

# Symbolic-Connectionist Representational Model for Optimizing Decision Making Behavior in Intelligent Systems

R. John Martin<sup>1</sup>, Sujatha<sup>2</sup>

<sup>1</sup>School of Computer Science and Engineering, Bharathiar University, Coimbatore, India

<sup>2</sup>Department of Computer Science, Bharathi Women's College (Autonomous), Tamil Nadu 600108, India

---

## Article Info

### Article history:

Received Apr 2, 2017

Revised Jul 30, 2017

Accepted Aug 14, 2017

### Keyword:

Cognitive process

Connectionist model

Decision making

Machine learning

Neural network

---

## ABSTRACT

Modeling higher order cognitive processes like human decision making come in three representational approaches namely symbolic, connectionist and symbolic-connectionist. Many connectionist neural network models are evolved over the decades for optimizing decision making behaviors and their agents are also in place. There had been attempts to implement symbolic structures within connectionist architectures with distributed representations. Our work was aimed at proposing an enhanced connectionist approach of optimizing the decisions within the framework of a symbolic cognitive model. The action selection module of this framework is forefront in evolving intelligent agents through a variety of soft computing models. As a continuous effort, a Connectionist Cognitive Model (CCN) had been evolved by bringing a traditional symbolic cognitive process model proposed by LIDA as an inspiration to a feed forward neural network model for optimizing decision making behaviours in intelligent agents. Significant progress was observed while comparing its performance with other variants.

Copyright © 2018 Institute of Advanced Engineering and Science.  
All rights reserved.

---

## Corresponding Author:

R. John Martin,  
School of Computer Science,  
Bharathiar University,  
Coimbatore, India.  
Email: jmartin.in@gmail.com

---

## 1. INTRODUCTION

Optimizing the agent based decision making systems is a centre of intelligent systems research. Agent based decision making is fundamentally different from human decision making and make decisions according to programmed procedures. In cognitive modelling, decision making is viewed as a high level mental process which involves judging multiple options in order to choose one, so as to fulfil the objective of the decision making agent [1], [2]. This paper describes the effort of moulding a connectionist model for optimization by incorporating the principles of LIDA (Learning Intelligent Distribution Agent) cognitive architecture [3]. LIDA facilitates the cognitive process of decision making through its 'action selection' module [4], [5].

The LIDA action selection module is being adopted to evolve connectionist neural architecture to enhance its computational ability. The cognitively motivated computational connectionist model will be used for various optimization problems with considerably good impact in its decision making agents. Even though cognitive modelling (symbolic) and neural modelling (connectionist) use two different approaches in building intelligent agents, there are recent developments in combining both approaches (symbolic-connectionist) to reinforce the field of cognitive agent building [6-8].

The following sections of this paper describe the concepts and principles behind the computational methods to bringing out an enhanced Connectionist Cognitive Network (CCN) model for optimizing real

time decision making problems. This will be a motivation for the prospective researchers to approach the new dimension of cognitive model building.

## 2. ACTION SELECTION MODULE OF LIDA

The LIDA cognitive architecture extensively defines the higher order cognitive process of decision making with its action selection module. The action selection phase is basically a learning phase where several processes operate in parallel in the complete cognitive cycle of LIDA (Figure 1) for the decision which will lead to the next cycle. Various associations formed within the entities of its architecture by reinforcing old selections occur as the conscious broadcast reaches Perceptual Associative Memory. New events from the conscious broadcast are encoded as knowledge patterns in Transient Episodic Memory. Potential action patterns, together with their contexts and expected results, are learned into Procedural Memory from the conscious broadcast. This is more similar to the training pattern of the feed forward networks of learning.

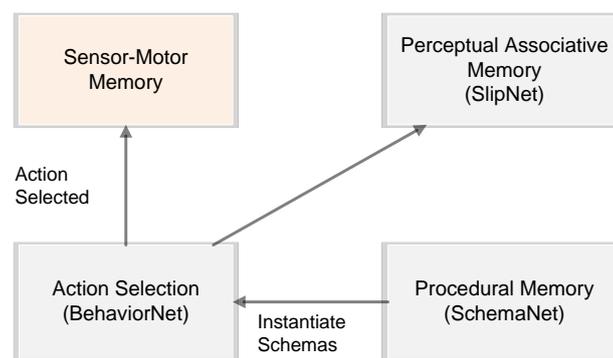


Figure 1. Action-selection module of LIDA cognitive architecture

Alongside of this learning, using the conscious contents, possible schemas for the action behavior are evolved from the Procedural Memory. Same action pattern is sent to Action Selection, where it competes to be the behavior selected for this cognitive cycle. The selected behavior triggers Sensory-Motor Memory to produce a suitable motor plan for the behavior pattern to be carried out [9]. This part of the cognitive cycle is the motivation for the proposed connectionist approach of decision making.

