International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-3, September 2019

Modeling of Action's Semantic Memory Incorporated with Procedural and Skill Memory to Perform Tasks

Rahul Shrivastava, Prabhat Kumar, Sudhakar Tripathi

Abstract*: In this paper, a computational model is proposed to mimic an action's semantic, procedural and skill learning's by an abstract modeling of cortical columns of the Neocortex, Basal ganglia and Cerebellum brain region. In proposed work, the action semantic Learning makes a robot capable to learn an action in terms of their body parts movement sequence that allows it to recognize the learnt action by seeing as well. Whereas in procedural, it allows to learn tasks in the form of action's hierarchy and makes it capable to capture the environmental features as a context for action's activations. The skill memory also been added in the proposed work which allows an agent to translate the action as per the current demand of the action. Also, the model has used Vnect model of computer vision to map the human motion into sequence of 3D skeleton of human body, therefore the model can learn by seeing, like humans. In experimental work, the model is tested on vague samples of few actions, where the model is found robust in action recognition task and performed well as per the expectations.

***Keywords*: Action Semantic learning, hierarchical procedural learning, motor skill learning.**

I. INTRODUCTION

In Today's world, from space agencies to military organizations all are looking for robots to perform their complex operations in environment which are not favorable for humans like exploration of distant planets, repairing of satellites, working in nuclear power plants, living in space stations for their maintenance and performing rescue operations etc. Each complex task can be an action sequence, where each action corresponds to some motor level semantics i.e. the sequence of body parts movement, and these actions can also be dependent on some environmental contexts for their activations. A human can perform all these learning's by self, by using their vision and some specific brain regions like Neocortex area that helps in the semantic learning by processing the human vision input [1], Basal ganglia (BG) helps to learn the sequences of actions in context to tasks and environmental features [2,3,4,5], and the cerebellum which is known to be a skill memory meant for the precise movement of the body parts as per the task requirement [6,8,9,10,11,12]. An abstract modeling of functionalities of the mentioned brain regions can lead a robot to perform complex operations and in decision making without any human help.

Although, in past, human cognition of performing tasks has been mimicked to some extent, but they have not followed the biological models of above-mentioned brain regions and their theories, like the SOAR architecture [13], ACT-R [14], SOAR-RL [15], CRAM [16] and the ICARUS [17]. In SOAR and other mentioned models, operators are used to perform procedures or tasks using inferences, but not provided any linkup of actions to the body parts movement (motor level action) and to the environmental contexts which is required to get captured to activate actions. Although, Q learning can be used to learn motor level semantics of any action using obtained reward and the punishment during interaction with an environment, but it requires many iterations for a robot to learn and it is poor to learn the action hierarchies as well.

To fill the above-mentioned research gap of previous models, we have proposed an abstract model for the neocortex, basal ganglia and the cerebellum which is able to learn by seeing the performed actions of others and also able to recognize them by seeing, just like humans. The model has used the Vnect model [7]; a computer vision model that can map the body parts into a 3D space and to track the movement as well. The output is then sent to the proposed model of Neocortex to capture the motor level semantics of an action. The model of Basal ganglia is then used to capture the environmental contexts of actions in few exposures as well as able to learn the action hierarchies corresponding to a task. Furthermore, the abstract model for cerebellum is proposed to execute the motor level action with higher accuracy. This paper will be organized as follows. Section II, describes the proposed computational modeling work. Experimental results are given in Section III. Finally, the conclusions are summarized in Section IV.

II. PROPOSED MODEL

The model proposes a mechanism for robots to learn how to perform tasks, where the agent learns the action semantics and the contexts of the actions by just seeing the videos of actions. Since, the Neocortex is responsible

Manuscript published on 30 September 2019
* Correspondence Author
Rahul Shrivastava*, Department of Computer Science & Engineering, National Institute of Technology, Patna, Bihar, India. Email: rahul.cspg15@nitp.ac.in
Prabhat Kumar, Department of Computer Science & Engineering, National Institute of Technology, Patna, Bihar, India. Email: prabhat@nitp.ac.in
Sudhakar Tripathi, Department of Computer Science & Engineering Rajkiya Engineering College Ambedkar Nagar, UP, India. Email: p.stripathi@gmail.com

for semantic learning's in humans therefore it has been modeled in abstract manner to learn the semantic of actions. Similarly, the BG is responsible for learning the sequences of actions in context of tasks, therefore it has been modeled as sequence learner and context learner as well. The abstract model of Cerebellum is also included for controlling the actions on motor level as per the task requirement as it performs in humans. The proposed framework is shown in the Fig.1.

In the model, the Neocortex is presented as a cortical column of neurons to learn the semantic of any action as shown in HTM [1]. The Neocortex is subdivided into five parts, first is the joint layer which takes input directly from the Vnect, which takes the joint angle values corresponding to each body joint, frame wise. Second, is the action layer which associates the different joints into a single pose and then it learns the temporal relation of movement. Third is the ES (environmental states), which is having neurons that can either be activated or inactivated, the activation pattern of all neurons combinedly represent a state, where the state can be get associated with the current supervised input i.e. actions, so that the action can be recalled corresponding to currently represented state. Fourth, is the object body interaction system, which carries several nodes whose activation is decided on the basis of fulfillment of associated condition which are related to the object body interaction, which is explained in brief in the OBI system. Since, on the task level, each task learns the actions along with the coactive object, therefore, a RO (recognized object) system is also present which recognizes the object, that is essential for the object dependent action learning.

2.1. Neocortex and the Lower level action learning

An action is a sequence of several human poses or gestures, under which an agent can change their poses with respect to time. Since, every action can be done in various different manner which makes the action recognition a difficult task, for example: the action "Hi" can be done while you are sitting, while you are walking or standing, in all these different manners, identifying the which body part movement in a particular sequence is just suffice for the action "Hi" is a big difficulty. Therefore, an action semantic memory in the model has been proposed which is inspired from the HTM (neocortex model) to learn only the lower level action (i.e. an action which is not further divisible). The neocortex based HTM model is already discussed in the preliminary section. In action semantic memory, the neocortex model has been presented with so many modifications as shown in the Fig. 1, the layer is divided into a joint layer and the lower level action layer. The joint layer is subdivided into the neuron columns, where each column is corresponding to each key joint of the body, and each neuron of the column represents the state of the joint as shown in the Fig. 2.

