CONTENTS

Introduction

THERE ARE EASIER THINGS to make than a brain. Driven by the promise and resources of biomedical research, developmental neurobiologists are trying to understand how it is done. Driven by the promise and advances of computer technology, researchers in artificial intelligence (AI) are trying to create one. Both are fields of contemporary research in search of the principles that can generate an intelligent system, a thing that can predict and decide, and maybe understand or feel something. In both developmental neurobiology and AI based on artificial neural networks (ANNs), scientists study how such abilities are encoded in networks of interconnected components. The components are nerve cells, or neurons, in biological brains. In AI, the term *neuron* has been readily adopted to describe interconnected signaling components, looking back on some 70 years of ANN research. Yet, to what extent the biological analogy is useful for AI research has been a matter of debate throughout the decades. It is a question of how much biological detail is relevant and needed, a question of the type of information necessary to make a functional network. The information problem underlies both fields. What type of information is necessary to wire a brain? What do biologists mean when they say something is "encoded by genes," and how is genetic information transformed into a brain? And finally, to what extent is the same type of information required to wire up biological brains or to create artificial intelligence?

This book is about the information problem and how information unfolds to generate functional neural networks. In the case of biological

brains, prior to learning, the information for developmental growth is encoded in the genome. Yet, there are no chapters about brain regions or their connectivity to read in the genome. In fact, compared to the information necessary to describe every detail necessary to make a functioning brain, there is rather little information available in the genome. Growth requires genetic information plus time and energy. Development happens in steps that occur in space and time in an ordered fashion. The outcome is a system that would require more information to describe than was needed to start its growth. By contrast, most ANNs do not grow. Typically, an artificial network with initially random connections learns from data input in a process that is reminiscent of how biological brains learn. This process also requires time and energy. Learning also occurs in steps, and the order of these steps matters. There are important similarities and differences between these stepwise, time- and energy-consuming processes. The current hope for AI based on ANNs is that the learning process is sufficient and that a developmental process analogous to biological brains can therefore be omitted. Remarkably, there was a time in neurobiology research almost a hundred years ago when scientists felt much the same about the brain itself. It was inconceivable where the information for wiring should come from other than through learning. The idea was that, just like ANNs today, the brain must initially be wired rather randomly, and subsequent learning makes use of its plasticity.[1] But if this were so, how could, say, a monarch butterfly be born with the ability to follow thousands of miles of a migration route that it has never seen before?

As temperatures drop in the fall in North America, millions of monarch butterflies migrate for up to 3,000 miles to overwinter in Mexico. Remarkably, millions of butterflies distributed over close to 3 million square miles in the north all target only a few overwintering sites that cover less than a single square mile. Many theories have been put forth as to how a butterfly could do this.[2,3] Similarly remarkable, an individual sea turtle will return over thousands of miles to the very beach where it was born—many years later. We do not know how sea turtles do it, but it is conceivable that they had learned and can remember something

about a place where they had once been before. This is where the story of the monarch butterfly turns from remarkable to downright unbelievable. The butterflies that started out in the north will overwinter in the south until temperatures rise next spring. They then start flying north again, but only a few hundred miles. At different places in the southern United States they stop, mate, lay eggs and die. A new generation of monarchs picks up the trail north, but again only for a few hundred miles. It usually takes 3–5 generations for a full round trip.[2] By the time temperatures drop again in the fall in North America, a monarch butterfly is about to embark on the 3,000-mile trip south to a precise location that was last visited by its great-great-grandfather. Where is this information coming from?

The currently almost exclusive focus of AI on ANNs is a highly successful, but recent development. It followed several decades during which AI and machine learning focused on formal, symbol-processing logic approaches, rather than the somewhat enigmatic neural networks. For most of its history, AI researchers tried to avoid the complexities and messiness of biological systems altogether.[4, 5] How does information about the role of a gene for a neuronal membrane protein help to program an intelligent system? The history of AI is a history of trying to avoid unnecessary biological detail in trying to create something that so far only exists in biology. The observation begs the question what information can safely be deemed "unnecessary." To address this question, we need to look at biological and artificial brain development from the information perspective. An assumption and hope of AI research has long been that there is a shortcut to creating intelligent systems. We may not yet know what shortcuts work best, but it seems a good idea to at least know exactly what it is we are trying to leave out in attempts to create nonbiological brains. My hope is that an understanding of the way information is encoded and transformed during the making of biological brains proves useful in the discussion what can and cannot be shortcut in the making of AI. This is the story of a neurobiologist tracking down that information.

The Perspective of Neurobiological Information

The biological brain is a complicated network of connections, wired to make intelligent predictions. Common analogies for brain wiring include circuit diagrams of modern microprocessors, the electrical wiring installations in skyscrapers or the logistics of transportation networks in big cities. How are such connections made during brain development? You can imagine yourself trying to make a connection by navigating the intricate network of city streets. Except, you won't get far, at least not if you are trying to understand brain development. There is a problem with that picture, and it is this: Where do the streets come from? Most connections in the brain are not made by navigating existing streets, but by navigating streets under construction. For the picture to make sense, you would have to navigate at the time the city is growing, adding street by street, removing and modifying old ones in the process, all the while traffic is a part of city life. The map changes just as you are changing your position in it, and you will only ever arrive if the map changes in interaction with your own movements in it. The development of brain wiring is a story of self-assembly, not a global positioning system (GPS).

