Chapter 3

Biological cognition – semantics

In reading what follows, it is important to keep in mind that the ideas captured here represent the beginning, not the end, of a research program. For this reason, it is perhaps worth stating what the following architecture – the semantic pointer architecture (SPA) – is *not*. First, it is not testable by a single, or small set, of experiments. Being the foundations of a research program, the SPA gains some amount of credibility when it gives rise to successful models. Failures of such models lead to a reconsideration of those specific models and, when systematic, a reconsideration of the program itself.

Second, the SPA is not a completed theory of mental function: we have not yet actually built a fully cognitive brain (you will not be surprised to learn). In what follows I describe models of perception, action, and cognition. And, I describe these in a way that combining them is trivial. Nevertheless, there are many behaviors involving each of these aspects of a cognitive system that I will not discuss.

Third, even theoretically speaking, the coverage of the SPA is uneven. Some aspects of cognition are more directly addressed than others – the SPA is undeniably a work in progress.

These qualifications aside, there are still compelling reasons to pursue the SPA. First, while it is not a completed, unified theory of mental function, it *is* an attempt to move towards such a theory. Such attempts can be useful in both their successes and failures: either way, we learn about constructing a theory of this kind. As discussed earlier, some have suggested that such a theory does not exist (section 1.2). But the behavioral sciences are far too young to think we have done anything other than scratch the surface of possible cognitive theories.

A second reason to pursue the SPA is its close connection to biological considerations. A main goal of this work is to show how we can begin to take biological detail seriously, even when considering sophisticated, cognitive behavior. Even if it is obvious that we should use as much available empirical data as possible to constrain our theories in general, actually doing do so requires well-specified methods for doing so. In what follows, the application of the SPA provides several specific instances in which we can address questions, and draw on empirical data, in new and interesting ways (see e.g., sections 3.4, 4.6, 5.7, 6.2, etc.).

A third reason to pursue the SPA is its generality. I have attempted to demonstrate the generality of the approach by choosing a broad set of relevant examples which demonstrate, but do not exhaust, the principles at work. Of course, one book is not enough to adequately address even a small portion of cognitive behavior. In short, the intent is to provide a method and an architecture that opens the way for a much wider variety of work than can possibly be captured in a single book, or for that matter, done in a single lab.

I leave a detailed discussion regarding the consequences of the architecture for chapter 10. Nevertheless, keeping in mind the main motivations behind the SPA should help situate the following introduction to it.

3.1 The semantic pointer hypothesis

Underlying the semantic pointer architecture is the semantic pointer hypothesis. Its purpose is to bridge the gap between the neural engineering framework (NEF) – a theory that tells us how a wide variety of functions may be implemented in neural structures – and the domain of cognition – which is in need of ideas about how such neural structures give rise to complex behavior. In the next four chapters, I describe and demonstrate five central aspects of the architecture: semantics, syntax, control, memory and learning. In the subsequent chapter, I provide a high-level integration of these discussions in a characterization of the semantic pointer architecture (SPA) for biological cognition. In the remainder of the book I explicitly compare the SPA to past suggestions for cognitive architectures, and discuss important practical and conceptual differences.

Let me begin with a simple statement of the semantic pointer hypothesis:

> Higher-level cognitive functions in biological systems are made possible by semantic pointers. Semantic pointers are neural representa-

tions that carry partial semantic content and are composable into the complex representational structures necessary to support cognition.

There are two aspects of this hypothesis that must be expanded in detail. First, I must indicate how semantic information, even if partial, will be captured by the representations that we choose to identify as semantic pointers. Second, I must give a characterization of how to construct complex structures given those same representations. Given the characterization of neural representation in the previous chapter, it is natural to begin with the assumption that semantic pointers are vectors in a high-dimensional state space. Consequently, I will address these two aspects in that context. I will address the semantics (both perceptual and motor) of these representations in the present chapter. The construction of representational structures (i.e. syntax) is addressed in the next chapter.

