UNIVERSITY OF MESSINA

DEPT. OF COGNITIVE SCIENCE, PSYCHOLOGY, EDUCATION AND CULTURAL STUDIES
GRADUATE STUDIES IN COGNITIVE SCIENCE - XXXIII COURSE

Neural Models of Contextual Semantic Disambiguation

Arianna Maria Pavone

A DISSERTATION
SUBMITTED TO DEPARTMENT OF COGNITIVE SCIENCE,
PSYCHOLOGY, EDUCATION AND CULTURAL STUDIES
AND THE COMMITTEE ON GRADUATE STUDIES
OF UNIVERSITY OF MESSINA
IN FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY IN COGNITIVE SCIENCE

DIRECTOR OF GRADUATE STUDIES
Prof. Alessandra Falzone

ADVISOR
Prof. Alessio Plebe

I certify that I have read this dissertation and that, in my opinion, it is fully adeguate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Prof. Alessandra Falzone
(Director of Graduate Studies)

I certify that I have read this dissertation and that, in my opinion, it is fully adeguate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Prof. Alessio Plebe
(Advisor)

Approved for the University Committe on Graduate Studies.

Preface

In human language several ambiguities cannot be resolved without simultaneously reasoning about an associated context. Often, the context can be best catpured from the visual scene referred by the sentence. If we consider the sentence "I take a photograph of a chimpanzee in my pajamas", looking at language alone, it is unclear if it is the person or the chimpanzee wearing the pajamas.

In this dissertation we focus on the contextual effects on semantics: on the one hand we investigate such contextual effects on a disambiguation task using neural computational simulation; on the other hand we propose a novel context sensitive cognitive account of similarity.

Going a little more in detail in our disambiguation task, provided with a sentence, admitting two or more candidate interpretations, and an image that depicts the content of the sentence, it is required to choose the correct interpretation of the sentence depending on the image's content. Thus we address the problem of selecting the interpretation of an ambiguous sentence matching the content of a given image.

This type of inference is frequently called for in human communication that occurs in a visual environment, and is crucial for language acquisition, when much of the linguistic content refers to the visual surroundings of the child [8, 11].

This kind of task is also fundamental to the problem of grounding vision in language, by focusing on phenomena of linguistic ambiguity, which are prevalent in language, but typically overlooked when using language as a medium for expressing understanding of visual content. Due to such ambiguities, a superficially appropriate description of a vi-

sual scene may in fact not be sufficient for demonstrating a correct understanding of the relevant visual content.

Regarding our new contextual account of similarity, we will suggest that most of the traditional similarity models which have been proposed over the years can converge on a generalized model of similarity in which the context plays a fundamental role in order to overcome all the criticisms raised over the years to each of the traditional similarity models.

From the neurocomputational point of view, our models are based on the Eliasmith's Neural Engineering Network (NEF) [27] and Nengo[1], the python library which serves as an implementation of the NEF. The basic semantic component within NEF is the so-called Semantic Pointer Architecture (SPA) [129], which determines how the concepts are represented as dymanic neural assemblies.

[1]Nengo is available at `https://www.nengo.ai`

Table of Contents

Introduction

In recent years, a number of different disciplines have begun to investigate the fundamental role that context plays in different cognitive phenomena. The problem of context spans from the abstract level of semantics down to the level of neural representations. It has increasingly been studied also for its role in influencing mental concepts and, more specifically, linguistic communication has been the area of study that has traditionally explored these issues.

The term context is not easy to define: it is something that cannot be specified independently of a specific frame and it may play quite different roles within alternative research paradigms. In wider terms, as stated by [38], we can define the context as a "frame that surrounds the event and provides resources for its appropriate interpretation". However, in order to obtain a more complete understanding of what context stands for, it is necessary to investigate how it interacts with cognitive phenomena at three different levels: the linguistic, the cognitive and the neural level [99].

There is a long tradition in linguistics and pragmatics which invokes context to help account for aspects of meaning in language that go beyond the scope of semantics. The main elements of context has roots that dates back to the past, and regards the degree to which truth-functional semantics depends on context. Gottlob Frege raised the point in his uncompleted 1897 volume Logik, and though he was not explicitly using the term context, he underlined how for many expressions, fixing their truth value requires supplemental information, coming from the circumstances, the "frames", in which such expressions are pronounced.

The first clear elucidation of the dependence of language on context was proposed by [120]. He takes up Frege's idea that a word has meaning only if related to the meaning of the whole sentence and if its meaning is perceived by both interlocutors, speaker and listener.

At a cognitive level the issue regards concepts and the degree to which they are de-

pendent on context. [7] has been one of the first to underline how difficult it is to conceive them as stable subjective entities, while it appears more appropriate to think of categories as dynamically constructed and tailored to specific contexts, or as *ad hoc* categories. A recent review of studies of the cognitive perspective on the linguistic issue of context can be found in [2]. Just as in the strictly linguistic domain, they find in the wider cognitive view a variety of positions, some that minimize the destabilizing effect context has on concepts, such as that of [71], or others that assume a more intermediate position such as that of [82], that while acknowledging the fundamental role context might play in concepts, sustain that a characterizing stable nucleus of mental concepts is also a part.

From the neuroscientific point of view, the context has started to have a certain relevance and the issue regards whether the activation of neurons, in response to the same stimuli, can significantly change due to contextual factors, with increasing studies on the functioning of the hippocampal region, an area traditionally considered as being a site of contextual effects processing. In particular, signs of environmental context are represented in the place cell, and can influence discrimination decisions between visual stimuli and emotional fear responses. Neurons in area CA1 (primary area of the Cornu Ammonis) instead codify contextual overlap of a temporal nature, so that memories of events taking place in brief temporal sequences subsequently have facilitated reciprocal re-activation. Cognitive neuroscience is now starting to consider in a systematic way how context interacts with neural responses [126]. The way context drives language comprehension depends on the effects of context on the conceptual scaffolding of the listener, which in turn, is the result of his neural responses in combination to context.

The kind of ambiguity addressed in this work is the canonical case of structural ambiguity, technically known as Prepositional Phrase Attachment, where a sentence includes a prepositional phrase that can be attached to more than one higher level phrases [50]. The attachment resolution is context dependent, we deal specifically with the case when depends on the visual context.

In addressing the analysis of contextual effects with a neuro-computational approach, a preliminary question concerns the optimal way to encode concepts in clusters of activation vectors. A theoretical criterion to ascertain that the chosen coding is effective,

and cognitively plausible, is to verify how the coding behaves in front of one of the main relationships that exists between concepts: semantic similarity.

The fact that human assessments of similarity are fundamental to cognition is a widely shared opinion. Similarity, is fundamental for learning, knowledge and thought, since only our sense of similarity allows us to order things into kinds so that these can function as stimulus meanings. Reasonable expectation depends on the similarity of circumstances and on our tendency to expect that similar causes will have similar effects.

On the other hand the context plays a fundamental role in the perception of the similarity between two different concepts: depending on the context in which they are compared, two entities can appear more or less similar to each other.

For the reasons reported above, although not central to the main problem of this work, a specific study on this aspect was therefore included in this thesis, proposing a context sensitive similarity model based on SPA.

This dissertation is organized as follows. In Chapter 1 we introduce the Neural Engineering Framework (NEF), a general methodology that allows you to build large-scale, biologically plausible, neural models of cognition. It acts as a neural compiler: you specify the properties of the neurons, the values to be represented, and the functions to be computed, and it solves for the connection weights between components that will perform the desired functions. Importantly, this works not only for feed-forward computations, but recurrent connections as well. It incorporates realistic local error-driven learning rules, allowing for online adaptation and optimization of responses. In Chapter 1 we also present Nengo, a Python library that allows the construction and simulation of large-scale brain models using the NEF method.

In Chapter 3 we schematically discuss the information flow and interactions among the main brain regions involved in contextual behavior, as emerged by recent advancement in neurosciences and present a circuit for simplified model of the CA3 hippocampal region involved in the retrieval of semantic relations between two items (item-item association) and between an item and a context (item-context association).

In Chapter 4 we focus on the contextual effects of visual scenes on semantics, investigated using neural computational simulation. Specifically we address the problem of

selecting the interpretation of sentences with an ambiguous prepositional phrase, matching the context provided by visual perception. More formally, provided with a sentence, admitting two or more candidate resolutions for a prepositional phrase attachment, and an image that depicts the content of the sentence, it is required to choose the correct resolution depending on the image's content. We evaluated the ability of our model in resolving linguistic ambiguities on the LAVA (Language and Vision Ambiguities) dataset, a corpus of sentences with a wide range of ambiguities, associated with visual scenes.

Finally in Chapter 5 we propose a context sensitive plausible model of semantic similarity, according with the main guidelines of the other traditional models known in literature, based on the Semantic Pointer Architecture which is at the basis of the NEF. We show how our new proposed model is able to give a solution to traditional criticisms against traditional models of similarity.

1

A Framework for Neural Semantics

A first critical choice is the identification of a suitable neural framework to be adopted all along this work. The two main requirements we seek is the biological plausibility and the possibility of modeling at a level enough abstract to deal with full images and with words in sentences. The two requirements are clearly in stark contrast to each other. On the extreme of biological plausibility we find neural simulators describing the current flow in neural compartments, like NEURON [43] and GENESIS [13], used for realistic simulations of small volumes of the brain, as done in the European Brain Project [78]. There is no hope of investigating cognitive phenomena with such models. On the other extreme of high level of abstraction and aptitude towards the investigation of cognitive phenomena there is the tradition of so-called *connectionism* [117]. These artificial neural models become widespread around the 1990s in cognitive science, especially in the psychology of language development [29, 72], but their application was limited to highly simplified small experiments, using toy visual and linguistic stimuli.

Today the legacy of connectionism has been taken up by the family of algorithms collected under the name *deep learning*. Unlike the former artificial neural networks, deep learning models succeeds in highly complex cognitive tasks, reaching even human-like performances in some visual tasks [137]. However, the level of biological plausibility

of deep learning algorithms is in general even lower than in connectionism, these models were developed with engineering goals in mind, and exploring cognition is not in the agenda of this research community [100]. In our model we will also include a very simple deep learning component, but only for the low-level analysis of the images. This choice makes the model simpler, by exploiting the ease of deep learning model in processing visual stimuli. It would have been easy to solve also the crucial part of our problem, the semantic disambiguation, through deep learning, but this would have been of little value as a cognitive model.

An interesting compromise between biological plausibility and level of abstraction is *Topographica* [10], designed with the main purpose of simulating cortical maps. Although it has been successfully used in exploration of neural semantics at the level of early language acquisition [98], Topographica finds its best use in modeling responses in visual areas. Currently, the neural framework that can simulate the widest range of cognitive tasks, by adopting a unified methodology with a reasonable degree of biological plausibility, is Nengo (Neural ENGineering Objects) [27]. The idea behind Nengo dates back to 2003, thanks to the former NEF (Neural Engineering Framework) [28], which defines a general methodology for the construction of large cognitive models, informed by a number of key neuroscientific concepts. In brief, the three main such concepts are the following:

- the *Representation's principle*: neural representations are defined by the combination of nonlinear encoding of spikes over a population of neurons, and weighted decoding over the same populations of neurons and over time;
- the *Transformation's principle*: transformations of neural representations are functions of the variables represented by neural populations. Transformations are determined using an alternately weighted decoding;
- the *Dynamic's principle*: neural dynamics are characterized by considering neural representations as state variables of dynamic systems. Thus, the dynamics of neurobiological systems can be analyzed using control (or dynamics systems) theory.

According to the listed principles, the basic computational object in Nengo is a popu-

lation of neurons that collectively can represent a multidimensional entity. The meaningful entity is retrieved from the neural activation by the following equation (pp.395–397 of [27]):

$$\vec{x} = \sum_{i,j}^{N,M} e^{-\frac{t-t_{i,m}}{\tau}} \vec{d}_i \tag{1.1}$$

where N is the number of neurons in the population, and M is the number of spikes that happen in the time windows of the computation; $t_{i,m}$ is the time when the i-th neuron in the population has fired for the m-th time; $\vec{d}_i$ is the i-th raw of the $N \times D$ *decoding* matrix $\vec{D}$ with D the dimension of the entity to be represented; τ is the time constant of decay of the postsynaptic activation. The activity of the neurons in a population depends from the *encoding* of their input that can be multidimensional with a dimension different from D.

A fundamental extension of the general neural population, ruled by equation (1.1), is the the Semantic Pointer Architecture (SPA), used when representing entities at higher cognitive level, i.e. conceptual and linguistic. In addition to the encoding and the decoding features, SPA structures allow a number of high level operations, that may correspond to conceptual manipulation, with some degree of biological plausibility. The foundation of these conceptual operations is in the mathematics of *holographic* representations, as theorized by Tony [97]. One of the basic operations is the *binding* of two SPA, computed by circular convolution (p.406 of [27]):

$$\vec{x} \circledast \vec{y} = \mathscr{F}^{-1}\left(\mathscr{F}(\vec{x}) \cdot \mathscr{F}(\vec{y})\right) \tag{1.2}$$

where $\mathscr{F}$ is the discrete Fourier transform and $\mathscr{F}^{-1}$ is its inverse, and $\cdot$ is the element-wise multiplication. Informally, SPA together with its associated operations, can be thought as a neural process that compresses information in other neural processes to which it points and into which it can be expanded when needed, providing shallow meanings through symbol-like relations to the world and other representations, and expanding to provide deeper meanings with relations to perceptual, motor, and emotional information, support complex syntactic operations. They also help to control the flow of information through a cognitive system to accomplish its goals. Thus semantic pointers have semantic, syntactic, and pragmatic functions, just like the symbols in a rule-based system, but with a highly distributed, probabilistic operation.

There are several reasons that have lead to this choice, one of the main reasons is that Nengo is developed from the NEF (Neural Engineering Framework), The NEF defines a general methodology for the construction of large cognitive models, and in particular of biologically plausible neural models. It acts as a neural compiler, that is it allows to specify the properties of the neurons, the values to be represented, the functions to be calculated and allows to find solutions for the connection weights between the components of the network that will perform the required functions. This system also allows to simulate complex dynamic models such as integrators, oscillators, and incorporates realistic error-based learning rules, allowing the adaptation and optimization of responses. The motivations that drive our research interest towards the creation of biologically plausible cognitive models and towards realistic models of neurons are increasingly shared by the scientific community.

In the first place, having biologically realistic (and plausible) models allows us to better verify the theories on which these models are based. If you want to understand exactly how a brain works, it is not enough to reproduce its correct behavior; it is necessary that this be done in exactly the same way that the brain would do it. That is, you should be able to have models of activation and neural connectivity that are comparable to natural ones. This drives our interest toward NEF models rather than algorithmic models of reasoning simulation.

In this first chapter I will limit my introduction to the basic elements of Nengo, and the important role that they have within my work. I will not include here one component, the Semantic Pointer Architecture, because it is charged by non trivial philosophical implications, and it deserves a deeper discussion in a next chapter. This first chapter is dedicated to the general explanation of how the main components of Nengo work, I will also show some examples of computational calculation through the use of some mathematical functions implemented in Python that will show us some essential arithmetic operations.

1.1 NEF - A Framework for simulating the brain

The NEF is the acronym that indicates the Neural Engineering Framework. The idea behind the NEF dates back to 2003, thanks to a book written by Eliasmith and Charles H. Anderson, entitled Neural Engineering [27] were many mathematical models and theories on how the biological natural systems were able to implement numerous dynamic functions were presented. From these theories the NEF takes origin. Eliasmith and Anderson focused on ”low-level” systems, including parts of the brainstem involved in controlling stable eye position. They used these methods to better understand more general issues about neural functions, such as how the variability of neural spike trains and the timing of individual spikes relate to information that can be extracted from spike patterns.

The NEF lends itself to a wide variety of applications. The reason for this is because it does not make assumptions about what specific functions the brain performs. Rather, it is a set of three principles that can help determine how the brain performs some given function. As John Miller once suggested, the NEF is a kind of neural compiler [25].

If you have an idea about the high-level function of the brain area in which you are interested, and you know some information about how neurons respond in that area, the NEF provides a way of connecting populations or layers of neurons together to realize that function. This, of course, is exactly what a compiler does for computer programming languages.

However, things are not so simple when it comes to the brain and its functions, building models with the NEF can be an iterative process: first, you gather from the neural system and generate a hypothesis about what it does; then you build a model using the NEF and see if it behaves like the real system; then, if it does not behave consistently with the data, you alter your hypothesis or perform experiments to figure out why the two are different. What the NEF offers is a systematic method for performing these steps in the context of neurally realistic models.

As I mentioned previously, the core of NEF are three principle, that will be shown in the following section:

- The *Representation's principle*: neural representations are defined by the combination of nonlinear encoding (exemplified by neuron tuning curves and neural spiking)

and weighted linear decoding (over populations of neurons and over time);

- The *Transformation's principle*: transformations of neural representations are functions of the variables represented by neural populations. Transformations are determined using an alternately weighted linear decoding;
- The *Dynamic's principle*: neural dynamics are characterized by considering neural representations as state variables of dynamic systems. Thus, the dynamics of neurobiological systems can be analyzed using control (or dynamics systems) theory.

These three principles will be described one by one and practical examples will be provided on how they work.

1.1.1 The Representation's Principle

A central principle in NEF is that of adaptability. We can adapt the theoretical information we have about codes, to understanding the representation of neural systems. Codes are defined in terms of complimentary encoding and decoding procedures between two alphabets, as for the morse alphabet for example. To encode it, it is necessary to find a link between the set of lines and the dots that compose it and the alphabet in roman letters. The encoding procedure is the mapping from the roman alphabet to the morse code alphabet, and the decoding procedure is its inverse [73].

To characterize representation in a neural system, we must identify the relevant encoding and decoding procedures and the relevant alphabets. Regarding the encoding procedure one typical example is the mapping of stimuli into a series of neural spikes. Encoding is what neuroscientists typically measure and is what is partly captured by the tuning curves, a tuning curve is a graph of neuronal reaction as an operation of a ongoing stimulant attribute, like wavelength, orientation, or frequency. This pattern of firing is often depicted by neuroscientists as a spike raster. Spike rasters show the only response to the stimulus that is sent to other neurons.

However, we can not limit simply to the encoding procedure when we talk about neural representation, but it is very important to talk about decoding procedures.

Characterizing the decoding of neural spikes into the variable they represent is as important as characterizing the process of encoding variables into neural spikes, for example, if no information about a stimulus can be extracted from the spikes of the encoding neurons, then it makes no sense to say that they represent the stimulus. Representations, at a minimum, must potentially be able to stand in for the things they represent.

Essentially there are two kinds of encoding procedures: linear and non-linear. As for the linear one, we can say that nonlinear encoding maps a continuously varying parameter like stimulus intensity into a series of discontinuous spikes; in the linear ones the decoding procedure is linear in the sense that the responses of neurons in the population are weighted by a constant (the decoder) and summed up to give the decoding.

The NEF employs a specific method for determining appropriate linear decoders given the neural responses to stimuli. There are two aspects to the decoding of the neural response that must be considered. These are what Eliasmith [27] called the population and temporal aspects of decoding. The population decoding accounts for the fact that a single stimulus variable is typically encoded by many different (i.e. a population of) neurons. The temporal decoding accounts for the fact that neurons respond in time to a typically changing stimulus. Ultimately, these two aspects determine one combined decoding. However, it is conceptually clearer to consider them one at a time. Population decoders are determined by finding the weighting of each neuron tuning curve, so that their sum represents the input signal over some range.

1.1.2 The Transformations's Principle

The transformation principle is another key principle of the NEF. This is because transformations (or calculations) can also be characterized using decoding. And this can be done by identifying a transformation decoder which is a particular type of decoding, a polarized decoding. That is, in determining a transformation we extract information other than what the population is taken to represent. The bias, then, is away from representational, decoding of the encoded information.

For example, if we think that quantity x is encoded in a certain neural population, when we define the representation for this population we determine the decoders that es-

timate x. However, when we define a transformation, we identify decoders that estimate some functions, $f(x)$, of the amount represented. In other words, we find the decoders that, rather than extracting the signal represented by a single population, extract a *transformed* version of that signal [53].

1.1.3 The Dynamic's principle

Dynamic processes play an extremely important role in the cognitive sciences and neurobiological studies, dynamic processes are identifiable even in the simplest nervous systems. Functions like moving, eating and perceiving a world in continuous evolution and change are clearly dynamic processes. It is not surprising, then, that single neural cells have almost always been characterized by neuroscientists as essentially dynamic systems [73].

The modern theory of control, which allows both the analysis and the synthesis of complex dynamical systems, has been developed precisely because the understanding of complex dynamics is essential to construct something that works in the real world. Thanks to its general formulation, this theory can be applied to chemical, electrical, digital or analogical systems.

The third principle of the NEF is the suggestion that the space representations of neural populations are the state variables of a dynamical system defined using control theory.

1.2 Nengo

Nengo is a Python library that allows the construction and simulation of large-scale brain models using the NEF method. Nengo is able to create sophisticated neural simulations using a few lines of code. Moreover it is strongly extensible and flexible, it is in fact possible to define the types of neurons and the training rules of neural networks, it is possible to take input directly from an hardware, as well as to simulate your model on different neural simulators. Nengo (Neural ENGineering Objects) is a graphical neural simulation environment developed over the last several years by the research group at the Centre for Theoretical Neuroscience at the University of Waterloo, it derives from the

fruitful work of Chris Eliasmith and his doctoral student Eric Hunsberger [51] who have worked for years on the creation of the NEF.

1.2.1 The main Nengo's objects

Nengo is based on Python, an high level programming language that supports different programming paradigms, such as the object-oriented one (with support for multiple inheritance), the imperative and the functional one, and offers a strong dynamic typing. Python comes with an extremely rich built-in library, which together with automatic memory management and exception-handling constructs makes it one of the richest and most convenient languages to use.

In computer science, object-oriented programming (OOP, Object Oriented Programming) is a programming paradigm that allows to define software objects able to interact with one another through the exchange of messages. It is particularly suitable in contexts in which interdependence relationships can be defined between the concepts to be modeled (containment, use, specialization).

Specifically, in object-oriented programming, the object is an instance of a class, unique and separate from other objects (according to the concept of encapsulation) with which it can however "to communicate".

Nengo is as the others, a library that enriches the Python programming language of particular functionalities and allows the realization of biologically plausible neural networks, using the methodologies proposed by the NEF. In this first chapter, I briefly introduce the characteristics of Nengo and of the objects that make up the library. During the presentation phase, I will propose several examples to show how each Nengo object works and interacts with other objects in the network.

The Node Object

The Node object (`nengo.node`) is an external object to a neural network that is used both to provide Nengo objects with non-neural inputs and to process their outputs [27]. The nodes can also accept input values and realize arbitrary mathematical transformations in order to check the input data provided in the simulations carried out on a Nengo neural

network. The nodes, therefore, typically are not part of the brain model but are used to generate (through their simulation) the stimuli, which are acquired by sensors or other environment variables, and which can not be generated by the brain model itself. Nodes can also be used to test models by providing specific input signals to specific parts of the model itself, so they can simplify the process and input / output interface of a neural network when used as interfaces with internal elements of a Nengo network.

The Ensemble Object

a group or population of neurons is implemented by using an Ensemble object (`nengo.ensemble`). When an Ensemble is created, a wide variety of parameters can be set. Some of these parameters are set in order to reflect some physiological properties of the neurons that are modeled, while others are set in order to improve the accuracy of the transformations that will be simulated by the neurons themselves. A population of neurons can also contain just a single neuron.

The Connection Object

The Connection object (`nengo.connection`) it allows to connect two objects belonging to a Nengo model. A connection is defined by two parameters, that is, the source object and the destination object. This connection is unidirectional, in the sense that the information is transmitted from the first argument of the connection (the source) to the second element (the destination). Almost all Nengo objects can act as a source or as a destination for a connection. However, it is necessary that both objects have the same size. For example, if an object named node has a dimension of 2 and the object named `ensemble` has a size of 1, the following connection can not be created:

```
1 nengo.Connection(node, ensemble)
```

However, it is possible to create both the following two connections:

```
1 nengo.Connection(node[0], ensemble)
  nengo.Connection(node[1], ensemble)
```

The Network Object

The Network Object (`nengo.network`) it represents a neural network and can contain Ensemble, Node, Connection or other Network objects [5]. A Network is primarily used to group objects and connections together for display purposes. However, it is possible to use a Network to allow the reuse of the code.

The Probe Object

The Probe object (`nengo.probe`, it is used to collect data related to simulations and it is used in any context where it is necessary to explore and display simulation data for analysis. For this reason, a Probe object does not affect the brain model and its simulation data in any way. Any Nengo object can be used for the exploration in the context of a Probe object (apart from Probe objects themselves), depending on the different attributes in the object that can be viewed. To understand what can be displayed through a Probe object, you can view the `probeable` attributes of an object.[73]

1.3 The Representation's Principle with Nengo

In this section we will show in detail some examples of application of the principle of representation in Nengo.

1.3.1 Representation with a single neuron

The example shown in this section illustrates how to build and manipulate a single LIF (leaky integrated-and-fire) neuron, which is a standard model of neuron. In this example, a population of neurons formed by a single element is managed. The Figure 1 1.1 shows the Python code of our example. As a first step the neural model (line 6) is created, which represents our simple network and inside which the population of neurons is inserted. A group of neurons is identified in Nengo by an Ensemble object. In our case, when the Ensemble object is created (line 8), the number of neurons it contains is specified (line 9), its size (in our case it is a simple scalar value) and various other parameters that define the characteristics of reaction to the signal.

```
1  import numpy as np
2  import matplotlib.pyplot as plt
3  import nengo
4  from nengo.dists import Uniform
5
6  model = nengo.Network(label='Singolo Neurone')
7  with model:
8      neuron = nengo.Ensemble(
9                               1,
10                              dimensions=1,
11                              intercepts=Uniform(-.5, -.5),
12                              max_rates=Uniform(100, 100),
13                              encoders=[[1]])
14     cos = nengo.Node(lambda t: np.cos(8 * t))
15     nengo.Connection(cos, neuron)
16     cos_probe = nengo.Probe(cos)
17     spikes = nengo.Probe(neuron.neurons)
18     voltage = nengo.Probe(neuron.neurons,'voltage')
19     filtered = nengo.Probe(neuron, synapse=0.01)
20
21 with nengo.Simulator(model) as sim:
22     sim.run(1)
```

Figure 1.1: Python code for a single neuron

In particular, the neuron is excited by increasing values of the input signal, and this is indicated by the value of `encoders` set at 1 in line 13. Furthermore, the neuron is excited for signal values higher than -0.5, as indicated in line 11. Finally the maximum value of the signal produced by the neuron is set at 100Hz, as indicated in line 12. In Fig. 1.2 (top right) the tuning curve of the neuron is shown. Notice how the curve assumes values of 0 when the input signal is below the value -0.5. The neuron begins to generate impulses, gradually increasing, starting from the value -0.5 to reach the maximum value (100 Hz) when the input signal reaches the value 1. This curve represents the response of the neuron to the input signal and is independent of the simulation. After defining the characteristics of our neuron, with the help of a Node object, an input signal is simulated, represented by the cosine function $\lambda(t) = \cos(8t)$ (line 14). The Input signal is connected to the population of our network (formed by a single neuron) through a Connection object (line 15). Fig. 1.2 shows (at the top left) the input signal generated in the time span of one second, i.e. for $t = 0, \ldots, 1$. In the subsequent lines of code (lines 16-19), through the Probe objects, data for display and analysis are collected. Finally, in line 21, a Simulator object is created to manage the simulation of our model. This simulation is started for the

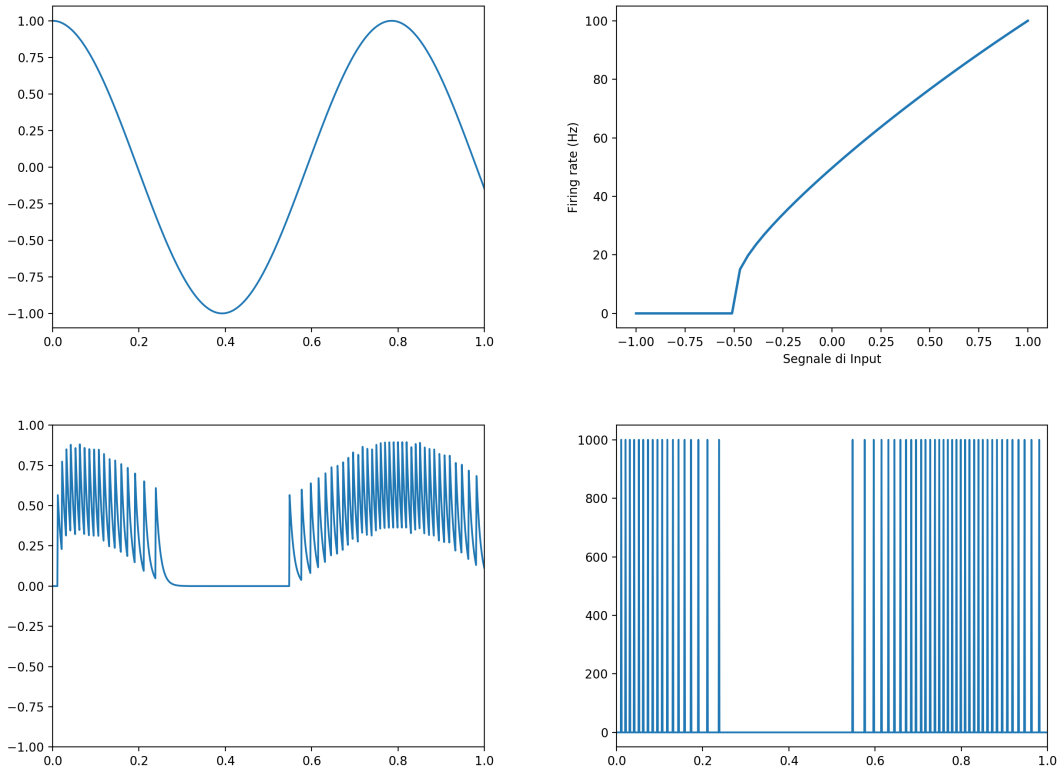


Figure 1.2: (Upper left) the input signal identified by the function $\lambda(t) = \cos(8t)$, per $t = 0, \ldots, 1$; (Top right) the tuning curve of the neuron; (Bottom left) the response of the neuron to the stimulus produced by the input signal; (Bottom right) the impulses produced by the single neuron, filtered through a 10ms post-synaptic filter.

duration of a second in line 22 [27].