When engineers design the electrical wiring in a building or a computer microchip, they have the final product in mind. We make blueprints to understand and build engineered systems with precise outcomes. A blueprint shows a picture of the final product, the endpoint. A blueprint also contains all the information needed to build that product. It largely doesn't matter in what order the pieces are put in, as long as everything is in place when you flip the on switch. But there is no blueprint for brain connectivity in the genes. There is also no such information coming from the environment. If neither the genes nor the environment contain endpoint information of connectivity, what kind of information do they contribute?

Genetic information allows brains to grow. Development progresses in time and requires energy. Step by step, the developing brain finds itself in changing configurations. Each configuration serves as a new basis for the next step in the growth process. At each step, bits of the

genome are activated to produce gene products that themselves change what parts of the genome will be activated next—a continuous feedback process between the genome and its products. A specific step may not have been possible before and may not be possible ever again. As growth continues, step by step, new states of organization are reached. Rather than dealing with endpoint information, the information to build the brain unfolds with time. Remarkably, there may be no other way to read the genetic information than to run the program. This is not a trivial statement to make, and it will take some explaining. If there is no way to read the genetic code other than running it, then we are principally unable to predict exact outcomes with any analytical method of the code. We can simulate it all right, but the result would not have been predictable in any way other than actually running the whole simulation. The information is in the genes, but it cannot be read like a blueprint. It really is a very different type of information that requires time and energy to unfold.

The Perspective of Algorithmic Information

Scientists in nonbiological fields are more familiar with this type of information. There is a simple game, where you draw lines of X's or O's (or black dots versus blanks) based on simple rules that produce remarkable patterns. Imagine a single X in a row of an infinite number of O's and a simple rule that determines for each triplet of X's and O's whether there is an X or an O in the next row. To find out the next line, you read the first three characters, write the output X or O underneath the center of the triplet below, then move one character and do it again for the next partially overlapping triplet. One rule, called rule 110, looks innocently enough like this:[6]

Triplet in previous row:	XXX	XXO	XOX	XOO	OXX	OXO	OOX	OOO
..determines in next row:	O	X	X	O	X	X	X	O

For example, starting with one X: .. OOOOOXOOOOO..
will lead to the next row: .. OOOOXXOOOOO..

Repeating this process again and again, using each previous line to apply the rule and write the next one below, will create a two-dimensional pattern (you will find the result in figure 2.3 on page 96). The repeated application of defined rules is an iteration. A ruleset that uses the output of each preceding step as the input of the next step defines an algorithm. The two-dimensional pattern is the outcome of algorithmic growth based on the iterative application of simple rules. But what does this game have to do with brain development? Shockingly, for the simple rule shown above, the two-dimensional pattern turns out to be so surprisingly complicated that it was proven to contain, at some point of its pattern growth process, any conceivable computation. Mathematicians call this a universal Turing machine or "Turing-complete." This is not an intuitive concept. The information content of the underlying code is absurdly low, yet it can produce infinite complexity. What is more, there is no analytical method to tell you the pattern at iteration 1,000. If you want to know, you must play the game for 1,000 rounds, writing line by line. These systems are called cellular automata and are a beloved model for a branch of mathematics and the research field of Artificial Life (ALife). Some ALifers consider AI a subfield. Many AI researcher don't care much about ALife. And neither of them care much about developmental neurobiology.

In information theory, the cellular automaton described above highlights an important alternative to describing complete endpoint information. Instead of a precise description of every detail of the pattern after 1,000 iterations, a complete description of the system is also possible by providing the few simple rules plus the instruction "apply these rules 1,000 times." The information required to generate the complete system is also known as Kolmogorov complexity in algorithmic information theory. Data compression algorithms do exactly that. An image of a uniformly blue sky is easily compressed, because its algorithmic information content is low (paint the next 10,000 pixels blue). By contrast, a picture where every pixel has a random different color and no repeating patterns cannot easily be compressed. In the case of the cellular automaton, Kolmogorov complexity is very low, while endpoint information required to describe the system becomes infinite with infinite

iterations. The algorithmic information content required to create the system are a few instructions plus time and energy, while the endpoint information content is enormous in the case of many iterations.

The rule 110 cellular automaton provides us with a simple example of an algorithmic growth process that can generate more information based on simple rules, and yet its output can only be determined by letting it grow. "More" information is defined here as the information needed to describe the output if there were no growth process. However, in contrast to biological systems, rule 110 can only produce one fixed outcome with every iteration based on a set of rules that never change. For these reasons alone, rule 110 cannot be a sufficient model for biological systems. Yet, rule 110 teaches us that unpredictable unfolding of information is possible even with very simple rules in a deterministic system. For rule 110 there is a proof, the proof of Turing universality. For biological growth based on the genetic code, we face many more challenges: The rules are more complicated and change with every iteration of the running algorithm, and stochastic processes are central to its run. If a simple system like rule 110 can already be unpredictable, then we should not be surprised if algorithmic growth of biological systems turns out to be unpredictable. However, the proof for biological systems seems currently out of reach. The idea that information unfolding based on genomic information cannot be mathematically calculated, but instead requires algorithmic growth or a full simulation thereof, is a core hypothesis of this book.

