A-mesa-in-desert vs a-needle-in-haystack:

A fitness landscape on weight space of an application using spiking neurons under rate coding.

Akira Imada

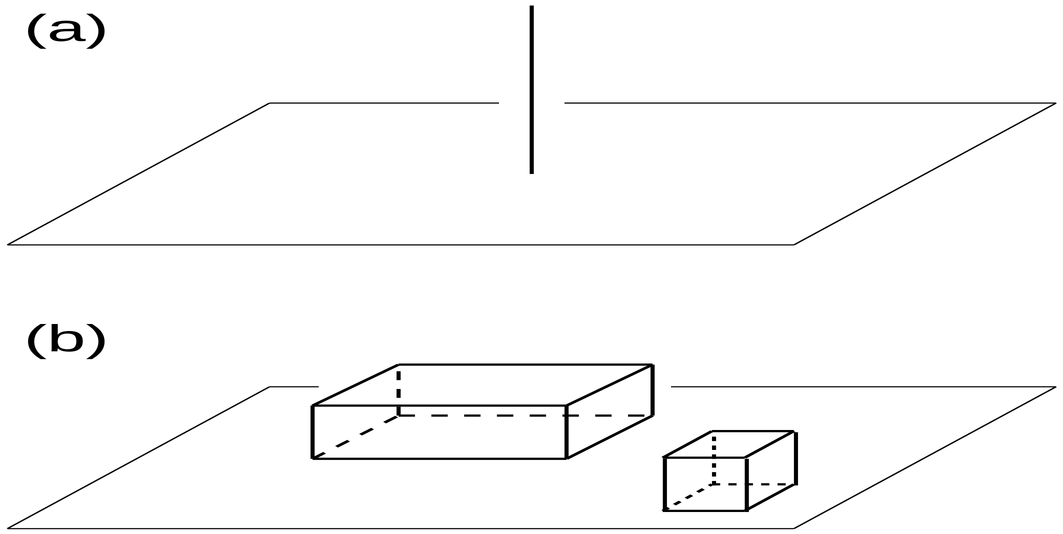
Brest State Technical University
Belarus

We came accross

an extremely difficult
PEAK
to be searched for

in a
FITNESS LANDSCAPE

explored by an
EVOLUTIONARY COMPUTATION


(a)
(b)

To realize an

ASSOCIATIVE MEMORY

we use

SPIKING NEURONS

whose learning is by

EVOLUTIONARY COMPUTATIONS.

□ Contents

(cont'd)

3. Results of our Experiments.

- **Downhill Walk**
- **Hillclimbing**
- **Hinton & Nowlan's Experiment of Baldwin Effect**
  - **Effect of Lifetime Learning**

4. Summary & Conclusions.

□Associative Memory:

- **stores patterns**
  - **in a distributed way (among neurons),**
- **recalls patterns**
  - **from noisy and/or partial input.**

$\Downarrow$ **i.e.**

- **gives us**
  - **perfect recollection from imperfect information**

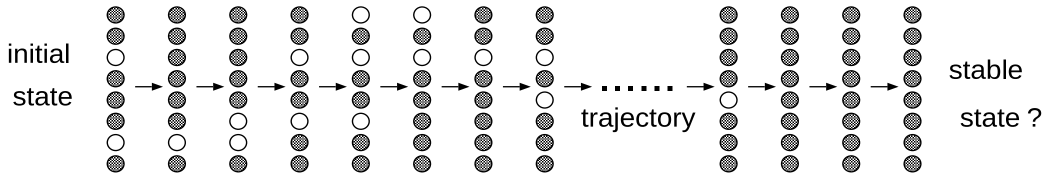
□The Associative Memory can be realized with:

