Generating goal-directed visuomotor plans based on learning using a predictive coding type deep visuomotor recurrent neural network model

Minkyu Choi[1], Takazumi Matsumoto[1], Minju Jung[1,2] and Jun Tani[1,*]

Abstract— **The current paper presents how a predictive coding type deep recurrent neural networks can generate vision-based goal-directed plans based on prior learning experience by examining experiment results using a real arm robot. The proposed deep recurrent neural network learns to predict visuo-proprioceptive sequences by extracting an adequate predictive model from various visuomotor experiences related to object-directed behaviors. The predictive model was developed in terms of mapping from intention state space to expected visuo-proprioceptive sequences space through iterative learning. Our arm robot experiments adopted with three different tasks with different levels of difficulty showed that the error minimization principle in the predictive coding framework applied to inference of the optimal intention states for given goal states can generate goal-directed plans even for unlearned goal states with generalization. It was, however, shown that sufficient generalization requires relatively large number of learning trajectories. The paper discusses possible countermeasure to overcome this problem.**

I. INTRODUCTION

Recently robotics researchers have focused more on studies on learnable robot by using advanced schemes of deep learning [2], [4], [6], [7]. Obvious benefit is that learning by robots themselves can ease difficulty in describing precise models of the robots and their environment by the users. The most popular approach in applying deep learning to robots is to use convolutional neural network (CNN) for developing visuomotor mapping possibly by using reinforcement learning framework [3], [4]. Another interesting approach is using the framework of predictive coding [8], [9] to robot learning problems [1], [5], [6], [7], [11], [14]. The predictive coding framework can allow robots to develop task specific internal models by extracting latent causality between intention states and the resultant outcomes of perceptual sequences through learning of accumulated sensory-motor experiences. Yamashita & Tani [14] and Noda et al. [7] showed that a set of skilled behaviors like manipulating objects can be learned for robust generation using the predictive mechanism in this framework. Hwang et al. [6] showed that imitation learning using pixel level dynamic vision can be performed successfully by using predictive coding type deep visuomotor deep RNN model. Although the current application of the predictive coding is limited to simple prediction of action outcomes, it can be applied to more cognitively challenging problems involved with optimal action planning and their dynamic execution for achieving arbitrarily given goals. The current paper presents the first step toward such research goals by reporting a set of results from our robotic experiments.

The basic ideas and trails shown in the current study is briefly described in the following. The predictive coding scheme is implemented into a neural network model, referred to as predictive coding type deep visuomotor recurrent neural network model (P-DVMRNN). A real arm robot with vision is tutored for object-directed behavior generation tasks such as grasping an object for placing it on a goal target sheet. The tutoring is repeated for teaching a set of different trajectories dealt with large variation in positions such as for the object and the target goal sheet. The tutoring of each goal-directed trajectory for a particular task provides a robot with related multimodal perceptual experience consisting of pixel level vision and proprioception in terms of the joint angles which are extended in time as synchronized.

A set of visuo-proprioceptive sequences obtained through tutoring of a particular task is used for off-line training of the P-DVMRNN model. P-DVMRNN model learns to regenerate each training trajectory by inferring the corresponding intention state. Here, the intention state which is represented by the internal neural activity in the model network encodes the way of the robot intending to interact with the environment. It is noted that each intention state is self-determined in the course of learning. Consequently, after adequate training with good generalization it is expected that a causal mapping from the intention state space to the corresponding perceptual sequence space can be developed in the model network. After successful learning, the model network can generate mental image for visuo-proprioceptive sequence for the intention state inferred for the tutoring sequence. Moreover, it is assumed that an intention state located neighboring among those inferred in the training can generate analogous one by possibly interpolating those trained trajectories if generalization in learning can be done successfully.

Let us consider further extension of the scheme to involve with goal-directed planning as the main objective in the current study. Suppose that goal state is given in terms of the corresponding perceptual state such as a visual frame image of a robot putting an object on a goal sheet. Then, the problem of planning is to infer the corresponding intention state which can achieve the specified perceptual state in the distal step by inversely applying the acquired causality

*Corresponding author

[1] Okinawa Institute of Science and Technology (OIST), Okinawa, Japan

[2] Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea

{minkyu.choi8904, takazumi, minju5436, tani1216jp}@gmail.com

between the intention to the perceptual sequence. Although it would be trivial to generate corresponding trajectories to a prior learned goal states, the same may not be assured for the case of unlearned goal states. We examine this issue by conducting robotic experiments by changing task difficulty. Although all the robot tasks considered in the current study might be relatively simple, our trial should be the first one for applying the predictive coding framework to the learning-based robot action planning by using deep learning scheme. The paper will focus on some difficulty we encountered in terms of generalization in planning and will discuss how the problem could be resolved by improving the scheme in future.