A Shared Perspective

Biologists like to talk about the genes that contain a certain amount of information to develop the brain, including its connectivity. But in order to appreciate the information content of genes, we must understand the differences and consequences of information encoding for a self-assembling system versus a connectivity map. The genetic code contains algorithmic information to develop the brain, not information that describes the brain. It can be misleading to search for endpoint information in the genes or the mechanisms of the proteins they encode.

Address codes, navigational cues and key-and-lock mechanisms all follow such a rationale and make intuitive sense. And they all exist as molecular mechanisms, in brain wiring as elsewhere in biology. But they are part of unfolding algorithmic information, not endpoint information of brain connectivity. As the brain grows, different genes are turned on and off in a beautiful ballet in space and time, endowing each individual neuron with a myriad of properties that play out and change in communication with its neighbors. The neuron navigates as the city map grows and changes in interaction with the neuron's own movement in it.

The study of genes in developmental neurobiology is a success story from at least two perspectives. First, in the quest for molecular mechanisms. What a gene product does at any point in time and space during brain development tells us something about a part of the growth program that is currently executed. But information about a specific molecular mechanism may only be a tiny part of the information that unfolds in the wake of a random mutation in the genome. A mutation can lead to more aggressive behavior of the animal. And yet, the mutation may well affect some metabolic enzyme that is expressed in every cell of the body. The molecular function of the gene product may tell us nothing about animal behavior. How the molecular mechanism of this gene is connected to the higher order behavior may only be understood in the context of the brain's self-assembly, its algorithmic growth.

Many mutations have been found that change predispositions for behavioral traits, yet there may be only very few cases that we could reasonably call "a gene for a trait." Most gene products contribute to develop the trait in the context of many other gene products, but do not contain information about the trait itself. A mutation, selected by evolution for behavioral changes, must change either brain development or function. If the effect is developmental, then we have to face the information problem: There may be no way to know what the altered code produces other than running the entire process in time (or simulating it on a computer). There may be no shortcut. This is the problem with the street navigation analogy: You have to navigate a changing map on a path that only works if the map changes just as you are navigating it.

The full route on the map never existed, neither at the beginning nor at the end of your trip, but instead the route was made in interaction with your own actions. This is the essence of self-assembly.

We can study self-assembly either as it happens in biology or by trying to make a self-assembling system from scratch. As of 2020, biological neural networks (i.e., brains) are still unparalleled in their intelligence. But AI is on it. And yet, self-assembly is not a major focus of AI. For many years, AI focused on formal symbol-processing logic, including enormous expert systems built on decision-making trees. As recently as the early 2000s, the victory of formal, logical symbol-processing AI was declared. Since then, just when some thought we were done with neural networks, a revolution has taken place in AI research. In the few years since 2012, practically every AI system used to predict what friends or products we allegedly want has been replaced with neural networks. "Deep learning" is the name of the game in AI today.

The ANNs we use as tools today are not grown by a genetic code to achieve their initial architecture. Instead, the initial network architecture is typically randomly connected and thus contains little or no information. Information is brought into an ANN by feeding it large amounts of data based on a few relatively simple learning rules. And yet, there is a parallel to algorithmic growth: The learning process is an iterative process that requires time and energy. Every new bit of data changes the network. And the order matters, as the output of a preceding learning step becomes the input of the next. Is this a self-assembly process? Do we ultimately need algorithmic growth or self-assembly to understand and create intelligence? One obvious problem with the question is that the definition of intelligence is unclear. But the possible role of self-assembly may need some explaining, too.

In the search for answers, I went to two highly respected conferences in late summer 2018, an Artificial Life conference themed "Beyond Artificial Intelligence" by the International Society for Artificial Life and the Cold Spring Harbor meeting "Molecular Mechanisms of Neuronal Connectivity." I knew that these are two very different fields in many respects. However, my reasoning was that the artificial life and artificial intelligence communities are trying to figure out how to make something

that has an existing template in biological systems. Intelligent neural networks *do* exist; I have seen them grow under a microscope. Surely, it must be interesting to AI researchers to see what their neurobiology colleagues are currently figuring out—shouldn't it help to learn from the existing thing? Surely, the neurobiologists should be equally interested in seeing what AI researchers have come up with, if just to see what parts of the self-assembly process their genes and molecules are functioning in.

Alas, there was no overlap in attendance or topics. The differences in culture, language and approaches are remarkable. The neurobiological conference was all about the mechanisms that explain bits of brains as we see them, snapshots of the precision of development. A top-down and reverse engineering approach to glimpse the rules of life. By contrast, the ALifers were happy to run simulations that create anything that looked lifelike: swarming behavior, a simple process resembling some aspect of cognition or a complicated representation in an evolved system. They pursue a bottom-up approach to investigate what kind of code can give rise to life. What would it take to learn from each other? Have developmental biologists really learned nothing to inform artificial neural network design? Have Alifers and AI researchers really found nothing to help biologists understand what they are looking at? I wanted to do an experiment in which we try to learn from each other; an experiment that, if good for nothing else, would at least help to understand what it is that we are happy to ignore.