In Fig. 1.2 (below) are shown the results of the simulation performed on our network. In particular, on the left, the response of the neuron to the stimulus generated by the input signal is represented. Notice how the neuron begins to emit pulses when the input signal is above the -0.5 value. On the right are the impulses produced by the single neuron, filtered through a 10ms post-synaptic filter.

In Fig. 1.3 are shown the results of the simulation performed on the same neural network, considering an input signal linked to the mathematic function $\lambda(t) = 4t^3 - \frac{5}{2}t + \frac{1}{5}$. In particular, on the left, is represented the response of the neuron (in blue) to the stimulus generated by the input signal (in orange). On the right are the impulses produced

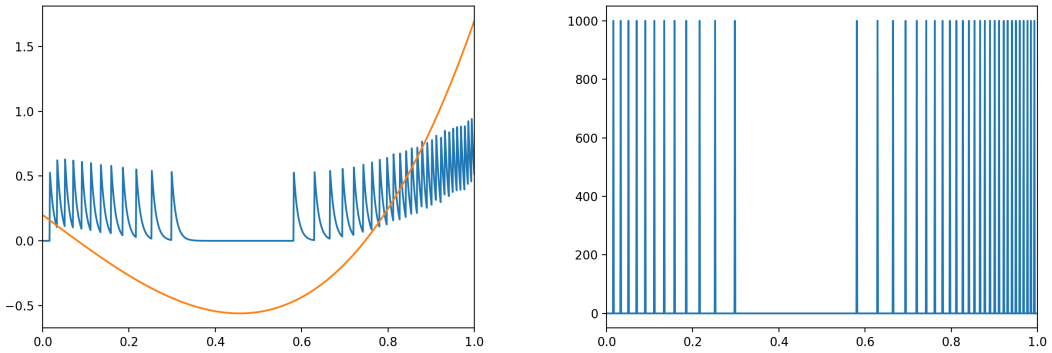


Figure 1.3: The results of the simulation performed on a neural network formed from a single neuron, considering an input signal linked to the mathematic function $\lambda(t) = 4t^3 - \frac{5}{2}t + \frac{1}{5}$. On the left, the response of the neuron 25 is represented (in blue) to the stimulus generated by the input signal (in orange). On the right are the impulses produced by the single neuron, filtered through a 10ms post-synaptic filter.

by the single neuron, filtered through a 10ms post-synaptic filter.

1.3.2 Representation with a pair of neurons

In this example we show how to create a neural network formed by a pair of neurons with a complementary behavior. As in the previous example, the neurons that form the network are LIF neurons and their behavior is characterized in such a way that one undergoes complementary stimuli to the other. Specifically, the first neuron will increase its spike frequency in response to positive signals, while the latter will increase its spike frequency in response to negative signals. This is the simplest population formed by a pair of neurons capable of providing a reasonable representation of a scalar value.

Fig. 1.4 shows the Python code used to manage a couple of neurons. As done in our previous example, we used a Node object for the simulation of the input signal, represented by the cosine function $\lambda(t) = \sin(8t + 2)$ (line 12). The Input signal is connected to the population of our network (formed by a pair of neurons) through a Connection object (line 13).

The population of neurons (defined in line 7) is characterized by a pair of neurons (line 8) whose maximum spike value is equal to 100 Hz (line 10). However, the two neu-

```
import numpy as np
import nengo
from nengo.dists import Uniform

model = nengo.Network(label='Due Neuroni')
with model:
    neurons = nengo.Ensemble(
                              2, dimensions=1,
                              intercepts=Uniform(-.5, -.5),
                              max_rates=Uniform(100, 100),
                              encoders=[[1], [-1]])
    sin = nengo.Node(lambda t: np.sin(8 * t+2))
    nengo.Connection(sin, neurons, synapse=0.01)
    sin_probe = nengo.Probe(sin)
    spikes = nengo.Probe(neurons.neurons)
    voltage = nengo.Probe(neurons.neurons, 'voltage')
    filtered = nengo.Probe(neurons, synapse=0.01)

with nengo.Simulator(model) as sim:
    sim.run(1)
```

Figure 1.4: Python code for the management of a pair of neurons

rons have a complementary behavior defined by the line 11 encoders: the first neuron is stimulated for increasing values of the input signal, while the second neuron is stimulated for decreasing values. Consequently, since their lower stimulation limit is, for both, equal to -0.5 (line 9), the first neuron will start responding to stimuli for values of the input signal above -0.5, while the second neuron will begin to respond to stimuli for input signal values lower than 0.5. This behavior is highlighted by the graph in Fig 1.5 (top right) in which the tuning curves of the two neurons are shown.

In Figure 51.5 also shows the results of the simulation performed, for a total time of one second, on the neural network just described, formed by a pair of neurons. Specifically, at the top left is the response of the neurons (in blue) to the stimulus generated by the input signal (in orange). In the lower left corner are the impulses produced by the pair of neurons, filtered through a 10ms post-synaptic filter, while the lower right side shows the response of the neurons to the input signal stimuli.

It can be observed how the first neuron is stimulated for values of the input signal between -0.5 and 1, with a spike frequency that gradually increases as the input signal approaches the value 1. In a complementary way the second neuron is stimulated for input signal values between 0.5 and -1, with a spike frequency gradually increasing as

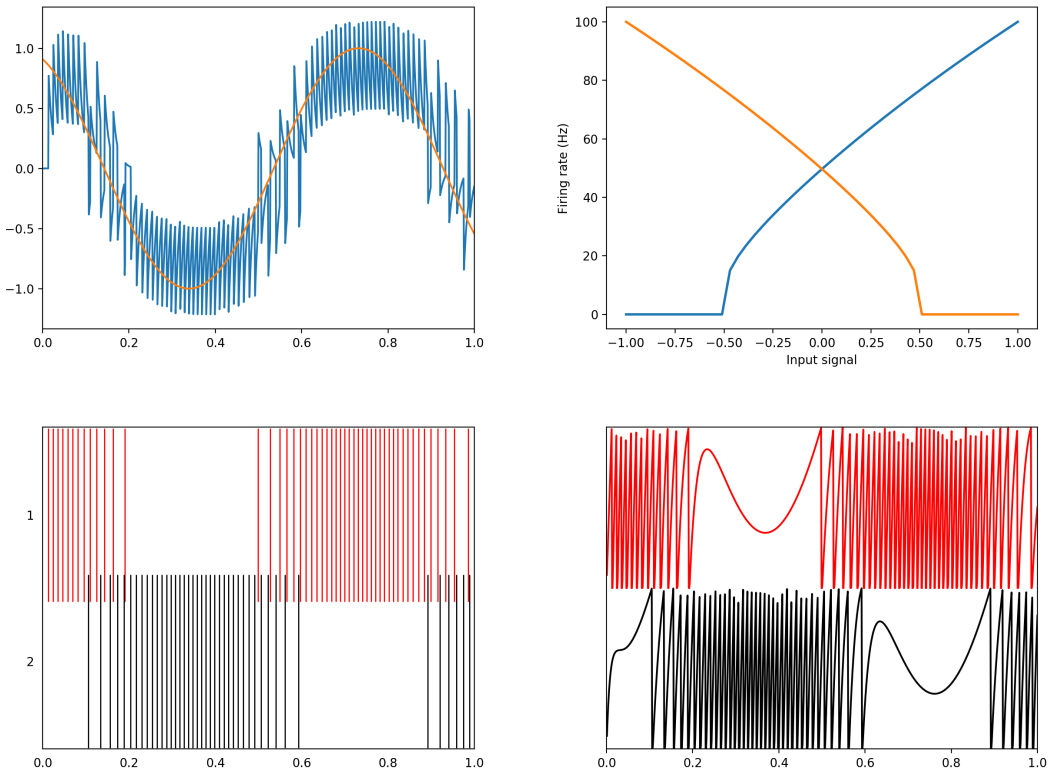


Figure 1.5: The results of the simulation performed, for a total time of one second, on the neural network just described, formed by a pair of neurons. Specifically, at the top left is the response of the neurons (in blue) to the stimulus generated by the input signal (in orange). In the lower left corner are the impulses produced by the pair of neurons, filtered through a 10ms post-synaptic filter, while the lower right side shows the response of the neurons to the input signal stimuli.

```
import numpy as np
import nengo

model = nengo.Network(label='100 Neuroni')
with model:
    A = nengo.Ensemble(100, dimensions=1)
    sin = nengo.Node(lambda t: 2*t*t - np.sin(t))
    nengo.Connection(sin, A, synapse=0.01)
    sin_probe = nengo.Probe(sin)
    A_probe = nengo.Probe(A, synapse=0.01)
    A_spikes = nengo.Probe(A.neurons)

with nengo.Simulator(model) as sim:
    sim.run(1)
```

Figure 1.6: Python code for the management of a pair of neurons

the input signal decreases approaching the value -1. The response of the two neurons to the stimuli provided by the input signal can well represent the input signal in response to which they are activated. We will see in the next sections how the increase in the number of neurons in the population the representation of the input signal will be significantly more accurate.

1.3.3 Representation with a Neural Population

In this example we show how to generate a simple neural network consisting of a population of 100 neurons. As in previous cases, the population consists of LIF-type neurons whose response properties to the input signal are defined randomly.[51]

La Fig. 1.6 The figure 6 shows the Python code used to manage this population of neurons. As in previous cases, a Node object is used for the simulation of the input signal, represented by the cosine function $\lambda(t) = 2t^2 - \sin(t)$ (line 7). The population of 100 neurons is created in line 6. The absence of the parameters that define the properties of response to the input signal, indicates that these are generated randomly. The input signal is then connected to the population of our network (formed by a pair of neurons) through a Connection object (line 8). In Figure 1.7 it is shown the graphs related to the tuning curves (on the left) of the 100 neurons that make up the population, and the curve that represents the input signal supplied to the population of neurons (on the right). Notice how the direction of stimulation response, the maximum spike value, and the lower limit

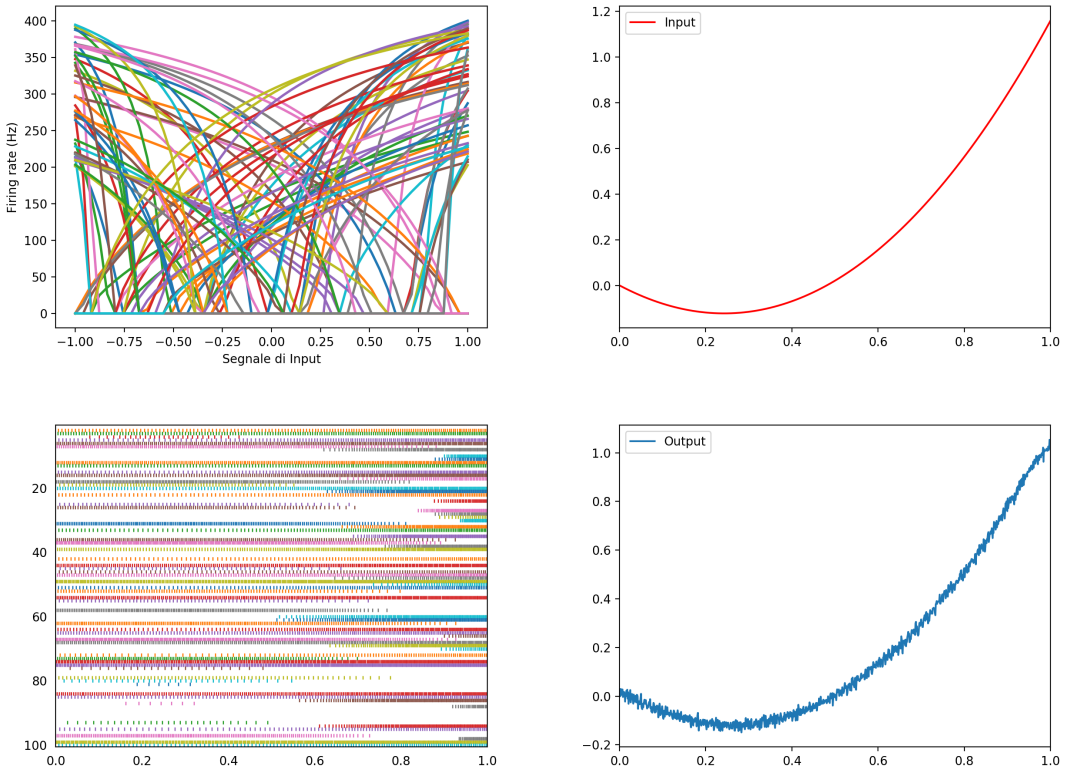


Figure 1.7: The graphs related to the tuning curves (on the left) of the 100 neurons that make up the population, and the curve that represents the input signal supplied to the population of neurons (on the right). (Below) The impulses produced by the population of neurons (on the left), filtered through a post-synaptic filter of 10ms and (on the right) the response of the neurons to the stimulus generated by the input signal.

at which each individual neuron responds are significantly different.

In Figura 1.7 (below) are instead shown the graphs of the results of the simulation performed, for a total time of one second, on the neural network just described. Specifically on the left are the impulses produced by the population of neurons, filtered through a 10ms post-synaptic filter. On the right is the neuron response to the stimulus generated by the input signal.

It can be observed that the accuracy in the representation of the input signal is very good for this population of neurons. This can be evidenced by the fact that the input signal and the output signal generated by the neurons are very similar.

```
import numpy as np
import nengo

model = nengo.Network(label='Reppresentazione 2D')
with model:
    neurons = nengo.Ensemble(100, dimensions=2)
    sin = nengo.Node(lambda t: np.sin(t))
    cos = nengo.Node(lambda t: np.cos(t))
    nengo.Connection(sin, neurons[0])
    nengo.Connection(cos, neurons[1])
    sin_probe = nengo.Probe(sin, 'output')
    cos_probe = nengo.Probe(cos, 'output')
    neurons_probe = nengo.Probe(neurons, 'output', synapse=0.01)

with nengo.Simulator(model) as sim:
    sim.run(5)
```

Figure 1.8: Python code for managing a population of 100 neurons

1.3.4 Representation of a pair of signals

In this section we will construct a simple neural network which is able to reacting to an external 2-dimensional signal. In Nengo, this type of signal is managed by a pair of vectors (ie sequences of real values) mono-dimensional. The neural network uses two input communication channels within the same population of neurons.

La Fig. 1.8 shows the Python code used for the simulation of this neural network. The neural model provides for the presence of only one group of neurons, created through a single Ensemble object, called`neurons` of 100 neurons (line 6). La specifica `dimesione=2` indicates that the neuron population is able to manage two distinct input channels. Two Node objects are used for the simulation of the two input signals which, in our example, are represented by the two sine and cosine functions, ie $\lambda_1(t) = \sin(t)$ e $\lambda_2 = \cos(t)$ (lines 7-8). The two input signals are then connected to the ensemble object through the use of two Connection objects (lines 9-10). However, note that the first signal is connected to the first component of the Ensemble object (`neurons[1]`) while the second signal is connected to the second component of the Ensemble object (`neurons[2]`).

In Figure 1.9 it is shown the graph of the simulation result performed, for a total time of 5 seconds, on the neural network just described. The two outputs produced in response to the two input signals are represented in blue and orange, respectively. Notice how the two signals are faithfully reproduced from the synaptic response of the neuron population.

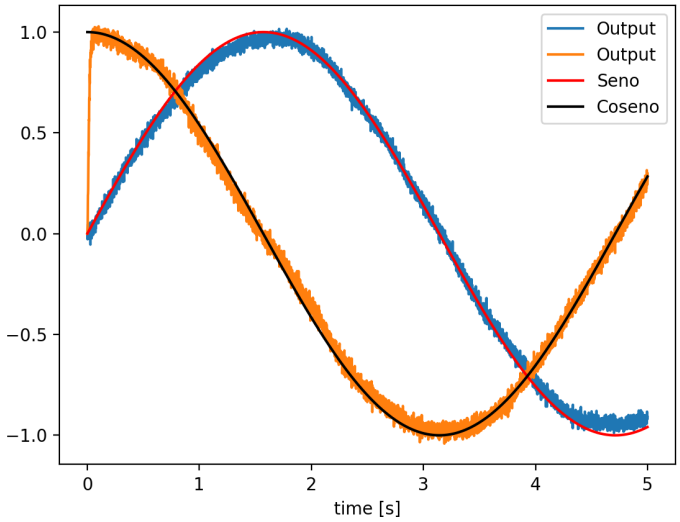


Figure 1.9: The graph of the simulation result performed, for a total time of 5 seconds, on the neural network just described. The two outputs produced in response to the two input signals are represented in blue and orange, respectively. Notice how the two signals are faithfully reproduced from the synaptic response of the neuron population.

1.3.5 Representation of the sum of two signals

When two input signals converge in the same population of neurons, they combine each other to generate a single signal whose values are given by the sum of the values present in the two original signals. This property is also valid when the two input signals come from the output of two different populations of neurons that react to some external stimulus [138].

In this section we will construct a simple neural network capable of simulating the addition of its input signals. The neural network uses two communication channels within the same population of neurons. In this way the addition of the two inout channels is a result that is obtained automatically, since the signals coming from different synaptic connections interact linearly.

In Figure 1.10 it is shown the Python code used for the simulation of this neural network. The neural model provides for the presence of three groups of neurons, created through three Ensemble objects, A, B e C, of 100 neurons each (lines 5-7). Two Node

```
import nengo

model = nengo.Network(label='Addition')
with model:
    A = nengo.Ensemble(100, dimensions=1)
    B = nengo.Ensemble(100, dimensions=1)
    C = nengo.Ensemble(100, dimensions=1)
    input_a = nengo.Node(output=0.5)
    input_b = nengo.Node(output=0.3)
    nengo.Connection(input_a, A)
    nengo.Connection(input_b, B)
    nengo.Connection(A, C)
    nengo.Connection(B, C)
    input_a_probe = nengo.Probe(input_a)
    input_b_probe = nengo.Probe(input_b)
    A_probe = nengo.Probe(A, synapse=0.01)
    B_probe = nengo.Probe(B, synapse=0.01)
    C_probe = nengo.Probe(C, synapse=0.01)

with nengo.Simulator(model) as sim:
    sim.run(5)
```

Figure 1.10: Python code for managing a population of 100 neurons

objects are used to simulate the two input signals which, in this example, are represented by the two constant functions $\lambda_1(t) = 0.5$ e $\lambda_2 = 0.3$ (lines 8-9). The two input signals are then connected to the A and B populations, respectively, through the use of two Connection objects (lines 10-11).

The linear combination of the signals produced in response from the A and B populations is realized by channeling these rials as stimuli for the C population, through the creation of two parallel connections that drive the signal from the two A and B populations. B to the C population (lines 12-13).

In Figure 1.11 (upper left) it is shown the graphs of the input signals produced by the two Node objects and the results of the simulation performed, for a total time of 5 seconds, on the neural network just described. The synaptic response of the A population is indicated in blue, whereas the population of the B population is orange, while the synaptic response of the C population is indicated in green. It can be seen that the signals coming out of the two populations A and B, faithfully representing the two functions produced by the input signals, converge simultaneously as input of the B population. As a consequence, the latter group of neurons faithfully reproduces the sum of the two signals, producing the desired output in response.

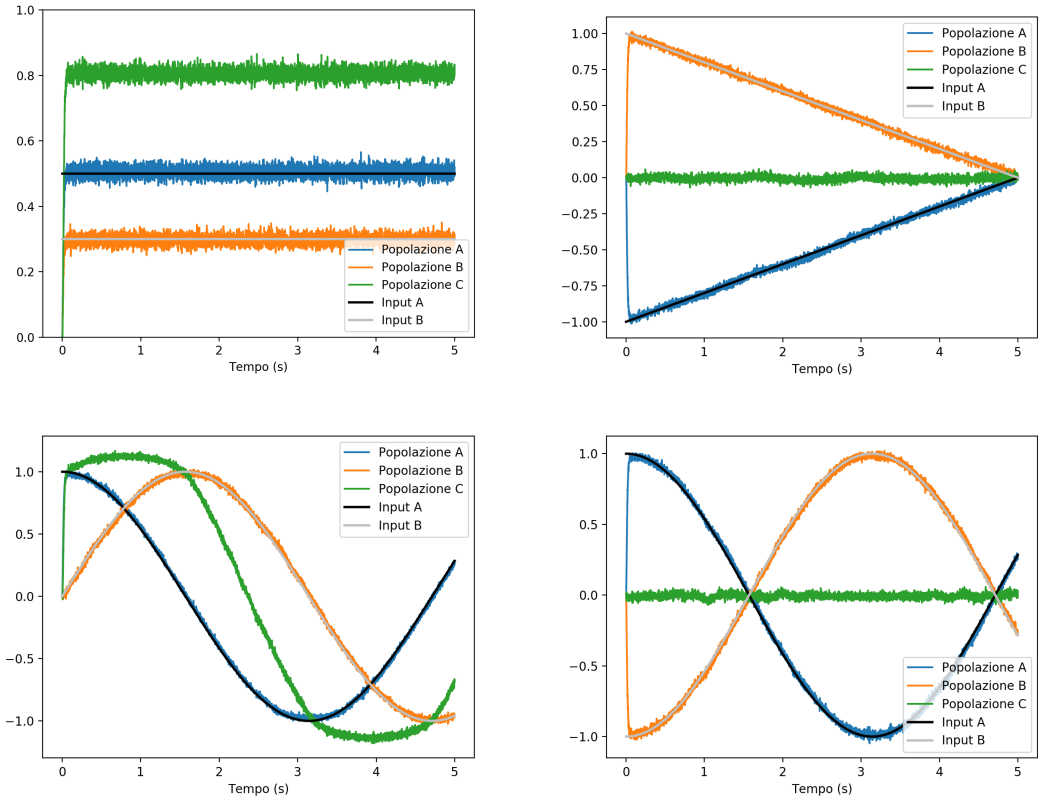


Figure 1.11: The graphs of the simulation performed on a neural network able to represent the sum of two distinct input signals. Specifically, the sum functions of two distinct input signals are represented. Specifically, the functions are represented, $\lambda_1(t) = 0.5$ and $\lambda_2(t) = 0.3$ (in alto a sinistra), $\lambda_1(t) = \sin(t)$ and $\lambda_2(t) = \cos(t)$ (lower left), $\lambda_1(t) = \sin(t)$ and $\lambda_2(t) = -\sin(t)$ (lower right).

```
import numpy as np
import nengo

model = nengo.Network(label='Squaring')
with model:
    A = nengo.Ensemble(100, dimensions=1)
    B = nengo.Ensemble(100, dimensions=1)
    sin = nengo.Node(lambda t: 0.5 * np.sin(t))
    nengo.Connection(sin, A)
    def ampl(x): return x[0] * 2
    nengo.Connection(A, B, function=ampl)
    sin_probe = nengo.Probe(sin)
    A_probe = nengo.Probe(A, synapse=0.01)
    B_probe = nengo.Probe(B, synapse=0.01)

with nengo.Simulator(model) as sim:
    sim.run(20)
```

Figure 1.12: Python code for managing a population of 100 neurons

In Figure 1.11 it is also shown the result of the same simulation, performed on three different neural networks in which the input signals introduced in the A and B populations have been changed. Specifically we tested the functions $\lambda_1(t) = 1 - \frac{1}{5}t$ and $\lambda_2(t) = \frac{1}{5}t - 1$ (top right), $\lambda_1(t) = \sin(t)$ e $\lambda_2(t) = \cos(t)$ (lower left), $\lambda_1(t) = \sin(t)$ and $\lambda_2(t) = -\sin(t)$ (lower right). In all cases it is possible to notice how the output signal produced by the B population faithfully reproduces the sum of the two input signals.

1.4 The Transformation Principle with Nengo

In this section I will present some examples of application of the transformation principle in Nengo.

1.4.1 An amplification channel

As seen in the previous examples, a population of neurons can emit an output in response to a signal external to the network, produced by a Node object, or in response to a signal inside the network, produced by one or more populations of neurons [138]. In the previous section, for example, a neural network was created in which a population of neurons emitted a signal in response to the stimulations produced by two different populations of neurons, thus producing a signal that represented the sum of these stimulations.

In this first example of transformation, we show how a population of neurons can be used, not only to represent faithfully or to combine the input received from other populations within the network, but is also able to apply simple linear transformations to the signal of input received. To this end we will create a neural network capable of amplifying the signal received from the outside.

In Figure 1.12 it is shown the Python code used for the simulation of this neural network. The neural model predicts the presence of two groups of neurons, created through three Ensemble objects, A and B, of 100 neurons each (lines 6-7). An Node object is used for the simulation of the input signal which, in our example, is represented by the function$\lambda(t) = 0.5\sin(t)$ (line 8). The input signal is then connected to the A populations through the use of a Connection object (line 9).

To realize the transformation of the signal it is essential to act on the connection realized between the population A and the population B. Specifically, we define a Python function called `ampl`, which amplifies the signal by doubling the input value. We set `ampl`$(x) = 2x$ (line 10).

Then a connection is defined between the A population and the B population (line 11) in which the function `ampl` it is used as a signal filter. This allows the output signal from the Ensemble A object to arrive in the Ensemble B object, with a double intensity.

La Fig. 1.13 shows the graph relating to the simulation performed on the neural network just described for a total time of 20 seconds. In blue the signal produced by the population of neurons A is shown, which faithfully reproduces the external input signal, while in orange the output signal produced by the population of B neurons is shown, whose value amplifies that produced by A.

The transformation made in this example can be modified to involve any function on the input signal. For example in Fig. 1.13 (on the right) the 20-second simulation graph is shown on a neural network very similar to the one described in this section, where the transformation function is defined by $f(x) = x^2$. Notice how the signal produced by the B population (in orange) is actually the square of the output signal generated by the first population of neurons (in blue).

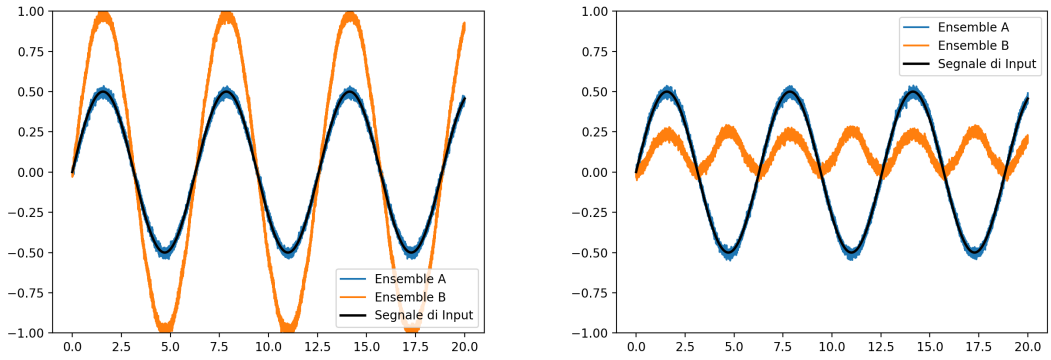


Figure 1.13: The graph relating to the simulation performed on the neural network which carries out an amplification of the signal (on the left) is the square of the signal (on the right). In blue the signal produced by the population is shown that faithfully reproduces the external input signal, while in orange the output signal produced by the transformation is shown.

1.4.2 Arithmetic operations

In this section we will show how, through the transformation of the signal, it is possible to implement arithmetic operations such as multiplying two values. The model that implements this type of transformation can be thought of as a combination of the model used for the representation of a pair of signals and the one used for the representation of the square of a signal, shown in the previous section.

