

How biological concepts and evolutionary theories are inspiring advances in machine intelligence

Abigail R. Gutai¹ and Thomas E. Gorochoowski^{1,2,*}

¹ School of Biological Sciences, University of Bristol, Life Sciences Building, Tyndall Avenue, Bristol, UK

² BrisSynBio, University of Bristol, Life Sciences Building, Tyndall Avenue, Bristol, UK

* Correspondence should be addressed to T.E.G. (thomas.gorochoowski@bristol.ac.uk)

Abstract

Since its advent in the mid-twentieth century, the field of artificial intelligence (AI) has been heavily influenced by biology. From the structure of the brain to evolution by natural selection, core biological concepts underpin many of the fundamental breakthroughs in modern AI. Here, focusing specifically on artificial neural networks (ANNs) that have become commonplace in machine learning, we show the numerous connections between theories based on coevolution, multi-level selection, modularity and competition and related developments in ANNs. Our aim is to illuminate the valuable but often overlooked inspiration biologists have provided AI research and to spark future contributions at this intersection of biology and computer science. Although recent advances in AI have been swift, many significant challenges remain requiring innovative solutions. Thankfully, biology in all its forms still has a lot to teach us, especially when trying to create truly intelligent machines.

Keywords: artificial intelligence; artificial neural networks; genetic algorithms; evolution; neuroevolution; machine learning;

Introduction

In 1950, Alan Turing posed the question – “Can machines think?”¹ Since then the field of Artificial Intelligence (AI) has strived to find ways to create more intelligent machines². While many theories have been developed over the past seventy years, it has only been during the last few decades that many practical systems have been realised. This recent resurgence stems less from new methodologies but more from the availability of suitably vast data sets that can be used to accurately train AI models and sufficiently powerful computers able to perform this task. Together these capabilities have led to a revolution where machine learning and AI techniques are now being applied far and wide across science and with growing success^{3–10}.

The most ambitious goal in AI is the creation of Artificial General Intelligence (AGI), that is AI which can learn to perform tasks as well as humans ^{11–13}. An evolutionary approach to this problem is a promising direction due to the open-ended nature of evolution ¹⁴. Unlike other methods that rely heavily on human intervention and which result in systems with inherently limited capabilities (e.g. the manual reverse engineering of brain circuits to understand key behaviours ¹⁵), systems that can evolve are able to adapt and complexify themselves as needed to the task at hand. Even though these biologically inspired perspectives hold great value, biologists often do not see the links between their research and AI, or fully appreciate the contributions they could make to this emerging field.

Complex information processing and problems solving occurs at all levels of biological organisation, from signal transduction ¹⁶, non-neural cell communication ¹⁷, neural cognition, and collective brains ¹⁸. In this work, we focus on how biological concepts and evolutionary principles have been applied to Artificial Neural Networks (ANNs) as they are commonly used in AI today. However, it should be noted that these ideas could be applied to all forms of machine learning including those inspired by non-neural biological systems. We demonstrate the inspiration that biology and evolution has provided AI research, some of the key breakthroughs they underpin, and future directions that will require bringing these fields even closer together. We begin by briefly introducing ANNs and methods from evolutionary computing that are central to many recent advances in machine learning. These offer a path towards the creation of open-ended and self-evolving systems that will be likely to be essential for AGI. We then use examples to demonstrate the core biological concepts that support recent advances in the design of ANNs, discuss specific techniques inspired by evolutionary phenomena that can address some of their inherent limitations, and look towards the possibility of using engineered biological substrates for future AI systems. We end by highlighting some of the next steps and the important role that biologists can play in developing the intelligent machines of tomorrow.

Artificial neural networks

Human brains consist of an interconnected network of billions of neurons from which our intelligence emerges (**Figure 1a**). ANNs are a highly simplified model of these real neural networks ¹⁹. They consist of layers of nodes used to represent neurons and directed edges of varying weights to capture the strength of synaptic connections between these (**Figure 1b**). In general, they are structured so that an input layer connects to numerous hidden layers (that are not observed by a user) before an output layer is reached. Each node transmits its current state (i.e. activity level) to neighbours and these transmissions are modulated according to the edge weight between the nodes. Each node then integrates this information (generally

summing all input states) and updates its state according to an activation function. This commonly takes the form of a sigmoidal shaped function that has an output between 0 and 1 capturing whether the node is inactive (state = 0) or firing (state = 1). This state is then transmitted to the next layer by the same process until the output layer is reached.

There are many ways that information can be encoded as input to an ANN and the types of output it can produce. When analysing images, it is common for input nodes to correspond to the intensities of individual pixels and for output nodes to represent various classifications such as 'cat' or 'dog' that are active depending upon the contents of the image. In this scenario, the ability for the ANN to correctly recognise which animal is present in an image is acquired through learning and, like in real brains, involves the adjustment of the edge weights. Correct outputs result in a strengthening of specific connections similar to how intrinsic reward systems modify synaptic weights in animal brains ²⁰. With sufficient training, an ANN can reliably label pictures of dogs and cats it has never seen before.

The precise mechanism of adjusting synaptic strengths in brains is still an open research question but the dominant method in ANNs is through the supervised method of back propagation of errors, often referred to simply as backpropagation or gradient descent ²¹. Backpropagation works by comparing the output of an ANN to its expected value to calculate what is termed the loss function which is fundamentally a measurement of error. The backpropagation algorithm moves backwards from the output layer to the input layer adjusting the weights at each step such that the loss function (error) is reduced. Crucially, this process can be performed efficiently by a computer allowing an ANN's weights to rapidly converge to a configuration that accurately captures the desired input-output relationship.

In addition to the feed-forward multi-layer architectures described above, other forms of ANN are possible. A prominent example is Recurrent Neural Networks (RNNs) which allow for additional connections between layers and individual neurons (e.g. feedback) ²². These more complex architectures provide additional capabilities, such as the ability to use internal states as memory for processing variable length inputs, which has caused RNNs to be extensive use in text and sequence processing tasks. Unfortunately, backpropagation does not always work so well for RNNs as they lack a single feed-forward flow of information through the network ²³. Therefore, alternative approaches to learning are often used, such as evolutionary based methods discussed later.

The development of ANNs caused a lot of initial excitement but it was soon realised that they would have limited success in producing the complexity of human cognition. Brains have been refined over millions of years of evolution to have architectures that form an ideal basis for learning. The product of this evolutionary optimisation is the diverse range of abilities animals are born with such as walking at birth or basic capabilities we take for granted such

as ‘objectness’ (i.e. being able to distinguish objects from their environment), whereas conventional ANNs rely on learning alone, resulting in a number of deficiencies ²⁴.

ANNs are also susceptible to stagnation at local optima in their fitness landscape. For example, each ANN has an associated fitness landscape for the task they are confronted with (**Figure 1c**). Each dimension corresponds to a different parameter which can be adjusted (e.g. edge weight), resulting in changes in fitness (e.g. improved prediction accuracy). Often, the landscape has many dimensions and a complex topology with many peaks and troughs, making an exhaustive search impossible or impractical. Training methods often involve incremental movement toward high fitness levels allowing them to converge to local optima but hindering their ability to discover global optima that would require exploring initially counterintuitive (i.e. low fitness) regions known as deceptive domains. For example, a biped controlled by an ANN where fitness is the total distance travelled will initially fall in the direction of the finish; balance and coordination appear as a deceptive domain. This particular deceptive domain can be avoided by using an easing period ²⁵. However, this requires reliance on human intervention and detection of the deceptive domain, which is not always as intuitive as in this locomotive example. Recently, it has been shown that local minima may not pose a problem when the ANN is over parameterised (i.e., is very deep with many intermediate layers) ^{26–28}, demonstrating that the core limitations in some learning models are still not fully understood.

