14

What Neuroevolution Can Tell Us About Biological Evolution?

In previous chapters, several examples were given of using neuroevolution to discover behavior for intelligent agents. The goal was to construct artificial agents that could perform complex tasks in aid to humans, potentially in virtual worlds, household robots, autonomous vehicles, etc. However, the approach can also be useful in the other direction, i.e. using neuroevolution to understand biological intelligence. Why do certain neural structures exist in the brain, i.e. what do they do and how did they come about? How do the genetic and environmental influences combine to construct an individual? What are the stepping stones in the evolution of intelligent behavior? How do behaviors such as herding, hunting, and communication emerge? This chapter will review progress towards answering these questions, and identifies further opportunities in them.

14.1 Understanding Neural Structure

Neuroscience aims to understand how the brain produces behavior. The neural structures in the brain are highly organized into nuclei, or collections of neurons, and pathways between them, and the goal is to identify what functions they each perform individually and through interactions. Single-cell recordings have been used for a long time to identify such function at a low level, for instance identifying cells that respond to a particular location in the visual field, and a line of a particular orientation and direction of movement in it (Hubel and Wiesel 1968). More recently, several broader imaging techniques have been developed to look at larger areas of the brain at once: voltage-sensitive dye imaging can visualize entire maps, diffusion tensor imaging entire pathways, and, EEG, MEG, and fMRI even the entire brain at once (Chemla and Chavane 2010; Meoded et al. 2017; Lenartowicz and Poldrack 2010). Sensory and motor functions are already understood relatively well, and much progress is made in delineating higher functions such as reasoning and language.

However, one important perspective that is often missing in such inquiries is that the structures are a product of evolution. Part of what we observe today may not be explained simply as serving a function in some optimal sense. Some of the structure is there because evolution needed to discover it: It may not be optimal or necessary, but is instead a remnant of evolutionary stepping stones. Humans still have tailbones even though we no longer have tails. Speech organs look the way they do because they evolved from mastication elements (MacNeilage 1998). Similarly, in order to understand brain structures and behavior fully, it may be necessary to understand their evolutionary origins.

Although the brain microstructure varies between individuals, the high-level organization is remarkably consistent between individuals and between species. Evolution has come up with a successful solution and has created many variations of it that occupy multiple niches in the world. A possible approach to understanding the brain is to create artificial worlds, place artificial agents in them to face various challenges, and evolve their brains to construct behaviors that allow them to survive and be successful. By manipulating the environment, it may be possible to determine what structures are likely to evolve and why. To the extent that they match those observed in biology, it may be possible to gain insight into biology.

For instance, in one such grid-world simulation, an agent first needed to navigate to a zone where food items are located, while avoiding poison obstacles, and then to remain in that zone and forage (Figure 14.1; Aharonov-Barki, Beker, and Ruppin 2001; Ruppin 2002). The agents were controlled by a fully recurrent binary neural network with five sensory, four motor, and 6-41 hidden neurons. After successful behavior had evolved, the hidden neurons were analyzed through conventional neuroscience methods of lesioning and receptive field analysis. Remarkably, the successful networks had evolved a command neuron (or a few) that essentially switched the network between the navigation and foraging behaviors. The network starts by navigation, but as soon as the agent consumes a food item, the command neuron switches it into foraging. Such command neurons emerged in evolution because they resulted in higher fitness: Individuals that were able to separate the navigation and foraging behaviors found the food zone faster, avoided poison better, and were able to forage more efficiently than those that mixed the two behaviors.

Interestingly, command neurons are found in many biological systems as well, including aplysia, crayfish, and even lobsters and crabs (Edwards, Heitler, and Krasne 1999; Teyke, Weiss, and Kupfermann 1990; Combes, Meyrand, and Simmers 1999; DiCaprio 1990). They generally switch motor behaviors on and off based on sensory input, similar to the command neuron that were evolved in the simulation. Thus, the simulation demonstrates computationally not only how such a network implements effective behaviors, but also can arise in evolution as a solution to a computational need.

Beyond the single-neuron lesion and receptive field analysis, the full access that computational networks provide makes it possible to analyze the solutions in more detail. For instance, multiple small perturbations to the network's neurons or connections can be introduced, and the contribution of each of these elements quantified by estimating its Shapley value (a game-theoretic measure of contribution to a collaboration; (Keinan et al. 2006)). Such an analysis makes it possible to identify the role of each element in constructing a function, and it also makes it possible to prune the network by removing elements that do not contribute significantly. Although developed for analyzing evolved artificial networks, the technique could in principle be adapted to neuroscience, for instance based on multiple lesions, or on perturbations caused by TMS (transcranial magnetic stimulation).

Neuroevolution simulations can be useful in evaluating hypotheses about function of specific circuits. For instance, facilitating synapses (Markram, Wang, and Tsodyks 1998) have been observed to activate postsynaptic neurons not only based on current input but also based on a rate of activation change in the past. Most likely they play a role in processing temporal sequences, but they may also be useful in compensating for propagation delays (Lim and Choe 2006; Kwon and Choe 2009). Although such delays are not taken into account in abstract neural networks, in biological networks delays are an important factor.