The model (whose code is shown in Fig. 1.14) is composed of four populations of neurons, two of which (called *A* and *B*) are used for the representation of external input signals, the third (called `combined`) it is used for the two-dimensional combination of the two input signals, while the fourth population (called *D*) is used for the non-linear transformation of the two input signals.

Once again, two Node objects are used for the simulation of the input signal that, in our example, are represented by broken functions defined through the Python model `picewise` (linee 13-14). I segnali di input vengono poi collegati alle popolazioni *A* e *B*, rispettivamente, attraverso l'utilizzo di due oggetti Connection (lines 16-17).

The population `combined` in line 10, it is defined as a population of two-dimensional neurons. The connection between the *A* and *B* populations and the two components of the

```
import numpy as np
import nengo
from nengo.dists import Choice
from nengo.utils.functions import piecewise

model = nengo.Network(label='Multiplication')
with model:
    A = nengo.Ensemble(100, dimensions=1, radius=10)
    B = nengo.Ensemble(100, dimensions=1, radius=10)
    combined = nengo.Ensemble(220, dimensions=2, radius=15)
    prod = nengo.Ensemble(100, dimensions=1, radius=20)
    combined.encoders = Choice([[1, 1], [-1, 1], [1, -1], [-1, -1]])
    inputA = nengo.Node(piecewise({0: 0, 2.5: 10, 4: -10}))
    inputB = nengo.Node(piecewise({0: 10, 1.5: 2, 3: 0, 4.5: 2}))
    correct = piecewise({0: 0, 1.5: 0, 2.5: 20, 3: 0, 4: 0, 4.5: -20})
    nengo.Connection(inputA, A)
    nengo.Connection(inputB, B)
    nengo.Connection(A, combined[0])
    nengo.Connection(B, combined[1])
    def product(x): return x[0] * x[1]
    nengo.Connection(combined, prod, function=product)
    inputA_probe = nengo.Probe(inputA)
    inputB_probe = nengo.Probe(inputB)
    A_probe = nengo.Probe(A, synapse=0.01)
    B_probe = nengo.Probe(B, synapse=0.01)
    combined_probe = nengo.Probe(combined, synapse=0.01)
    prod_probe = nengo.Probe(prod, synapse=0.01)

with nengo.Simulator(model) as sim:
    sim.run(5)
```

Figure 1.14: Python code for managing a population of 100 neurons

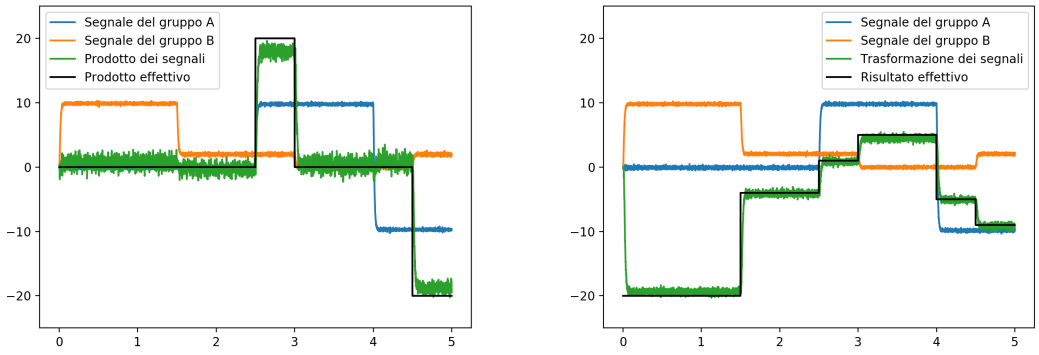


Figure 1.15: The graph relating to the simulation carried out on the neural network that produces the product of two signals (on the left), ie the transformation $f(x,y) = xy$, and the arithmetic transformation $f(x,y) = \frac{1}{2}x - 2y$ (on the right). In both cases x and y they represent the two input signals.

`combined` is carried out on lines 18-19.

Finally, a Python function called `product`, which has the task of calculating the product of the two components of the signal x received in input (line 20), and a connection is made between the population `combined` and the D population, conveyed by the transformation defined by the function `product`.

In Fig. 1.15 (on the left) shows the graph of the signals generated during the simulation produced by our example, with a total duration of 5 seconds. The graph shows the representation of the two input signals (in blue and orange, respectively) and the signal produced in response by the D population. Note how this signal is sufficiently close to the function (represented in black) that identifies the actual product between the two input signals.

La Fig. 1.15 (on the right) shows instead the graph of the signals generated during the simulation produced on a neural network that implements the arithmetic transformation $f(x,y) = \frac{1}{2}x - 2y$, in which x and y they represent the two input signals. Also in this case the graph shows the representation of the two input signals (in blue and orange, respectively) and the signal produced in response by the population that applies the non-linear transformation.

```
import nengo
from nengo.utils.functions import piecewise

model = nengo.Network(label='Integratore')
with model:
    A = nengo.Ensemble(100, dimensions=1)
    input = nengo.Node(piecewise({0: 0, 0.2: 1, 1: 0, 2: -2, 3: 0, 4: 1, 5: 0}))
    tau = 0.1
    nengo.Connection(A, A, transform=[[1]], synapse=tau)
    nengo.Connection(input, A, transform=[[tau]], synapse=tau)
    input_probe = nengo.Probe(input)
    A_probe = nengo.Probe(A, synapse=0.01)

with nengo.Simulator(model) as sim:
    sim.run(6)
```

Figure 1.16: Codice Python per la gestione di una popolazione di 100 neuroni

1.5 Dynamic Transformations with Nengo

In this section we will show some examples of application of the principle of dynamic transformation in Nengo.

1.5.1 Simulation of a integrator

In mathematics integration is the process that determines the integral of a function, also called quadrature. The integrator is therefore a tool that can determine the integral of a function. The neural model presented in this section implements a mono-dimensional neural integrator, ie an instrument capable of determining the integral of a function obtained as an input signal.

This example shows how a population of neurons can be used for the implementation of stable processes of dynamic transformation. These dynamic transformations are at the basis of the implementation of memories, noise cancellation, statistical inference, and many other neural processes.

The model (whose code is shown in Fig. 1.16) it is composed of only one population of neurons, called *A*, and defined as a group of 100 neurons (line 6). An Node object is used for the simulation of the input signal that, in our example, for simplicity is represented by a broken function defined through the Python model `picewise` (line 7). The input signal is then connected to the *A* population through the use of a Connection object

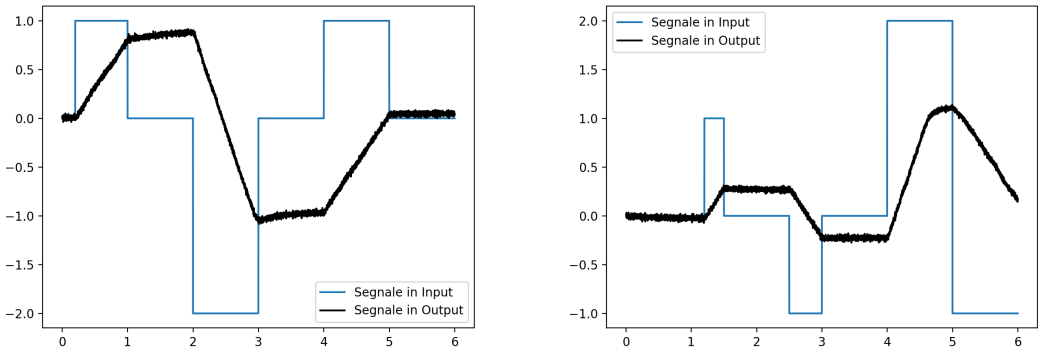


Figure 1.17: The graphs produced by the simulation (of 6 seconds) of a neural model that implements an integrator.

(line 10).

Instead, dynamic transformation is implemented through the creation of a neural connection between the *A* population and itself (line 9). This requires that the output signal generated by the group of neurons, under the stimulus of the input signal, is added to the latter generating a second input which, therefore, varies dynamically over time.

After launching the model simulation for a period of 6 seconds, the graphs is shown in Fig. 1.17 (on the left). Specifically, the input signal is shown in blue, while the output signal produced in response to the stimulation of the neurons is shown in black. Observe how, from the moment the input signal takes on a positive value, the output signal grows steadily. Vice versa, when the input signal takes on a negative value, the output signal decreases proportionally. In the time intervals in which the input signal takes the value 0, the integrator assumes instead maintains its constant value, as it is logical to expect. In Fig. 1.17 (on the right) the graph of the output signal produced by the same neural model is also shown under the stimulus of a different input function.

Because the integrator was built through a neural model, it does not behave perfectly. By running the simulation several times it is possible to highlight the errors produced by the model. These errors can be significantly reduced by increasing the number of neurons in the population.

2

The Semantic Pointer Architecture

This chapter details the cognitive architecture called *Semantic Pointer Architecture* (SPA). Like any cognitive architecture, SPA represents only a hypothesis of how concepts are represented in our neural system and of the functioning of cognitive processes that lead to intelligent behaviors [129].

2.1 The semantic pointer

At the basis of the SPA is the hypothesis of the existence of semantic pointers. The purpose of introducing this hypothesis is to bridge the gap between the neural structure used in the NEF, based on the idea that a wide variety of functions can be implemented in neural structures, and the domain of cognition (which needs ideas on how such neural structures give rise to complex behaviors). In SPA, the syntax is inspired by supporters of the *symbolic approach* who argue for the presence of syntactically structured representations in the head, while the semantics is inspired by connectionists who argue that vector spaces can be used to capture important semantic features .

2.1.1 Physical characterization of a semantic pointer

Semantic pointers are a means of linking more general neurological representations (symbols) to the central ones of cognition. They can be generated from raw perceptual inputs or can be used, for example, to guide a motor action. Specifically, according to this hypothesis, higher-level cognitive functions in biological systems are made possible by such pointers. Such pointers are neural representations that carry a partial semantic content and are composable in the representational structures necessary to support a more complex process of cognition.

2.1.2 Mathematical characterization of a semantic pointer

A semantic pointer is mathematically characterized as a *vector* of dimension n, that is, as an element of a vector space n -dimensional. A n -dimensional vector is made up of a sequence of n numeric values, called components of the vector.

For example, the sequence $(2,6,3)$ represents a vector in the vector space at 3 dimension; the sequence $(2,-4,0)$ represents a vector in a space of 2 dimension; while $(6,1,2,-3,-8)$ represents a vector in vector space at 5 dimension.

To understand what a vector is and how it can be represented graphically, let's consider the 2 dimensional vector space ($n = 2$) represented by the Cartesian plane, very familiar to us. In Fig. 2.1 (on the left) the Cartesian plane is shown in which three vectors are highlighted. They are graphically represented as segments starting from the origin (the point $(0,0)$) and ending in the point (x,y), characteristic of each vector. Each vector in the 2 dimension plan is therefore uniquely defined by its 2 components: x and y. The three vectors represented in Fig. 2.1 (on the left) are the vectors $(2,2)$, $(-2,3)$ and $(-4,1)$. In a 3 dimension vector space, like the one shown in Fig. 2.1 (right) each vector is defined by a set of 3 numbers (x,y,z). The vector represented in the figure is defined by the triple $(5,8,3)$.

A high-dimensional vector space is a natural way of representing semantic relationships in a cognitive system. Although it is difficult to visualize high-dimensional spaces, we can understand how this statement can work in the representation of concepts in a three-dimensional space (a relatively low-dimensional space). We refer to this represen-

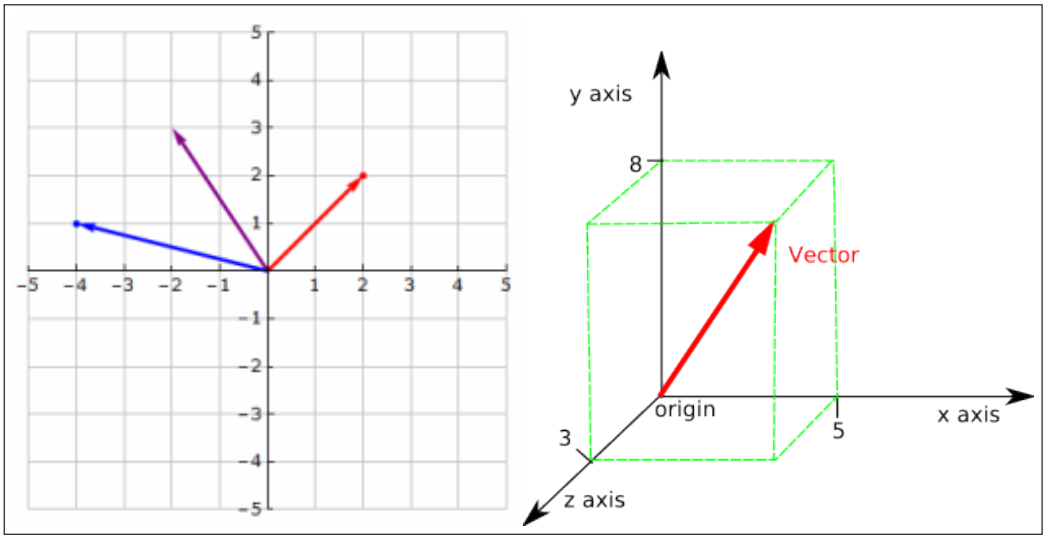


Figure 2.1: The mathematical representation of a vector. On the left is a Cartesian plane (a 2-dimensional vector space) in which the three vectors $(2,2)$, $(-2,3)$ and $(-4,1)$ are represented, each of which is defined by a set of two numbers (x,y). In a 3-dimensional space, such as the one shown on the right, each vector is defined by a set of 3 numbers (x,y,z). The vector represented in the figure is defined by the triple $(5,8,3)$.

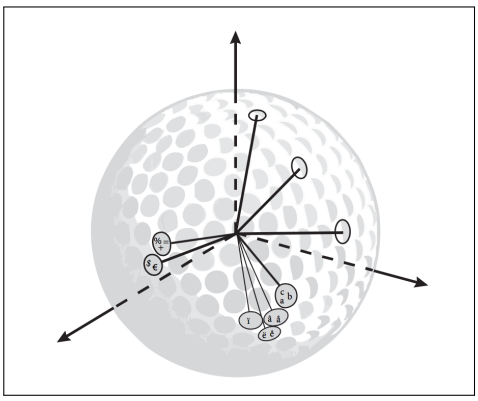


Figure 2.2: A conceptual golf ball. The surface of the ball represents conceptual space and the dimples on the surface represent concepts. Specific examples of a concept are identified by points that are within a given dimple. Semantically similar concepts (and examples) are found in relatively close proximity to semantically dissimilar concepts.

tation as a "conceptual golf ball", like the one shown in Fig. 2.2. The surface of the ball represents conceptual space and the dimples on the surface represent concepts. Specific examples of a concept are identified by points that are within a given dimple. Semantically

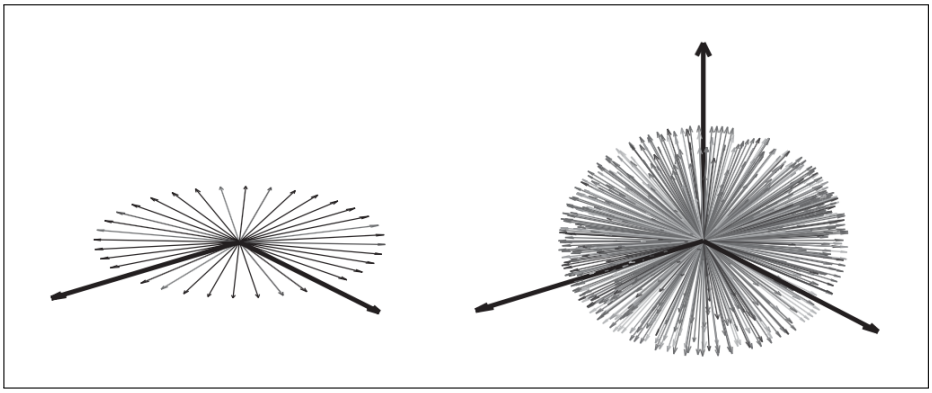


Figure 2.3: The conceptual spaces available on the surface of a hypersphere with a given dimensionality. The figure shows the number of unit vectors that can be inserted into a disk (on the left) and a ball (on the right), under the constraint that there must be a minimum angle of ten degrees between any two vectors. The disc will contain at most 36 vectors while the ball will contain 413 vectors.

similar concepts (and examples) are found in relatively close proximity to semantically dissimilar concepts.

However, the semantics represented by even simple human concepts appear to be extraordinarily rich, extending beyond simple mapping to a single vector space, albeit at a very high dimension. Multi-dimensional vector spaces allow the presence of a greater number of vectors and, consequently, the possibility of defining a wider set of concepts.

Fig. 2.3 shows the conceptual spaces available on the surface of a hypersphere with a given dimensionality. The figure shows the number of unit vectors that can be inserted into a disk (on the left) and a ball (on the right), under the constraint that there must be a minimum angle of ten degrees between any two vectors. The disc will contain at most 36 vectors while the ball will contain 413.

These scaling considerations suggest that 500-dimensional semantic pointers should be appropriate for human cognitive simulations.

2.1.3 Functional characterization of a semantic pointer

Although a semantic pointer can be represented through a mathematical vector, it takes its name from the world of computing. This is because its functional characteristics are very similar to those performed by a computer pointer.

A *pointer*, in computer science, is a number that indicates the address of information stored somewhere in the computer's memory. What is interesting is that operations can be performed on the pointers themselves which result in indirect operations on the information identified by the pointer itself, although the information object of the operation is never explicitly used. Most of these manipulations are pretty straightforward. For example, if I need to pass a data structure to a function that can use it, I can simply pass the pointer to it, which is typically much smaller than the data structure itself. As a result, far less information has to be moved around the system, while still making relevant data available for later use.

Consequently, having the information pointer available does not indicate anything about the type of information that will be found when you access the address it points to. To access the information and data present at the address specified by the pointer, it is necessary to *dereference* the pointer.

On the basis of these principles it is possible to state that pointers perform a symbolic function, that is, they operate as symbols. Symbols, after all, derive their computational usefulness from the arbitrary relationship they have with their contents. And symbols are often meant to act as labels for more sophisticated data structures (such as schemas, scripts, etc.), just as pointers act as labels for whatever is present at their address.

In short, the semantic pointer hypothesis suggests that the brain manipulates compact, address-like representations to exploit the efficiency and flexibility offered by such representations. In connection with this, such neural representations are able to act as symbols in the brain.

However, the arbitrary relationship between a pointer and its content is problematic for biological systems, because the relationships between neural representations are more often learned. In contrast, the relationships between symbols in a digital computer are determined by human design decisions. It is precisely the inability to adequately capture semantics that raises what has become known as the *symbol problem* (Harnad, 1990).

Instead, the hypothesis suggested in this theory is an extension of the standard notion of a computer pointer. In particular, the term *semantic* in a *semantic pointer* refers to the fact that the representations that play the role of a pointer themselves contain semantic information. That is, the relationship between a semantic pointer and the more sophisticated

data structure it points to is not arbitrary. Specifically, the semantic information contained in a semantic pointer is usefully thought of as a *compressed* version of the information contained in a more complex representation.

2.1.4 The semantic pointer at a glance

Putting these previous considerations together, semantic pointers are neural representations that are appropriately described by high-dimensional vectors, are generated by the compression of more sophisticated representations, and can be used to access those more sophisticated representations through decompression (a kind of dereferencing) .

A semantic pointer can therefore be described by the following three characterizations:

1. *physical characterization*: semantic pointers are activities that occur in a biological neurological network;

2. *mathematical characterization*: semantic pointers are described by vectors in a high-dimensional space;

3. *functional characterization*: semantic pointers are compressed representations that point to more complete semantic content.

While it may be possible, and even useful, to talk about SPA in this way, semantic pointers themselves are neural representations. Treating them mathematically, physically or functionally does not mean treating them at different levels of detail.

2.2 Deep semantics and superficial semantics

In psychology, interest in semantics is often expressed by asking questions about what kind of semantic processing is needed to perform different cognitive tasks. In short, the question of interest becomes: in what respects do we need *deep* semantic processing and which can be effectively justified by *superficial* semantic processing?

The distinction between deep and superficial processing can be traced back to Allan Paivio's theory of double coding [91, 90]. This theory suggests that different information,

such as perceptual and verbal information, are processed in distinct semantic channels: *deep semantics* and *superficial semantics*, in fact.

Much of the semantic processing we perform on a daily basis appears to be of a profound type. Typical deep semantic processing occurs when language processing or execution of specific movements occurs. Other semantic elaborations, such as perceptive ones, which allow us to correlate symbolic concepts to specific signals from the outside world, are managed by a superficial semantic level.

Several studies have shown that superficial semantic processing is much faster than deep semantic processing, and therefore it is not surprising that, after a perceptual stimulus, concepts that are statistically more highly correlated are listed first. Deep processing takes longer, but offers a richer characterization of the meaning of the concept.

Given the important distinction between deep and superficial semantics and the evidence that these two levels are processed differently in the brain, a central question is: "How can both deep and superficial processing be incorporated into the model?".

The central hypothesis of the SPA outlines the answer to this question: semantic pointers contain partial semantic information. At this point it is necessary to develop the following aspects: describe in detail how the partial semantic information carried by semantic pointers is generated (and therefore how they are able to capture surface semantics) describe how to use semantic pointers to access the deep semantics to which they are related (i.e. how to dereference pointers).

2.2.1 Characterization of partial semantics

As already specified above, the surface semantics captured by a semantic pointer can be thought of as a kind of *compressed* representation of complex semantic relationships.

Compression (in information theory) is the technique that allows a more compact representation of data, thus using a smaller amount of space. Compression can come in two forms: the first is the *lossless* compression, used in the well-known `zip` method, which allows the compression of data that can then be perfectly reconstructed by applying an inverse procedure to compression. The second form is *lossy* compression, such as the well-known `jpeg` method for compressing images. It involves higher compression at the

cost of losing some of the information in the original image. This means that it is no longer possible to obtain the original information from its compressed version.

In this context, semantic pointers are assumed to use lossy compression. For this reason we speak of *partial semantics* information carried by semantic pointers. They only provide an indication of the semantic concept to which they refer. To obtain a more detailed representation of this concept it is necessary to de-reference the pointer, that is to access the information to which the pointer refers.

2.2.2 Hierarchical structure of semantic cognitive processes

The process of generating surface semantics allows for a representation that can remain closely linked to deep semantics in a neural architecture.

We can begin to formalize this by assuming that there are some visual data (e.g. an image), y, which is generated by the external visual world and drives neural activity. If we assume that the purpose of perceptual systems is to build and use a statistical model of this data, then the system must understand a function $p(y)$ that describes the probability of each state, so that it can use that information to disambiguate future data.

The brain probably approaches this distribution by constructing what is called a *parameterized model*. Such a model identifies a small number of parameters that capture the overall shape of the ideal model, thus giving only an approximation of the latter. For example, if the data statistics are in a famous Bell (or Gaussian) curve, we can model the data with an equation like:

$$p(y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(y-\hat{y})^2/2\sigma^2}$$

So, to *capture* all the data passed using our model, we just need to remember two parameters, $\hat{y}$ (the mean) and σ (the standard deviation), and the equation that describes their relationship. This is much more efficient than explicitly remembering each value of $p(y)$ for each value of y.

Perhaps the best known example of such a structure in neuroscience is the *visual hierarchy*. For object recognition, this hierarchy begins with the retina and proceeds through the thalamus to the visual areas V1 (primary visual cortex), V2 (secondary visual cortex), V4 and IT (inferotemporal cortex) [32].

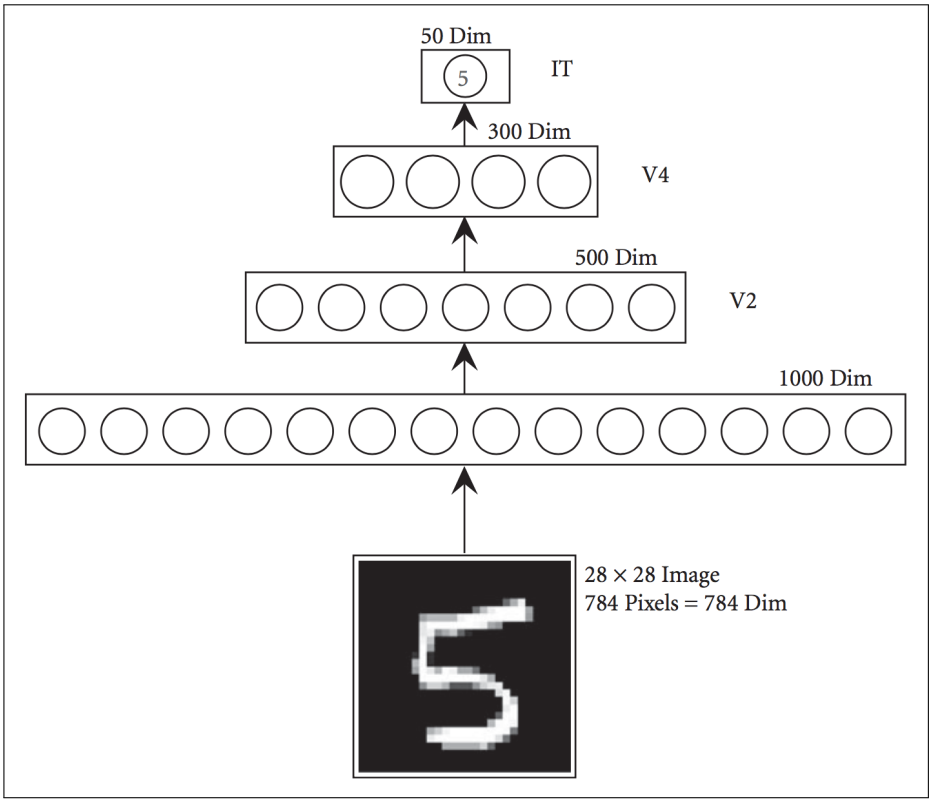


Figure 2.4: A hierarchical statistical model. An original image consisting of 784 pixels (its size is therefore $n = 784$) is compressed into a 50 -dimensional representation through a series of hierarchical statistical models. Each level can be thought of as corresponding to a level of the visual hierarchy for object recognition as indicated. The dimensionality of the state space at each level is indicated above the level itself.

Fig. 2.4 shows a hierarchical statistical model in which the original image composed of 784 pixels (its size is therefore equal to $n = 784$) is compressed into a 50 -dimensional representation through a set of intermediate hierarchical statistical models. Each level can be thought of as corresponding to a level of the visual hierarchy for object recognition as indicated.

In a hierarchical statistical model, each higher level in the hierarchy attempts to build a statistical model of the underlying level. Taken together, the layers define a complete model of the original input data. This type of hierarchical structure naturally allows for the progressive generation of more complex functions at higher levels and progressively acquires higher-order relationships in the data. Furthermore, such models embody relation-

ships between hierarchical levels that are reminiscent of the variety of neural connectivity observed in the cortex: feedforward, feedback, and recurrent (intercalary) connections are all essential. The power of these methods for generating effective statistical models is impressive [9].

For example, the representation in V1 is used to guide neurons in V2, and so on up the hierarchy. By the time we get to the top level of the hierarchy, we get a much smaller (i.e. compressed) dimensional representation that summarizes what was presented at the lowest level. This compressed representation is nothing more than a semantic pointer.

Once this statistical model is built, we can map it to a mechanical model using the NEF. The structure of this statistical model is inspired by a specific cortical mapping, and so the NEF can respect that mapping, while determining how neurons can implement the necessary underlying calculations.

We can therefore say that capturing the representation of a deep semantics and linking it to a higher-level representation solves the problem of symbolic representation, if we can show how those higher-level representations can function as symbols (this problem will be addressed later).

If this hypothesis is plausible, then the representational structure underlying the SPA should be clear: the semantic pointers, generated by a perceptual processing, can be compressed through a form of processing and used to obtain a form of superficial semantics by acting, at the same time, as a symbol. When it is necessary to obtain the deep semantics of such a representation, then the semantic pointer can be used to stop the upper layer of the perceptual network that generated it so that the network can reconstruct the vector representation that carries its deep semantics.