Genetic algorithms

Various computer-based evolutionary techniques were developed independently in the 1960s, which today fall under the umbrella of evolutionary computation ²⁹. Genetic Algorithms (GAs) ³⁰ are one of the most commonly used and work by encoding a potential solution to a problem as a bit string (e.g. 01010100011...) where each bit corresponds to the presence/absence (1/0) of a particular characteristic (e.g. a link between two nodes in an ANN). A population of solutions (e.g. strings) is generated and the highest performing are chosen to mate, intermediate performing variants continue to the next generation and low performing strings perish before the next cycle. Mating works similarly to crossing over in meiosis (**Figure 1**). In the prophase of meiosis, maternal and paternal chromosomes align to form chiasmata at random homologous loci. The sequences of DNA between these chiasmata are exchanged. In GAs, a random point along the string is selected and bits are swapped to create two new offspring. Mutations occur rarely (e.g. for 1 in every 10,000 bits for classical GAs) switching a 1 to a 0 or *vice versa*. Closely adjacent bits are less likely to be separated by crossing over (similar to genetic linkage), so tend to be synergistic and form partial solutions similar to genes in biology. Recombination of good partial solutions (i.e. sub-strings) enables the search to

‘jump’ between optima focusing in the most promising parts of the search space, while purging poor partial solutions which would otherwise accumulate³¹. Because mutations are stochastic rather than guided by a fitness gradient, and populations of solutions need to be tracked over time, GAs are computationally expensive to run. However, they are able to avoid local optima making them particularly well-suited to solving problems with complex fitness landscapes.

It should be noted, that while bit strings were originally the basis of most GAs, many modern systems that now fall under the more general umbrella of Evolutionary Algorithms (EAs)³² make use of more complex data structures to represent a possible solution (i.e. specific ANN) and provide tailored functions that allow for mutations and the combination (i.e. crossover) of multiple solutions.

Neuroevolution

Neuroevolution is the artificial evolution of ANNs where edge weights are determined by EAs instead of backpropagation. Specifically, an EA is used to evolve a population of bit strings which represent the connection weight matrix of an ANN. The fitness of each bit string is determined by the performance of their resulting ANNs at a given task. The field emerged in the 1990s³³ and EAs are now argued to be a competitive alternative to backpropagation³⁴.

Neuroevolution became popular due to the development of the NeuroEvolution of Augmented Topologies (NEAT) system³⁵. In NEAT, ANNs evolve from a minimal architecture and, in addition to adjusting their weights, mutations can affect the topology of the network by adding and removing nodes and connections. This allows optimisation to occur simultaneously with complexification, mirroring how our brains have evolved from primitive smaller ones or how a mature brain develops from the limited capacity of an infant brain. Complexification enables the elaboration of existing optimised features so enters larger search spaces in more promising domains than would otherwise exist with a large, fixed topology. Structural elaboration is often initially detrimental to ANNs and takes multiple generations to optimise. So, to prevent new innovations being lost due to this initial loss in fitness, competition is limited to structures which emerge at the similar stages via a process called ‘speciation’. This enables the preservation of innovations with future potential, further enabling exploration of deceptive domains. NEAT significantly outperforms neuroevolved ANNs with a fixed topology in pole balancing³⁵, a standard benchmark learning task. In addition, the evolution of topology eliminates reliance on human design prior to evolution which requires insight that may not always exist.

Just as pure learning has its deficiencies, so does pure evolution. As discussed previously, using hill climbing techniques such as backpropagation to adjust ANN weights enables a local optimum to be efficiently reached but the global optimum may never be

discovered. In contrast, pure evolution is more likely to locate the region of a global optimum but won't necessarily locate its peak because it does not directly follow a local fitness gradient³⁶. As such, these approaches are complementary, just as in nature. Evolution shapes the fundamental innate behaviours of an organism based on the experiences of ancestors, yet the precise conditions and events experienced by each individual in a lineage will differ subtly. Learning operates on the innate framework of behaviour to optimise it specifically to the individual's unique life experience. In other words: evolution discovers the fittest innate behaviour, and learning optimises it so it can be best used. Coupling evolutionary computing with pure learning, such as via neuroevolution with backpropagation, strengthens this analogy with nature and exploits the best qualities of each approach. This may be done using neuroevolution to evolve an ANN structure and training with backpropagation to optimise the edge weights, or perhaps by alternating between the two or a combination as in nature³⁷.

Evo devo and indirect encoding

Modern ANNs can contain billions of connections. If using neuroevolution, each ANN would need a string of at least an equal length and as a large population of these ANNs is also required, the space to store each iteration of the evolutionary process would become prohibitive. In contrast, the human genome sequence holds approximately 1.5 GB of information, which is not enough to directly encode each of the trillions of synaptic weights in the brain, let alone the entire organism. Despite this, it is possible for the genome to encode the phenome because it does so *indirectly*. The full connective architecture of the brain is not mapped explicitly in the genome, instead cellular interactions lead to a cascade of developmental processes which results in a structural phenotype emerging with desired high-level statistical features. Genes are reused and affect multiple phenotypic traits creating regular patterns and symmetries with repeating motifs³⁸. For example, most multicellular organisms display some form of symmetry and numerous features which are derived from a single innovation, e.g., multiple legs and hair follicles³⁹. If the form and neural architecture for each leg were encoded by a different gene, four independent mutations would be required for any adaptation. This would likely introduce a deceptive domain because a single mutation may result in an inefficient stance and uncoordinated behaviour and walking gait. If all legs are encoded by one gene, then the transition becomes trivial because only one mutation is required⁴⁰. This property is essential for coordinated evolution of structure and behaviour and is also the basis of many important exaptations – features that acquire functions for which they were not originally adapted or selected. For example, the legs, wings, antennae and mandible of an insect are all homologous variations of the same appendage with distinct specialised functions³⁹.

Stanley and Ambrosio incorporated this property of indirect encoding for ANN parameters which they called Hypercube-based NEAT (HyperNEAT) ⁴⁰. Unlike earlier where NEAT is used to evolve the ANN directly, in HyperNEAT it is used to evolve compositional pattern producing networks that generate highly regular connection topologies ⁴¹. These patterns are then used to encode the parameters for the functional ANN (**Figure 3b**). The result is an ANN that can efficiently exploit regularities in a problem. For example, HyperNEAT can automate the evolution of highly coordinated four-legged gaits which requires genetic reuse for each leg ⁴². Previous work has demonstrated that indirect encoding can be used to produce coordinated movement ²⁵, whereas when limited to direct encoding this is only possible by simplifying the task by manual decomposition ^{43,44}. Additionally, as HyperNEAT represents the problem geometrically, solutions are scalable without further evolution. Therefore, high dimensional tasks can be tackled using a low resolution/complexity system making them computationally cheaper to train.

Complexification through development, competition and evolution

A few billion years ago, living cells began simply – morphically, behaviourally and genetically in a simpler environment and with less competition. From then on, the elaboration of genomes enabled new behaviours and structures. This adaptive radiation of life has created multiple kinds of organisms with different metabolic processes occupying different niches. As these organisms interact with their environment, they cause changes opening new niches for further diversity to arise. This leads to a continually growing web of interactions with multiple trophic levels and competitive coevolution (**Figure 2**).

Conventional evolutionary computing lacks this open-ended quality, usually halting as it stagnates at a final solution. This is due to many evolutionary computing techniques involving a single static selection pressure based on the performance of the task of interest. However, many of the complex behaviours we are interested in recreating in AI such as language are the product of gradual complexification, rather than an explicit selection pressure for language in isolation. Such processes are thought to be a key ingredient for autonomous creation of complex behaviours ⁴⁵.

A relatively new technique in evolutionary computing called novelty search has shed light on the shortcomings of conventional approaches. In novelty search, behaviours are selected using a pre-designed novelty metric rather than an explicit selection for performance alone. Once the search for the simplest behaviours is exhausted, more complex behaviours are generated and searched, enabling a diversification and complexification of the solutions found. In a biped locomotive task, novelty search significantly outperforms fitness-based approaches despite being agnostic about what behaviour is adaptive for walking ⁴⁶. This study

highlights the deficiency of using a single fixed selection pressure based on task performance – by completely ignoring the objective it's possible to explore deceptive domains associated with the task fitness. However most behaviours are useless for the researchers goals, so to focus efforts, 'Quality Diversity' combines selection for both novelty and objective, resulting in only the fittest representatives for a diversity on behaviours creating competition within niches to breed the fittest solutions from all possible strategies ^{47–49}. Novelty search bares some similarity to natural complexification, with 'novelty' representing newly available niches, giving some support to the importance of gradual complexification in the discovery of complex behaviours. However, it is a vast simplification of open-ended evolution. Despite the apparent drive for complexity, there is no evidence of an explicit selection for it. Rather it is a by-product of a variety of evolutionary phenomena such as arms-races, gene duplication events ⁵⁰ and major evolutionary transitions ⁵¹. Evolutionary computing techniques which incorporate natural complexification have been successful in producing more complex behaviours. By starting small, task performance can be optimised in a simple fitness landscape, and then by gradually complexifying the AI, the solution enters larger search spaces in more promising domains than it would otherwise have starting there.

