Chapter 6

The Adaptive Value of the Dual Route Architecture

A proper understanding of the large diversity of processes and architectures in the human brain, probably, requires knowledge about its evolutionary history and the environmental demands placed on it in the past. Particularly the field of emotions has witnessed the emergence of a neurobiological model motivated by evolutionary reasoning (LeDoux, 1996; for related ideas see Oatley & Johnson-Laird, 1987, Tooby & Cosmides, 1990). LeDoux (1996) suggested that parallel direct and indirect pathways have evolved in the brain for fear processing and that the functional differences in terms of speed and accuracy between the two are adaptive. An innovative aspect of this approach is that the evolutionary reasoning focuses on internal cognitive (and affective) processes instead of on externally observable behavior, as was the case in the older ethological approaches (Darwin, 1965). Evolutionary psychology is often criticized for its lack of empirical constraints, but we attempted to investigate LeDoux’ evolutionary justification experimentally by applying Beer’s (1990) method of computational neuroethology. This comprises the computational simulation of neural processes by artificial neural networks (e.g., Armony, Servan-Schreiber, Cohen, & LeDoux, 1995), of evolutionary processes by genetic algorithms (Holland, 1975), and of artificial environments giving rise to selection pressures (Beer, 1990).

Evolutionary psychology has been attacked for providing only "just so" stories that can be tailored to fit any phenomenon at hand (e.g., Gould & Lewontin, 1979; Looren de Jong & van der Steen, 1998). In a discussion of this criticism Ketelaar and Ellis (2000) acknowledge the problems which stem from the absence of an appropriate research tool other than conceptual speculation. They are more optimistic on the present state of evolutionary psychology, however, and argue that speculation has become more rigorous resulting often in formal hypotheses that can be tested empirically.

Evolutionary theorizing can be constrained with the help of a computational optimization procedure gleaned from evolutionary biology, called genetic algorithms (GAs). They were introduced to artificial intelligence research by John Holland (1975) and were defined by Goldberg (1989) as "... search algorithms based on the

4 This chapter is a modified version of Den Dulk, Heerebout and Phaf (In Press).
mechanics of natural selection and natural genetics." (p.1). Biological systems contain their genetic information in minute structures called chromosomes. Likewise, GAs encode the solutions to a problem in a string, which involves a coded parameter set (often with binary features) that is subject, just as in nature, to mutations and crossovers. Mutation may, for instance, randomly flip a bit from 0 to 1 or from 1 to 0, and, generally, has a low probability of occurring. Mutation can improve performance by occasionally suggesting a new partial solution. With crossover, two strings selected for reproduction are combined to produce a new solution. Often a crossover point at which the two parent strings are cut is chosen randomly. After parts of both strings have been interchanged they are 'glued' together to produce new strings.

Starting from an initial population that may consist of random strings, the search for optimal solutions proceeds through reproduction. An organism has a certain chance to live and reproduce, which is usually referred to as its fitness. Fit organisms are more likely to live and reproduce than less fit organisms, or from a slightly different perspective: fit genes live, others die. From generation to generation, this leads to a higher overall fitness of the population and to better solutions. In practice, strings are pseudo-randomly selected, with high-fitness strings having a better chance of survival and reproduction. According to its fitness, a new solution is inserted at the appropriate position (rank) and 'pushes the worst solution off the table'. Because the best solutions are retained, the highest-ranking fitness value increases monotonically. The choice of a suitable fitness criterion eventually determines which problem is solved by the GA.

Previously, GAs mostly served as a powerful optimization tool in artificial intelligence (Goldberg, 1989), for instance in design and scheduling problems with many boundary conditions. They were also used to optimize neural network models to perform particular tasks (i.e., written number recognition; Happel & Murre, 1994), but not linked directly to specific neurobiological models. The increasing availability of computational power, however, allows for the simulation of more complex environmental conditions and, thus, enables the relatively novel application of GAs to evolutionary psychology. GAs have already been applied to a limited number of psychological issues. Brinkers and den Dulk (1999; see also den Dulk & Brinkers, 2000) have, for instance, made a computational study of the emergence of altruism in viscous populations.

Here we will apply such a method to LeDoux' (1996) hypothesis that the architecture of a direct and indirect pathway, as found in neurobiological research, is adaptive. The indirect pathway goes from the thalamus via the cortex to the amygdala, and the direct path from the thalamus to the amygdala. This direct path is faster, has less capacity, and is more coarsely grained than the indirect pathway. According to LeDoux' interpretation the fast path is primarily important in life-threatening situations, when a fast response to a potentially dangerous stimulus can be life saving, whereas an inappropriate fear response does not seem too harmful. When the speed of the reaction is not critical, however, a more accurate evaluation through the indirect path, in which also the false alarms are corrected, is more productive. Due to the bias towards fast fear-reactions, indirect processing will tend to restore balance and mostly inhibit these reactions. In the example LeDoux gives someone sees a curved object lying on the forest floor. The direct route could identify this as a snake and initiate a fear response. If, however, the object is in fact a stick the slower but more accurate processing through the indirect route can feed additional information to the amygdala and inhibit the fear response.
This dual pathway model may be used to explain a much broader range of affective phenomena than only fear conditioning for which it was initially developed. Behavioral research on human subjects has suggested that a similar mechanism might be responsible for the results found in affective priming studies (e.g., Murphy & Zajonc, 1993). In these experiments the preference judgement of a neutral, meaningless, stimulus was influenced by a prime only when it was presented very briefly (4ms) and not at longer presentations (1000ms). An explanation for this is that there is a direct affective influence as the consequence of fast processing done by a simple quick system, whereas at longer presentations there is some kind of inhibition of this primary response coming from more developed systems. Although, it is not at all clear whether the same anatomical structures are involved in these phenomena, it is in itself interesting that a similar mechanism seems to underlie it. The kind of architecture suggested by LeDoux, might reflect a more general principle in affective processing (also suggested by Buck, 2000).

This main question of this chapter is whether the architecture suggested by LeDoux can be adaptive, and under what specific environmental circumstances it will have its adaptive value. To investigate this we will combine genetic algorithms (as presented in Chapter 2) with neural networks (as presented in Chapter 3, 4 and 5). The genetic algorithm will be used to optimize the weights of the neural network. To test the adaptive value we will simulate an evolutionary process of an organism possessing a neural network controlling its behavior. In the genetic algorithm procedure the organism will be exposed to specific selection pressures, which are assumed to have led to the evolution of the dual route architecture (i.e., dangerous predators and a limited amount of food). This simulation setup was inspired by work done by Randall Beer (1990), who used a simple simulated organism, similar to the Braithenberg vehicles (Braithenberg, 1984). In Beer's simulations an artificial organism (which we will call an agent) was optimized to navigate towards a food patch. Here the organisms evolved in a similar setup but predators as well as food were present in the environment.

Starting points

LeDoux' (1996) reasoning is best summarized in the following quote: “Although the cortical pathway provides the amygdala with a more accurate representation than the direct pathway to the amygdala from the thalamus, it takes longer for the information to reach the amygdala by way of the cortex. In situations of danger, it is very useful to be able to respond quickly. The time saved by the amygdala in acting on the thalamic information, rather than waiting for the cortical input, may be the difference between life and death. It is better to have treated a stick as a snake than not to have responded to a possible snake.” (LeDoux, 1996, p. 166).

In this quote LeDoux mentions a number of aspects thought to be relevant for the adaptive value of the dual route architecture. The direct route processes information faster than the indirect route. The indirect route has as advantage that it has a more accurate representation. This seems a simple concept, but there are few artificial neural networks suited for this, because they do not incorporate time explicitly in the neural processing. The stimuli processed in the direct route also take absolute priority over all other processing. They are of vital importance and may decide between ‘life and death’. Not only do the stimuli processed through the direct route have big consequences, they also are time critical, thus need fast reaction, whereas other factors, thought important, may not need such very quick reactions. The most significant issue here is the tension between speed and accuracy, which is
resolved by the dual routes. The next sections discusses how this tension is reflected in the simulation.

**Selection pressure for speed**

In LeDoux' example a predator (a snake), is used as the source of danger. Although he could have used another source of danger, the predator is a particularly suitable one, because the predatory defense system for active escape and avoidance was probably among the first functional systems to evolve (Archer, 1979). The snake might be a particularly apt example, because according to Ohman (1985) 'the anti-reptile defense system of the early mammals provided the prototype around which later predatory defense systems were built' (p. 128). It is also important that the threat of a snake needs a quick reaction. The time criticality needs to be significant enough to make evolution prefer the shorter less accurate pathways in an encounter with a predator.