Learning the semantics of lower level action (LLA), is divided into three steps, which are explained below:

i) Finding the activation pattern corresponding to the pose or gesture:

The Vnect gives the joint values of all body joints (body joints are shown in the Fig. 7), frame wise. The joint value can lie from 0 to 1, 1 means that the joint is fully open to its maximum limit (for hand elbow at 180 degree the joint value will be 1, and at the angle of 30 degree the joint value will be 0). At the moment of each new frame input from the Vnect, the Joint layer first compares the previous joint values with the current joint values, and if the joint value changes in two consecutive frames above a threshold value then only the joint value will allowed to be learnt (means that the joint is actually indulged in an action). The joint layer is having a joint column corresponding to each joint (shown in Fig. 2), where each column is subdivided into the joint state nodes, each state node is anchored by one of the possible joint value. In the initial, all joint column will be empty, as the input starts to come, it creates the joint state nodes whose value will be initialize with the current joint value, and if already joint state node is present then it will check for the match with the existing joint state nodes of the column, according to eq. 1, where the difference of the stored joint value and the current joint value of the joint state node is to be taken, if the difference comes with in the threshold limit or the pose match threshold (β) value then the matched joint state node will be activated, otherwise a new joint state node will be created whose value will be initialize with the current joint value.

ii)

For example: let say joint column1, which is corresponding to the joint1, each node of the column is associated with a joint state value, joint 1= [1, 0.7, 0.5, 0.3,0.1]. If the current state of the joint 1 is 0.45 and β is 0.09, then the third node of the set joint column1 will get activated, and other will remain deactivated, because the third one is associated with the 0.5 which is closer with the within the limit of third node of the joint column. This is how the corresponding to current pose or gesture, an activation pattern will be obtained in terms of the nodes of the joint columns.

$$\text{IF} \quad | j_{p,q}^{stored\_js} - j_{q}^{current\_js} | < \beta \qquad (1)$$

$$\text{Then} \quad j_{p,q}^{act} = 1$$

Where, the $j_{p,q}^{stored\_js}$ is the stored joint value of the p^{th} state of the q^{th} joint, $j_{q}^{current\_js}$ is the current joint value of the q^{th} joint, $j_{p,q}^{act}$ is the activation of the p^{th} state of joint q, and β is the pose match threshold.

d.

The optimum value of pose match threshold is very much required, it must be balanced, because, if the value is smaller, then, it would not allow any noisy action to recognize, and if the value is much higher then it will recognize each action falsely true.

Learning a pose as coactive joint state node:

To learn a pose, as a coactive joint state node an action layer is there in the neocortex. The neocortex in our model is meant only for the lower level action, therefore the action is divided into the action columns, where each column is divided into the temporal nodes.

Each temporal node is corresponding to a time in the action. The activated temporal node captures a pose or can say the activation pattern of joint nodes which is generated at any particular time during an action. Capturing a pose is making of associations between the currently activated temporal node of the action column and the currently activated joint state nodes. In the very first exposition of an action every K^{th} frame will only be given as input to the model for learning.

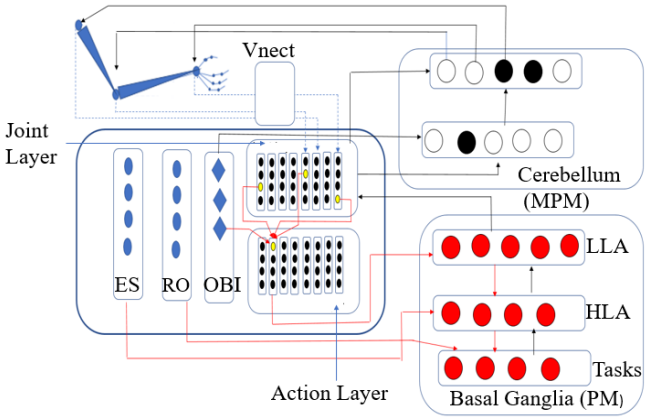


Fig.1 Proposed Model for Neocortex, Basal Ganglia and the Cerebellum interaction System

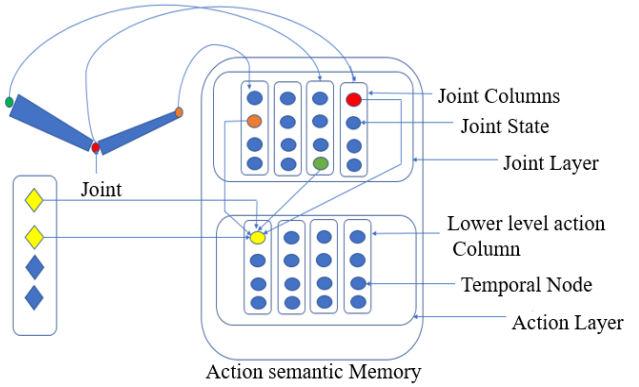


Fig. 2. Lower level Action learning using Joint and the Action Layer.

Let say, an Action_column1is corresponding to LLA1 which is having ten temporal nodes (t^1, t^2….t^{10}), then its means that the LLA1 can learn 10 different poses, where each node learn the pose generated in its corresponding time step, where t^1 gets associate with the very first pose generated while action, and t^{10} will gets associate with last pose generated in the action. The association between the temporal nodes of the action column and the joint state nodes of the joint column is a weight learning between the two, where the weights changes according to the eq. 2 which is a Hebbian weight learning. The weight learning equation is having two terms, first terms says the weight increases when both nodes gets activated, and the second term says that the weight decreases when the post nodes activates (temporal node) and the pre-node (joint state node). Here the temporal node will be considered of the action column which is corresponding to the supervised LLA.

$$\Delta w_{t,j}^{s,i} = ((1-\Delta w_{t,j}^{s,i}) * act^{s,i} * act^{t,j} * lr) - (\Delta w_{t,j}^{s,i} * (1-act^{s,i}) * act^{t,j} * fr) \quad (2)$$

Where, $\Delta w_{t,j}^{s,i}$ is the weight between the t^{th} temporal node of the j^{th} LLA column, and the s^{th} joint state of the i^{th} joint column of the joint layer, lr is the learning which is equals to 0.3, and fr is the forgetting rate which is equals to 1.

iii) Frame Matching while learning LLA:

The previous step will be followed only in the first exposition of an action, because here, every kth frame has been taken as input to learn. But in further expositions, there is a requirement of finding the best match frame with the currently activated temporal node or the previous associated learnings with the temporal node which is shown in the Fig. 3(b). Instead of matching with all frames, here a window of 'j' frames has been taken for match, where a match score will be calculated for each frame of the window according to eq. 3. If case, if any i^{th} frame matches with the currently activated temporal node then the window will shift to the next of the matched frame, i.e. i+1 and then association will form similar to the previous step, according to the eq. 2. If none of the frame matches within the limit of threshold then window will shift to the j frames, i.e. the window size. After association with the current activated temporal node, the next in sequence temporal node will get activated and similar process will be followed with the next activated window.

