

SYDE 556/750
Simulating Neurobiological Systems
Lecture 1: Introduction

Andreas Stöckel

Based on lecture notes by
Chris Eliasmith and Terrence C. Stewart

January 7, 2020



Accompanying Readings: Chapter 1 of Neural Engineering

Contents

1 Overview	1
1.1 Overall Goal of This Course	1
1.2 Theoretical Neuroscience and Levels of Organization	1
1.3 Theoretical Neuroscience vs. Theoretical Physics: An Analogy	3
2 Neural Modelling	3
2.1 Problems with current modelling approaches	4
2.2 The Brain	6
2.3 Kinds of Data From the Brain	9
2.4 Summary	11
3 The Neural Engineering Framework and the Semantic Pointer Architecture	12
3.1 Neural Engineering Framework	12
3.2 Semantic Pointer Architecture	13
3.3 The NEF and SPA as tools for hypothesis generation	13
3.4 The Nengo Simulator	14
4 Representation, Transformation, and Dynamics	14
4.1 Representation	14
4.2 Transformation	17
4.3 Dynamics	17
5 Examples	18
6 Conclusion	19

1 Overview



Note: This entire lecture is meant to give you a broad, whirlwind overview of the material in the course. We are going to revisit most of the topics mentioned here in much more detail in the upcoming weeks.

1.1 Overall Goal of This Course

The overall goal of this course is to explore methods for modelling and simulating large-scale neurobiological systems. By large-scale we mean that we are focusing on the properties of nervous systems and less on the interaction between their individual components (i.e., individual neurons/synapses).

Put differently, our goal is to systematize the construction of brain models – or at least the constructions of models of brain subsystems.

Beyond the fact that building brain models is the most interesting thing in the world¹, why should you be excited about this course? What are potential applications of the material we are going to discuss?

1. Figuring out how the brain works

- Health applications (better treatment of diseases).
- Philosophical implications (new ways of understanding the mind and who we are).

2. Building better AI systems

- Understanding how the brain works may help us to build better AI systems.
- “Classical AI” (structured; symbolic) ↔ “Deep Learning” (unstructured, “sub-symbolic”, “grounded”). We are discussing solutions that are somewhere in between, i.e., systems that are structured but still able to learn and use grounded symbols.

3. Programming neuromorphic hardware

- Neuromorphic hardware systems are inspired by brain-like information processing. The methods discussed in this course facilitate programming these systems [1].

1.2 Theoretical Neuroscience and Levels of Organization

The approach we are taking with respect to building models of neurobiological systems can be broadly characterized as being under the umbrella of *Theoretical Neuroscience*.

Nervous systems can be described at multiple levels of organization, each level catering to different kinds of questions. For example, analysing nervous systems at the level of individual neurons and synapses tends to answer questions about how computations are performed

¹ I will leave it up to the reader to decide whether this is a subjective statement.

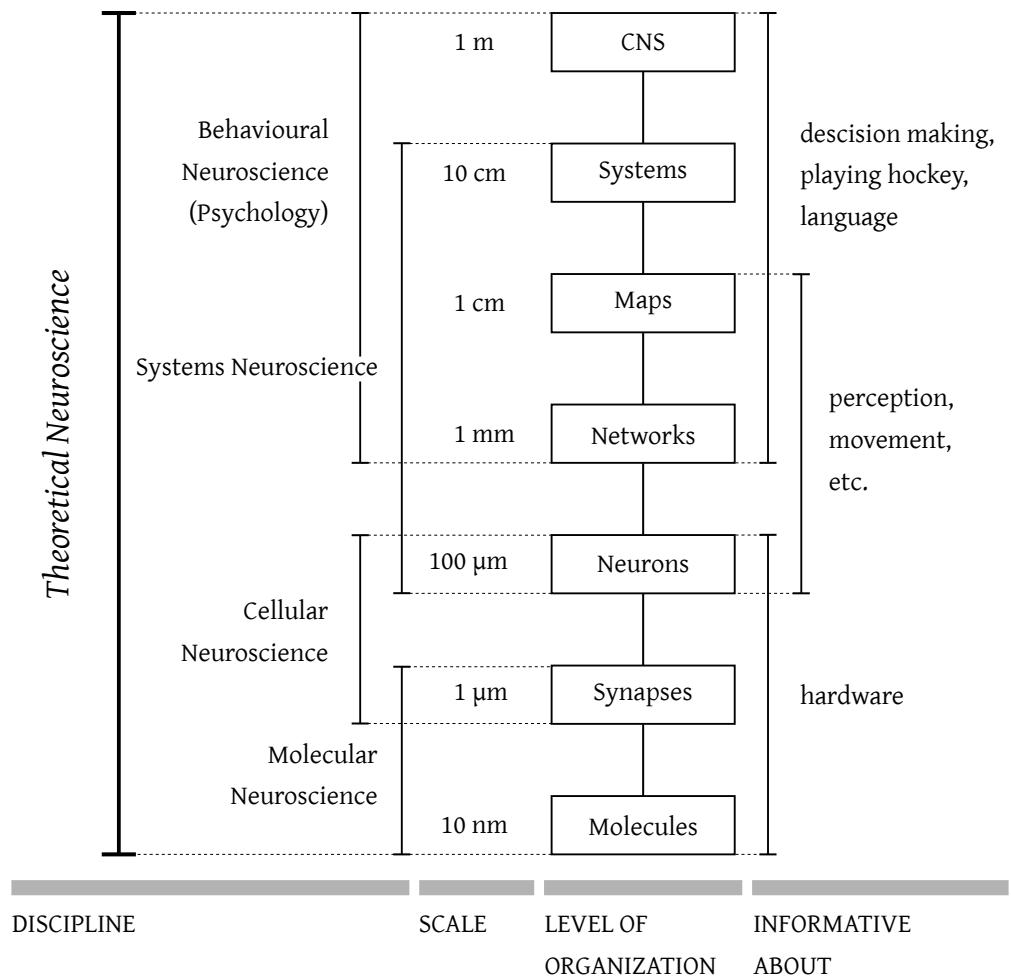


Figure 1: The scope of theoretical neuroscience and levels of organization.

in a neural substrate. Questions about motor control and perception are better answered by looking at entire networks (e.g. the retinal microcircuits for vision, or the cerebellar microcircuit for motor control). Since each of these organizational levels is interesting on its own, there are entire subdisciplines of neuroscience that focus on a narrow range of levels (fig. 1).

When trying to answer broader questions, such as “How does the mind work?”, concerning oneself with just one level of organization is no longer sufficient. We need techniques that help us bridge multiple levels of organization. Furthermore, we need some way of quantifying the progress towards our goal, as well as systematic methods that reduce the overwhelming complexity we are facing.

The aim of Theoretical Neuroscience is to work towards these goals by generating compact models that summarize what has been learned in the individual subdisciplines and that foster communication between them [2].