In many types of simulated neural networks, for example back propagation of error (Rumelhart et al., 1986), information propagation is instantaneous. When input activation is fed to such a network it is propagated to the output nodes immediately in a single sweep. In such networks there is, thus, no advantage of a direct route over an indirect route, because activation over both routes will reach the output layer at the same time, i.e. instantaneous. It is, however, also clear that this type of instantaneous processing is highly unrealistic, and that there are always time costs involved in real neural processing. The simulation presented here will take into account the time it takes to propagate activation over neurons and axons.

**Selection pressure for accuracy**

In order to obtain selection for accuracy there needs to be some demand upon the perceptual discrimination system. In a task in which the agent is only confronted with predators any stimulus will indicate the presence of a predator, thus no discrimination needs to be made. Therefore, the simulation should also include some other objects, which need a different kind of response. In this simulation food was chosen as the competing stimulus, because the contrast between the approach reaction required for food and the avoidance reaction required for predators, makes it easier to interpret the behavior observed in the simulation.

In addition, discrimination between predator and food should not be too easy. A very simple discrimination problem can be solved in a single layer of connections, and an indirect route for more elaborate processing would not be necessary at all. The problem should thus be hard enough to make the organism gain something from more extensive processing through the indirect route. The postulated trade-off between speed and accuracy will not show up in a too simple problem, because both optimal speed and perfect accuracy can be achieved in just a single route. Therefore, task complexity has to be at a sufficiently high level to demand elaborate processing. In some of the simulations this was obtained through overlap in the input pattern of the two types of stimuli.

We will now discuss the implementation of the simulation environment in detail, and after that the five simulations that were performed in this artificial environment. Analyses of the results showed that only when there was i) delay of processing in the neural network, ii) a sufficient discrimination problem and iii) a big impact of predator avoidance for the chance to reproduce, the dual-route architecture evolved.
The General Simulation Setup

The general methodology of this project was to simulate an artificial evolution and investigate the neural networks of the organisms which evolved in it. Within this evolutionary process new agents were born, tested on fitness, and those with the highest fitness were selected for reproduction. The fitness of the agents was determined by putting them into a virtual environment in which they had to avoid predators and collect food.

The environment in which the agents had to survive was 400 by 400 length units. It was torus shaped, i.e. when an agent went over the edge of the map on one end it came back on the other end. This was the case for the left and the right as well as the top and the bottom sides.

The rules of the simulated world were based on real world physics laws. We simulated dynamical processes as a function of time and two spatial dimensions. Such continuous rules of physics makes the simulation more complicated compared to the more common grid world simulations, which use distinct cells and discrete time steps (as in Chapter 2). For our simulations, however, the time factor is essential, because the reaction times are the subject of investigation. An additional advantage of real world physics rules is that choosing the rules is less arbitrary, not just any rule can be chosen, but only those that resemble real world physics. Of course the continuous physics laws had to be approximated in the computer simulation with discrete time steps (iterations) and floating point variables which have limited precision.

Physics laws were applied for the following aspects: Elasticity in collisions (as described by Francis, 1973); calculation of position, speed and acceleration (with respect to both translation and rotation); and friction (for translation and rotation). Friction was considered to be linearly increasing with speed.

The agent

The agent had a round body equipped with two pairs of smell sensors which were placed at some distance from each other at the front of the agent (see Figure 34). The agent also possessed two motor actuators on each side of the body, which exert a force straight backwards on the environment, resulting in a forward force on the agent’s body. When one of the actuators is more highly activated than the other, the agent will turn to the side of the less active actuator. By selectively activating one or the other actuator the agent can navigate through the environment. Sensors and actuators are connected by a simple neural network, which we shall specify later. The agent has a body with a radius of 10 length units. The actuators and sensors are located at the edge of the agent’s body. The actuators are at 90 and -90 degrees from the front. The sensors are at 45 and -45 degrees from the front. The distance between the sensors was thus $10\sqrt{2} (= 14.1421)$ length units.
Chemotaxis

The principle by which the simulated agents navigate in the artificial world resembles chemotaxis, which works as follows. The predators and food in the environment emit a chemical that can be detected by the agent’s smell sensors. The perceived strength of the smell at a certain point depends on the strength of the source and the distance from the edge (not the center) of the source (see Figure 35) to the sensor. Since the smell sensors are positioned at some distance to each other on the agent’s body there is a difference in stimulation strength between the two sensors. This difference can be used to navigate towards or away from the smell emitting object in the environment (See Appendix A for the exact function and parameters used for the smell gradient). A smell consisted of two components (A and B). Different smells differed in the intensity of these components.

Figure 34. The agent’s body, with two sensor pairs at the left and right front of the agent. There is one motor actuator at each side. A three layer neural network connects the sensors and the actuators. The network depicted is a somewhat simplified form, it shows two instead of four nodes in both hidden and input layer.

Figure 35. The decrease of smell with distance with a source of strength 25 (arbitrary units).
**Neural network**

The neural network connecting sensors and actuators consisted of three layers of nodes, which had feed-forward connections from input to hidden layer, from hidden to output layer, and also direct connections from input to output nodes. The input layer consisted of four nodes receiving direct input from the sensors (see Appendix B), the hidden layer of four nodes, and the output layer of two nodes that activated the actuators. Though the connections started out at zero, the unidirectional connections from all nodes in one layer to all nodes in another layer could achieve a non-zero weight (from -10 to 10). Because the simulated evolution fully determined the connection weights, and, thus, which connections would eventually be used, the availability of all potential connections allowed for the development of dual processing in the network. The complete network consisted of 32 connections, 16 (4x4) connections between input and hidden layer, 8 (4x2) between hidden and output, and 8 (4x2) between input and output.

Activation transfer between nodes was governed by the standard weighted activation rule (Rosenblatt, 1962) which was combined with a sigmoid function to squash the activation between 0 and 1 (see Appendix B).

![Symmetry Line Diagram](image)

**Figure 36.** Connections to the left and right side of the symmetry line were implemented to have identical values. This means that for all connections a mirror connection was determined, which has the same connection strength. The figure illustrates how the mirror images of connections were determined. The connection from node i to j is the mirror image of connection from i' to j', both have strength W1. The connection from i' to j is the mirror image of the connection from i to j', both have strength W2.

The connection strengths in the neural network obeyed a left-right symmetry (see Figure 36). This means that each connection in the network had a mirror image, which had the same connection strength. To assign the mirror weight pairs we had to assign mirror node pairs first. For the input and output layers the two nodes simply formed a pair, for the hidden layer this was just an arbitrary choice. Two nodes were treated as being on the left side, the other two were treated as being on the right side.
of the symmetry line. Because of this symmetry the number of connection strengths that had to be specified was, thus, half of the number of connections. This was done to reduce the search space for the genetic algorithm, and also because it tends to produce good solutions for this specific task, which demands symmetric behavior.

By adding connections directly from input to output as well as through the hidden layer the direct and indirect route are, in a way, already in place before the simulation starts (although the connections were initialized as zero). It would in principle also be possible to simulate the evolutionary development of the architecture as well. Such an approach would, however, be much harder to analyze, and with the current method it is still possible to establish whether the two routes are adaptive by analyzing the way the connection strengths develop. Depending on the condition in which the agent is put, we expect preference for either food or predators in the direct route. If LeDoux' hypothesis is true and the evolutionary conditions are met in the simulation, we should expect avoidance behavior to be primarily encoded in the direct route, and approach towards food primarily encoded in the indirect route. This preference should not be found in conditions where there is no advantage for a dual route architecture.

![Figure 37. An overview of the artificial world. In this case a single agent is in a world with ten food patches, and one predator.](image)

**Food and Predators**

Food was simulated by a round motionless object of radius 10, which emitted a smell consisting only of component B. When an agent made contact with a food patch it was considered eaten, and the food patch disappeared. There were always ten food patches in the world. At the moment a food patch was eaten by the agent a new one appeared at another position.
There were always six predators in the world. The predators were identical to the agent except for two aspects. First, they emitted a smell, whereas the agent did not. The smell consisted only of component A. Second, the connections of the predators were fixed at values that always made them chase the agent. The predators were not modified in the evolutionary process. When a predator made contact with an agent, it was assumed to be caught by the predator, and the agent died. Figure 37 is a snapshot of the simulation world with all elements in it.

**Energy**

The agent could gather energy by consuming food patches. When a food patch was eaten the energy contained in the food patch, which was always 1.0, was added to the agent's energy.

When the motor actuators of the agent were activated it burned energy. The amount of energy burned was particularly high when the activation was near its maximum activation of 1.0 (See Appendix C for the function and parameters used).

**The genetic algorithm**

To optimize the connection strengths in the neural network a genetic algorithm was used. This is a cyclic process based on Darwin's idea of natural selection. First a generation of individuals is tested on performance in the environment. Then individuals with the highest fitness are selected for reproduction, and the generation containing the new individuals is tested again.

