Procedia Computer Science

Volume 41, 2014, Pages 255–263

BICA 2014. 5th Annual International Conference on Biologically Inspired Cognitive Architectures

Modeling Mechanisms of Cognition-Emotion Interaction in Artificial Neural Networks, since 1981

Stevo Bozinovski
South Carolina State University.
sbozinovski@scsu.edu

Abstract
The paper describes modeling of cognition-emotion interaction implemented in a neural network named Crossbar Adaptive Array in 1981. The architecture was proposed to meet two challenges: solving the delayed reinforcement learning problem for neural networks, and building a self-learning system (no advice and no reinforcement from environment) based on a neural network. The architecture introduced computation of feelings, and their interaction with learning and decision making mechanisms. It also introduced genetic environment as provider of initial emotions to the neural network. Receiving initial emotions from the genetic environment, the architecture learns using the proposed emotion backpropagation mechanism. Some developments after 1981 are also discussed.

Keywords: Crossbar Adaptive Array, cognition-emotion interaction, self-learning systems, consequence driven systems

1 Introduction

There are various understandings of what emotion is, and how it is represented (e.g Plutchink 1962). The issue is similar to the issue of understanding intelligence, where one view of what intelligence is, is to say that intelligence is what is measured by the IQ test. Analogously, we can say that emotion is what the Emotions Profile Index (Plutchnik and Kellerman 1974) measures. Computer programs have been used to draw emotional facial expressions from such a test (Bozinovski et al 1991). However, this paper considers on a more fundamental question related to emotion, and that is modeling cognition-emotion interaction (Hudlicka 2004). The question has been addressed many times in psychology (Schachter and Singer, 1962), in artificial intelligence since 1967 (Simon 1967) and in robotics since 1981 (Sloman and Croucher 1981). In neural networks the issue was first considered in 1981, on which this paper will be mostly about.

Artificial neural networks were related to decision making since their proposal (McCulloch and Pitts, 1943). Their potential for pattern recognition and concept (class) formation has been studied since the early works on Pandemonium (Selfridge 1958). The learning process was modeled in a network named Perceptron since 1958 (Rosenblatt 1958, 1962). The array representation of neural networks was used since early work on Learning Matrices (Stainbuch 1961). The response-sensitive

Selection and peer-review under responsibility of the Scientific Programme Committee of BICA 2014

doi:10.1016/j.procs.2014.11.111

teaching methods were studied since 1962 (Glushkov 1962). The reinforcement learning was applied since 1973 (Widrow et al. 1973) when a classification of learning system was introduced, dividing learning systems into supervised, unsupervised, and reinforcement learning systems. Feelings and emotions in neural networks were introduced in 1981 (Bozinovski 1981a, 1981b, 1982a, 1982b), in a network named Crossbar Adaptive array (CAA) This paper will be focused on cognition-emotion interaction introduced in the CAA network.

2 Adaptive Networks Group, 1981: Delayed Reinforcement Learning Challenge

The idea of a *need of a cognition-emotion interaction in an artificial neural network* appeared in response to the challenge presented in 1981 in front of the Adaptive Networks Group (ANW) of the Computer and Information Sciences (COINS) department of University of Massachusetts at Amherst. The group pursued general motto stated by the Project Officer, Harry Klopf, of "building goal seeking systems from goal seeking components". The group was mainly focused on the concept of Reinforcement Learning, and in 1981 the challenge of Delayed Reinforcement Learning was stated as well as the challenge of Assignment of Credit. Until then, the group worked on systems with immediate reinforcement for an action taken. The challenge of delayed reinforcement learning is how to engineer a secondary reinforcement mechanism (Keller and Schoenfeld, 1950), actually a *reinforcement backpropagation,*, from a situation when reinforcement was received to the situation back in time when the corresponding action was taken. As the Assignment of Credit challenge, the problem is which neurons will be affected and which synaptic weights should be updated.

Two instances of the general delayed reinforcement learning problem were considered: the path-in-a-maze learning task (challenge presented by Rich Sutton) and the pole-balancing learning task (challenge presented by Chick Anderson).

Two approaches were taken and two different architectures were proposed. The architectures were named Actor/Critic (A/C) architecture and Crossbar Adaptive Array (CAA) architecture. The programming work and actual experiments were carried out by Sutton (for the AC architecture) and Bozinovski (for the CAA architecture). The dynamics of cart-pole balancing system was programmed by Chuck Anderson.