Fig. 2.5 shows a hierarchical statistical model of the functioning of this semantic theory implemented in our neural model. The model shown in the figure uses the `jpg` compression metaphor to simulate the compression of semantic pointers from lower to higher levels. The highest level is represented by the most superficial vector space, that of symbols. The lowest level represents the vector space of deep semantics. Each element of the i level allows to obtain an element of the $i+1$ level (highest level) through a compression process, obtaining a more compact representation of it. In a very similar way, each element of the i level allows to obtain an element of the $i-1$ level (lowest level)

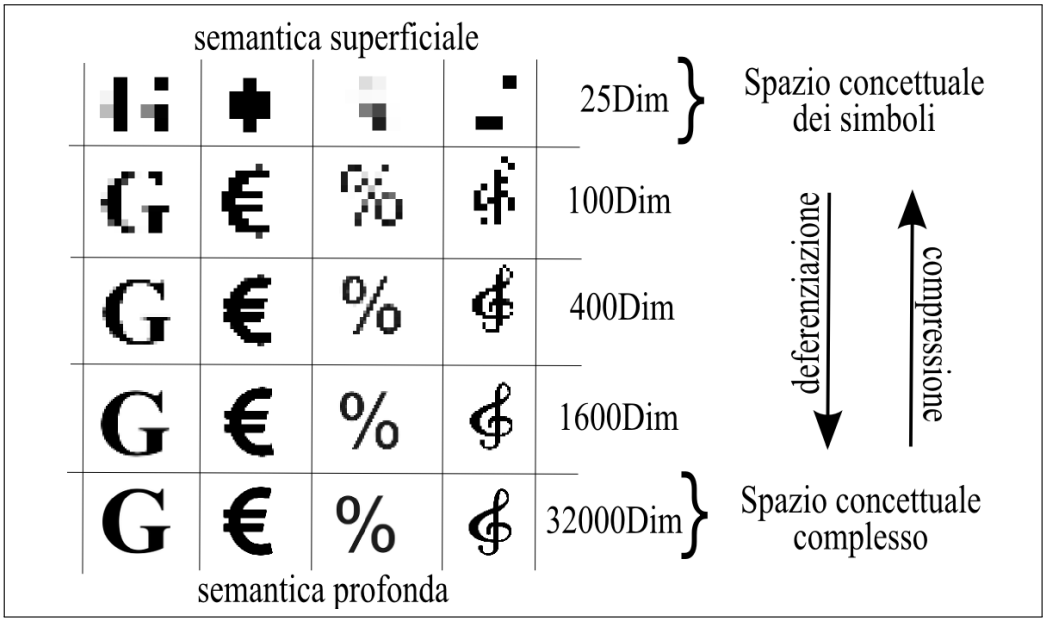


Figure 2.5: A hierarchical statistical model of the functioning of semantic theory implemented in our neural model. The model shown in the figure uses the `jpg` compression metaphor to simulate the compression of semantic pointers from lower to higher levels. The highest level is represented by the most superficial vector space, that of symbols. The lowest level represents the vector space of deep semantics. Each element of the i level allows to obtain an element of the $i+1$ level (highest level) through a compression process, obtaining a more compact representation of it. In a very similar way, each element of the i level allows to obtain an element of the $i-1$ level (lowest level) through a process of deferencing typical of computer pointers, obtaining a more detailed representation of it.

through a process of deferencing typical of computer pointers, obtaining a more detailed representation

It is worth noting that until now we have used the term *semantic pointer* to indicate only the top of these hierarchies. However, it must be stressed that this is a simplification used to help introduce this notion. Clearly, the representations on the other levels of the hierarchy are also compressed representations, albeit with a higher dimension (than the first in the hierarchy). Such representations can also be used by other parts of the system as pointers to more detailed semantic information. For explanatory purposes, and perhaps for efficiency reasons, the most compressed representations are *paradigmatic semantic pointers*, but they don't necessarily have to be the only type.

Three crucial characteristics of this model are therefore useful for understanding the

characterization of the semantics provided by the SPA. In particular:

1. the model constructs high-level representations of the perceptual input that acquire statistical relationships that describe the structure of the domain;

2. such representations are compressed (ie, smaller) representations of the input; is

3. such representations can be dereferenced to provide detailed perceptual information.

2.2.3 Dual semantic processes

It is interesting to note that these same three characteristics can be useful for characterizing the biological representations that guide the motor system. However, the task of motor systems seems very different from that of perceptual systems. The motor system, in fact, does not need to classify the stimuli presented. Indeed, while perceptual systems often need to map a high-dimensional and ambiguous state (e.g., images generated by highly nonlinear environmental processes) to a much smaller set of states (e.g., categories of objects), systems Motors must often map a small set of states (e.g., desired achievement goals) to an ambiguous, high-dimensional state (e.g., any one of the possible configurations of muscle strains on the body that results in a target). This allows us to notice how these two tasks are almost opposite.

Sometimes, however, the opposition is a highly informative relationship. In mathematical terms, problems that share their structure in this way are called *dual problems*. It is helpful to identify the duality between problems since if we know how to solve one of these problems, then we can solve the other.

For example, suppose you want to move your hand towards a given object. Once the desired target state has been identified, it is delivered to the cortical motor system. The first level of this system can then determine which direction the arm must move to reach that target state. Once the direction of movement has been determined, it can be used as a control signal for a controller that is further down the hierarchy.

Of course, specifying the direction of movement does not determine how twists are applied to the joints of the arm that perform that movement. This, then, would be the

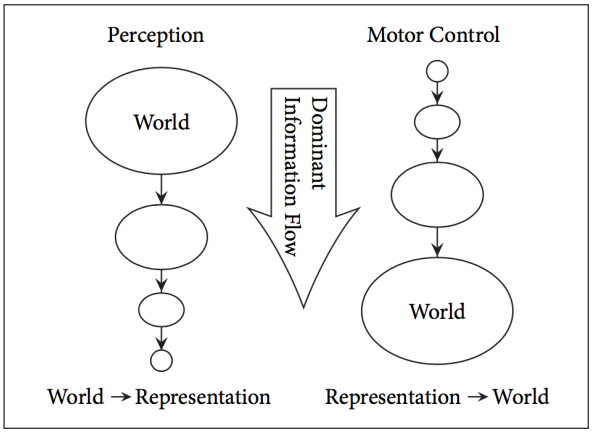


Figure 2.6: The perceptual and motor systems, although very complicated, can be treated as dual. The perceptual system encodes relationships using hierarchically compressed representations. The motor control system reverses this process (albeit using similar structure and representations) to determine high-dimensional nonlinear control signals from a low-dimensional signal.

function of the next lower level in the control hierarchy. Once the specific forces required to move the arm have been determined, the specific tensions that must be produced in the muscles to realize those forces must be determined by the next lower level in the hierarchy. Ultimately, this progression results in motor neuron activity causing muscle contraction and arm movement.

There is several evidence that this type of control structure is actually present in our brain [141, 55, 89], i.e. a control structure in which each level of the hierarchy has explicit models of the behavior of other levels.

Notice how both features depend on a hierarchy that is *executed* in both directions. In the forward direction (up the hierarchy), the perceptual hierarchy allows for the classification of visual stimuli; in the opposite direction, it allows the generation of deep semantics. While I haven't discussed it in detail, both directions also help the system learn the appropriate representations. For the motor hierarchy, the forward direction (down in the hierarchy) allows for the control of a sophisticated body using simple commands, as well as the generation of deep semantics. The reverse direction allows online adaptation of the motor control to various perturbations.

Again, both directions will allow the system to learn appropriate representations (i.e.,

synergies). So it is clear that in both cases, the semantic pointers operating at the top of the hierarchy can be thought of as being attached to a *memory* and can be used in the hierarchy to obtain details of that memory.

Of course, this use of the term *memory* is, unlike a computer, necessarily constructive. We can make moves that we have never done before, and we can imagine writing something we have never seen before. This observation supports the notion that perceptual and motor systems are well characterized in building statistical models of the inputs received in the past.

3

A Computational Model of the Neural Circuits for Contextual Behaviour

We argue that the default modus operandi of the brain is to process each input in relation to the context in which it occurs. Inputs to the brain are not represented in their absolute values or categories. Instead, representations of inputs are made exclusively in relation to other inputs-past and present. These contextual comparisons operate both in time and in space, comparing the inputs either against the past or against other locations. By comparing rather than isolating, the brain works predominantly with discontinuities in the inputs, charging its computational power by the changes and differences in sensory activations.

In this chapter we analyze the role of context in some of the main activities of the brain, focusing our attention on the item-item association and the itme-context one. We take a closer look at the anatomy of the area of the brain that appears to be involved in contextual behavior and, specifically, the hippocampus and the CA3 region. We then provide a model based on NEF and SPA that can implement some of the elementary brain functions related to contextual behavior.

From our preliminary experimental results it seems that the proposed model is able to respond adequately, with a certain precision, to the stimuli received in input.

3.1 The Brain as a Context Machine

The contextual nature of brain functioning can be detected already at the level processing of a single receptor cell. A receptor cell transforms the physical input - e.g., light, sound, temperature, mechanical force-into neural signals. While doing so, receptors observe the temporal context. The strength of a response to a given input will depend typically on the strengths of the preceding inputs. If the new input is considerably stronger than the past ones, the response will be vigorous. If this difference is small, so will be the response. This is explained by the phenomenon of adaptation. If a cell is exposed to a constant high-intensity stimulus, the vigor of the response reduces. The cell adapts to the stimulus. The strongest outputs are generated not necessarily by the stimuli of highest absolute amplitudes, but by those imposing largest changes to the previous inputs. Sensory cells are not the only ones that adapt. Neurons in the central nervous system adapt well too. An orientation sensitive cell in the primary visual cortex will respond with high firing rates if a stimulus of a novel, not previously presented orientation is shown.

At the stimulus onset, a cortical cell will respond with high firing rate. However, as the stimulus remains unchanged, this vigorous response ceases quickly, usually within less than 100 ms - as if the stimulus is interesting for only a short time period and only its change is relevant. Interestingly, when the stimulus is removed, the cells in the primary visual cortex will respond vigorously again (i.e., off-response), with firing rates often higher than that to the initial appearance to the stimulus [86]. This illustrates the significant extent to which the past context affects the responses to the present stimuli. Apparently, the past tells the brain how much the present matters.

In general, it appears that the basic circuitry of the brain is utterly incapable of maintaining the absolute information about the physical world (e.g., temperature, mechanical force, light intensity, etc...). Instead, the brain has no other source of information but that presented relative to the context in which it emerged. Evidently, when a big brain gets wired, the result a massive context machine.

The eye movements during a fixation may be an example of such an active quest for change. Our eyes, when healthy, make micromovement with a frequency of 30-100 Hz and of miniature amplitude, which seem to play an essential function in enabling visual

perception. If these movements are abolished (actually their effects are abolished through innovative optic devices), our perception of the world fades away [81]. If the surrounding world does not change, the brain makes it change. A constancy of the input seems to bring the perceptual fire to a still. The dynamics of change seems to keep marry-go-round.

Animals including humans engage in goal-directed behavior flexibly in response to items and their background, which we call contextual behavior [67]. Although the concept of context has long been studied, there are differences among researchers in defining and experimenting with the concept.

Contextual behavior can be categorized into three subcategories as follows by considering the types of interactions among context, item, and response:

- *Contextual response selection*, which refers to the emission of different types of responses to the same item depending on the context in the background.
- *Contextual item selection*, which occurs when there are multiple items that need to be chosen in a contextual manner.
- *Contextual item-response*, which occurs when multiple items and multiple contexts are involved and the selection takes place whereby the animal either chooses an item or inhibits such a response depending on item-context paired association.

Context, therefore, provides a unifying cognitive framework in the brain from stimulus interpretation to behavior. In the current review, we will define the term context as follows. Context refers to an external physical stimulus that is present in the animal's background and is multiplex (i.e., composed of complex elemental information), and it should be directly associated with purposeful behavior.

3.2 Neural Circuits for Contextual Behaviour

In this section we schematically discuss the information flow and interactions among the main brain regions involved in contextual behavior, as emerged by recent advancement in neurosciences. Although the circuits and the types of information processed are often discussed from the viewpoint of spatial-nonspatial information processing for memory, it has

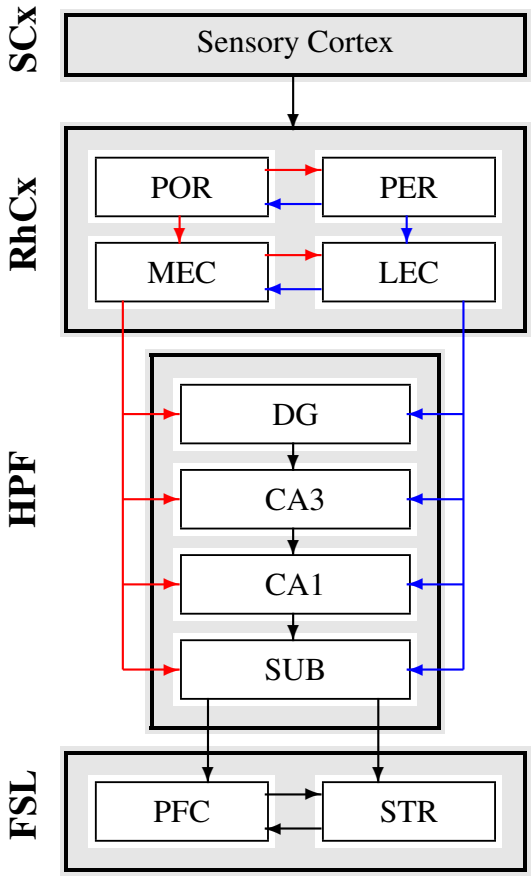


Figure 3.1: A schematic illustration of the information flow and interactions among the brain regions (selectively chosen) involved in contextual behavior (connections are simplified for illustrative purposes).

been argued [67] that the same circuits are involved in processing background contextual information and item information within the context.

The literature suggests [33, 59, 34, 56, 24] that the Rhinal Cortical regions (RhCx) and the Hippocampal Formation (HPF) play key roles in mnemonically categorizing and recognizing contextual representations and the associated items. In addition, it appears that the Fronto-Striatal cortical loops (FSL) in connection with the contextual information-processing areas critically control the flexible deployment of adaptive action sets and motor responses for maximizing goals. Fig. 3.1 shows a schematic picture of such information flow and interactions.

Via primary and higher-order sensory cortices (SCx), multimodal perceptual informa-

tion from the environment enters the RhCx. Specifically the RhCx includes the post-rhinal cortex (POR), the perirhinal cortex (PER), the medial entorhinal cortex (MEC), and the lateral entorhinal cortex (LEC). The RhCx is used for explicit (episodic and semantic) memory. In addition POR and MEC are involved in scene memory and spatial memory while PER and LEC are involved in visual memory and item memory.

Such areas in the RhCx may interact with each other to varying degrees as indicated by arrows. It is hypothesized that qualitatively different information-processing streams exist in the RhCx, denoted by red arrows (contextual information) and blue arrows (item information). The qualitatively different information streams continue as the information enters the hippocampal formation (HPF) in which associative processes occur between these two streams.

The hippocampus, including the dentate gyrus (DG), has the shape of a curved tube, which has been compared to a seahorse, and a ram's horn (Cornu Ammonis). Its abbreviation CA is used in naming the hippocampal subfields CA1, CA2, CA3, and CA4 [4]. The hippocampal subfields (DG, CA3, CA1) and subiculum (SUB) all receive contextual-noncontextual information in parallel. Serial information processing across the hippocampal subfields to SUB also occurs at the same time. Apparently, the CA3 (with the conjoint effort of DG) plays key roles in recognizing the original as well as modified contextual environments in comparison to memory representations [80, 112]. The subiculum has been instead recently involved in working memory [109].

The information regarding the contextual interpretation of the environment and its associated items then interacts with the fronto-striatal networks (FSL) including the prefrontal cortex (PFC) and striatum (STR) in a goal-directed manner before final response behavior is determined. The FSL is then involved in planning decision making, and specifically the STR serves as input for the Basal Ganglia and is involved in reward recognition and in motor and action Planning.

The four regions (SCx, RhCx, HPF, and FSL) interact with each other via various feedforward and feedback connections to realize coherent, inter-regional bottom-up and top-down communications toward goal-directed behavior.

3.3 Role and Anatomy of the CA3 Region

The hippocampus has a remarkable anatomical structure. It is defined by the dentate gyrus (DG) and Cornu Ammonis (CA). The CA is anatomically and functionally differentiated into distinct subfields named CA1, CA2, CA3 and CA4. Computations that occur in the hippocampal subfields have been experimentally investigated explosively in the last 10 years although most studies heavily focused on spatial contextual representation. The CA3 region has attracted major attention in recent years [16] for its specific role in memory process, susceptibility to seizures and neuro-degeneration.

It would be rare in natural settings to encounter the same context in exactly the same physical conditions every time an animal experiences the context repeatedly. this can be due to differences in lighting conditions, viewpoint, degradation or loss of some elements in the context, etc. To use contextual memory and learned contextual behavior, however, is critical to reliably identify the same context despite some minor changes in its surrounding as well as to recognize some significant differences. The hippocampal circuits appear to perform such critical computations.

The hippocampus is often implicated in forming new associative memories, storing memories independently of each other, retrieving memories from partial cues, and flexibly applying stored memories to novel situations. David Marr [79] was the first to suggest that recurrent collaterals enable a region to act as an auto-association network capable of pattern completion, the process by which incomplete or degraded representations are filled-in based on previously stored representations. Pattern completion allows for accurate generalization in the face of noise or partial sensory input.

The CA3 subfield, for example, contains auto-associative networks whereby pattern completion recovers the learned contextual representation despite some modification/ loss/ noise in the original context [79, 83, 88, 68]. The same network appears to perform non-linear operations to orthogonalize similar, yet significantly different contextual information into a separate contextual memory (pattern separation) and the DG subfield seems also critical during these processes

Internal connectivity in the CA3 subfield is more rich than in other hippocampal regions. It contains pyramidal neurons, a type of multipolar neurons found in areas of the

brain including the cerebral cortex, the hippocampus, and the amygdala. Such pyramidal neurons are organized with recurrent axon collaterals of CA3, i.e. axons that circle back to the dendrites of cells within the same region forming a recursive feedback loop. Such pyramidal neurons ramify extensively making excitatory contacts with neighboring excitatory and inhibitory neurons [70].

This circuit is implicated in encoding spatial representations [87] and episodic memories [119]. Computational models suggest that the dentate gyrus and CA3 subfields of the hippocampus are responsible for discrete memory representations by using pattern separation and pattern completion when a modified external stimulus is recognized as an old memory or encoded as a new memory [1].

The CA3 region receives inputs from the entorhinal cortex either directly via the perforant path or indirectly from the dentate gyrus via the mossy fibers [3].

Kesner [57] reviews contributions from the CA3 region subfields CA3a, b and c in acquiring and encoding spatial information. Mnemonic functions depend on synaptic interactions of CA3 associative networks, operating as an attractor, with inputs from the dentate gyrus and entorhinal cortex [58]. Fields CA3a and b encode spatial information in short-term memory and also support retrieval by spatial pattern completion. In contrast, the CA3c field may support pattern separation via interactions with the DG. The output subfields, CA3a and b process information sequentially communicating with the CA1 region via the Schaffer collaterals.

The CA3 system, through its recurrent collateral connections, is presented as a single attractor enabling fast, one trial, associations between any spatial location and an object or reward [114]. A recurrent structure permits memory completion during recall from any subset of acquired links. This theory permits associations between time, objects and rewards to provide the temporal order needed for episodic memory.

Cerasti and Treves [14] discussed how self-organizing recurrent connections enable spatial representations to be acquired in the CA3 area. A simplified network model shows that self-organization can lead to the emergence of multiple loosely organized discrete point-like attractors which differ from structures associated with a single, continuous attractor.

The view is that the CA3 region functions as an auto-associative memory [80, 83,

130, 88]. An auto-associative memory is a recurrent network that stores patterns in its feedback connections and can reconstruct these patterns when only a partial version of them is presented. Thus, the actual storing place are the recurrent connections and this idea could explain why there are so remarkably many in CA3.

An auto-associative memory can only store patterns that are not similar or mutually correlated [80]. By nature, however, the neural activation in the input region of the hippocampus, the entorhinal cortex (EC), is not uncorrelated [52]. Thus, it has been suggested that the DG performs pattern separation during the storage phase [83, 130, 88]. It decorrelates the patterns of the EC and projects the separated versions of the patterns to CA3 for storage. A large number of cells with low activity and the sparse projection of mossy fibers support pattern separation computationally [113, 131]. Hence, this view explains the appearance of yet other prominent hippocampal characteristics. Finally, it has been proposed that the role of CA1 is to decode the highly transformed patterns in CA3 back to their original versions in the EC.

It has been also suggested [23] that the hippocampus and its subregions support associative processes as observed in paired-associated learning tasks. In [49, 111] it has been suggested that the hippocampus and specifically the CA3a and b auto associative network is responsible for the formation and storage of associations.

3.4 A SPA Model of the CA3 Region

In this section we describe the circuit of our simplified model of the CA3 hippocampal region involved in the retrieval of semantic relations between two items (item-item association) and between an item and a context (item-context association).

3.4.1 Paired Item-Item and Item-Context Association Path

Semantic relation between concepts is a well understood phenomenon in cognitive science. It is assumed that when two concepts, x and y, are thought simultaneously, they may become linked in memory. Subsequently, when one thinks about x, then y is likely to come to mind as well. Thus multiple links to a concept in memory make it easier to be retrieved because of many alternative routes to locate it.

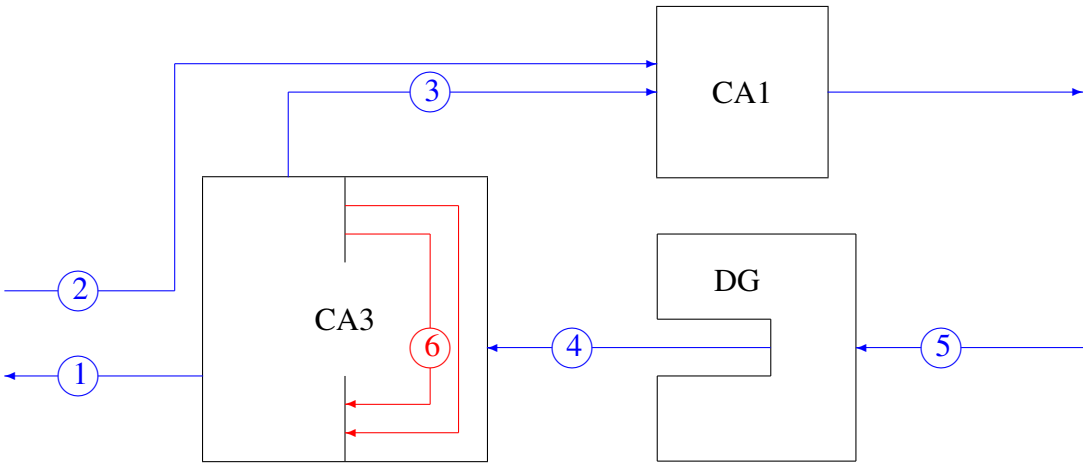


Figure 3.2: A schematic representation of the hippocampal tri-synaptic circuit. Neurons of the EC project to the DG along the perforant path (5). Granule cells in the DG project to the CA3 field of the hippocampus via the mossy fiber (4). The CA3's pyramidal cells project heavily onto themeselves via recurrent collaterals (6) (depicted in red) and also to the CA1 through Schaffer collaterals (3). The fimbria (1) is the principal output pathways of the hippocampus that also brings in commissural (2) input from the collateral hippocampus.

However, such similarity doesn't define a symmetric semantic relation between x and y. In real world representation of associative networks relations are, indeed, represented by direct edges, i.e. the measure of the relation between x and y could be different from the measure of the relation between y and x. thus In our model we assume unidirectional semantic associations between two concepts.

For example, the concepts "gasoline" and "car" are undoubtedly related in real world, thus if we think to gasoline the concept of "car" comes to mind with a great probability. However the contrary is not true, if we think to a car probably other concepts come to mind with higher probability, like "road" or "parking". So we can say that "gasoline" is more related with "car" than viceversa.

Going in depth into details, in our model based on the SPA an item-item association between a source item x and a target item y, both identified by two semantic vectors, is achieved by binding x to a special semantic vector, SOURCE-ITEM, and by binding x with a special semantic vector TARGET-ITEM. Such bindings are then superimposed to obtain the semantic relations between these two items.

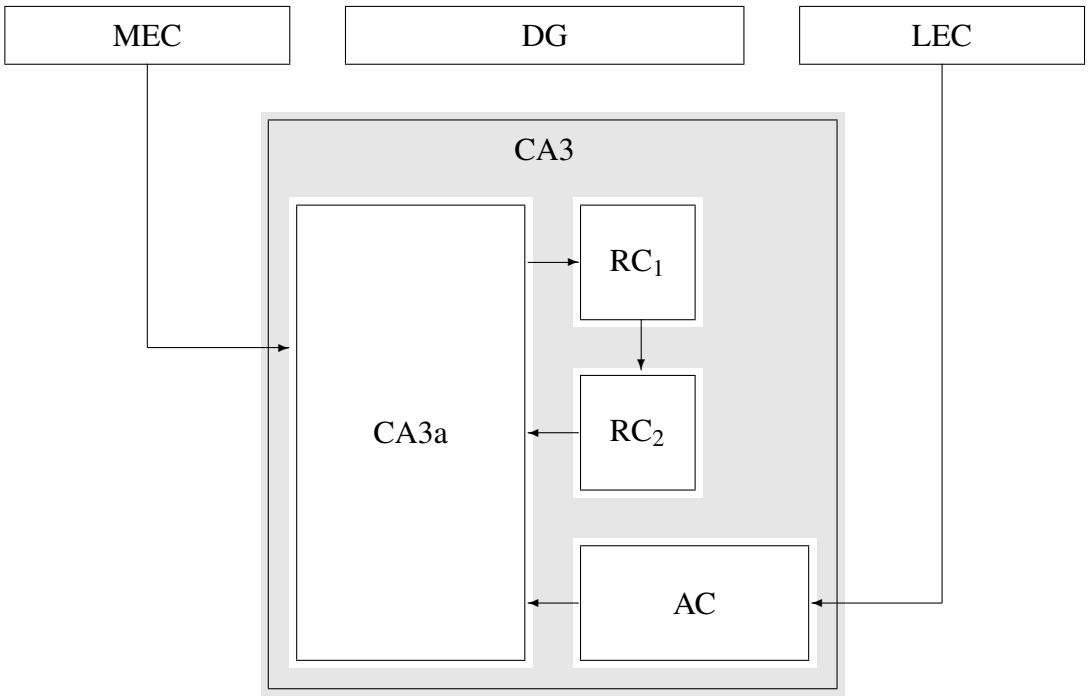


Figure 3.3:

Formally we assume that such association, from item x to item y, is stored in memory as a semantic vector defined as

$$x \otimes \text{SOURCE-ITEM} + y \otimes \text{TARGET-ITEM} \tag{3.1}$$

Similarly an item-context association between a source item x and a target context C is achieved by binding x to a special semantic vector, SOURCE-CONTEXT, and by binding C with a special semantic vector TARGET-CONTEXT. Such bindings are then superimposed to obtain the semantic relations between the Item and the corresponding context.

Formally we assume that such association, from item x to context C, is stored in memory as a semantic vector defined as

$$x \otimes \text{SOURCE-CONTEXT} + C \otimes \text{TARGET-CONTEXT} \tag{3.2}$$

A schematic representation of the Association Module of the CA3 region in the Hippocampus is depicted in Fig.3.4. It consists in two serially connected Components: the Item-Item Association component (IIA) and the Item-Conrtext Association component (ICA), both of them consistsing in two subfields.

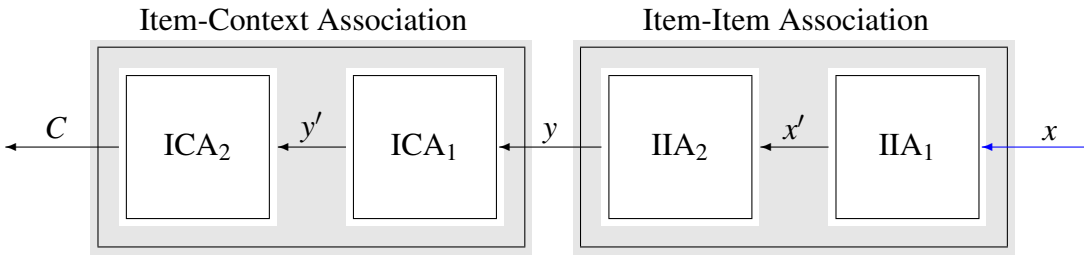


Figure 3.4: A schematic representation of the Association Module of the CA3 region in the Hippocampus. The input signal x enters the first Item-Item Association component (IIA_1); its output, the signal $x' = [x \otimes \text{SOURCE-ITEM}]^*$, enters the second Item-Item Association component (IIA_2); the output signal $y = [x' \otimes \text{TARGET-ITEM}^{-1}]^*$ enters the first Item-Context Association component (ICA_1); The output of ICA_1 is the signal $y' = [y \otimes \text{SOURCE-CONTEXT}]^*$, which enters the second Item-Context Association component (ICA_2); the output of the whole module is the signal $C = [y' \otimes \text{TARGET-CONTEXT}^{-1}]^*$

Assume that the sensorial composition of item x, as input signal INPUT, is sent by the LEC region and received as input by the IIA_1 component. In order to perform a transfer from item x to item y we use the following two step operations. First in IIA_1 the vector INPUT is bound with the vector SOURCE-ITEM. Subsequently the resulting output vector is unbound in IIA_2 using the following relation

$$\text{OUTPUT} = [\text{INPUT} \otimes \text{SOURCE-ITEM}]^* \otimes \text{TARGET-ITEM}^{-1} \tag{3.3}$$

where OUTPUT is the semantic vector associated to the target item y.