The procedure of the genetic algorithm is depicted schematically in Figure 38. After a generation was tested, the weak (i.e., least fit) individuals were removed. The remaining individuals were transferred to the new generation, which was supplemented by new children born from the survivors. Reproduction took place until the population had reached three times the size it had after the removal of unfit individuals. After reproduction, the cycle started again with a new fitness test for all individuals (i.e., the children and the survivors from the previous cycle). The longer an agent survived the more accurate became its fitness estimate, because its fitness was averaged over all tests.

![Figure 38. Cycle gone through in each generation. Creating new individuals as children of the current population. Testing fitness of all individuals, removing the weak individuals. Then again new children are generated based on the remaining individuals.](image)
**The fitness measure**

To measure the fitness of an agent, it was placed in the environment, with ten food patches and one predator. The agent could move around in this environment for a number of time steps. The test was ended when the agent was caught by a predator, or the maximum number of iterations of 5000 was reached. In practice this maximum number of iterations was reached only very rarely. The amount of energy the agent had collected during its life was taken as the fitness measure. Total energy collected seemed a good measure because it reflects how well an organism copes in the environment and also leads to selection for economic energy consumption (i.e., an agent should only move when it leads to an energy increase). Predator avoidance also had an effect on the total amount of energy collected, although indirectly. The longer the agent lived the more food patches it could collect.

The initial positions of the agent, predator and food were chosen randomly. This initial random placement could, of course influence the fitness measure. For instance, an agent would live very short if it was placed right in front of a predator. To be able to compare different agents, the same random positioning was used in a single test of all agents within a generation. In a generation the test was repeated twelve times with different initializations, so that an average fitness for that individual in that generation could be calculated. One agent was, thus, never tested twice with the same initialization.

**Creating new generations**

The first generation consisted of 18 agents, with all connections set to zero. The size of a new generation was not fixed, but was regulated by a set of rules (for details see Appendix D). These rules determined for every individual whether it is transferred to the next generation or not. A new generation also consisted of new agents that were mutations of old agents. The effect was that when there is still much variation in fitness, the population becomes small, and when there is little variation it becomes large. The idea behind this mechanism is that a large variation in fitness indicates that there is much room for improvement, which makes the task easy and can be done by a small population. When there is little variation in the population the algorithm is close to convergence to a local optimum in the search space, which means it is better to have a large population to perfect the current solution, and to search for new lines of improvement.

**Creating children**

New children are created until the population has become three times the size it was after removing weak individuals. For each child two parents are selected. For the first parent we use tournament selection. One half of the agents in the population is randomly selected. From this subgroup the agent with the highest fitness is selected as parent1.

The second parent is selected from the whole population with the probability of selection depending on the ‘distance’ to parent1. The 16 connection strengths on which agents can vary can be seen as a 16 dimensional Euclidian space, in which each individual is positioned at a certain point. The distance between two agents can thus be seen as the distance between these two points in that space. The mating procedure determines that the further they are apart, the smaller the chance they will be paired for mating. These chances are used in a Roulettewheel selection (see Appendix D).
The essence is that the chance of selection as parent2 is inversely proportional to the distance to parent1.

Not in all cases both parents are used to create the child, this depends on the amount of cross-over. The cross-over parameter determines the chance that there is cross-over for each pair of parents. If there is cross-over, a child will be constructed from a combination of the connection strengths of both parents. This is done by picking two random points in the array of connections, the part inside those two points is inherited from parent1 the rest is inherited from parent2. Without cross-over the child will be constructed from the connections of parent1 alone (see Appendix D).

After the connections of the parent(s) have been copied to the child there is also a mutation of the connection strengths, determined by the mutation parameter. The mutation obeys a poisson distribution. The mutation parameter determines the width of this distribution (see Appendix D, for more details see Heerebout, 2001)

**Simulation 1: Optimization of the Agent**

This first simulation functions as a control simulation to which other simulations could be compared. This simulation will lack some of the conditions that are assumed to make dual route processing adaptive. Activation of a node, for instance, not yet entailed a time delay (see Appendix B) Thus, we do not expect that a network configuration corresponding to LeDoux’ hypothesis will develop. There will, however, be an extensive analysis of the solution to which the network converges, in order to be able to compare it to the simulations that do have the necessary conditions. These simulations also provide an opportunity to become acquainted with the methods of analysis that are specific to this application of genetic algorithms to neural networks. A number of different methods of analyses were applied to obtain a more general insight of the way the individuals preformed their task.

**Simulation 1: Method**

A simulation was run with the procedure described above. All simulations were run over 300 generations. All the parameter values are shown in Table 1.

**Simulation 1: Results**

**Fitness:** The simulation shows a gradual increase in average fitness over the 300 consecutive generations tested (Figure 39). The variance from test to test could be quite high because of the different initializations of the world. The amount of variation is increased, however, by the method of presenting all agents with the same initialization. As explained in the simulations setup section, this was done to reduce variance in the tests between agents. This is an advantage if one wants to select the best agents from a generation, but this is a disadvantage if one wants to determine whether there is an increases in fitness from one generation to the next. Another factor increasing variation from generation to generation is the variable population size. Small populations have a large variability from test to test compared to large populations. To get a better impression of the progress from generation to generation the fitness values are presented as averages over ten consecutive generations.

**Network analysis:** The most basic approach to analyze the selected individual may be to make a schematic representation of the connection configuration of the network (see Figure 40). The nodes in the network are depicted as circles and the connections are depicted as lines between nodes. Connections with an arrow at the
end indicate excitatory connections. Connections with a circle at the end indicate inhibitory connections. Both the excitatory and inhibitory connections were categorized in three groups according to their strength (see index of Figure 40). The thickness of the line indicates the connection strength.

Figure 39. Fitness development over 300 generations. The vertical axis represents the fitness. On the horizontal axis are the generations in clusters of ten.

Figure 40. A schematic view of a representative individual of the last generation in Simulation 1. ‘O’ stands for output, ‘I’ for input, ‘H’ for hidden. In the subscripts the first character stands for food (F) or predator (P), the second character stands for left (L) or right (R). For the hidden units a number denotes their identity. Nodes H2 and H4 are mirror images of nodes H1 and H3 respectively.
In the direct path the connections from food input (IFL and IFR) to the output layer (OL and OR) have a very characteristic pattern. The excitatory connections go from left input to right output, and right input to left output. This is a pattern we call crossed-excitation. The behavior which is a consequence of such an organization is that if there is a source to the right, the right sensor will be activated stronger than the left sensor (because of the smell gradient). Due to the crossed pattern, the left actuator becomes more activated, and the agent makes a turn towards the source on the right. This approach behavior makes sense because the inputs respond to food input. In this particular case the pattern is even more characteristic because there are also inhibitory connections going from left input to left output, and from right input to right output. We will call this pattern parallel-inhibition, which slows down the forward movement and strengthens the turn of the agent towards food. Both crossed-excitation and parallel-inhibition are seen in approach behavior.

From predator input (IPL and IPR) to output layer (OL and OR) there is a pattern we call parallel-excitation: excitatory connections from left input to left output, and right input to right output. This connection scheme will make the agent turn away from a predator smell source. This configuration will cause quite a strong forward movement and less of a turning movement, because there are no crossed-inhibitory connections, and the excitatory connections themselves are quite strong. This solution seems to make sense, because speed is quite important to escape from a moving predator, whereas a food source needs a sharp turn.

The effect of the connections over the hidden units is harder to interpret. Although the location of the hidden units is arbitrary, nodes H1 and H3 will be treated as located on the left side in the agent, and H2 and H4 on the right side. By doing so the network can still be explained in terms of crossed and parallel connection schemes, making talking and thinking about the network much easier.

With respect to the indirect route, hidden units H1 and its mirror node H2 had no connections to the output layer and, thus, had no effect on the functioning of the network. The connections towards H1 and H2 are grayed to indicate they had no effect on behavior. Nodes H3 and its mirror H4 had a crossed-excitation pattern to the output nodes. The effect of the connection patterns onto H3 and H4 are, therefore, the inverse of that same pattern towards the output layer directly. From the food inputs (IFL and IFR) to H3 and H4 there was the characteristic pattern of parallel-inhibition and crossed-excitation, which would have caused avoidance behavior if it was connected directly to the output nodes. But, as we have seen, it is inverted into approach behavior because the hidden units have a crossed-excitation pattern to the output nodes. As far as the food is concerned the direct and indirect route, therefore, do not show qualitatively different functions.

From the predator inputs (IPL and IPR) to H3 and H4 there is parallel-excitation and crossed-inhibition. This would cause approach behavior if directly connected to output, but again this is inverted by the crossed-excitation pattern from hidden to output. Again direct and indirect routes cause similar behavior, also with regard to the predators.

In conclusion the network analysis shows a quite consistent pattern. The function of the direct and indirect route seems to be quite similar. For both the direct route and the indirect route there is approach behavior for predators, and avoidance behavior for food.