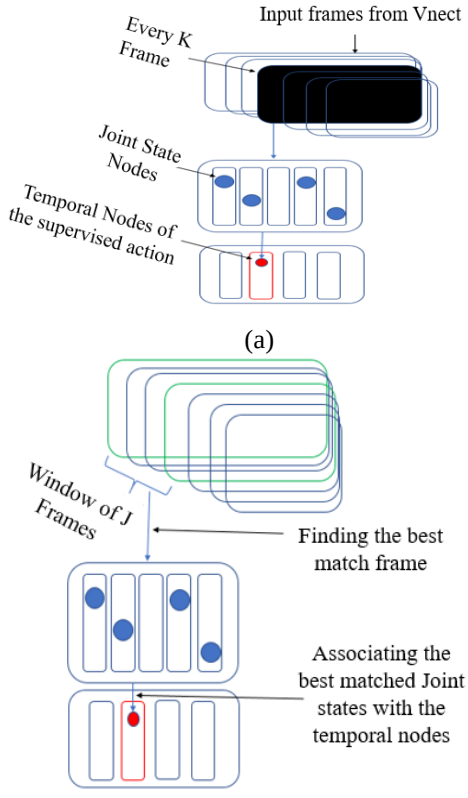


Fig. 3: Frame matching

$$M(T_l^t, W^f) = \frac{\sum w(T_l^t, S_j^i) * (1-(j^f - S_j^i))}{\sum w(T_l^t, :)} \quad (3)$$

Where, $M(T_l^t, W^f)$ is the match score of f[th] frame of Window 'W' with the t[th] temporal node of the l[th] action column.

If $M(T_l^t, W^f)$ is the highest of all frames of the activated window W, and score is above a threshold value then the frame f will be the selected for learning.

If $(1-(j^f - S_j^i)) * w_{t,l}^{i,j} > \beta 1$ then form association between the t[th] temporal node of l[th] column, and i[th] joint state node of j[th] column according to eq. 2.

2.2. Object body interaction system:

Movement of body parts during an action not only depends on the action semantics, if the agent is dealing with an object. For example: In action "pick bottle" by follow just semantics, a humanoid can't perform this action until the agent does not learn the motivation of action and their semantics in terms of object interactions. Therefore, an OBI system is proposed in the neocortex whose task is to identify the motivation of an action just by seeing.

The OBI system is shown in the Fig. 4, which consists of several OBI nodes, where each node is associated with a predefined condition. Each node can have only two states, either it will be ON when the associated condition is true, or OFF when the associated condition is false.

In Fig. 4, the associated condition with the node 1, tells that the node will get activated only when the concerned object comes within the reach of agent. Node 2, get activated only when the agent is not in the hand reach of the concerned object. Node 3 get activated when the concerned object is in the hand touch of the agent. Node 4 get activated when the hand is above the object, like this, several other OBI nodes can be created, which tells the information of interaction with an object.

As we know, each object related action is always be associated with some motivations that triggers the agent to perform action. For example, in action "navigation towards the object", requires the object must be out of hand reach, otherwise a human would prefer to move their hands to pick the object. But how the robot will get to know about what to do in what situations. The obvious solution is the OBI nodes, if robot founds that the some OBI nodes get deactivated in the end of action which was previously activated in the initial of action, it shows that the deactivating the node is the motivation of the action or can say stopping condition for the action (means that the motivation of the action is completed, and the action is no longer required).

Similarly, for the action "touch object" the starting condition will be the node 3 (means that the object is in the hand reach) and the stopping condition will be the node 4 (means that the distance of the object and the palm is zero). This is how the agent will get to know about the starting and the stopping conditions without any explicit learning.

Next the usages of OBI nodes in the LLA layer nodes of the BG, temporal nodes of the action layer of neocortex and in the cerebellum, are described.

1.2.1 Association formation between OBI and the LLA layer nodes of BG:

During learning, the OBI nodes makes associations with the coactive LLA layer nodes of the BG, the association tells the starting and the stopping condition for the LLA layer nodes.

For example, The behavior of the OBI node, in which, the node in the initial of action is activated and at the final of action is deactivated, gets associated with the positive weights (node 1will become the starting condition for the LLA), and the node which is deactivated in the initial of action and activated at the final of action will gets associated with the negative weights (node 2 will become the stopping condition for the action) with the LLA node of the BG. These associations help in choosing the LLA node during performing any task, stops the previous chosen LLA on achieving the target and helps to skip some LLA from doing (when the target is already achieved).

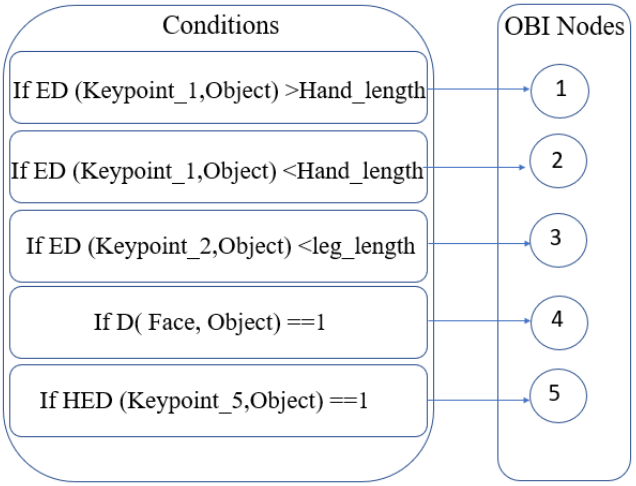


Fig. 4. Object Body Interaction system

1.2.2 Association of OBI nodes with the temporal nodes of the neocortex for action recognition:

Previously, in the object free actions, the temporal nodes are used to learn the semantics of an action, which helps in the recognition and as well as in performing the action. But in case of object related action, the temporal nodes of the action column learn the OBI nodes to recognize the action only. To perform the action, cerebellum is used, which is explained in below in the next section.

The OBI nodes which changes their states get associate with the current activated temporal node of the action column which is corresponding to the supervised action. Here, only two temporal nodes are required, one to represent the state before the action, another one is to represent the state which is generated as a result of action. The OBI node which turns inactive to active get associate with the first temporal node, and the OBI node which turns inactive to active get associates with the second temporal node.

In the case of action " movement towards object", the node 1 and node 2 both have changed their states during the action, therefore both nodes gets associates with the temporal nodes by a weighted synapses, where node1 will be associated with the temporal node 1 and node 2 with the temporal node 2 of the action column corresponding the action " movement towards object".

Retrieval Number C4227098319/19©BEIESP
DOI: 10.35940/ijrte.C4227.098319
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

2.3 Cerebellum and the Object related action learning:

In object dependent actions, joint states derive from the current state (location w.r.t body key points) of the considered object. Since, the object position can be different while doing learning an action in different iterations, therefore, joint states will not get repeated and associations will not be formed between the temporal and the joint nodes. In such case, an action semantic, will get learnt by the cerebellum as shown in Fig.5.