Assume now that the sensorial composition of item x, as input signal INPUT, is received as input. In order to perform a transfer from item x to context C we use the following two step operations. First we bind INPUT with the vector TARGET-CONTEXT, Then we unbind the resulting semantic vector using the following relation

$$\text{OUTPUT} = [\text{INPUT} \otimes \text{TARGET-CONTEXT}]^* \otimes \text{SOURCE-CONTEXT}^{-1} \tag{3.4}$$

3.4.2 Recurrent Collaterals Path

In the CA3 region of the hippocampus, pyramidal cells excite other pyramidal cells and interneurons. The axons of CA3 pyramidal cells spread throughout most of the region to form an associative network. These connections were first drawn in [118, 108]. Their

physiological properties were explored to understand epileptiform discharges generated in the region. Synapses between pairs of pyramidal cells involve one or few release sites and are weaker than connections made by mossy fibers on CA3 pyramidal cells.

The hippocampus is known to be involved in spatial learning and memory in rodents. Some of the most convincing evidence for this is the presence of place cells in areas CA3 and CA1 of the hippocampus and of many other types of spatially selective cells in neighboring areas

Strong constraints on the neural mechanisms underlying the formation of place fields in the rodent hippocampus come from the systematic changes in spatial activity patterns that are consequent on systematic environmental manipulations. We describe a network model of area CA3 in which local, recurrent, excitatory, and inhibitory interactions generate appropriate place cell representations from location- and direction- specific activity in the entorhinal cortex.

3.5 Preliminary Evaluation

In this section we show and discuss the results of a first preliminary evaluation of our schematic model of the CA3 hippocampal region, based on the SPA, obtained by testing the output of its components using a simplified small-world scenario consisting in two contexts and four distinct items.

To simplify vectors representation we assume that any concept of our small world scenario can be depicted by three different sensorial treats, and specifically a *shape*, a *scent* and a *color*. Such sensorial treats are represented in our associative memories by the semantic vectors SCENT, SHAPE and COLOR, respectively.

We consider two different Contexts, schematically identified by symbols A and B, defined as scene covered by a white sweet square and a scene covered by a black sharp rectangle, respectively. Formally the two contexts are represented in memories by the following two semantic vectors, CONTEXTA and CONTEXTB.

$$\begin{aligned} \text{CONTEXTA} &:= \text{SCENT} \otimes \text{SWEET} + \text{COLOR} \otimes \text{WHITE} + \text{SHAPE} \otimes \text{SQUARE} \\ \text{CONTEXTB} &:= \text{SCENT} \otimes \text{SHARP} + \text{COLOR} \otimes \text{BLACK} + \text{SHAPE} \otimes \text{RECTANGLE} \end{aligned}$$

We also consider four different basic items, schematically identified by symbols X, Y, Z and W. Specifically X is a sweet red circle, Y is a sharp blue rectangle, W is a sharp yellow point and Z is a sweet green point. Formally the four basic items are represented in memories as the following four semantic vectors.

$$\begin{array}{llll}
\text{ITEMZ} & := \text{SCENT} \otimes \text{SWEET} & +\text{COLOR} \otimes \text{GREEN} & +\text{SHAPE} \otimes \text{DOT} \\
\text{ITEMW} & := \text{SCENT} \otimes \text{SHARP} & +\text{COLOR} \otimes \text{YELLOW} & +\text{SHAPE} \otimes \text{DOT} \\
\text{ITEMX} & := \text{SCENT} \otimes \text{SWEET} & +\text{COLOR} \otimes \text{RED} & +\text{SHAPE} \otimes \text{CIRCLE} \\
\text{ITEMY} & := \text{SCENT} \otimes \text{SHARP} & +\text{COLOR} \otimes \text{BLUE} & +\text{SHAPE} \otimes \text{TRIANGLE}
\end{array}$$

Context similarity are identified by two different semantic vectors, SIMILARA and SIMILARB, referring to context A and context B, respectively. Roughly speaking our model identifies with vector SIMILARA (SIMILARB) any ambiguous context whose treats are recognized as similar those of context A (context B). Formally we assume that the representations of the following two semantic vectors are located in our memories.

$$\begin{array}{ll}
\text{SIMILARA} & := \text{SIMILAR} \otimes \text{CONTEXTA} \\
\text{SIMILARB} & := \text{SIMILAR} \otimes \text{CONTEXTB}
\end{array}$$

Once we have described the small world in which our experimental tesst take place, we evaluated the answer given in output by our model on four different tasks: the item-item response, the item-context response, the item-context response with related inputs, and the item-context response on approximate inputs.

As regards the first task, that is the item-item response, we have imagined a symmetric association of equivalent value between the items Z and X in relation with Context A, while we have assumed a relationship between the elements Z and W in relation with context B. Specifically, we assume that Z and X are closely related to each other in context A, while in context B the item input Z is closely related with item W. Such associations were implemented within the model memory using equations (3.1) and (3.2).

Just by way of example, this scenario could be easily described if we assume that in the context of scents (context A) Z is related with X since both are sweet. While in the context of shapes (context B) Z is related with W since both are depicted as dots.

Subsequently, once assumed that such associations are present in memory, the model

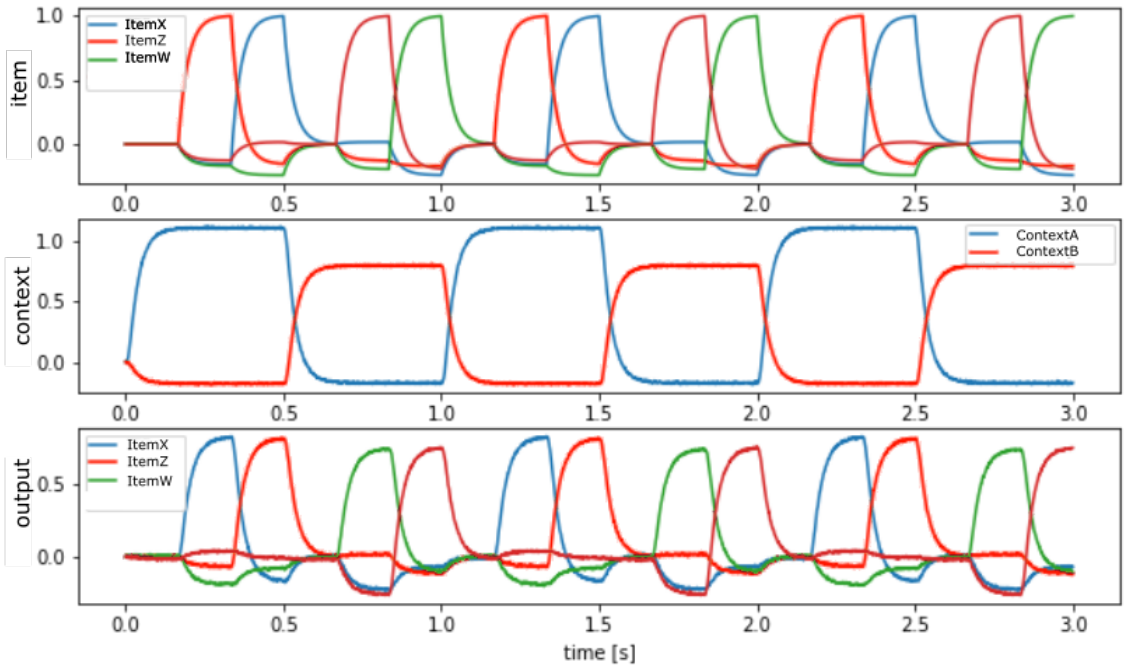


Figure 3.5: The model response to an item-item contextual cue when two items X and Z are related within context A, while two items Z and W are related within the context B. The results are perfectly in line with what one would have expected: the stimulus of Z within context A provides the expected results with high precision, that is X. Similarly the output obtained from the stimulus Z on context B, is W as expected.

was tested by supplying the stimulus of item Z in correlations with the two different contests, A and B, evaluating the response obtained from the model on the basis of the output provided by equations (3.3) and (3.4).

As can be seen from the results shown in the Figure 3.5, the results are perfectly in line with what one would have expected: the stimulus of Z within context A provides the expected results with high precision, that is X. Similarly the output obtained from the stimulus Z on context B, is W as expected.

These results show how the model is able to provide the correct item-item association when the stimulus is provided within the correct context. Interestingly the response to the same stimulus provided within a different context is instead less strong than the first. Furthermore, the model provides a correct answer even in the presence of several associ-

ations of the same item. As the context changes, the response of the model to the same stimulus changes.

With regard to the second task, the item-context response, our model showed equally satisfactory results.

Once again it is assumed that in context A the elements Z and X, as well as the elements W and Y, are strongly correlated with each other. It is also assumed that in a second context B the elements Z and W are equally correlated. In our example in the context of the scent Z and X are related, both being sweet, as are W and Y, both being sharp. Similarly in the context of shapes the elements Z and W are closely related, both being points.

During the testing phase we assumed that the relationships discussed above were already present in memory and we evaluated the response of the model by providing as input pairs of signals, both relating to items.

We have noticed, for example, that when Z is paired with X, the system outputs the signal relative to context A, when instead Z is paired with W the model reacts by returning context B. In a similar way , when W is paired with Y, the system outputs the signal relating to context A, when instead W is paired with Z the model reacts by returning context B.

The results obtained in this first experimental evaluation of our model relating to contextual behavior suggests to us how it is possible to use the NEF and the SPA to adequately simulate the brain responses based on the stimulus received at the input, taking into account the information relating to the context in which this evaluation is made.

In the next chapter we will tackle a specific problem of linguistic disambiguation, through the use of contextual information, using a neuro-computational model based once again on NEF and SPA.

4

Visual Resolution of Linguistic Ambiguities.

The results discussed in this chapter were presented in preliminary form in [**?**] and appeared in the Proceedings of the 12th International Conference on Neural Computation Theory and Application (NCTA) part of International Joint Conference on Computational Intelligence (IJCCI) [95]. The same results, in extended form, will be presented at the AISC [93]. The paper published at NCTA 2020 got the *Best Student Paper Award*

Understanding language goes hand in hand with the ability to integrate complex contextual information obtained via perception. In this chapter use context information to address a kind of ambiguity which can be considered as a canonical case of structural ambiguity, technically known as Prepositional Phrase Attachment, where a sentence includes a prepositional phrase that can be attached to more than one higher level phrases [50]. The attachment resolution is context dependent, we deal specifically with the case when depends on the visual context.

Specifically, provided with a sentence, admitting two or more candidate interpretations, and an image that depicts its content, it is required to choose the correct interpretation of the sentence depending on the image's content. Thus we address the problem

of selecting the interpretation of an ambiguous sentence matching the content of a given image.

In [12] the authors presented a novel task for grounded language understanding: disambiguating a sentence given a visual scene which depicts one of the possible interpretations of that sentence. To this end, they introduced a new multimodal corpus containing ambiguous sentences, representing a wide range of syntactic, semantic and discourse ambiguities, coupled with videos that visualize the different interpretations for each sentence. They addressed this task by extending a vision model which determines if a sentence is depicted by a video and demonstrated how such a model can be adjusted to recognize different interpretations of the same underlying sentence, allowing to disambiguate sentences in a unified fashion across the different ambiguity types.

To enable the systematic study of visually grounded processing of ambiguous language, It has been createed a new corpus, named LAVA (Language and Vision Ambiguities). This corpus contains sentences with linguistic ambiguities that can only be resolved using external information. The sentences are paired with short videos that visualize different interpretations of each sentence. The sentences encompass a wide range of syntactic, semantic and discourse ambiguities, including ambiguous prepositional and verb phrase attachments, conjunctions, logical forms, anaphora and ellipsis.

Overall, the corpus contains 237 sentences, with 2 to 3 interpretations per sentence, and an average of 3.37 videos that depict visual variations of each sentence interpretation, corresponding to a total of 1679 videos.

Using this corpus, the authors of [12] addressed the problem of selecting the interpretation of an ambiguous sentence that matches the content of a given video. The sentence tracker produces a score which determines if a sentence is depicted by a video. This earlier work had no concept of ambiguities; it assumed that every sentence had a single interpretation. They extend this approach to represent multiple interpretations of a sentence, enabling us to pick the interpretation that is most compatible with the video. To summarize, the contributions of their work are threefold. First, they introduceed a new task for visually grounded language understanding, in which an ambiguous sentence has to be disambiguated using a visual depiction of the sentence?s content. Second, they released a multimodal corpus of sentences coupled with videos which covers a wide range

of linguistic ambiguities, and enables a systematic study of linguistic ambiguities in visual contexts. Finally, they presented a computational model which disambiguates the sentences in our corpus with an accuracy of 75.36%.

4.1 Definition of the Task

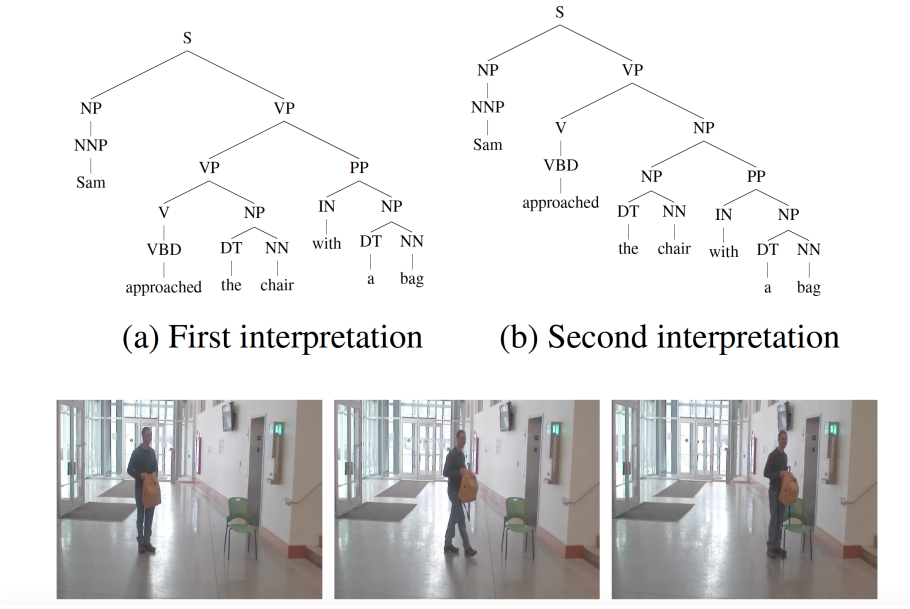
Following the same line as in [12] we provide a framework for the study of language understanding with visual context by introducing the task of grounded language disambiguation. This task requires to choose the correct linguistic representation of a sentence given a visual context depicted in a video. Specifically, provided with a sentence, n candidate interpretations of that sentence and a video that depicts the content of the sentence, one needs to choose the interpretation that corresponds to the content of the video.

To illustrate this task, consider the example in Figure 4.1, where we are given the sentence *Sam approached the chair with a bag* along with two different linguistic interpretations. In the first interpretation, which corresponds to parse 1(a), Sam has the bag. In the second interpretation associated with parse 1(b), the bag is on the chair rather than with Sam. Given the visual context from figure 1(c), the task is to choose which interpretation is most appropriate for the sentence.

Such task introduces a wide range of well known syntactic, semantic and discourse ambiguity classes. While the ambiguities are associated with various types, different sentence interpretations always represent distinct sentence meanings, and are hence encoded semantically using first order logic. For syntactic and discourse ambiguities we also provide an additional, ambiguity type specific encoding as described below.

- *Syntax:* Syntactic ambiguities include Prepositional Phrase (PP) attachments, Verb Phrase (VP) attachments, and ambiguities in the interpretation of conjunctions. In addition to logical forms, sentences with syntactic ambiguities are also accompanied with Context Free Grammar (CFG) parses of the candidate interpretations, generated from a deterministic CFG parser.

- *Semantics:* The corpus addresses several classes of semantic quantification ambiguities, in which a syntactically unambiguous sentence may correspond to different



(a) First interpretation (b) Second interpretation

Figure 4.1: An example of the visually grounded language disambiguation task. Given the sentence *Sam approached the chair with a bag*, two potential parses, (a) and (b), correspond to two different semantic interpretations. In the first interpretation Sam has the bag, while in the second reading the bag is on the chair. The task is to select the correct interpretation given the visual context (c).

logical forms. For each such sentence we provide the respective logical forms.

- *Discourse* The corpus contains two types of discourse ambiguities, Pronoun Anaphora and Ellipsis, offering examples comprising two sentences. In anaphora ambiguity cases, an ambiguous pronoun in the second sentence is given its candidate antecedents in the first sentence, as well as a corresponding logical form for the meaning of the second sentence. In ellipsis cases, a part of the second sentence, which can constitute either the subject and the verb, or the verb and the object, is omitted. We provide both interpretations of the omission in the form of a single unambiguous sentence, and its logical form, which combines the meanings of the first and the second sentences.

Figure 4.2 lists examples of the different ambiguity classes, along with the candidate interpretations of each example.

	Ambiguity	Example	Linguistic interpretations	Visual setups
Syntax	PP	Claire left the green chair with a yellow bag.	Claire [left the green chair] [with a yellow bag].	The bag is with Claire.
			Claire left [the green chair with a yellow bag].	Bag is on the chair.
	VP	Claire looked at Bill picking up a chair.	Claire looked at [Bill [picking up a chair]].	Bill picks up the chair.
			Claire [looked at Bill] [picking up a chair].	Claire picks up the chair.
	Conjunction	Claire held a green bag and chair.	Claire held a [green [bag and chair]].	The chair is green.
			Claire held a [[green bag] and [chair]].	The chair is not green.
		Claire held the chair or the bag and the telescope.	Claire held [[the chair] or [the bag and the telescope]].	Claire holds the chair.
			Claire held [[the chair or the bag] and [the telescope]].	Claire holds the chair and the telescope.
Semantics	Logical Form	Claire and Bill moved a chair.	**chair**(x), **move**(Claire, x), **move**(Bill, x)	Claire and Bill move the same chair.
			chair(x), **chair**(y), **move**(Claire, x), **move**(Bill, y), $x \neq y$	Claire and Bill move different chairs.
		Someone moved the two chairs.	**chair**(x), **chair**(y), $x \neq y$, **person**(u), **move**(u, x), **move**(u, y)	One person moves both chairs.
			chair(x), **chair**(y), $x \neq y$, **person**(u), **person**(v), $u \neq v$, **move**(u, x), **move**(v, y)	Each chair moved by a different person.
Discourse	Anaphora	Claire held the bag and the chair. It is yellow.	It = bag	The bag is yellow.
			It = chair	The chair is yellow.
	Ellipsis	Claire looked at Bill. Also Sam.	Claire looked at Bill and Sam.	Claire looks at Bill and Sam.
			Claire and Sam looked at Bill.	Claire and Sam look at Bill.

Figure 4.2: An overview of the different ambiguity types, along with examples of ambiguous sentences with their linguistic and visual interpretations. Note that similarly to semantic ambiguities, syntactic and discourse ambiguities are also provided with first order logic formulas for the resulting sentence interpretations. Table 4 shows additional examples for each ambiguity type, with frames from sample videos corresponding to the different interpretations of each sentence.

4.2 The ambiguity testbed

The testbed of our study is the LAVA (Language and Vision Ambiguities) corpus, recently introduced by [12]. Such corpus contains sentences with linguistic ambiguities that can only be resolved using external information. The sentences are paired with short videos that visualize different interpretations of each sentence. Such sentences encompass a wide range of syntactic, semantic and discourse ambiguities, including ambiguous prepositional and verb phrase attachments, conjunctions, logical forms, anaphora and ellipsis. Overall, the corpus contains 237 sentences, with 2 to 3 interpretations per sentence, and an average of 3.37 videos that depict visual variations of each sentence interpretation, corresponding to a total of 1679 videos. Each sentence involves two or more entities in one among four categories (person, bag, telescope and chair).

In their paper [12] also addressed the problem of selecting the interpretation of an ambiguous sentence that matches the content of a given video. In our case also, the road to

Figure 4.3: Examples of the six ambiguity classes described in table 2. The example sentences have at least two interpretations, which are depicted by different videos. Three frames from each such video are shown on the left and on the right below each sentence

solve the ambiguities is in pairing sentences with images that visualize the corresponding scene. Specifically, we focused on sentences where exactly three distinct entities belong-

ing to three distinct categories are involved (among person, bag, telescope and chair). For instance, given the sentence ”Sam approached the chair with a bag”, three different categories involved: person (Sam), chair and bag. In addition two different linguistic interpretations are plausible: the first interpretation assumes that Sam has the bag while approaching the chair, while the second one assumes that the bag is on the chair while Sam is approaching.

In addition, in our work, the ambiguous phrases examined by the system are related to the preposition “with”: the system is able to solve the ambiguity thanks to the given image and therefore to understand who the proposition refers to, for example: “Dany approached the chair with a yellow bag”, the system in this sentence is able to recognize to whom it refers “with”. Thus, our corpus contains 81 sentences, with 2 to 3 interpretations per sentence.

4.3 Background

Mammals rely a great deal on their visual system and as a result, one of the primary sources of information on the external world arrives in the form of visual input. It is not by chance that the initial vocabulary acquired by very young children is made up to a large extent by nouns referring to visible objects [8, 11]. Therefore, independently from the issue of context dependence, there is a significant body of research on the mapping between the visual world and human language spanning from developmental psychology [64], connectionist models [110] to biologically plausible neurocomputational models [101, 102]. While these studies manage important challenges in the cognitive processing of language and vision, they do not address explicitly the issue of linguistic ambiguities, which is instead central in the work of [127]. A substantial fraction of the activity of this group focused on the processing of ambiguous language providing evidence for the importance of visual information for linguistic ambiguity resolution by humans [125], results confirmed by other recent studies [17].

It is worth considering also different perspectives on the interaction between language and vision, aiming at application, in particular web application, where efficient multimodal representations are much needed [65, 124]. Currently, one of the most investigated

application is the automatic labeling of images and video clips [54]. The issue of ambiguities is clearly a great challenge for applications too, and has been addressed for several specific aspects. [6] tackle lexical ambiguity, proposing a model that disambiguate words with more than one sense, by detecting and recognizing the related object in the visual scene. There are few models that use vision for resolving kind of ambiguities different from the lexical one. One of the most investigated discourse ambiguities is due to coreference, the phenomenon of linguistic expressions that refer to the same entity in a discourse, as in the case of pronominal anaphora. A popular computational approach for coreference resolution based on text only is the Stanford CoreNLP system [66]. There are studies that attempt to solve coreferential ambiguities by using the visual context [107, 60], outperforming the Stanford model. Prepositional phrase ambiguity is addressed specifically by [20], proposing an approach to simultaneously perform semantic segmentation and prepositional phrase attachment resolution for captioned images. They showed that their semantic segmentation and prepositional phrase attachment resolution modules have complementary strengths, and that joint reasoning produces more accurate results than any module operating in isolation. Their vision and language approach significantly outperforms the text-based Stanford Parser.

[123] addressed the problem of selecting the interpretation of an ambiguous sentence that matches the content of a given video, by introducing a sentence tracker which produces a score with the aim of determining if a sentence is depicted by a video. This earlier result had no concept of ambiguities and it assumed that every sentence had a single interpretation [12]. extended the approach of [123] to represent multiple interpretations of a sentence, enabling to pick the interpretation that is most compatible with the video. To enable the systematic study of visually grounded processing of ambiguous language, they introduced the corpus LAVA adopted in our work, already described above. In addition they presented a computational model which disambiguates the sentences in the LAVA corpus with an accuracy of 75.36%.

All the approaches to visual syntactic disambiguation listed above differ in aims from our model, cognitive understanding is not in their agenda. On the contrary, our purpose is to explore processes of syntactic disambiguation compatible with human cognition, in the same vein of earlier models of vision and language interaction mentioned at the be-

ginning of this section. These models did not addressed the challenging issue of syntactic ambiguity, as we do now.

4.4 Our Model

Without restricting the general case we can assume that the world is populated by objects that are grouped into categories $\mathscr{C} = \{C_1, C_2, \cdots\}$. The small world of LAVA is populated with a limited number of categories, whose instances can appear in images, corresponding to scenes viewed by the agent. Images are matrices $\mathbf{I}$ of pixels in two dimensions. Inside an image $\mathbf{I}$ ti is possible to cut submatrices $\mathbf{B}_{x,y}^{(C)}$ with centers at coordinates x, y, and size suitable to contain objects of category C. The submatrices $\mathbf{B}_{x,y}^{(C)}$ bear a resemblance with foveal images captured during saccadic movements gazing over different portion of the visual scene, with the purpose of recognizing each single object. Unlike natural saccades, in our model the centers of the detailed views are not driven by top-down mechanisms, are instead sampled at fixed regular intervals, scanning the whole image:

$$X = \langle x_1, x_1 + \Delta_x, \cdots, x_1 + N\Delta_x \rangle, \tag{4.1}$$

$$Y = \langle y_1, y_1 + \Delta_y, \cdots, y_1 + M\Delta_y \rangle. \tag{4.2}$$

The visual component of the model is made by a set of deep convolutional neural networks, tuned to recognize one of the possible categories $C \in \mathscr{C}$. Each network is a function $f_{(C)}(\cdot)$ estimating the probability of an object of category C to be inside a submatrix $\mathbf{B}_{x,y}$:

$$f_{(C)}() : \mathbb{R} \times \mathbb{R} \to [0..1] \tag{4.3}$$

By applying the deep convolutional neural network $f_C(\cdot)$ to all the submatrices of an image $\mathbf{I}$, a vector $\vec{p}^{(C)}$ of probabilities to find an object of category C in the discrete horizontal positions X is constructed. An element $p_i^{(C)}$ of $\vec{p}^{(C)}$ is computed as following:

$$p_i^{(C)} = \max_{y \in Y} \left\{ f_{(C)} \left(\mathbf{B}_{x_i,y}^{(C)} \right) \right\} \tag{4.4}$$

The rationale behind equation (4.4) is that in an interior environment the displacement of objects – therefore their spatial relationship – appears mainly in the horizontal dimension of the retinal projection of the scene. It is therefore possibly to capture the probabilistic

locations of objects as vectorial representations corresponding to scanning the scene horizontally along X. We can now compose equation (4.4) into a family of functions $\phi^{(C)}(\cdot)$ that, given and image $\mathbf{I}$, return probability vectors $\vec{p}^{(C)}$:

$$\phi^{(C)}() : \mathbb{R} \times \mathbb{R} \rightarrow [0..1]^N \tag{4.5}$$

Let us move on the linguistic part of LAVA and of the stimuli to the model. The full set of sentences in LAVA use words from a closed lexicon $\mathscr{L}$, and within this lexicon there are two subsets relevant for our model. One is the lexicon of words $\mathscr{W} \subset \mathscr{L} = \{W_1, W_2, \cdots\}$ used to name objects of the categories in $\mathscr{C}$. In the case of LAVA we can assume a deterministic reference function:

$$c(W) : \mathscr{W} \rightarrow \mathscr{C} \tag{4.6}$$

associating every word W to a category C. There is then a smaller lexicon of prepositions, the grammatical category responsible for the contextual ambiguities: $\mathscr{P} \subset \mathscr{L} = \{\texttt{with}, \cdots\}$.

A sentence in the LAVA is an ordered set $\mathscr{S}$, with elements $S_i \in \mathscr{L}$, from which a simple preprocessing extracts three key words:

$$\mathscr{S} \rightarrow \begin{cases} W_{\mathrm{P}} & \text{noun under the head of the preposition} \\ W' & \text{first noun possible head of the prepositional phrase} \\ W'' & \text{second noun possible head of the prepositional phrase} \end{cases} \tag{4.7}$$

The noun W_{P} is easily found by searching in $\mathscr{S}$ the first element $S_i \in \mathscr{P}$, and then searching the first element S_j with $j > i$ such that $S_j \in \mathscr{W}$. The other two nouns W' and W'' are the only two possible elements $S_{l,k} \in \mathscr{W}$ with $l \neq j, k \neq j$. Let us call $W_{\mathrm{H}} \in \{W', W''\}$ the correct head of the prepositional phrase.