To frame the idea of Theoretical Neuroscience more clearly, let us consider another field of science that has been working on a problem that is *almost* as complex as trying to answer the question of “who we are”.

1.3 Theoretical Neuroscience vs. Theoretical Physics: An Analogy

Table 1: Comparison between theoretical physics and theoretical neuroscience.

	Theoretical physics	Theoretical neuroscience
Quantify phenomena	$\mathbf{F} = m\mathbf{a}$	$\dot{\mathbf{x}} = \mathbf{D}\mathbf{a}$
Summarize lots of data	motion of objects	neural representation of information
Speculative (generate hypotheses)	true for all velocities	true for all stimuli

Theoretical Neuroscience as a field can perhaps be best described by comparing it to Theoretical Physics, which is concerned with the nature of reality, or, phrased as a question, attempts to understand “What exists?”. Just as in neuroscience, physics is divided into numerous sub-fields, with particle physics at the smallest and astrophysics at the largest scale. The goal of theoretical physics is to find overarching descriptions that apply to all sub-fields – examples being conservation of energy, or (as an example of a useful, but clearly limited model) Newtons laws of motion (table 1). Both apply to celestial objects and everyday objects alike, i.e., span multiple levels of scale and organization.

To summarize, both fields are similar in that they attempt to find quantifications that summarize large corpora of data across vastly different scales using similar mathematical methods. The main difference lies in the question that is being answered (“What exists?” vs. “Who are we?”). Furthermore, since biology tends to be highly nonlinear, limiting the potential of mathematical analysis, Theoretical Neuroscience has to rely on even more numerical simulations than Theoretical Physics.

2 Neural Modelling

To make sure that a theory from Theoretical Neuroscience, or, more broadly, *any* theory about cognition, is useful, we should be able to use it to build models that make predictions about cognitive systems.



Note: Note that this implies that our theories (or “frameworks”) must be precise enough to be turned into a mathematical model – for example, many theories from Psychology do *not* pass this requirement.

Since our models will get complex quite quickly, we must most certainly use some kind of computer simulation. Correspondingly, Theoretical and Computational Neuroscience are often quite closely related.

In general, researchers often distinguish between the following kinds of models:

- **Single neuron models**

May include spikes, spatial structure, various ion channels, etc.

- **Small network models**

May either use spiking neurons, rate neurons, mean fields, etc.

- **Large network/cognitive models**

May either use biophysics, pure computation, anatomy, etc.

Ideally, we would like to have a framework that allows us to include levels of detail below any higher level as desired. Of course, the “correct” highest level depends on the question that is being asked. Our goal in this course is to be able to build models that elicit some kind of high-level behaviour, or *function* – either through motor control, or by solving cognitive tasks. We will refer to this kind of model as a “functional model”. As a consequence, we focus less on single neuron and small network models on their own, since these are often not sufficient to produce the functions we are interested in.



Note: The term *functional model* may be a little unfortunate since *functional* is quite overloaded. As described above, we use *functional* solely in the sense of “producing a behaviour” or “solving a cognitive task”.

2.1 Problems with current modelling approaches

Classically, there have been two different approaches for constructing the kind behavioural models we are interested in: “bottom-up”, large-scale neural models and “top-down” cognitive models. Both approaches have various problems that – in our opinion – disqualify them from being useful theories in the “Theoretical Neuroscience” sense defined above.



Note: We are *not* saying that the approaches described below are useless! As with any scientific project, “usefulness” depends on the question that is being asked. In the context of this course, we are trying to build “functional models” (in the sense defined above), and the problems discussed below are solely with respect to this goal.

Bottom-up, large-scale neural models Examples of this kind of approach would be (parts of) the Human Brain Project (HBP; [3]) or the DARPA SyNAPSE research program [4, 5]. Here, the general idea seems to be to collect as much information as possible about “canonical microcircuits” and to simulate these in large scale on either supercomputers or specialised neuromorphic hardware (see [6] for a more thorough discussion of the HBP).

The idea of canonical microcircuits may be attractive from a theoretical perspective. However, this seems to ignore that the brain consists of highly specialised structures. As we will discuss next, different parts of the brain are very dissimilar in terms of connectivity, cell types, as well as the type of input and output.

Furthermore, it is unclear how these projects hope to produce higher level behaviour beyond matching low-level neurophysiological data. Surely, intelligent behaviour does not just “emerge” by blindly repeating the same circuits over and over again.

Top-down cognitive models Top-down approaches can be characterized by foremost focusing on the behaviour and to a much lesser degree on the implementation of the behaviour in a neural substrate. This kind of approach is often taken in Cognitive Science and (if a computational model is provided) in Psychology. A variety of modelling frameworks exists that describe cognition in terms of simple computational building blocks that can be connected. Examples of modelling frameworks that follow this particular approach include ACT-R and Soar.

The problem with these approaches is that the models constructed in these frameworks are disconnected from neuroscience, which makes it hard to make predictions about e.g. neural activities or, on a higher level, reaction times and error rates.

Top-down modelling frameworks also have the problem of having no “bridging laws”. For example, if we were to build one part of our model in greater biological detail, it would be unclear how to build an interface between these levels of organization. One could argue that this is a bit like having rules of chemistry that never mention that “it’s all built out of atoms and electrons”.

Because of these missing “bridging laws”, it is also not really practical to constrain the model according to data from neuroscience. For example, the mathematical equations that are being used to describe computation within the model are not constrained. However, as we will see in the upcoming weeks, a neural substrate may not be particularly good at computing certain functions.

As mentioned in the above note, these restrictions are surely fine for many research questions. One might (conversely) ask whether we actually understand the brain enough to provide “bridging laws” and to accurately constrain our models. The desire to provide a framework that links high-level behaviour down to neural activities may be met with the same scepticism as a computer science student who sets out to understand how a word processor works, but ends up worrying about transistors and solid-state physics.



Note: One problem with this analogy is that in the case of computers we – the scientific community – understand all the laws that bridge between different levels of organization/abstraction. Trying to define similar laws for cognition is *exactly* what we try to do in Theoretical Neuroscience.



Aside: For more fun but (perhaps flawed) analogies between neuroscience and computer science see “Could a Neuroscientist Understand a Microprocessor?” [7].

2.2 The Brain

We ended the last section asking whether we actually know enough about brains in order to expertly constrain our models and to define bridging laws. So, what *do* we know about our brains?

Let's start with some basic facts about the human brain:²

- **Weight:** 2 kg (2% of the body weight)
- **Power consumption:** 20 W (25% of the body's total power consumption)
- **Surface area:** 1500 cm² to 2000 cm² (roughly four A4/letter pages of paper)
- **Number of neurons:** 100 billion (10^{11} , 150 000 mm⁻²)
- **Number of synapses:** 100 trillion (10^{14} , about 1000 per neuron)



Note: If we were able to simulate 1000 neurons in realtime at about 1 W (which is roughly the case for simple neuron models and specialised neuromorphic hardware, not counting any interconnect), a computer simulating the entire brain would consume about 100 MW, i.e., the power output of an entire medium-scale power plant. That is almost 7 orders of magnitude more than what biology requires.

