24 April 2017

A Model of How the Brain Learns

John H Andreae

A Short Introduction to
Multiple Context Associative Learning (MCAL),
and the PP System.

History

It is known that brains use networks of neurons to learn and that neuroscientists are studying where different things are learned in human and other brains. The model to be described here is not made of neurons but if it turns out to be the right sort of model, neuron net versions of this model should be helpful in understanding how the human and other animal brains learn.

The model is called PURR-PUSS, or PP for short, and it is to be realized in computing hardware, suitable for installing in a robot as the learning part of a brain for that robot.

Of course, we are now hearing on a regular basis how **artificial neural networks** (**ANN**s) are performing statistical functions important for the operation of the human brain. In particular, **deep learning** has had dramatic successes with statistical tasks like speech and face recognition, but Daniel Dennett[1] has reservations: "... although they will give us great answers to hard questions like never before, *they won't be able to tell us why."* (p.316)

Surprisingly, I am not aware of any system that makes PP obsolete, even though it was invented 43 years ago in 1974[2] with two lasting ideas: **multiple context** and **novelty goals**. The first idea was helped by theoretical work of my postgraduate student John Cleary and the second idea was a result of my attempt to get the system to explore its world.

A pervasive idea in the field of **artificial intelligence** (**AI**) since the 1960's has been that the brain is under the control of a very elaborate computer program with learning as a secondary process governed by that program. In my first book *Thinking with the Teachable Machine,*[3] I compared the programs of AI at that time with the water system in a man-made house, while claiming that PP was more like a river flowing down to the sea. PP has **no top level program** that determines what it does. PP can be taught by reward and punishment to do what others want it to do (**reinforcement learning**[4]), but when reinforcement is withdrawn it continues to use what it has learned while seeking its own novelty goals. Now that Bayesian predictive processing (also called PP by Andy Clark[5]) has taken centre stage in AI, my PP is no longer out on its own but my **multiple context** system isn't just statistical: it has a computational base with a statistical overlay. In AI, learning is still something to be optimized by exposure to large amounts of data with particular tasks in mind. However, with robots needing to be able to respond immediately and intelligently to new stimuli, such as words and objects, ANNs with a computational base must be just around the corner. Perhaps my PP will help in their design.

Another key argument in my 1977 book[3] referred to the limitations of formal systems shown by Gödel's Theorem and explained the importance of embodiment for human and robot intelligence:

> Now, every machine, robot or computer *on its own* (isolated from its surroundings) is a formal system and will never be able to do some of the things that we can do. ... Sometimes people think of their bodies as cages within which they live, but bodies are quite the opposite of cages. My body connects me to the world around me in such an intimate and interactive way that I become part of that world. ... It is not possible at the present time to give PURR-PUSS an adequate body for intimate interaction with the world around her, so Teacher has to provide the missing link. By giving PURR-PUSS a close, interactive link with Teacher, we defer the need for an adequate walking-and-talking body until such time as we can provide it. (p.6)

40 years later PP is still waiting for someone to provide it with an adequate robot body!

Templates

There is no question that we use context[6] in our deliberations, but there is so much context for whatever we do that there must be some limit on how much context we use. My assumption has been that the human body with all its sensorimotor equipment has evolved over millions of years in a mostly rural environment, so the context used by the human brain is largely determined by that sensorimotor equipment and the tasks it has been used for. George Miller's[7] famous 7 ± 2 for the number of items we have immediate access to, suggests how far back in time we need to go when including context for our immediate decisions. Our sensors and muscle movements limit what is available to us in those 7 ± 2 time-slots, but this is still an enormous amount of context, so we need to take into account the pre-processing done by (e.g.) statistical neuron nets in our visual, auditory and muscle control systems, and confine our attention to high level sensorimotor information such as recognized sounds, composite motor commands and the results of object and face recognition. PP has a **Short Term Memory** (**STM**) for holding this high level sensorimotor information.