There have been a wide variety of proposals regarding the role that semantics and syntax should play in our theories of cognition. Traditionally, a cognitive system was thought to be largely a symbol processing system that relied on syntax to respect the semantics of the symbols and combination of symbols found in the system. However it did not depend on the semantics of the symbols for determining how they were processed. Essentially, semantics came along for the ride.

It has been suggested for some time, that a high-dimensional vector space is a natural way to represent semantic relationships between representations in a cognitive system. It is difficult to visualize high-dimensional spaces, but we can get a sense of what this claim amounts to by thinking of concepts in a 3D state space, as shown in figure 3.1. I refer to such a visualization as a conceptual golf ball. The surface of the ball represents the conceptual space, and the dimples in the surface represent concepts. Concepts that are semantically similar should lie in close proximity. Of course, things will get crowded quickly in 3 dimensions. The important contribution of high-dimensionality is that the amount of surface area available to put concepts increases exponentially ???verify, as shown in figure 3.2.

An obvious drawback with such spaces is that it is not immediately clear how complex representational *structures* can be placed in such a space (since the space has no obvious structure to it): I return to this issue in the next chapter. However, even without worrying about representational structure *per se*, the semantics of even simple human concepts seems to be extraordinarily rich, extending beyond a simple mapping to a single, even very high-dimensional, vector space (Barsalou, 2009, 1999). The richness of human conceptual behavior suggests that it is un-

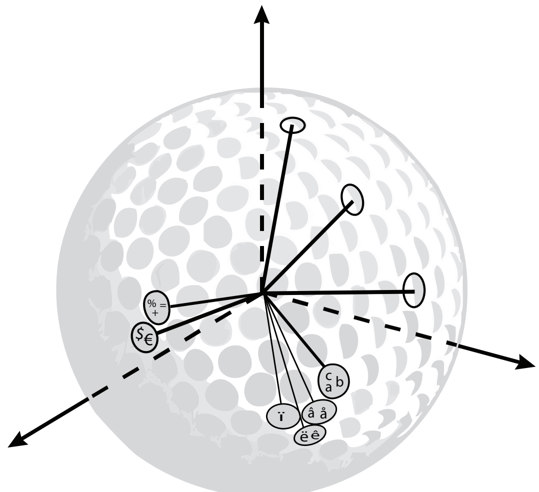

Figure 3.1: A conceptual golf ball depicting a 3D state space. Dimples in the ball represent concepts and 3D vectors from the center of the ball to its surface are used to track the dimples. The proximity of the dimples to each other reflects the similarity of the concepts they represent.

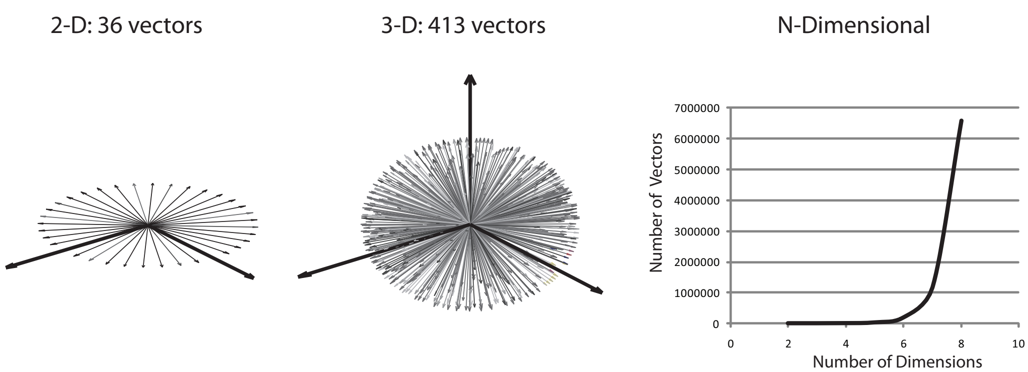

Figure 3.2: The scaling of the available conceptual space on the surface of a hypersphere with dimensionality. The figure shows the number of unit vectors that can be placed into a disk and ball under the constraint that there must be a minimum angle of ten degrees between any two vectors. The rightmost plot extends the result from intuitive two and three dimensional shapes to higher dimensional spheres.

likely that all of the aspects of a concept are actively represented at the same time. In earlier work, I have made a distinction between 'occurrent' and 'conceptual' representation (Eliasmith, 2000). Conceptual representations are representations that address the traditional questions about concepts that have preoccupied cognitive scientists and their predecessors for centuries. I believe there are theories of concepts consistent with what I say here (see section 10.3.3). However, my focus is on occurent representations. These are identifiable with the NEF-style representations described in the previous chapters: they are neural activities.