Brain Structure As can be seen in figs. 2a and 2b, the human brain has several visually clearly distinct features. The most prominent of these structures is the Cerebral Cortex (the “folded” part at the top in fig. 2b; each peak of a “fold” is called *gyrus*, the corresponding trough is called *sulcus*). Furthermore, we can clearly see the Cerebellum (the sponge-shaped “appendix” at the back of the brain), as well as the Brain Stem (the grey region extending into the spinal cord).

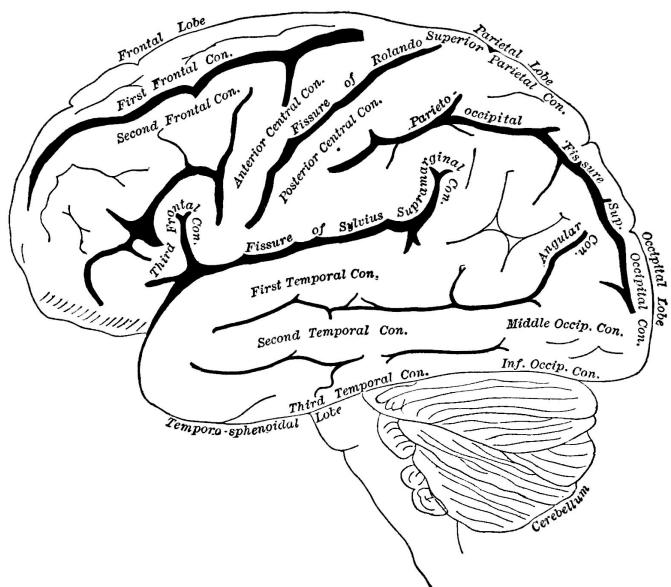
Cerebral cortex has historically been subdivided into different “lobes” (fig. 2a), particularly the “Frontal Lobe”, “Temporal Lobe” (temple ≈ lateral/side), “Occipital Lobe” (occiput ≈ lat. back of the head), “Parietal Lobe” (paries ≈ lat. “wall”; top of the head). As it turns out, these subdivisions are not only purely anatomical, but also have functional significance.

Smaller features include the locus coeruleus, thalamus, amygdala, hypothalamus, substantia nigra, etc., which are not (well) visible in the illustrations here.

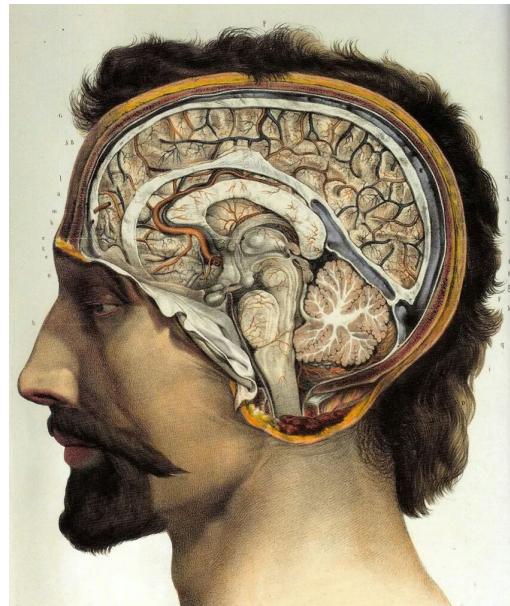
To summarize, the brain is highly structured, even at a macro-scale. We should take this structure into account when building cognitive models! Furthermore, there are many latin and greek names that you can remember (but don't feel obliged to) and impress your friends.



Note: An introductory neuroscience book [8] or a clinical neuroanatomy book (for example [9]) may be good places to start in case you want to learn more about brain structure.

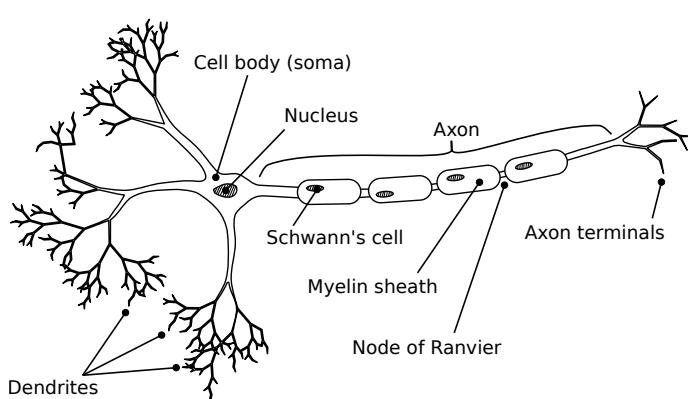


(a) Labelled lateral view of the left hemisphere

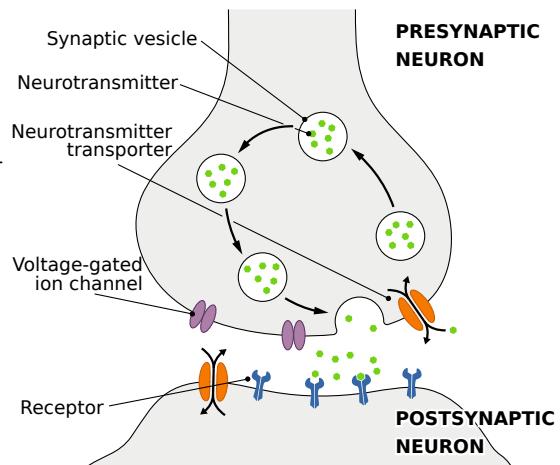


(b) Sagittal cross-section

Figure 2: Illustrations depicting the human brain. **(a)** From *Popular Science Monthly, Volume 35* (1889) via Wikimedia. **(b)** Illustration by Jean-Baptiste Marc Bourgery, *Traité complet de l'anatomie de l'homme* (1831 to 1854) via Wikimedia.



(a) Single neuron



(b) Chemical synapse

Figure 3: Neurons and Synapses. **(a)** Sketch of a schematic biological model neuron. Input spikes arrive at synapses located at the dendrites and are processed in the cell body. Resulting output spikes travel along the axon to the axon terminals, where they connect to other neurons or a neuromuscular junction. From [10] (fig. 2.3); inspired by [8] (fig. 2.2). **(b)** Diagram detailing synaptic transmission at a chemical synapse. From [10] (fig. 2.10); adapted from Wikimedia.