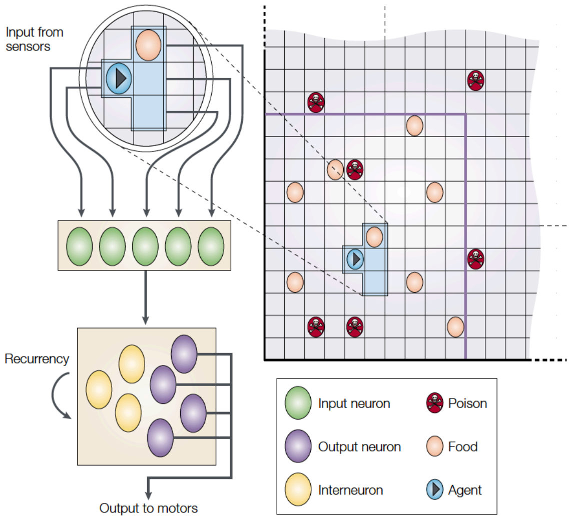


Figure 14.1: **Evolution of command neurons in a navigation and foraging task**. In the simulated grid world, there are a number of poison and food items. The agent needs to first navigate to the 10 × 11 bottom left area where the food items are, eat as many of them as possible, and avoid poison items at all times. The agent's behavior was controlled by neural networks that were evolved through genetic algorithms over time. Some of the evolved interneurons act as command neurons, switching the behavior from navigation to foraging as soon as the first food item is consumed. Similar command neurons have been observed in biology; the experiment demonstrates how they may arise as an advantage in evolving effective behavior in the domain. (Figure from Ruppin 2002)

Information from the sensors takes time to propagate to neurons that react to it, and proper responses to e.g. a moving object require compensating for these delays. With neuroevolution, it is possible to construct facilitating synapses that play this role, resulting in more accurate performance in tasks such as pole balancing with synaptic delays. Such compensation amounts to rudimentary prediction, and suggests that coping with synaptic delays may be a foundation for predictive mechanisms, which have been proposed to underlie much of cognitive processing (Hawkins and Blakeslee 2004; Hawkins and Ahmad 2016).

Neuroevolution simulations can also be used to target specific biological behaviors. For instance, such experiments have been useful in understanding locomotion circuits in animals (Chiel, c, and Gallagher 1999; Beer, Chiel, and Gallagher 1999). Such circuits are often called CPGs, or central pattern generators, because they provide a cyclical activity pattern than can be used to control the gait through multiple muscles (Buzsáki 2006; Steuer and Guertin 2019). Such networks are relatively small, consisting of three to five continuous-time recurrent neurons. However, they generate complex dynamics that also changes over time. The simulations made it possible to characterize such dynamics mathematically and experimentally, and demonstrate how such neural systems can be composed of multi-stable dynamic building blocks. In some cases it was possible to assign functional roles to these blocks, in others they remained opaque as supporting interneurons.

These mathematical characterizations of CPGs were expanded into simulations of actual locomotion in lampreys and salamanders, both in swimming and walking (Ijspeert 2008; Ijspeert et al. 2007). The evolved networks coordinate the oscillatory patterns of the CPGs as inputs to the two legs on each side of the body, resulting in motions required for effective propulsion. Remarkably, such evolved controllers resulted in more robust patterns and flexible control than a model that was built by hand. Also, the oscillation patterns and the connectivity structures were closer to those observed in biology, again demonstrating how the biological structures may arise from evolutionary pressure to perform well wrt. a behavioral challenge in a physical environment. Moreover, the same circuit can control both swimming and walking, as well as transitions between them, potentially demonstrating a crucial phase in the vertebrate evolution from aquatic to terrestrial.

Beyond pattern-generator circuits, a more general question concerns network building blocks. Evolved neural networks often include identifiable motifs, i.e. patterns of connectivity that occur more frequently than they would in randomly generated networks (Kashtan et al. 2004; Kashtan and Alon 2005). It turns out that these same motifs can also be found in biological networks. Thus, computational simulations can then be used to identify what function they may perform. For instance, the feedforward loop motif can be used to filter information, generate pulses, and increase responses, and the single-input motif can generate time-varying gene expressions. Evolved neural networks can then demonstrate how behavior is composed of such building blocks, for instance uncovering spatial specialization in a visual pattern recognition circuit.

Beyond understanding motif function, neuroevolution can be used to illustrate how motifs, and more generally modules, emerge. It turns out that if the network is evolved to simply solve one task, they are unlikely to arise. However, if the environment requires solving multiple goals composed of different combinations of subgoals, and the goals change over time, modular network structure and motifs do arise. In this manner, evolution finds modularity as an effective way to discover subfunctions that can be used to construct multiple behaviors. Indeed, the modular structure of the brain supports this hypothesis: many areas of the brain participate in many tasks in different combinations. Even the visual areas are used in some language tasks and vice versa, suggesting that their computational function is more general than just one modality. Neuroevolution studies can thus demonstrate this general principle as a solution arising from the complexity of tasks the animal has to solve.