With this distinction in hand, acknowledging the richness of conceptual behavior leads to the realization that it is a mistake to assume that there are occurent representations in the brain that carry the full semantic content of any given concept. There is simply too much information related to any particular conceptual representation to be able to actively represent and manipulate it all. This is why the semantic pointer architecture employs 'pointers'.

A pointer, in computer science, is a set of numbers that indicate the address of a piece of information stored somewhere in memory. What is interesting about pointers, is that manipulations of pointers themselves can be performed which result in a use of the information identified by the pointer, despite the fact that that information itself is never explicitly addressed. Most such manipulations are quite simple. For instance, if I am creating a list of data that needs to be grouped, I can store each piece of data with a pointer to the next data item in the list. Such a 'linked list' provides a flexible method for traversing, removing, and adding items to a collection of data. Similarly, if I need to pass a data structure to a function which may use it, I can simply pass the pointer, which is typically much smaller than the data structure itself. As a result, much less information needs to be moved around within the system, while still making the relevant data available for subsequent use.

One notable feature of pointers in computer science, is that a pointer itself and the information contained at its address are arbitrarily related. As a result, having the pointer itself often indicates nothing about what sort of information will be found when the address to which it points is accessed. Since decisions about how to use information often depend on what the content of that information is, pointers are often 'dereferenced' during program execution. Dereferencing occurs when the data at the address specified by the pointer is accessed. In a computer, this is a relatively cheap operation because memory is highly structured, and the pointer is easy to interpret as an address. Given these features, pointers are reminiscent of symbols. Symbols, after all, are supposed to gain their computational utility from the arbitrary relationship they hold with their contents (Fodor, 1998).

And symbols are often thought to act like labels for more sophisticated data structures (such as schemas, scripts, etc.) just as pointers act as labels for whatever happens to be at their address.

The SPA suggests that neural representations, especially those implicated in cognitive processing, share central features with this traditional notion of a pointer. In short, the SPA suggests that the brain manipulates compact, address-like representations to take advantage of the significant efficiency and flexibility afforded by such representations. Relatedly, such neural representations may be able to act like symbols in the brain.

However, the arbitrary relationship between a pointer and its contents seems problematic for biological systems, because relationships between neural representations are most often learned. In contrast, the relationships between symbols in a digital computer are determined by human design decisions. It is precisely the failure of such decisions to adequately capture semantics that raises what has become known as the 'symbol grounding problem' (Harnad, 1990). This, in short, is the problem of defining how symbols get their semantics – a difficult problem indeed, and one over which much ink has been spilled. As a result, it would be too hasty to simply import the notion of a pointer directly from computer science into our understanding of biological cognition.

Consequently, the hypothesis I am suggesting here is an extension of the standard notion of a pointer. In particular, the 'semantic' in 'semantic pointers' refers to the fact that the representations that play the role of a pointer contain semantic information themselves. That is, the relationship between a semantic pointer and the memory to which it points is not arbitrary. In section 3.4, I provide a detailed example in the visual system of how semantic pointers relate to the memories to which they point. For now, it is useful to think of the semantic information that is contained in a semantic pointer as a compressed version of the information contained in a more conceptual memory. With this rough characterization in hand, we are in a position to consider recent research on semantics and examine detailed examples of how motor and perceptual semantic pointers can function in a neural architecture.

3.2 Semantics: An overview

When constructing a cognitive architecture, there has been an historical tendancy to start with syntactic considerations and build on them by introducing more and more sophisticated semantics. However, psychologists and linguists have more recently argued that much syntactic processing can be best explained by sophisticated semantic processing.[1] In contrast to traditional approaches, this put semantics in the driver's seat. As a result, much recent empirical work on semantics has focused on understanding what *kind* of semantic processing is needed to perform various cognitive tasks. In short, the question of interest has become: for which behaviors do we need 'deep' semantic processing, and which can be effectively accounted for by 'shallow' semantic processing?