In cerebellum system, along with the joint angles, the body key point distances from the concerned object has been taken as input to predict the correct joint angle values as output so that the key point can reach to the desired goal location. Input to the cerebellum are joint states and the key point distances from the concerned object, which is generated before the start of an action. The joints states, which are generated after the action, have been taken as a supervised input to the cerebellum to evaluate the error for learning using deep neural network.

The distances of different key points from the object (Euclidean distance) is calculated w.r.t all three dimensions (x, y and the z axis). Instead of all key points, only those key points have been considered which changes the most between the frames, similar with joint angle values. The object body interaction (OBI) system (shown in the Fig. 4) take care of all the key points and their distances with the concerned object. Here, a three layered deep Neural network is used to generate the correct motor output values or the joint states corresponding to the input joint states and the distances of the key points from the object. The error to the final layer is the difference between the supervised input and the current output generated by the network.

$$E^{ol} = Correct\_Output\ (js^1, js^2, \ldots js^n) - Network\_Output\ (js^1, js^2, \ldots js^n) \quad (4)$$

$$\Delta^{j+1} = \left(H_{input}^{J+1} > 0\right) * E^{j+1} \quad (5)$$

$$E^{j} = W^{j,j+1} * \Delta^{j+1} \quad (6)$$

$$W^{j,j+1} = lr * W^{j,j+1} * \Delta^{j+1} * O^{j} \quad (7)$$

Where, E^{ol} is the error of output layer, E^{j} is the error of the j^{th} hidden layer, Js is the joint state, O^{j} is the output of the j^{th} hidden layer, I^{j} is the input to j^{th} the hidden layer, Δ^{j+1} is the gradient of $(j+1)^{th}$ hidden layer and *lr* is the learning rate.

The values which the cerebellum takes lie between the [0 ,1]. The joint state value is already explained in the step 1, but the joint which does not changed during the action will be set to -1, here -1 represents that the joint is in turn off state during the learning (in Fig. 4, it is shown by the yellow color nodes). The distance of a key point from the object will be -1 when the object is not in the reach of the key point (key point can't reach to object by changing the states of the joints which are in the ON state) or when the key point is not moving, otherwise it will become set to between 0 and 1, where the value of 1 represents the maximum distance from the object.

During learning an action this could be happen that some joints have changed their state but they are not relevant to the task, now how would the agent would know which joint is irrelevant. The answer is OBI system which is explained in above section.

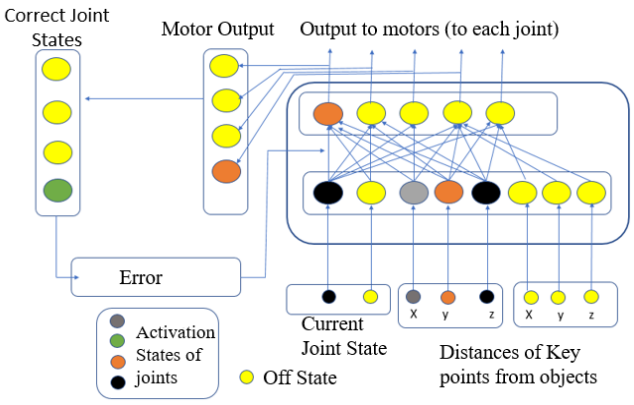


Fig. 5. Learning in Cerebellum

2.4 Learning a Higher-level action in Basal Ganglia:

In the abstract model of BG, it is subdivided into three layers, task layers, low level action (LLA) layer and the high-level action (HLA) layer. The low level actions are those which are not further divisible into actions like move, touch, grasp, kick etc., on the other hand higher level are those which are further divisible into a sequence of several lower level actions like the action "fetch bottle" is further divisible into to a sequence of "move toward bottle" → "touch the bottle" →"grasp the bottle" → "move back to the previous location". These higher-level actions are the part of the basal ganglia, in which the BG learns the sequence of both the higher level and the lower actions nodes. The task layer contains the different task node like "making tea', cooking food" etc. The task layer nodes learn the sequence of HLA nodes which are activated during the task. Since the HLA can be an object dependent action, like "fetch bottle", so the task layer nodes learn the object sequence too which are activated in sequence during the task.

Since, the HLA consists of LLA sequence and Tasks consists of the HLA sequence, therefore the neocortex and the cerebellum are needed to be exposed to both LLA and HLA before learning the tasks, and exposed to LLA before learning of HLA.

The learning hierarchy is shown in the Fig. 5. While learning an HLA, the HLA layer node corresponding to the supervised HLA, learns the activation values of each recognized LLA's. The activation value represents the position of LLA in the sequence. The activation values of previously recognized LLA always gets updated at the moment of current LLA recognition. At the moment of recognition, the activation value of the recognized LLA will be set to 1, which then starts decay with every new recognition, according to eq. 8. Therefore, the LLA which recognized earlier in sequence would decay more and its activation value will be the of all LLA in sequence, and vice versa. Finally, at the end of the HLA, the HLA node store the current activation pattern of the LLA, in the form of weights. If the weighted connection value is 0 means that the LLA is not present in the HLA.

$$act^{l} = act^{l} * (1-\alpha) \qquad (8)$$

Where, act^{l} is the activation of l^{th} LLA node and $\alpha \in (0, 1)$ is the decaying factor.

In figure 6, the HLA "fetch" is learnt as a sequence of LLA "touch", "grasp", "move_handup" and "move back to goal location". In HLA "fetch", the LLA which is having the highest activation represent that the LLA will come in the last of the sequence, the LLA which is having the lowest activation represent that the LLA will come in the beginning of the sequence.

Similarly, the tasks are also learnt as sequence of HLA, the HLA which is having the highest activation in any task, represent that the HLA comes in the last of the task, but here HLA's activation values are not enough to define tasks, concerned object is also need to be defined along with the action, that is why the task nodes in the model has leant activation value of coactive object and HLA. The activation value of objects too decay as similar to HLA.

In the Fig. 6, the task "making tea" is shown, where the task node has learnt the activations of both the activated object and the HLA. The problem of this learning that the repetition of object and the HLA is not allowed, however, to overcome this problem the model used the multiple copy of the object and the HLA on repetition, as shown in the Fig. 6, where multiple instances of fetch in HLA layer, 'Moveto_goallocation' are created in LLA layer.