Because neuroevolution is an optimization method, it can also be used in a different role in understanding neural structure: Instead of evaluating their evolutionary origins, to optimize the model parameters. Biophysical models are created with objectives and constraints derived from experimental data. They often contain parameters that are difficult to set right to match the data, but can provide insights into the biological structures and processes. Neuroevolution can be effective in this role: It has been used for instance in optimizing the spiking patterns on the Izhikevich model of hippocampal neurons (Venkadesh et al. 2018) and fitting multicompartmental models to multilocation patch-clamp and microelectrode array data (Druckmann et al. 2007; Buccino et al. 2024). Interestingly, as discussed in Section 11.4, neural network implementations in hardware often utilize spiking neural networks to reduce energy consumption; it has turned out useful to optimize their structure and hyperparameters through evolution C. Schuman et al. 2022; Iranmehr et al. 2019. Neuroevolution can thus realize the potential of such biologically more accurate models, suggesting how behavior can arise from the biophysical properties expressed in their parameters.

Neuroevolution simulations can also be used to explore other hypotheses about the development of modularity and organization. One such hypothesis is to minimize the total wiring length, as will be discussed next.

14.2 Evolutionary Origins of Modularity

Given that the primary role of the brain is to process information, it is natural to try to explain its entire structure and function in computational terms. However, it is sometimes useful to recognize that the brain is also a physical organ, and there are physical requirements that must be met. For instance, some of the brain structure may be due to the need to maintain efficient metabolism, i.e. to bring oxygen and nutrients to the cells, including the vasculatory structure and the blood-brain barrier. While bigger brains in general are more powerful, the size of the brain is limited by the birth canal. Some of the growth mechanisms after birth may exist to compensate for it, rather than be driven entirely by need to construct an efficient information processing system. Similarly, the overall organization, with gray matter on the outside and white matter on the inside, and the highly convoluted surface with gray matter, amounts to an efficient use of the available space.

The need to minimize wiring length is an important principle that may have affected the evolution of brain structure more generally (Sporns and Betzel 2016; Horvát et al. 2016). In particular, it may be the evolutionary origin of modularity. This is an interesting possibility because modularity is also a powerful functional principle. While a tightly connected system may in principle provide more complex functionality, it is more difficult to construct, maintain, and adapt a system where everything depends on everything else. For instance in engineering, modular structures are often used because they make such processes easier. For these same reasons, evolution may have favored modular designs as well.

However, such pressures are relatively weak compared to simply performance, and it has been difficult to demonstrate this theory biologically and computationally. In contrast, it turns out possible to demonstrate that minimization of wiring length can play a primary role in the evolution of modularity; the functional advantages then emerge as a secondary, reinforcing side effect (Clune, Mouret, and Lipson 2013).

Computational experiments were set up to compare the evolution of neural networks in a visual object recognition task under two conditions: with a single objective of maximizing performance alone, and with two objectives of maximizing performance and minimizing wiring length simultaneously. Since wiring length is presumably less important for survival than performance, it was set to affect selection only 25% of the time. Wiring length was measured as the total squared length of all connections and NSGA-II was used to construct a Pareto front of the two objectives.

The task involved an eight-pixel retina where an object might appear either in the left or right half, or both (Figure 14.2). Note that it is indeed possible to decide whether there is an object on the left/right half before combining these decisions; the task should therefore lend itself to modular solutions. Performance was measured simply as the percentage of correct answers. Simple feedforward networks with three hidden layers were evolved in