The three key words W_{P}, W', W'' find a correspondence in the model in terms of three Nengo SPA items: $\vec{V}_{\mathrm{P}}$, $\vec{V}'$ and $\vec{V}''$. The processed sentence $\mathscr{S}$ is linked with an image $\mathbf{I}$ in which the objects of categories referred by W_{P}, W' and W'' are searched:

$$\vec{p}_{\mathrm{P}} = \phi^{(c(W_{\mathrm{P}}))}(\mathbf{I}), \tag{4.8}$$

$$\vec{p}' = \phi^{(c(W'))}(\mathbf{I}), \tag{4.9}$$

$$\vec{p}'' = \phi^{(c(W''))}(\mathbf{I}). \tag{4.10}$$

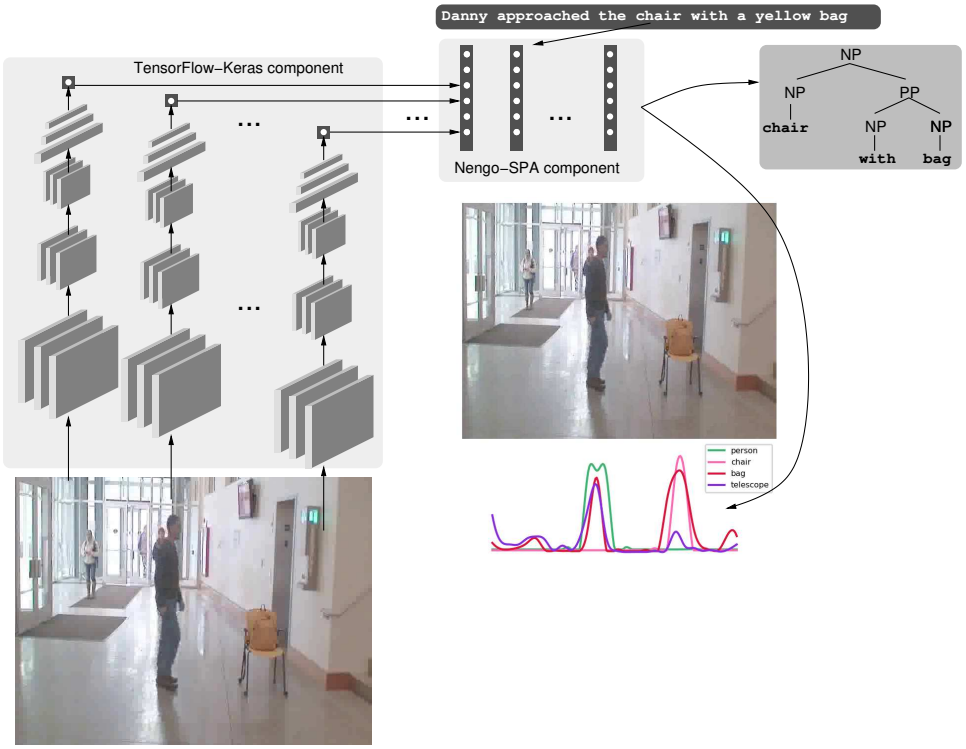


Figure 4.4: Sketch of the neural model.

These vectors, expressing probabilities of locations of the three categories along the horizontal view of the agent, are bind to the corresponding Nengo SPA items, using NEF $\circledast$ operator, introduced in equation (1.2). We can express the binding in our case as a function $b(\cdot) : \mathbb{R}^N \to \mathbb{R}^N$:

$$b\left(\vec{V}_{\mathrm{P}}\right) = \vec{V}_{\mathrm{P}} \circledast \vec{p}_{\mathrm{P}}, \tag{4.11}$$

$$b\left(\vec{V}'\right) = \vec{V}' \circledast \vec{p}', \tag{4.12}$$

$$b\left(\vec{V}''\right) = \vec{V}' \circledast \vec{p}'', \tag{4.13}$$

and the disambiguate item $\vec{V}^*$ is selected as following:

$$\vec{V}^* = \arg \min_{\vec{V} \in \{\vec{V}', \vec{V}''\}} \left\{ \zeta \left(b\left(\vec{V}_{\mathrm{P}}\right), b\left(\vec{V}\right) \right) \right\} \tag{4.14}$$

where $\zeta\left(\vec{V}_1, \vec{V}_2\right)$ is a measure of similarity between the two Nengo SPA items $\vec{V}_1$ and $\vec{V}_2$.

Therefore, the predicted head of the prepositional phrase W_{H}^{*} is the lexical item associated with $\vec{V}^{*}$.

The combined deep convolutional and Nengo SPA neural processes are sketched in Fig. 4.4.

4.5 Experimental results

We evaluated the performance of our neural model, described in the previous section, on the LAVA dataset. Each sentence and its associated picture in the dataset was processed, predicting the lexical item that most likely is the head of the ambiguous propositional phrase. The code we used for constructing the neural model and computing our experimental results is available for download at `http:www.github.com/alex-me/nencog`.

Here we will present first a set of qualitative results useful for illustrating the neural processes performed by the model, and then the quantitative results of the disambiguation task.

Fig. 4.5 illustrates few qualitative results of the intermediate stage of the process, when the vectors $\vec{p}^{(C)}$ of probabilities to find an object of category C in the discrete horizontal positions X are computed. Vectors are generated by applying the deep convolutional neural network $f_C(\cdot)$ to all the submatrices of a given image according to equation (4.4). Fig. 4.5 include the following 6 examples:

ID code in the LAVA corpus	sentence
`00022-9570-9660`	: Danny approached the chair with a yellow bag
`00022-18590-18700`	: Danny left the chair with a yellow bag
`00022-22420-22510`	: Danny left the chair with a green bag
`00022-54050-54160`	: Danny approached the chair with a blue telescope
`00022-55780-55850`	: Danny approached the chair with a blue telescope
`00029-24110-24210`	: Danny looked at the chair with a blue telescope

The vectors of the kind shown in Fig. 4.5 that are relevant in the sentence, are then associated with the three Nengo SPA units $\vec{V}_{\mathrm{P}}$, $\vec{V}'$, $\vec{V}''$ by means of equations (4.11), (4.12), (4.13). All SPA units are populations of spiking neurons, which vectors evolve in time following equation (1.1). This evolution is shown in Fig. 4.6 for a small number of examples. All plots show the evolution in time of the three vectors related with Nengo

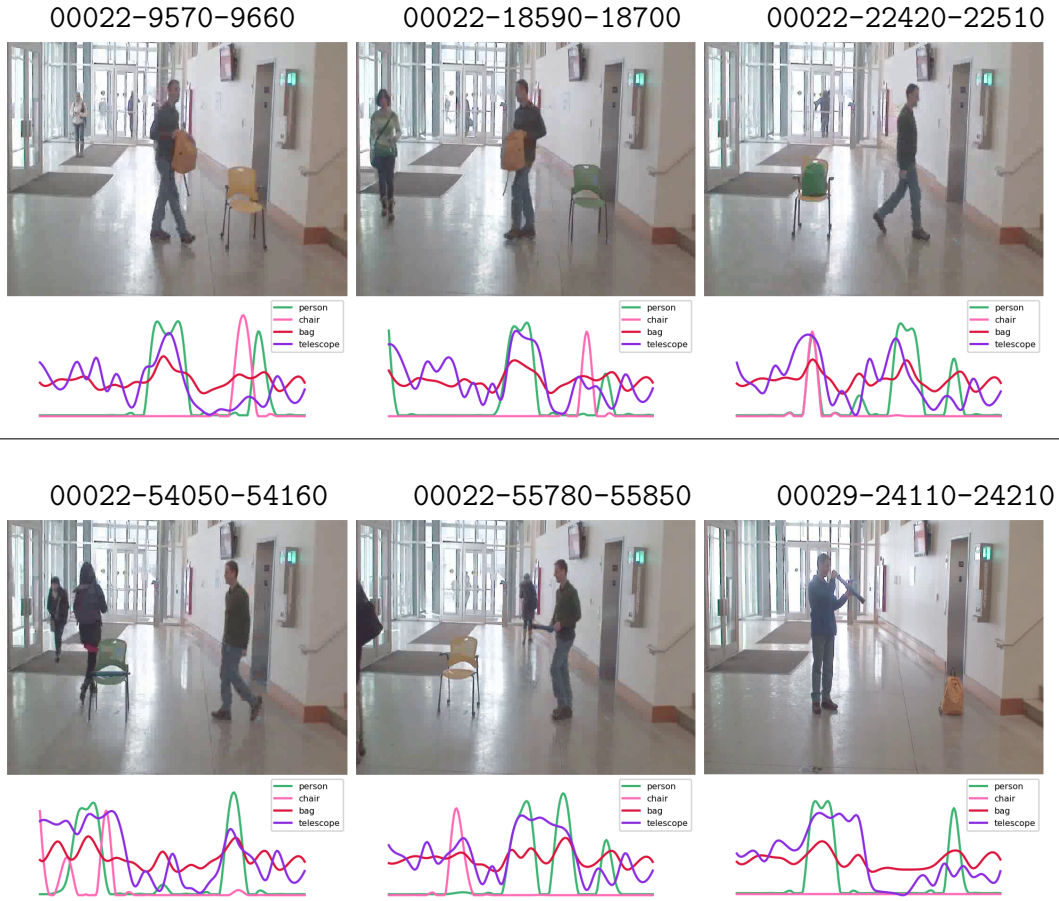


Figure 4.5: The vectors $\vec{p}^{(C)}$ of probabilities to find an object of category $C \in \{$ `person`, `bag`, `telescope`, `chair` $\}$ in the discrete horizontal positions X computed for 6 different images of the LAVA corpus. Vectors are generated by applying the deep convolutional neural network $f_C(\cdot)$ to all the submatrices of a given image according to equation (4.4). Images are labeled with their corresponding codes in the LAVA corpus.

W_P =bag		
$W_H \setminus W_H$	person	chair
person	.73	.27
chair	.00	1.00
accuracy	0.87	

W_P =telescope			
$W_H \setminus W_H$	bag	person	chair
bag	.25	.75	.00
person	.00	.96	.04
chair	.07	.20	.73
accuracy	0.77		
overall accuracy	0.81		

Table 4.1: Overall and detailed accuracy obtained by the model when tested on the LAVA dataset. The results are grouped for the two possible W_P, and for each one the matrix of errors is shown, with the true W_H as rows, and the predicted W_H^* as columns. The cumulative accuracy for the two sets is shown, as well as the overall accuracy.

SPA units $\vec{V}_P$, $\vec{V}'$, $\vec{V}''$. The crucial aspect for the purpose of the disambiguation is that the final shape of the vectors is such that between $\vec{V}'$ and $\vec{V}''$ the most similar to $\vec{V}_P$ will be the SPA unit associated with the correct word W_H. This final similarity can be appreciated in the four examples of Fig. 4.6.

Table 4.1 presents the quantitative results of the model over all the processed LAVA sentences. The total set of sentences has been divided into two categories, those with W_P=bag and those with W_P=telescope. In the first set the possible correct heads of the prepositional phrase W_H can be person or chair, while in the second set the possibilities are bag, person, and chair. For each of the sets the matrix of errors is reported, showing the fractions of lexical element that the model has predicted as head of the prepositional phrase, given the correct head word. It can be seen that the overall accuracy of the model is good, over 80%, and that it is slightly lower when W_P=telescope.

As in the case of the experiments by [12], the most significant source of failures are poor object detection. Objects in the LAVA corpus are often rotated and presented at angles which turns out to be difficult to recognize. It turns out moreover that some object classes, like the telescope and the bag, are much more difficult to be recognized. It can be

observed in Fig. 4.5 that objects of the classes `bag` and `telescope` are the most difficult to be recognized due to their small size and to the fact that hands tend, in most cases, to largely cover them. Conversely objects of the classes `person` and `chair` are generally well detected and generate a much more accurate probability vector. We have assessed this source of error by evaluating the pure visual recognition accuracy, which is of 80% for person objects, of 79% for chair object, of 67% for the bag object, and as low as 60% for telescope. Note that we deliberately avoided to include in the model a state-of-the-art deep learning model that would have easily achieved better recognition rates, but loosing biological plausibility. Moreover, with our model we have been able to evaluate the disambiguation performances that takes into account uncertainty in the visual process. As seen in the performances shown in Table 4.1, disambiguation is more reliable that the pure visual object recognition.

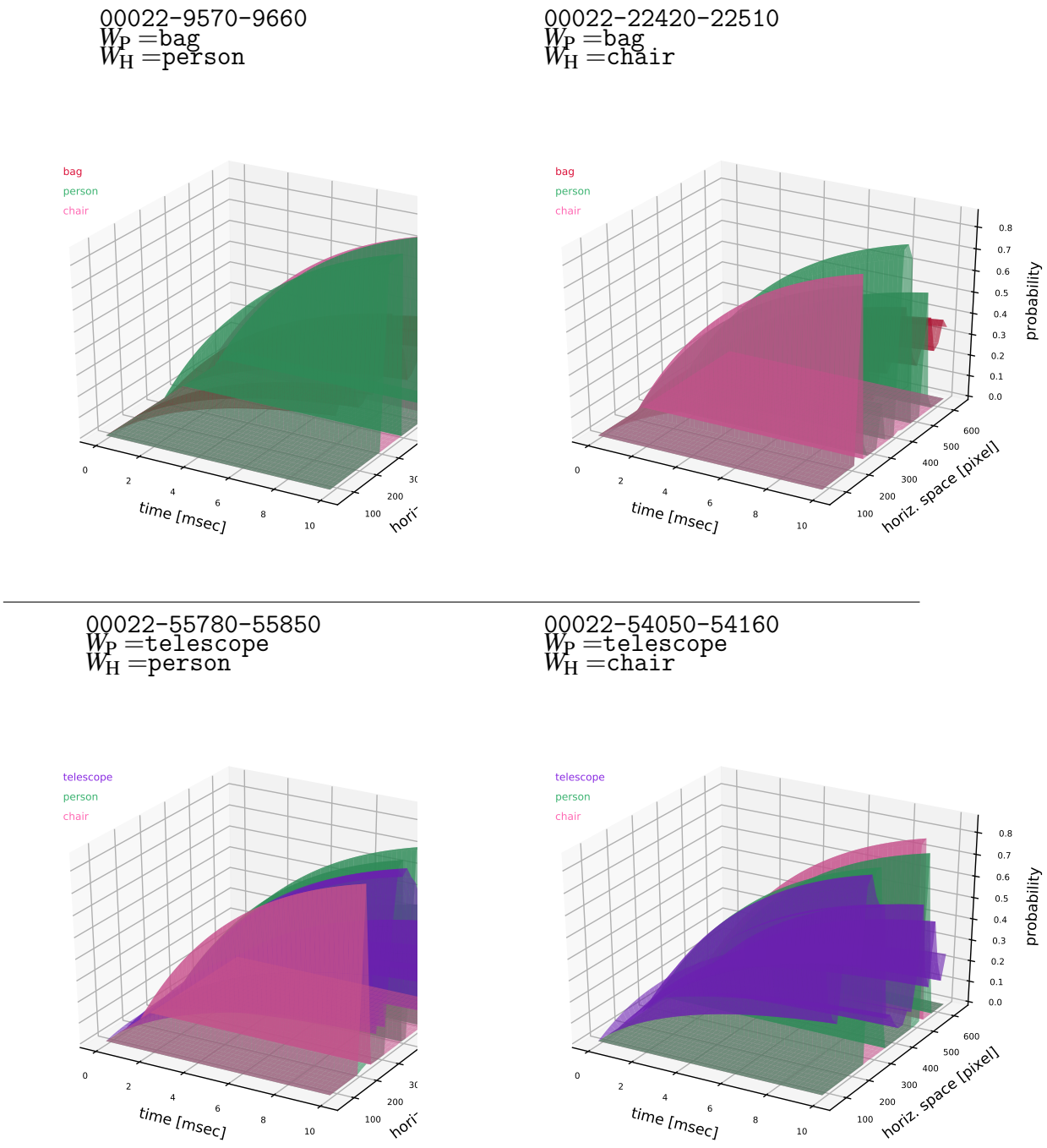


Figure 4.6: Evolution in time of the Nengo SPA neural populations associated with the three key words in the disambiguation tasks. In the example in the top left the noun under the head of the preposition is `bag`, and the head of the prepositional phrase is `person`, it is visible how at the end of the evolution the SPA vector associated with `bag` become more similar to the vector associated with `person`, with respect to the vector associated with `chair`. Exactly the opposite happen in the example in the top right, where this time the noun under the head of the preposition, `bag` has `chair` as head of the prepositional phrase. Similarly in the bottom row for `telescope` as the noun under the head of the preposition. The scenes corresponding to the four examples can be seen in Fig. 4.5

.

5

A Context Sensitive Account of Similarity Based on the SPA

The results discussed in this chapter has been accepted for presentation at the AISC Midterm Conference 2020 *Cognitive Science meets the Humanities* [94], organized by the *Italian Association of Cognitive Science*.

5.1 Why Similarity?

In addressing the analysis of contextual effects with a neuro-computational approach, a not insignificant issue concerns the optimal way to encode concepts in clusters of activation vectors. A theoretical criterion to ascertain that the chosen coding is effective, and cognitively plausible, is to verify how the coding behaves in front of one of the main relationships that exists between concepts: semantic similarity.

On the other hand the context plays a fundamental role in the perception of the similarity between two different concepts: depending on the context in which they are compared, two entities can appear more or less similar to each other.

For the reasons reported above a specific study on this aspect was therefore included in this Chapter, proposing a context sensitive similarity model based on SPA.

Specifically in this chapter, we propose an alternative, unifying solution to the current challenges in concept research. Using methods for characterizing representational states in neural systems [27, 28], we describe concepts in terms of processes involving mental representations as *semantic pointers* (see Chapter 2). just to remind the reader of the fundamental concepts, semantic pointers are neurally instantiated, symbol-like representations that can be transformed in numerous ways to yield further representations that function to support cognitive processes like categorization, inference, and language use.

5.1.1 Background

How does our brain compute semantic similarity between two concepts? Such kind of computation involves cognitive processes on which much work has been made, giving birth to conflicting and often criticised theories.

The world is an orderly enough place where similar objects and events tend to behave similarly. Take frogs and toads as an example. They are similar in many respects: in their form, internal biology, behaviour and diet just to list a few. This is way one can often successfully generalise from what one knows about frogs to what it could be said about toads.

The fact that human assessments of similarity are fundamental to cognition is a widely shared opinion. Similarity, is fundamental for learning, knowledge and thought, for only our sense of similarity allows us to order things into kinds so that these can function as stimulus meanings. Reasonable expectation depends on the similarity of circumstances and on our tendency to expect that similar causes will have similar effects [105].

Thus assessments of similarity are important since they provide methods for predicting as many important aspects of our world as possible [45].

For example to assess the similarity between frogs and toads can help in generalise to toads some information learned about frogs. More formally if the similarity between two concepts x and y increases, the probability of correctly inferring that y has p upon knowing that x has p increases accordingly [128]. This relationship is also used contextually. For instance leopards and jaguars are similar in anatomy but quite dissimilar in behaviour. This difference successfully predicts that people are likely to make anatomical but not

behavioural inferences from leopards to giaguars.

What is particularly relevant for our work is that when one has complete knowledge about the reasons why x has a property p, then general similarity between x and y is no longer relevant to generalisations [37, 106]. The knowledge itself completely guides whether to infer if y as property p is appropriate. We tend to rely on similarity to generate inferences from x to y and categorise objects into kinds when we do not know exactly what properties are relevant or when we cannot easily separate an object into separate properties.

Similarity is an excellent example of a domain-general source of information. Even when we do not have specific knowledge of a domain, we can use similarity as a default method to reason about it. Another argument for the importance of similarity in cognition is simply that it plays a significant role in psychological accounts of problem solving, memory, prediction, and categorisation. If a problem is similar to a previously solved problem, then the solution to the old problem may be applied to the new problem [46, 115, 116]. In addition similarity is able to provide an elegant tool for examining the structure of our mental entities and the processes that operate on them.

Not surprisingly, similarity has also played a fundamentally important role in psychological and cognitive experiments and theories. For example, in many experiments people are asked to make direct or indirect judgments about the similarity of pairs of objects. A variety of experimental techniques are used in these studies, but the most common are to ask subjects whether the objects are the same or different, or to ask them to produce a number, between say 0 and 10, that matches their feelings about how similar the objects appear

5.1.2 Our Proposal

We argue that such recent theory of biological cognition [27] is able to define a more plausible model of semantic similarity, according with the main guidelines of the other traditional models known in literature, on the one hand, and giving a solution to traditional criticisms against these models [134, 63], on the other.

The model of similarity based on the SPA we propose in this chapter is a geometric

model in that it takes similarity to be the distance between semantic pointers in some n-dimensional mental comparison space, thus computing measurements like the traditional mental distance model [121]: the shorter the distance between two semantic pointers, the more similar the two concepts represented by the semantic pointers are. The difference is that in the traditional model computations are made in the same comparison space, while the SPA model assumes the coexistence of several comparison spaces. Thus computations my change their measurements depending on the context, addressing many of the criticisms raised against the distance model [134, 63], mostly concerned with symmetry, minimality, and the triangle inequality. As a consequence many mathematical techniques for deriving spaces from data can be used, as multidimensional scaling [121] and latent semantic analysis. The SPA provides for a computation that can be more or less accurate on the basis of the semantic level in which the computation is made: shallow measurements work on small low-level semantic spaces and are therefore faster, while deeper measurements work on larger high-level semantic spaces and are therefore slower. This highlights how much the SPA model has in common with the featural [134] and the structural [31] models, where the components of a semantic pointer can be assimilated to the features of a concept, and where similar concepts to a first shallow analysis may turn out to be more distant after a detailed comparison. We also argue that, when concepts belonging to different contexts are compared, the brain must do the further work of converting corresponding pointers laying on different semantic spaces. This may increases the semantic distance of the concepts and highlights how much this model has in common with the transformational model [40] where the work done for transformations corresponds to the effort necessary to change context.

In Section 5.2 we present the four main cognitive models for similarity proposed in the literature, and specifically the geometric, the feature based, the alignment based, and the transformational models. In Section 5.3 we give a description of our new account of similarity proposed in this chapter and discuss, in Section 5.4, about some criticisms rised along the years against the standard models and about their settlement in the context of our new account of similarity based on the semantic pointers model.

5.2 Models of Similarity

Despite the strong presence of similarity judgments in our reasoning, an accurate model of similarity has yet to be agreed upon. However a number of theoretical accounts of similarity have been proposed in the last decades (see [48] and [39] for a detailed survey on similarity models) bringing a significant influence on important research areas such as statistics, automatic pattern recognition, data mining, and marketing. In this section we give a brief description, or at least a hint, of the four main cognitive models for similarity proposed in the literature. We will refer to them as the geometric, the feature based, the alignment based, and the transformational models.

A model of similarity $\mathscr{M}$ should describe how the elements of a universal set of entities $\mathscr{E}$ are represented by our cognitive system. Based on this representation, given two elements $x, y \in \mathscr{E}$, the model provides a way to compute similarity between x and y (which may be different from the similarity between y and x if $\mathscr{M}$ admits non-symmetric similarities).

Formally the model defines a *similarity function*, $\delta : \mathscr{E} \times \mathscr{E} \to \mathbb{R}$, which associates to each ordered pair, (x,y), of elements in $\mathscr{E}$ a similarity value $\delta(x,y)$. In some cases it is possible to have a *dissimilarity function* $\bar{\delta}(x,y)$ which computes the dissimilarity between x and y, instead of their similarity. The dissimilarity value is always inversely proportional to the corresponding similarity value.

Similarity models have played a fundamentally important role in psychological experiments and theories. A variety of experimental techniques have been used in these studies, but the most common are to ask subjects whether two given objects are the same or different, or to ask them to produce a number, say between 1 and 10, that matches their feelings about how similar (or dissimilar) the objects appear.

5.2.1 Geometric Models

One of the most influential models for similarity if the *geometric model* (also called *mental distance model*) [15, 132, 133] where entities are represented as points in a mental n-dimensional space and the similarity between such entities is inversely proportional to the distance between the corresponding points (Shepard, 1962; Thurstone, 1927; Torger-

son, 1965).

Thus in a geometric model which assumes a mental space of dimension n, each entity $x \in \mathscr{E}$ is represented by a point with n coordinates, $x = \langle x_1, x_2, .., x_n \rangle$, and the mental distance function $\bar{\delta}$ is a metric in the mathematical sense of the term. This implies that the dissimilarity $\bar{\delta}(x,y)$, between two entities $x, y \in \mathscr{E}$, is associated with the distance of the corresponding points:

$$\bar{\delta}(x,y) = \left[\sum_{k=1}^{n} |x_k - y_k|^r \right]^{\frac{1}{r}}$$

where r is a parameter that allows different spatial metrics to be used. The most common spatial metric is the *Euclidean metric*, obtained with $r = 2$, and where the distance between two points is the length of the straight line connecting the points x and y.

Another common spatial metric is the *city-block metric*, obtained with $r = 1$, and where the distance between two points is the sum of their distances on each dimension. Such spatial metric is often appropriate for psychologically separated dimensions such as color and shape, brightness and size or any given set of separated measurable perceptions.

One of the most interesting properties (and source of criticism) of geometric models is that the distance function $\bar{\delta}$ must be in accord with the following three axioms whose meaning is well known:

(a)	$\bar{\delta}(x,y) \geq \bar{\delta}(x,x) = 0$	(distance minimality)
(b)	$\bar{\delta}(x,y) = \bar{\delta}(y,x)$	(distance symmetry)
(c)	$\bar{\delta}(x,y) + \bar{\delta}(y,z) \geq \bar{\delta}(x,z)$	(triangle inequality)

Concrete implementations of a geometric model is commonly obtained by multidimensional scaling (MDS) [121, 121] which is a technique that creates a space displaying the relative positions of a number of objects, given the table of the distances between them. In this context the input of an MDS program may be similarity (dissimilarity) judgments, confusion matrices, correlation coefficients, or any other measure of pairwise similarity. However it is well known that the distances given by the similarity ratings may not be satisfiable in any given dimensional space. It turns out indeed that a perfect reconstruction of the similarities among a set of n entities can be obtained only when a high enough dimensionality (in the worst case $n-1$ dimensions) is used. Thus one of the goals of the MDS analysis is to keep the number of dimensions as small as possible.

5.2.2 Feature Based Models

The most influential alternative to geometric models of similarity is the *feature based model* (also known as *feature contrast model* or simply *contrast model*), which has been introduced by Amos Tversky In 1977 [134]. The idea lying behind a feature based model of similarity is that subjective assessments of similarity not always satisfy the three axioms required by geometric models of similarity.

Specifically a feature based model represents an entity $x \in \mathscr{E}$, as a collection of features and the similarity between two entities is expressed as a linear combination of the measure of the common and distinctive features. Specifically if we denote by $[x \cap y]$ the number of features that x and y have in common and by $[x-y]$ the number of features that are in x and not in y, then the similarity $\delta(x,y)$, between two entities $x, y \in \mathscr{E}$, is computed as follows:

$$\delta(x,y) = \alpha[x \cap y] - \beta[x-y] - \gamma[x-y]$$

where α, β, and γ are weights for the common and distinctive components.

There are no restrictions on what may constitute a feature. A feature may be any property, characteristic, or aspect of a stimulus. In addition features may be concrete or abstract.

One advantage of the feature based model is that it can account for violations in any of the metric distance axioms. For instance it is able to implement asymmetric similarity between entities since β may be different from γ so that $\beta[x-y]$ may be different from $\gamma[x-y]$.

For the sake of completeness we notice that a number of earlier models turn out to be similar to the contrast model introduced by Tversky, by applying a ratio function as opposed to a linear contrast of common and distinctive features. Just to cite few of them:

(a)	$\delta(x,y) = [x \cap y]/[x]$	Bush and Mosteller (1951)
(b)	$\delta(x,y) = [x \cap y]/([x]+[y])$	Eisler and Ekman (1959)
(c)	$\delta(x,y) = [x \cap y]/[y \cap x]$	Sjoberg (1972)

In all these models the fundamental premise, that entities can be described in terms of constituent features, has been a powerful idea in cognitive psychology and has influenced much work.

Neural network representations, for instance, are often based on features, where entities are broken down into binary vectors of features where ones identify the presence of features and zeros their absence. In such context the similarity distance value is typically computed using the fairly simple function, i.e. the *Hamming distance*, formally given by $\delta(x,y) = [x-y] + [y-x]$.

5.2.3 Alignment Based Models

In the *alignment based models* (also known as *structural models*) comparison of two entities x and y is computed not just by matching their features, but by determining how such features correspond to, or align with, one another.

One of the most interesting aspects of alignment based models is that they make purely relational similarity possible [31] since matching features influence similarity more if they belong to parts that are placed in correspondence, and parts tend to be placed in correspondence if they have many features in common [36, 74].

Another interesting aspect of such models is that concerns alignable and nonalignable differences [75]. Nonalignable differences between two entities are attributes of one entity that have no corresponding attribute in the other entity while alignable differences are those that require the elements of the entities first be placed in correspondence. Consistent with the role of structural alignment in similarity comparisons, alignable differences influence similarity more than nonalignable differences [76] and are more likely to be encoded in memory [77].

Such models deserve a special mention since they have proven to be useful in explaining consumer preferences of commercial products, suggesting, for example, why new products are viewed more favorably when they improve over existing products along alignable rather than unalignable differences [139].