So I assembled a seminar series, a workshop, about the common ground of both fields. The seminars are presented from the perspective of a neurobiologist who wants to know how our findings on brain development relate to the development of ANNs and the ultimate goal of artificial general intelligence. Many neurobiologists feel that ANNs are nothing like the biological template, and many AI scientists feel that their networks should not try to resemble biology more than they currently do. The seminars are therefore presented with a broad target audience in mind: there is so little common ground that it is easily shared with any basic science-educated layperson. The average neurobiologist is a layperson when it comes to AI, and most ANN developers are

laypeople when it comes to neurobiology. Developmental neurobiologists may feel they are not missing anything by not following the bottom-up approach of AI, and ANN developers may feel they are safe to ignore biological detail. But to decide what is not needed, it helps to at least know what it is we are choosing to not know.

One of the best outcomes of good seminars are good discussions. And here I didn't need to search long. Going to conferences with these ideas in mind has provided me for years with experiences for how and where such discussions can go. I started writing this book with these discussions in mind. Initially, I only used them as a guide to pertinent questions and to identify problems worth discussing. As I kept on going back to my own discussions and tried to distill their meaning in writing, it turned out all too easy to lose their natural flow of logic and the associations that come with different perspectives. So I decided to present the discussions themselves. And as any discussion is only as good as the discussants, I invented four entirely fictional scientists to do all the hard work and present all the difficult problems in ten dialogs. The participants are a developmental geneticist, a neuroscientist, a robotics engineer and an AI researcher. I think they are all equally smart, and I do hope you'll like them all equally well.

The Ten Seminars

The seminars of the series build on each other, step by step. Preceding each seminar is a discussion of the four scientists who exchange questions and viewpoints in anticipation of the next seminar. The series starts with **The Historical Seminar: The Deeply Engrained Worship of Tidy-Looking Dichotomies**, a rather unusual seminar on the history of the field. The "field" being really two fields, developmental neurobiology and AI research, this seminar provides an unusual and selective historical perspective. Their shared history puts each other's individual stories in the spotlight of shared questions and troubles. Both struggle with remarkably similar tension fields between seemingly opposing approaches and perceptions. There are those who feel that the approaches, hypotheses and analyses must be rigorously defined for any

outcome to be meaningful. Then there are those who feel that, like evolution, random manipulations are okay as long as one can select the ones that work—even if that means giving up some control over hypotheses, techniques or analyses.

Both fields begin their shared history by independently asking similar questions about information. The discovery of individual nerve cells itself was a subject of divisive contention. Even before scientists were sure that separable neurons exist, concerns were already raised about the information necessary to put them all together in a meaningful network. Much easier to envision the network as a randomly preconnected entity. And when early AI researchers built their very first networks with a random architecture, they did so because they felt it had to be like that in nature—where should the information have come from to specifically connect all neurons? A randomly connected network contains little or no information; the network has to grow smart through learning. In biology, the dominance of this view was challenged already in the 1940s by studies that focused on the precision and rigidity of connectivity that is not learned. This work marked a turning point that led neurobiologists to ask questions about how network information can develop based on genetic information. By contrast, today's artificial neural networks used in typical AI applications still only grow smart by learning; there is no genetic information. Yet, years in both fields played out in similar tension fields between precision and flexibility, between rigidity and plasticity. The fields may not have talked much to each other, but they mirrored each other's troubles.

The historical background forms the basis for three sessions. The first session explores the types of information that underlie biological and artificial neural networks. The second session builds on the information-theoretical basis to discuss the approaches taken by biologists to understand how genetic information leads to network information—the missing element in most ANNs. The third session connects algorithmic growth to learning and its relevance for AI.

Each session consists of three seminars. The first session starts with **Seminar 2: From Algorithmic Growth to Endpoint Information**, which deals with the difference between information required to make

a system and information required to describe a system. Genes contain information to develop neuronal connectivity in brains; they don't contain information that describes neuronal connectivity in brains. We are facing one of the hardest problems right from the start, mostly because human intelligence lacks intuition for this kind of information. The core concept is algorithmic growth. A set of simple rules is sufficient to create mindboggling complexity. But what is complexity? The journey to understand information encoding is intricately linked to this question. If a cellular automaton based on a very simple rule set can produce a Turing-complete system, including unlimited complexity of patterns, where is the information coming from? The algorithmic information content of the rules and the game is sufficient to create the entire system. This is very little information, and there is clearly no complexity there. On the other hand, the analysis of the pattern created by such a cellular automaton reveals unlimited depth. To describe the pattern requires a lot of information, something we like to call complex. All the while, the cellular automaton is a deterministic system, meaning repeated runs with the same rules will always produce the same pattern. The information for the development of this precision is somehow in the rules, but only unfolds to our eyes if the rules are applied iteratively, step by step, in a time- and energy-consuming process. This is the idea of algorithmic growth. The brain develops through algorithmic growth. Yet, in contrast to the cellular automaton, brain development includes nondeterministic processes and the rules change during growth. How useful is the analogy of the cellular automaton in light of these constraints? This question brings us back to the information that is encoded by the genetic code. When we discuss genes, we focus on biological neural networks. In the process, we learn about the type of information and the consequences of growth and self-assembly that define the network's properties. These are the types of information that are typically left out in ANN design, and they may thus serve as a survey of what exactly is cut short in AI and why.