3 The Actor/Critic Architecture

The Actor/Critic architecture (Barto et al, 1983), shown in Figure 1, has two main functional units, named Associative Search Element (ASE) and Adaptive Critic Element (ACE). Functioning of the AC architecture can be described by following equations:

Associative Search Element (ASE)
Action computing part computes action y,

$$y = f(\Sigma w_i x_i + noise_i) \qquad (1)$$

where *f(.)* is some neural activation function, in this case a threshold function giving 1 for nonnegative and -1 for negative arguments. Here *noise*$_i$ is a *Gaussian* noise for ensuring random movement.
The learning part (memory update) is computed as

$$w_i(t+1) = w_i(t) + \alpha r\hat{}(t)e_i(t) \qquad (2)$$

where α is positive constant defining learning step size, $r\hat{}$ is the internal reinforcement, and e_i is the eligibility trace of the *i*-th input pathway, computed as

$$e_i(t+1) = \delta e_i(t) + (1-\delta)y(t)x_i(t) \qquad (3)$$

where *″, 0≤ ″<1,* is the forgetting parameter of this learning equation

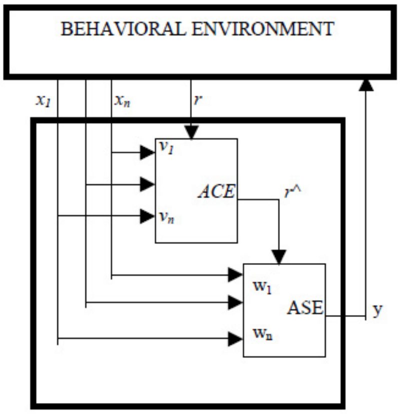


Figure 1. The Actor/Critic (A/C) Architecture (Sutton 1981)

Adaptive Critic Element
This element has structure of the same dimension as the memory of ASE, and the memory elements are denoted as v_i. Given situation x the *prediction of reinforcement* is computed as

$$p=\Sigma v_i x.$$

Using that reinforcement prediction, the internal reinforcement is computed as

$$r\hat{}(t) = r(t) + \gamma p(t) - p(t-1) \qquad (4)$$

where r is the external reinforcement given by the environment and γ (here γ =.95) is a discount factor for the predicted reinforcement.
The learning rule for this memory structure is a *second order equation*

$$v_i(t+1) = v_i(t) +\beta \,(r(t) + \gamma p(t) - p(t-1))\tau_i(t) \qquad (5)$$

where τ_i is a trace of the input x_i, with dynamics described as

$$\tau_i(t+1) = \lambda\tau_i(t) + (1-\lambda)x_i(t) \qquad (6)$$

where λ, $0\leq\lambda<1$, is the forgetting factor of this learning equation.

4 The Crossbar Adaptive Array Architecture

The objective of designing the Crossbar Adaptive Array (CAA) architecture was twofold:

1) *Build a self-learning system*, a system that will depend neither on external reinforcement nor external advice
2) *Solve the delayed reinforcement learning problem.*

In order to solve the self-learning part, a genetic mechanism was introduced. In addition to the Behavioral Environment, where the agent controlled by the CAA architecture behaves, a Genetic Environment was introduced, where from the CAA architecture inherits its initial memory. That memory is connected to situations in the environment by evolution, and gives intrinsic emotions, such as “if you feel cold it is unpleasant situation, avoid it”. The concept of “cold = unpleasant” is genetically built into the agent when it is created and when initial genetic string (chromosome) is received from the genetic environment. The CAA also is able to exporting the memory as a chromosome after learning, following the principle of Lamarckian evolution. Figure 2 shows the CAA architecture