2.5 Lower level action recognition and their retrieval:

In action recognition, the match score has been calculated corresponding to each LLA layer node of the basal ganglia. The node which is having the highest match score of all and above the threshold will be the recognized action. The recognition starts with the generation of activation pattern of the joint states as explained in section 3.1. If the action is not an object related then recognition will be dependent on the score of temporal nodes which derives from the joint state node activation pattern, otherwise the score will be derived from the OBI activation pattern.

During recognition, in object free action, different joint state nodes get activated, as per eq.1, and then an activation pattern will be obtained corresponding to the current pose or the input frame. In recognition too, a window of input frames is used to anchor the current activated temporal node (at the initial, the top most temporal node of the column will be activated) with the best matched frame, so that the window can be shifted to find the best match frame with the next activated temporal node in the window. If the match score of temporal node with the best matched frame of the window (calculated according to eq. 9 given in the algorithm 1) does not crosses the threshold score (i.e.β3) then the current temporal node will not get anchored in the window and it will remain active to find the best match frame in the next shifted window. SoftMax function is used to calculate a confidence score corresponding to each action, whenever, confidence score of any action crosses the threshold value (i.e. β4= 0.7) then the action will be declared as recognized action.

The recognition process of LLA is shown in the Algorithm 1, where firstly the current temporal node of each column is initialized to 1, and then the set current window for each action column. Next a while loop will run until we get a winner. In each iteration of loop, corresponding to current window a score is calculated for the current activated temporal node of the action, which is match score with the highly matched frame of the current window of the corresponding action. The match score of the temporal node will be added to the overall score of the action, which then passed to SoftMax activation function to calculate the confidence score of each action, if any of the action cross the threshold value (β4) then the action will be chosen as a winner and the recognized action.

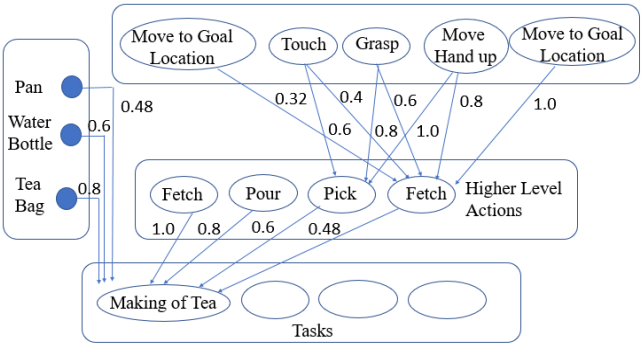


Fig. 6. Action Hierarchy System and their sequence learning

Algorithm 1: LLA Recognition

Initialize the current activated temporal node array to 1, length of the array equal to the number of actions. Initialize the current window W, corresponding to each action. In all action column, 1st temporal node will be activated to find the best match frame in window

While (winner = = -1 OR current window size of all action = = 0)

For all action l

For each frame of current window

$$M(T_l^t, W^f) = \frac{\sum w(T_l^t, S_j^i) * (1-(j^f - S_j^i)}{\sum w(T_l^t, :)}$$

End

If $Max\ (T_l^t, W_l) > \beta 3$

Current activated temporal node $t^l = t^l + 1$

Score(l) = Score (l) + $Max\ (T_l^t, W_l)$

End

End

Confidence_Score(l) =Exp (score(l) / sum (Exp(score))

If Confidence_Score(l) > β4

Winner =l

End

End

Where, $M(T_l^t, W^f)$ is the match score of current activated t^{th} temporal node of l^{th} action with the f^{th} frame of current window of action l, $\sum w(T_l^t, :)$ is the summation of weights belongs to the t^{th} temporal node of l^{th} action column.

In case the action is object related action their recognition would depends on the OBI nodes, if any OBI node changes their state while action then the input from the OBI node will treated as 1 otherwise it will be treated as '0', and the temporal node activation will be similar as calculated above.

$$Act_j^t = \frac{\sum_{k=1}^{k=r} W(i,j,k) * OBI^K}{\sum_{k=1}^{k=m*n} W(i,j,k)} \quad (10)$$

Where, the OBI^k is the input from the k[th] OBI node, r is the number of OBI node, Act_j^t is the activation of the t[th] temporal node of the j[th] action column and W^{obi} is the weight matrix between the temporal node and the OBI node.
During recognition object dependent action, at the moment, when an OBI node changes their state, the input from the OBI node to the temporal node will be considered as high otherwise low. Each temporal node calculates their score using the associative weights and the input coming from the OBI nodes (high/low), according to eq.11. For recognition, corresponding to each LLA, each temporal node should be matched above the threshold limit (β) If condition satisfies then score of the action will be calculated according to eq. 11. The highest scorer will be the recognized LLA.

$$\forall t \qquad S_l^t = \frac{\sum input^y * OBI(l,t,y)}{\sum OBI(l,t,:)} > \beta$$

$$Score(l) = \sum S_l^t \quad (11)$$

Recognized Action (LLA) = max (Score)

Where, $\sum S_l^t$ is the activation of t[th] temporal node of action column l, score(l) is the confidence score of l[th] LLA.
The retrieval of LLA, require the recalling of joint states which are associated with temporal nodes of the action column, where the action column is corresponding to the recalled action. The recalling order of the temporal nodes will be top to bottom. But if the action is object related then the recalling would require cerebellum and the current OBI activations.

2.6 Higher level action detection and their retrieval:

In recognition, a score will be calculated corresponding to each HLA, the highest scored HLA will be the recognized HLA. Since, HLA is a sequence of LLA's, therefore its recognition requires the recognition of its constituent LLA. After each recognition of LLA corresponding to the input frames, the activation value of the current activated LLA will be set to 1, and the activation values of previous activated recognition of LLA decays as per the eq7, as result of that a new activation pattern always generated with every new recognition of LLA. Each HLA node has already learnt the LLA ordered sequence in the form of their activation values. Therefore, during recognition, corresponding to each HLA node, a score (*MatchScore*) has been calculated. The *MatchScore* is summation of differences of all associated activation values of LLA nodes with their corresponding current activation values, as calculated in eq. 12. If the *MatchScore* of any of the HLA node is highest of all and higher than the threshold value (β5) then the HLA will be declared as a winner or the recognized HLA.

$$MatchScore_j^t = \frac{\sum_{i=1}^{i=|I|} (1 - |act_i^t - act_i^j|)}{N^j} \quad (12)$$

Winner action= $MatchScore_j^t$ >β5 and max ($MatchScore_j^t$)

where $MatchScore_j^t$ the Match Score of a j[th] HLA at t[th] time step, l is the number of LLA's, act_i^t is the activation of l[th] LLA at t[th] time step,
There can be chances of error in the LLA recognition, which some LLA's can be skipped or not be recognized. As a result of that the recognition of HLA will get affected, and the HLA which are lengthy in sequence will be get affected more. So, here the threshold value of the HLA match threshold will be dependent on the Sequence length, and which can be calculated as (1- α) * N, where the α is the decay factor and N is the number of allowed skipped LLA's (LLA which are not recognized).
The recalling an HLA is just recalling the LLA in the ordered sequence, where the LLA which is having the lowest activation in the HLA will be recalled first and so on.