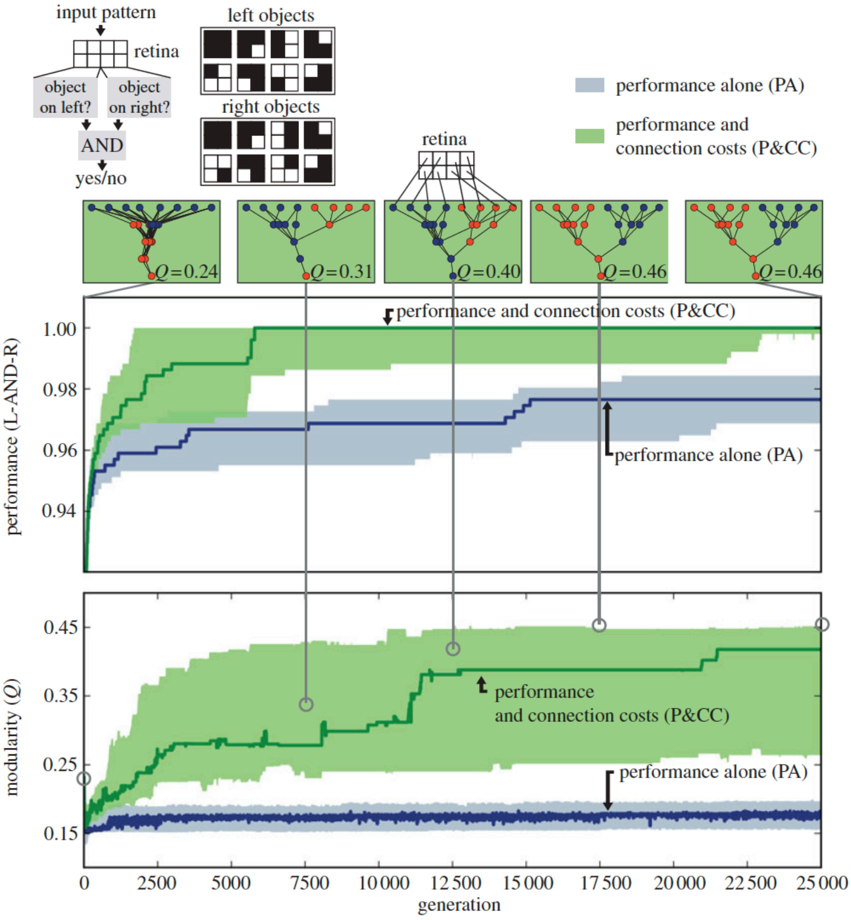


Figure 14.2: **Evolution of modularity based on maximizing performance and minimizing wiring length.** The goal was to evolve a visual system to locate and identify objects. (*a*) Objects appear on the left and/or the right side of the retina and the network needs to decide whether there is an object in both. (*b*, *d*) With the objective to minimize wiring length, more modular networks evolve over time. (*c*) Modular networks also perform better, although there are some well-performing nonmodular networks as well. Computational simulations thus suggest that wiring length is the primary evolutionary pressure behind modularity; performance and adaptability pressures may further enhance it. (Figure from Clune, Mouret, and Lipson 2013)

this task. They had integer weights and thresholds, and mutations to add or remove a connection and increase or decrease a weight or a threshold. The networks were initially set up randomly; their modularity was measured by first dividing the networks optimally into modules, and then comparing the density of connections within each module to that of a randomly connected network (M. E. Newman 2006).

In 25,000 generations, the performance+wiring-based evolution resulted in more modular networks than the performance-based evolution. Such structural modularity resulted in functional modularity as well: The modules often corresponded to making a decision on the left or the right side. Interestingly, many such networks actually performed better than those that were evolved only to maximize performance. They were generally smaller and therefore perhaps easier to optimize; a good nonmodular network may also be more difficult to find. The networks with the shortest wiring length were more likely to be modular. However, evolution did find some well-performing nonmodular networks as well, suggesting that modularity does not arise from performance alone.

The modular networks also turned out to be more evolvable. In further experiments, networks were evolved in a sequence of two tasks: they were first evolved to answer whether an object appeared both left and right, and once they had learned this task, further evolved to answer whether an object appeared in either left or right (the opposite order of tasks was also run). The modular networks required fewer generations to adapt to the new environment, and they were more modular than in an unchanging environment. The results thus suggest that modularity evolves primarily due to wiring length, once it is there, it is further enhanced by the need to adapt. Thus, neuroevolution simulation can be used to gain insights into the origins of modularity in biology.

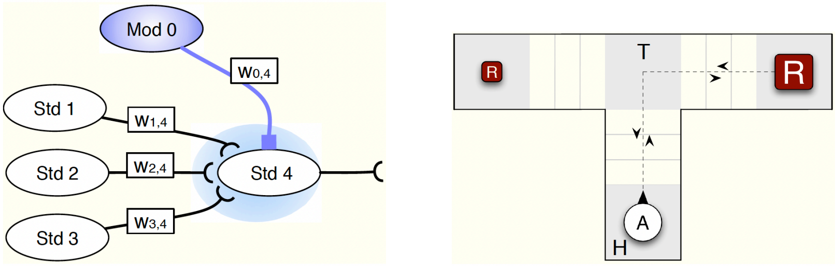
14.3 Understanding Neuromodulation

As has been mentioned several times in this book, there are many biological constraints and mechanisms that are likely to have an effect on neural function, but are not included in the standard neural network models. Another such mechanism is neuromodulation. That is, some neurons have a multiplicative effect on the weighted sum of inputs, or on the Hebbian weight change. Such modulation can lead to more complex behavior and more powerful adaptation.

For instance, backpropagation can be extended to multiplicative neurons in a straightforward manner. The gradient descent equations can be derived for such connections, resulting in sigma-pi units (sigma represents the sum of inputs, pi represents the product of multiplicative inputs). This method results in smaller networks: for instance, XOR can be represented in just three units: one computing AND, one OR, and one selecting between them multiplicatively (Pollack 1987; Rumelhart, Hinton, and Williams 1986). Scaling up, such networks have been useful in for instance recognizing whether a string adheres to a particular grammar: a single symbol at a wrong place can change the decision, which behavior can be represented well by multiplicative connections (Giles et al. 1991). Such networks can be evolved just as well as weighted-sum networks, achieving the same benefits.

