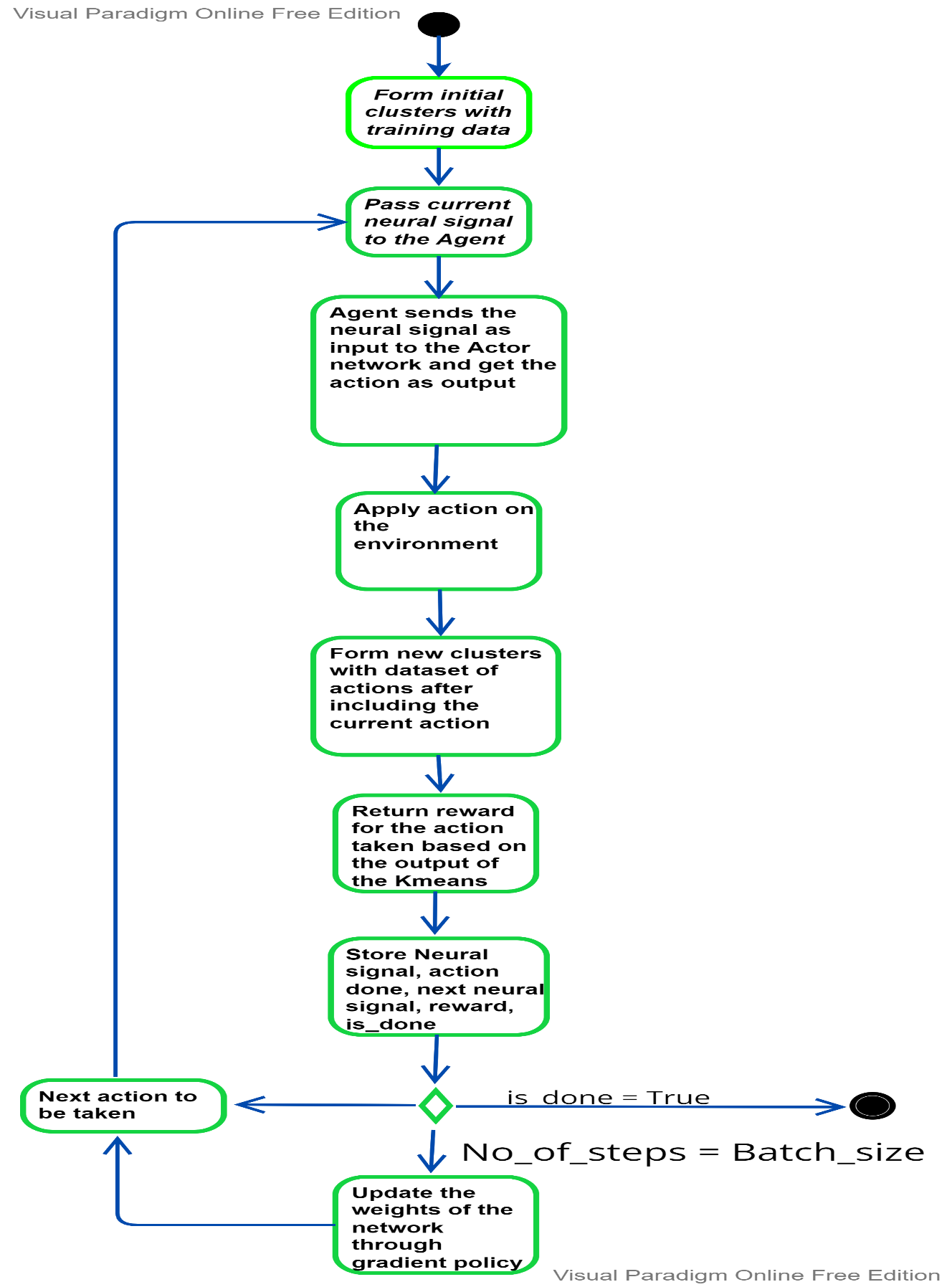
1. Action needs to be dependent on at least one parameter**,** ex: - to predict the weather which depends on temperature, humidity etc.
2. Action space should be upper bounded by maximum value in observation space**.**
3. Need to have the correct number of dimensions in vector while representing Actions, States. (e.g., dimensions of action are equal to the input layer size of the Actor-Critic Network)
4. Need to specify the Observation space (cause) (Continuous), Action space(effect)(continuous).
5. Need to call memory method after every iteration to store all the parameters of that instance into the memory
6. Must have flag variable (short term/working memory) which stores the number of iterations, after batch size number of iterations must call the update method so that agent can update his policy through feedback***.***