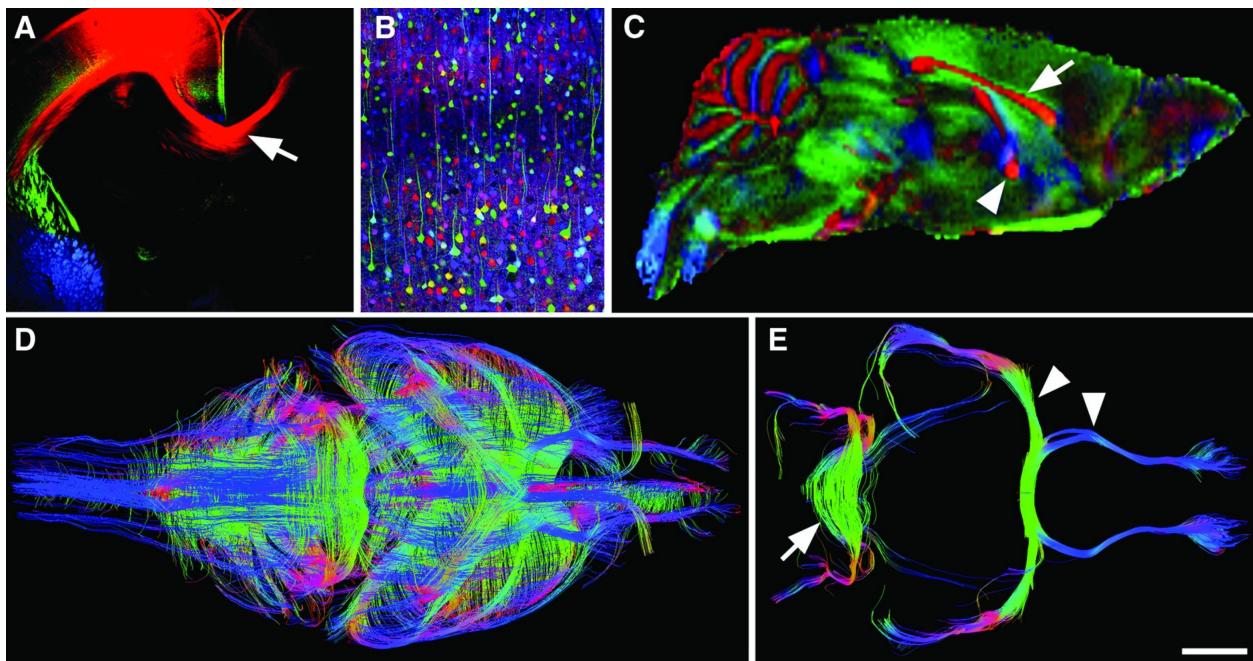


Figure 4: Panels showing different techniques for labelling axon tracts and circuits in the mammalian brain. **A** Carbocyanine dye labeling of a developing embryonic mouse brain (day 17). **B** shows a so called “brainbow” image of a mouse brain in which individual neurons are coloured in different hues. **C** Diffusion-weighted magnetic resonance image. **D, E** Tractography images of high angular resolution imaging (HARDI/q-ball). Image and captions from [11] (fig. 7).

Neurons and Synapses The smallest unit of computation in the brain is usually assumed to be a single nerve cell, or *neuron* (fig. 3a). The interfaces between neurons are called *synapses*. *Synapses* are either “chemical” (fig. 3b), or “electrical” (in some regions, e.g., retina).

In contrast to what the text-book illustration of fig. 3a might suggest, neurons come in various visually distinct shapes and are often organised in a very chaotic fashion (cf. panel **B**, fig. 4). Estimates of the number of neuron types range between 100 and 1000. Each neuron has between 1-200 000 inputs and outputs; the total axon length in the brain is estimated to be 72 km (cf. panels **D, E**, fig. 4).

Advances in scanning techniques allow researchers to reconstruct neural networks down to a nanometer scale, which means that we will have access to the “wiring diagram” of entire brain regions (see this National Geographic Video and this TEDx Talk by Jeff Lichtman).



Aside: Scales of convergence and divergence. Granule cells in the cerebellum are one of the smallest neuron type in the brain. They receive input from exactly 4-5 pre-synaptic neurons. Since more than half of all neurons in the brain are granule cells, the *median* number of receiving synapses per neuron is 4-5. Each Granule cell projects (diverges) onto 400-600 Purkinje cells. Each Purkinje cell receives input from 175 000 granule cells.

² For those of you who still find this a little too abstract, take a look at this video showing an unfixed (i.e., untreated) human brain (warning: dead human body parts). In case you'd like to experience this in real life, take PSYCH 784.

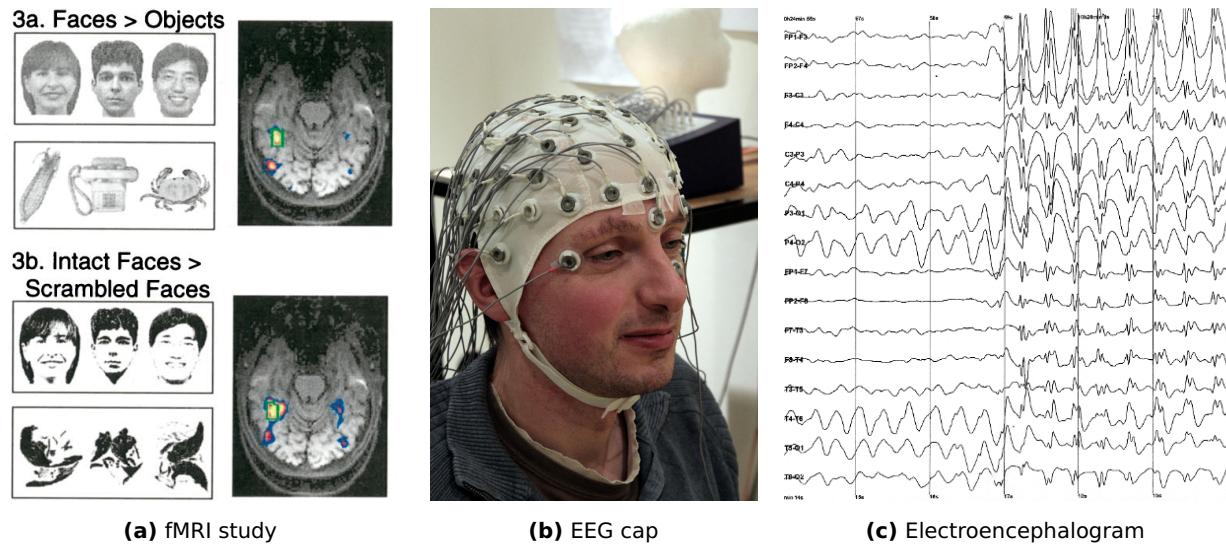


Figure 5: Non-invasive brain data. **(a)** fMRI study of the “Fusiform Face Area” in the “Fusiform Gyrus” [12] (fig. 3). **(b)** EEG cap (image from Wikimedia). **(c)** Electroencephalogram (image from Wikimedia)

2.3 Kinds of Data From the Brain

We now have a rough overview of the overall *structure* of the nervous system, both on a macro- and micro-scale. Gathering this kind of structural data is the subject of neuroanatomy and cellular neuroscience, and of course, we should use these data – for example macro- and micro-scale connectivity – to constrain our models. But apart from these structural properties, what kind of empirical data from living brains do we have access to that can be used to either further constrain our model, or to test its predictions?