5.2.4 Transformational Models

Extending the alignment based account of similarity, *transformational models* [140] assume that a cognitive system uses an elementary set of transformations when computing the similarity between two entities. In this context, if we assume that each entity is de-

scribed by a sequence of features, the similarity is assumed to decrease monotonically as the number of transformations required to make one sequence of features identical to the other increases.

Thus while the correspondences among features of two entities are explicitly stated in an alignment based model, such correspondences are implicit in a transformational based model. For this reason a potentially fertile research direction is to combine the alignment based account for representing the internal structure of the features of entities with the constraints that transformational accounts provide for establishing a psychologically plausible measure of similarity [44, 84].

One of the most important criticisms against these models is to specify or justify what transformational operations are possible or plausible.

5.3 An Account of Similarity based on the SPA

Eliasmith's theory of biological cognition introduces semantic pointers as neural representations that carry partial semantic content and are composable into the representational structures necessary to support complex cognition. Semantic pointers are symbol-like representations that result from the compression and recursive binding of perceptual, lexical, and motor representations, effectively integrating traditional connectionist and symbolic approaches.

In [] the authors proposed a computational model using semantic pointers that is able to replicate experimental data from categorization studies involving each prior paradigm. It is argued that a framework involving semantic pointers can provide a unified account of conceptual phenomena.

In this section we briefly introduce the reader to the semantic pointer theory and describe how they are used to represent concepts in Eliasmith's theory of biological cognition. Then we give our account of similarity based on semantic pointers.

5.3.1 An Account of Similarity based on the SPA Framework

As already discussed in Chapter 2 a *semantic pointer*, in its most basic form, can be thought of as a compressed representation of a specific concept associated with a specific

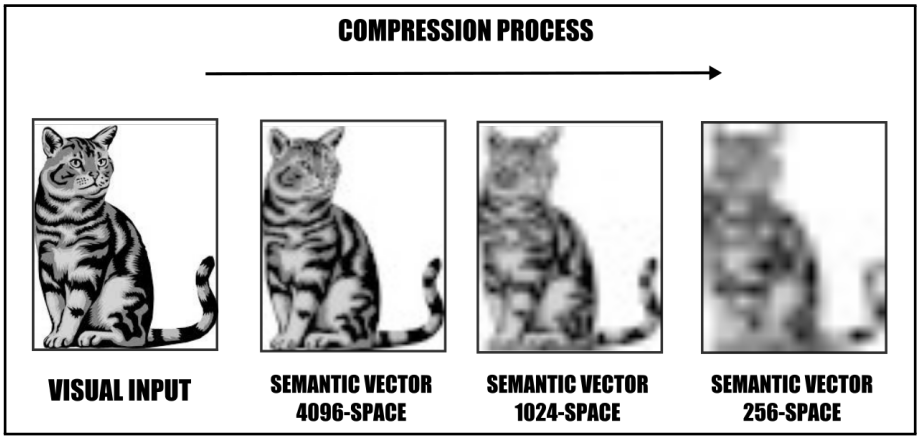


Figure 5.1: A hierarchical statistical model. An original visual input is compressed into a 256-dimensional representation through a series of hierarchical statistical models. Each level can be thought of as corresponding to a level of the visual hierarchy for object recognition as indicated. The dimensionality of the state space at each level is indicated below the level itself.

domain or within a specific context. It can act as a summary representation of a concept or category.

In this chapter we briefly review some of the main features of semantic pointers which are relevant for the discussion in this Chapter.

A semantic pointer can be represented as a vector in an n-dimensional space. In this context, the value n represents the dimension of the vector space in which the pointer is defined. This dimension is closely related to the size of the population of neurons that give rise to this compressed representation.

Typically, such representations are generated from perceptual inputs. A classic example is what happens with the vision of an object within one's visual field, although similar representations can be generated through hearing, touch or other perceptual mechanisms.

As highlighted in chapter 2, visual perception initially gives rise to the activity of a very large population of neurons on the first level of the visual cortex and will then be encoded by a high-dimensional semantic pointer. Through some transformations the neural populations in the underlying layers of the visual cortex make it possible to obtain increasingly compact representations of the input, providing, to all effects, semantic summaries of the original input. It is therefore possible that at the end of the perception process a

very compressed representation of the perceptual input is produced (see Fig. 5.1).

This transformation of the input representation is consistent both with the decrease in the number of neurons observed in the deeper hierarchical layers of the visual cortex, and with the development of hierarchical statistical models of neural inspiration for the reduction of dimensionality.

One of the main characteristics of semantic pointers lies in the possibility of being linked together in more structured representations containing lexical, perceptual, motor or other kind of information. And it is important to keep in mind that any given semantic pointer can be manipulated independently of the network used to generate it, and that the structured representations resulting from such a binding are themselves semantic pointers.

In the NEF the binding of two semantic pointers is performed using a process called circular convolution [27]. It is not our interest to go into technical details of this operation in this chapter (see Chapter 2 for more details) and for this reason we can limit ourselves to saying that the circular convolution can be thought of as a function that merges, by binding them together, two input vectors into a single output vector of the same dimension. Overall, the result of this binding operation is a single representation that captures the relationships between the input pointers involved in the binding.

The fact that the dimension of the resulting vector is the same as the dimension of those involved in the binding implies that part of the information contained in the starting vectors has been lost. The vector obtained from the ligature can therefore be seen as a compressed representation of the relationship that binds two or more concepts together.

The binding process can be repeated an indefinite number of times and, more interestingly, it can be reversed to obtain one of the semantic pointers that have given rise, through the binding, to the compressed representation of a more complex concept. However, it is important to know that this reconstruction can only be done in an approximate way.

We are now ready to introduce the *Semantic Pointer Model of Similarity* (SPMS), a similarity metric model that associates the degree of similarity of concepts to the distance between the corresponding semantic pointers in a system of mental spaces. This definition brings the new model of similarity very close to the traditional geometric model, however

it has also some points in common with other classic similarity models.

Specifically, as occurs in the traditional geometric model of similarity, the smaller the distance between two semantic pointers within a mental space, the more similar are the two concepts represented by semantic pointers. Consequently the farther the distance the less similar they are.

The key difference between the SPMS and the traditional geometric model lies in the fact that the latter assumes the existence of a single mental space while the SPMS provides for the presence of a system of comparison spaces, thus more than one mental space. Such many spaces for comparison operate on different contexts of judgment, grouping the semantic pointers not only on the basis of the their semantic category, but also on the basis of the level of detail or abstraction through which the concepts are represented, and therefore on the basis of the size of the semantic pointers representing concepts.

This implies that while in the traditional geometric model all judgments are expressed with reference to the same mental space of comparison, in the new SPMS the space of comparison changes according to the semantic pointers involved in the judgment and therefore according to the context in which the judgment takes place or according to the level of detail on the basis of which the judgment is made.

It can therefore happen that two objects can have different measures of similarity between them if they are compared in different mental spaces, or in different contexts. For each context, such as appearance, color, shape, taste, etc. the space of comparison changes.

It can also be observed that, if we limit the computation of similarity to a single space of comparison, as in the traditional geometric model, we should also assume that all judgments of similarity are equally complex, but our experience suggests us without any doubt that it is not. The SPMS reflects this idea by allowing two objects to have different measures of similarity based on the fact that they can be compared at different levels of detail.

This scenario suggests the presence of a system of mental spaces, or geometric spaces, of different dimensions, coexisting within the same model. Larger spaces would host semantic pointers capable of representing complex concepts and at a greater level of detail. Mental spaces of smaller dimensions would instead host smaller semantic pointers, with

a higher level of compression and which refer to a synthesis or a more abstract version of the corresponding concepts hosted on spaces of higher dimension.

This model also fits well with some of the principles underlying feature based models in accordance with which entities can be described in terms of constitutive features. A complex concept is in fact constituted by the binding (through the convolution process) of various elementary concepts, which actually represent their constitutive features. The larger the number of such constitutive features, the larger the size of the resulting semantic pointer should be in order to allow the discrimination of information related to the constituent elements. In fact, too much compressed semantic pointers would not be able to hold an enough amount of information.

It is easy to imagine how two semantic pointers obtained by binding sets of constituent elements with a substantial intersection are very close to each other within the same geometric space and are consequently perceived within the model as very similar. On the other hand, when two semantic pointers are obtained by binding very different vectors, it is plausible to think that they are located at a greater distance within the geometric space.

Furthermore, it can be assumed that these geometric spaces are not isolated but can be connected to each other, in more than one way. For example, as we have already said in our introduction to semantic pointers, they in effect point to more compressed representations of the same concept, thus defining a connection between mental spaces with higher detail and smaller mental spaces, in which the same concepts are represented with a higher level of compression. Furthermore, it should be observed how the binding of semantic pointers can lead to the creation of new semantic pointers that represent concepts on different contexts with respect to the contexts on which the individual constituent elements of the binding resided, creating real bridges or connections between different mental spaces.

Calculating the distance, or similarity, between concepts residing in different mental spaces, even if connected to each other, could therefore require a more complex formula than the simple calculation of the geometric distance between two vectors in the same space. This highlights a common feature between the SPMS and transformational models in which similarity is assumed to decrease monotonously as the number of transformations required to make one sequence of features identical to another increases. Even in the

new model, in fact, one could imagine how the similarity can decrease in a monotonous way as the number of transformations required to pass from the context to the other increases.

To better clarify the details of this new semantic similarity model based on the SPA in the next section we will consider some of the objections raised to the main semantic similarity models known in the literature and we will characterize in more detail how the SPMS can allow to solve some of these objections.

5.4 Criticisms to Standard Models and their Settlement

Over the years a number of criticisms against traditional models for similarity has been raised, especially against the geometric model [85]. Making an exhaustive list of all the objections that have been raised would be too difficult and is a goal that goes beyond the scope of this work. Already by itself the fact that the SPMS has features in common with almost all of the traditional similarity models seen in the previous section highlights how it inherits some of the positive aspects they propose.

However in this section we analyze some of the main objections raised against some of the traditional models in a more detailed way and show how the new model can naturally provide a plausible justification for such objections.

Specifically our analysis mainly focuses on the objections to the three axioms of the geometric model, the reflexive relation, the symmetric relation and the triangular inequality, on the objection relating to the limit to the number of nearest neighbors that can be assigned to a single concept and to the objection related to the lack of specific structure in feature-based models.

5.4.1 The reflexive relation

One of the first objections raised against the geometric model of similarity is that relating to the minimality of distance. At the basis of this objection, the hypothesis has been advanced that some concepts are, at a perceptual level, more similar to other concepts rather than to themselves. Such hypothesis would therefore be in conflict with the first of the three axioms underlying the geometric model.

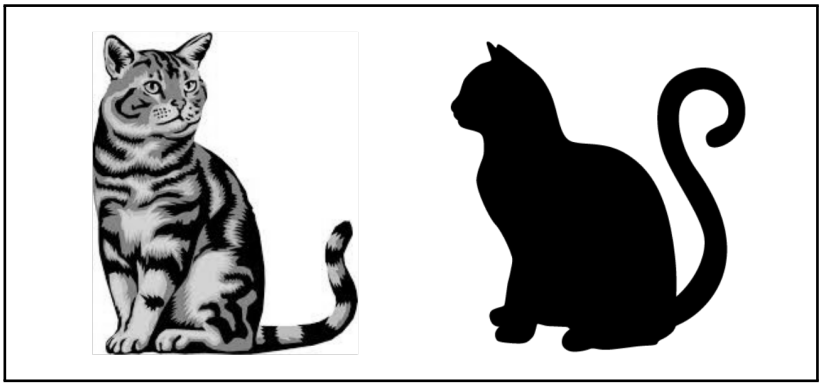


Figure 5.2: Pictures of two cats. The cat on the left presents a greater level of detail and consequently the elaboration of all the constituent elements and their consequent representation through a semantic pointer would take longer than that required for processing the visual input represented by the cat on the right. The latter is in fact much simpler and has much less details.

Podgorny and Garner in [103] hypothesize in their study, for example, that the letter C is more similar to the letter O than W is to itself. This hypothesis is made on the basis of an experimental study in which the reaction time has been used as a measure of similarity of two perceptual inputs: in this context, longer reaction times indicate a lower degree of similarity while shorter reaction times indicate a degree of higher similarity.

Based on similar considerations it has been also hypothesized that the letter S is more similar to itself than the letter W is to itself.

The SPMS suggests a justification for Podgorny & Garner's observations that do not necessarily lead us back to Laakso & Cottrell's conclusions. The higher reaction time in response to a perceptual input (in this case a visual input) can be justified by the fact that detailed and more complex perceptive inputs take longer to be processed by the visual cortex. It is plausible to imagine that, given the richness in the details, the corresponding semantic pointer resides within a large mental space whose processing is consequently slower. Furthermore, the observation and comparison of details involves the decomposition of the starting semantic pointer into its constituent elements, an operation that takes a long time to perform. On the other hand much simpler perceptual inputs can reach a higher level of compression and can therefore be processed faster.

In other words, there is a significant amount of time it takes a semantic pointer to go

through a transformation in the brain. Therefore, depending on the size and nature of the input, compressing or decompressing some semantic pointers can take longer than others.

Consider the two input images depicted in Fig. 5.2. They both represent a cat. However the cat on the left presents a greater level of detail and consequently the elaboration of all the constituent elements and their consequent representation through a semantic pointer would take longer than that required for processing the visual input represented by the cat on the right. The latter is in fact much simpler and has much less details. In other words, the processing of the visual information on the left requires a longer reaction time since the information it carries is much more.

This also suggests that perhaps, in many circumstances, the reaction time is not the best measure of similarity for calculating the similarity between two concepts, at least not in all cases.

5.4.2 The symmetric relation

The symmetric property typical of the geometric similarity model has also often been criticized. This property implies that, given two concepts x and y, the measure of similarity of x towards y should be the same if computed for y towards x.

Obviously, in a geometric model this property always holds since the distance from x to y is the same as that from y to x. However, some studies highlight that this property is not always valid, thus highlighting how the geometric model is not the most suitable for experimenting the measure of similarity between two concepts.

A famous criticism in this direction is that raised by Laakso & Cottrell and according to which it is assumed that North Korea is perceived much more like China than China can be perceived similar to North Korea. More recently, Polk et al. [104] found that when the frequency of colors is experimentally manipulated, rare colors are judged to be more similar to common colours than common colours are to rare colours.

Such criticism is again based on the idea that the judgments of similarity are all formulated in the same mental space of comparison. By adopting the SPMS and the idea that the spaces of comparison may change with the context or with the size of the semantic pointers, these problems can be overcome.

It can easily be assumed that China is better known than North Korea. In other words, people generally know much more details about China (size, population, language, currency, economy, politics, etc.) than about North Korea. China is therefore a more relevant concept than North Korea.

Consequently when people think of China or compare it with something else, many more details come to mind than when people think to North Korea. This implies that the semantic pointer representing China can be much richer and made up of many more constituent elements than the semantic pointer representing the North Korea concept.

On the SPMS, making this kind of similarity judgment is not as simple as measuring a distance between two distinct points within a mental space. Rather, the constituent elements of the two semantic pointers are examined in order to identify those of the first concept that correspond to those of the second. However, it seems to matter which of the two concepts is introduced first, namely the question "how similar is x to y?" is semantically different from the question "how similar is y to x?". In the first question x is introduced first and then the subject refers to the constituent elements of x that he knows and that he relates to the known constituent elements of y. In the second question y is introduced first and then the symmetrical process takes place .

Consider a simple example where 6 features of x are known:

- x has a rectangular shape
- x is taller than wide
- x is colored red
- x is rough to the touch
- x is very nice
- x speaks Russian

While only 3 features of y are known:

- y is red
- y is rectangular in shape

- y speaks Russian

According to these definitions, x is a more relevant concept than y.

When x is introduced before y, the comparison is made by evaluating the features of x and trying to put them in association with those of y. In the specific case, the subject recognizes 3 features in common with y and 3 features that do not find any correspondence. Roughly, we could say that the similarity of x to y would be estimated at 50%.

Otherwise, when y is introduced before x, the comparison is made by evaluating the features of y and associating them with those of x. In our example, the subject finds 3 features out of 3 in common with x, from which it follows that the similarity of y with x would be valued at 100%.

This example provides a justification for why it seems that a less relevant concept is more like a more relevant concept than the other way around. These are counter-intuitive judgments that are due to a different knowledge of the concepts being compared.

5.4.3 Triangle inequality

Regarding the triangle inequality Tversky and Gati [135] found some violations when it is combined with an assumption of segmental additivity. Specifically it turns out that in a standard geometric model, given three concepts x, y and z, we would aspect that $\delta(x,y)+\delta(y,z)=\delta(x,z)$, if x, y, and z lie on a line. In general, if they do not lie in the same line we have that $\delta(x,y)+\delta(y,z)\geq\delta(x,z)$.

However, it is easy to find violations of these assumptions in the common perception of similarity between concepts. Imagine considering the degree of similarity between the concepts dog, cat and blind. It is easy to hypothesize that the cat - dog couple and the dog - blind couple are pairs of concepts that are very close to each other in terms of similarity: the first because both concepts refer to two common domestic animals, the second because the dog is very often used as an aid for blind people. However, it is difficult to assume that the distance between cat and blind is small enough to justify the triangular inequality in a geometric model.

Once again the problem from which this objection arises is that it is assumed that the comparison is made within the same mental space. If, on the other hand, one accepts that

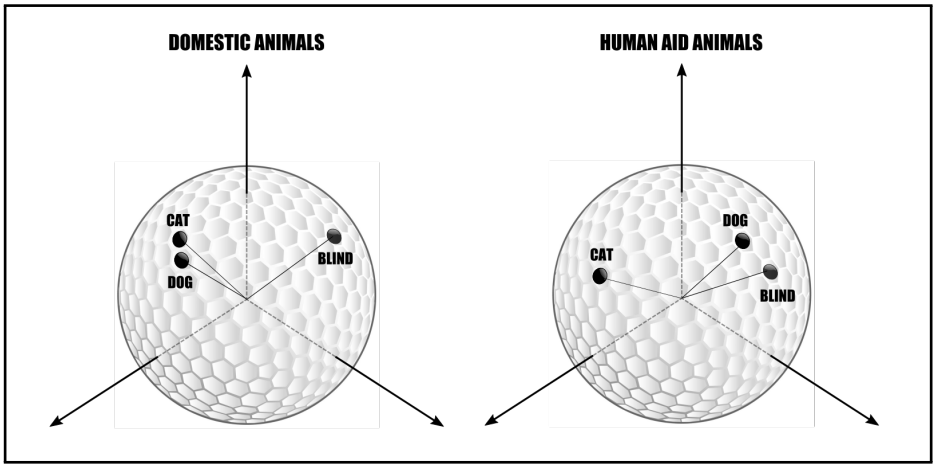


Figure 5.3: The comparison between the concepts dog, cat and blind in the SPMS. The comparison between dog and cat is made within the context relating to domestic animals in which the two concepts are very close. Similarly the comparison between dog and blind is done within the context relating to animals as an aid to humans, in which once again the two concepts are very close. Within these two contexts, however, the concepts of cat and blind are located in very distant points

different comparisons can take place in different mental spaces, there is an easy justification for this inconsistency.

In the SPSM model the comparison between dog and cat would be made within a context, that relating to domestic animals, in which the two concepts are very close. Similarly the comparison between dog and blind would be done within a second context, that relating to animals as an aid to humans, in which once again the two concepts are very close. Within these two contexts, however, the concepts of cat and blind would probably be found in very distant points, justifying the hypothesis from which the objection arises (see Fig. 5.3). Whether the comparison is made on the first or second context probably depends on whether the term cat or the term blind is introduced first, as discussed in the previous section.

In some cases it is also possible that one of the two terms does not have its own representation within a specific context. For example, the concept of cat may not appear within the context of animals as an aid to humans. In this case, calculating the similarity between the two concepts would also require additional work necessary to move from one context to another.

5.4.4 Limit on the number of closest neighbors

A further criticism of geometric similarity models is due to the upper limit on the number of closest neighbors a concept can have in a mental space. Tversky & Hutchinson in [136] proposed a first study in which this criticism was raised. Based on this study, it is suggested that geometric models impose an upper limit on the number of points that can share the same closest neighbor. A much more restrictive limit is implied in the hypothesis that the data points represent a sample of a continuous distribution in a multidimensional Euclidean space (see Fig. 5.3). By analyzing 100 datasets Tversky & Hutchinson show that most perceptual inputs satisfy the geometric-statistical limit, while many conceptual input sets exceed it.

Carol Krumhansl [61] propose a solution that allows to improve the geometric similarity models in order to solve this specific problem. On the basis of this proposal, the dissimilarity between two concepts is modeled in terms of both the distance between elements in a mental space and the spatial density in the proximity of the compared elements. In this context, spatial density is defined as the number of elements positioned in proximity to the element within the mental space. The more concepts there are in the proximity of the element, the greater the spatial density of the element. It follows that the elements are more dissimilar if they have many elements surrounding them (their spatial density is high) than if they have few neighboring elements.

By including spatial density in the similarity calculation violations of the principles of minimality, symmetry and triangular inequality can also potentially be explained, as well as part of the influence of context on similarity. However, the empirical validity of the spatial density hypothesis has been questioned several times [18, 19, 62, 135].

The SPMS model naturally justifies the criticism raised by Tversky & Hutchinson [136] about the upper limit on the number of points that can share the same closest neighbor. In fact, if we hypothesize that the comparisons between concepts can take place in different mental spaces, each of which identifies a different context, we could also hypothesize that the same concept can admit different representations, on the basis of the context in which it is evaluated. If you think of the concept of cat in the context of domestic animals, then probably the representation of this concept by means of a semantic pointer will contain

as main constituent elements such qualities as tenderness, small size, soft hair, intelligence, cleaning, etc. If, on the other hand, we think of the concept of cat in the context of predatory animals, then probably the semantic pointer it represents of this concept will contain other qualities as main constituent elements such as speed, shrewdness, instinct, intelligence, etc. Some constituent elements will probably be in common between the two representations, but they would still be two distinct semantic pointers inserted in two different mental spaces, because they refer to different contexts.

This hypothesis, valid only if we hypothesize that we can have different mental spaces correlated with each other, allows us to overcome this specific criticism. The number of elements that can share the same closest neighbor is in fact higher than that imposed by the traditional geometric model, since it is possible to find the same concept within different mental spaces.

In our example, within the first mental space the concept of cat could have as its closest neighbor the concept of dog, as a pet. Similarly, within the second mental space, the cat may have the lynx as its closest neighbor, as a small predatory feline.

This hypothesis is much more true if we consider the fact that as the details with which concepts are evaluated increase, we move from small mental spaces to large mental spaces, while remaining within the same context. In our previous example, if at a superficial evaluation (therefore at a high degree of compression of the pointers) it is easy to hypothesize that the concept of cat is associated with that of dog, it is true that if we imagine increasing the degree of detail with which we evaluate the concept of cat, adding for example some considerations on the breed, we could hypothesize that the concept of European cat has as its closest neighbor that of the Carthusian cat rather than the concept of dog.

5.4.5 Unstructured representations in features based models.

One of the main criticisms leveled against feature-based models is that they use unstructured representations, where the representation of concepts is simply given by a set of unrelated features.

To solve this problem, over the years some solutions have been proposed based mainly

on two basic ideas, namely that of organizing the characteristics on the basis of a propositional structure and that of organizing them on the basis of a hierarchical structure.

In the propositional approach [92] the characteristics that are part of a concept are related to each other by statements drawn mainly from the visual domain, such as above, near, right, inside, greater than, etc.

In a very similar way in the hierarchical approach, characteristics represent entities that are incorporated into each other, that is, related to each other by statements such as ”part of” or ”a type of”.

The SPMS model natively uses a model very similar to the propositional approach. Indeed, it has been shown in various studies how structured representations of concepts, such as those we find formulated in natural language used every day, are essential for the explanation of a wide variety of cognitive behaviors.

Typical structured representations are natural language expressions such as ”The dog chases the boy”. The underlying structure of this expression is the grammar of the language, while its component parts are the individual words. In an artificial language, such as those typically used in computers, a similar sentence can be represented with the structured representation chases (dog, boy). This representation is structured because the position in which each term appears determines its grammatical role. The verb is the first, the agent is the second and the complement is the third. If the order of the elements changes, the meaning of the structured representation also changes.

The possibility of being able to link two different vector representations at the base of the SPMS is of fundamental importance in the definition of our similarity model since, if we are able to link vectors together, then we are able to define a role for each pointer that does part of a complex structure, tagging vectors of content with vectors having a structural role. For example, it would be possible to associate through the ligature the semantic pointer to chase (with content function) to the semantic verb pointer (with structural function), to indicate that the role played by chasing is a verbal role.

6

Conclusions and Future Works

In this research we focused on the contextual effects on semantics, investigating such effects on a disambiguation task using neural computational simulation and proposing a novel context sensitive cognitive account of similarity.

In our disambiguation task described in Chapter 4, provided with a sentence, admitting two or more candidate interpretations, and an image that depicts the content of the sentence, it is required to choose the correct interpretation of the sentence depending on the image's content. Thus we address the problem of selecting the interpretation of an ambiguous sentence matching the content of a given image.

This kind of task is also fundamental to the problem of grounding vision in language, by focusing on phenomena of linguistic ambiguity, which are prevalent in language, but typically overlooked when using language as a medium for expressing understanding of visual content. Due to such ambiguities, a superficially appropriate description of a visual scene may in fact not be sufficient for demonstrating a correct understanding of the relevant visual content.

Regarding our new contextual account of similarity introduced in Chapter 5, we suggested that most of the traditional similarity models which have been proposed over the years can converge on a generalized model of similarity in which the context plays a fundamental role in order to overcome all the criticisms raised over the years to each of the traditional similarity models.

From the neurocomputational point of view, our models were based on the Eliasmith's Neural Engineering Network (NEF) [27] and Nengo[1], the python library which serves as

[1]Nengo is available at `https://www.nengo.ai`

an implementation of the NEF. We described the NEF and Nengo in Chapter 1. The basic semantic component within NEF is the so-called Semantic Pointer Architecture (SPA) [129], which determines how the concepts are represented as dymanic neural assemblies. We described the details of the SPA in Chapter 3.

In the next sections we will briefly give some general consideration about this field and discuss some possible future developments of our research highlighting some applications in which our neural model could be employed.

We thank the reviewers of the doctoral thesis for giving some interesting insights on which to investigate and for suggesting some interesting directions in which to continue our research. Below we discuss in detail a couple of the suggestions received during the review phase.

6.1 General Consideration

Despite the enormous efforts being made in this direction, the question of what is the advantage of running simulations on this type of large-scale neural networks remains controversial. However, the motivations that drive research interest towards the creation of biologically plausible cognitive models and towards realistic models of neurons are increasingly shared by scientific community.

In the first place, having very realistic biologically models allows to better verify the theories on which these models are based. If you want to understand exactly how a brain works, it is not enough to reproduce its correct behavior; this needs we need a fair compromise, since a excess of realism in the way the brain would work renders the models useless [26, 96, 30]. However, it should be possible to have models of neural activation and connectivity that are "comparable" to natural ones. The same effects of degeneration should be observed neural, injury, deep brain stimulation and even various drug treatments.

This pushes our interest towards the NEF model rather than towards algorithmic simulation models of reasoning. Secondly, the creation of realistic cognitive models is able to suggest new types of algorithmic solutions. In fact, when using NEF, you do not get an exact implementation of any specified algorithm. Neurons approximate that algorithm,

and the accuracy of that approximation depends not only on the neural properties, but also on the functions that are calculated. This implies that, instead of applying any calculation to the model, the NEF forces us to use the basic operations available for neurons, allowing us to understand which classes of algorithms cannot be implemented in the human brain.

Although many researchers believe that the great challenge in depicting large-scale neural networks is to improve the biological realism of simulations, the idea that the main challenge is not biological realism is increasingly widespread; rather it consists in defining the general principles of neural computation that could allow the artificial brain to show the robust flexibility characteristic of biological cognitive processes. The NEF and the Semantic Pointer Architecture, on which it is based, are taking their first steps in this direction.

6.2 The Winograd Challenge

One of the possible future applications of our neural model for context-based linguistic disambiguation described in Chapter 4 is the "Winograd Schemas", based on the anaphora resolution for dealing with commonsense queries.

A *Winograd schema* is a challenge where a pair of sentences that differ in only one or two words and that contain an ambiguity is resolved in opposite ways in the two sentences. Its solution requires the use of world knowledge and reasoning for its resolution. The schema takes its name from a well-known example by Terry Winograd:

> The city councilmen refused the demonstrators a permit because they [feared/advocated] violence.

If the word is "feared", then "they" presumably refers to the city council; if it is "advocated" then "they" presumably refers to the demonstrators.

In his paper, "The Winograd Schema Challenge" Hector Levesque [69] proposes to assemble a set of such Winograd schemas that are

- easily disambiguated by the human reader (ideally, so easily that the reader does not even notice that there is an ambiguity);

- not solvable by simple techniques such as selectional restrictions;
- Google-proof; that is, there is no obvious statistical test over text corpora that will reliably disambiguate these correctly.