In this and the following simulations one representative individual from the final generation was selected for further analysis. Averaging over the individuals did
not seem appropriate because in principle different types of solutions can be found within one generation. In practice most agents within the final generation were virtually identical, thus this choice would not have influenced any of the conclusions and only in rare cases slightly deviating individuals were discarded.

**Figure 41.** The mapping charts for food input in Simulation 1. Above row: Output activation plotted for various food inputs presented to the intact network. Middle row: the same for the direct route only. Bottom row: the same for the indirect route only.

**Mapping charts:** By inspecting the connection weights we cannot always understand their effects on actual behavior. The next analysis will be a step closer to behavior. In this analysis the agent was presented with specific input patterns and we recorded the corresponding output activation. Input activations for food and predators were always presented separately. For each type of input we presented the network with combinations of left and right input activations. We systematically varied the input intensities between 0.0 and 1.0 in steps of 0.1. Resulting in 121 input combinations (11 different input activations for left the right input). By representing these data in charts, such as in Figure 41 and Figure 42, a better understanding can be
The adaptive value of the dual route architecture

The adaptive value of the dual route architecture gained from the network. We will call these charts mapping charts. On the x and z-axis are the various input activations, on the y-axis the corresponding output activations are plotted. Output activation for the left and right output activation is represented in separate charts. Because of the left-right symmetry of the network the outputs for the left and right output nodes are each other’s mirror images. Representing them both is, thus, in a way, redundant. Nevertheless we choose to show both because this makes it easier to understand the resulting behavior. This method of representing the stimulus-response mapping was adopted from Beer (1990).

Figure 42. The mapping charts for predator input in Simulation 1. Above row: output activation for intact network. Middle row: outputs for direct route only. Bottom row: outputs for indirect route only.

Figure 41 shows the mapping charts for food smell input. The top row shows the mapping of the intact network, the middle row shows the mapping over only the direct route (indirect connection strengths were set to zero), and the bottom row shows the mapping over the indirect route (direct connection strengths were set to
zero). The chart for the left output activation of the intact network is high at the right \((I_L < I_R)\), low at the left \((I_L > I_R)\), and has a steep curve along the middle diagonal where inputs are equal \((I_L = I_R)\). This means that if the right input is higher than the left input, the left output will become active. This is again the familiar approach behavior we expect for food input. The patterns for the direct and intact network look very similar. A difference between the routes is, however, that the direct route does not have such a sharp transition as the indirect route.

The mapping chart for predator input (Figure 42) shows the same kind of pattern, but exactly reversed. All three charts show the same pattern, but again the transition in the direct route is not as sharp as in the indirect route or intact network.

The mapping charts shown in the previous section seem pretty similar for the two routes. One explanation could be that the analogous development is a form of redundancy, which can be useful in the evolutionary process. For instance, when an agent reproduces mutation could cause a defect in one of the pathways. With both pathways performing the same function, the other pathway could take over. A parent with redundant pathways would, thus, have a higher chance that its children will survive. To investigate whether there is such redundancy an extra test was performed. The fitness of the final generation was again tested in hundred fifty replications. Three different conditions were run: with the intact network, with only the direct route (lesion to the indirect route) and with only the indirect route (lesion to the direct route). If the direct and indirect route indeed perform identical functions, and the other one is only there as backup in case of a mutation, we should expect same fitness values for all three conditions.

![Fitness plot](image)

Figure 43. The average fitness plotted during the 150 replications.

The differences between the conditions are quite remarkable (see Figure 43). The intact network reaches a fitness of about 7.5. The direct route condition reached a fitness of 3.9 and the indirect route condition reached a fitness of 1.7. It is interesting to see that, although the mapping charts of the various conditions seem quite similar, their performance does not. Also, it is not one of the two routes which is solely responsible for the intact network's high fitness. Both the direct and indirect routes perform worse than the intact network, and the intact network even performs better
than the sum of the fitnesses of the direct and indirect routes. Apparently the joint operation of both routes is necessary to reach optimal performance.

If we take a closer look at the mapping charts (Figure 41 and Figure 42) in order to find a difference that could explain these results, a contender could be the difference in the transition at the diagonal where inputs are equal \( (I_L = I_R) \). The transition at the diagonal is only a small part of the surface, but an important part. If the agent smells a predator, it will turn its back to the predator, and its left and right inputs will then become equal. This will also be the case if the agent turns its nose straight towards food. So, during a run most of the time the output activation will be determined by the outputs close to the diagonal.

For the predator mapping charts (Figure 42) the slope of the transition is quite steep in the indirect route and in the direct route it is quite flat. A sharp transition of the indirect route allows it to make sharp turns. The amount of the output activation along the transition area in the indirect route is quite low, about 0.2. So, an agent controlled by only the indirect route, will probably turn quickly because of the sharp transition, but because of the low output will not move very fast. If we look at the transition area in the direct route, we see the output is quite high at the diagonal, about 0.8. Because the slope is quite flat it will probably not turn very fast, but once it has turned away its speed will be quite high. The same kind of differences between the routes can be found for food input in (Figure 41), we assume the same principles apply here. Overall, it seems that the indirect route is primarily important for turning and the direct route for speed.

![Figure 44](image-url)

**Figure 44.** Rotation in degrees and translation in distance units, averaged over all time steps during an agent's life. Measured for intact, direct and indirect networks.

**Rotation and translation tests:** To investigate whether this was indeed the case the agent's average translation and rotation per time step (iterations), which was recorded during the fitness tests, was plotted for all three conditions (see Figure 44). It can be seen that, as was expected, the direct route generates higher speed than the indirect route, and the indirect route, turns more than the direct route. Still, in the intact network speed and rotation are both much higher than the summation of speed and rotation in the direct and indirect processing conditions. So, the idea that the indirect route provides rotation and the direct route provides the speed is too
simplistic. Although there is some specialization for speed and rotation both depend primarily on the interaction between the two routes, which ultimately allows for the most adaptive behavior of the agent.

**Simulation 1: Discussion**

We did not find a dual route encoding corresponding to LeDoux' dual processing hypothesis. No such solution was expected, however, because processing was instantaneous, and therefore the time factor played no role. Nevertheless, some qualitative differences were found between the two routes. The direct route coded the fast forward movement, whereas the indirect route tuned the direction more finely. This was the case for both food and predator input. The sharper distinction between left and right input in the indirect route could be interpreted as a more fine grained distinction. This is something which would fit in with LeDoux' way of thinking. However, this distinction is true for both food and predator input, so it does not address the matter of our interest, i.e., a difference in the coding of predators versus food.

**Simulation 2: Introducing Delay of Signal Transmission**

It is a biological fact that it takes time to propagate signals along axons and synapses. In the previous simulation the whole network was updated from input to output in one upsweep in each time step. So, if an agent sensed a stimulus, it could respond to it instantaneously, there was no delay between stimulus and response at all, and, thus, no difference between the routes in the time it took to process the stimulus. In such a situation we should not expect the evolution of the dual route architecture.

In this simulation a mechanism was added to simulate temporal processing. Other neural networks, which have some temporal factor in their activation rule, often have a gradual build up of activation. For example, Beer (1990) used a leaky integrator in the activation rule of the network. This causes the output activation to gradually change to a level that 'corresponds' to the current input activation, but it still allows for instantaneous propagation of activation. For the purpose of this simulation, however, something different is needed. If there is a delay, the output should not be smaller in the initial stages, but it should not be there at all. A real delay is needed, not a gradual build up.

Temporal processing was implemented by delaying the activation in a neuron for a small period before it was transmitted to the next neuron. The signal was assumed to be somewhere along the axons and dendrites during that delay. Because of the architecture there was one such delay in the direct route. This was in the input layer simulating the time it takes to transmit the signal from input to output. For the indirect route there were two such delays. One in the input layer, simulating the delay of transmission from input to hidden, and one in the hidden layer, simulating the delay of transmission from hidden to output. No matter how long this delay lasted, the delay through the indirect route would always be twice as long as the delay through the direct route.

In choosing the number of time steps the signal is delayed we have to consider that the size of the delay has to be in a realistic proportion to the speed of movement in the environment. It should not be too large so that the animal cannot react appropriately at all, and not be too small so that it becomes negligible, and thus will have no effect on the evolution of the network. We explored three possible values for the delay parameter (expressed in number of time steps) in order to determine what would be an appropriate value.
The delay was simulated by storing the activations temporarily in what is called a FIFO queue (First In – First Out). It is a queue in which data is pushed at one end and popped at the other end. This way, activations entering the neuron are transmitted to the next layer only after all other activations in the queue have been popped out. The length of the queue, parameter $L$, is the length of the delay. If this length is 1, a value put into the queue, is popped out immediately. In this case there is thus no temporal delay, exactly as in the previous simulation. If $L=2$ it will keep the new activation value for one iteration before popping it out. If it is 3 it will delay it two iterations, etc. The delay in the neural networks of the predators remained unchanged ($L=1$, no delay) to be able to compare performance between different $L$ values and the previous simulation. Simulations were run with $L$ values of 2, 3 and 4. No simulations of $L = 1$ were run, instead the results of the previous simulation were used as comparison.