A major drive for complexification is the interactions between coevolving organisms, particularly antagonistic ones. Predator and prey, for instance, perpetually create new selection pressures for their counterparts. A famous example of this is the arms-race between bats and eared moths ⁵². The ability of bats to locate moths using sonar clicks drove the evolution of sonar detection in moths. This was followed by bats altering the intensity and frequency of their sonar click for stealth. Moths then evolved their own sonar clicks to 'jam' the frequency of the bats or act as acoustic aposematism. Such arms-races are never-ending, a phenomenon known as the Red Queen Principle ⁵³, and result in individuals constantly adapting to keep up with new innovations. Hillis was the first to simulate an arms-race for evolutionary computing ⁵⁴. He used a host/parasite interaction to evolve more efficient sorting networks. The function of a sorting network is to arrange numerical data in ascending or descending order. In the experiment, the fitness of the sorting network was based on its sorting success (i.e. number of elements in the correct order). The numerical data represented the parasite which coevolved with the sorting network and its fitness was based on sorting failures. As the parasite evolved, the sorting problems it produced gradually increased in difficulty. This gradual increase in task difficulty enabled more efficient learning and discovery of better sorting strategies. Increasing the difficulty of a task over time has also proven a useful approach in training ANNs, helping to avoid deceptive domains in a technique known as 'incremental evolution' ⁵⁵. Often it involves manually designing progressively harder tasks for the ANN, which is laborious. However, the same process occurs automatically in simulated arms-races due to the simultaneous evolution of opponents.

Multiple selection pressures enable exaptation, where features which were selected for under one pressure can be adapted for by another⁵⁶. For example, feathers initially evolved for warmth, then for gliding after jumping from a height, and eventually for flight⁵⁷. Feathered flight would likely not have evolved without the selection for warmth because primitive feathers would have been costly and ineffective for flying, and thus, a deceptive domain. Mouret and Doncieux demonstrated that exaptation can occur in AI systems by implementing multiple selection pressures⁵⁸. In their experiments, artefacts of increased complexity arose when exposed to selection pressures besides that for task performance, again highlighting the deficiency of using only a single selection pressure to guide evolution.

The evolution of the brain, from smaller primitive brains to complex ones, inspired the development of ANN topological complexification in NEAT. The problem is first approached with a simple ANN and gradually the number of nodes and connections are increased as the solution is elaborated. This also reflects the evolution of innate behaviours where complex behaviours are built upon simpler ones⁵⁶.

A factor often overlooked in evolutionary computing which limits complexification is the evolution of morphology, i.e., there are a limited number of behaviours possible with a single fixed morphology. This is highly relevant in robotics. Behavioural adaptations often depend on physical adaptations. For example, the adaptation of the bat locating its prey using echolocation first relied on the vocal physiological adaption in addition to the sensory perception of its own reflected sonar. Simultaneous evolution of morphology and behaviour²⁵ opens up a vast possibility of phenotypes to be explored compared to using a single fixed morphology⁵⁹. As mentioned earlier, one way to capture this feature in ANNs is by employing CPPNs which enable the structure of an ANN to dynamically evolve and explore potentially useful topologies for a task at hand⁶⁰.

Complexification occurs not only gradually with evolution but also due to learning within the lifetime of an individual. It is well understood that humans and other animals learn most efficiently if they begin with simpler tasks and gradually expand their knowledge and skills⁶¹. For example, musical students begin practice at grade one and slowly improve their skills before attempting a grade eight piece. This is so effective in producing complex cognition that it is effectively 'built in' to cognitive development: the initial cognitive limitations and slow development of some animals, particularly humans, is thought to be adaptive to facilitate incremental learning during early life, rather than just being an artefact of incomplete development⁶². For example, infants are only able to focus on objects ~10 cm away, allowing them to learn from simpler sensory information without the complication of size inconsistency and distance. This form of gradual learning can be applied to AI too. For example, two pure-learning ANNs which play Go against one another both begin at as novices and gradually become more skilled, learning better strategies faster than if they were trained against master

opponents to begin with. The major benefit of self-play is that it eliminates the need to obtain human game data across a range of skill levels or the need to spend time playing against humans in real-time, which is slow and limited in scope.

It is likely that a combination of multiple kinds of complexification is necessary for open-ended evolution⁶³. For example, implementing just competitive coevolution alone can cause stagnation due the loss of the Red Queen gradient, where one species discovers a strategy so strong that it cannot be beat by an opponent. Whereas, when paired with topological complexification, the space of possible behaviours continually broadens, helping to establish effective open-ended arm-races⁶⁴.

Complexification is particularly effective using HyperNEAT because solutions are scalable. Gauci and Stanley found that HyperNEAT-evolved ANNs that first learn to play the board game Go on a 5×5 board could subsequently learn faster and discover better strategies when then faced with a larger 7×7 board⁶⁵. HyperNEAT also shows promise for exaptation because it enables the duplication of clusters of nodes, similar to gene duplication events⁵⁰, and therefore previous behaviours can continue to evolve simultaneously with the new exaptation creating multimodal functionalities. Huizinga and Clune evolved multimodal ANNs with HyperNEAT by using multiple selection pressures⁶⁶. It is unclear if similar success can be achieved with coevolution because it is difficult to control or identify the selection pressures that emerge⁶⁷. So far, coevolution using HyperNEAT has not been attempted but is a promising direction to explore. Sims used a simpler developmental approach to simultaneously coevolve the morphology and behaviour of 3D creatures in a head-to-head challenge to catch a ball²⁵. This resulted in the successful evolution of a diversity of interesting strategies but there was no control to test whether indirect encoding was an important component in the results.

Modularity, hierarchy and division of labour

From genetic to societal levels, selection across scales is thought to have played a role in major evolutionary transitions (**Figure 3**). For example, the eukaryotic cell arose from a proto-eukaryote's engulfment of and subsequent mutualism with a prokaryote which became the mitochondria. The reproduction of the mitochondria eventually depended on the reproduction of the eukaryotic cell and *vice versa*⁶⁸. Due to their aligned reproductive fates, there is a selection for cooperation and this manifests in a division of labour – the mitochondria provide optimal conditions for aerobic respiration and the host capitalises on this while providing the energy and resources required. This example of cellular compartmentalisation enables much greater efficiency within the cell and facilitates further beneficial variation⁶⁹. Another major benefit of compartmentalisation is its ability to allow otherwise incompatible processes to

coexist within a collective system. Such separation is particularly important for the filamentous cyanobacteria *Anabaena*, which under nitrogen limiting conditions allow for the differentiation of vegetative cells into heterocytes along their filament chains⁷⁰. Heterocytes are cells built to exclude oxygen containing thick walls, mechanisms to remove oxygen, and having photosynthesis deactivated. This is crucial as the nitrogenase enzyme expressed in heterocytes that fixes nitrogen can only function in an anaerobic environment. Therefore, by compartmentalising this biochemical function into a different cell type, filaments of *Anabaena* can survive in very nutrient low environments, vegetative cells fixing carbon and heterocytes fixing nitrogen simultaneously from the air, even though the two processes are biochemically incompatible. The ability to separate such functional processes into compossible 'modules' is also believed to play a crucial role in the evolvability of natural systems^{71–73}.

The equivalent of such a division of labour in computing is the functional modularity that computer software and hardware often exhibit. From an engineering perspective, modularity is useful because components can be replaced, upgraded, or reused more easily without having to redesign or construct an entire system from scratch. In AI, modularity is also important because it can simplify a problem by decomposing a complex task into smaller ones that are easier to solve. Many types of AI exhibit levels of modularity which are analogous to biological ones, such as in GAs, ANNs and multi-agent systems.