Multiple context in PP is made up of hierarchies of **templates** for each **event-type**. In my latest, and probably last, experiment[8] the simulated robots had 9 action event-types (Body Move, Speech and Hand Pointing were the main ones) and 32 stimulus event-types (including Hearing, Touch, and various Visual event-types saying which robots were pointing at what and giving the relative positions of robots and other items in their small World). An event-type may refer to a specific collection of events, like BodyMove which included the events Forward, Left, Right, Pat and Slap. Or it may, like the Speech event-type, include an unlimited number of events (Words). PP had 67 templates for these event-types, 32 for predicting actions and 35 for predicting stimuli.

Templates are central to the structure of PP, because they <u>prescribe what contexts can be formed</u> as well as prescribing the <u>event-type of events that can be associated with the contexts.</u> By 'associated' I mean 'immediately following'[9]. A context plus associated event is an **association.** Since any context may be followed by different associated events on different occasions, there may be more than one association with the same context. (Later, we will see that action-predicting associations with the same context are lumped together in a node of a network.) The order of the event-types in a context is arbitrary: a context is just a collection of event-types. Here is a simple template:

Formal description of template: [SW: Speech,2 Speech,1 >> Speech]

Name of template: SW

Context of template: Speech,2 Speech,1

Symbol for "predicts" : >>

Associated event-type: Speech (It could be written Speech,0)

Speech,2 is the Speech event-type of an event 2 steps earlier than the associated event.

Speech,1 is the Speech event-type of an event 1 step earlier than the associated event.

Example of an association: [SW11: "A" "Short" >> "Introduction"]

TEMPLATES use EVENT-TYPES	[SW: Speech,2 Speech,1 >> Speech]
to prescribe	
ASSOCIATIONS using EVENTS	[SW11: "A" "Short" >> "Introduction"]

This template SW is prescribing two processes: how to store or strengthen an association and how to use an association that is already stored. Storing and using occur at different stages of PP's processing cycle, so the two cannot be confused. For storing, the template says " If two consecutive Speech events occur and they are followed by another Speech event, store the latest Speech event in the context of the previous 2 events as an association. If the association has already been stored with that associated event, strengthen it. If the association has been stored but not with that associated event, store a new association with the same context and the new associated event." For using, the template says "If two consecutive Speech events occur and they already exist as the context in an association of this template, then predict that the associated Speech event will follow next. If more than one associated event has been stored with the context of the two consecutive Speech events, predict that the **preferred associated event** will follow (using extra probabilistic data stored with the associated events, as will be explained later)." Associations are stored in **Long Term Memory** (**LTM**).

An association stored by the template SW can be seen as the learning of a **rule** "if the context occurs then do or predict the associated event", but the rule has to be seen as just one rule in a collection of rules, with some cooperating and some competing. This storage and use of associations is a form of **associative learning.** To distinguish a system like PP from earlier forms of associative learning, such as the conditioned reflex, it is called **multiple context associative learning** (**MCAL**).

It was mentioned above that PP had no top-level program to control it, so how is it controlled? In my view, the answer to this question is an important feature of PP. This is what makes it a river-like system. PP is **controlled by its associations** and it starts with none! It starts with a **blank slate** or *tabula rasa* attributable to John Locke[10] in 1690:

Let us then suppose the mind to be, as we say, white paper void of all characters, without any ideas. How comes it to be furnished? Whence comes it by that vast store which the busy and boundless fancy of man has painted on it with an almost endless variety? Whence has it all the materials of reason and knowledge? To this I answer, in one word, from EXPERIENCE. In that all our knowledge is founded; and from that it ultimately derives itself. Our observation, employed either about external sensible objects, or about the internal operations of our minds perceived and reflected on by ourselves, is that which supplies our understandings with all the materials of thinking.

Locke clearly envisaged the importance of learning for intelligence and anticipated many of the ideas that have driven the design of PP.