- **Fully-connected Neural Network Model**

  (Hopfield type)
- **Artificial Immune System Model**
- **Spiking Neurons**

  (To be more biologically plausible)
- **etc.**

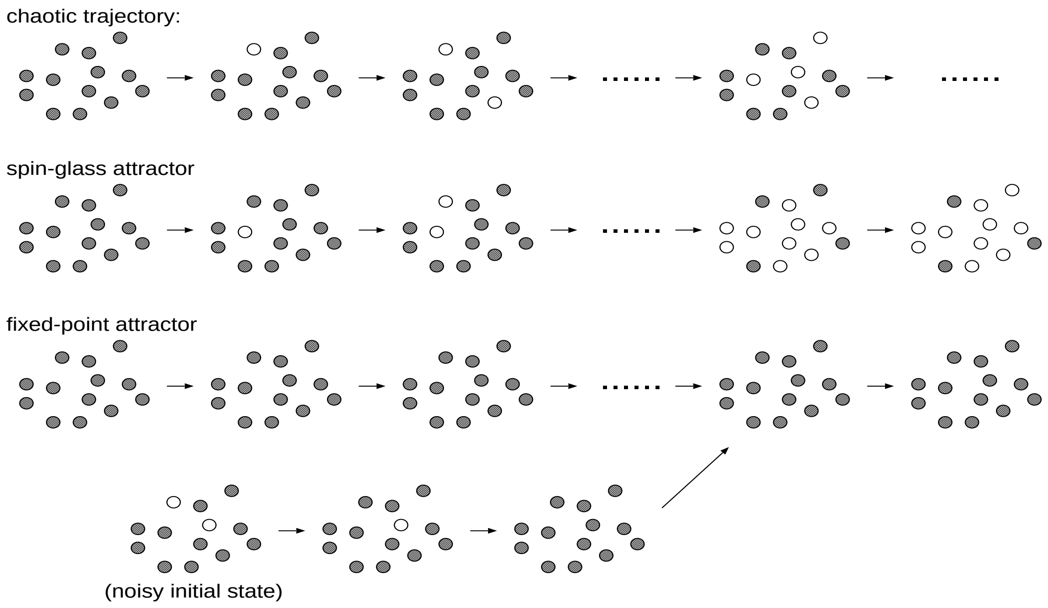
associative memory = dynamical system

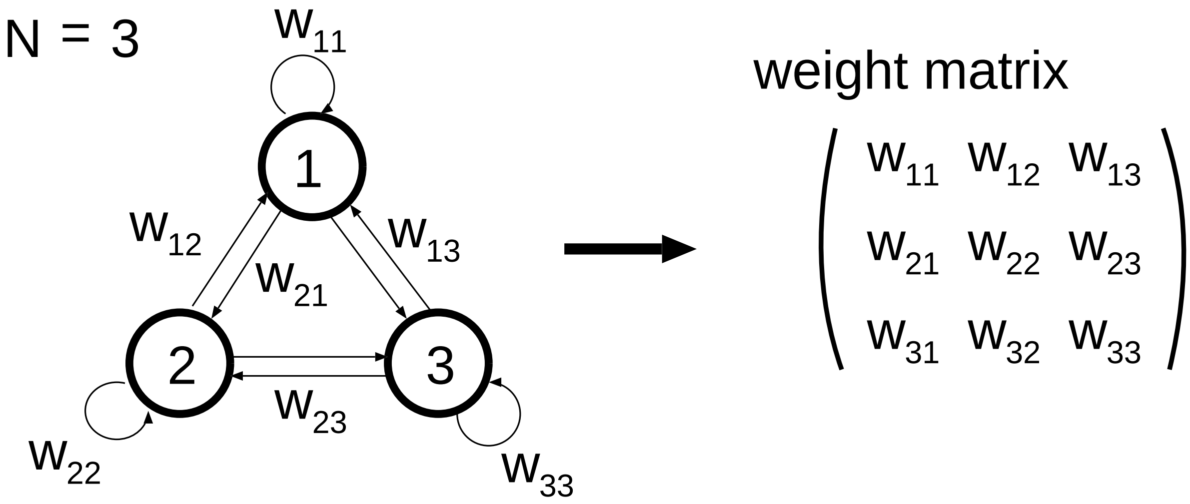


converge

initial state $\xrightarrow{\text{converge}}$ equilibrium state

u+du $\longrightarrow$ u


chaotic trajectory:
......
......
spin-glass attractor
......
fixed-point attractor
......
(noisy initial state)



N = 3
W11
W12
W13
W21
W23
W22
W33
1
2
3
weight matrix
W11 W12 W13
W21 W22 W23
W31 W32 W33

□ State Transition of the Hopfield Model

$$s_i(t+1) = sgn(\sum_j^N w_{ij} s_j(t))$$

□ Hebbian Weights

$$w_{ij} = \frac{1}{N} \sum_{\mu=1}^{p} \xi_i^{\nu} \xi_j^{\nu}, \quad (i \neq j), \quad w_{ij} = 0.$$

(To store p patterns: $\mathbf{x}^{\mu} = (\xi_1^{\mu}, \xi_2^{\mu}, \cdots \xi_N^{\mu}) \quad \mu = 1, 2, \cdots, p$)

□ Hopfield Model has a Crucial Drawback.