As the lowest levels of biology, genetic information is encoded in linear DNA sequences composed of four bases. Although having a simple linear structure, sub-sequences encode higher-level functional elements such as genes, and these are further grouped together into chromosomes that ultimately make up the final genome of an organism. Each component in this hierarchy has a distinct role in synthesis and the proliferation of each component depends on the success of the complete phenotype⁷⁴. A similar phenomenon also occurs during the evolution of GAs where bit strings are used to encode the genome of a potential solution. Crossing over operations allow for partial solutions to become localised within a string so that they can propagate effectively. These sub-sequences can then be further combined at larger length scales establishing a hierarchy of modularity even within this simple system. By viewing a complex task as a collection of smaller interacting subtasks, it is possible to tackle larger problems in smaller chunks, facilitating search in even highly deceptive spaces. Because of this, genetic algorithms have been found to better tackle tasks with complex fitness landscapes compared to ANN-based deep learning approaches³⁴.

At the neural level, brains are also modular being composed of many distinct regions dedicated to different cognitive processes. For example, the Broca's area is involved in articulation of language. Upon injury of the Broca's area, a patient may significantly struggle to speak their native language but have no problem understanding it because language comprehension is located separately in Wernicke's area. ANNs tend to be non-modular, or

‘monolithic’, so changes made to one part may have unpredictable knock-on effects elsewhere due to their highly integrated structure. Thus, the learning of a new skills can cause catastrophic forgetting of an ANN’s previous function. In contrast, in modular biological brains, the acquisition of a new skill such as piano playing does not tend to alter existing skills such as mental arithmetic. Modular ANNs can be evolved by including selection based on a human-designed metric of modularity⁵⁸ or by including connection costs as a selective pressure⁷⁵. The latter was based on the suggestion that modularity in the brain evolved initially to reduce the amount of synaptic connections to save energy⁷⁶. Modular ANNs outperform monolithic ANNs and they are more evolvable to new tasks because they are able to reuse modules corresponding to regularities in the task. They also suffer less from catastrophic forgetting⁷⁷.

At higher levels of organisation, social groupings are common between organisms (e.g., parental care, herds and flocks). There are many benefits to group living. An extreme example of this are the eusocial insects whose colonies form what are termed a ‘superorganism’. These colonies are composed of distinct castes – such as queens, workers, and soldiers – which are morphologically and behaviourally specialised to their task. In AI, a problem can be tackled using multiple cooperating agents if fitness can be measured by the sum of their behaviours. For example, OpenAI created a human-relevant example of how this works using a game of hide and seek⁷⁸. Pairs of seekers played against pairs of hiders where each individual is controlled by its own ANN. The hiders worked together to build a shared wall around themselves before the seekers finished counting. Once the seekers discovered they could use ramps to breach the wall, the hiders exhibited a division of labour where one began constructing the wall while the other confiscated the ramps – a strategy that would have been impossible by either alone due to the short countdown.

Despite the three described levels of modularity above, most of the present AI literature involves implementing only a single level within a system, which may not always enable sufficient decomposition of a task at hand. The performance of GAs, for instance, is known to deteriorate as the dimensionality of the search space increases⁷⁹. There are some cases that demonstrate combining GAs and multi-agent modularity increases performance: Potter and De Jong coevolved multiple cooperating bit strings⁷⁹, and Miikkulainen *et al.* used neuroevolution to coevolve predators for a prey-capture task which caught more prey together than separately⁸⁰. However, the hierarchy of modularity in current artificial systems is modest compared to that found in biology. In nature, group selection occurs on numerous levels simultaneously, establishing an extensive hierarchy of task decomposition which scales with complexity covering genomes⁸¹, endosymbiosis, cell compartmentalisation⁸², multicellularity⁵¹, tissues and organs, organisms, sex⁸³, social groups⁷⁴, and inter-specific mutualisms⁸⁴ to name but a few. In effect, a complete ecosystem also exhibits modularity through niche differentiation, which operates to reduce competition^{85,86}.

Another important feature of group living is the exchange of information between members. An example of this is the hive-mind of social insects. When looking for a new nest, colony members split up and independently search the environment. Upon their return they communicate the quality of the locations they have found until a consensus is reached on where the whole colony should move next. Insect communication like this has inspired a method of exploring complex fitness landscapes known as Particle Swarm Optimisation (PSO) where a 'swarm' of potential solutions move in response to both local features that an individual solution experiences as well as features gathered and communicated by other solutions in the swarm ⁸⁷. This type of approach has been used in place of backpropagation to train ANNs and evolve their structure with some limited success ⁸⁸. Communication has also inspired information sharing in ANN-based systems. For example, in their prey-capture simulations, Miikkulainen *et al.* found that a cooperating group of NEAT-evolved predators caught more prey if information was shared among them ⁸⁰.

Engineering biologically based AI systems

Beyond inspiration, future developments in bioengineering may allow for a radical rethink as to AI's connection to biology, allowing it to become more closely integrated within the computer science domain. Rather than running simulations of artificial neural networks on electronic computers, which are often ill-suited to the task, why not grow living neural networks into desired interconnection patterns and exploit the innate capabilities of these cells for learning and computation? Some progress has been made on this front ⁸⁹ and while we are still far from understanding the molecular complexity of living neural networks, this does not prevent us using some of their capabilities in new ways. The field of synthetic biology is a great example of this, where simple biologically based systems are built using biological parts (not always fully understood) as a means to repurpose them in new ways ⁹⁰. This has led to many breakthroughs and the creation of modified living systems able to convert waste into valuable chemicals and materials ^{91,92}, act as new forms of therapeutic to treat disease ⁹³, and even perform novel distributed computations ^{94–96}.

There have already been several attempts at creating systems that can learn by implementing AI models using simple biomolecular parts. Examples include the creation of a four-input perceptron based on metabolic circuits as a basis for more advanced sensing applications ⁹⁷ and the use of DNA strand displacement to implement basic artificial neural networks, including a Hopfield associative memory consisting of four fully connected artificial neurons ⁹⁸.

In addition to using existing models of AI as a basis for developing new biologically based system, the biological substrates themselves might also offer novel information

encoding and processing capabilities that go beyond that which is easily achievable using electronic computers ^{94,99–101}. This would provide a chance to potentially rethink how AI is implemented and spark creative approaches that harness the unique qualities of biology. There has even been some speculation as to the ‘cellular supremacy’ of biological systems in terms of their potential computational power compared to their classical electronic counterparts ⁹⁹. For example, key features in biology such as the use of continuous and noisy signals, multiple levels of organisation/hierarchy, and the innate ability of biology to adapt and evolve may be key ingredients for implementing intelligence, but which are difficult or potentially impossible to implement efficiently using classical computers.

Using biology as a substrate for AI may also offer new ways to tackle other challenges. Biological systems are nearly always highly robust to the failure of individual parts, are energy efficient, and are self-replicating and repairing. Existing electronic AI systems can be difficult to deploy because of the computational power and energy requirements they are generally fragile to hardware failure, placing concerns over their safe deployment. Using engineered biology to implement AI systems could open avenues for systems that can be deployed in virtually any environment being able to exploit local resources to generate energy and including processes to continually monitor and repair faulty biomolecular components implementing the AI system. They even could be designed to exploit evolution itself ¹⁰² and create truly autonomous and self-adaptive systems.

Conclusion

The prospect of building AGI depends upon our ability to create systems which are able to work in an autonomous, innovative and open-ended way – much like biology. Here, we have shown how some of the core characteristics, concepts and theories underpinning biological systems have greatly influenced the design and evolution of ANNs to date (**Figures 2 and 3**). But this is only a fraction of what might be possible. Biological systems evolve, function, and learn across many levels of organisation simultaneously and in concert with their changing environment. Seeing an AI system not as a single isolated entity, but instead as a computational ecosystem of specialised modules working across scales in a dynamic and multi-faceted environment, could be an important step towards realising some of the additional capabilities we crave.