2.3.1 Non-invasive

Non-invasive data sources allow us capture data from a nervous system *in vivo*, without requiring direct access to the brain and without inflicting permanent damage to the subject. Correspondingly, we are (fortunately) mostly restricted to this kind of data when studying human brains.

fMRI *Functional magnetic resonance imaging* is a measure of whole-brain activity based on changes in blood flow. Often this is based on the so called blood-oxygen-level dependent (BOLD) contrast (fig. 5a). A catalogue of data can be found at <https://neurosynth.org/>.

- + Whole-brain, 3D reconstruction (individual activity voxels, volume elements)
- Medium spatial resolution (millimeters)
- Low temporal resolution (seconds)
- Signal is hard to interpret (differences, indirect, i.e., does *not* measure spiking)
- Has to be averaged over multiple trials

EEG *Electroencephalography* measures the electric activity at the scalp using an array of electrodes (figs. 5b and 5c). Being isolated from the brain through several layers of bone and tissue, this method can at most capture large-scale communication between areas.

- + High time resolution
- Low spatial resolution
- Orange circle Relatively cheap
- Orange circle Artefacts (eye movement, swallowing)

2.3.2 Invasive

In contrast to non-invasive data sources, invasive data sources require direct access to the brain and may result in permanent damage.

Lesion Studies Here the question is what effect damaging a certain part of the brain has. This may either be part of an experiment (in animals) or the result of an accident/disease (in humans). For example, lesion of the occipital cortex leads to “blindsight” (blindness where some visual information is still unconsciously present), lesion of the inferior frontal gyrus results in loss of the ability to speak (Broca’s area), lesion of the posterior superior temporal gyrus results in loss of the ability to understand speech (Wernicke’s area), lesion of the fusiform gyrus prevents recognition of faces and other visually complex objects, and lesion of the medial prefrontal cortex may affect moral judgment (controversial; see: Phineas Gage).

- + Informative about the functional relevance of an area
- Often permanently damaging (though techniques for reversible lesions exist)

Single cell recording Here, electrodes (one or many) are placed in the brain. The electrode might not necessarily be placed right at (or in) a neuron, so it will pick up the *local field potential* (LFP; a superposition of neural activity from multiple cells). It is common to amplify the recorded signal and to send it to a speaker, which makes spikes audible (Video).

- + High temporal resolution (microseconds)
- + High specificity (single or few neurons)
- Limited to a few cells
- Damaging over time

Multi-electrode recordings Here a so called “tetrode” (an electrode with multiple tips) or a “Microelectrode Array” (MEA; “Utah Array”) is inserted into the brain (Video).

- + High temporal resolution (microseconds)
- Orange circle Up to ≈ 100 cells with one array
- Orange circle Requires post-processing (e.g., extraction of individual neurons from LFPs)
- Damaging over time

Calcium Imaging Use fluorescent calcium indicator to indicate the presence of Ca^{2+} ions inside a neuron, which is an indicator for neuronal activity. Calcium indicators can either be chemical (i.e. an externally added substance) or produced by the cell through genetic modification ( Video: Fish Embryo,  Video: Stalking Fish).

- + High temporal resolution (milliseconds) - Local
- + High spatial resolution (individual cells) - Invasive

Optogenetics Modify individual types of neurons genetically, so they can either be stimulated or inhibited through light ( Video).

- + High temporal resolution (milliseconds) - Local
- + Targets individual cell types
- + Can examine function of brain circuits - Invasive

2.4 Summary

To summarize, we have access to a vast amount of data describing the brain at both the macro- and micro-scale, and both in terms of its anatomy (i.e., structure) and physiology (i.e., mechanisms within the living system).

Scientists have constructed neural models at the scale of single detailed individual neurons, as well as millions ( HBP Video) and billions ( SyNAPSE Video) of neurons. Still, we don't have good theories that connect behaviour and cognition to low-level neural data. In the words of Churchland and Sejnowski (1994), neuroscience is "data-rich and theory-poor". This is still true today.



Note: Problematically, neuroscientists do not agree as for what the right level of detail of neural models is. When the SyNAPSE project led by IBM's Dharmendra S. Modha announced the simulation of a "cat-scale" brain with 1 billion neurons in 2009, the founder of the Human Brain Project (HBP), Henry Markram, denounced the SyNAPSE project as a "hoax and PR stunt" and called for Modha to be "strung up by the toes" [13].

Markham's rage did not – as one might expect – stem from the fact that IBM's "cat brain" did not do what one would expect from a functional brain (controlling behaviour), but merely from the fact that IBM's neurons were "too simple" and therefore cannot possibly constitute a full brain model. Markham's HBP and its predecessor, the BlueBrain project, make use of physiologically and anatomically extremely detailed neuron models that are statistically connected. Yet, as of 2020, the HBP has similarly failed to produce any "functional" brain model.

Of course, this leaves that question as for what the right level of detail is to be counted as a "brain model". Perhaps we should, as we have discussed above, focus on what actually

matters – connecting neural models to high-level behaviour.

3 The Neural Engineering Framework and the Semantic Pointer Architecture

Now that we have spent some time discussing other approaches aiming at large-scale modelling of neurobiological systems and pointed out their potential shortcomings, it is time to introduce the approach at neural modelling we are going to discuss in this course. In particular, we are going to have a look at two different modelling techniques: the *Neural Engineering Framework*, and the *Semantic Pointer Architecture*.

3.1 Neural Engineering Framework

The *Neural Engineering Framework* (NEF) has been proposed by Eliasmith and Anderson in their book “Neural Engineering” in 2003 [14]. We will spend the first half of this course on discussing the NEF.

The goal of the Neural Engineering framework is to treat the brain as a purely physical system and to use techniques from engineering – such as control theory, information theory, and signal processing theory – to aid a *functional* understanding of the brain in theoretical neuroscience. After all, building physical systems that perform a certain function and – to this end – *understanding* physical systems in terms of their potential function lies at the heart of engineering practice.³ Of course, a major difference lies in the fundamental components usually employed by nature and, conversely, by engineers. Natural systems have evolved, and their individual parts, such as neurons, are thus inherently diverse and “messy”. Engineers on the other hand often rely on precisely manufactured parts. So when applying engineering methods to neural modelling, we somehow have to take this messiness into account. As we will see later, we can actually use the “messiness” of neural systems to our advantage when building models.

The Neural Engineering Framework provides a systematic way to translate a vector-valued dynamical system into a spiking neural network. In other words, the NEF can be thought of as a *neural compiler*, that turns a mathematical description into a spiking neural network. Most importantly, this translation process can be informed by the vast corpus of neuroscientific data collected over the past decades. To put it differently: given a mathematical model, and a set of neurobiological constraints, including neural tuning, neuron count, connectivity, and time constants, we can create a neural network that fulfils these constraints and approximates the mathematical model.