2.7 Task recognition and their recalling:

Since, using different objects in same actions will be considered as different tasks, therefore it is very much required to learn a task in the sequence of HLA along with the coactive object (i.e. object used while performing HLA) as shown in the Fig. 5, where the task of making tea is learnt as a sequence of HLA and the used object in the corresponding HLA of the sequence. The task layer of the BG, is having several nodes corresponding to each unique task, where the weights of the task node with the HLA nodes of the BG and the RO (recognized object) layer node of the neocortex, represent the order of the HLA and corresponding object in the sequence.
Similar to HLA recognition, a score is calculated corresponding to each task node, which has been considered the current activation of HLA's (calculated in eq. 12) and objects used in the corresponding HLA.
The algorithm for the task recognition is shown below. The algorithm starts just after the time step of last task recognition. Since, each task node is associated with the activation values of HLA's and objects as ordered sequence value, therefore, during recalling, each task node compares their associated values with the current activation pattern to calculate score. With each new recognition of HLA and the object, set the activation values of current HLA and the object to 1, and decay the activations of all other HLA and the object according to eq. 7. Now, corresponding to each task node, calculated a match between the associated activation value of an individual HLA node in the task node with their current activation value, according to eq. 13, similarly for the object nodes. According to eq. 14.

$$matchvalue\_HLA_j^i = 1 - D_1 \quad (13)$$

$$D_1 = | current\_act\_HLA^{j} - asscociated\_act\_HLA_i^{j} |$$

$$matchvalue\_object_j^{i} = 1 - D_2 \quad (14)$$

$$D_2 = | current\_act\_object^{p} - associated\_act\_object_i^{p} |$$

Where, $current\_act\_HLA^{j}$ is the current activation value of j[th] HLA node, $asscociated\_act\_HLA_i^{j}$ is the associated activation value of j[th] HLA node with the i[th] task node. $current\_act\_object^{p}$ is the current activation value of p[th] object node, and $associated\_act\_object_i^{p}$ is the associated activation value of p[th] object node with the i[th] task node.

If the match value of any individual HLA node equals to the match value of any of the object node for a particular task node, then it means that the task node has already learnt the given combination of the HLA and the object node. Next, a total score of the task node is calculated (TS^{i}), which is a summation of calculated match value for each matched combination in the task node as calculated in eq. 15.

$$TS^{i} = \sum_{k=1}^{k=M} Matchvalue\_HLA_j^{i} \quad (15)$$

Where, M is the number of combinations of HLA and the objects which are already learnt by the i[th] task node, and currently recognized in the current input pattern.

$$MS^{i} = \frac{TS^{i}}{N} \quad (16)$$

If $MS^{i} > \beta 6$ && MS^{i} = max (MS)

Then winner OR recognized task = i

Where, MS^{i} is the match score of the i[th] task node, and N is the number of nonzero associated activations with the task node. After calculation of total score of the task node, a match score of the task node has been calculated which is a ratio of total score to the number of non-zero associated activation values of HLA nodes with the task node as calculated in eq.16. Match Score value of the task node represents how closer the current activation sequence with the learnt activation sequence. A task node which is having the highest match score value of all, and the match score value crosses the threshold value (β6) then the task node will be declared as a winner and the recognized task. This iterative process will run until a winner is selected. After the recognition of task, score of all task node reset to 0. But whenever anyone performs a learnt task by follow the different order of actions from the learnt order sequence then the model won't recognize the task. For example: if anyone performs the task of "making a cup of tea", in which the action Fetch a pan comes first then pour water into the, and then add sugar, tea and in the last put gas on. This similar task can be done by follow different action sequence, like first fetch a pan, then put pan on gas, put gas on, then add milk, sugar and tea bag, in such cases the model falsely says negative to the action.

2.8 Action Translation by the Cerebellum:

The action translator is used to translate the stored poses according to the current working parameters. The cerebellum works in two ways, first is used to generate the motor output directly on the basis of current motor parameters distances and the motivations, where the joint information can't be stored in the joint columns or where the capturing the semantic into the joint columns is not possible. Second is the way of translating the stored frames, where each temporal node of the action column of the corresponding action will be recalled in sequence , and at each recalling of temporal node, it checks weather the current action move will follow the motivation of the action or not, if it follows then it allow to continue the action otherwise it translates the one of the parameter which can maximize the motivation. For example: the action 'movement toward object' requires the previous distance from the object is to minimize with each step of the movement which is already been captured by the cerebellum using the object interaction system. Suppose the agent in other direction of object, and it recalls the joint states but it moves which lead them in another direction, but the cerebellum will project the movement in all direction and see which direction will be the best to follow the motivation in this way the cerebellum works.

III. IMPLEMENTATION AND RESULTS:

For simulation, humanoid has been designed in the MATLAB 2019 as shown in Fig. 7, which consists of 20 joints, each joint is having own range of movement same as human. Model is validated on the basis of given LLA's and HLA's. In LLA's recognition, the accuracy of the model is a ratio of the number of the correct recognition to the total number of testing action, and in case of performing the LLA, the error of the model is an average of differences between the desired and the real location of body key points. In case of HLA's recognition, error is the ratio of the number of correct predictions to the total number of testing sample. The frame rate of Vnect is 33 frame per second, learning all frame can increase the space and the time complexity that is why some frames are skipped to process. The frames in which movement has not been take place are skipped to store and about only ten frame has been considered to store their joint states, and after receiving each action the cerebellum learns only the object body interaction movement corresponding to the current parameters like joints states, distances and the goal states which it can identify by OBI nodes, In each action learning, cerebellum uses 1000 epochs to converge the weights so that the correct output it can generate.

3.1 LLA recognition and the pose match threshold:

Since, the model has used a wide window to match the frames, so that the speed mismatch problem can be overcome. Next, the accuracy of the model has been evaluated at different pose match threshold value. At the lower match threshold value (β), the joint value of currently exposed pose to the model does not match with in the threshold limit (β),

as a result of that the activation value of the temporal nodes gets affected and the overall score of the correct action will decrease, along with the incorrect actions. But here the model does not set any threshold limit on the score of LLA to be get recognized. In case if the pose match threshold value increases then the score of correct action along with the incorrect action can be increased. Here, simply the highest scorer LLA will be the recognized action therefore the threshold value does not affect the recognized LLA very much, but still it must be balanced so that a difference can be maintain between the score of the correct and the incorrect actions. The match score results are shown with respect to the different pose match threshold, as shown in the Fig. 8.