## 3. SYMBOLIC REPRESENTATIONAL MODEL OF DECISION MAKING

Modeling of human decision-making is quite challenging especially under a complex and uncertain environment. Ethical decisions are among the more complex decisions that agents face. Ethical decision making can be understood as action selection under conditions where constraints, principles and values play a central role in determining which behavioral attitudes and responses are acceptable [10].

Many decisions require having to select an action when information is unclear, incomplete, confusing, and even false, where the possible results of an action cannot be predicted with any significant degree of certainty, and where conflicting values can inform the decision-making process. In order to make the process of action selection to be unfussy by an agent, a bottom-up approach has been derived from action selection phase of LIDA cognitive architecture [11]. The Figure 2 is an expansion of action selection module described in the previous section and is viewed as a symbolic representational model.

This symbolic representational model has three layers for optimizing the decision which will be derived as a new schema. The first layer is *schemanet* that receives input from the procedural memory, which is very similar to the input layer of the feed forward connectionist network architecture. The hidden layer of the feed forward net is associated with *slipnet* in the cognitive model which is used to evaluate the adequacy of alternatives for the action selection process. The output layer of the neural net is molded with third part of the cognitive model i.e, *behaviornet*. The new selection (new schema) is fed back to the sensor motor memory for learning. This conceptual framework of cognitive model is the motivational factor to the emergence of enhanced symbolic-connectionist network model for optimizing decisions in intelligent systems.

**4. BACK PROPAGATION NETWORK AS A CONNECTIONIST MODEL**

In connectionist neural network, the most popular method of learning is called Back propagation (BPN) [12]. Learning in feed-forward networks usually supervised learning, meaning that the network is trained by providing it with input and matching output patterns. The computational ability of BPN in optimizing diversified decision making scenarios has also paved way for implementing cognitive models of decision making [13-15].

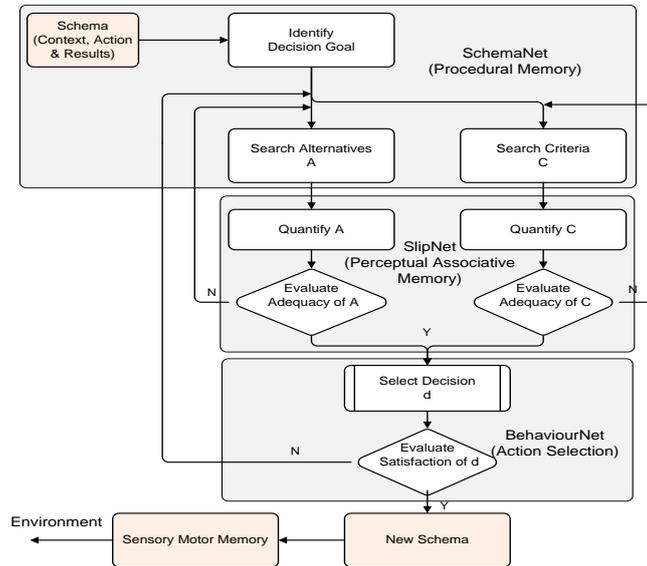


Figure 2. Symbolic representational model of action-selection

The standard form of back propagation algorithm is illustrated in Figure 3. Its computational phases are being presented here for more clarity.

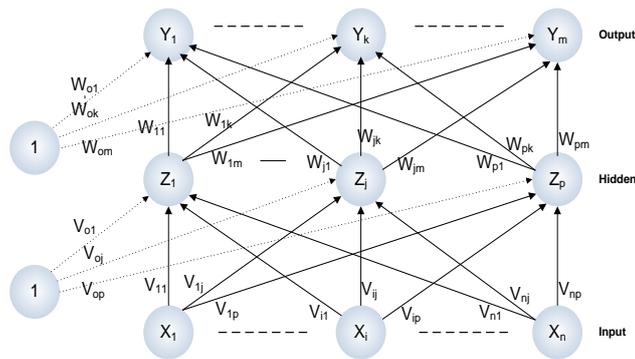


Figure 3. Multilayer feed forward neural Network model

The topology consists of three layers of units namely input, hidden and output layers. The connections between the layers feeding their inputs to the other layer with weighted connections. All the weights are changeable in such a way to learn the network to obtain the desired goal. The network learns by adjusting its weights until to find a set of weights that produce the correct output for every further input.