![Fitness Development for $L=2$](image1)

![Fitness Development for $L=3$](image2)

![Fitness Development for $L=4$](image3)

Figure 45. The fitness development for $L=2$, 3 and 4 over generations in Simulation 2. On the y-axis the fitness and on the x-axis the generation. To reduce variation we took the average over 10 iterations for each point plotted.
Figure 46. A comparison of four mapping charts in Simulation 2. All charts are made with only the indirect route intact and plotting the left output activation for combinations of food input. The $L=1$ condition is taken from the previous simulation.

**Simulation 2: Results**

**Fitness:** The fitness curves of all three conditions initially rise and then flattens off (see Figure 45). As was to be expected, higher $L$ resulted in lower fitness curves. This simply indicates that a delayed response is a disadvantage in an environment with instantaneous reacting predators. The biggest difference was between $L=1$, of the previous simulation, and $L=2$ (compare Figure 39 and Figure 45). Thus, the single delay at $L=2$ already had a noticeable influence on performance.

**Mapping charts:** Figure 46 shows four mapping charts for each $L$ value. They were all made with only the indirect route intact and plot the left output activation for combinations of food input. The $L=1$ chart is taken from Simulation 1. It can be seen that for $L=1$ the difference in output activation between the left side of the surface ($I_L > I_R$) and the right side of the surface ($I_L < I_R$) is larger than any of the others. Also, the transition between the high and low surfaces on the diagonal ($I_L = I_R$) is quite steep. This means that as soon as one input becomes larger than the other, the output pattern changes dramatically. For food input this means the agent will make a sharp turn towards the food and will keep its nose right on target. For $L=2$ and $L=3$ the difference between the left and right side of the surface is smaller, while for $L=4$ there is no effect of input whatsoever, the surface is completely flat. The complete lack of any contribution of the indirect route in the $L=4$ condition is probably caused by the fact the delay through the indirect route has become too large to contribute anything useful (in the indirect route with $L=4$ the delay is 2 x 4 time steps). In some situations a late response might even be harmful, for instance, a direct route response generated by a nearby stimulus might get interfered by an indirect route response generated by a previous stimulus.
Simulation 2: Discussion

This simulation investigated the influence of the $L$ parameter in order to make an appropriate choice for the delay in the subsequent simulations. The fitness of the agents clearly decreased with higher delays. This was expected because the agent’s later reaction will cause it to get caught more often, or might cause it to slip past a food patch if it cannot turn fast enough. In fact, this lower fitness is a prerequisite for our simulation setup, because the whole idea of the dual route architecture is that it can improve fitness by reacting faster. The biggest difference in fitness is between the $L=2$ condition and the $L=1$ condition. Such a significant decrease in fitness at $L=2$ suggests it can benefit from a dual route architecture.

The comparison of the mapping charts shows that a time delay of $L=4$ is probably too large, because processing by the predators is not delayed to a similar degree. There is no development of the indirect route at all. This is probably because delay in the indirect route (2x4 time steps) becomes so long it cannot contribute anything useful and may even be harmful. For the following simulations $L=2$ is used, although $L=3$ also seems a suitable candidate.

Simulation 3: Overlapping Input Patterns

Simulation 1 did not converge on an organization as in LeDoux’ dual route model. This was, as we mentioned, expected because two of the requirements mentioned by LeDoux were lacking. One such requirement was related to the need for speed. Since there was no delay in that network, the supposedly faster direct route could not add anything extra to the network. The other missing requirement was related to the need for accuracy. In the quote cited in the introduction, LeDoux mentions ‘more accurate representations’ in the cortical pathway. With a simple discrimination task as in Simulation 1, however, both the direct route and indirect route perform perfect discrimination. Thus, the supposedly more accurate indirect route did not have any additional value in this simple discrimination task. In Simulation 2 the temporal functioning of the network was introduced, in the current simulation a more challenging discrimination task will be introduced.

In LeDoux’ (1996) example a person walking in a forest is startled by a stick which is mistaken for a snake. In LeDoux’ theory the direct route is responsible for the startle reaction and the indirect route corrects this initial response. The false alarm, initiated by the direct route, is inhibited by the more accurate indirect route. LeDoux based this ideas on laboratory experiments of conditioning. In these studies the direct route alone appeared to be sufficient for conditioning simple stimuli. When discrimination was made harder, by making it depend on the context, or when similar tones had to be discriminated, the cortical pathways were necessary (LeDoux, 1996).

In this simulation the discrimination task will be made harder by mixing up the smells that predators and food are emitting. In the previous simulations we assumed that the smell of predators and food consisted of only one component. So, the input of the smallest amount of one of these smells would immediately identify the source. In this simulation these two distinct components will both be emitted by both predators and food. They will only differ in the strength of these smell components. In order to discriminate between these two smell sources there needs to be a comparison between these two intensities.
Simulation 3: Method

We chose for a relatively simple method of implementing the overlap in predator and food smell. The smell component, which was emitted by predators, was labeled A and the component emitted by food was labeled B. To predator smell component B was now added and to food smell component A was added. These added components were emitted at half the strength of the original components. The original components remained at full strength.

Because the task was now somewhat harder, it took longer for the simulation to converge on a stable fitness value. Pilot simulations showed the fitness was still rising at 300 generations, therefore, the simulation was now extended to 1000 generations. The $L$ parameter was set at 2, as determined in the previous simulation. In all other respects, the settings were the same as in the previous simulation.

Simulation 3: Results

Fitness: As in the previous sections we plotted the fitness averaged over 10 generations. For the first 300 generations there was a sharp increase in fitness (Figure 47). After generation 300 the increase slowed down. From iteration 800 on it may have reached an asymptotic level of about 4.6.

![Figure 47. Fitness averaged for groups of 10 generations as a function of generations in Simulation 3.](image)

Network analysis: As in Simulation 1 a network analysis was made of the network of a representative individual of the final generation (Figure 48). It should be noted, however, that the connection scheme will be harder to understand, because both smell components are present on both food and predators.

The behavior with respect to component B seems to be primarily determined by the indirect route. The direct connection strengths from B-inputs are so small they are not represented with the chosen criteria, and so are the connection strengths to nodes $H_3$ and $H_4$. Nodes $H_3$ and $H_4$ do have a small contribution because of the bias activation of 0.1 that all nodes have, but this does not depend on the input. The only important nodes are, thus, $H_1$ and $H_2$, which receive crossed-excitation and parallel-inhibition. Nodes $H_1$ and $H_2$ copy their activations to the output nodes by parallel-excitation and crossed-inhibition. The effect is approach behavior towards component B.
The component A input through the direct route is processed by a characteristic pattern of parallel-inhibition and crossed-excitation. This is a very surprising finding, however, since it would result in approach behavior towards component A, which is strongest in predators. Although the indirect route might be able to correct for this, it remains surprising that the direct route, which is thought to be important for the initial fast avoidance reaction, is doing exactly the opposite as expected.

The indirect route for component A has strong parallel-excitation and crossed-inhibition from input to nodes H₁ and H₂. From these nodes to output there is again parallel-excitation. Ignoring interaction effects, this would result in avoidance behavior for component A. The contribution of the direct and indirect route, thus, seem to act as opposing forces with respect to component A. How the agent will behave in the presence of predators, is thus not clear from this analysis. Further, nodes H₁ and H₂ receive parallel-inhibition and crossed-excitation from component B-inputs. This pattern is opposite to what is received from the component A-inputs, which complicates matters because component B is also present, at half strength, in predators.

Figure 48. Schematic view of a representative individual of the last generation in Simulation 3. A difference with the schematic network of the first simulation is that the first character of the subscripts for inputs now represents component A (A) and B (B), instead of food (F) and predator (P). All other labels remain the same.

Mapping charts: How this all works out in practice can be better understood with the aid of the mapping charts for predator input (Figure 50). The chart shows
that the agent still moves away from the predator, but its entire surface is quite high. This means the agent is constantly moving fast when it smells predators, confirming earlier findings that fast forward movement is a good strategy in the presence of predators. The performance of the lesioned network can be understood on the basis of what we have seen in the network analysis. The agent does indeed move towards the predators as a consequence of the direct route, but the indirect route corrects this, and turns the agent away from the predator. The influence of the direct route on the intact mapping chart is that it makes the overall surface higher. It remains remarkable however, that the direct route of the agent steers it towards predators. A differential functioning of the two routes with respect to predators seems to have been obtained, but it is completely opposite to the one postulated by LeDoux.