***6. WORKFLOW***



***6.1 Overall working of Model***



******

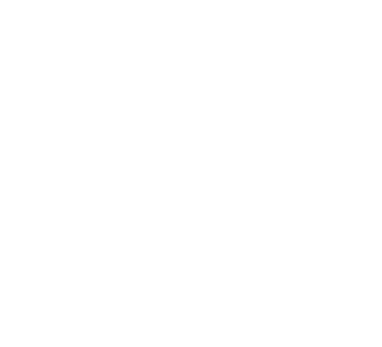
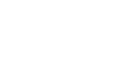
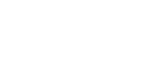
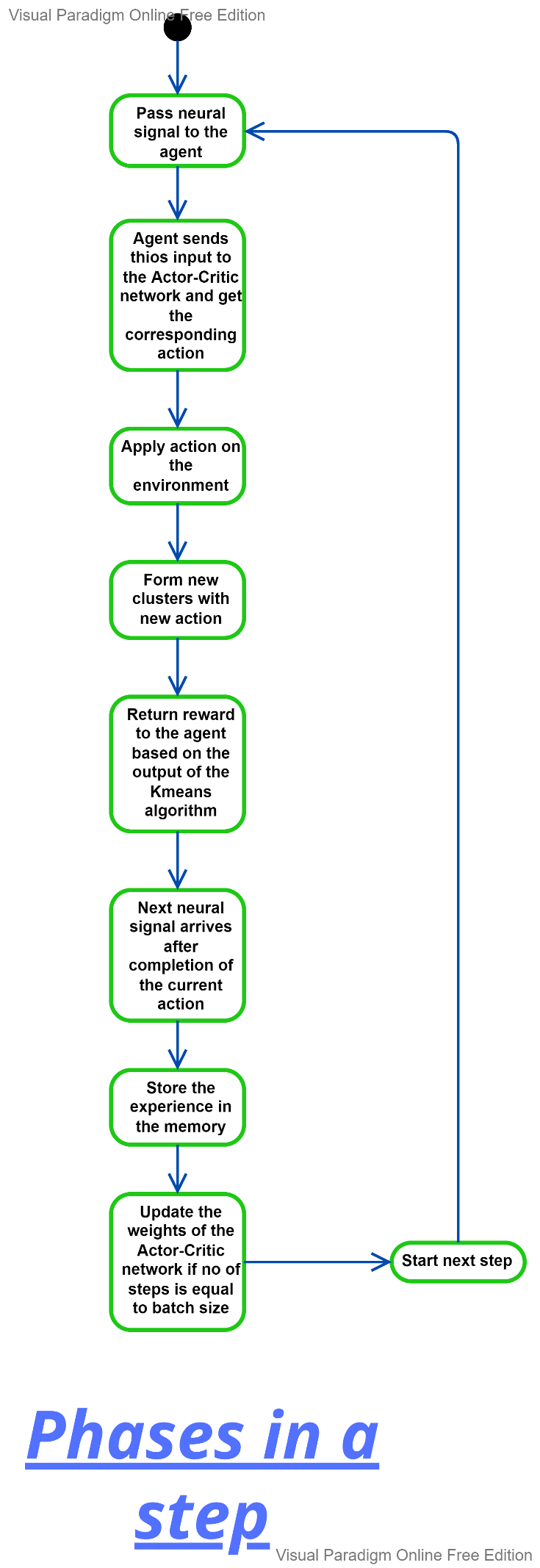


Fig - 3

******



**Fig – 4**

***6. SAMPLING***

***6.1 Sample size***

* Training Dataset size = 50 (Brain Tumour Dataset) to form initial clusters.
* No of Episodes = 10
* Actor Critic Network size: - Input Layer = (1,1), Hidden Layer1 = (1,256), Hidden Layer2 = (256,256), Output Layer = (256,1)

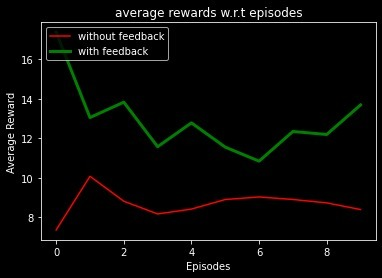


* No of steps in each episode = 50
* Store total reward for each episode in a list.
* Then calculate the average reward for each episode using the following formula**,**

* Store average reward for each episode calculate average reward for each episode by taking total reward of first episode, then taking the total reward of 2 episodes and soon which means calculating the average reward for each episode after every episode**.**

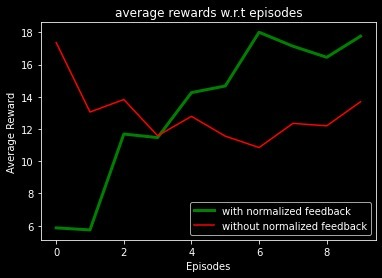
***7. RESULTS***

***7.1 With feedback protocol***

******

**Fig - 5**

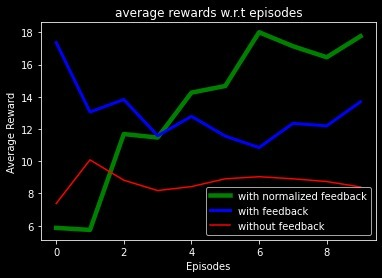
*Fig - 5 is the plot of the learning with feedback protocol vs without feedback protocol.*

***7.2 With Normalised feedback protocol***

**Fig - 6**

*Fig - 6 is the plot of the learning with normalised feedback protocol vs without normalised feedback protocol.*

***7.3 without Feedback vs with feedback vs with normalised feedback***

******

**Fig - 8**

*Fig – 8 is the showing the domination of normalised feedback over without normalised feedback and without feedback.*

***Conclusion***

Thus, having agent who updates based on feedback can let the robotic arm to take the best action than without feedback w.r.t reward. Reward is given by the Environment; good reward means agent is functioning well according to environment which means adapting to environment. So, with feedback learning process speeds up and model adapts to environment early than without feedback as the learning is linear. Also, there is no guarantee that model can explore whole environment if it doesn’t have feedback since random actions are taken if there is no feedback.

***REFERENCES***

*1 https://1509.02971.pdf(arxiv.org)*

*2 https://link.springer.com/article/10.1007/s11229-018-1713-z*