Of course, you don't have to start PP without associations, but learning is the easiest and probably the best way to acquire associations. Even if not started with a blank slate, PP will need reflexes, or a more primitive brain, or something to imitate in order to keep it going in its world when its associations cannot select actions. PP has plenty of inbuilt or innate structure, including the templates, (but no associations when it starts) so it isn't touched by Steven Pinker's arguments against a blank slate in his book of that name.[11]

The next question is "If the associations are in control, where are the associations going to take PP?" Certainly we can give PP reward and punishment, which I prefer to call approval and disapproval because the reward and punishment are volatile, disappearing if they are not continued. But if PP does no more than seek approval and disapproval given to it by a teacher or whatever, then that approval and disapproval is controlling PP through the associations. If that happened all the time, PP would be a reinforcement learning system and it would lack all the interesting features of our own brains, the curiosity and creativity that make us special.

At first, my intention was to enable PP to explore its world in an interesting way, when it wasn't being given approval and disapproval. The method I hit upon was to make any new association a novelty goal and then, when PP used that association again, to stop it being a novelty goal. A novelty mark was put on any association stored for the first time and the mark was removed when it was used again. What surprised me was that it made teaching PP with approval easier because when I had taught part of a task and went on to the next part of the task, PP no longer reacted to the removal of approval for the first part. Novelty goals in the next part took over from the removed approval until approval was given at the end of that next part. Approval used to teach something wasn't needed to keep PP using what was learned.

Soon I came to realize that novelty goals were doing much more than just making teaching with approval easier. Not only did they help to integrate new associations into PP's LTM, but they were goals made by PP's interaction with its world through its own body, not goals given to it by an external agent. These were PP's **own** goals even if most of them would be of little use. Now PP could be controlled by its own associations and its own goals. It would still need some teaching by approval, by disapproval and through imitation, but it could also branch out on its own, if allowed to. Here was the germ of free will and creativity, even if they were still a long way off.

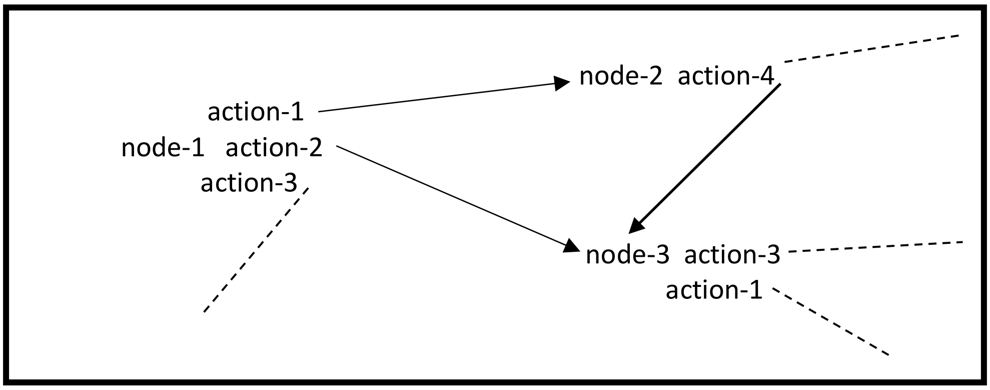
Acting in the World

While PP is deciding on its actions by means of contexts of events from its body and world, it does not need associations that predict what stimulus events the body and world will provide next. The body and world are supplying them. Only when we consider planning, imagining and thinking to oneself will associations that predict the stimuli from its body and world be needed.

PP has two kinds of goal. I have talked about the novelty goals. The other goals are approval goals. If approval is given just after an action is chosen by an association, that action in the association is marked as an approval goal. Similarly, if disapproval is given, the action is marked as disapproved. If several associations contribute to the choice of action, the actions in all the contributing associations are marked as approval goals. PP is designed to try to reach the contexts of associations which have actions marked as goals. How does it do this?

Obviously, PP has to have routes to its goals and must choose the most promising route.

When PP moves from having the context of one association in its STM to having another context of the same template in its STM by means of the associated action of the first association, that associated action has the number of the second context attached to it with a count of 1 to show that the transition has occurred once. The count is increased by 1 every time the transition is repeated and it is used to calculate the **transition probability** of that transition. Because of these transitions between contexts of a template, a network of contexts is formed, where **nodes** are contexts and **connections** are the transitions caused by associated actions. The diagram has 3 nodes and 7 connections from part of an imagined network.



Because there can be several associations with the same context, a node, which represents a context, will have as many actions as there are associations with that context. So node-1 in the diagram has 3 associations with the same context, node-2 has 1 association and node-3 has 2 associations with its context.

