Getting emotional with evolutionary simulations: the origin of affective processing in artificial neural networks
Heerebout, B.T.

Citation for published version (APA):
Appendices

Appendix A
To create offspring the first parent was chosen by ‘tournament selection’. This was the agent with the highest fitness within a random subset of agents of one third of the population. To select the second parent agent, an adapted ‘roulette-wheel selection’ was applied. For each agent a portion of a roulette wheel (i.e., proportional to the probability of selection) was allocated. The probability of an agent (Pi) to be selected was calculated with the formula:

\[
P_i = 0.5 \left( \frac{\text{ReversedDistance}_i}{\text{SumReversedDistance}} + \frac{\text{ReversedReducedFitness}}{\text{SumReversedReducedFitness}} \right)
\]

(1)

This formula combined an agent’s the resemblance to the first parent and the agent’s reduced fitness. The resemblance to the first parent, its ReversedDistance, was measured as the inverse of the Euclidean distance between the networks, which in this context were represented as points in a multidimensional space, defined by the networks’ connections. The agent’s ReducedFitness was its fitness minus 90% of the lowest occurring fitness in the current generation.

The connection strengths of the parents were used to create two offspring. Half of the time each offspring was a copy of one of the parents and the other half they were a mixture of both parents. In case of a mixture, two random parts of the weights would be swapped. In addition, all the weights, ranging between –10 and 10, would mutate slightly by adding M, which was calculated with the formula:

\[
M = -0.1 \cdot \log \left( \frac{1}{r} - 1 \right)
\]

(2)

where r was a random value drawn from a uniform distribution between 0 and 1. The offspring were added to the population and the process was then repeated until the population size had tripled. All agents were considered as parents anew with the creation of each offspring pair.

Appendix B
The entities had a round shape with a radius of 10 length units. The environment was a torus shaped square which sides measured 400 length units. Smells were emitted by all
entities in the environment and dissipated in all directions equally with geometrically
decaying intensity ($S$) according to the formula:

\[
\begin{align*}
\delta < \delta_{\text{MAX}} : & \quad S = \frac{1}{1 + \delta} \cdot S_{\text{MAX}} \cdot \left(1 - \frac{\delta}{\delta_{\text{MAX}}}\right) \\
\delta \geq \delta_{\text{MAX}} : & \quad S = 0 
\end{align*}
\]

with $S_{\text{MAX}}$ = maximal smell intensity at the source (set to 25), $\delta$ = distance to the
source (in length units) and $\delta_{\text{MAX}}$ = maximal distance at which the source was smellded
(set to 100). The entities emitted multiple smells in different ratios. Plants emitted both smell ‘a’ and ‘b’ with the intensity of ‘a’ being half as strong as ‘b’. With predators smell ‘b’ was half as strong as smell ‘a’.

The agent’s motor actuators exerted a force against the environments surface
that propelled the agent straight ahead if both forces were equal or curved if the one was
greater than the other. Each time step that force was exerted, the agent’s energy was
reduced by an amount $E_{\text{nergy}}$ according to the formula:

\[
E_{\text{nergy}} = \frac{\text{MaxEnergy}}{1 - (\text{MotorAct}_{t} - \text{MaxAct})} \cdot \frac{\text{MotorAct}_{t}}{\text{MaxAct}}
\]

where $\text{MotorAct}_{t}$ = activation of the actuator at time $t$, $\text{MaxAct}$ = the actuator’s
maximum activation (set to 1.0), $\text{MaxEnergy}$ = maximum consumption per actuator per
time step (set to 0.001). In addition, the agent used a static energy cost of 0.001 per time
step.

**Appendix C**

In the third stage agents are removed from the population. The probability of staying in
the population for another generation would be determined in three steps. First a relative
factor (RF) was calculated for each agent with the formula:

\[
RF = b \cdot \frac{(F_{i} - F_{\text{min}})}{F_{\text{max}} - F_{\text{min}}} + o
\]

with $F_{i}$ as the fitness of agent $i$, $F_{\text{min}}$ as the lowest fitness in the current population,
$F_{\text{max}}$ as the highest fitness in the current population, $o$ as the lower limit of RF (set to
0.15) and $b$ as the range of RF values (set $1 - o = 0.85$). In the second step the relative
factor was used to calculate the population factor which accounted for the influence of
the current population size ($N$) with respect to the initial population size ($N_{0}$). If the
current population was large, the agents’ probability to be removed increased and vice versa. The population factor (PF) was calculated according to the formula:

\[ N \geq N_0 : \quad PF = RF \frac{N_0}{N} \]

\[ N < N_0 : \quad PF = 1 - \left[ (1 - RF) \cdot \left( 1 - \left( 1 - \frac{N}{N_0} \right)^2 \right) \right] \quad (6) \]

In the third step the probability that the agent would remain in the population for another generation was calculated by multiplying the PF by 0.99. This ensured that even the fittest individual could sometimes be removed from the population.

**Appendix D**

The weights of the network’s connection determine an agent’s behavior. The network is symmetrical in the sense that all nodes have a mirror node and thus, for each connection running from node i to j there is a mirror connection from mirror node i’ to j’. Activations are of course unique, otherwise the output nodes’ activation would always be equal and the agent could only move forward. The symmetry greatly reduces the number of mutations required to evolve a functional behavior and is correct under the assumption that the agent should give an exact mirrored response in an exactly mirrored environment. The activation \( y_i \) of all nodes was calculated with the formula:

\[ y_i = \sigma \left( \sum_{j=1}^{N} \omega_{ij} y_j + \theta_i + S_i \right) \quad (7) \]

The formula of the sigmoid was:

\[ \xi \geq 0 : \quad \sigma(\xi) = \frac{\xi}{1 + \xi} \]

\[ \xi < 0 : \quad \sigma(\xi) = 0 \quad (8) \]

with \( \omega_{ij} \) = the weight of the connection from unit j to node i, \( N \) = the number of connections to node i, \( \theta_i \) = the bias of node i (0.1 in the simulations) and \( S_i \) = the smell input to node i from the corresponding sensor (only for input layer). In order to create a temporal difference between processing over long neural pathways and short neural pathways, a time delay was introduced by storing a node’s activation for one time step. Hence, the activation that was calculated for a node at time t was propagated over it’s sending connections at time t+1. As a consequence, input signals were processed to the output nodes at t+1 via the direct projections and at t+2 via the indirect connections, through the hidden layer.