The distinction between deep and shallow processing can be traced back to Allan Paivio's (1986; 1971) Dual-Coding Theory. This theory suggests that perceptual and verbal information are processed in distinct channels. In Paivio's theory, linguistic processing is done using a symbolic code, and perceptual processing is done using an analog code, which retains the perceptual features of a stimulus. Paivio (1986) provides a lengthy account of the many sources of empirical evidence in support of this theory, which has been influential in much of cognitive psychology including work in working memory, reading, and human computer interface design.

More recently, this theory has been slightly modified to include the observation that both channels are not always necessary for explaining human performance on certain tasks (Simmons et al., 2008; Glaser, 1992). Specifically, simple lexical decision tasks do not seem to engage the perceptual pathway. Two recent experiments help demonstrate this conclusion. First, Solomon and Barsalou (2004) behaviorally demonstrated that careful pairings of target words and properties can result in significant differences in response times to determining if a property belongs to a target word. For instance, when subjects were asked to determine if the second word in a pair was a property of the first word, false pairings that were lexically associated took longer to process. For example, a pair like 'cherry-card' resulted in 100ms quicker responses than a pair like 'banana-monkey'. Second, Kan et al. (2003) observed that fMRI activation in perceptual systems was only present in the difficult cases for such tasks. Together, this work suggests that deep processing is not needed when a simple word association strategy is sufficient to complete the task.

Nevertheless, much of the semantic processing we perform on a daily basis seems to be of the 'deep' type. Typical deep semantic processing occurs when we understand language in a way that would allow us to paraphrase its meaning, or answer probing questions about its content. It has been shown, for instance,

[1]Research in the field of 'cognitive linguistics' largely adopts this stance (George Lakoff, Ronald Langacker, Charles Fillmore,). ref??? johnson-laird mental models),

that when professional athletes, such as hockey players, read stories about their sport, the portions of their brain that are involved in generating the motor actions associated with that sport are often active (Barsalou, 2009). This suggests that deep semantic processing may engage a kind of 'simulation' of the circumstances described by the linguistic information. Similarly, when people are asked to think and reason about objects (such as a watermelon), they do not merely activate words that are associated with watermelons, but seem to implicitly activate representations that are typical of watermelon backgrounds, bring up emotional associations with watermelons, and activate tactile, auditory, and visual representations of watermelons (Barsalou, 2009).

Consider the Simmons et al. (2008) experiment which was aimed at demonstrating both the timing and relative functions of deep and shallow processing. In this experiment participants were each scanned in an fMRI machine twice. In one session, the experimenters were interested in determining the parts of the brain used during shallow semantic tasks, and during deep semantic tasks. As a result, participants were asked two questions: first, "For the following word, what other words come to mind immediately?"; and second, "For the following word, imagine a situation that contains what the word means and then describe it?" The experimenters found that in response to the first question, language areas (such as Broca's area) are most active. In contrast, in response to the second question participants engaged brain areas that are active during mental imagery, episodic memory, and situational context tasks (Kosslyn et al., 2000; Buckner and Wheeler, 2001; Barr, 2004). In other words, from this session it was evident that simple lexical association activated language areas, whereas complex meaning processing activated perceptual areas, as would be expected from the Dual-Coding Theory.

During the other scanning session, participants were asked to list, in their heads, answers to the question "what characteristics are typically true of X?", where X was randomly chosen from the same set of target words. When the two different scanning sessions were compared, the experimenters were able to deduce the timing of the activation of these two different areas. They found that the first half of the 'typically true of X' task was dominated by activation in language areas, whereas the second half of the task was dominated by activation in perceptual areas. Consistent with the earlier behavioural experiments, this work shows that shallow processing is much more rapid, so it is not surprising that highly statistically related properties are listed first. Deep processing takes longer, but provides for a richer characterization of the meaning of the concept.