II. Method

In the predictive coding framework, all three processes of learning, recognition, and generation can be conducted by means of the prediction error minimization. Firstly, the learning is a process to map between intention states and the resultant perceptual sequences by self-determining the corresponding intention state for each sequence and connectivity weights for minimizing the prediction error. In the case of using RNN models for implementing the predictive coding scheme as like in the current study, the intention state can be represented by the initial states of internal neural units by utilizing the initial sensitivity characteristics of the RNN dynamics. Recognition is a process to infer inversely the corresponding intention state for a given target perceptual sequence. Finally, plan generation is to infer the corresponding intention state to achieve a goal state given at the distal step. The intentions state inferred is used to generate perceptual sequence reaching to the goal state. Next, we show how this predictive coding idea can be implemented in the current proposed neural network model.

A. Neural Network Architecture

Our network architecture (shown in Fig. 1) uses a recurrent neural network (RNN) based on predictive coding [8] capable of learning, generating, and recognizing multi-modality perceptual sequence inputs. The network has two closely related paths dedicated to processing visual input and motor joint angles respectively. At each time step, visual input in the form of a frame captured from an RGB camera is provided to the network, as well as the corresponding motor joints angles from a robot arm. The visual image and joint angles are fed as inputs to the lowest layer of the network and processed through three layers, then finally merged in the highest layer. The outputs predicting both the next visual input and joint angles are generated based on the internal neural activity in the lowest layer. Each layer is only connected to its neighbors (above/below) and the adjacent visual/motor counterpart. These structural characteristics enable network to process incoming data in an hierarchical manner [6], [15].

For the visual path, convolutional Long Short-term Memory (LSTM) networks [16] are used as the basic building blocks of the network in order to process spatial and temporal information simultaneously [6], [13], [15], [16]. In each path, there are two streams of information: top-down and bottom-up. The top-down connection projects the current prediction to the lower layers, while the bottom-up connection carries information from outside the network or errors between predictions and actual inputs. Top-down connections in vision path utilize convolutional operations and pooling, while the bottom-up connections are implemented as a transposed convolution [17]. The size of the feature maps for each layer is designed to be half of the previous lower level.

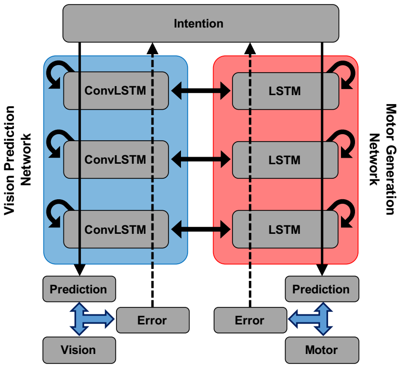


Fig. 1: Overall network architecture. The proposed predictive coding type deep recurrent neural network model dealing with visuomotor sequences.

For the motor path, which operates on lower dimensional data compared to the visual path, LSTM is used. Similar to the feature maps in the visual path, the number of neurons in the motor path decreases along the hierarchy in the network. In order to improve learning with low dimensional data, sparse encoding is utilized [6], [14]. In this work, each motor joint is represented by a 10 dimensional sparsely encoded vector.

The number of layers in both visual and motor pathways are identical, and as mentioned previously the layers at the same level in both visual and motor pathways are connected to each other horizontally. In Fig. 1, the lateral connections at each level enable the whole network to exchange information between the two paths. Thus, this lateral connection is key for the network to closely couple the two modalities and maintains the link between a given visual input and motor joint angle. The network forms the common internal states of the highest layer by recognizing both visual input and current motor state from bottom-up connections. Since the raw visuomotor information is processed hierarchically through multiple layers from the lowest layer to highest layer, this internal state presents abstract information of the current environment as well as robot's state. Based on this abstract representation, the model is able to predict future visuomotor input by projecting it towards the lowest layer, generating a pixel-level visual prediction and a sparsely encoded joint angle prediction. A breakdown of each layer in the two network paths is shown in Table I.