Seminar 3: From Randomness to Precision explores what happens when we add noise to algorithmic growth. Both an elementary set of rules for a one-dimensional cellular automaton or a genetic code will

deterministically produce identical results with every run in a precise computer simulation. But nature is not a precise computer simulation, or at least so we think. (Yes, the universe could be a big deterministic cellular automaton, but let's not go there for now.) Biology is famously noisy. Noise can be annoying, and biological systems may often try to avoid it. But noise is also what creates a pool of variation for evolution to select from. From bacteria recognizing and moving towards sugar to the immune system recognizing and battling alien invaders, nature is full with beautifully robust systems that only work based on fundamental random processes that create a basis for selection. We will have some explaining to do, as we transition from the idea of simple rules that yet produce unpredictably complex outcomes on one hand to perfectly random behavior of individual components that yet produce completely predictable behavior on the other hand. Intuition may be of limited help here.

Awe and excitement about brain wiring mostly focuses on the exquisite synaptic specificity of neural circuitry that ensures function. As far as specific connectivity is absolutely required for precise circuit function, synaptic specificity has to be rigid. On the other hand, the brain develops with equally awe-inspiring plasticity and robustness based on variable neuronal choices and connections. In particular, neurons that find themselves in unexpected surroundings, be it through injury or a developmental inaccuracy or perturbation, will make unspecific synapses with the wrong partners. In fact, neurons are so driven to make synapses that scientists have yet to find a mutation that would prevent them from doing so as long as they are able to grow axons and dendrites and contact each other. Neurons really want to make synapses. If the right partner can't be found, they'll do it with a wrong partner. If a wrong partner can't be found, they'll do it with themselves (so-called autapses). This is what I call the *synaptic specificity paradox*: How can synaptic specificity be sufficiently rigid and precise to ensure function, if individual neurons are happy to make unspecific synapses?

The answer is closely linked to algorithmic growth: promiscuous synapse formation can be permissible, or even required, depending on when and where it occurs as part of the algorithm. For example, many

neurons have the capacity to initially form too many synapses, which contain little information. Through subsequent steps of the growth algorithm, this pool of synapses will be pruned and refined, thereby increasing the information content in the network. Rules for the weakening or strengthening of synapses are a core functional principle of all neural networks, both biological and artificial. This reminds us of brain function, learning and memory. But remarkably, neuronal activity can be part of the growth algorithm, long before there is even an opportunity for meaningful environmental input or learning. I call this *post-specification*, the specification of synapses late in the developmental algorithm, following initially more promiscuous synapse formation. By contrast, synaptic *pre-specification* occurs when only certain neurons get to see each other in space and time during their period of synaptogenic competency, i.e., the time window when they can make synaptic connections. If the patterns of the running algorithm restrict the synaptic partners that get to see each other, the problem of identifying the partner is greatly facilitated. The more spatiotemporal positions pre-specify partnerships, the more promiscuous, random synapse formation is permissible.

Random processes therefore need not be an enemy of precision in neural networks. Instead, random processes are abundantly utilized during algorithmic growth of the brain, just as in so many other biological processes. But random developmental processes do not necessarily produce variability in the outcome; they can also lead to almost perfectly precise synaptic connectivity patterns. And random developmental processes give rise to two of the most astonishing properties of biological brain wiring: flexibility and robustness. Connections not only change with experience, but also rewire in response to injury and developmental perturbation. ANNs also have some of these properties. And yet, historically, both neurobiology and AI had a rather strained relationship with randomness. Even today, most neurobiologists and ANN developers will consider noise as something to avoid, rather than as a design principle for a network. An understanding of the roles of noise will bring us closer to appreciating how to make networks flexible and robust in addition to making them with precision.

Seminar 4: From Local Rules to Robustness brings us back to the essence of self-assembly: local interactions during algorithmic growth. In order for local interactions to flexibly react to changing environments, local agents must be able to make their own decisions, independent of, and unknowing of, the larger system they create. This is the concept of autonomous agents. If the individual players of a soccer game would not make their own decisions, the game would be boring. If the players would not follow a common set of rules, the game would fall apart. The local interactions, the players' decisions and flexibility, make the game interesting (if this kind of game happens to be interesting to you) and robust. The outcome is unpredictable at the level of the individual game, but the average outcomes over seasons are remarkably predictable. Which of the two, the individual game or the average season, is more interesting is in the eye of the beholder. For biological systems the beholder is evolutionary selection. For example, whatever local molecular and cellular interactions lead to different fingerprints may leave the outcome unpredictable at the level of the individual thumb, but perfectly predictable and robust at the level of selectable functionality.