Some of the actions on nodes will be marked as goals, either approval or novelty goals. Some may be marked as disapproved. When PP has a particular context in its STM corresponding to one of the nodes in a network for a particular action event-type, it needs to know which action on that node to prefer, if there are several. This is where Bayesian conditional probability calculations made by a process called **leakback** give each node the **expectation of reaching a goal** from that node, and give each action on that node the expectation of reaching a goal by choosing that action with the context of that node. Leakback starts from all the goals and works backward through the network until either a maximum depth to the calculation has been reached or the expectations are below a minimum limit. This is not the place to bother the reader with the details, which can be found in my 1998 book *Associative Learning for a Robot Intelligence*[12]. Computer scientists may well have more efficient algorithms than mine for calculating the expectations.

The associations are rules which give PP its computational base. The network of transition probabilities give PP a statistical overlay. A computational, rather than statistical, base is needed for the complex task of using language because we learn new words and phrases from a single exposure to them. Statistical processes require many samples of their data. Unlike statistical processes, MCAL results can be analysed down to the last association.[8]

So far, I have been talking about one template and the network of nodes formed by it. The main idea of multiple context is to make available to PP contexts from throughout the action-stimulus-delay space for delays up to about 8 steps in the past.

Multiple Context

If PP is considered to be a model of the human or animal brain, then the choice of templates is assumed to have been selected by evolution. In this case, it may be helpful to imagine that a template is an area of the neocortex (or perhaps the hypothalamus) with dendrites entering the area from sources of the event-types of the context, and axons exiting to a command centre for the associated event-type of the template. A neuron in the area, behaving like an association, would carry out the AND function needed to identify a particular context of events on its dendrites and its axon would travel to the particular action event in the command centre. This structure should be treated as no more than a rough mental aid until one day neuroscientists discover exactly how the neurons in the brain learn.

Imagine a large graphic table with a column for the each action event-type and each stimulus event-type and a row for each number of steps delay. This table would describe STM, which is the space of event-types and delays available to the contexts of templates. We would like the templates chosen to have contexts spread over this space, with the longer delays covered more thinly. Here is a much smaller table with two possible contexts of many, indicated by blue and red blobs. Each of the two contexts would belong to a different template. Remember the auxiliary event-type has zero step delay because it refers to an event occurring now.

Event-type:	Speech Action	Speech Stimulus	Hand Action	Body Move	Eye Move	Vision Stimulus	Touch Stimulus
One step delay:	🔵		🔴	🔵	🔴	🔴	🔵 🔴
Two steps delay:		🔵				🔴	
Three steps delay:						🔴	
...							

Not all contexts are of equal "strength". In general, those contexts with more event-types will be stronger than those with fewer because every event-type restricts what situations the context can apply to. A context with no event-types is out of the question because it would include every situation. A template with the blue context in the table allows associations with contexts containing just one particular speech action, one speech stimulus 2 steps back, one body move and one touch stimulus. The contexts of the associations would ignore other events and delays.

At this point, the reader could well be confused by my two uses of the word "context". The context part of a template prescribes event-**types** and their delays, which the **events** in the context of an association of that template must satisfy. The template might say that a speech action delayed by 2 steps was required; an association of that template must have an actual speech action obtained with that delay. I won't usually need to distinguish between template contexts and association contexts because with one I will be talking about event-types and with the other I will be talking about events.

We need to give templates with stronger contexts higher priority so that PP will prefer to use their associations over the associations of templates with weaker contexts. There is no doubt that if one context is a subset of another context, the former will be weaker than the latter, and must be given a lower priority than the latter. When the contexts share few event-types, but appear to be of similar strength, they can be given equal priority. In choosing actions, PP tries the highest priority templates first and goes to lower priority templates only if it can't find associations of the higher priority templates that tell it what action to choose. However, once the choice has been made, all associations with the current context are updated with the choice.

The templates of PP should be chosen so that a wide range of tasks would be learnable in the particular body and world of the robot, for which PP is to be the learning brain. With a real robot in the real world, I would expect the choice of templates to take a considerable amount of experimentation. So far, I have had experience only with simple simulated robots and worlds.