The set would then be presented as a challenge for AI programs, along the lines of the Turing test. The strengths of the challenge are that it is clear-cut, in that the answer to each schema is a binary choice; vivid, in that it is obvious to non-experts that a program that fails to get the right answers clearly has serious gaps in its understanding; and difficult, in that it is far beyond the current state of the art.

A contest, entitled the Winograd Schema Challenge was run once, in 2016. At that time, there was a cash prize offered for achieving human-level performance in the contest. Since then, the sponsor has withdrawn;

What we would like to verify in our future studies is the applicability of our context-based neural disambiguation model to the challenge described above. In this context, the model would receive as input a sentence extracted from the challenge, together with two further sentences that allow to carry out a disambiguation on the basis of the context. It would already be an excellent result to be able to obtain small margins of error on the first set of phrases proposed by the challenge, namely those easily disambiguated by the human reader.

6.3 Gardenfors' Conceptual Spaces

In his book "Conceptual Spaces - The Geometry of Thought" [35], Peter Gardenfors presents a pioneering theory for representing conceptual knowledge, the basic construct of human thinking and reasoning [47]. Gardenfors' theory is closely related to our proposal for a new semantic similarity model introduced in Chapter 5 and has some commonalities. This suggests a more accurate comparison between the two theories, also from a computational point of view.

The conceptual level of the Gardenfors' theory is not seen as an alternative to traditional approaches of knowledge representation in artificial intelligence, namely symbolic or sub-symbolic methods. Instead, it is meant to complement both approaches.

His work is highly recommendable and worth reading, as it does not only tackle many fundamental problems of knowledge representation such as grounding [42], concept formation and similarity comparisons [41], but also outlines novel and enlightening ideas how to overcome these. The book introduces the notion of a conceptual space as a framework for representing knowledge at the conceptual level. It is motivated by contrasting it to other levels of knowledge representation: The highest level, the symbolic level, conceptualizes the human brain as a computing device.

Knowledge is represented based on a language consisting of a set of symbols. Logical operations can be performed on these symbols to infer new knowledge. Human reasoning is modeled as a symbol manipulation process. Classical, symbolic artificial intelligence does not very well support central cognitive processes such as the acquisition or formation of new concepts and similarity comparisons.

The lowest level, the sub-symbolic knowledge representation, is oriented towards the neuro-biological structure of the human brain. Concepts are implicitly represented via activation patterns within the neural network. Learning is modeled by modifying the activation of neurons. Explicit representation of knowledge and concepts is not possible.

Specifically interesting for our purposes, at the intersection between the symbolic and the sub-symbolic level, Gardenfors introduces the conceptual level. With some degree of similarity to our model his theory of conceptual spaces is based on semantic spaces with a geometric structure: A conceptual space is formed by a set of quality dimensions. One or several quality dimensions model one domain. An important example used is the color domain represented by the quality dimensions hue, saturation and brightness. Conceptual spaces have a cognitive foundation because domains can be grounded in qualities perceivable by the human sensory apparatus. Concepts are represented as conceptual regions described by their properties on the quality dimensions.

Also in this case, and as happens in our model of similarity, the geometric structure of conceptual spaces makes it possible to determine distances and therefore provides an inherent similarity measure by taking the distance in the conceptual space as indicator of the semantic similarity. The notion of similarity is an important construct for modeling categorization and concept formation. Using similarity for reasoning can also reflect well the vagueness typical for human reasoning.

The strong focus on the cognitive foundation makes his theory particularly valuable. It contains many challenging claims which are related to various disciplines by giving evidence from a wide range of literature.

Unfortunately, Gardenfors describes his theory only at a very abstract level and forbears from describing algorithms for the formalization of his theory. The realization of a computational model for conceptual spaces bears many practical problems which still have to be solved. Moreover, no empirical evidence is given for his pioneering, sometimes revolutionary ideas. However, these shortcomings should be considered as challenges to solve in the future.

Bibliography

[1] Ahn, J., Ahn, J., and Lee, I. (2014), *Intact CA3 in the hippocampus is only sufficient for contextual behavior based on well-learned and unaltered visual background*, Hippocampus 24(9). doi: 10.1002/hipo.22292

[2] Airenti, G. and Plebe, A. Editorial: Context in communication: A cognitive view. *Frontiers in Psychology*, 8:115, 2017.

[3] Amaral, D. G., and Witter, M. P. (1983). *The three-dimensional organization of the hippocampal formation: a review of anatomical data*. Neuroscience 31, 571-591. doi: 10.1016/0306-4522(89)90424-7

[4] Amaral, D., Lavenex, P. (2006). *Chapter 3. Hippocampal Neuroanatomy*. In Andersen P, Morris R, Amaral D, Bliss T, O'Keefe J. *The Hippocampus Book*. Oxford University Press. ISBN 978-0-19-510027-3.

[5] Aubin, S., Voelker, A. R., and Eliasmith, C. Improving with practice: A neural model of mathematical development. *topiCS*, 9(1):6–20, 2017.

[6] Barnard, K. and Johnson, M. Word sense disambiguation with pictures. *Artificial Intelligence*, 167:13–30, 2005.

[7] Barsalou, L. W. Ad hoc concepts. *Memory and Cognition*, 11:211–217, 1983.

[8] Bates, E., Dal, P. S., and Thal, D. Individual differences and their implications for theories of language development. In Paul Fletcher and Brian Mac Whinney, editors, *Handbook of child language*, pages 96–151. Basil Blackwell, Oxford (UK), 1995.

[9] Beal, M.J. (2003). Variational algorithms for approximate Bayesian inference. Physics. University of Cambridge. PhD Thesis.

[10] Bednar, J. A., Choe, Y., De Paula, J., Miikkulainen, R., Provost, J., and Tversky, T.. Modeling cortical map with Topographica. *Neurocomputing*, 58–60:1129–1135, 2004.

[11] Bornstein, M. H., and Cote, L. R.. Cross-linguistic analysis of vocabulary in young children: Spanish, dutch, french, hebrew, italian, korean, and american english. *Child Development*, 75:1115–1139, 2004.

[12] Berzak, Y., Barbu, A., Harari, D., Katz, B., and Ullman, S.. Do you see what i mean? visual resolution of linguistic ambiguities. In *Conference on Empirical Methods in Natural Language Processing*, pages 1477–1487, 2015.

[13] Bower, j. M., and Beeman, D. *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System*. Springer-Verlag, New York, second edition, 1998.

[14] Cerasti, E., and Treves, A. (2013). *The spatial representations acquired in CA3 by self-organizing recurrent connections*. Front. Cell. Neurosci. 7:112. doi: 10.3389/fncel.2013.00112

[15] Carroll, J. D., & Wish, M. (1974). Models and methods for three-way multidimensional scaling. In D. H. Krantz, R. C. Atkinson, R. D. Luce, & P. Suppes (Eds.), Contemporary developments in mathematical psychology(Vol.2,pp.57–105). San Francisco: Freeman.

[16] Cherubini, E. and Miles, R. (2015), *The CA3 region of the hippocampus: how is it? What is it for? How does it do it?*, Front. Cell. Neurosci. 9:19. doi: 10.3389/fncel.2015.00019

[17] Coco, M. I., and Keller, F. The interaction of visual and linguistic saliency during syntactic ambiguity resolution. *Quarterly Journal of Experimental Psychology*, 68:46–74, 2015.

[18] Corter, J. E. (1987). Similarity, confusability, and the density hypothesis. Journal of Experimental Psychology: General, 1 1 6, 238–249.

[19] Corter, J. E. (1988). Testing the density hypothesis: Reply to Krumhansl. Journal of Experimental Psychology: General, 117, 105–106.

[20] Christie, G., Laddha, A., Agrawal, A.,Antol1 Yash Goyal, S. Kochersberger, K., and Batra, D. Resolving language and vision ambiguities together: Joint segmentation & prepositional attachment resolution in captioned scenes. In *Empirical Methods in Natural Language Processing*, pages 1493–1503. Association for Computational Linguistics, 2016.

[21] Deshmukh, S. S., and Knierim, J. J. (2011). *Representation of nonspatial and spatial information in the lateral entorhinal cortex*. Front. Behav. Neurosci. 5:69. doi: 10.3389/fnbeh.2011.00069

[22] Deshmukh, S. S., Johnson, J. L., and Knierim, J. J. (2012). *Perirhinal cortex represents nonspatial, but not spatial, information in rats foraging in the presence of objects: comparison with lateral entorhinal cortex*. Hippocampus 22, 2045?2058.

[23] Eichenbaum, H., Cohen, N. J. (2001). Oxford psychology series; no. 35.From conditioning to conscious recollection: Memory systems of the brain. Oxford University Press.

[24] Eichenbaum, H., and Lipton, P. A. (2008). *Towards a functional organization of the medial temporal lobe memory system: role of the parahippocampal and medial entorhinal cortical areas*. Hippocampus 18, 1314?1324.

[25] Eliasmith, C. Cognition with neurons: A large-scale, biologically realistic model of the Wason task. In G. Bara, L. Barsalou, and M. Bucciarelli, editors, *Proceedings of the 27th Annual Meeting of the Cognitive Science Society*, 2005.

[26] Eliasmith C. How we ought to describe computation in the brain. Stud Hist Philos Sci. 2010 Sep;41(3):313-20. doi: 10.1016/j.shpsa.2010.07.001. PMID: 21466123.

[27] Eliasmith, C. *How to build a brain: a neural architecture for biological cognition*. Oxford University Press, Oxford (UK), 2013.

[28] Eliasmith, C., and Anderson, C. H. *Neural Engineering Computation, Representation, and Dynamics in Neurobiological Systems*. MIT Press, Cambridge (MA), 2003.

[29] Elman, J. L., Bates, E., Johnson, M. H., Karmiloff-Smith, A. Parisi, D. and Plunkett, K. *Rethinking innateness A Connectionist Perspective on Development*. MIT Press, Cambridge (MA), 1996.

[30] Eliasmith, C., Trujillo, O. (2014) The use and abuse of large-scale brain models, Current Opinion in Neurobiology, 25, pp. 1–6, ISSN 0959-4388.

[31] Falkenhainer, B., Forbus, K. D., Gentner, D. (1989). The structure-mapping engine: Algorithm and examples. Artificial Intelligence, 41, 1–63.

[32] Felleman, D. J. & Van Essen, D. C. (1991). Distributed hierarchical processing in primate visual cortex. Cerebral Cortex, 1, 1–47.

[33] Fyhn, M., Molden, S., Witter, M. P., Moser, E. I., and Moser, M. B. (2004). *Spatial representation in the entorhinal cortex*. Science 305, 1258?1264.

[34] Furtak, S. C., Wei, S. M., Agster, K. L., and Burwell, R. D. (2007). *Functional neuroanatomy of the parahippocampal region in the rat: the perirhinal and postrhinal cortices*. Hippocampus 17, 709?722.

[35] Gardenfors, P. (2000) The Geometry of Thought. A Bradford Book.

[36] Goldstone, R. L. (1994). The role of similarity in categorization: Providing a groundwork. Cognition, 52, 125–57.

[37] Goodman, N. (1972). Seven strictures on similar- ity. In N. Goodman (Ed.), Problems and projects (pp. 437–446). New York: The Bobbs-Merrill Co.

[38] Goodwin, C. and Duranti, A. Rethinking context: an introduction. In Alessandro Duranti and Charles Goodwin, editors, *Rethinking context: Language as an interactive phenomenon*, pages 1–42, Cambridge (UK), 1992. Cambridge University Press.

[39] Hahn, U. (2003). Similarity.

[40] Hahn, U., Chater, N., and Richardson, L. (2003). Similarity as transformation. *Cognition*, 87:1–32.

[41] Hahn, U., Ramscar, M. (2001) Similarity and categorization. Oxford University Press.

[42] Harnard, S. (1990) The Symbol Grounding Problem. Physica D: Non-linear Phenomena, 42:335-346.

[43] Hines, M. and Carnevale, N. The NEURON simulation environment. *Neural Computation*, 9:1179–1209, 1997.

[44] Hofstadter, D. (1997). Fluid concepts and creative analogies: Computer models of the fundamental mechanisms of thought. New York: Basic Books.

[45] Holland, J. H., Holyoak, K. J., Nisbett, R. E., & Thagard, P. R. (1986). Induction: Processes of inference, learning, and discovery. Cambridge, MA: Bradford Books/MIT Press.

[46] Holyoak, K. J., & Koh, K. (1987). Surface and structural similarity in analogical transfer. Memory & Cognition, 15, 332–340.

[47] Holyoak, K., Morrison, R. (2005) The Cambridge Handbook of Think-ing and Reasoning. Cambridge University Press

[48] Holyoak, K. J., Morrison, R. G., Goldstone, R. L., and Son, J. Y. (2012). Similarity.

[49] Hasselmo, M. E., Wyble, B. P. (1997). Free recall and recognition in a network model of the hippocampus: Simulating effects of scopolamine on human memory function. Behavioural Brain Research, 89(1-2), 1–34.

[50] Hindle, D. and Rooth, M. Structural ambiguity and lexical relations. *Cognitive Linguistics*, 19:103–120, 1993.

[51] Hunsberger, E. and Eliasmith, C. Spiking deep networks with LIF neurons. *CoRR*, abs/1510.08829, 2015.

[52] Hafting, T., Fyhn, M., Molden, S., Moser, M. B., Moser, E. I. (2005). *Microstructure of a spatial map in the entorhinal cortex*. Nature 436: 801-806. 10.1038/nature03721

[53] Ive, J., Gkotsis, G., Dutta, R., Stewart, R., and Velupillai, S. Hierarchical neural model with attention mechanisms for the classification of social media text related to mental health. In Kate Loveys, Kate Niederhoffer, Emily Prud'hommeaux, Rebecca Resnik, and Philip Resnik, editors, *Proceedings of the Fifth Workshop on Computational Linguistics and Clinical Psychology: From Keyboard to Clinic, CLPsych@NAACL-HTL, New Orleans, LA, USA, June 2018*, pages 69–77. Association for Computational Linguistics, 2018.

[54] Karpathy A, Fei-Fei L (2015) Deep visual-semantic alignments for generating image descriptions. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition

[55] Kawato, M. (1995). Cerebellum and motor control. In M. Arbib (Ed.), The handbook of brain theory and neural networks. Cambridge, MA: MIT Press.

[56] Kerr, K. M., Agster, K. L., Furtak, S. C., and Burwell, R. D. (2007). *Functional neuroanatomy of the parahippocampal region: the lateral and medial entorhinal areas*. Hippocampus 17, 697-708.

[57] Kesner, R. P. (2013). A process analysis of the CA3 subregion of the hippocampus. Front. Cell. Neurosci. 7:78. doi: 10.3389/fncel.2013.00078

[58] Kesner, R. P., and Hunsaker, M. R. (2010). The temporal attributes of episodic memory. Behav. Brain Res. 215, 299-309. doi: 10.1016/j.bbr.2009.12.029

[59] Knierim, J. J., Lee, I., and Hargreaves, E. L. (2006). *Hippocampal place cells: parallel input streams, subregional processing, and implications for episodic memory*. Hippocampus 16, 755?764.

[60] Kong, C., Lin, D., Bansal, M., Urtasun, R., and Fidler, S. What are you talking about? text-to-image coreference. In *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 3558–3565, 2014.

[61] Krumhansl, C. L. (1978). Concerning the applicability of geometric models to similarity data: The interrelationship between similarity and spatial density. Psychological Review, 85, 450–463.

[62] Krumhansl, C. L. (1988). Testing the density hypothesis: Comment on Corter. Journal of Experimental Psychology: General, 117, 101–104.

[63] Laakso, A. and Cottrell, G. (2005). *Churchland on connectionism*. Cambridge University Press.

[64] Landau, B., Smith, L. B. and Jones, S. The importance of shape in early lexical learning. *Cognitive Development*, 3:299–321, 1988.

[65] Lazaridou, A., The Pham, N., and Baroni, M. Combining language and vision with a multimodal skip-gram model. *CoRR*, abs/1501.02598, 2015.

[66] Lee, M., Chang, A., Peirsman, Y., Chambers, N., Surdeanu, M., and Jurafsky, D. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39:885–916, 2013.

[67] Lee, I., and Lee, C. H. (2013), *Contextual behavior and neural circuits*, Front. Neural Circuits, doi:10.3389/fncir.2013.00084

[68] Lee I, Kesner RP. Encoding versus retrieval of spatial memory: double dissociation between the dentate gyrus and the perforant path inputs into CA3 in the dorsal hippocampus. Hippocampus. 2004; 14:66–76.

[69] Levesque, H. (2011) The Winograd Schema Challenge, Commonsense.

[70] Lorente de No, R. (1934). *Studies on the structure of the cerebral cortex: continuation of the study of the ammonic system*. J. Psychol. Neurol. 46, 113-177.

[71] Machery, E. By default: Concepts are accessed in a context-independent manner. In Stephen Laurence and Eric Margolis, editors, *The Conceptual Mind: New Directions in the Study of Concepts*. MIT Press, Cambridge (MA), 2015.

[72] MacWhinney, B. editor. *The Emergence of Language*. Lawrence Erlbaum Associates, Mahwah (NJ), second edition, 1999.

[73] Markram, H. The blue brain project. *Nature Reviews Neuroscience*, 7:153–160, 2006.

[74] Markman, A. B., & Gentner, D. (1993). Structural alignment during similarity comparisons. Cognitive Psychology, 25, 431–467.

[75] Markman, A. B., & Gentner, D. (1993b). Splitting the differences: A structural alignment view of similarity. Journal of Memory & Language, 32, 517–535.

[76] Markman, A. B., & Gentner, D. (1996). Commonalities and differences in similarity comparisons. Memory & Cognition, 24, 235–249

[77] Markman, A. B., & Gentner, D. (1997). The effects of alignability on memory. Psychological Science, 8, 363–367.

[78] Markram, H., Muller, E., Ramaswamy, S., and Reimann, M. W. Reconstruction and simulation of neocortical microcircuitry. *Cell*, 163:456–492, 2015.

[79] Marr, D. (1970) A theory for cerebral neocortex, PRSLB, vol.176, pp. 161–234

[80] Marr, D. (1971) *Simple memory: a theory for archicortex*. Philosophical Transactions of the Royal Society of London, Series B 262: 23-81. 10.1098/rstb.1971.0078

[81] Martinez-Conde, S., Macknik, S. L., and Hubel, D. H. (2004). *The role of fixational eye movements in visual perception*. Nature Review of Neuroscience, 5, 229-240.

[82] Mazzone, M. and Lalumera, E. Concepts: Stored or created? *Minds and Machines*, 20:47–68, 2009.

[83] McNaughton, B. L., and Morris, R. G. M. (1987). Hippocampal synaptic enhancement and information storage within a distributed memory system. Trends Neurosci. 10, 408–415. doi: 10.1016/0166-2236 (87)90011-7

[84] Mitchell, M. (1993). Analogy-making as perception: A computer model. Cambridge, MA: MIT Press.

[85] Nickerson, R. S. (1972). Binary classification re- action time: A review of some studies of human information-processing capabilities. Psychonomic Monograph Supplements, 4 (whole no. 6), 275–317.

[86] Nikolic, D., Hausler, S., Singer, W., and Maass, W. (2009). *Distributed fading memory for stimulus properties in the primary visual cortex*. PLoS Biology, 7, e1000260.

[87] O'Keefe, J., and Nadel, L. (1978). *The Hippocampus as a Cognitive Map*. Oxford, UK: Oxford University Press.

[88] O'Reilly, R. C., McClelland, J. L. (1994). *Hippocampal conjunctive encoding, storage, and recall: avoiding a trade-off*. Hippocampus 4: 661-682. 10.1002/hipo.450040605

[89] Oztop, E., Kawato, M., & Arbib, M. (2006). Mirror neurons and imitation : A computationally guided review. Neural Networks, 19, 254–271.

[90] Paivio, A. (1971). Imagery and verbal processes. Holt, New York, NY: Rinehart and Winston.

[91] Paivio, A. (1986). Mental representations: A dual coding approach. New York: Oxford University Press.

[92] Palmer, S. E. (1975). Visual perception and world knowledge. In D. A. Norman & D. E. Rumelhart (Eds.), Explorations in cognition (pp. 279–307). San Francisco: W. H. Freeman.

[93] Pavone, A., Plebe, A. (2020) *A Cognitive Model for Resolving Semantic Ambiguities by Context Information*. Accepted to AISC Midterm Conference 2020 *Cognitive Science meets the Humanities*

[94] Pavone, A., Plebe, A. (2020) *A Cognitive Account of Similarity Based on the Semantic Pointer Architecture*. Accpted to AISC Midterm Conference 2020 *Cognitive Science meets the Humanities*

[95] Pavone, A., Plebe, A. (2020) *Neural Semantic Pointers in Context*. Proceedings of the 12th International Conference on Neural Computation Theory and Application (NCTA) part of International Joint Conference on Computational Intelligence (IJCCI), pp.447-454

[96] Piccinini, G., Craver, C. Integrating psychology and neuroscience: functional analyses as mechanism sketches. Synthese 183, 283–311 (2011).

[97] Plate, T. A. (2003), *Holographic Reduced Representation: Distributed Representation for Cognitive Structures*, Holographic Reduced Representation: Distributed Representation for Cognitive Structures, CSLI Publications Stanford, CA, USA. ISBN:1575864290

[98] Plebe, A., and De La Cruz, V. M. *Neurosemantics – Neural Processes and the Construction of Linguistic Meaning*. Springer, Berlin, 2016.

[99] Plebe, A., and De La Cruz, V. M. Neural representations in context. In Antonino Pennisi and Alessandra Falzone, editors, *The Extended Theory of Cognitive Creativity – Interdisciplinary Approaches to Performativity*, pages 285–300. Springer, Berlin, 2020.

[100] Plebe, A., and Grasso G. The unbearable shallow understanding of deep learning. *Minds and Machines*, 29:515–553, 2019.

[101] Plebe, A., Mazzone, M., and De La Cruz, V. M. First words learning: A cortical model. *Cognitive Computation*, 2:217–229, 2010.

[102] Plebe, A., Mazzone, M., and De La Cruz, V. M. A biologically inspired neural model of vision-language integration. *Neural Network World*, 21:227–249, 2011.

[103] Podgorny, P., & Garner, W. R. (1979). Reaction time as a measure of interintraobject visual similarity: Letters of the alphabet. Perception & Psychophysics, 26, 37–52.

[104] Polk, T. A., Behensky, C., Gonzalez, R., & Smith, E. E. (2002). Rating the similarity of simple perceptual stimuli: Asymmetries induced by manipulating exposure frequency. Cognition, 82, B75–B88.

[105] Quine, W. (1969). Natural kinds. *Ontological Relativity and Other Essays*, pages 114–138.

[106] Quine, W. V. (1977). Natural kinds. In S. P. Schwartz (Ed.), Naming, necessity, and natural kinds (pp. 155–175). Ithaca, NY: Cornell University Press.

[107] Ramanathan, V., Joulin, A., Liang, P., and Fei-Fei, L. Linking people in videos with 'their' names using coreference resolution. In *Proc. of European Conference on Computer Vision*, pages 95–110, 2014.

[108] Ramon, Y., Cajal, S. (1899). Textura del Sistema Nervioso del Hombre y de los Vertebrados.

[109] Riegert, C., Galani, R., Heilig, S., Lazarus, C., Cosquer, B., Cassel. J. C. (2004). *Electrolytic lesions of the ventral subiculum weakly alter spatial memory but potentiate amphetamine-induced locomotion*. Behav. Brain Res. 152 (1): 23?34. doi:10.1016/j.bbr.2003.09.011

[110] Rogers, T. T. and McClelland, J. L. *Semantic Cognition - A Parallel Distributed Processing Approach*. MIT Press, Cambridge (MA), 2006.

[111] Hetherington, M. M., Rolls, B. J. (1996). Sensory-specific satiety: Theoretical frameworks and central characteristics. In E. D. Capaldi (Ed.), Why we eat what we eat: The psychology of eating (p. 267–290). American Psychological Association.

[112] Rolls, E. T. (1996). *A theory of hippocampal function in memory*. Hippocampus 6, 601?620.

[113] Rolls, E. T. (2007). *An attractor network in the hippocampus: Theory and neurophysiology*. Learning & Memory 14: 714-731. 10.1101/lm.631207

[114] Rolls, E. T. (2013). *A quantitative theory of the functions of the hippocampal CA3 network in memory*. Front. Cell. Neurosci. 7:98. doi: 10.3389/fncel.2013.00098

[115] Ross, B. H. (1987). This is like that: The use of earlier problems and the separation of similarity effects. Journal of Experimental Psychology: Learning, Memory, and Cognition, 13, 629–63 9.

[116] Ross, B. H. (1989). Distinguishing types of superficial similarities: Different effects on the access and use of earlier problems. Journal of Experimental Psychology: Learning, Memory, and Cognition, 1 5, 456–468.

[117] Rumelhart, D. E. and McClelland, J. L. editors. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. MIT Press, Cambridge (MA), 1986.

[118] Schaffer, K. (1892). Beitrag zur histologie der ammonshornformation. Arch. Mikroskop. Anat. 39 611?63210.1007/BF02961541

[119] Scoville, W. B., and Milner, B. (1957). *Loss of recent memory after bilateral hippocampal lesions*. J. Neurol. Neurosurg. Psychiatry 20, 11-21. doi: 10.1136/jnnp.20.1.11

[120] Searle, J. R. Literal meaning. *Erkenntnis*, 13:207–224, 1978.

[121] Shepard, R. N. (1962). The analysis of proximities: Multidimensional scaling with an unknown distance function. Part I. Psychometrika, 27, 125–140.

[122] Shepard, R. N. (1962). The analysis of proximities: Multidimensional scaling with an unknown distance function. Part II. Psychometrika, 27, 219–246.

[123] Siddharth, N., Barbu, A., and Siskind, J. M. Seeing what you're told: Sentence-guided activity recognition in video. In *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 732–739, 2014.

[124] Socher, R., Ganjoo, M. Christopher, Manning, D., and Ng, A. Zero-shot learning through cross-modal transfer. In *Advances in Neural Information Processing Systems*, pages 935–943, 2013.

[125] Spivey, M. J., Tanenhaus, M. K., Eberhard, K. M. and Sedivy, J. C. Eye movements and spoken language comprehension: Effects of visual context on syntactic ambiguity resolution. *Cognitive Psychology*, 45:447–481, 2002.

[126] Stark, S. M., Reagh, Z. M., Yassa, M. A., and Stark, C. E. L. What's in a context? cautions, limitations, and potential paths forward. *Neuroscience Letters*, 2018.

[127] Tanenhaus, M. k., Spivey-Knowlton, M. J., Eberhard, K. M. and Sedivy, J. C. Integration of visual and linguistic information in spoken language comprehension. *Science*, 268:1632–Science, 1995.

[128] Tenenbaum, J. B. (1999). Bayesian modeling of human concept learning. In M. S. Kearns, S. A. Solla, & D. A. Cohn (Eds.), Advances in neural information processing systems 1 1 (pp. 59–68). Cambridge, MA: MIT Press.

[129] Thagard, P. Cognitive architectures. In Cambridge University Press, editor, *The Cambridge handbook of cognitive science*, 2011.

[130] Treves, A., Rolls, E. T. (1994). *Computational analysis of the role of the hippocampus in memory*. Hippocampus 4: 374-391. 10.1002/hipo.450040319

[131] Treves, A., Tashiro, A., Witter, M. P., Moser, E. I. (2008). *What is the mammalian dentate gyrus good for?* Neuroscience 154: 1155-1172. 10.1016/j.neuroscience.2008.04.073

[132] Torgerson, W. S. (1958). Theory and methods of scaling. New York: Wiley.

[133] Torgerson, W. S. (1965). Multidimensionsal scaling of similarity. Psychometrika, 30, 379–393.

[134] Tversky, A. (1977). Features of similarity. *Psychological Review*, 84:327–352.

[135] Tversky, A., & Gati, I. (1982). Similarity, separability, and the triangle inequality. Psychological Review, 89, 123–154.

[136] Tversky, A., & Hutchinson, J. W. (1986). Nearest neighbor analysis of psychological spaces. Psychological Review, 93, 3–22.

[137] VanRullen, R. Perception science in the age of deep neural networks. *Frontiers in Psychology*, 8:142, 2017.

[138] Voelker, A. R., and Eliasmith, C. Methods for applying the neural engineering framework to neuromorphic hardware. *CoRR*, abs/1708.08133, 2017.

[139] Zhang, S., & Markman, A. B. (1998). Overcoming the early entrant advantage: The role of alignable and nonalignable differences. Journal of Marketing Research, 35, 413–426.

[140] Wiener-Ehrlich, W. K., Bart, W. M., & Millward, R.(1980). Ananalysis of generative representation systems. Journal of Mathematical Psychology, 21 (3), 219–246.

[141] Wolpert, D. M. & Kawato, M. (1998). Multiple paired forward and inverse models for motor control. Neural Networks, 11, 1317–1329.