In neural networks, both development and function vitally depend on individual neurons behaving as autonomous agents. The growing tip of a neuron employs random exploration of its environment through filopodia, tiny fingerlike protrusions. The neuron must be allowed to individually and locally decide whether it likes something it senses on the left or on the right using these protrusions. Similarly, the ability to learn in both biological and artificial neural networks relies on individual neurons, and individual synapses, to adapt their function. The concept of autonomous agents has made repeated stage appearances in AI. In all cases, the actions of autonomous agents only make sense in the context of a process that develops in time. The agents' decisions and collective actions set the stage for higher order organization that develops step by step. They are part of self-assembly in space and time. And this brings us, at the end of session 1, back to the question of types of information. It is possible to describe, in arbitrary detail, the precise angles of the movements of every soccer player or a neuron's growing protrusions. However, at what level of detail a situation must be described

in order to understand a distinct step of the underlying growth algorithm is not an easy question to answer.

In the second session we approach this question by diving into the realities of players and rules during the self-assembly of the biological brain prior to learning. All three seminars in this session focus on those aspects of the neurobiological history and their outcomes that are critical from the perspective of information theory: When and where *does* the information get into the network? ANNs used in AI today do not encode much information prior to learning; they are engineered and switched on for training. The second session is therefore all about biology, but with the goal to understand what it is exactly that ANNs are leaving out.

A powerful way to study brain development is experimental perturbation through mutation of the genetic code. After all, evolution did it before: genetic changes that affect development result in changes to the brain. The evolutionary approach is based on trial and error and does not require a need to predict the outcome of a genetic change as long as it can be selected. Selection of heritable, meaningful changes are evolution's way of reprogramming the brain. But what are these meaningful changes to the genome? Are there special genes for the brain and behavior, or could any mutation in the genome help to reprogram the brain through information unfolding during development?

The second session starts with **Seminar 5: From Molecular Mechanisms to Evolutionary Programming**, in which we will explore these questions by analyzing how mutations can reprogram animal behavior. We will approach the answer through a discussion of programming by evolution: If a mutation causes heritable, meaningful and selectable change, then evolution can use it to rewire and reprogram the network. For this to work, it is not necessary that the functional mechanism of the protein encoded by the mutated gene is in any intuitive or direct way related to connection specificity. Rather, the effect of a mutation has to be such that the developmental process, and the unfolding of information that comes with it, reproducibly change the network. In this way, a behavioral predisposition can certainly be caused by a single mutation, yet there need not be a single "gene for that behavior."

The fact that single mutations in the genome can reprogram animal behavior is well established. Pioneering experiments with fruit flies have surprised and enlightened this field for more than 50 years. Examples include mutants that affect courtship behavior and the internal clock that predictively guides behavior through the daily cycle of day and night. Importantly, the way such mutations were (and still are) found is based on accelerated evolution in the lab. The first step is to dramatically increase the occurrence of random mutations without any prediction as to what this might cause. The second steps is to let thousands of the randomly mutagenized animals develop. The third step is to take those flies that survived the high mutation rate and assay them for behavioral alterations. These forward genetic screens for behavioral mutants led to the successful identification of mutants with altered behavior; over the years, several genes harboring these mutations were discovered. Some of the best studied of these genes are those where a molecular function directly relates to the behavior. There are beautiful examples, but they may be the exceptions. Most mutations that modify animal behavior affect genes that function in surprising ways during developmental growth, often at many steps or in many different cells. Such mutations can lead to heritable, meaningful and selectable behavioral change, but not through specific molecular mechanisms that are related to the behavior itself. Mutations may cause unpredictable developmental alterations that nonetheless lead to reproducibly different brains based on changes in network development or function. Those are the mutations that served evolution in the slow, trial-and-error reprogramming of brains and their astonishing behavioral innate programs. There is no single gene solely responsible for the spider's ability to weave a species-specific web or the monarch butterfly's ability to migrate a route of thousands of miles over a succession of generations. If our goal is to understand the programming of a neural network that accomplishes such feats, we must step beyond the idea of a single gene coding for a single behavior. We must learn how evolution reprograms the abilities of networks, including human intelligence.

Seminar 6: From Chemoaffinity to the Virtues of Permissiveness deals with the historical and ongoing quest of developmental

neurobiologists to understand underlying molecular mechanisms. The invariable hope is to draw direct lines from mutations to genes to the gene products' molecular mechanisms in trying to decipher neural network design. Neurobiologists prefer to characterize those genes whose gene products execute molecular mechanisms that make intuitive sense with respect to neuronal connectivity, hence the terms "guidance molecules" or "chemoattractants." This is such a powerful idea and prominent concept in developmental neuroscience that we need to discuss examples of such molecules and their roles during algorithmic growth in some detail.