- N neurons store only $2N$ patterns at most (Gardner).

$$p < 2N.$$

□ From the Hopfield Model to Spiking Neuron Model

Influence from other neurons
via
ELECTRIC CURRENTS

$\Downarrow$

changes
MEMBRANE VOLTAGE
of the neuron.

(cont'd)

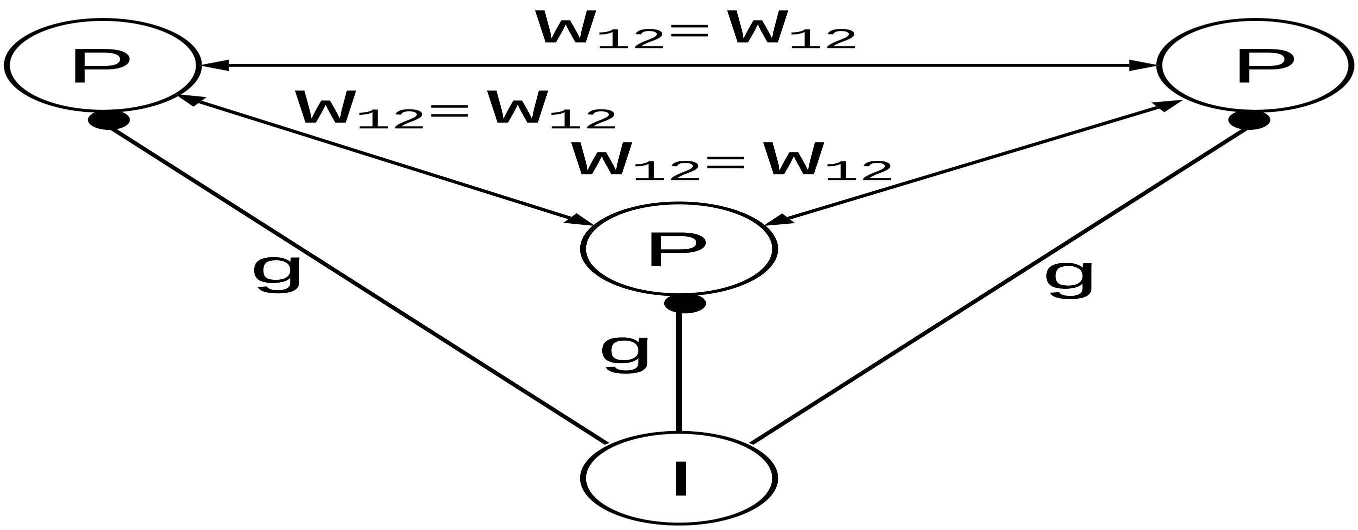
$\Downarrow$

If it exceeds a threshold

$\Downarrow$

the neuron emits
a SPIKE.

□ Pyramidal Cells & Interneurons



PYRAMIDAL CELL

$\|$
(positive current)
$\Downarrow$

PYRAMIDAL CELL

$\Uparrow$
(negative current)
$\|$

INTERNEURON

□ Response of a Single Neuron to an External Stimulus:

- **Spike rate vs stimuli (Naka-Rushton function)**

$$S(P) = \begin{cases} MP^n/(\sigma^n + P^n) & \text{if } P \geq 0 \\ 0 & \text{if } P < 0 \end{cases}$$

(cont'd)

- **Spike rate under time-variant stimulus:**

$$\frac{dr(t)}{dt} = \frac{1}{\tau}(-r(t) + S(P))$$

□ Response of Multiple Neurons

- **Stimulus to a Pyramidal Cell**

$$P_i = (\sum_{j=1}^{N} w_{ij} \cdot R_j - g \cdot G)_+^2$$

of

$$S(P_i) = MP_i^n/(\sigma^n + P_i^n)$$

(cont'd)