This experiment, and many others emphasizing the importance and nature of deep semantic processing, have been carried out in Larry Barsalou's lab at Emory

University. For the last two decades, he has suggested that his notion of 'perceptual symbols' best characterizes the representational substrate of human cognition. He has suggested that the semantics of such symbols are captured by what he calls 'simulations'. Indeed, the notion of a 'simulation' has often been linked to ideas of deep semantic processing (e.g., Allport, 1985; Damasio, 1989; Pulvermüller, 1999; Martin, 2007). Consequently, they would no doubt agree with Barsalou's claim that deep semantic processing occurs when "the brain simulates the perceptual, motor, and mental states active during actual interactions with the word's referents" (Simmons et al., 2008, p. 107). Indeed, his data and arguments are compelling.

However, the important missing component of his theory is how such symbols and simulations can be implemented and manipulated by the brain. In a discussion of his work in 1999, one recurring critique was that his notion of 'perceptual symbols' is highly under-defined. For instance, Dennett & Viger pointedly note "If ever a theory cried out for a computational model, it is here" (1999, p. 613). More to the point, they conclude their discussion in the following manner:

> We want to stress, finally, that we think Barsalou offers some very promising sketchy ideas about how the new embodied cognition approach might begin to address the "classical" problems of propositions and concepts. In particular, he found some novel ways of exposing the tension between a neural structure's carrying specific information about the environment and its playing the sorts of functional roles that symbols play in a representational system. Resolving that tension in a working model, however, remains a job for another day.

Indeed, in the conclusion to a recent review of his own and others' work on the issue of semantic processing, Barsalou states (Barsalou, 2009, p. 1287):

> Perhaps the most pressing issue surrounding this area of work is the lack of well-specified computational accounts. Our understanding of simulators, simulations, situated conceptualizations and pattern completion inference would be much deeper if computational accounts specified the underlying mechanisms. Increasingly, grounding such accounts in neural mechanisms is obviously important.

This, of course, is the purpose of the semantic pointer architecture: to provide a neurally grounded account of the computational processes underwriting cognitive function.

Given the important distinction between deep and shallow processing, and the evidence that these are processed in different ways in the brain, a central question for the SPA is: how can we incorporate both deep and shallow processing? The central hypothesis of the SPA outlines the answer I will pursue in the next three sections: that semantic pointers carry partial semantic information. The crucial steps to take now are to: 1) describe exactly how the partial semantic information carried by semantic pointers is generated (and how they capture shallow semantics); and 2) describe how semantic pointers can be used to access the deep semantics to which they are related (i.e., how to dereference the pointers).

3.3 Shallow semantics

In essence, the shallow semantics captured by a semantic pointer can be thought of as a kind of 'compressed' representation of complex relations that underly deep semantics. Compression comes in two forms: 'lossless' like the well known .zip methods used to compress computer files; and 'lossy' which loses some of the information in the object that was compressed. I take semantic pointers to be lossy compressions of the information they are generated from. To demonstrate the utility of lossy compression, and its relevance to cognition, let us consider a recently proposed class of lexical semantic representations. These representations have been developed by researchers who build algorthims to do automatic text processing. These same representations have been shown to capture may of the word-similarity effects that have been extensively studied by psychologists (Deerwester et al., 1990; Landauer and Dumais, 1997).

These representations are constructed by having a computer process a very large corpus of example texts. During this processing, the computer constructs what is called a term-document frequency matrix (see figure 3.3). The columns of this matrix index specific documents in the corpus. The rows in the matrix index words that appear in those documents. When the computer reads a particular document, it counts the number of times any word occurs in the document, and adds that value to the appropriate cell of the matrix. This way, if we look down the columns of the matrix we can determine how many times each word appears in a given document. If we look across the rows of the matrix, we can determine how many times each word appears in each document.