In search of information encoding for brain wiring, the holy grail has been the search for mechanisms that *instruct* the neuron where to make a connection. The idea of instructive mechanisms contrast with permissive mechanisms, which may be necessary to allow growth, but do not guide it actively. Oddly, the most widely used attractant for the guidance of growing neuronal protrusions in experimental culture is nerve growth factor—NGF. This is a molecule that the neuron needs to grow. By providing NGF only on the left, but not on the right, we can make neurons grow robustly to the left. This is clearly instructive. But wait, it's a growth factor! The neuron simply will not grow where it is not present. That's rather permissive. Obviously, a permissive mechanism (like a growth factor) can contribute to the neuron's choice where to grow. From an information-theoretical perspective, the information for the directionality must have previously been provided in the location of the growth factor, which may lay out an entire path. The factor itself may be permissive, but the path it marks is instructive. Which brings us to the information needed to mark the path—and that brings us back to algorithmic growth where, step by step, paths may be laid out through the interactions of many autonomous agents, including the growing neuron itself. The path may not exist either at the beginning or the end of the neuron's journey, but results from the interactions of the neuron with its surroundings as it grows. Some molecules on neuronal and non-neuronal surfaces convey local and temporally restricted attractive or repulsive signals. Yet other molecular mechanisms alter the propensity of the neuron to further grow extensions at all or gain or lose the

capacity to make synapses, or alter its mechanical interactions with the surroundings. In the context of algorithmic growth, the composite of all these factors determines the rules for each step in the algorithm. A genetic instruction need not be attached to a molecular mechanism of a single gene product. Instead, *composite instructions* are fleeting states of the system defined by the molecular and cellular contexts that happen to emerge at any given time and place during algorithmic growth.

Seminar 7: From Genes to Cells to Circuits is all about levels, from molecules to neural circuits, as we move towards the function of neurons in the network. How is it that in the field today the study of neural circuit function is obviously a question to be studied at the level of cells, while the study of the same neural circuit's assembly is obviously a question to be studied at the level of molecules? This brings us back to the type of information encoded in the genome and its relation to processes at other levels. Single genes usually do not describe processes at higher levels, even though a specific mutation in a single gene can heritably and meaningfully change that process.

The levels problem is particularly pertinent when we are trying to span all the levels from the immediate effects of a mutation in the genome to a behavioral trait. Genome-wide association studies try to establish probabilities for a given genomic variation to be associated with a specific behavior. The probabilistic nature of the data and the difficulty to establish causality in such experiments is directly linked to the nature of algorithmically unfolding information.

Neuronal function is the level at which grown biological networks and engineered artificial networks meet. But in the biological template neuronal activity can in fact be part of the genetically encoded growth algorithm. Neuronal activity is part of information unfolding. As we have already discussed in the context of synaptic post-specification, activity is known to kick in before there is any environmental input. Correlated neuronal activity is one of the ingredients of algorithmic growth that require a random process to function robustly. It also provides a bridge to neural network function and AI.

The third session is all about transitions. First, there is the transition from development to function in neural networks. Next, the transition

from the naïve to the trained and functional network, and with it the transition from biological to artificial networks. The transition from dull to intelligent. And then there is the transition from separate biological and artificial networks to their interactive future. In all cases, the idea of information unfolding in a time- and energy-consuming manner serves as framework to assess possibilities and limitations.

In **Seminar 8: From Development to Function** we explore in what ways developmental self-assembly is relevant for network function. We will start with the burden of evolution and development for biological brains. Evolution had to work with the outputs of previous states, no matter how inappropriate they may have been when selection pressures changed. Evolutionary change happens in steps, however small, in a necessarily sequential manner. And the process takes, of course, time and energy. As a result, brains feature some remarkable and apparently nonsensical oddities that only make sense in light of development—and the way development was slowly shaped over millennia of evolutionary modification.

These kinds of biological oddities, and messiness, led computer enthusiasts who were trying to develop AI in the '80s to take pride in ignoring what their neuroscience colleagues were doing. "We can engineer things much better," they may have thought, so why learn about the nonsensical solutions biology had to put up with?

And yet, if we avoid the burden of developmental history by starting with a randomly connected network prior to learning, the first problem we are confronted with is the time and energy it takes to train. And training, again, is a stepwise, time- and energy-consuming process. The order of input matters. And the ultimate function of the network is made possible, and burdened, by its training history. We will explore how information is stored in biological and artificial networks. How does the neural network save a four-digit PIN? The amount of bits needed to store this information is clearly defined in computational terms. Yet, neural networks save and retrieve this information flexibly and robustly, even if random neurons in the network drop out. In addition, the biological network does not have a separate training and function period. Learning is inseparable from using the network; storing is

inseparable from retrieving information. And again, we meet an evolutionary principle and the power of sequences in time. Many bits of information in the biological network—memories—can only be accessed by going through a sequence in time. Try saying your phone number in reverse order. How is this information content programmed, stored and retrieved?

If self-assembly is any guide, then information has to enter by changing the sequential, auto-associative network, which means it changes algorithmic information. Maybe memories should not be understood as stored entities at all, but rather as algorithmic rules sufficient to recreate them with a certain robustness, flexibility and variability. This bring us back to the cellular automaton that does not store the memory of the pattern at iteration 1,000, but instead the information to recreate this state. We will explore to what extent this process resembles algorithmic growth, and how it transitions from development to function.