On considering a feed forward network with  $n$  input ( $i$ ) units,  $p$  hidden ( $j$ ) units and  $m$  output ( $k$ ) units, the back propagation training will be performed in two phases namely forward pass and backward pass.

**4.1. Forward Pass**

One of the set of  $n$  training input patterns is applied to the input layer.

$x_p = (x_{p1}, x_{p2}, \dots, x_{pn})$  which may be a binary or real- vector.

The activations of the neurons in the hidden layer are calculated by using the sum of the activations of the numbered input layer units they are connected to multiply by their respective connection weights and passing it through a transfer function.

Net input to hidden layer unit j is calculated as

$$net_j = \sum_{i=1}^n w_{ji} x_i$$

The hidden layer output, i.e. take net input of unit j and pass it through a transfer function

$$oh_j = \int(net_j)$$

The activations of the hidden layer units calculated are then used in updating the activation of the output units. Net input to output unit k is calculated as

$$net_k = \sum_{j=1}^L w_{kj} oh_j$$

The transfer function for the output of output unit k

$$oo_k = \int(net_k)$$

#### 4.2. Backward Pass

The error signal for each input is calculated by using the difference between the actual activation of each output unit ( $oo_k$ ) and the desired target activation ( $d_k$ ) for that unit.

$$(d_k - oo_k)$$

Delta ( $\delta$ ) for each output unit is then calculated.

$$\delta_{o_k} = (d_k - oo_k) oo_k (1 - oo_k)$$

The error signals for the hidden layer units are then calculated by taking the sum of the deltas of the output units a particular hidden unit connects to multiply by the weight that connects the hidden and output unit.

Error signal for each hidden unit j

$$\sum_{k=1}^W \delta_{o_k} w_{kj}$$

Delta term for each hidden unit j is equal to its error signal multiplied by its output, multiplied by (1 - its output).

$$\delta h_j = (oh_j)(1 - oh_j) \sum_{k=1}^W \delta_{o_k} w_{kj}$$

The weight error derivatives for each weight between the hidden and output units are calculated by taking the delta of each output unit and multiplying it by the activation of the hidden unit it connects to. These weight error derivatives are then used to change the weights between the hidden and output layers.

$$wed_{jk} = \delta_{o_k} (oh_j)$$

The weight error derivatives for each weight between the input unit  $i$  and hidden unit  $j$  are calculated by taking the delta of each hidden unit and multiplying it by the activation of the input unit it connects to (i.e. that input pattern  $x_i$ ). These weight error derivatives are then used to change the weights between the input and hidden layers.

$$wed_{ij} = \delta h_j(x_i)$$

To change the actual weights themselves, a learning rate coefficient  $\eta$  is used, which controls the amount the weights are updated during each back propagation cycle. The weights at a time  $(t + 1)$  between the hidden and output layers are set using the weights at a time  $t$  and the weight error derivatives between the hidden and output layers using the following equation.

$$w_{jk}(t+1) = w_{jk}(t) + \eta(wed_{jk})$$

In a similar way the weights are changed between the input and hidden units

$$w_{ij}(t+1) = w_{ij}(t) + \eta(wed_{ij})$$

The similarities between the action selection model of LIDA cognitive cycle and the BPN algorithm is the motivating factor for modelling of the enhanced Connectionist Cognitive Network (CCN).

In BPN, each unit in the network receives an error signal that describes its relative contribution to the total error between the actual output and the target output. Based on the error signal received, the weights connecting the units in different layers are updated. The two passes are repeated several times for distinct input patterns and their targets, until the error between the actual output of the network and its target output is convincingly small for all the units of the set of training inputs. Feed forward connectionist networks likely to develop internal relationships between units so as to organize the training data, as a result they develop an internal representation that enables them to generate the desired outputs when given the training inputs. The same internal representation can be applied to inputs that were not used during training; the new inputs are classified by the network according to the features they share with the training inputs.

## 5. CONNECTIONIST COGNITIVE NETWORK (CCN) MODEL

Connectionist Cognitive Network (CCN) is a conceptual framework which combines the functionalities of LIDA based action selection model and feed forward neural network model. The architectural similarities of both models are illustrated in the CCN architecture (Figure 4). Similar kinds of frameworks have done with the cognitive model of action selection proposed by LIDA. There are research outcomes that show the importance of bringing cognitive and connectionist approaches together to produce out performed intelligent agents [16].