TABLE I: Breakdown of the size & number of convolution feature maps (vision) and LSTM layers (motor) per layer

	In/Out	L1	L2	L3	L4
Vision Path	64×64×1	32×32×40	16×16×80	8×8×80	4×4×12 Shared
Motor Path	40	1024	1024	16	

B. Training

During training, our model learns to predict/generate a set of training sequences by inferring the corresponding intention states represented by initial states (IS) of the internal units for each sequence as well as connectivity weights using back-propagation through time (BPTT) [18] towards the direction of minimizing the prediction error. In this model, IS are self-determined for each training sequence.

C. Inference of intention for novel sequences

With the trained network parameters (for e.g., weights and biases), it is expected that each training sequence can be regenerated using the corresponding IS in a closed-loop manner[1]. When a novel perceptual sequence pattern is given to the learned network, it can be recognized by inferring the corresponding IS values by means of error regression (ER) scheme for minimizing the prediction error without changing connectivity weights. When the learning can be done with sufficient generalization, it is generally assumed that a novel sequence pattern which is similar to a particular trained one tends to be inferred with a similar IS value.

For example, consider a scenario in which we wish to find the IS (h_0) which encodes a given sequence in a simple RNN. Since we have no information of the actual IS, we set the current IS to a random value and start searching. In this search, we first generate a prediction output using the randomly set IS in a closed-loop manner as noted previously. Given the random IS, the output ($O_{1:T}$) of this process is unlikely to match our target sequence ($T_{1:T}$), producing a prediction error ($E_{prediction}$).

$$E_{prediction} = \sum_{t=1}^{T} ||O_t - T_t||^2 \qquad (1)$$

This prediction error ($E_{prediction}$) is then back-propagated through time (BPTT) [18]. Unlike training, during the ER process, model parameters such as weights and biases are left unchanged. Only the IS is optimized for prediction error minimization. This process is iterated multiple times until the predicted output follows the target sequence by minimizing prediction error. Once the optimal IS is found, the network is able to generate (or decode) the corresponding sequence.

D. Planning

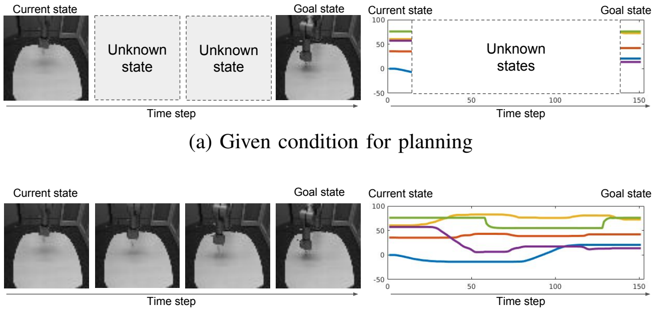
This subsection describes how goal-directed action plans can be generated by extending the scheme of the ER. Let us suppose that the robot waits for a goal to be specified while staying at predefined home position posture. We consider that a goal state is given in terms of its corresponding perceptual state, i.e., visual state ($V_{target,T}$) and joint angle state ($M^j_{target,T}$, where j is an index of joints and J is the number of joints, $j \in J$). Then the problem to solve is to generate an optimal visuomotor sequence which can rationally connect the perceptual state in the home posture in the initial step ($V_{target,1}$ and $M^j_{target,1}$) and the one in the goal state. Fig. 2a presents the available information for making plans and Fig. 2b shows the generated visuomotor sequence connecting initial step and goal step. Because the model network can generate various possible visuomotor sequences by changing the IS based on learning, it is considered that an optimal IS for generating such sequence can be searched by using the aforementioned ER scheme. A difference in the ER scheme for the plan generation is that the target perceptual states are given only partially, at the initial state and the end state. Thus, the prediction error which will be used to optimize an IS by ER is given as follows:

$$\begin{aligned} E_{prediction} =& ||V_{out,1} - V_{target,1}||^2 + ||V_{out,\hat{T}} - V_{target,T}||^2 \\ &+ \sum_{j=1}^{J} KL(M^j_{out,1}||M^j_{target,1}) \\ &+ \sum_{j=1}^{J} KL(M^j_{out,T}||M^j_{target,T}) \end{aligned} \qquad (2)$$