Practically speaking, there are a number of important subtleties to consider when constructing these kinds of representations to do actual textual processing. For instance, such matrices tend to get very large, since many standard corpora

	Moby Dick	Hamlet	Alice in Wonderland	Sense and Sensibility
boat	330	0	0	0
enemy	5	0	0	1
mad	36	17	14	0
manners	1	1	1	34
today	1	0	1	9
tomorrow	1	0	1	15

Figure 3.3: Term-document frequency matrix. Each row of the matrix shows the number of occurrences of a specific word, written in the first column, in the four selected texts specified by the other columns. An extremely crude classification system based on this matrix might categorize Moby Dick, Hamlet, and Alice in Wonderland together based on their themes of madness, while distinguishing Moby Dick based on its abnormally frequent references to boats. The row vectors of the words 'today' and 'tomorrow' are also noteworthy for being quite similar, suggesting some level of semantic similarity between the words.

have over 20,000 documents and 100,000 unique words (Fishbein, 2008). Consequently, methods for reducing the size of the matrix are often employed. These methods are chosen to ensure that the most important statistical relationships captured by the original matrix are emphasized as much as possible. For present purposes, we can simply consider the raw matrix shown in figure 3.3.

The reason such a matrix can capture semantic information, is because we expect semantically similar words to occur in the same documents. This is because most of the considered documents are short, and on one or a few basic themes – like the books and stories encountered in early childhood. So, if we compare the row-vectors of semantically similar words, we expect them to have similar vectors, because those words will appear in many of the same contexts. Semantically unrelated words will have vectors that are not very similar. Notice that like semantic pointers, these representations of words are first and foremost high-dimensional vectors.

Importantly, with this matrix in hand, we are in a position to construct a representation of a document that has never been seen before. The simplest way to do this is to add up all of the row-vectors of the words that appear in the document, each time they appear. This document representation can then be compared to other documents we have seen before, and grouped with those that have the most similar document vectors. While simple, this method is surprisingly effective at categorizing documents. Using a wide variety of text corpora, well above 90% of new documents can be correctly classified with this representation (Yang and Liu, 1999).

More importantly, this same representation has been used by researchers to capture psychological properties of language. For example, Paaßet al. (2004) demonstrate how prototypes[2] can be extracted using such a method. The resulting representations capture standard typicality effects.[3] As well, this kind of representation has been used to write the Test of English as a Foreign Language (TOEFL). The computer program employing these representations scored 64.4%, which compares favorably to foreign applicants to American universities, who scored 64.5% on average (Landauer and Dumais, 1997).

In sum, semantic representations which capture basic statistics of word use can effectively support certain kinds of linguistic processing observed in human subjects. As mentioned earlier, these representations are high-dimensional vectors, just like semantic pointers. That is, they capture precisely kind of semantics that semantic pointers, in themselves, are proposed to capture in the SPA. In short, the SPA suggests that shallow semantic processing can be performed without using the pointer as a pointer: i.e., by relying solely on the semantic content captured by the pointer representation itself. There are many other methods for generating the semantics of high-dimensional vectors. Some have been shown to help capture many developmental, old-age related, and reasoning phenomena (Rogers and McClelland, 2004).

However, consideration of the kinds of experiments discussed in the last section suggests that there is an important and distinct role for deep semantic processing. This role is clearly not captured by the kinds of simple lexical associations used to generate these shallow representations for automatic text categorization. And, these shallow semantics are, in general, not sufficient to address the symbol grounding problem identified in section 3.1. After all, the word representations generated in this manner bear no relation to the actual objects that they pick out: instead, they model the statistics of the text. To address both deep semantics and

[2]Prototypes of categories are often used to explain the nature of concepts (Smith, 1989). It has been a matter of some debate how such prototypes can be generated.

[3]Typicality effects are used to explain why subjects rate some concept instances as being more typical than others. These effects are often explained by the number of typical features that such instances have (the more typical features an instance has, the more typical it will be). Typical instances are both categorized more quickly and produced more readily by subjects. The prototype theory of concepts has been successful at capturing many of these effects.

symbol grounding, in the next section I turn to a very different method of generating semantic pointers, one connected more directly to biological mechanisms.