While our focus here has been on ANNs as a basis for intelligent machines, the ability for single cells and vast collectives to display intelligent behaviours mean that future inspiration for AI could come from our emerging understanding of computation in developmental biology ¹⁰³, swarm cognition ¹⁸, as well as information processing in chemical reaction networks ¹⁰⁰ and evolving populations ¹⁰⁴. In fact, the more we investigate biological systems, the more we

realise the diverse and unintuitive ways information is encoded and processed by biology that goes beyond the capabilities of modern electronic computers. Perhaps concepts such as ‘liquid brains’¹⁰⁵ and knowledge regarding distributed biological cognition need to find their way into computer science for progress to accelerate?

Discovering the ingredients needed to create a true AGI is one of the major challenges facing science today¹¹. Even though AI research is often perceived as computer science, biology has provided important inspiration for many modern approaches and may form an important substrate for building the possibly “living” AI systems of the future.

Acknowledgements

We would like to thank Jon Bridle, Claire Grierson and Simeon Castle for their insightful comments on a draft of this work and anonymous reviewers for their valuable feedback and suggestions. T.E.G. was supported by BrisSynBio, a BBSRC/EPSRC Synthetic Biology Research Centre (grant BB/L01386X/1), and a Royal Society University Research Fellowship (grant UF160357). This study did not involve any underlying data.

Author Contributions

All authors contributed to the writing and editing of the work. T.E.G. supervised the work.

Declaration of Interest

None.

References

1. Turing, A. M. I. Computing Machinery and Intelligence. *Mind* **49**, 433–460 (1950).
2. Cave, S. & Dihal, K. Hopes and fears for intelligent machines in fiction and reality. *Nat. Mach. Intell.* **1**, 74–78 (2019).
3. Li, B., Hou, B., Yu, W., Lu, X. & Yang, C. Applications of artificial intelligence in intelligent manufacturing: a review. *Front. Inf. Technol. Electron. Eng.* **18**, 86–96 (2017).
4. Stafford, I. S. *et al.* A systematic review of the applications of artificial intelligence and machine learning in autoimmune diseases. *Npj Digit. Med.* **3**, 30 (2020).
5. de Almeida, A. F., Moreira, R. & Rodrigues, T. Synthetic organic chemistry driven by artificial intelligence. *Nat. Rev. Chem.* **3**, 589–604 (2019).
6. Goldenberg, S. L., Nir, G. & Salcudean, S. E. A new era: artificial intelligence and machine learning in prostate cancer. *Nat. Rev. Urol.* **16**, 391–403 (2019).
7. Schneider, P. *et al.* Rethinking drug design in the artificial intelligence era. *Nat. Rev. Drug Discov.* **19**, 353–364 (2020).
8. A Dixon, T., C Curach, N. & Pretorius, I. S. Bio-informational futures. *EMBO Rep.* **21**, e50036 (2020).
9. Richards, B. A. *et al.* A deep learning framework for neuroscience. *Nat. Neurosci.* **22**, 1761–1770 (2019).
10. Moen, E. *et al.* Deep learning for cellular image analysis. *Nat. Methods* **16**, 1233–1246 (2019).
11. Ben Goertzel. Artificial General Intelligence: Concept, State of the Art, and Future Prospects. *J. Artif. Gen. Intell.* **5**, 1–48 (2014).
12. Sam Adams *et al.* Mapping the Landscape of Human-Level Artificial General Intelligence. *AI Mag.* **33**, (2012).
13. Goertzel, B. & Pennachin, C. *Artificial General Intelligence (Cognitive Technologies)*. (Springer-Verlag, 2007).
14. Ruiz-Mirazo, K., Umerez, J. & Moreno, A. Enabling conditions for 'open-ended evolution'. *Biol. Philos.* **23**, 67–85 (2008).
15. Mnih, V., Heess, N., Graves, A. & Kavukcuoglu, K. Recurrent models of visual attention. in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2* 2204–2212 (MIT Press, 2014).
16. Lyon, P. The cognitive cell: bacterial behavior reconsidered. *Front. Microbiol.* **6**, 264 (2015).
17. Baluška, F. & Levin, M. On Having No Head: Cognition throughout Biological Systems. *Front. Psychol.* **7**, 902 (2016).

18. Couzin, I. Collective minds. *Nature* **445**, 715–715 (2007).
19. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
20. Schultz, W. Predictive Reward Signal of Dopamine Neurons. *J. Neurophysiol.* **80**, 1–27 (1998).
21. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning representations by back-propagating errors. *Nature* **323**, 533–536 (1986).
22. Yu, X. Si, C. Hu, & J. Zhang. A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Comput.* **31**, 1235–1270 (2019).
23. Pascanu, R., Mikolov, T. & Bengio, Y. On the Difficulty of Training Recurrent Neural Networks. in *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28* III-1310-III–1318 (JMLR.org, 2013).
24. Zador, A. M. A critique of pure learning and what artificial neural networks can learn from animal brains. *Nat. Commun.* **10**, 3770 (2019).
25. K. Sims. Evolving 3D Morphology and Behavior by Competition. *Artif. Life* **1**, 353–372 (1994).
26. Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B. & LeCun, Y. The loss surfaces of multilayer networks. *J. Mach. Learn. Res.* **38**, 192–204 (2015).
27. Dauphin, Y. N. *et al.* Identifying and Attacking the Saddle Point Problem in High-Dimensional Non-Convex Optimization. in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2* 2933–2941 (MIT Press, 2014).
28. Lu, H. & Kawaguchi, K. *Depth Creates No Bad Local Minima.* (2017).
29. Fogel, D. B. *Evolutionary Computation: The Fossil Record.* (Wiley-IEEE Press, 1998).
30. Holland, J. H. Genetic Algorithms. *Sci. Am.* **267**, 66–73 (1992).
31. Muller, H. J. The relation of recombination to mutational advance. *Mutat. Res. Mol. Mech. Mutagen.* **1**, 2–9 (1964).
32. Reeves, C. Evolutionary computation: a unified approach. *Genet. Program. Evolvable Mach.* **8**, 293–295 (2007).
33. de Garis, H. Genetic Programming: Building Artificial Nervous Systems Using Genetically Programmed Neural Network Modules. in *Machine Learning Proceedings 1990* (eds. Porter, B. & Mooney, R.) 132–139 (Morgan Kaufmann, 1990). doi:10.1016/B978-1-55860-141-3.50019-5.
34. Such, F. P. *et al.* *Deep Neuroevolution: Genetic Algorithms Are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning.* (2017).
35. Stanley, K. O. & Miikkulainen, R. Evolving Neural Networks through Augmenting Topologies. *Evol. Comput.* **10**, 99–127 (2002).

36. E. Galvan & P. Mooney. Neuroevolution in Deep Neural Networks: Current Trends and Future Challenges. *IEEE Trans. Artif. Intell.* **1**, 1–1 (5555).
37. Liu, H., Simonyan, K. & Yang, Y. DARTS: Differentiable Architecture Search. *ArXiv abs/1806.09055*, (2019).
38. Stiles, J. & Jernigan, T. L. The Basics of Brain Development. *Neuropsychol. Rev.* **20**, 327–348 (2010).
39. Angelini, D. R. & Kaufman, T. C. Insect appendages and comparative ontogenetics. *Dev. Biol.* **286**, 57–77 (2005).
40. K. O. Stanley, D. B. D'Ambrosio, & J. Gauci. A Hypercube-Based Encoding for Evolving Large-Scale Neural Networks. *Artif. Life* **15**, 185–212 (2009).
41. Stanley, K. O. Compositional pattern producing networks: A novel abstraction of development. *Genet. Program. Evolvable Mach.* **8**, 131–162 (2007).
42. J. Clune, B. E. Beckmann, C. Ofria, & R. T. Pennock. Evolving coordinated quadruped gaits with the HyperNEAT generative encoding. in *2009 IEEE Congress on Evolutionary Computation* 2764–2771 (2009). doi:10.1109/CEC.2009.4983289.
43. G. S. Hornby, S. Takamura, T. Yamamoto, & M. Fujita. Autonomous evolution of dynamic gaits with two quadruped robots. *IEEE Trans. Robot.* **21**, 402–410 (2005).
44. Téllez, R. A., Angulo, C. & Pardo, D. E. Evolving the Walking Behaviour of a 12 DOF Quadruped Using a Distributed Neural Architecture. in *Biologically Inspired Approaches to Advanced Information Technology* (eds. Ijspeert, A. J., Masuzawa, T. & Kusumoto, S.) 5–19 (Springer Berlin Heidelberg, 2006).
45. Soros, L. & Stanley, K. Identifying Necessary Conditions for Open-Ended Evolution through the Artificial Life World of Chromaria. *Artif. Life Conf. Proc.* **26**, 793–800 (2014).
46. Lehman, J. & Stanley, K. O. Abandoning objectives: Evolution through the search for novelty alone. *Evol. Comput.* vol. 19 189–223 (2011).
47. Pugh, J. K., Soros, L. B. & Stanley, K. O. Quality Diversity: A New Frontier for Evolutionary Computation. *Front. Robot. AI* **3**, 40 (2016).
48. Mouret, J.-B. Evolving the Behavior of Machines: From Micro to Macroevolution. *iScience* **23**, 101731 (2020).
49. Stanley, K. O., Clune, J., Lehman, J. & Miikkulainen, R. Designing neural networks through neuroevolution. *Nat. Mach. Intell.* **1**, 24–35 (2019).
50. Martin, A. P. Increasing Genomic Complexity by Gene Duplication and the Origin of Vertebrates. *Am. Nat.* **154**, 111–128 (1999).
51. Szathmáry, E. & Smith, J. M. The major evolutionary transitions. *Nature* **374**, 227–232 (1995).