where $V_{out,t}, V_{target,t}, M^j_{out,t}, M^j_{target,t}$ are the visual predicted output at step t, the visual target at step t, the j^{th} joint angle predicted output at step t and the j^{th} joint angle target at step t respectively. $\hat{T}$ is a target step for the robot producing a target output. The time step at which the robot would achieve the given goal state is not known so it may be different from the ground truth target T. Therefore, during the optimization process, $\hat{T}$ is inferred. During the ER process, based on the current IS, a closed-loop prediction is generated until a predefined step T_{max} which is long enough to achieve the goal state. Then the ground truth visual target image $V_{target,T}$ is compared against all generated prediction output frames (from $V_{out,1}$ to $V_{out,T_{max}}$) and it generates errors for each step. In order to promote the model to achieve the goal faster, in a more optimized way, compensation value 1.01^t is multiplied to the prediction error calculated for each time step. Among those predicted frames, the frame that has the smallest compensated error compared to $V_{target,T}$ is set as $V_{out,\hat{T}}$ and the respective output step $\hat{T}$. As noted in Section III, this can result in a shorter sequence of steps to reach the goal. In this work, as sparse coding is applied to the motor joint angles, Kullback-Leibler divergence (KL) is used to measure error between motor targets and outputs.

When the IS for visual path is found by optimization, the robot is able to generate motor joint angles associated with each predicted image. As shown in the network architecture, the two modalities of vision and motor are correlated through lateral connections. Therefore, by searching optimal IS for

[1] Closed-loop: giving the previous time step's output as the current step's input, as opposed to Open-loop: input to each time step is given from ground truth data.



(a) Given condition for planning

(b) Generated plan for vision and motor

Fig. 2: Planning by error regression. For both (a) and (b), left figures show visual data and right graphs present motor joint angles

one modality, it is possible to induce the other modality. Inducing motor joints angles is therefore possible by optimizing IS in the visual path and vice versa. The error from Equation 2 describes the case when the both the visual and motor target is given. However, when only a target from one modality is given, eliminating the corresponding target term from Equation 2 will yield a new prediction error. For example, in case the motor target is not given, the term $\sum_{j=1}^{J} KLD(M_{out,T}^{j}||M_{target,T}^{j})$ should be removed.

III. Experiments

In this section, we describe the experimental procedures and results obtained using an arm robot and camera connected to our network. Three tasks with different behavioral complexity were considered. The goals of the three tasks were, 1) reaching to a single point in the task space, 2) reaching to two points sequentially in the task space, and 3) grasping an object and putting it on a goal target sheet in the task space[2].

For the reaching tasks, our robot was configured to used 4 of its 7 joints, while in the grasping task, 5 joints including an end effector were used. The camera was in a fixed position facing the workspace and robot. While collecting training and testing data, both visual data and motor joint angles were sampled at 10Hz (for the grasping task, this was reduced to 2.5Hz). Each frame from the camera was resized to 64×64 pixels and 8 bit grayscale before being provided to the network.

Data collection was conducted in two phases: first, a human operator moved the robot by hand following a set of randomly generated positions. After the joint angles were recorded, the robot recreated the recorded trajectory and captured the video of the motion. For testing purposes, we only used the initial and final states of visuomotor trajectories from a test set and compared the prediction to the ground truth test trajectories. Because this data was generated by a human operator, it will naturally have noise and fluctuations. If our model is able to generalize the training trajectories to reach the goal state, it should be able to ignore the unnecessary pauses and fluctuations. Finally, both the pixel level vision and joint angles were normalized to $[-1, 1]$, and we utilized the Adam optimizer for training the network [19].

A. Experiment 1: Reaching

For the first experiment, a frame from the target vision data showing the last position of the robot arm is given as its goal state. To successfully accomplish this task, the network must generate a plausible prediction for both visual input and corresponding joint angles forming a trajectory of the arm robot. This experiment used a table (74cm x 74cm) for the task space where 100 reaching trajectories for training were generated by randomly allocating the target position on the table. Testing was done on 40 randomly sampled positions that were not part of the training set.