3.4 Deep semantics for perception

It should be clear from the preceding discussion that there are two questions of interest when considering the semantic pointer architecture. First, how are the shallow semantics of the pointers themselves generated? Second, how can the pointers be used to engage the deep semantic processing system when appropriate? The discussion in the previous section addresses only the first question. This is useful for demonstrating the relevance of shallow semantics to characterizing central aspects of language processing, a task that is undeniable cognitive.

In this section, I will pursue another method for generating shallow semantics that demonstrates how the resulting representation remains linked to deep semantics in a neural architecture. However, the task I will consider is more perceptual. In section 4.7, I consider how these different aspects of semantic processing can be integrated.

Let us begin with object recognition in vision. Many of the most impressive results in machine vision employ *statistical modeling* methods. It is important to note that the word ‘model’ in statistics – and in the next few paragraphs – is not used in the same way as it is throughout most of this book, and in most of cognitive science. In statistics, the term ‘model’ refers to an equation that captures relationships: there is no expectation that the elements of the equation pick out objects in the world. In contrast, ‘model’ in non-statistical usages typically refers to abstractions whose parts are expected to map onto objects in the world. The neural models I describe throughout take their abstract parts (neurons, brain areas, etc.) to map onto real parts of the brain. These latter models are sometimes called ‘mechanistic’ models, to distinguish them from statistical models.

In general, statistical modeling methods are centrally concerned with characterizing the (unreliably) measured state of the world, and identifying important patterns in those often noisy, measured states. In short, these methods have been developed to describe complex relationships given real-world data. This may sound familiar: the lexical representations described in the previous section are a kind of statistical model, attempting to describe the relationships between words in real-world text data (and there are many spelling errors, non-words, etc.). Considered generally, describing complex real-world relationships with uncertain information is exactly the problem faced by biological systems.

For objection recognition, we can begin to formalize this problem by supposing there is some visual data, y, which is generated by the external visual world and drives neural activity. If we suppose that the purpose of perceptual systems is to construct and use a statistical model of this data, the system must figure out some function $p(y)$ that describes how likely each state y is, so it can use that information to disambiguate future data. For instance, if I am in North America and a medium-sized brown animal is coming in my direction, I can assign probabilities to the various kinds of animal it might be (e.g. groundhog, dog, and rabbit are high, capybara and wallaby are low). Assigning those probabilities is an example of using the model, $p(y)$, that I have constructed based on past data.

Since the real world is extremely complex, the ideal statistical model will also be enormously complex (as it is the probability of all possible data at all times). As a result, the brain probably approximates 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. For example, if all of the data y lie in the famous Bell curve (or Gaussian distribution), we can model the data with an equation like:

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

Then, to 'capture' all past data using our model, we only need to remember two parameters, $\bar{y}$ (the mean) and σ (the standard deviation), and the equation describing their relationship. This is much more efficient than remembering each value of $p(y)$ for each value of y explicitly.

To build such a model, the system needs to estimate the parameters. Of course, to do any such estimating, the system needs data. As a result, a kind of bootstrapping process is necessary to construct this kind of model: we must use data to estimate the parameters; then we use our best estimate of the parameters to interpret any new data. Despite this seeming circularity, extremely powerful and general algorithms have been designed for estimating exactly these kinds of models (Dempster et al., 1977). Such models have also been extensively employed in building connectionist-type models, and have been suggested to map to biological neural networks.[4]

Note, however, that the methods for model inference do not specify the structure of the model itself (i.e. the relationships between the parameters). In the

[4] A good place to start for state-of-the-art applications of these methods is Geoff Hinton's web page at `http://www.cs.toronto.edu/~hinton/`. For discussion of biological mappings of these methods see Friston (2003).

artificial neural network application of these methods, this structure is often 'biologically inspired'. One notable feature of brain structure that has proven a very useful starting point is its hierarchical nature. The best known example of this structure in neuroscience is the visual hierarchy. For object recognition, this hierarchy begins with the retina, and proceeds through thalamus to visual areas V1 (primary visual cortex), V2 (secondary visual cortex), V4, and IT (inferotemporal cortex) (Felleman and Essen, 1991).