The authors hope that this framework will eliminate the limitations of a BPN through the adoption of cognitive model by reducing number of learning steps, so that the computing time will also be less compared with pure BPN. The major improvement is achieved by enhancing the momentum factor on updating weights.

During action selection phase of LIDA cognitive cycle, each cognitive cycle will produce the new schema which would be based on the previous schemas generated in previous cycles. The previous schemas are being referred from perceptual associative memory during the current cycle as illustrated in Figure 1 and subsequently in Figure 2.

The weight from one of more previous training patterns must be stored in order to use momentum. Here, the new weights for training step  $i+2$  is based on  $i$  and  $i+1$ . This makes the current weight adjustment with a fraction of the recent weight adjustment. The weight updating formula which differentiates normal feed forward network is

$$w_{jk}(t+1) = w_{jk}(t) + \eta(wed_{jk}) + \mu[w_{jk}(t) - w_{jk}(t-1)]$$

A slight enhancement in this algorithm which will guarantee the convergence is by limiting the momentum factor with the range  $0 < \mu < 0.5$  instead of  $0 < \mu < 1$ . This is very similar to scenario of fine tuning the

decision alternatives through evaluating the search criteria in the perceptual associative memory as illustrated in Figure 2.

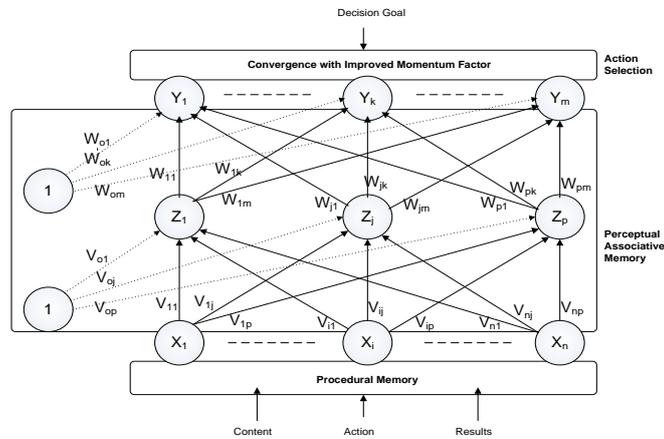


Figure 4. Connectionist cognitive network model

6. RESULTS AND DISCUSSIONS

The performance of the proposed algorithm is verified by simulating an agent for figure print identification as a benchmark problem. The convergence behavior of the proposed algorithm (CCN) is compared with well known BPN training algorithms such as gradient descent (BPN-GD), gradient descent with adaptive learning rate (BPN-GDA) [17]. Epoch numbers, mean squared errors (MSE) and execution times will be considered for evaluating the convergence performance of the algorithms. To have effective comparison in each algorithm, the parameter values and termination condition are considered as constants. To get best results the initial weights are set to random and uniformly generated between [-4, +4]. The learning rate coefficient is assigned the value 0.01.

The figure print identification is a well known database in the pattern recognition. Three classes of data set each has 30 instances, totally 90 patterns are used. Out of 90 patterns, 70 patterns are used for training and the remaining 20 used for testing. The input values are normalized before applied to the network. Table 1 shows the results of network convergence while running the respective algorithms.

Table 1. Network Convergence

Algorithm	Training Parameters	Epochs	MSE @ Training	MSE @ Testing	Time (msecs)
BPN-GD	$\eta = 0.01$	395	0.00030	0.00951	143
BPN-GDA	$\eta = 0.01$	358	0.00029	0.00719	129
Proposed CCN	$\eta = 0.01, \mu = 0.09$	139	0.00028	0.00692	67

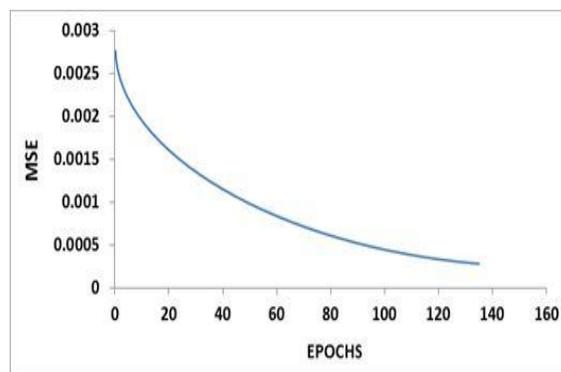


Figure 5. Network learning curve of CCN

The proposed CCN algorithm converges at 139<sup>th</sup> iteration while other two BPN algorithms at the epochs of 395 and 358 respectively. It also takes minimum time of 67 msec to converge. But BPN-GD and BPN-GDA have taken 143 msec and 129 msec respectively to reach the termination with condition MSE=0.0003. The notable variation found in the proposed CCN is its testing MSE i.e 0.692, which is minimum among all algorithms. The learning curve drawn against epochs and MSE for the proposed algorithm is shown in Figure 5.