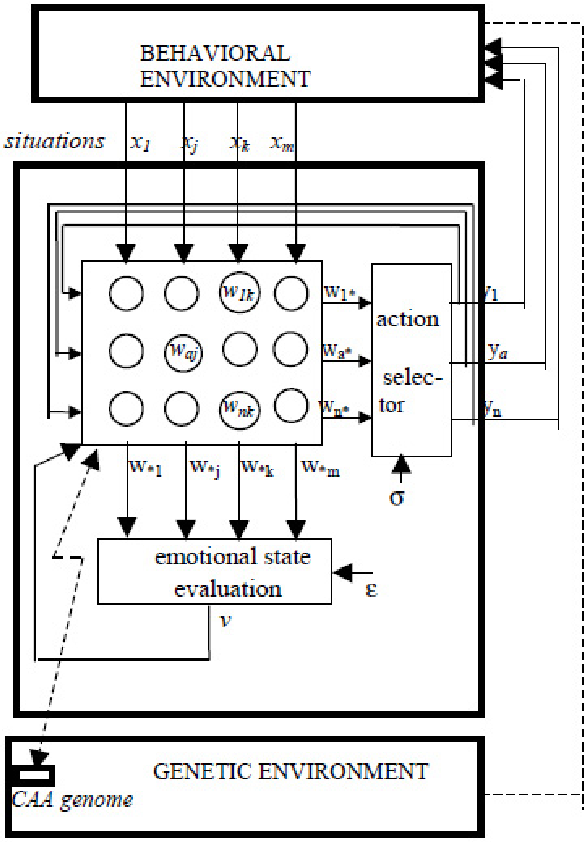


Figure 2. The Crossbar Adaptive Array (CAA) architecture (Bozinovski 1981)

The situations are received by the memory structure of the CAA architecture, which in crossbar fashion computes cognition (action decision and learning) and emotion. So the cognition-emotion interaction is computed in the same memory structure. The architecture represents an *artificial brain architecture* where both cognition part and emotion computing part (limbic system) share the same memory structure.

Decision making process, action computation

Being in situation x_j, ($x_j = 1$ all other inputs equal zero) CAA computes its action as

$$y_a = sgn(\max_{a=1,..,n} \{\Sigma_{j=1,..,m} w_{aj}x_j + \sigma_a\}) \qquad (7)$$

where σ_a is a random number from a uniform distribution between -0.5 and ,+0.5. It represents the nature of the CAA *searching strategy*. Before learning, this term is dominant and CAA searches for a path in the problem space, and after learning, this term does not affect the behavior. The function *sgn(.)* gives 1 for nonnegative and 0 for negative argument.

Emotion feeling process

After receiving the next situation, x_k, (x_k=1 others zero) which is *consequence of action a in a previous situation* x_j, the emotion in that situation is computed as

$$v_k = sign(\Sigma_{a=1,..,n} w_{ak}x_k - \varepsilon_k),$$

Since only one component in the input vector **x** is 1, the output emotion vector has also only one nonzero component. So the overall computation can be computed in several ways for example

$$v = \sum_{k=1,.,m} v_k = \max_{k=1,.,m} \{v_k\} = v_k \qquad (8)$$

In the above equations *sign(.)* gives 1 for positive, 0 for zero, and -1 for negative arguments. The *emotional sensitivity* parameter ε_k can be a random number or a number received by the initial genome, and it is used in environments where a system should learn to pass an unpleasant situation in order to reach a goal situation. It is a threshold after which the system senses emotion. Overall emotion in situation x_k depends only of the emotion obtained from the consequence situation $x_{k'}$.

CAA crossbar learning rule

The CAA has only one, learning equation:

$$w_{aj} = w_{aj} + v_k \qquad (9)$$

which, written as difference equation is

$$w_{aj}(t) = w_{aj}(t-1) + x_j(t-1)y_a(t-1)v(t) \qquad (10)$$

Equations (9) and (10) are two forms of the *CAA crossbar learning rule*, which is represented by only one, *first order* learning equation.

*T*he **CAA crossbar learning procedure** (pseudocode) in each time step has four activity steps

```
1) state $j$: compute action $a$ biasing on $w_{*j}$ , sent it to the environment
          the environment gives back the state $k$
2) state $k$: compute emotion $v_k$ using $w_{*k}$ . then compute overall emotion $v$
3) state $j$: increment $w_{aj}$ using $v$ (emotion backpropagation and learning)
4) change state: $j=k$; goto 1
```

5 The 1981 Experiments in Cognition-emotion Interaction

Both A/C and CAA architectures tried to solve two instances of the delayed reinforcement learning problem (or credit assignment problem). The two instances were path in a maze learning and pole balancing learning.

The instance of learning path in a maze

The A/C architecture and its approach toward solution of delayed reinforcement learning problem, was tested on the types of mazes from animal learning theory, shown on the left in Figure 3. The CAA approach used mazes motivated by the 1981 VAX/VMS computer game Dungeons and Dragons, with desirable and undesirable states, example of which is shown to the right side of Figure 3. While CAA approach used emoticons in representing states, the A/C approach did not use them, here it is added for denoting the goal state.