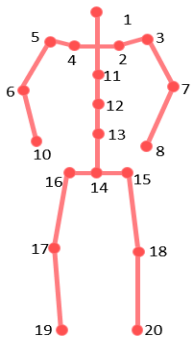


Fig. 7. Simulated Humanoid

3.2 HLA recognition and the HLA match threshold:

The Recognition of HLA is also a one of the key phases in the task recognition, where the recognition of HLA gets affected by the HLA match threshold value and the sequence length of the HLA. Since, some lower level action may be skipped from the recognition because of the error in the LLA, then the HLL must have optimal HLA match threshold value so that, even at higher error rate in the LLA, the HLA can be get truly recognized.

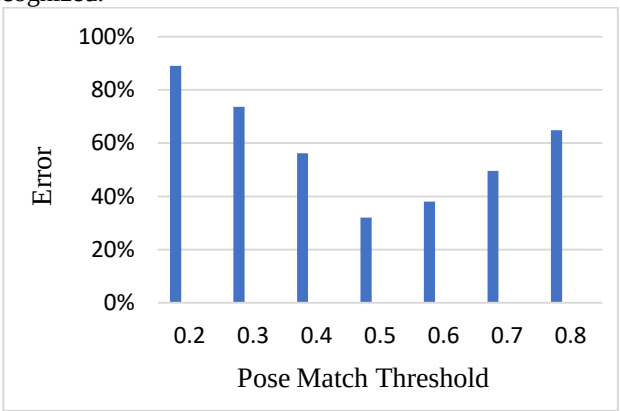


Fig. 8: Lower Level Action recognition accuracy with respect to different pose match threshold values

The recognition accuracy of HLA with respect to different threshold values and different action length (HLA match threshold) is shown by the help of line graph in Fig. 9. Here, the noise has been added up with the original learnt actions where the randomly LLA are skipped from the learnt sequences. The line graph shows, due to the skipping of random LLA's, the difference between the stored and the current activations has increased. In the lengthy sequence HLA, the average gap will be higher between the activations, therefore the recognition accuracy will be lower in the lengthy sequence HLA and it can be increase with the increase in the threshold value. The graph validates the proposed formula for the HLA match threshold (given in step 6 of the ASM learning), as the formula calculates the approx. the same value for the threshold limit at which the given action has outputted higher accuracy.

3.3 Action Speed and the Accuracy:

The action speed is also play one of the important roles in LLA recognition, as the action speed can generate a contrast between two similar actions like "slapping" and "touching chick" that is why it is important feature to consider. Since, we consider only the K^{th} frame of the input to learn, in case of fast action, so many poses can be skipped, therefore the frame rate must be dynamic based on the action speed. An action speed is a measurable in the terms of the change in the joint values per frame (only one joint will be considered which have the maximum change). If the change is higher, then the value of K must be decrease and Vice versa.

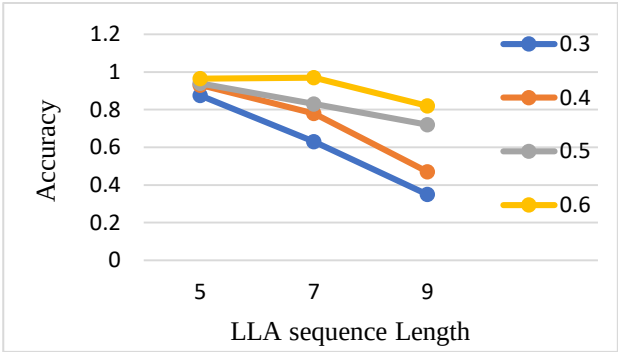


Fig. 9. HLA recognition accuracy with respect to different action length and the HLA match threshold

The standard frame rate for the perfect animation is 25 frames per second, since the frame rate of VNect is 33 frame per second, so that it can capture at least 5-6 poses if we takes the k value of 6, but in case of slower action, the joint value changes very slowly as result of this at same value of K it will generate several poses and a fast version of the similar action may not get recognized, but if we changes the frame rate based on the action speed then the number of poses will be similar and the fast action will also get recognized as slow action and vice versa. Therefore, the action the frame rate must be a constant for all action if we want to train the contrast in speed of actions, otherwise the dynamic frame rate will give higher accuracy. But if we present the fast and slow version of an action as two different action then definitely the action.

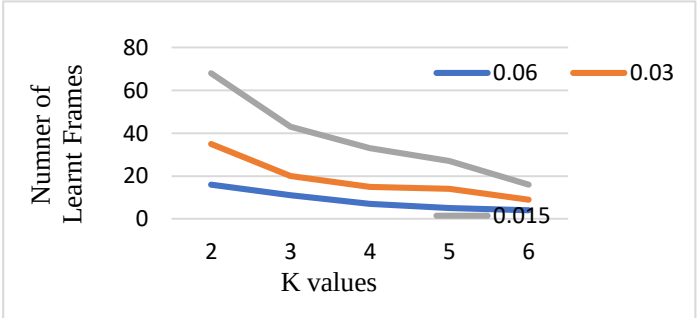


Fig. 10. Average number of frames generated at three different average speed of the action.

The results of the LLA recognition at different speed and the K value is observed and plotted in the Fig. 10, and then plotted accuracy opposite to different K value at different speed of the action, their results are shown by the help of line graph in Fig. 11.

Since, the higher value of K can decrease the number of learnt poses which can increase the number of temporal nodes and the time complexity. On the other hand, if the number of poses is higher in an action more will be the LLA recognition accuracy which is possible at low value of K. So, here motive is to find the balance between the number of frames and the accuracy which is achieved at the K =6.