³ This is akin to Richard Feynman’s famous chalkboard note “What I cannot create, I do not understand.” [15]

3.2 Semantic Pointer Architecture

The second technique – which we are going to discuss, among other topics, during the second half of the course – is the *Semantic Pointer Architecture* (SPA). The SPA is a mathematical tool for building models of high level cognition. It is based on what's commonly referred to as a *Vector Symbolic Architecture* (VSA) and has been described in the book *How to Build a Brain* by Eliasmith published in 2013 [16].

VSA provide a way to implement discrete symbolic architectures as they have been used in classical artificial intelligence research (“Good Old Fashioned AI”; GOFAI) on top of continuous vector spaces. The SPA in particular describes how such “symbol vectors”, also called “semantic pointers”, can be generated from sensory input, combined – or bound to – other semantic pointers, and decompressed back into either the sensory or motor domain.

3.3 The NEF and SPA as tools for hypothesis generation

While the NEF and SPA are independent concepts, they together provide a powerful tool for scientific exploration in theoretical neuroscience and cognitive science. In particular, the NEF provides a way to represent vector spaces in a neural substrate, and, conversely, the SPA provides a way to implement a cognitive architecture on top of vector spaces. In a sense, the NEF describes the computational substrate, whereas the SPA can make use of – but is not limited to – this substrate. Looking back at fig. 1 we see that the NEF and SPA together model biological systems from synapses and neurons up to entire central nervous systems. So far, no other modelling approach can achieve this. We will have a look at some examples of models that have been built using these methods in section 5.

Of course, merely using these techniques does not magically mean that the models we build are good models of human cognition. In fact, depending on the kind of question we are trying to answer, our models are most definitively wrong, but that is just the nature of models. The true strength of any modelling technique is in aiding *hypothesis generation*.

For example, in the context of the NEF and the SPA, we might have an hypothesis as for how a certain cognitive task could be solved in terms of the SPA. We furthermore know from empirical research which brain regions are involved in solving this task, and as such have data from neuroscience. We can now use this data as constraints when mapping the model onto the brain using the NEF. The resulting spiking neural network model can then be simulated and probed in various ways. This allows us (a) to *verify* our hypothesis by comparing high-level behaviour data (e.g., timings and error rates) between human/animal subjects and our model and (b) to make *predictions* about what we should be able to measure in the case of experiments that have not been performed on real subjects.

3.4 The Nengo Simulator

In the end, the NEF and the SPA are just mathematical concepts. In order to aid scientists in pursuing the kind of hypothesis-driven research outlined above, we need tools that facilitate the construction of such models. The “nengo” (originally short for “Neural Engineering Objects”) neural network simulation package is such a tool. It not only provides convenient interfaces for building NEF and SPA models (as well as other kinds of artificial neural networks), but can also execute the resulting model on a variety of hard- and software-backends such as GPUs and analogue and digital neuromorphic hardware.

nengo was originally developed by members of the Computational Neuroscience Research Group (CNRG) at the University of Waterloo [17]. It is now developed by a company called ABR (Applied Brain Research). nengo is available free-of-charge for non-commercial use. It can be found at <https://nengo.ai> and <https://github.com/nengo>.

 **Note:** As part of the assignments we are going to write our own software implementation of the NEF. Correspondingly, for the sake of learning the underlying theory, you cannot use nengo at first. However, you are asked to use nengo for the last assignment, and are free to use it for the final project. So it definitely doesn't hurt to have a look at it.

4 Representation, Transformation, and Dynamics

The Neural Engineering Framework is based on three principles: Representation, Transformation, and Dynamics. We will have a closer look at each of these principles in upcoming lectures, but for now, let's quickly summarize what each of these principles is about.

4.1 Representation

In order to solve complex tasks, nervous systems must construct *representations* of their environment. A reasonable definition of *representation* would be to say that a neuron *represents* some aspect of the environment, if this neuron is active (or conversely, exactly not active) whenever a certain stimulus is present. In other words, if there is a causal relationship between the stimulus and the neural activity. However, in general, the question of how exactly neurons represent information is still open.

In accordance with the above, we think of representations in the NEF as the mapping between a *value* that is being processed in the network and the corresponding *activity* of a *group* of neurons. Mathematically speaking, the neural activity can be seen as a vector $\mathbf{a}(t)$ in a high-dimensional vector space, whereas the corresponding value $\mathbf{x}(t)$ may be situated in a lower-dimensional vector space.

For this concept of representation to be useful, we need to define a way to *decode* the represented value from the neural activity, and also a way to *encode* a value as neural activities.

The NEF defines the decoding process in terms of a linear decoder, and the encoding process in terms of a nonlinear encoding.

Note that we talked about representations in terms of the activity of a *population* – also referred to as an *ensemble* – of neurons, and not in terms of the activity of individual neurons. This is an important difference that separates the NEF from many “classical” modelling approaches in neuroscience. The NEF focuses on neuron populations as the smallest representational unit, and not individual neurons.