For food input the Mapping charts of the intact network show a nice curve which indicates approach behavior (Figure 49). The direct route shows a completely flat surface and does not seem to contribute at all. The Mapping chart of the indirect route confirms that most of the food approach behavior is attributable to the indirect route.

Figure 49. The output values for the left and right motor actuator for various left-right input combinations of food in Simulation 3. The top row depicts the intact network, the middle row only the direct route, the bottom row only the indirect route.
Figure 50. The output values of the left and right motor actuator for various left-right input combinations of predators in Simulation 3. The top row depicts the intact network, the middle row only the direct route, the bottom row only the indirect route.
Figure 51. Graphs showing the turns of the agent with respect to the starting position in Simulation 3. The rows correspond with the number of iterations after the start of the input. The agent receives input representing food (left), and a predator (right).

**Rotation analysis:** The results of the previous simulations, where the direct route appeared to be encoding predator approach behavior, called for some further investigation. If our understanding of the mechanism is correct, the encoding of predator approach behavior in the direct route would have as consequence that the agent initially turns towards the predator and then turns away again as soon as the indirect route starts to contribute to the output activation. This is because a signal that
enters the network takes one time step to transmit its activation over a connection. Thus, at the second time step the agent’s behavior is entirely determined by the connections of the direct route. One time step later the indirect route starts to contribute, and now its’ behavior is determined by the intact network. In this analysis we decided to observe more directly in the domain of interest, the behavior of the agent over time.

For this analysis the agent was fixed at a certain position and provided with predator or food input for 5 iterations, and its behavior was recorded per time step. Thirteen different input patterns were presented corresponding to different locations of the smell source. The distance from the agent to the predator or food patch was always 20 units, and the angle, as measured from the agents, was varied between -90 and 90 degrees in steps of 15 degrees. This resulted in 13 different input sets for predators and 13 for food. A negative angle means the predator or food is to the left of the agent, and a positive angle means it is located to its right. The agent’s angle relative to its original angle, was recorded at each of the five iterations. Due to the way the physics was implemented (i.e., there was no direct influence on movement, but a force was applied which gradually influenced movement) there was a delay in the effect of the input. In a natural situation the angle of input would change as a consequence of the agents movement. In the test it was held constant, however, to make the results easier to interpret. The angles depicted in Figure 51 are represented in degrees. Note, that the food and predator charts do not have the same scale.

The left column of Figure 51 shows the reaction to the food input. In the first iteration there is no response at all. The input has not reached the output layer yet. On the second iteration there is the first reaction, but it is small because the agent has to build up speed. During the next three iterations the rotation moves on in this direction. The result is a clear turn towards the food, with larger angles showing the largest turn.

The right column of Figure 51 shows the reaction to predator input. The first reaction seen at the second iteration is a turn towards the predator. On the third iteration the influence of the indirect path starts to push in the opposite direction, but it has not yet undone the speed it had in the original direction. In the fourth iteration the agent has been able to turn away from the predator, with the biggest turn for the input angle of 30 and -30. In the fifth iteration this trend is continued.

The food appears to be more important to the fitness of the agent than the presence of a predator. Because it is not very well able to distinguish between a good patch and a predator it first moves towards any smell and only later, through the indirect route, is capable of discriminating between the two.

**Simulation 3: Discussion**

It was made harder to discriminate between food and predator by mixing the smell components. Whereas in Simulations 1 and 2 detection of a single component was enough to establish the identity of the source, now a comparison between smell intensities was necessary to identify the source. By making the problem harder, the direct route would need to generalize, while the indirect route would still be able to make a more accurate discrimination.

What we found was indeed a generalization in the direct route. Food and predators were treated the same, but it was exactly the opposite as expected. All smell sources were approached, i.e., treated as food, by the direct route. It was, apparently, most adaptive to initially treat a smell source as food, until later information indicated otherwise. One possible explanation for this may lie in the design of the fitness
function. The height of the fitness is completely determined by the number of food patches collected by the agent. Indirectly the number of food patches is also affected by predator avoidance. If the agent can avoid a predator, it will live longer and will thus be able to gather more food. The profit margin may, however, be small. Running away from the predator costs energy. These costs might be too high compared to what is gained from the food collected while on the run. With inefficient parameter settings it could even occur that the agent loses energy while avoiding predators and gathering food. In that case it would be best to just run towards food and ignore predators, or even to run directly towards predators to minimize the losses. The agent’s life will be short, but at least it will not end with a highly negative amount of energy. Instead of attempting to tune the parameters of the fitness function a new fitness measure will be introduced in the next section. This measure attaches more importance to the agent’s ability to stay ‘alive’ for a long time.

**Simulation 4: Emphasizing Predator Avoidance**

To test whether the results of the previous simulation were indeed caused by the characteristics of the fitness function, the previous simulation was repeated, but now with a different fitness function, which explicitly rewarded predator avoidance.

**Simulation 4: Method**

In this simulation we measured the number of iterations it took before the agent was caught by a predator. This measure of lifetime was multiplied by the amount of energy left (measured in the same way as in the previous simulation) and this was taken as the fitness measure.

**Simulation 4: Results**

**Fitness:** The simulation was run for 1000 generations. The development of the fitness over generations is shown in Figure 52. The absolute value of the fitness can of course not be compared to the previous simulations, because it is qualitatively different. The overall shape of the curve is, however, similar to the previous simulation. There seem to be somewhat more fluctuations in the curve, this is probably because the new fitness measure varies more widely, i.e., the lifetime of the agents fluctuates more than the total collected energy. We assume that the small decrease of the curve near generation 1000 is also due to random fluctuation.

**Network analysis: In**

Figure 53 a schematic representation of the network of a representative individual is shown. The connections from the A-inputs to the outputs (I\textsubscript{AL} and I\textsubscript{AR} to O\textsubscript{L} and O\textsubscript{R}) are all excitatory. Thus, input from component A is mainly causing forward movement. Because the excitatory parallel connections are a little bit stronger than the crossed connections it will cause avoidance from the source.

The direct connections from the B-inputs to the outputs (I\textsubscript{BL} and I\textsubscript{BR} to O\textsubscript{L} and O\textsubscript{R}) are crossed-inhibitory, which also turn the agent away from the source. Since the direct route tendency is to avoid both smell components, both predators and food will be avoided.

Of the connections in the indirect route the connections to nodes H\textsubscript{1} and H\textsubscript{2} can be neglected because these have only small outgoing connections. From nodes H\textsubscript{3} and H\textsubscript{4} there are crossed-excitative and parallel-inhibitory connections to the outputs. H\textsubscript{3} and H\textsubscript{4} receive parallel activation and crossed-inhibition from the B-inputs (I\textsubscript{BL} and
I_{BR}), and the connection strengths from the A-inputs are quite small. The indirect route thus contributes to turning behavior towards a source of component B, which is strongest in food.

This network analysis thus indicates that this simulation resulted in a way of encoding which corresponds to LeDoux' hypothesis. The direct route encodes avoidance for both predator and food, whereas the indirect route encodes food-approach behavior.

---

![Graph](image)

**Figure 52.** Fitness (averaged over groups of ten consecutive generations) as a function of generations in Simulation 4.
Figure 53. Schematic view of the connections of a representative individual in Simulation 4. Connections to H1 and H2 have been grayed because H1 and H2 have only negligible outgoing connections strengths.

**Mapping Charts:** The mapping charts for food input confirm that the direct path (middle row) and indirect path (bottom row) perform completely different functions (see Figure 54). Their surfaces are almost mirror images of each other. The direct path turns the agent away from food whereas the indirect path turn it towards food. In the top row of the figure, where both paths contribute to the output, we see that the corrective function of the indirect path is dominant, and in effect the agent tends to turn towards food.

The agent’s response to predator input (Figure 55) is the same for the direct and indirect routes (middle and bottom row, respectively). The Mapping charts of the direct route have a much more horizontal surface, indicating that the agent turns away slowly from predators. Overall this surface is much higher than that of the indirect route, thus, the agent will respond with fast movement when presented with predator input. The surface of the indirect route has a steeper slope, resulting in a strong turn away from the predator. The intact network has a high flat area near the diagonal ($I_L = I_R$), resulting in fast forward movement (direct path). Further removed from the diagonal the difference between left and right output activation becomes larger, for these inputs the agent will thus respond with turning behavior.

**Rotation analysis:** The network analysis and Mapping charts both seem to indicate a solution in accordance with LeDoux’ theory. In the rotation analysis it can be seen whether presentation of food will indeed cause an initial avoidance reaction followed by a correction. This will be done by monitoring the first five iterations after presentation of a stimulus, using the same method as described in the previous simulation.