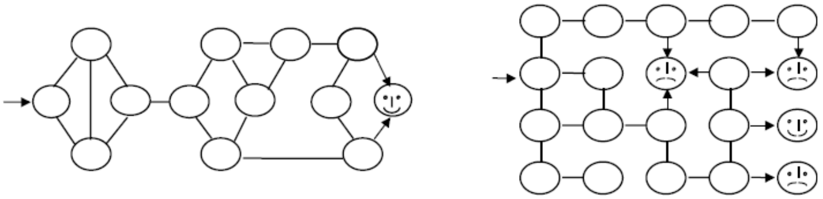


Figure 3. Examples of mazes considered by AC architecture (left) and by CAA architecture (right)

During 1981 the only architecture that solved the path in a maze learning problem was the CAA architecture (Bozinovski 1981a) and reported outside the ANW group in 1982 (Bozinovski 1982a, 1982b).

The instance of learning to balance a cart-pole system

The cart-pole balancing problem is a benchmark problem in control theory and artificial intelligence, and assumes a cart to which it is attached an inverted pendulum. The system is controlled by a force F which moves the cart left and right such that the pendulum stays in upward position. A controller receives information about angle (θ) and angular velocity (ω) of the pole, and applies set of actions {+F, -F}.

For this problem A/C approach used the Michie-Chambers representation (Michie and Chambers, 1968) with 162 environment states. It restricts the movement of the pole to a relatively small angle and also movement of the cart to a relatively small left-right discourse. Negative reinforcement is received by the controller if either cart or pole goes beyond the restrictions.

The CAA approach used simpler representation, with only 10 states, taking two assumptions:

1) there is no need to consider limits of the cart displacement *in order to demonstrate learning*; the force F is constant so it already is a limitation for displacement; it suffices to consider only the pole displacement optimization (between $-\theta_{lim}$ and $+ \theta_{lim}$), and

2) for set of action force, instead of using binary set {+F,-F}, use ternary set {- F,0,+F}

With those assumptions the pole balancing problem was represented as emotionally colored, nondeterministic graph of a state space as shown in Figure 4.

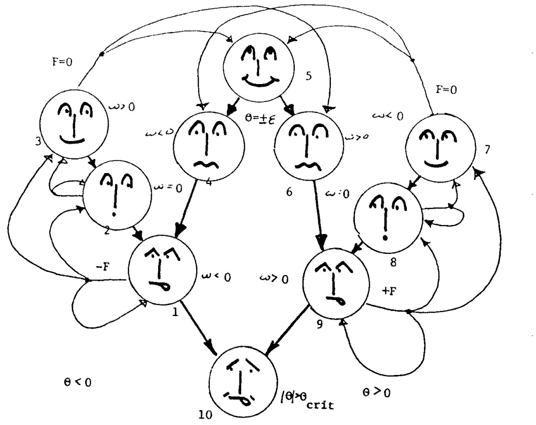


Figure 4. Emotionally colored state space for the CAA approach of the cart-pole balancing problem (1981)

With this design, a simple delayed reinforcement learning controller

$F = g(\theta, \omega)$,

where θ is angle and ω is angular velocity, was designed which learned a simple control heuristics:

*if sign*θ*sign*ω *< 0 then F = sign*θ*, otherwise F = 0*

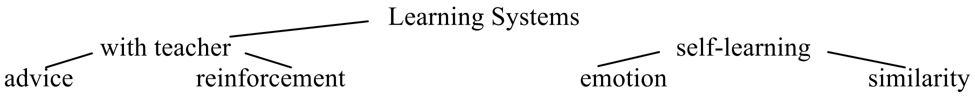
The negative reinforcement was received when the pole fall down, at angle of 90^0.

The CAA approach solved this problem in 1981 (Bozinovski 1981b) The AC approach solution was achieved after 1981 and was described in (Barto et al 1983).

6 Discussion: Contribution of the 1981 CAA Architecture to Cognition-emotion Interaction

This discussion is a 33 years (third of a century) look back on the work which introduced emotions and to explored cognition-emotion interaction in neural networks. .The following are the pioneering contributions in 1981 to cognition-emotion interaction:

- **Solving the delayed reinforcement learning problem** (learning with delayed rewards and learning with delayed punishments) and assignment of credit problem for neural networks
- **Introducing a working self-learning (emotion based learning) system.** Before that all the neural networks learned with external teacher who corrects their action errors, either by advising the correct action, of just giving reinforcement (action evaluation but not advising the correct action).
- **Introducing new taxonomy of learning systems,** described by the following tree:



- **Introducing interaction between cognition and emotion** built in a neural network. CAA computes in a crossbar fashion from the same memory structure both decisions for actions and evaluations (emotions) of consequences of actions, and learns by backpropagation of the emotion signal. The CAA learning rule explicitly contains emotion backpropagation w_{aj} = w_{aj} + v, where v is overall feeling of consequence of action y_a in situation x_j
- **Introducing genetic environment which defines initial emotions.** CAA is a neural network that includes the genetics environment, in addition to the behavioral environment. Initial emotions are genetically introduced to CAA agent from the genetic environment, and they contain emotional description of both dangerous and favorable situations in the behavioral environment.
- **Introducing a learning procedure (pseudocode), not just learning equations.** The CAA description (Bozinovski 1982b) was the first one that gave a pseudocode, procedure of learning
- Introducing the concept of **searching strategy** in neural networks, not a "noise" used before.
- **Introducing parallel programming in neural network implementation.** The CAA controller was running on one terminal (Bozinovski), in parallel with the cart-pole dynamics running on another terminal (Anderson). The inter-process communication was done via VAX/VMS mailboxes, which was programmed by Bob Heller.
- **Introducing a third action in modeling the control of a cart-pole balancing system.** Instead of {-F, +F} set of possible actions, the CAA experimental work introduced the three set actions {-F, no action, +F}. It simplified the control of the system.
- Introducing explicitly the concept of **state evaluation** and that way implicitly pointing out to relation to Dynamic Programming, eight years before explicit connection between reinforcement learning and Dynamic Programming was established. The CAA crossbar memory keeps only the action values, and computes the state values when needed.
- The CAA architecture was a **forerunner** (Barto 1997**)** and **precursor** (Barto 2007) of the **eight years later proposed Q-learning method** (Watkins, 1989). Eight years after CAA was proposed, Watkins (Watkins 1989, Barto et al 1990) proposed a learning procedure which basically uses the CAA approach, but explicitly relates the approach to Dynamic Programming, rather than to emotion. The Q-table is exactly the same as the CAA memory. The element w_{aj} in CAA has the same role as the Q(a, j) value in Q-learning. Although

Watkins proposed the Q-learning independently, two relevant factors should be mentioned 1) Watkins was participant of the European Workshop Session on Learning, which took place in Bled, Yugoslavia, 13-15 may, 1987, where Bozinovski presented his work on CAA architecture, and Watkins his work on search, which was not related to Q-learning. 2) Sutton who was office-mate with Bozinovski during 1980-1981, was influential in use of Q-learning as reinforcement learning method, he was one of the academic advisors for Watkins' PhD thesis in 1989.

- Introducing **Consequence Driven Systems,** pointing out the concept of consequence related to emotion (Bozinovski et al. 1996) and decision making (Barett et al. 1998)
- The CAA architecture approach pointed to motivation research in neural networks (Bozinoski 2002, Barto et al. 2004). It also was a reason for proposing a basic law of psychology (Bozinovski 2003).
- The CAA architecture is a first **artificial brain** architecture since it includes both a cognitive part of computing decisions for behaviors, and a part (limbic system) that computes emotion as evaluations of consequences of those behaviors. And both parts share the same memory structure.

Acknowledgement

The 1981 Adaptive Networks (ANW) Group of the university of Massachusetts at Amherst consisted of the following members: Nico Spinelli (Principal Investigator), Harry Klopf (Project Officer), Michael Arbib, Andrew Barto, postdoc at that time, and graduate students Richard Sutton, Charles Anderson, Ted Selker, Jack Porterfield, and Stevo Bozinovski. The work on neural networks project was funded by the Air Force Base in Dayton, Ohio. Informal discussions were often carried out with Eva Hudlicka and Marta Steenstrup. Animal Learning courses were taken by Sutton and Bozinovski with professor John Moore and Cognitive Psychology course on similarity was taken by Bozinovski with professor James Chumbley, both professors with Psychology Department. The author wishes to thank the ANW group for providing environment where the CAA architecture was proposed.

References

Barett C., Thord R., Reidys C. (1998) Simulations in decision making for socio-technical systems. in M. Beckmann, B. Johansson, F. Snickars, R. Thord (eds.) Knowledge and Networks in a Dynamic Economy, p. 59-82, Springer Verlag

Barto, A. (1997). Reinforcement learning. In O. Omidvar and D. Elliot (Eds.) *Neural Systems for Control,* p. 7-30, Academic Press.