## 7. CONCLUSION

The strategy adopted to make a unique Connectionist Cognitive Network (CCN) model by associating the BPN and LIDA Cognitive model of action selection is computationally proved with its enhanced network performance compared with other two variations of BPN. The convergence speed of the proposed algorithm in terms of time and epochs is shown by simulating the benchmarking problem of figure print verification. The highlight of the proposed algorithm is the minimum MSE observed while testing. This can be improved by training with more input patterns. The proposed CCN model should also be compared with other leading architectures like SVM to know its limitations.

It is evident that this initiative will eliminate the limitations of both traditional symbolic and connectionist approaches and will lead us to a new dimension of higher order cognitive agent building.

## REFERENCES

- [1] Young-Jun Son, et al., "An extended BDI model for human behaviors: decision-making, learning, interactions, and applications," In Proceedings of the 2013 Winter Simulation Conference: Simulation: Making Decisions in a Complex World (WSC '13), IEEE Press. 2013; 401-411.
- [2] Yingxu W, Guenther R, "The Cognitive Process of Decision Making," *International Journal of Cognitive Informatics and Natural Intelligence*. 2007; 1(2):73-85.
- [3] Stan Franklin, et al., "LIDA: A Systems-level Architecture for Cognition, Emotion, and Learning," *IEEE Transactions on Autonomous Mental Development*. 2014; 6(1):19-41.
- [4] Wallach, W, et al., "Conceptual and Computational Model of Moral Decision Making in Human and Artificial Agents," *Topics in Cognitive Science*. 2010; 2(3):454-485.
- [5] John Martin R, Sujatha S, "Volitional Decision Making on Interactivity as a Result of Multi-cyclic cognitive Processes and Emotions," *Journal of Computing Technologies*. 2015; 4(10):2278 – 3814.
- [6] Thomas, et al., "Connectionist models of cognition. In Ron Sun (ed.)," *The Cambridge Handbook of Computational Psychology*," Cambridge University Press. 2011; 23-58.
- [7] Ilin, et al., "Cognitively Inspired Neural Network for Recognition of Situations," *IJNCR*. 2010; 1: 36-55.
- [8] Friston K., Park H, "Structural and functional brain networks: from connections to cognition," *Science*. 2013; 342(6158).
- [9] Dong D, Franklin S, "Sensory Motor System: Modeling the Process of Action Execution," Proceedings of the 36th Annual Conference of the Cognitive Science Society. 2014; 2145-2150.
- [10] Lebiere C, Anderson JR, "Cognitive constraints on decision making under uncertainty," *Front. Psychology*. 2011; 2(305).
- [11] John Martin R, Sujatha S, "Bottom-up Approach of Modeling Human Decision Making for Building Intelligent Agents," *Indian Journal of Science and Technology*. 2016; 9(4).
- [12] David E. Rumelhart, James L. McClelland, "Parallel Distributed Processing: Explorations in the Microstructure of Cognition," *Psychological and Biological Models*, MIT Press, Cambridge. 1986; 2.
- [13] Manas Singhal, Maitreyee Dutta, Manish Trikha, "Signature Verification using Normalized Static Features and Neural Network Classification," *International Journal of Electrical and Computer Engineering (IJECE)*. December 2016; 6(6): 2665-2673.
- [14] Arpad Kelemen, Yulan Liang, Stan Frankli, "Learning High Quality Decisions with Neural Networks in Conscious Software Agents," *WSEAS Transactions on Systems*. 2005; 9(4):1482-1492.
- [15] Narasinga Rao M R, et al., "Predictive Model for Mining Opinions of an Educational Database Using Neural Networks," *International Journal of Electrical and Computer Engineering (IJECE)*. 2015; 5(5):1158-1163.
- [16] Fodor, Jerry A, et al., "Connectionism and cognitive architecture: A Critical Analysis," *Cognition*. 1998; 28:3-71.
- [17] G. Á. Werner and L. Hanka, "Tuning an artificial neural network to increase the efficiency of a fingerprint matching algorithm," IEEE 14th International Symposium on Applied Machine Intelligence and Informatics (SAMII), Herlany. 2016; 105-109.