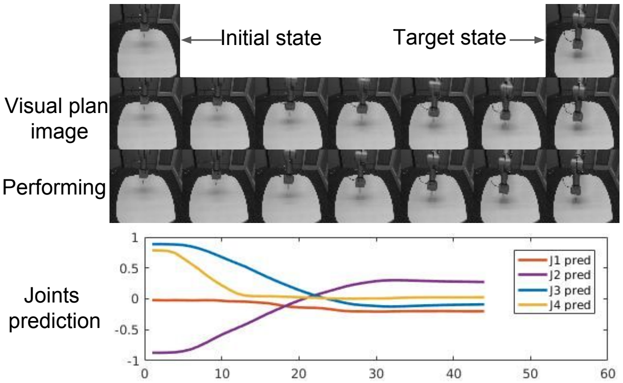


Fig. 3: Example results from the reaching experiment. The upper row shows images of the initial and target visual states in the top level, visual image sequence generated for plan in the second level, and actual visual input perceived during execution of the plan in the bottom level. The lower row shows joint angles generated for the plan. Fig. 4 and Fig. 5 are presented in the similar manner.

As described in equation 2, the IS of the network is optimized based on the errors between the visual predictions and visual targets given in the first and the last frames and then motor joint angles are generated based on the optimized IS. Fig. 3 shows the results of this experiment. Although there is some visible blurring in prediction outputs (second row), the overall shape of the arm and its movement are maintained. We also observe that the motor joint angles were successfully generated even though the target values of the joints were not provided to the network.

As mentioned previously, since the training patterns were generated by a human operator, the trajectories are inconsistent. However, the robot reaches the goal state faster than the similar training trajectories. This suggests the model can generate more optimized trajectories that still reach the goal state, by generalizing training patterns.

To evaluate planning performance, we measured the distance between the final position reached based on the plan and the target position specified. Given the imprecision in the visual input, if the deviation at the end of the trajectory

[2] Videos can be found at https://sites.google.com/site/academicpapersubmission/p-dvmrnn

TABLE II: Success rate for experiment 1 with varying training set sizes

Training set size	25	50	100
Average deviation	5.3cm	3.2cm	2.6cm
Success rate	45%	70%	84%

was less than 4cm (i.e., less than 3 pixels), the result was judged to be successful. Considering the overall size of robot arm (approximately 80cm), a 4cm error is believed to be reasonable. Over 40 test data points, the recorded average deviation was 2.6cm with a maximum deviation of 5.4cm. The overall success rate of experiment 1 was 84%.

Although some degree of position generalization was achieved as shown by the success rate of 84% in the test generation, it was true that a relatively large amount of tutoring trajectories were used for the learning. Therefore, we examined how much the position generalization depended on the amount of tutoring trajectories. For this purpose, the same experiment was repeated by reducing number of tutoring trajectories, to 50 and 25. The result of the success rate in test generation is summarized in Table II. It can be seen that the success rate decreased significantly when the number of tutoring trajectories was reduced. It can be said that by using the current model, a reasonable success rate in test generation requires a relatively large amount of tutoring data of around 100 trajectories even for a relatively simple task as like the current one.

B. *Experiment 2: Reaching two points*

For the second experiment, we extended the first task by adding an intermediate target that the robot must touch before reaching the goal. The intermediate target was marked by a filled circle with a diameter of 12cm. The goal state was given as the last visual frame showing the intermediate target marker and the arm in the final position. To accommodate the two distributions of locations, the task space was expanded to 100cm by 100cm.

The task for the robot is to 1) touch the intermediate marker and then 2) move to the final position. For this task, if the robot touches a point within the marker and reaches the final position with a deviation of the end effector of less than 4cm, the trial is regarded as successful. For training the network, 100 training sequences were collected. Fig. 4 shows the target, predicted and actual visual frames as well as joint angles for one trial. The overall success rate was 75% for this task.

C. *Experiment 3: Moving an object*

For the third experiment, we added an object to the task space for the robot to manipulate. The object and the target circle are placed in two randomly sampled locations as in experiment 2 and the task for the robot was to 1) grasp the object and 2) place it in the target circle. The object was a plastic cylinder with a diameter of 5cm and a height of 10cm. The target circle and workspace were the same as in experiment 2. 100 training sequences and 50 test sequences are used. In testing, if the robot grasped the object and placed it upright anywhere within the target circle, it was considered a success.