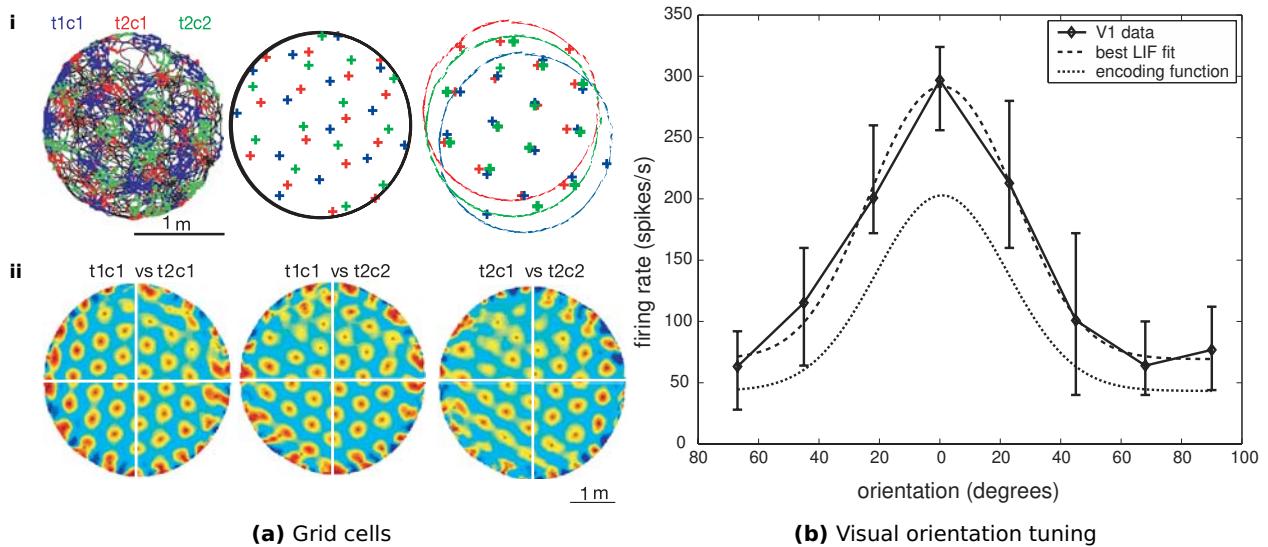


Figure 6: Examples of neural representation. **(a)** Grid cells. Top left: trajectory of the experimental animal while moving through space. Colours (ref, blue, green) correspond to the activity of three cells recorded from the dorsocaudal medial entorhinal cortex (dMEC). Top centre: peak activity locations for each cell. Top right: the individual cell peak activities are shifted to coincide with each other, highlighting the repeating structure in each cell. Bottom: cross-correlations between the individual cell activities. Figure copied from [19], fig. 3. **(b)** Example of visual orientation tuning of a cell in primary visual cortex of a macaque monkey. Figure copied from [14], fig. 3.1.



Example: Famous examples of neural representation are place- and grid-cells. Here neural activity is correlated with the animal being at a certain location (place cells, [18]), or within a certain patch of a multi-scale hexagon grid (grid cells, [19], see fig. 6a). We are going to revisit grid cells when we talk about spatial semantic pointers. Another, more classical example is orientation tuning in primary visual cortex. Here, cells represent the orientation of line-shaped features in the visual field. See fig. 6b.



Note: When talking about representations in this course, we generally refer to “transient” activities of the network, i.e., relatively short-lived signals. Of course, longer-lived states – such as *long-term memories* – are also represented in the brain somehow, but we will talk about this separately when discussing learning.



Note: Having to think about representation in terms of vector spaces, may seem to be somewhat limiting. However, as we will see throughout the course, vectors are a very general mathematical construct that can be used to represent all kinds of things, including symbols, (bandlimited) functions over space and time, probability distributions, etc. If you’d like to, take a look at Chapter 3 in [14], which discusses function representation (we

are most likely not going to discuss this particular chapter in detail; though we are going to discuss this concept in the context of representing functions over time).

4.2 Transformation

Of course, merely being able to represent values within biologically plausible neural networks is not sufficient to explain complex behaviours. To this end, we need some notion of *computation*, i.e., the ability to *transform* a represented value \mathbf{x} using a function $f(\mathbf{x})$, and to store the resulting value \mathbf{y} in another neural population.

In the NEF, we compute functions by choosing the “right” connection weights between two populations. In particular, since we have information about how the represented value $\mathbf{x} \in \mathbb{R}^m$ can be *decoded* from, let’s say, a source population A , and how values $\mathbf{y} \in \mathbb{R}^n$ are *encoded* in a target population B , we can compute optimal connection weights \mathbf{W}^f that realize a function $f : \mathbb{R}^m \rightarrow \mathbb{R}^n$.

 **Note:** The way in which we normally choose connection weights in the NEF is fundamentally different from popular approaches for training artificial neural networks, such as *backpropagation*. Since we first think about what the individual neuron populations (comparable to a “layer” of neurons) *represent*, we can optimally solve for connection weights on a connection-by-connection basis.

When “training” a network using back-propagation, the training procedure alters the connection weights across multiple layers simultaneously, hence changing what is being represented by individual neurons/populations of neurons. This makes back-propagation more powerful, since we don’t have to know what the right representation and transformation is supposed to be. However, it also makes “training” inherently less interpretable, much slower, and less robust (with respect to hyperparameters). In other words, in the context of modelling neurobiological systems, there is no reason why we should use something such as backpropagation, as long as we have a good idea as for what the values being represented are, and which functions have to be computed to get from one representation to the next.

That being said, the NEF can be combined with backpropagation-like online-learning approaches in order to build systems that *learn* over time. Again, we will talk about that during the second half of the course when we discuss learning.

4.3 Dynamics

So far, the transformations we have talked process information in a *feed-forward* fashion. However, most brain areas make heavy use of *recurrent* (“backwards projecting”, or “cyclical”) connections. As it turns out, we can systematically use recurrent connections to implement

arbitrary dynamical systems, i.e., equations of the form

$$\frac{d}{dt} \mathbf{x}(t) = f(\mathbf{x}(t), \mathbf{u}(t)),$$

where $f : \mathbb{R}^m \times \mathbb{R}^n \rightarrow \mathbb{R}^m$ defines the differential equation, \mathbf{x} is the current system state, and \mathbf{u} is the input to the differential equation. Comparably to the feed-forward connections discussed above, we can easily find optimal recurrent connection weights \mathbf{W}^f that approximate a certain dynamical system.

This is particularly useful for modelling phenomena such as (short-term) memory. For example, the following differential equation describing an *integrator*

$$\frac{d}{dt} \mathbf{x}(t) = \mathbf{u}(t),$$

can be used to “remember” an input \mathbf{u} across time. Being able to implement dynamical systems allows us to incorporate approaches from control-theory into our models. This is helpful when trying to build motor control systems, but also for inference related tasks (cf. “filters”, “observers” in control theory).

The ability to implement models with arbitrary dynamics is a central feature that sets the NEF apart from other modelling techniques. For example, the NEF provides a systematic approach for constructing stable attractor networks, which have been hypothesised to be an important building block of nervous systems for decades [20].



Note: If time permits, we are going to discuss some newer, alternative approaches for building dynamical systems in a spiking neural substrate, such as the FORCE [21, 22] (an online dynamics learning approach) and EBN [23, 24] (a method focused on efficient information transmission between neurons) at the very end of this term.

5 Examples

As mentioned above, the NEF can be seen as a neural compiler. Given a quantitative description of a behaviour (e.g. an algorithm), we can solve for the connections between neurons that will approximate that behaviour. This works for a wide variety of neuron models, both detailed and simple neuron models (next lecture). In particular, as one would expect in real biological systems, the number of neurons affects the model’s accuracy, while the neuron properties influence timing and computation. This means that we can make predictions about the behaviour of a system.

The following videos show some examples of the NEF and the SPA in action:

- Vision: character recognition
- Vision: >1000 categories
- Problem solving: Tower of Hanoi
- Spaun: digit recognition
- Spaun: copy drawing
- Spaun: addition by counting
- Spaun: pattern completion

6 Conclusion

To summarize, the approaches we are going to discuss in this lecture – the Neural Engineering Framework (NEF) and Semantic Pointer Architecture (SPA) being among them – allow us to model neurobiological systems at a scale unparalleled by competing approaches.

These techniques are particularly useful as a new way to test theories under neurological constraints, sometimes suggesting novel algorithms for solving problems in Artificial Intelligence (cf. the examples above). Having a framework for scientific exploration of large-scale brain models facilitates gaining a better understanding of the brain, potentially leading to new medical applications, and an understanding of the mind and who we are.