In **Seminar 9: From Algorithmic Growth to Artificial Intelligence** we focus on artificial neural networks and their relationship to self-organization and algorithmic growth. We will finally also discuss definitions of *self-assembly* and *intelligence*. Most ANNs are based on the idea of an engineered design, flipping the on switch and training the network. By contrast, in biological networks the information encoding goes hand in hand with the development of the brain. The brains of a newborn, a toddler or a 10-year-old are clearly recognizable for their developmental stages morphologically, functionally and by their behavioral output. The question is whether a tedious, years-long process of self-assembly is a desirable step to create an artificially intelligent system. More specifically, is there ever a need to grow a neural network, or is training a predesigned network like in deep learning sufficient, maybe equivalent, or even superior?

A common idea in ANN development is that the product of development is only hardware infrastructure. A lot of biological idiosyncrasies can be congealed in a single parameter, like the synaptic weight. These are shortcuts that have served ANNs well for many years and many tasks. Yet, a key question associated with this reduction is how it may limit learning. In biology, the single parameter contains none of the

parameter space necessary for evolutionary programming to modify an algorithmically growing network. Based on these considerations, we dive deeper into the way engineered ANNs do, and do not, function.

Finally, in **Seminar 10: From Cognitive Bias to Whole Brain Emulation**, we will discuss the consequences of algorithmic information storage in neural network for the function and interaction of biological and artificial networks. We will start with a discussion of heuristics, the probabilistic nature of any information in the network. Neural network function is less the computing of input in conjunction with stored data based on logical operations, and more a process of probabilistic alignment and selection of patterns based on previous experiences. Both biological and artificial networks are biased by their experience. An ANN that has only been trained with numbers 0 to 9 will interpret the picture of an elephant as a number from 0 to 9.

We are all well-trained neural networks, but our brains come with a history track, as do ANNs. New information is not stored independent of other safely stored information content. Instead, any new bit of information is processed in the context of the entire history of the network. The same experience means something different for every individual. And the better the information is aligned with previous experiences, the easier it is for the network to "believe" the new arrival. This simple thought has some interesting consequences for the concept of cognitive bias: in a network built on algorithmic growth, bias is a feature, not a bug of the system, whether we like it or not.

Finally, if information is stored as an algorithmic process that requires time and energy, can it be retrieved and transferred *in toto*? That is, what does the self-assembling brain teach us about the potential to upload or download our brains? If information is not stored in any dedicated bits, but as algorithmic rules sufficient to recreate that information, then bandwidth of connection may not be the biggest challenge for data transfer. In the development of AI, we continue the debate about how similar artificial systems should be to the biological analog. But if we want to extend or copy our own brains, a clearer understanding of how information is actually stored or retrieved is needed. We encounter the levels problem again. To generate artificial human

intelligence, what parts of the algorithmic growth of the human brain can be cut short? In the design of ANNs, shortcuts are useful to shorten computation time by throwing out irrelevant detail. This approach works, as long as we do not need or want to simulate, say, spatially and temporally restricted modulation of many synapses through diffusible neuromodulators. But if we want to simulate human intelligence, don't we need the neuromodulator, since circuit function requires synaptic changes that depend on the neuromodulatory context? We come to the question of the AI we want to generate. The shortcuts we choose in the development of artificially intelligent systems define what intelligence we get.

On Common Ground

This book was written with readers in mind that are interested in developmental biology or AI alike. However, those deeply immersed in either field will find much that is treated too superficially or from an unfamiliar perspective. I did not attempt to provide objective overviews over either field's history or main achievements; many great works already exist on both accounts and are referenced throughout the seminars. My goal was to identify common ground, with a focus on underlying questions of information encoding. My hope is that a reader with deeper knowledge in either topic will still find reason to smile when trying to think what it may read like for someone in the other field.

I am convinced that all concepts presented here have been part of many ideas in different contexts before. Algorithmic growth in particular is not a new concept. It is implicit in all developmental processes and any attempt to understand how the genome encodes growing things. Yet, intuitive and mechanistic thinking in either field rarely considers the implications of unpredictable information unfolding. Those familiar with self-organizing systems may find most concepts presented here oversimplified, or my definition of self-assembly (seminar 9) wanting. Similarly, developmental neurobiologists are likely to find much that could have been added from the boundless list and beauty of molecular mechanisms underlying neural network development and function. But

common ground lies more in the motivation, the desire to understand how neural networks grow smart, than in the details of the individual disciplines. On this account, I hope the historical perspectives presented throughout the seminars may provide helpful parallels.

I am aware that many ALife and AI researchers may feel that reading a book written by a neurobiologist is not likely to be helpful for their work, both for reasons of perspective and the inevitable focus on unhelpful biological "messiness." Similarly, some developmental neurobiologists may currently read a book or two on the application of deep learning to analyze their data, rather than to learn from the AI community about how "real" brains come to be. I started this project wishing there were a book that both would enjoy having a look at, or at least get sufficiently upset about to trigger discussion between the fields.

INDEX

Page numbers in *italics* refer to figures.