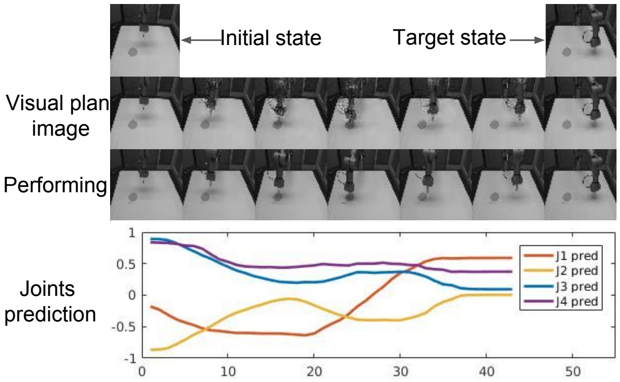


Fig. 4: Results of touching two points experiment

TABLE III: Success rate for experiment 3 with varying amounts of error allowed in grasping

	Strict grasping	Allowed error in grasping		
		1 pixel	2 pixels	3 pixels
Closed loop Prediction	48%	48%	64%	71%
Open loop Prediction	74%	74%	88%	93%

The difficulty of the task is considerably higher compared to the previous experiments, because the end effector must be moved accurately to grasp the object without sensory feedback. A successful trial is shown in Fig. 5. The overall success rate was 48%. The challenge here was primarily the low resolution of visual input and resulting inaccurate predicted trajectories. Due to the size and shape of the the object and the end effector, any deviation greater than 2cm often resulted in failure to grasp the object. Despite this, we noted that once the robot grasped the object, it was able to successfully place the object in the target circle (94% success rate, with an average deviation of 3cm).

In order to break down the performance in this task further, we considered the deviation from the center of the object to the center of where the end effector actuated. Allowing a 1 to 3 pixel error (1.3cm to 3.9cm deviation) in grasping, in line with previous tasks, the success rate was improved considerably as shown in Table III.

Additionally, we tested the ability of our model to produce one-step predictions. Unlike the previous tests, for one-step prediction the network observed ground truth visual data and motor joint angles after making a prediction at each timestep. This prediction scheme is employed in several other works [10], [11], [12]. As the model receives sensory feedback, it yields better results than closed loop prediction. This difference is shown in Table III as closed loop prediction and open loop prediction respectively.

IV. Conclusion

In this paper, we proposed a novel architecture for goal-directed action planning using a predictive coding type

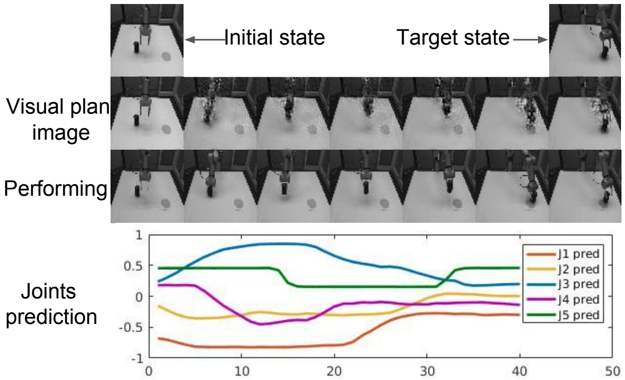


Fig. 5: Results of moving object experiment

deep dynamic neural network. In the robot experiment, the network learned to generate a set of visuo-proprioceptive sequences by self-determining the corresponding intention state in terms of the IS for each sequence as well as connectivity weights in the whole network. After learning, the network was able to generate optimal visuomotor plans for the specified goal states by inferring the corresponding IS with some degree of generalization. Our experimental results have shown that our architecture can produce not only near future predictions (one-step ahead) as used in existing works, but also far future states (multi-step ahead) for both visual and motor modalities. However, due to restrictions in image resolution used in the vision network, the robot frequently failed in a grasping task that required precise positioning. This issue can be ameliorated somewhat by increasing image resolution or adding additional sensory input (for e.g., depth perception or tactile sensation) at the expense of increased computational cost.

A significant issue we observed was that to achieve fair generalization in learning and plan generation required a relatively large amount of training data. As shown with the first experimental task, the success rate in reaching the goal state was significantly reduced as number of training sequences was decreased. How can we solve this generalization problem? One possible solution may be to introduce a variational Bayes (VB) scheme to the model network [20]. Recently, VB schemes have been introduced to several RNN models [21], [22]. RNN models using a VB scheme show better generalization in learning by extracting probabilistic structures hidden in perturbed sequence data when the regularization term of controlling entropy in the neural activity is adequately tuned [22]. Examination of such models applied to learning-based goal-directed planning of robots is left for future study.