An interesting question is whether neuroevolution would select for neuromodulation in order to solve a task, that is, whether it would emerge in evolution as an adaptive advantage. In one such experiment, neuromodulation was set to modify plasticity in Hebbian networks, i.e. those where a connection strengthens when both presynaptic and postsynaptic neurons are simultaneously highly active (Soltoggio et al. 2008). In contrast with backpropagation, which is an abstraction of learning in biological neural networks, Hebbian plasticity is an actual plasticity mechanism in biology. Connection weights were adapted as

$$w_{ji} = \eta \tanh(o_m)(Ao_jo_i + Bo_j + Co_i + D), \tag{14.35}$$



(*a*) Neuromodulation circuit (*b*) T-maze task

Figure 14.3: **Taking advantage of neuromodulation in the maze navigation task.** Neuromodulation offers a dimension of adaptation that may make it easier to solve complex tasks. (*a*) The three standard neurons activate the postsynaptic neuron through a weighted sum as usual. A modulatory neuron then amplifies the Hebbian adaptation of those weights. (*b*) The agent needs to traverse a corridor and then turn left or right to get to the larger reward; in a double maze (not shown), two such turns need to be made. The location of that reward changes periodically, and the agent needs to adapt its behavior accordingly. Networks evolved with modulation perform more reliably than non-plastic and non-modulatory networks, suggesting that evolution finds a way to take advantage of modulation even when it is not strictly necessary. (Figure from Soltoggio et al. 2008).

where η is the learning rate, o_m is the modulatory neuron output, o_j is the presynaptic activation and o_i is the postsynaptic activation, and A, B, C, and D are constants (Figure 14.3*a*). In this manner, the modulatory neuron controls whether the weight increases or decreases, and scales the magnitude of the Hebbian adaptation.

The approach was evaluated in the task of navigating a T-maze or double T-maze into a reward location, i.e. making the correct turn once or twice to get to the reward, and then navigating back to the starting location (Figure 14.3*b*). Each agent was tested 100 times, and at some point, the reward location changed, so it had to adapt its behavior. It could do so through recurrent connections that implemented memory, or by changing its weights through plasticity. The agent networks were evolved by inserting, duplicating, or deleting neurons, which could be either standard or modulatory, and by mutating the constants A, B, C, D, and η in Equation 14.35 and the real-valued weights through evolution strategy.

Even though the tasks were sometimes solved without plasticity and modulation, networks with plasticity evolved to perform significantly better in the 100 trials. Networks with modulation performed similarly in the T-maze, but significantly better in the double T-maze. The solutions had many different structures that were hard to interpret, but ablation studies showed that modulation plays an interesting role. When it was turned off from networks that were evolved with it, the networks still performed well locally, i.e. made turns and did not crash into walls. But they could often only turn in one direction, and could not navigate globally e.g. to find their way back to the starting location. This result suggests that neuromodulation is not simply an add-on that helps solve more complex tasks, but integrated into the dynamics of the navigation behavior. Successful behavior can be evolved without it, but solutions with modulation are easier to discover. They therefore evolve more reliably, resulting in better average performance.

A related experiment further suggested a possible biological mechanism for neuromodulation. In a stochastic reward optimization task, modulation activated reinforcement learning when it was most needed, allowing the system to adapt better to new scenarios (Soltoggio et al. 2007). Modulation was achieved through dynamics similar to dopaminergic activity recorded in the monkey's brain (e.g. Schultz 2024), giving it a computational interpretation.

The experiments thus show that the evolutionary process finds a way to utilize whatever dimensions of adaptation there are, rather than finding parsimonious solutions that ignore the dimensions that are not necessary. If neuromodulation is possible, neuroevolution will take advantage of it.

.

14.4 Developmental Processes

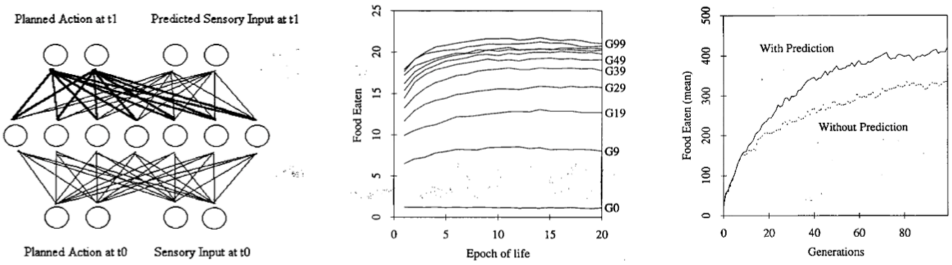
A fundamental question in cognitive science is how much of intelligent behavior in humans is innate, and how much is learned. This question is often referred to as the "nature vs. nurture" debate. Both of these factors play a role, of course, and are often synergetic through the process of development. Further, initial development, as well as long-term stability, can be driven by genetically directed learning, as will be reviewed in this section.

14.4.1 Synergetic development

Given the relatively small number of genes in the human genome (about 24,000; International Human Genome Sequencing Consortium 2004), a learning process is necessary to construct an organ as complex as the brain. On the other hand, genetic determination is also necessary: It can provide the overall structure, initialization, and a learning bias that then makes it possible to construct such complexity during the lifetime of the individual. Perhaps the clearest example of this process is language: All normal humans, and only humans, have an innate capacity for language. However, they need to learn a language in early childhood—language does not develop in isolation (Section 14.7.1).