In the right column of Figure 56 the reaction of the agent is shown to predator input. At $t_2$ the agent starts to turn away from the predator. The rotation is largest at input angles of 90 and -90 degrees. This direction of rotation continues in the remaining time steps.

The agent’s reaction to food input is shown in the left column of Figure 56. Its first reaction at $t_2$ is a turn away from the food source. At $t_3$, however, we see the onset of a rotation in the opposite direction. For all inputs the turn away from the food has become smaller, and for the smallest input angles the agent even has turned its nose towards the food again. At $t_4$ and $t_5$ the turn towards the food is continued. What we see is thus an initial avoidance reaction caused by the direct route at $t_2$, corrected by the slower approach reaction caused by the indirect route at $t_3$. Note, that at $t_3$, the direct route still activated the output nodes for avoidance, but the correction from the indirect route overrules this.
Figure 54. Left and right output activations as a function of food input combinations in Simulation 4.
Figure 55. Left and right output activations as a function of predator input combinations in Simulation 4.
Figure 56. Behavior of the agent receiving food input (left column) or predator input (right column), with the input source presented for a number of different angles in Simulation 4. The chart depicts the angle of the agent with respect to the starting position. The rows correspond with the number of iterations after the start of the input.

Simulation 4: Discussion

The purpose of this simulation was to show that under the right circumstances the network would develop differential processing in the dual route architecture as put forward by LeDoux (1996). In the previous simulation we also obtained a different
functionality of the two routes. In that simulation, however, the direct route treats both input types as food, whereas in the current simulation both input types are treated as predators. All three methods of analysis supported this conclusion.

It can be concluded that there are three factors important for the evolution of dual processing. First, processing of the stimuli should actually cost time (Simulation 2). Second, the stimuli have to be hard to discriminate, so that processing through the indirect route is required to respond appropriately (Simulation 3). Third, and this is the contribution of the current simulation, predator avoidance should make a significant contribution to the fitness of the individual.

Simulation 5: Overlap without Time Delay

LeDoux’ theory suggests that what caused the dual route processing found in the previous simulation was that it allows faster reactions to predators. An alternative reason for such development could be the historical order of development in evolution. As Gould and Lewontin (1979) have argued, many things in nature have not evolved because they are optimal solutions. The darwinian process adapts creatures to the local circumstances at a certain point in time, subsequent adaptations are built on top of that. If predator avoidance, for instance, happens to be a more important aspect for survival, it will develop first, and will probably develop in the direct route because it is easier to find a solution there due to the presence of only 8 connections. When inspecting various simulations during runtime we did indeed observe that the direct route converged earlier on a solution than the indirect route, the nature of this solution, however, depended on the conditions. Whatever developed in the indirect route would then depend on the particular solution first found in the direct route. The indirect route would develop those properties that were not fully developed by the direct route. Northcutt & Kaas (1995) have such a view on the development of the emotional systems.

To investigate this alternative hypothesis a simulation was performed which did not have the time delay but was otherwise identical to the previous simulation. If it is indeed the speed advantage that causes this solution, then no dual route organization should be without it. If the historical order of development is the cause then this simulation should be no different from the previous one.

Simulation 5: Method

The setup of this simulation was identical to the previous simulation except that there was no time delay. The $L$ parameter was set to 1 again (immediate update), thus, signals travelling through the indirect route would arrive at the output layer at the same moment as signals travelling through the direct route.

Simulation 5: Results

Fitness: After 1000 generations the fitness reached a value of about 6000, which is twice as high as the fitness reached in the previous simulation. This higher level of fitness is, of course, due to the fact that the agent is no longer at a disadvantage to the predator, which did not have any time delay in any of the simulations. The development of the fitness over the generations was quite remarkable (see Figure 57). Up until generation 600 there was hardly any increase in fitness. But after that point there was a sudden fast increase in fitness. Before the sudden jump the genetic algorithm was probably stuck in a local minimum of the fitness landscape, where it
escaped from at generation 600. It is not completely clear whether the fitness increases further beyond generation 1000, but in order to be able to compare simulations we did not prolong this simulation.

Figure 57. Fitness averaged of clusters of ten consecutive generations as a function of generations in Simulation 5.

Figure 58. Schematic view of the network of a representative individual in the last generation of Simulation 5.
Network analysis: The schematic representation of the connections in the network is depicted in Figure 58. From the A-inputs to outputs there are strong crossed-excitatory connections, and also weaker parallel-excitatory connections. Activation of the A-inputs would, thus, cause a forward movement and a slight turn towards the source of the A-component. Because smell A is stronger in predators, we would expect a turn towards predators. The connections from the B-inputs are parallel-inhibitory, and have no crossed connections, so activation of the B-inputs would also cause approach behavior. Because both components cause approach behavior both food and predators will be approached by the direct route in isolation.

Hidden units H3 and H4 are always inhibited by the inputs, so they will only have low activation and a marginal effect on behavior. For this reason the connections from these nodes have little effect. The connections from H1 and H2 to output have parallel-excitation and crossed-inhibition. This would result in avoidance behavior if these nodes directly received input. Because the A-inputs copy their activation through parallel-excitatory connections to these hidden nodes a source of smell A would be avoided. This effect is enhanced by small crossed-inhibitory connections from the A-inputs, and some small parallel-inhibitory connections the B-inputs.

Mapping charts: The charts for food input (Figure 59) show clearly that the direct path is mainly responsible for the behavior with respect to food. The chart for the indirect route is almost completely flat. There could in principle be inhibition of the output layer coming from the hidden layer, but this does not seem to be the case as the intact and direct charts are very similar.

Behavior with respect to predator input is less easy to interpret (see Figure 60). The direct route shows high output values, with a slightly tilted surface, which would correspond to approach behavior. The indirect route is, however, strongly tilted in the opposite direction, which will cause avoidance behavior. It is comprehensible how the behavior of the intact network relates to its components. It has the sharp curve on the diagonal, which is probably the contribution of the indirect route. And it has an overall higher surface than the indirect route, which is probably the contribution of the direct route.
Figure 59. Left and right output activations for food input in Simulation 5.
Figure 60. Left and right output activation for predator input combinations in Simulation 5.

**Simulation 5: Discussion**

The main finding from this simulation is that the connection configuration of the current simulation is very different from the previous simulation. The time delay that was present in the previous simulation is thus responsible for the dual route organization found there. In this simulation the direct route approached everything in the environment, whereas in the previous simulation it avoided everything. In this case it is thus not the order in which things develop which determines the eventual solution.

It was again found, as in earlier simulations (e.g., Simulation 1), that with predator input the direct route is primarily important for speed, whereas the indirect route imposes direction. Apparently, this solution is quite general and is not the consequence of optimization with respect to speed processing. In this case historical order of development may in fact have a causal role In the simulated evolution, complex processing solutions, as may develop in the indirect route, arise after more generations that solution requiring only few weight changes. Perhaps the direct route, which tends to develop early in our simulations, develops speed first because even
undirected movement can help to escape predators sometimes, whereas turning without speed, does not. Of course, the fast development of processing of both speed and direction in the direct route would be optimal, but due to the overlapping smell input this cannot be combined in a simple fashion with food approach behavior.

Conclusions

We set out to investigate the conditions under which adaptive dual processing is developed in a dual route architecture as hypothesized by LeDoux. To this end we set up an artificial environment in which the selection pressures were implemented. In the simulations, speed restrictions and critical selection criteria were systematically varied. When these factors matched LeDouxs' evolutionary reasoning most closely dual processing was indeed obtained in the dual route network.

Three factors were shown to be relevant for the evolution of the dual route architecture. First, there should be a difference in response time between the direct and indirect routes. Thus, there should be a time delay for each step in the neural network, and there should be a difference in the length of the two routes. Second, the 'problem' needs to be sufficiently difficult to actually necessitate the more accurate processing which can be added by the slow long route. This was implemented by encoding the two stimuli with overlapping patterns. Third, the importance of the time critical stimulus needs to be relatively high for it to have an effect. A factor not directly following from our simulations but which followed from our initial analysis is that stimuli need to differ in how critical a fast reaction is. In this simulation the predators needed a fast response, whereas the food did not.

The kind of research methods applied to this problem are quite new, especially as a tool to investigate psychological or neurobiological hypotheses. One limitation of this approach is that many parameters have to be chosen, and in a strict sense, conclusions can only be drawn with respect to these specific parameters. On the other hand if one investigates a psychological phenomenon in a laboratory experiment on human subjects, in a strict sense, we can only say something about human behavior in the specific settings of the experiment. In practice, we can generalize such results somewhat further because we can reasonably assume slightly different stimuli will show similar results. In principle, however, we can only demonstrate generality of a specific finding by doing more experiments under different circumstances (although restricted to the domain to which one intends to generalize) that produce the same results. Here too, generality can only be demonstrated by showing that the same principle applies to other setups with those same constraints. Such a setup could, for example, even make use of real robots instead of simulations. This is outside the scope of this study, however.