References

[1] Tani, J. (2016). Exploring Robotic Minds: Actions, Symbols, and Consciousness as Self-Organizing Dynamic Phenomena. New York: Oxford University Press.

[2] Cangelosi, Angelo, and Thomas Riga. ”An embodied model for sensorimotor grounding and grounding transfer: Experiments with epigenetic robots.” Cognitive science 30.4 (2006): 673-689.

[3] Hausknecht, Matthew, and Peter Stone. ”Deep recurrent q-learning for partially observable mdps.” CoRR, abs/1507.06527 (2015).

[4] Levine, Sergey, et al. ”End-to-end training of deep visuomotor policies.” The Journal of Machine Learning Research 17.1 (2016): 1334-1373.

[5] Tani, Jun, and Nolfi, Stefano. ”Learning to perceive the world as articulated: an approach for hierarchical learning in sensory-motor systems.” Neural Networks 12.7-8 (1999): 1131-1141.

[6] Hwang, Jungsik, et al. ”Dealing With Large-Scale Spatio-Temporal Patterns in Imitative Interaction Between a Robot and a Human by Using the Predictive Coding Framework.” IEEE Transactions on Systems, Man, and Cybernetics: Systems (2018).

[7] Noda, Kuniaki, et al. ”Multimodal integration learning of object manipulation behaviors using deep neural networks.” Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on. IEEE, 2013.

[8] Rao, Rajesh PN, and Dana H. Ballard. ”Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects.” Nature neuroscience 2.1 (1999): 79.

[9] Friston, Karl. ”The free-energy principle: a unified brain theory?.” Nature Reviews Neuroscience 11.2 (2010): 127.

[10] Rahmatizadeh, Rouhollah, et al. ”Vision-Based Multi-Task Manipulation for Inexpensive Robots Using End-To-End Learning from Demonstration.” arXiv preprint arXiv:1707.02920 (2017).

[11] Nagai, Yukie, and Minoru Asada. ”Predictive learning of sensorimotor information as a key for cognitive development.” Proc. of the IROS 2015 Workshop on Sensorimotor Contingencies for Robotics. 2015.

[12] Levine, Sergey, et al. ”Learning hand-eye coordination for robotic grasping with large-scale data collection.” International Symposium on Experimental Robotics. Springer, Cham, 2016.

[13] Jung, Minju, Jungsik Hwang, and Jun Tani. ”Self-organization of spatio-temporal hierarchy via learning of dynamic visual image patterns on action sequences.” PloS one 10.7 (2015): e0131214.

[14] Yamashita, Yuichi, and Jun Tani. ”Emergence of functional hierarchy in a multiple timescale neural network model: a humanoid robot experiment.” PLoS computational biology 4.11 (2008): e1000220.

[15] Choi, Minkyu, and Jun Tani. ”Predictive Coding for Dynamic Visual Processing: Development of Functional Hierarchy in a Multiple Spatiotemporal Scales RNN Model.” Neural computation 30.1 (2018): 237-270.

[16] Xingjian, S. H. I., et al. ”Convolutional LSTM network: A machine learning approach for precipitation nowcasting.” Advances in neural information processing systems. 2015.

[17] Zeiler, Matthew D., Graham W. Taylor, and Rob Fergus. ”Adaptive deconvolutional networks for mid and high level feature learning.” Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011.

[18] Werbos, Paul J. ”Backpropagation through time: what it does and how to do it.” Proceedings of the IEEE 78.10 (1990): 1550-1560.

[19] Kingma, Diederik P., and Jimmy Ba. ”Adam: A method for stochastic optimization.” arXiv preprint arXiv:1412.6980 (2014).

[20] Kingma, Diederik P., and Max Welling. ”Auto-encoding variational bayes.” arXiv preprint arXiv:1312.6114 (2013).

[21] Chung, Junyoung, et al. ”A recurrent latent variable model for sequential data.” Advances in neural information processing systems. 2015.

[22] Ahmadi, Ahmadreza, and Jun Tani. ”Bridging the gap between probabilistic and deterministic models: a simulation study on a variational Bayes predictive coding recurrent neural network model.” International Conference on Neural Information Processing. Springer, Cham, 2017.