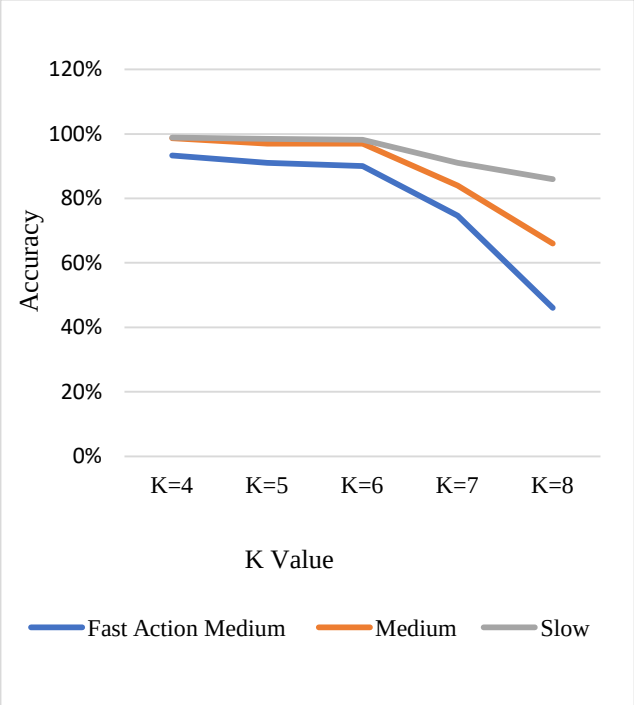


Fig. 11. Accuracy of LLA recognition at different K values and speed of actions

IV. CONCLUSION AND FUTURE WORK:

The Article has shown a unique way for the abstract modeling of Neocortex, Basal ganglia and the Cerebellum brain region in context to learning to perform tasks, where the model can learn the action semantics which can be used in action recognition as well as to perform actions in doing several complex tasks. In simulation results, the model has been found robust in recognition of actions which are performed at different speed and different manners. The model can work in all aspects of the human way of performing a task, from recognition to performing the tasks. The abstract model of hippocampus can be added in future version of the model, in which the agent can represent event activities using the recognized actions.

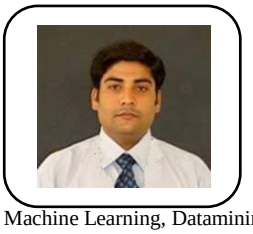
REFERENCES

1. Hawkins, Jeff, Subutai Ahmad, and Yuwei Cui. "A theory of how columns in the neocortex enable learning the structure of the world." *Frontiers in neural circuits*, vol.11, p. 81, 2017
2. Berns, G.S., Sejnowski, T.J.: A computational model of how the basal ganglia produce sequences. Journal of cognitive neuroscience, vol. 10(1), pp.108–121, 1998
3. Atallah, H.E., Frank, M.J., O'reilly, R.C.: Hippocampus, cortex, and basal ganglia: Insights from computational models of complementary learning systems. Neurobiology of learning and memory, vol. 82(3), pp. 253–267, 2004
4. Baston, Chiara, and Mauro Ursino. "A biologically inspired computational model of basal ganglia in action selection." Computational intelligence and neuroscience, vol. 2015, p.93, 2015
5. Yin, Henry H. "How basal ganglia outputs generate behavior." Advances in neuroscience, vol. 2014, 2014
6. Ozden, I., Sullivan, M.R., Lee, H.M. and Wang, S.S.H., Reliable coding emerges from coactivation of climbing fibers in microbands of cerebellar Purkinje neurons. *Journal of Neuroscience*, vol.*29*(34), pp.10463-10473, 2009
7. Mehta, D., Sridhar, S., Sotnychenko, O., Rhodin, H., Shafiei, M., Seidel, H.P., Xu, W., Casas, D. and Theobalt, C., Vnect: Real-time 3d human pose estimation with a single rgb camera. *ACM Transactions on Graphics (TOG)*, vol.*36*(4), p.44, 2017
8. Welsh, J.P., Lang, E.J., Suglhara, I. and Llinás, R., Dynamic organization of motor control within the olivocerebellar system. *Nature*, vol.374(6521), p.453, 1995
9. Lang, E.J., Sugihara, I. and Llinás, R., Olivocerebellar modulation of motor cortex ability to generate vibrissal movements in rat. *The Journal of physiology*, vol.*571*(1), pp.101-120, 2006
10. Van Der Giessen RS, Koekkoek SK, van Dorp S, De Gruijl JR, Cupido A, Khosrovani S, et al. Role of olivary electrical coupling in cerebellar motor learning. Neuron. vol.58, pp.599-612, 2008
11. Mukamel EA, Nimmerjahn A, Schnitzer MJ. Automated analysis of cellular signals from large-scale calcium imaging data. Neuron, vol.63, pp.747-760, 2009
12. Hoogland TM, De Gruijl JR, Witter L, Canto CB, De Zeeuw CI. Role of synchronous activation of cerebellar Purkinje cell ensembles in multi-joint movement control. Curr Biol. vol.25, pp.1157-65, 2015
13. Rosenbloom, P.S., Laird, J.E., Newell, A., McCarl, R.: A preliminary analysis of the soar architecture as a basis for general intelligence. Artificial Intelligence, vol. 47(1-3), pp.289–325, 1991
14. Anderson, J.R., Matessa, M., Lebiere, C.: Act-r: A theory of higher level cognition and its relation to visual attention. Human-Computer Interaction, vol. 12(4), pp.439–462,1997
15. Nason, S. and Laird, J.E.: Soar-RL: Integrating reinforcement learning with Soar. *Cognitive Systems Research*, vol.*6*(1), pp.51-59,2005
16. Beetz, Michael, Lorenz Mösenlechner, and Moritz Tenorth. "CRAM—A Cognitive Robot Abstract Machine for everyday manipulation in human environments." *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010
17. Langley, P., Choi, D. and Rogers, S., Interleaving learning, problem-solving, and execution in the ICARUS architecture. *Computational Learning Laboratory, CSLI, Stanford U., Tech. Rep*, 2005

AUTHORS PROFILE

Rahul Shrivastava has done his B.E. in CSE from BITS Bhopal (Madhya Pradesh). Currently, he is is pursuing Dual Degree (MTech + PhD) in CSE Dept at National Institute of Technology Patna, India. His research area includes Brain Computational modeling, Artificial Intelligence (Machine Learning, Deep Learning).

Prabhat Kumar is an Associate Professor in Computer Science and Engineering Department at National Institute of Technology Patna, India. He is a member of the International Federation for Information Processing (IFIP) Working Group (WG) 6.11: Communication Aspects of the E-World as well as member of NWG-13 (National Working Group 13) corresponding to ITU-T Study Group 13 "Future Networks, with focus on IMT-2020, cloud computing and trusted network infrastructures". Dr.Prabhat Kumar holds a Ph.D. in Computer Science and M.Tech. in Information Technology. He has chaired sessions at several international conferences held in India and abroad. He is also in the Program Committee of various National/International Conferences. He is a senior member of IEEE, professional member of ACM, life member of CSI, International Association of Engineers (IAENG), Indian Society for Technical Education (ISTE) and global member of Internet Society. His research area includes Wireless Sensor Networks, Internet of Things, Social Networks, Operating Systems, Software Engineering, E-governance etc.



Dr. Sudhakar Tripathi Associate Professor, Information Technology, Rajkiya Engineering College, Ambedkar Nagar, U.P. India. He has completed his Ph.D. from IIT(BHU) Varanasi. He has published many research articles in Reputed International Journal. His research area is artificial intelligence, Artificial Neural Networks, Computational Modeling and Research, Machine Learning, Datamining