- **Spiking Ratio of a Pyramidal Cell**

$$\frac{dr}{dt} = \frac{1}{\tau}(-r + MP^n/(\sigma^n + P^n))$$

becomes

$$\tau_R \frac{dR_i}{dt} = -R_i + \frac{100(\Sigma_{j=1}^{N} w_{ij} R_j - 0.1G)_+^2}{100 + (\Sigma_{j=1}^{N} w_{ij} R_j - 0.1G)_+^2}$$

(cont'd)

- **Spiking Ratio of Interneuron**

$$\tau_G \frac{dG}{dt} = -G - 0.07 \sum_{j=1}^{N} R_i$$

□ How to encode patterns?

$$\xi_i = \begin{cases} 1 & \text{if } R_i \geq M/2 \\ 0 & \text{if } R_i < M/2 \end{cases}$$

$\Rightarrow$ Rate Coding

- Rate Coding
- Time Coding

□ GA Implementation:

1. Represent a series of w_{ij} as a *population* of strings (*chromosome*).

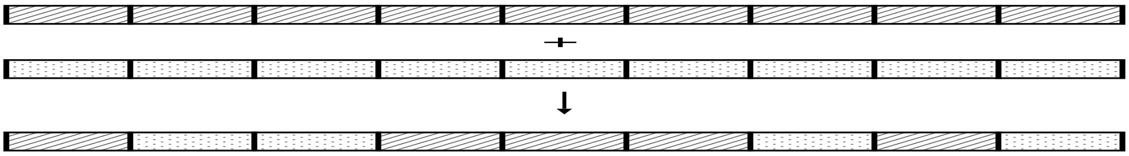
W_{11}	W_{12}	W_{13}	···	W_{21}	·········	W_{N1}	···	W_{NN}

2. Evaluate *fitness* by "How good is each individual?"

3. Generate an initial *population* at random.

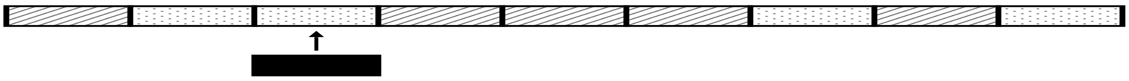
(cont'd)

4. Evolve them with
   - *Selection*
   - *Crossover*



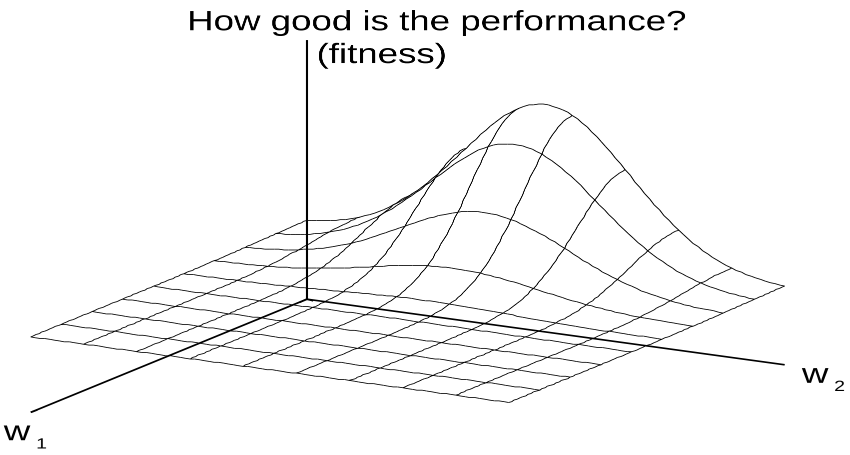
(cont'd)

– *Mutation*



5. Better Solutions from *generation* to *generation.*

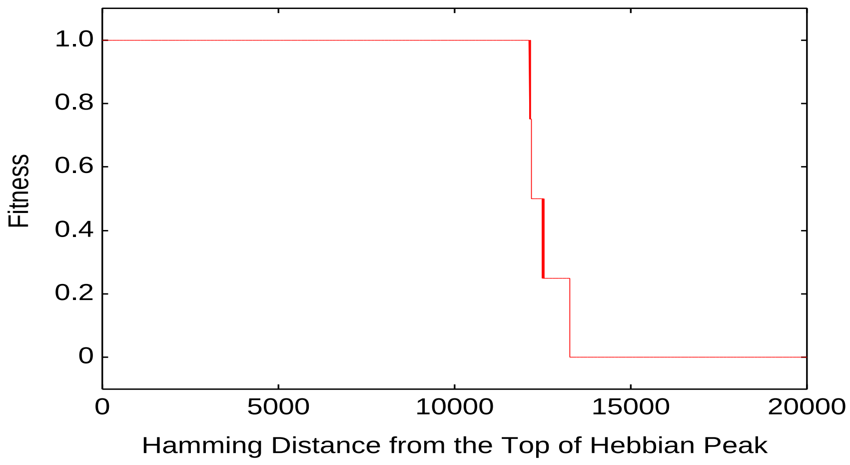
□ A Conceptual Illustration of Fitness Landscape