Event-types are not restricted to action and stimulus types. The delay of a **threading** event is increased only when the next event is not null. Null events don't count as an event at all. I have also experimented with an event being the whole context of a template. If the template of such a **context event** is its own template, the contexts become **re-entrant**. Discussion of such variants is not appropriate in this short account, but it is important to note that multiple context is open to many avenues of research.

Since there can be several networks of nodes predicting actions of the same type, there is a need for standardizing the expectations obtained by leakback in the different templates. I see this as being done statistically when PP can be run for very long interactions.

Speed and Memory

An important aspect of multiple context is the way associations are stored in LTM so as to provide fast processing. Each template is seen as having its own hardware processor and memory, so that increases in the number of templates don't slow down the overall system.

All processing is in a forward direction through associations, which means that we never need to work backwards from an associated event to find its context. In fact, contexts of associations don't need to be stored at all. The context of an association is just an address of, or pointer to, the node that holds the associated events for that context. The node is reached by "hashing" the context in STM as specified by the template. My first experiments with PP did, in fact, use hashing of contexts[3], but in developing the system it was important for debugging that the contexts could be checked, so I now use tree structures, instead of hashing, to go from the events of a context to the nodes of a template.

As robots are given more complex bodies, they will also be given higher-level event-types, so it may well be that templates with contexts of no more than 20 event-types and delays less than 9 steps will be sufficient. 1000 templates, each with memory for 100,000 nodes, could be taking us into a region comparable with the capacity of the human brain.

Until PP is put into an effective robot body and long interactions become possible, it is difficult to anticipate the problems and opportunities that might ensue.

Planning and Imagining

The LTM of PP holds its past experience. The action-predicting templates were used to generate that experience, while the stimulus-predicting templates gathered information about how PP's body and world reacted to those actions. The associations generated by the stimulus-predicting templates do not form networks of nodes leading to goals because goals are irrelevant for them but, for convenience, groups of stimulus-predicting associations with the same context are still called nodes. These nodes don't have expectations but do have **ages** which count the number of times the context has occurred and the number of times each predicted stimulus has occurred after the context.

PP attempts to make a **plan** every step unless it is **following** a plan. To make a plan, STM is copied into a **spare STM** and PP chooses an action using the spare STM in the same way as it chooses actions normally. Then with the updated spare STM and the stimulus-predicting templates, it finds the most likely stimuli from the body and world. Then, another action is chosen, more stimuli are obtained, another action is chosen, ... and so on until a goal is reached or predictions fail. If a goal is reached, a plan has been made and the plan can now be followed using marks placed on the action-predicting networks during the plan-making. In one of my experiments[13], when a plan was followed the associations responsible were put into a **belief memory,** to record that the associations had been confirmed. In that same experiment, PP was given a **trail memory,** which gave it an extended present, looking a few steps backward and forward, so that it had a better idea of where it was.

Moving through PP's experience in LTM with a spare STM, as is done in planning, could also be done for other reasons, such as answering questions or just thinking and imagining. I have also suggested that spare STMs could be used as a form of **working memory**, **WM**.

Experiments and Theory

Novelty goals, the reaction of the world, and the volatility of teacher-given approval and disapproval make a mathematical description of the behaviour of PP difficult – beyond me, anyway.

In my paper about PP with John Cleary[14] in 1976, we showed quite simply that PP could be taught to behave like a finite state machine (FSM). This is not surprising because a FSM follows a collection of rules which say "If you are in this state and you get this input then go into that state." To teach PP to behave like a FSM, we needed (i) an action-predicting template with a context comprising a stimulus event-type saying what state the FSM is in, and a stimulus event-type saying what the input is, together with an associated action event-type to move it into the next state, and (ii) a stimulus-predicting template with a context of the event-type for a state-to-state action and an associated stimulus event-type saying what the new state is. We could then lead PP through all the state transitions of the FSM, giving it the appropriate state, input and state-to-state action. PP would then have learned the FSM and all we had to do was supply the inputs for it to move through different states by itself. After PP does each state-to-state action learned with the first template, it predicts the new state with the second template.