With regard to the choices that had to be made for the parameters, we need to stress that they were based on general knowledge of physical systems and biological evolution and were generally independent of LeDoux' specific hypothesis. Although some tuning of parameters was necessary, this was only done to optimize the performance in the evolution of the agent. In fact, most of this tuning was done with single layer neural networks, before the simulations started, and thus, could not be influenced by our hypothesis with respect to the dual routes.

The evolutionary reasoning by which LeDoux (1996) justified the neurobiological architecture in fear conditioning, can be seen as an exponent of evolutionary psychology. The general idea in this reasoning is that in evolutionary history our species and its predecessors have encountered many similar challenges,
for which adaptive solutions have been found, and which have been genetically passed on to every new generation. It is of course an advantage that such behavior is inherited and does not need to be learned over and over in every individual. With respect to LeDoux’ dual route processing, inheritance is a particular advantage, because the possible penalty of death that could result from a late reaction leaves not much room for learning by trial and error. Such evolutionary reasoning, however, leaves much room for speculation about possible mechanisms. By applying the research tool of genetic algorithms (Holland, 1975) such evolutionary theorizing can be constrained. It forces us to specify the specific circumstances we expect to result in a certain solution, and it makes many of the preconditions for such an evolutionary path explicit. In this experiment we have shown that LeDoux’s evolutionary reasoning stands up to an empirical test, albeit, within the constraints we have mentioned above.

The neural network investigated here appears to be one of the simplest in which LeDoux’ dual processing framework can still develop. Undoubtedly, the actual neural network is much more complicated, and the computational evolution of this network would exceed present day computing capacities. The fact that two different network architectures, one artificial and one actual, both can show the postulated dual processing forms a strong argument in favor of the generality of LeDoux’ evolutionary processing theory.

Appendix A: Smell gradient

Equation 1. \( \delta < \delta_{MAX} : \quad S = \frac{1}{1 + \delta \cdot \text{Curve}} \cdot S_{MAX} \cdot \left(1 - \frac{\delta}{\delta_{MAX}}\right) \)

\[ \delta \geq \delta_{MAX} : \quad S = 0 \]

\( S \) = Smell intensity at the sensor,
\( S_{MAX} \) = Smell intensity of the source,
\( \delta \) = Distance between source and sensor,
\( \delta_{MAX} \) = The maximal distance at which the source could be smelled,
\( \text{Curve} \) = A parameter determining the curve of the hyperbolic function.

During these simulations \( \delta_{MAX} \) was set to 100, \( \text{Curve} \) to 1 and \( S_{MAX} \) to 25. For smell components of half strength, as used in the later simulations, \( S_{MAX} \) was 12.5.

Appendix B: Network activation rule

Equation 2. \( y_i = \sigma \left( \sum_{j=1}^{N} \omega_{ij} y_j + \theta_i + S_i \right) \)

The function for the sigmoid is:

Equation 3. \( \xi \geq 0 : \quad \sigma(\xi) = \frac{\xi}{1 + \xi} \)

\( \xi < 0 : \quad \sigma(\xi) = 0 \)

\( y_i \) = The activation of unit i,
\( \omega_{ij} \) = The strength of the connection from unit j to unit i,
The number of connections to unit $i$, $N$.

The bias of unit $i$ (0.1 in the simulations), $\theta_i$.

Smell input to unit $i$ from the sensors (only for input layer), $S_i$.

**Appendix C: Energy consumption per motor actuator**

**Equation 4**

$$ Energy_i = \frac{\text{MaxE}}{1 + \text{MaxA - MotorAct}_i/\text{MaxA}} $$

Energy$_i$ = The amount of energy burned at time $t$.

MotorAct$_i$ = The activation of the motor actuators at time $t$.

MaxA = The maximum activation of the motor actuators.

MaxE = The maximum energy the agent can use at a time step.

In all simulations MaxA was set to 1 and MaxE was set to 0.001.

**Appendix D: The genetic algorithm**

I. The chances of the rosette wheel

The procedure for the rosette wheel:

Add all $P_i$ (see Formula A.1) together until the sum is larger than $G$.

Select the individual $i$ who’s $P_i$ was added to the sum at the moment $G$ was exceeded.

With this procedure $P_i$ determines the chance with which $i$ will be selected. $P_i$ increases as $D_{ij}$ decreases or $F^*$, increases.

**Formula A.1:**

$$ P_i = c_1 \left( \frac{F_i}{\text{SumF}} \right) + c_2 \left( \frac{1}{\text{SumD}} \right) $$

$P_i$ = the probability that individual $i$ will be selected

$F_i$ = the fitness of individual $i$ – the lowest fitness of the population

$D_{ij}$ = the distance of individual $i$ to individual $j$

c$_1$ = the factor determining the weight of the distance $D_{ij}$

c$_2$ = the factor determining the weight of the fitness $F_i$

SumF = the sum of the fitness of all $N$ individuals

SumD = the sum of $1/D_{ij}$ over all individuals, except individual $j$

$G$ = $r_1 + r_2$

$r_1$ = a random number $r$, multiplied by $c_1$

$r_2$ = a random number $r$, multiplied by $c_2$

II. Circular cross-over

The procedure for cross-over forms one new DNA-string based on the two parent strings. Both parent strings are represented as a circle. For example Parent1 ($a_1,a_2,a_3,a_4,a_5$) and Parent2 ($b_1,b_2,b_3,b_4,b_5$) become:
Chapter 6

From Parent1 a part is taken away. This is replaced by the equivalent part of Parent2. If the starting point of the replaced part is between $a_4$ en $a_5$ and the endpoint between $a_1$ en $a_2$, then the progeny will look like this:

Crossover occurred with a probability of 0.6 in the simulations. Without crossover, the child is constructed from the connections of the first parent alone.

III. The mutation operator

Formula A.2:

$$M = -m \cdot \log\left(\frac{1}{r}\right)$$

- $m$ = mutation parameter (0.1 in the simulations),
- $r$ = a random number between zero and one,
- $M$ = the value that is added to the weight.

A mutated weight is equal to the sum of the old weight and $M$. If $r$ is equal to 0.5 then $M$ is zero, if it is smaller $M$ is negative, if it is larger $M$ is positive. The probability a weight increases is, thus, equal to the probability that it decreases. If the sum of the weight and $M$ is outside the domain [-10,10], the weights will be set to either -10 or 10. $M$ increases or decreases logarithmic with $r$. The effect of the function is that mutations will mostly be small, but will sometimes be large. Thus, most progeny will be similar to the parents (exploitation) but will sometimes be quite different (exploration).

IV. Determining the probability of survival

The probability of survival is calculated in three steps. First a probability is calculated based on the individual’s own fitness with respect to the highest and lowest fitness in the population (see Formula A.3).

Formula A.3:

$$P_i = \frac{b \cdot (F_i - F_{\text{min}})}{F_{\text{max}} - F_{\text{min}}} + o$$

- $P_i$ = the probability of survival after the first step
- $F_i$ = the fitness of individual $i$
- $F_{\text{min}}$ = the lowest fitness of the generation
- $F_{\text{max}}$ = the highest fitness of the generation
- $o$ = the lowest possible value of $P_i$ (0.15)
- $b$ = the possible range of $P_i$ (1-o = 0.85)
This function will normalize $P_i$ between 0.15 and 1.0. The lower boundary guarantees at least some chance of survival for individuals with a low fitness.

In step two the probability is corrected in order to maintain a population size around the initial population size. In case the current population is smaller than the original population the probability is increased, in case it is larger the probability is lowered (see Formula A.4).

Formula A.4:

$$\begin{align*}
N \geq N_0 : & \quad P_2 = P_1 \cdot \frac{N_0}{N} \\
N < N_0 : & \quad P_2 = 1 - \left(1 - P_1 \right) \cdot \left(1 - \left(1 - \frac{N}{N_0}\right)^2\right)
\end{align*}$$

- $P_2$ = the survival probability after the second step
- $P_1$ = the survival probability after the first step (Formula A.4)
- $N$ = the current population size
- $N_0$ = the initial population size (at time zero)

If the current population is larger than the initial population $P_i$ is multiplied with the quotient of the initial population size and the current population size. This is quotient is always between zero and one. If the current population is smaller it is multiplied with the inverse quotient. This second part needs some extra terms to keep the probability between zero and one.

In the third step the probability is multiplied with 0.99 (see Formula A.5).

Formula A.5: $P_3 = 0.99 \cdot P_2$

- $P_3$ = the eventual probability of survival
- $P_2$ = the probability of survival after step two

This part ensures that even the individual with the highest fitness has at least some chance to be removed from the population. The best individual would have a probability of 1.0 without it.