For many animals, the fundamental survival skills are there right after birth. For instance, newborn gazelles can run immediately, and whale calves can swim. For higher animals, there is a long period of development during which they are dependent on their caregivers. This period is exceedingly long for humans, and includes a series of critical periods during which skills such as walking, talking, and social intelligence develop in an order—and if they do not, the individual will not be able to develop them fully later (Robson 2023). This observation suggests that the relationship between evolution and learning, that is, the process of development, is more nuanced and structured than simply refinement of a genetic starting point.

In principle, evolution can discover complete solutions that do not need to be refined further. Most of evolutionary computation is also based on this approach. However, in constructing brains, evolution seems to have discovered a different approach, described theoretically as synergetic development (Elman et al. 1996b). Instead of specifying a complete solution, only the general structure is genetically determined, together with a learning mechanism that allows the animal to construct the full solution. These components are synergetic: The structure and initialization make learning most effective, and the learning mechanism



(*a*) Network architecture (*b*) Lifetime learning (*c*) Evolution of foraging

Figure 14.4: **Synergetic development in a foraging task.** The creatures evolve to navigate to food items, aided by development to predict the consequences of their actions. (*a*) The evolved network is trained to predict how its sensory inputs change as a result of its cations in the previous time step. (*b*) Their prediction ability improves over their lifetime throughout evolution; even in later generations (near G99), it is not genetically encoded. (*c*) The development of prediction allows evolution to discover better solutions faster. Thus, the experiment demonstrates the value of synergetic development. (Figures from Nolfi, Elman, and Parisi 1994)

is well suited for the structure and the environment. The minimally functional initialization and the critical periods are part of this synergy. That is, instead of a fully specified design, evolution has discovered a developmental process as the solution. This approach can be seen as an implementation of expressive encoding, with the power to discover solutions that would be difficult to find through direct evolution (Section 9.1.4).

Computational studies can be instrumental in verifying this theory. An early example is an experiment with simulated creatures foraging for food items randomly scattered in a 2D grid world (Nolfi, Elman, and Parisi 1994). They receive the current (t_0) angle and distance to the nearest food item as their input, and generate an action (turn left or right, move forward, or do nothing) at the next time step (t_1) as their output. The creature's fitness corresponds to the number of food items it finds. The optimal actions are not known, but the entire network can be evolved to discover successful foraging behavior.

However, in this experiment, the creatures also receive their previous action (at t_0) as additional input, and predict the sensory input at the next time step (t_1) as additional output. These additional outputs are known, and therefore the network can be trained through gradient descent to predict the consequences of its actions. This training takes place during the lifetime of the creature, and the weight changes are not encoded back to the genome.

Thus, lifetime learning establishes a developmental process. The creature learns to understand how its actions affect its environment, much like biological organisms learn to interact with their environment. Such learning allows it to perform better at the task for which it is evolved, and it guides evolution to generate individuals that can take better advantage of the learning process (Figure 14.4). Note that the prediction ability does not become encoded in the genes; the individuals start with poor ability even in later generations. Evolution instead utilizes learning as part of the synergetic developmental process. As a result, creatures that perform better are discovered faster.

In this manner, computational experiments can be used to gain insight into how development works and why it is so powerful. One such insight is that evolution establishes the proper learning biases, and learning provides the variance necessary to adapt to the world, as will be discussed in the next section.

On the other hand, it may also be possible to build more complex artificial systems by employing these same principles. Progress in such systems, and further opportunities, are reviewed in Section 4.2.

14.4.2 Development through genetically directed learning

One way to characterize the synergy of evolution and learning is through the general machine learning concepts of bias and variance. Biases exist in any learning system, making it more likely to learn certain kinds of behavior, and less likely to learn others. In contrast, variance means that it can learn a wide variety of patterns that exist in the training data. A pure evolutionary system can be seen as completely biased with no variance: The behavior is determined genetically, and there is no learning based on input. In contrast, a pure learning system has no bias, and only learns the patterns in the input.

Neither of such extremes is likely to be very successful. It is difficult to anticipate all possible input situations ahead of time, during evolution. On the other hand, it is difficult to learn a robust function through high variance; the system is likely to end up overfitting and not generalizing well to new situations. Thus, a developmental system is a way to strike a balance between these two effects. Evolution establishes the proper bias, making it easier for the learning system to acquire a useful, robust function from the inputs.

The biases can be most directly established by evolving the learning system itself. For instance, parameters for Hebbian learning can be incorporated into neuron definitions and evolved together with the network itself (Dario Floreano and Joseba Urzelai 1999). Through the lifetime of learning with these parameters, controllers in a robot navigation task can be evolved faster than without learning. Evolution converges to learning parameters that are the most effective, thus finding a proper balance between bias and variance.

A biological example of this process can be seen in the domain of constructing a pattern recognition system (Valsalam, Bednar, and Miikkulainen 2007; Miikkulainen et al. 2005). Indeed, visual systems of animals are believed to combine nature and nurture in a systematic way: The general structure is genetically determined to match the needs of the species, and then fine-tuned through learning. For example, retinotopy and orientation sensitivity exists even before birth in cats and monkeys, but the full structure is formed during the first few weeks after the eyes open. Human newborns have an innate preference for face-like patterns, which is refined to actual face preferences during the first few months of life. It can also help explain other species-specific visual functions that appear innate, such as detecting prey (e.g. flies in frog vision; (e.g. flies; Lettvin et al. 1940)).