The FSM is not very interesting as such, because it is known that language use requires more computation power, possibly up to the maximum which is the **Universal Turing Machine** (**UTM**). Now a UTM has two parts. First there is a FSM controlling a read-write head, and secondly there is an infinite tape that is read and written on, symbol by symbol with the read-write head. In 1981, Bruce MacDonald and I published a paper[15] that showed PP being taught and then emulating a UTM. Of course, there was no infinite tape, but Marvin Minsky[16] had already pointed out that the important feature of a tape was that it could be extended. Even humans couldn't have an infinite tape in their heads! PP had an extendable tape, held by associations, which would only run out when there was no more room for associations or it couldn't count any higher "in its head". Both limits were themselves extendable. This was achieved with 11 templates.

Another experiment[17] which was carried out for its theoretical implications in 1990, when computer memory and speed were still severely limited, showed that PP could handle nested subroutines and embedded clauses. To achieve this, we introduced an **auxiliary action,** which is an action not used to carry out the main task. *Raise Eyebrows* was the auxiliary action used for the experiment that was about solving a 5-puzzle. The human face has plenty of muscles so it is not unreasonable to expect these to be of use as auxiliary actions, especially as they are visible to a child learning by imitation. The auxiliary action was used to mark the contexts when a new subroutine (B, say) was being entered and when it ended. Of course, information has to be held about the subroutine (A, say) being left, when the lower subroutine B is entered, so that the subroutine A can be continued after return from the lower subroutine B. Holding this information is the responsibility of **working memory (WM)**.

My latest report[8] on PP is mainly about WM. First recognized by Bruce MacDonald[15], if an association keeps only the most recent of its associated events, it can be used, for example, for holding the most recent symbol in a particular position of a Turing Machine tape. It is called a **recency** association to distinguish it from a **choice** association that keeps all its associated events. (In fact all associations can be recency or choice if the associated events are put on a list with the most recent one always at the front.) The tape of a Turing Machine is its WM. The importance of WM was well illustrated by a sentence offered by Koenraad Kuiper[18] in 1980 as a challenge to PP's learning of language:

It was her Mini which the police believed Andrea to have been driving, wasn't it?

Koenraad points out that at least 8 words have to be carried along in WM by any system understanding the sentence. Most obviously, the first word determines the last and the negation of the second word 'was' determines the one from last. It is a long time since I designed my first learning machine, STeLLA[19], in 1962 and it is going to be a few years yet before a robot can learn from a blank slate to handle a sentence like Koenraad's.

The report on WM[8] describes an interaction between two similar robots, one being controlled by PP and the other by a teacher (me) in a small World. There is also a Cake object in the World which can be Squashed by one of the robots, if they get the Cake into a corner. PP started with a blank slate and has learned a lot but is still making plenty of mistakes. This conversation at the end of the run is luckily free of mistakes:

993	PP:	(turns left) *I turned left.*
	Tchr:	(moves forward) *Good. You push cake.*
994	PP:	(moves forward, pushing cake) *Next turn is mine so you move forward.**
	Tchr:	(moves forward) *Good. Next turn is yours so you push cake.*
995	PP:	(moves forward, pushing cake) *Next turn is mine so you wait.*
	Tchr:	(waits) *Good. Next turn is yours so you squash cake.*
996	PP:	(moves forward, squashing cake) *I squashed cake so next turn is yours.*
	Tchr:	(turns right) *Good. You squashed cake so next turn is mine.*

* PP has remembered, using WM, that the last squash on step 803 was by Teacher so it's PP's turn to squash.

Prospects

MCAL demolishes Fodor and Pylyshyn's claim[20] that *"This transition from associative to computational accounts of cognitive processes has the look of a true scientific revolution."* because PP is both associative <u>and</u> computational.

Associative learning doesn't need to be just statistical, like deep learning. When Dennett[1] says *"In effect, people can download and execute a virtual machine with no need for trial and error or* ***associative learning,*** *taking on hundreds of roles on demand, swiftly and reliably."* (p.303, my emphasis), 'associative learning', in the form of something like MCAL, may well be how people do it.