52. Conner, W. E. & Corcoran, A. J. Sound Strategies: The 65-Million-Year-Old Battle Between Bats and Insects. *Annu. Rev. Entomol.* **57**, 21–39 (2011).
53. Van Valen, L. A new evolutionary law. *Evol. Theory* **1**, 1–30 (1973).
54. Hillis, W. D. Co-evolving parasites improve simulated evolution as an optimization procedure. *Phys. Nonlinear Phenom.* **42**, 228–234 (1990).
55. Gomez, F. & Miikkulainen, R. Incremental Evolution of Complex General Behavior. *Adapt. Behav.* **5**, 317–342 (1997).
56. Lenski, R. E., Ofria, C., Pennock, R. T. & Adami, C. The evolutionary origin of complex features. *Nature* **423**, 139–144 (2003).
57. Gould, S. J. & Vrba, E. S. Exaptation-A Missing Term in the Science of Form. *Paleobiology* **8**, 4–15 (1982).
58. J. Mouret & S. Doncieux. Evolving modular neural-networks through exaptation. in *2009 IEEE Congress on Evolutionary Computation* 1570–1577 (2009). doi:10.1109/CEC.2009.4983129.
59. Auerbach, J. E. & Bongard, J. C. On the relationship between environmental and morphological complexity in evolved robots. *Proceedings of the 14th annual conference on Genetic and evolutionary computation* 521–528 (2012).
60. Cheney, N., Maccurdy, R., Clune, J. & Lipson, H. Unshackling evolution: Evolving soft robots with multiple materials and a powerful generative encoding. in *In Proceedings of the Genetic and Evolutionary Computation Conference. ACM* (2013).
61. Peterson, G. B. A Day of Great Illumination: B. F. Skinner's Discovery of Shaping. *J. Exp. Anal. Behav.* **82**, 317–328 (2004).
62. Turkewitz, G. & Kenny, P. A. Limitations on input as a basis for neural organization and perceptual development: A preliminary theoretical statement. *Dev. Psychobiol.* **15**, 357–368 (1982).
63. Fernando, C. & Szathmáry, E. Chemical, neuronal and linguistic replicators.
64. Stanley, K. O. & Miikkulainen, R. Competitive coevolution through evolutionary complexification. *J Artif Int Res* **21**, 63–100 (2004).
65. Gauci, J. & Stanley, K. O. Indirect Encoding of Neural Networks for Scalable Go. in *Parallel Problem Solving from Nature, PPSN XI* (eds. Schaefer, R., Cotta, C., Kołodziej, J. & Rudolph, G.) 354–363 (Springer Berlin Heidelberg, 2010).
66. Huizinga, J. & Clune, J. *Evolving Multimodal Robot Behavior via Many Stepping Stones with the Combinatorial Multi-Objective Evolutionary Algorithm.* (2018).
67. Cliff, D. & Miller, G. F. Tracking the Red Queen: Measurements of Adaptive Progress in Co-Evolutionary Simulations. in *Proceedings of the Third European Conference on Advances in Artificial Life* 200–218 (Springer-Verlag, 1995).
68. Gray, M. W. Mitochondrial Evolution. *Cold Spring Harb. Perspect. Biol.* **4**, (2012).

69. Gerhart, J. & Kirschner, M. The theory of facilitated variation. *Proc. Natl. Acad. Sci.* **104**, 8582 (2007).
70. Meeks, J. C. & Elhai, J. Regulation of Cellular Differentiation in Filamentous Cyanobacteria in Free-Living and Plant-Associated Symbiotic Growth States. *Microbiol. Mol. Biol. Rev.* **66**, 94 (2002).
71. Wagner, G. P. & Altenberg, L. Perspective: Complex Adaptations and the Evolution of Evolvability. *Evolution* **50**, 967–976 (1996).
72. Wagner, G. P., Pavlicev, M. & Cheverud, J. M. The road to modularity. *Nat. Rev. Genet.* **8**, 921–931 (2007).
73. Simon, H. A. The Architecture of Complexity. in *Facets of Systems Science* (ed. Klir, G. J.) 457–476 (Springer US, 1991). doi:10.1007/978-1-4899-0718-9_31.
74. Wilson, D. S. & Sober, E. Reviving the superorganism. *J. Theor. Biol.* **136**, 337–356 (1989).
75. Huizinga, J., Clune, J. & Mouret, J.-B. Evolving neural networks that are both modular and regular: HyperNEAT plus the connection cost technique. *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation* 697–704 (2014).
76. Clune, J., Mouret, J.-B. & Lipson, H. The evolutionary origins of modularity. *Proc. R. Soc. B Biol. Sci.* **280**, 20122863 (2013).
77. Ellefsen, K. O., Mouret, J.-B. & Clune, J. Neural Modularity Helps Organisms Evolve to Learn New Skills without Forgetting Old Skills. *PLOS Comput. Biol.* **11**, e1004128 (2015).
78. Baker, B. *et al.* *Emergent Tool Use From Multi-Agent Autocurricula*. (2019).
79. Potter, M. A. & De Jong, K. A. A cooperative coevolutionary approach to function optimization. in *Parallel Problem Solving from Nature — PPSN III* (eds. Davidor, Y., Schwefel, H.-P. & Männer, R.) 249–257 (Springer Berlin Heidelberg, 1994).
80. Mikkiläinen, R. *et al.* Multiagent Learning through Neuroevolution. in *Advances in Computational Intelligence: IEEE World Congress on Computational Intelligence, WCCI 2012, Brisbane, Australia, June 10-15, 2012. Plenary/Invited Lectures* (eds. Liu, J., Alippi, C., Bouchon-Meunier, B., Greenwood, G. W. & Abbass, H. A.) 24–46 (Springer Berlin Heidelberg, 2012). doi:10.1007/978-3-642-30687-7_2.
81. Melo, D., Porto, A., Cheverud, J. M. & Marroig, G. Modularity: Genes, Development, and Evolution. *Annu. Rev. Ecol. Evol. Syst.* **47**, 463–486 (2016).
82. Martin, W. Evolutionary origins of metabolic compartmentalization in eukaryotes. *Philos. Trans. R. Soc. B Biol. Sci.* **365**, 847–855 (2010).
83. Smith, J. M. *The Evolution of Sex*. (Cambridge University Press, 1978).
84. Frank, S. A. Genetics of Mutualism: The Evolution of Altruism between Species. *J. Theor. Biol.* **170**, 393–400 (1994).