□ Weights by Hebb's Rule

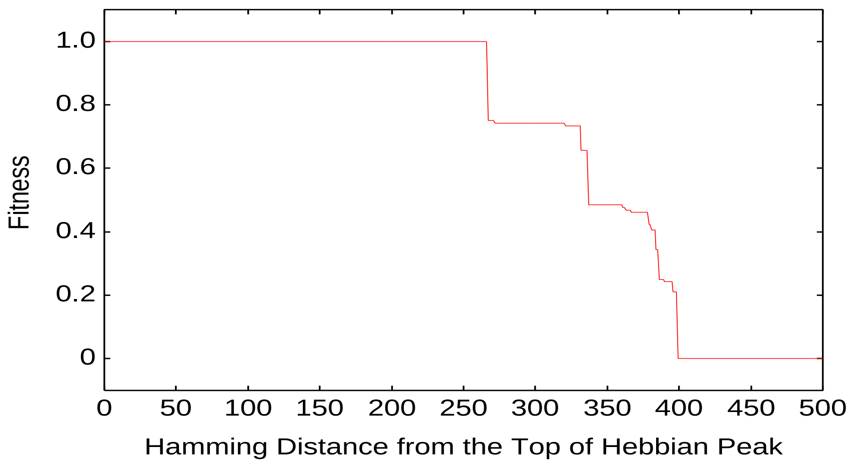
$$w_{ij} = sgn(R_i - M/2) \cdot sgn(R_j - M/2)$$

- **A Downhill Walk by Flipping zero to one:**



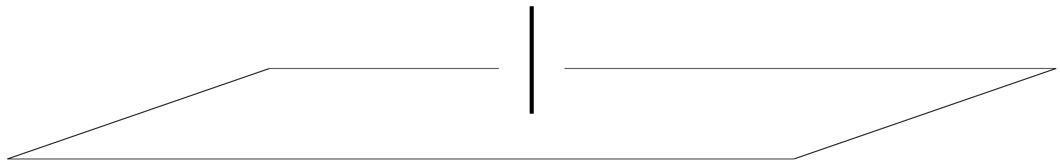
(cont'd)