The way such preferences are established is particularly interesting. While it is possible to specify some neural network structure genetically, such as retinotopy, a learning mechanism also exists and may be active even before birth. Evolution seems to have discovered a clever way to utilize it even in the process of creating the proper initial biases: Much of the initial structure can be constructed through the learning of internally generated patterns. Propagating activity waves in the retina allow orientation detectors to form; three-dot patterns in the ponto-geniculate-occipital loop may result in face preference (corresponding to the two eyes and the mouth). Thus, evolution does not need to specify a full visual system, and it does not even need to specify a full starting point for learning: It can instead specify a way of generating internal patterns that establishes useful species-specific biases.

To illustrate the power of this process, pattern-recognition neural networks were constructed in three different ways: purely through learning, purely through evolution, and through a combination of evolved prenatal pattern-generation and learning Valsalam, Bednar, and Miikkulainen 2007. The task consisted of recognizing hand-written digits in the well-known NIST dataset. Each evolved pattern generator encoded a distribution of Gaussians with different positions, rotations, and elongations. Their fitness was based on classification accuracy of the system that was first trained with the generated patterns, and then with the actual patterns in the dataset.

The learning mechanism was simple competitive learning. Each of the 10 neurons had a weight vector w, randomly initialized and then normalized to unit length:

$$w_i = \frac{w_i}{\sqrt{\Sigma_i w_i^2}}. \tag{14.36}$$

Each neuron responded to an input vector x through a weighted sum

$$y_j = \Sigma_i w_i x_i. \tag{14.37}$$

The weight vector of the winning neuron, i.e. the one with the highest response, was then rotated towards the input vector, i.e. first modified with

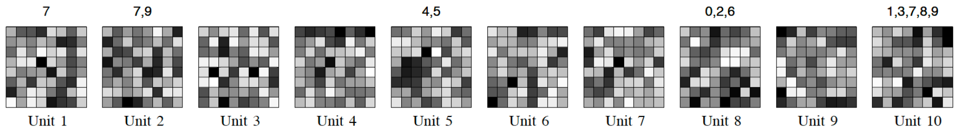
$$w_i(t+1) = w_i(t) + \eta(x_i - w_i(t)), \tag{14.38}$$

and then normalized to unit length. Competitive learning was used because it is a good model of biological (Hebbian) learning, and also because it is relatively weak and therefore depends more on bias.

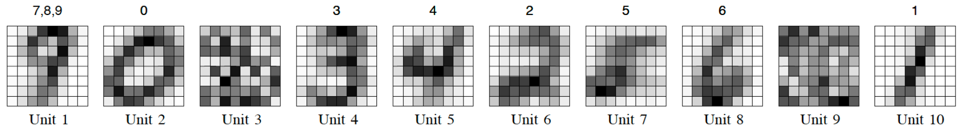
As expected, pure competitive learning developed weight vectors that resembled actual digits (Figure 14.5*b*). However, competitive learning is not very powerful, and usually did not learn to separate all digits. In particular, it had trouble with 7, 8, and 9 because they have many overlapping pixels. Direct evolution, in contrast, has no reason to learn weight vectors that resemble digits. The patterns it developed simply emphasized differences between digit categories, and formed a good foundation for separating them (Figure 14.5*c*). Pattern generation and learning resulted in a most interesting solution that clearly illustrates the importance of having a proper bias. Evolution created pattern generators that emphasized the different horizontal locations around the midline (Figure 14.5*d*). Only a few units learned these patterns, but it was enough to separate 7, 8, and 9 to different units (Figure 14.5*e*). As a result, the postnatal learning with actual examples created a reliable categorization of most examples (Figure 14.5*f*).

Thus, evolution was able to discover a proper bias so that even a simple learning system could perform well on this task. Although it was designed to illustrate a possible biological synergy of evolution and learning, the general approach may be useful in constructing complex systems in general,

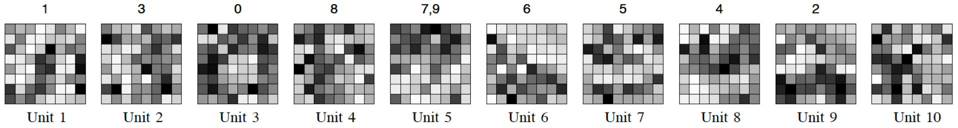
Moreover, the mechanism of internal pattern generation may play a role in the maintenance of such systems throughout the lifetime of the animal (Miikkulainen et al. 2005).