85. Guimerà, R. *et al.* Origin of compartmentalization in food webs. *Ecology* **91**, 2941–2951 (2010).
86. Jacobides, M. G., Cennamo, C. & Gawer, A. Towards a theory of ecosystems. *Strateg. Manag. J.* **39**, 2255–2276 (2018).
87. Poli, R. Analysis of the Publications on the Applications of Particle Swarm Optimisation. *J. Artif. Evol. Appl.* **2008**, 685175 (2008).
88. Chunkai Zhang, Huihe Shao, & Yu Li. Particle swarm optimisation for evolving artificial neural network. in *Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics. 'cybernetics evolving to systems, humans, organizations, and their complex interactions' (cat. no.0 vol. 4 2487–2490 vol.4 (2000).*
89. Rigby, M. *et al.* Building an artificial neural network with neurons. *AIP Adv.* **9**, 075009 (2019).
90. Cameron, D. E., Bashor, C. J. & Collins, J. J. A brief history of synthetic biology. *Nat. Rev. Microbiol.* **12**, 381–390 (2014).
91. de Lorenzo, V. *et al.* The power of synthetic biology for bioproduction, remediation and pollution control. *EMBO Rep.* **19**, e45658 (2018).
92. Gilbert, C. & Ellis, T. Biological Engineered Living Materials: Growing Functional Materials with Genetically Programmable Properties. *ACS Synth. Biol.* **8**, 1–15 (2019).
93. Din, M. O. *et al.* Synchronized cycles of bacterial lysis for in vivo delivery. *Nature* **536**, 81–85 (2016).
94. Greco, F. V., Tarnowski, M. J. & Gorochofski, T. E. Living computers powered by biochemistry. *The Biochemist* **41**, 14–18 (2019).
95. Nielsen, A. A. K. *et al.* Genetic circuit design automation. *Science* **352**, aac7341 (2016).
96. Daniel, R., Rubens, J. R., Sarpeshkar, R. & Lu, T. K. Synthetic analog computation in living cells. *Nature* **497**, 619 (2013).
97. Pandi, A. *et al.* Metabolic perceptrons for neural computing in biological systems. *Nat. Commun.* **10**, 3880 (2019).
98. Qian, L., Winfree, E. & Bruck, J. Neural network computation with DNA strand displacement cascades. *Nature* **475**, 368–372 (2011).
99. Grozinger, L. *et al.* Pathways to cellular supremacy in biocomputing. *Nat. Commun.* **10**, 5250 (2019).
100. Brophy, J. A. N. & Voigt, C. A. Principles of genetic circuit design. *Nat. Methods* **11**, 508 (2014).
101. Gorochofski, T. E. *et al.* Toward Engineering Biosystems With Emergent Collective Functions. *Front. Bioeng. Biotechnol.* **8**, 705 (2020).

102. Castle, S., Grierson, C. S. & Gorochofski, T. E. Towards an Engineering Theory of Evolution. *Nat. Commun.* **12**, 3226 (2021).
103. Pezzulo, G. & Levin, M. Re-membering the body: applications of computational neuroscience to the top-down control of regeneration of limbs and other complex organs. *Integr. Biol.* **7**, 1487–1517 (2015).
104. Kouvaris, K., Clune, J., Kounios, L., Brede, M. & Watson, R. A. How evolution learns to generalise: Using the principles of learning theory to understand the evolution of developmental organisation. *PLOS Comput. Biol.* **13**, e1005358 (2017).
105. Solé, R., Moses, M. & Forrest, S. Liquid brains, solid brains. *Philos. Trans. R. Soc. B Biol. Sci.* **374**, 20190040 (2019).

Figures and captions

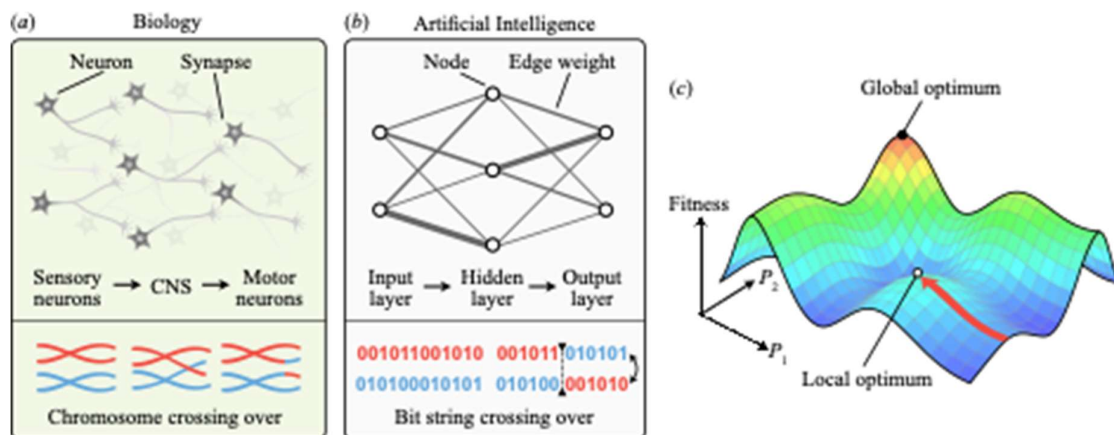


Figure 1: Basic concepts connecting biology and artificial intelligence. (a) Biological neural networks and crossing over during meiosis. Each neuron transmits an electrical impulse to multiple other neurons through synaptic connections. In this example, a sensory neuron is excited and transmits an electrical signal to the central nervous system (CNS) which results in transmission to a particular motor neuron prompting a response to the stimulus. In meiotic crossing-over, random loci along homologous chromosomes align, excise and exchange genetic material. (b) Artificial neural networks (ANNs) and crossing over in Genetic algorithms (GAs). Nodes represent neurons and are arranged in layers. Weights represent the strength of synaptic connections between pairs of nodes. In simple ANNs such as this, the input may be something like 'body mass', 'pitch of voice' and 'whisker length', with each encoded as an individual node. This is transmitted through one or multiple hidden layers, and output layer might consist of two nodes corresponding to a 'cat' and 'dog', respectively. In GAs, two mating bit strings swap their sequences. (c) Example of a fitness landscape. Horizontal axes correspond to two adjustable parameters (P_1 and P_2) and the vertical axis is the associated fitness. This landscape has multiple fitness peaks meaning that incremental improvement from some starting point (i.e. gradient ascent) may lead to a local optimum rather than global one (red arrow). Fitness landscapes tend to have many more dimensions. For example, the ANN in panel B has 12 adjustable parameters (edge weights) so its fitness landscape would have 13 dimensions.