References

- [1] Kwabena Boahen. "A Neuromorph's Prospectus". In: *Computing in Science Engineering* 19.2 (Mar. 2017), pp. 14–28. ISSN: 1521-9615. DOI: 10.1109/MCSE.2017.33.
- [2] Laurence F. Abbott and Peter Dayan. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. Computational Neuroscience. MIT Press, Oct. 2001. 480 pp. ISBN: 978-0-262-04199-7. URL: <https://mitpress.mit.edu/books/theoretical-neuroscience>.
- [3] Henry Markram. "The Human Brain Project". In: *Scientific American* 306.6 (2012), pp. 50–55.
- [4] Timothy P. Morgan. "DARPA Shells out \$21m for IBM Cat Brain Chip". In: *The Register. Science* (Aug. 18, 2011). URL: https://www.theregister.co.uk/2011/08/18/ibm_darpa_synapse_project/ (visited on 01/03/2020).
- [5] Paul A. Merolla et al. "A Million Spiking-Neuron Integrated Circuit with a Scalable Communication Network and Interface". In: *Science* 345.6197 (2014), pp. 668–673. ISSN: 0036-8075. DOI: 10.1126/science.1254642. eprint: <https://science.sciencemag.org/content/345/6197/668.full.pdf>. URL: <https://science.sciencemag.org/content/345/6197/668>.
- [6] Brent Komer and Chris Eliasmith. "A Unified Theoretical Approach for Biological Cognition and Learning". In: *Current Opinion in Behavioral Sciences* 11 (2016). Computational modeling, pp. 14–20. ISSN: 2352-1546. DOI: <https://doi.org/10.1016/j.cobeha.2016.03.006>. URL: <http://www.sciencedirect.com/science/article/pii/S2352154616300651>.
- [7] Eric Jonas and Konrad Paul Kording. "Could a Neuroscientist Understand a Microprocessor?" In: *PLOS Computational Biology* 13.1 (Jan. 2017), pp. 1–24. DOI: 10.1371/journal.pcbi.1005268. URL: <https://doi.org/10.1371/journal.pcbi.1005268>.
- [8] E. Kandel et al. *Principles of Neural Science*. 5th ed. McGraw-Hill Education, 2012.
- [9] Ryan Splitgerber. *Snell's Clinical Neuroanatomy*. 8th Edition. Nov. 6, 2018. 560 pp.

- [10] Andreas Stöckel. "Design Space Exploration of Associative Memories Using Spiking Neurons with Respect to Neuromorphic Hardware Implementations". MA thesis. Germany: Bielefeld University, 2015.
- [11] Alain Chédotal and Linda J Richards. "Wiring the Brain: The Biology of Neuronal Guidance". In: *Cold Spring Harbor perspectives in biology* 2.6 (June 2010), a001917–a001917. ISSN: 1943-0264. DOI: 10.1101/cshperspect.a001917. PMID: 20463002. URL: <https://www.ncbi.nlm.nih.gov/pubmed/20463002>.
- [12] Nancy Kanwisher, Josh McDermott, and Marvin M. Chun. "The Fusiform Face Area: A Module in Human Extrastriate Cortex Specialized for Face Perception". In: *Journal of Neuroscience* 17.11 (1997), pp. 4302–4311. ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.17-11-04302.1997. eprint: <https://www.jneurosci.org/content/17/11/4302.full.pdf>. URL: <https://www.jneurosci.org/content/17/11/4302>.
- [13] Sally Adey. *Cat Fight Brews Over Cat Brain*. Nov. 23, 2009. URL: <https://spectrum.ieee.org/tech-talk/semiconductors/devices/blue-brain-project-leader-angry-about-cat-brain> (visited on 01/04/2020).
- [14] Chris Eliasmith and Charles H. Anderson. *Neural Engineering: Computation, Representation, and Dynamics in Neurobiological Systems*. Cambridge, Massachusetts: MIT Press, 2003. 380 pp. ISBN: 978-0-262-55060-4.
- [15] Richard Feynman. *Richard Feynman's Blackboard at Time of His Death*. 1988. URL: <http://archives-dc.library.caltech.edu/islandora/object/ct1:483> (visited on 12/30/2019).
- [16] Chris Eliasmith. *How to Build a Brain: A Neural Architecture for Biological Cognition*. Oxford Series on Cognitive Models and Architectures. New York, New York: Oxford University Press, 2013. 456 pp. ISBN: 978-0-19-026212-9.
- [17] Trevor Bekolay et al. "Nengo: A Python Tool for Building Large-Scale Functional Brain Models". In: *Frontiers in Neuroinformatics* 7.48 (2014).
- [18] John O'Keefe and Lynn Nadel. *The Hippocampus as a Cognitive Map*. Oxford, United Kingdom: Oxford University Press, 1978. ISBN: 0-19-857206-9. URL: <http://cognitivemap.net/>.
- [19] Torkel Hafting et al. "Microstructure of a Spatial Map in the Entorhinal Cortex". In: *Nature* 436.7052 (Aug. 1, 2005), pp. 801–806. ISSN: 1476-4687. DOI: 10.1038/nature03721. URL: <https://doi.org/10.1038/nature03721>.
- [20] Chris Eliasmith. "A Unified Approach to Building and Controlling Spiking Attractor Networks". In: *Neural computation* 7.6 (2005), pp. 1276–1314.
- [21] David Sussillo and L.F. Abbott. "Generating Coherent Patterns of Activity from Chaotic Neural Networks". In: *Neuron* 63.4 (2009), pp. 544–557. ISSN: 0896-6273. DOI: <https://doi.org/10.1016/j.neuron.2009.07.018>. URL: <http://www.sciencedirect.com/science/article/pii/S0896627309005479>.

- [22] Wilten Nicola and Claudia Clopath. "Supervised Learning in Spiking Neural Networks with FORCE Training". In: *Nature Communications* 8.1 (Dec. 20, 2017), p. 2208. ISSN: 2041-1723. DOI: 10.1038/s41467-017-01827-3. URL: <https://doi.org/10.1038/s41467-017-01827-3>.
- [23] Martin Boerlin and Sophie Denève. "Spike-Based Population Coding and Working Memory". In: *PLOS Computational Biology* 7.2 (Feb. 2011), pp. 1–18. DOI: 10.1371/journal.pcbi.1001080. URL: <https://doi.org/10.1371/journal.pcbi.1001080>.
- [24] Martin Boerlin, Christian K. Machens, and Sophie Denève. "Predictive Coding of Dynamical Variables in Balanced Spiking Networks". In: *PLOS Computational Biology* 9.11 (Nov. 2013), pp. 1–16. DOI: 10.1371/journal.pcbi.1003258. URL: <https://doi.org/10.1371/journal.pcbi.1003258>.