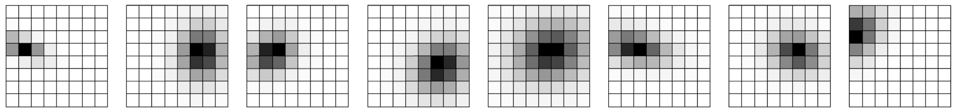
(*a*) Initial random weight vectors



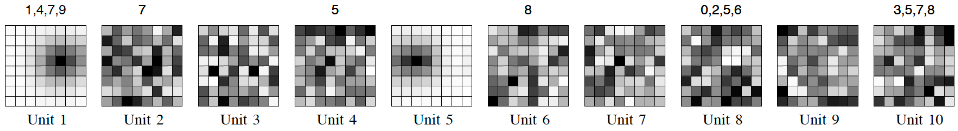
(*b*) Final competitive learning weight vectors



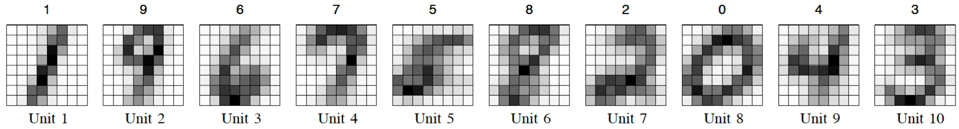
(*c*) Final evolved weight vectors



(*d*) Examples produced by an evolved pattern generator



(*e*) Weight vectors after prenatal training with evolved patterns



(*f*) Final weight vectors after additional competitive learning

Figure 14.5: **Synergy of evolution and learning through evolved pattern generators.** The task was to recognize handwritten digits on a 10×10 simulated retina; the recognition system consisted of 10 neurons that adapted through competitive Hebbian learning. (*a*) The weight vectors of each neuron (Unit) were initialized randomly. (*b*) When they learn through competitive learning, the final weight vectors resemble the inputs. However, learning is not very effective and e.g. 7, 8, and 9 are often confused. (*c*) When the weight vectors are evolved directly, they emphasize the differences that matter for classification. (*d*) The evolved patterns emphasize mostly the locations in the horizontal midline. (*e*) Prenatal training with such patterns takes place only in two units, but it is enough to separate 7, 8, and 9. (*f*) After postnatal learning with actual handwritten digit patterns, most examples are categorized correctly. Evolution thus discovers useful biases, and utilizes the learning mechanism itself to encode them, thus demonstrating synergy of evolution and learning. For animations of these processes, see https://neuroevolutionbook.com/neuroevolution-demos. (Figures from Valsalam, Bednar, and Miikkulainen 2007)

Environmental conditions often change, and the animal needs to adapt to such changes. If such adaptation is based purely on learning, it could easily overfit, and catastrophic forgetting could result. However, if pattern-generator-based learning continues together with learning from the environment, it can serve a stabilizing effect. Adaptation to new inputs is combined with continual adaptation to the fundamental patterns in the domain. Such learning could occur e.g. during REM sleep. This mechanism could potentially explain why animals learn altered environments only partially, and why they spend much time on REM sleep when their neural structures are most plastic. Evolved pattern generators can thus provide a mechanism for continual genetic influences on behavior. It could similarly be instrumental in keeping artificial systems both adaptive and stable.

A further aspect of the synergy between evolution and learning is that evolution can discover the actual learning mechanisms. For instance in the task of discovering repeated patterns in an input sequence with a spiking neural network, evolution discovered plasticity rules that made the task possible in three different settings (Jordan et al. 2021): with reward feedback (reinforcement learning), error feedback (supervised learning), and without feedback (correlation-based unsupervised learning). With cartesian genetic programming as the evolution method (Julian F. Miller 2011), the system discovered symbolic expressions for such plasticity, making it possible to interpret the underlying physical factors, such as homeostasis in the well-known spike-timing-dependent plasticity mechanisms (STDP; Song, Miller, and Abbott 2000).

Many of the metalearning methods reviewed in Chapter 11 and others optimize different aspects of the learning mechanisms (Confavreux et al. 2020; Najarro and Risi 2020a; Tyulmankov, Yang, and Abbott 2022; Gonzalez and Miikkulainen 2021b; Bingham and Miikkulainen 2022; Elsken, Metzen, and Hutter 2019). While often the goal is to simply improve machine learning performance, such methods can also lead to insights into the learning algorithms themselves. For instance, in an experiment where agents needed to adapt to changing reward locations in a Minecraft navigation task, evolution discovered innate reward neurons that made the search for the reward effective even without an explicit reward signal (Ben-Iwhiwhu et al. 2020). Neuroevolution thus discovered structures that facilitated learning during the lifetime of the agent. Such synergies result in more powerful machine learning, but also help us formulate speific hypotheses about biological adaptation.

14.5 Constrained Evolution of Behavior

Section 7.1 illustrated an important principle in evolution of complex behavior: It does not exist in a vacuum, but is constrained and guided by interactions with the environment and with other agents. Simulations of cooperative evolution can thus help us understand the origins of biological behaviors as well. Section 7.1 already demonstrated several such opportunities, including how role-based cooperation may emerge, how adaptive teams can evolve, and how an evolutionary arms race may result in sophisticated herding and hunting behaviors.

This section further expands and generalizes that principle. The guidance may originate not only from complex interactions with the environment, but from general constraints on what the agent can do. For instance, a physical body imposes limits on what movements are possible. Sensory perception is limited, and processing power in decision-making is