Barto A. Sutton R., Anderson C. (1983) Neuronlike elements that can solve difficult learning control problems, *IEEE Transactions and Systems, Man, and Cybernetics 13*: 834-846

Barto A., Sutton, R., Watkins C. (1990) Learning and sequential decision making. In M. Gabriel, J. Moore (eds.) *Learning and Computational Neuroscience: Foundations of Adaptive Networks*, MIT Press, p. 539-602

Barto A., Singh S., and Chentanez N. (2004) Intrinsically motivated learning of hierarchical collections of skills. In Proceedings of the International Conference on Developmental Learning (ICDL) p. 112-119, San Diego

Barto A. (2007) Temporal difference learning. Scholarpedia 2(11):1604

Bozinovski S (1981a) A self-learning system using secondary reinforcement, ANW internal report, COINS, UMass, November 25. 1981

Bozinovski S. (1981b) Inverted pendulum learning control. ANW Memo, COINS, UMass, December 10, 1981

Bozinovski S (1982a) A self-learning system using secondary reinforcement, *Abstracts of the Sixth European Meeting on Cybernetics and Systems Research - 1982,* Abstract 87, Austrian Society of Cybernetic Studies, Vienna

Bozinovski, S. (1982b). A self-learning system using secondary reinforcement. In R.Trappl (Ed.) *Cybernetics and Systems Research,* p. 397-402, North-Holland.

Bozinovski S., Martinovska C., Bozinovska L., Pop-Jordanova N. (1991) MEXYS2: A fuzzy reasoning expert system based on the subject emotions consideration, In. K. Adlassnig, G. Grabner, S. Bengtsson, R. Hausen (eds.) *Medical Informatics Europe*, Springer Verlag

Bozinovski, S., Stojanov G., Bozinovska L. (1996) Emotion, Embodiment, and Consequence Driven Systems In M. Mataric (ed.) Embodied Cognition and Action, AAAI Fall Symposium, MIT, Cambridge, MA, FS-96-02, p. 12-17

Bozinovski S. (2002) Motivation and emotion in the anticipatory behavior of consequence driven systems. In M. Butz, O. Sigaud, P. Gerard (eds.) Proceedings of the Workshop on Adaptive Behavior in Anticipatory Learning Systems. Edinburgh, p. 100-119

Bozinovski, S. (2003) Anticipation driven artificial personality: Building on Lewin and Loehlin. In M. Butz, O. Sigaud, P. Gerard (Eds.) *Anticipatory Behavior in Adaptive Learning Systems.* p. 133-150, Springer Verlag

Cannon R. *Dynamics of Physical Systems,* McGraw Hill, 1967

Glushkov V (1962) A theory of instruction for a class of discrete perceptrons (in Russian) *Zhurnal Vychislitelnoi Matematiki and Matematicheskoi Fiziki* 2: 317-335

Hudlicka E. (2004) Beyond cognition: Modeling emotion in cognitive architectures. Proc. 6th International Conference on Cognitive Modeling, Mahwah, NJ p. 118-123, Lawrence Erlbaum

Keller F, Schoenfeld W. (1950) *Principles of Psychology,* Appleton-Century-Croffts

McCulloch W., Pitts W. (1943) A logical calculus of the ideas imminent to neural activity, *Bulletin of Mathematical Biophysics* 5: 115-133

Michie D., Chambers R. (1968) BOXES: An experiment in adaptive control. In E. Dale, D. Michie (eds.) Machine Intelligence 2: 125-133, Oliver and Boyd

Plutchnik R. (1962) *The Emotions: Facts, Theories, and a New Model*, Random House

Plutchnik R., Kellerman H. (1974) *Emotions Profile Index. Test and Manual,* Western Psychological Services

Rosenblatt, F. (1958) The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review 65*, 386–408

Rosenblatt F. (1962) *Prunciples of Neurodynamics,* Spartan Books, 1962

Schacher S., Singer J. (1962) Cognitive, social, and physiological determinants of emotional state, *Psychological Review,* 69, 379-399

Selfridge O. (1958) Pandemonium – A paradigm of learning, Proc Symposium on Mechanization of the Thought Process, HMSO, London, p. 513-526

Simon H. (1967) Motivational and emotional controls of cognition, *Psychological Review* 74, 29-39

Sloman A., Croucher M. (1981) Why robots will have emotions. International Joint Conference on Artificial Intelligence, Vancouver

Steinbuch W. (1961) The Learning Matrix (in German) *Kybernetik* 2: 36-46

Watkins C. (1989) Learning from Delayed Rewards. Ph.D. Thesis. King's College, UK

Widrow B., Gupta N., Matra S. (1973) Punush/Reward: Learning with a critic in adaptive threshold systems. *IEEE Transaction on Systems, Man, and Cybernetics* 3: 455-465