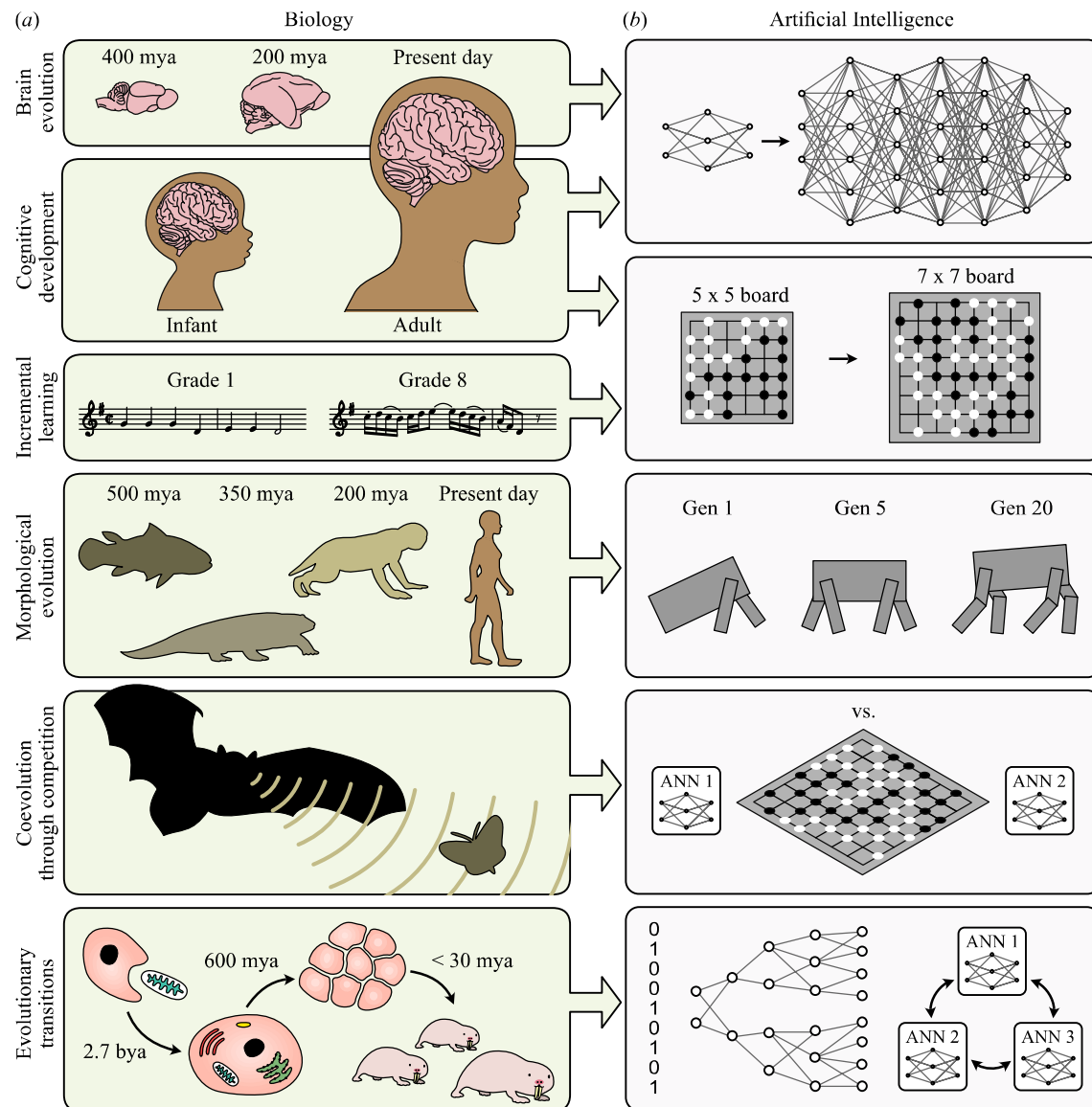


Figure 2: Complexification in biology and artificial intelligence. (a) In biology, complex cognition and morphology come about by a gradual accumulation of change. Interactions between coevolving organisms, such as bats and eared moths, continually drive evolution by creating new selection pressures. For behaviourally complex animals such as humans, cognition complexifies as the individual's brain develops into adulthood and they build their knowledge and experience. Evolutionary transitions to group living create more complex organisms, e.g. from single-celled organisms to multicellular organisms, and multiple organisms in a social group such as the eusocial naked mole rat. (b) Each example from biology is analogous to a method of complexification in AI such as simulating arms-races and evolving morphology with behaviour in robotics. Topological complexification in NEAT, where ANNs start out simple and gradually grow, enables complexification simultaneously with optimisation, and incremental evolution where learning simple tasks increases efficiency in

learning successively more complex tasks (e.g. learning the game 'Go' on a 5×5 sized board before playing on a 7×7 sized board).

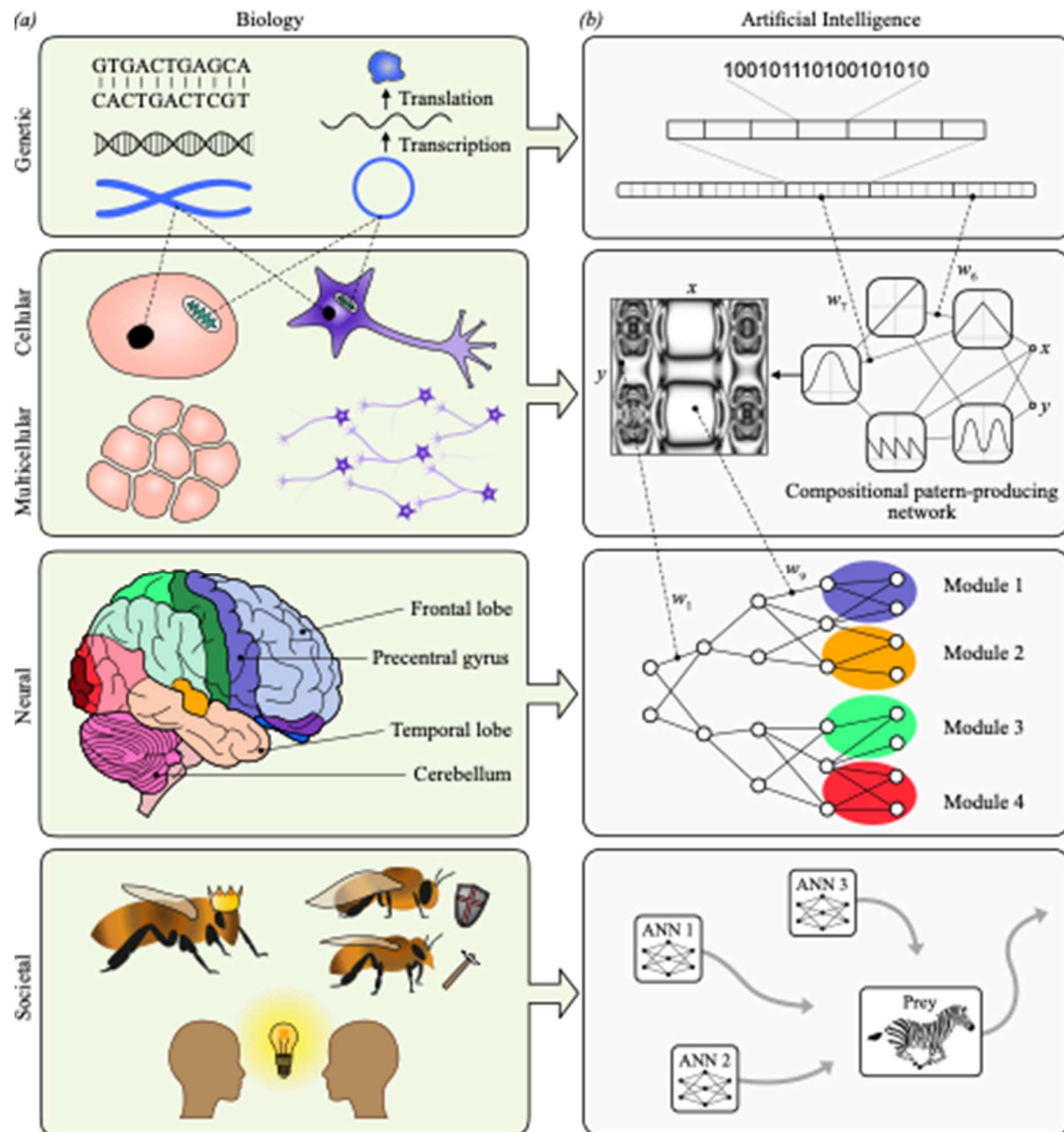


Figure 3: Hierarchy and modularity in biology and artificial intelligence. (a) Sequences of nucleotides form genes, and many genes are packed into chromosomes or plasmids. Cells are compartmentalised into organelles each specialised to their function, e.g. mitochondria are the location of respiration. In a multicellular organism, bodily processes are localised to specialised organs such as cognition in the brain. The brain is divided into many regions specialised for different cognitive processes (e.g. the prefrontal cortex is involved in logic and reasoning). Some organisms live complex social groups creating a 'superorganism' such as eusocial insect colonies. Communication between organisms creates a higher level of cognition, 'the hive mind', comparable to communications between neurons. (b) In GAs, group selection creates a hierarchy of complementary partial solutions within a bit string. Modularity can be created in ANNs where distinct clusters of nodes are responsible for different functions.

Individual agents such as ANNs can themselves represent a partial solution or function and exhibit a division of labour in a multi-agent system. Compositional pattern producing networks (CPPNs) can be used to indirectly encode ANNs, producing inherent benefits analogous to biological development. A bit string encodes the weight strengths of the CPPN (w_6 and w_7) which generates regular geometric 2D patterns which can be used to encode the weights of a functional ANN (w_1 and w_9).