- **A Downhill Walk by Flipping one to zero:**

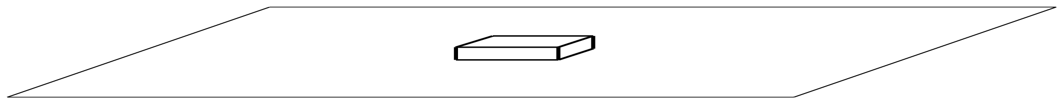


□ Search for a mesa

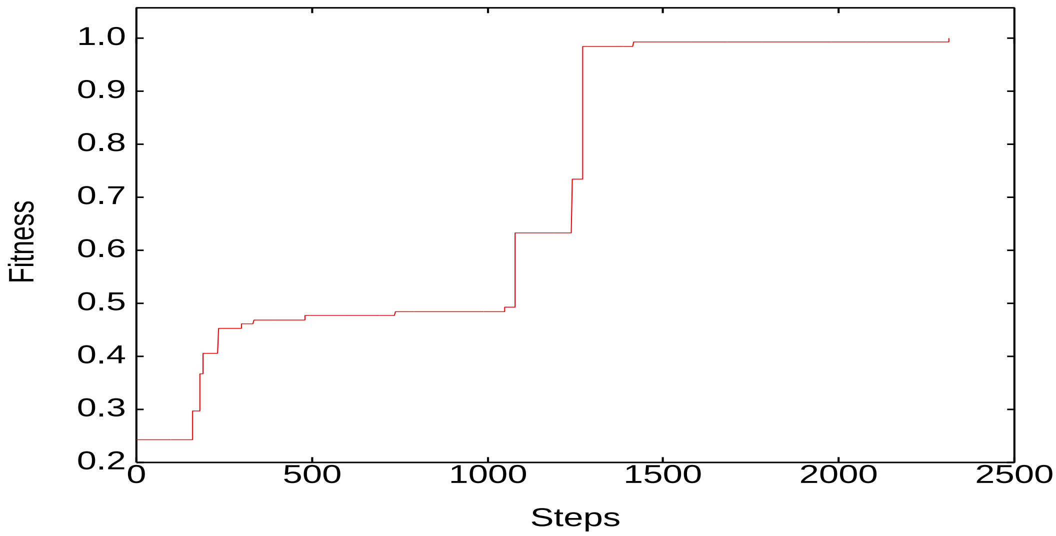
(a) A needle in haystack



(b) A mesa in desert



□ Hill climbing from very edge of sidewall.



□ Hinton & Nolan's Search for A-needle-in-haystack

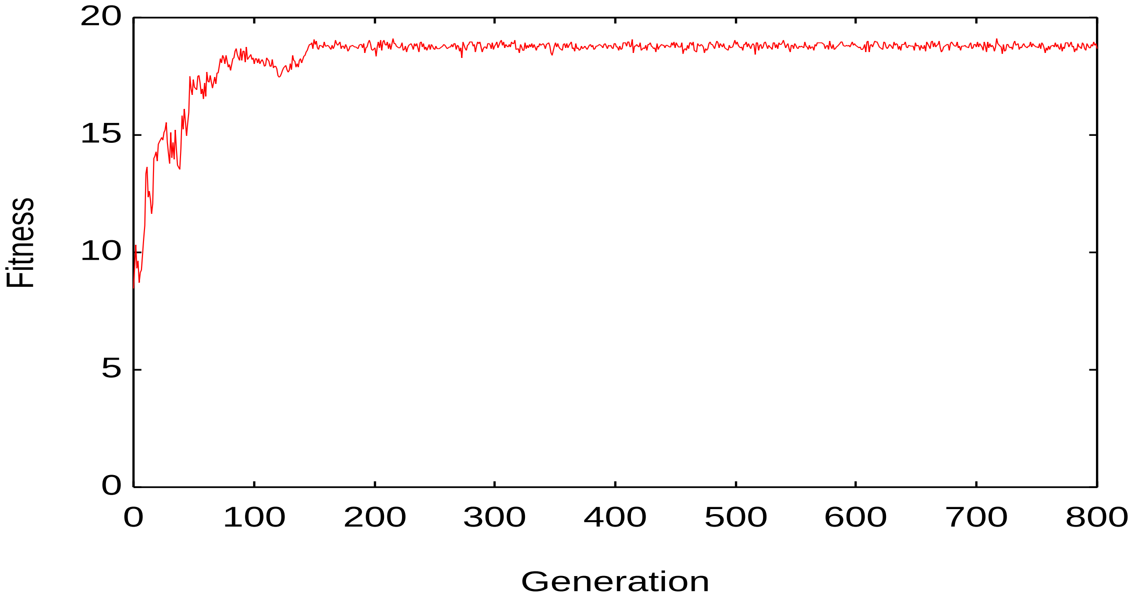
- A-needle ⇒ Only one configuration of 20 bits of binary string.
- Haystack ⇒ 2^{20} search points.
  - · say, (11111111110000000000) is assigned fitness one, while others zero.

(cont'd)

- *lifetime learning* of each individual.
  - about 25% are "1", 25% are "0", and the rest of the 50% are "?" .
  - During an evolution all the "?" position is assigned one or zero at random $\Rightarrow$ learning
  - Repeat the learning upto 1000 times
  - If it reaches the point of fitness one at the n-th trial, then the *degree to which learning succeeded* is calculated as

$$1 + 19 \cdot (1000 - n)/1000.$$

□ Search for a needle



□ Conclusion

- **Search for weights to store patterns is more difficult than a search for**
  - **· a needle in haystack.**

- **We guess it's due to very steep side wall of the peak.**
  - **· A mesa rather than a peak?**

(cont'd)

- **Can climbers take the uphill slope again?**
  - · **Yes, but only from the edge.**

- **Hinton & Nowran's Experiment?**
  - · **Effect of learning during life time (Baldwin Effect).**