Perhaps the greater aptitude of humans for learning language, compared with that of other animals, which is attributed by many to an innate universal grammar (UG), is due to the equivalent (in MCAL terms) of having innate templates for sounds and gestures extending over a longer range of delays. The design of MCAL event-types and templates for modelling human language learning would benefit enormously from the multi-language experimental data that has accumulated for and against UG[21]. It would be particularly pleasing if a MCAL solution could satisfy researchers on both the innate side[22] and the usage-based (or learning) side[23] of the UG debate.

Acknowledgements

In addition to John G Cleary and Bruce A. MacDonald, referenced in this account, Brian R. Gaines and Ian H. Witten have supported my work over a long period. This Report wouldn't have been written without the encouragement of Stewart Wilson. My wife, Molly's, editorial pen is behind all my writing.

References and Notes

1. Daniel Dennett: *From Bacteria to Bach and Back. The Evolution of Minds.* W. W. Norton & Co. 2017.
2. John H. Andreae (ed.): *Man-Machine Studies,*ISSN 0110 1188, nos. UC-DSE/4 and 5. 1974.
3. John H. Andreae: *Thinking with the Teachable Machine,* Academic Press, 1977.
4. Richard S. Sutton & Andrew G. Barto: *Reinforcement Learning,* MIT Press, 1998.
5. Andy Clark: *Surfing Uncertainty,* Oxford University Press, 2016.
6. Anticipating what follows, a context is a selection of events from STM, the events in STM being derived from the sensory and motor information provided by PP's body acting in PP's world.
7. George A. Miller: *The Magic Number Seven, Plus or Minus Two,* Psych. Rev. vol 63 no.2 81-96, 1956.
8. John H Andreae: *Working Memory for the Asssociative Learning of Language.* Research Gate: DOI:10.13140/RG.2.2.21755.72485
9. Association is a forward or predictive process in PP, as is all the processing in PP except Leakback.
10. John Locke: *An Essay Concerning Human Understanding.* First published 1690. Abridged and edited by Raymond Wilburn, London: J.M.Dent & Sons Ltd, 1947, p.26.
11. Steven Pinker: *The Blank Slate*. Penguin Books, 2002.
12. John H. Andreae: *Associative Learning for a Robot Intelligence,* Imperial College Press, 1998.
13. John H. Andreae: *A Multiple Context Brain for Experiments with Robot Consciousness,* IEEE Trans. Autonomous Mental Development, vol.3 no.4 312-323, 2011.
14. John H. Andreae & John G. Cleary: *A New Mechanism for a Brain.* Int.J.Man-Machine Studies, vol.8, no.1 89-119, 1976.
15. Bruce A. MacDonald & John H. Andreae:*The Competence of a Multiple Context Learning System.* Int.J. of General Systems. vol.7, 123-137.
16. Marvin L. Minsky: *Computation: Finite and Infinite Machines.* Prentice-Hall, 1967, p.118.
17. John H. Andreae, ShaunW. Ryan, Mark L. Tomlinson & Peter M. Andreae: *Structure from Associative Learning.* Int J. Man-Machine Studies. vol.39 1031-1050. 1993.
18. Koenraad Kuiper: *PURR-PUSSful Language Learning,* in Man-Machine Studies, ISSN 0110 1188, ed. JHA, no. UC-DSE/16, 5-10. 1980.
19. John H.Andreae: *STeLLA: A Scheme for a Learning Machine.* Proc.2nd IFAC Congress, Basel, 1963. In Automation and Remote Control, ed. Broida. Butterworths, 503, 1969.
20. Jerry A. Fodor & Zenon W. Pylyshyn: *Minds Without Meaning.* MIT Press, 2015. pp.11-12.
21. Maria T. Guasti: *Language Acquisition: The Growth of Grammar.* 2nd Edn. MIT Press, 2016.
22. Noam Chomsky: *What Kind of Cratures Are We?* Columbia University Press, 2016.
23. Michael Tomasello, ed.: *The New Psychology of Language. I & II.* Psychology Press, 2014.