In a hierarchical model, each higher level in the hierarchy attempts to build a statistical model of the level below it. Taken together, the levels define a model of the original input data (see figure 3.4). This kind of hierarchical structure naturally allows the progressive generation of more complex features at higher levels, and progressively captures higher-order correlations in the data. Furthermore, these kinds of model lead to relations between hierarchical levels that are reminiscent of the variety of neural connectivity observed in cortex: feedforward, feedback, and recurrent (interlayer) connections are all essential.

The power of these methods for generating effective statistical models is impressive (Beal, 1998). They have been applied to solve a number of standard pattern recognition problems, improving on other state-of-the-art methods (Hinton and Salakhutdinov, 2006). Furthermore, they have been shown to generate neuron tuning curves that look like those seen in visual cortex (Lee et al., 2007), when constructing models of natural images. In fact, many of the most actively researched models of vision are naturally interpreted as constructing exactly these kinds of statistical models.

To get a clearer picture of what this approach to perceptual modeling offers, and how it can be used to generate semantic pointers, let us turn to an example of such a model that was built in my lab by Charlie Tang. The purpose of this system is to construct representations which support recognition of a wide variety of handwritten digits presented as visual input. The input is taken from the commonly used MNIST database. Examples of the input are shown in figure 3.5. The model is structured as shown in figure 3.4, and is shown 60,000 examples out of this data set, and told how those examples should be categorized. Based on this experience, the model tunes its parameters to be able to deal with another 10,000 unseen, though similar, visual inputs in the data set.

To maintain biological relevance, the first layer of the model is trained on natural images, in order to construct an input representation that looks like that found in primary visual cortex. As shown in figure 3.6, the tuning curves capture many of the properties of V1 tuning, including a variety of spatial frequencies (i.e. narrowness of the banding), positions, orientations of the bands, and the edge-

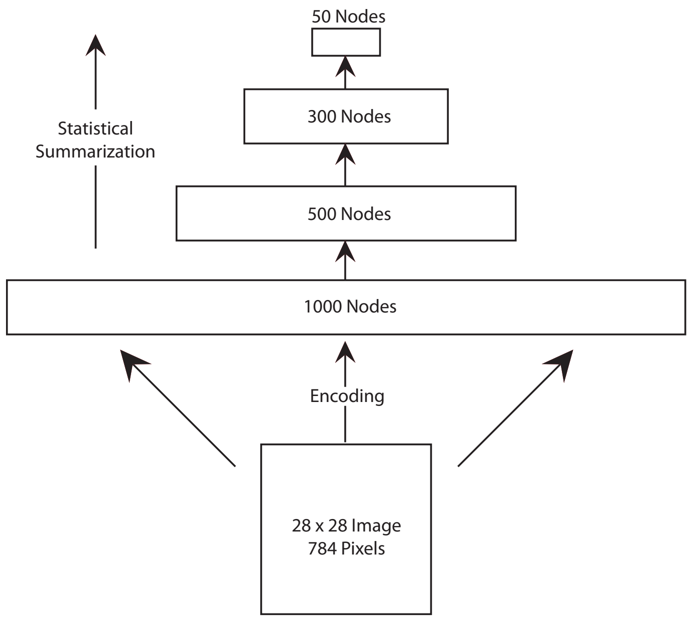

Figure 3.4: A hierarchical statistical model. An original image consisting of 784 pixels is compressed into a 50-dimensional compressed representation through a hierarchical series of statistical summaries. Note that the number of nodes at each level is given in terms of state space; that's the number of dimensions, not neurons!

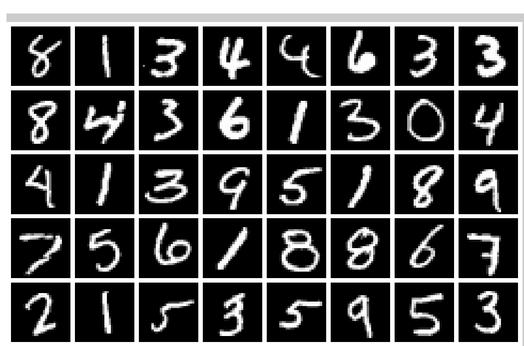

Figure 3.5: Input images from the NMIST database.

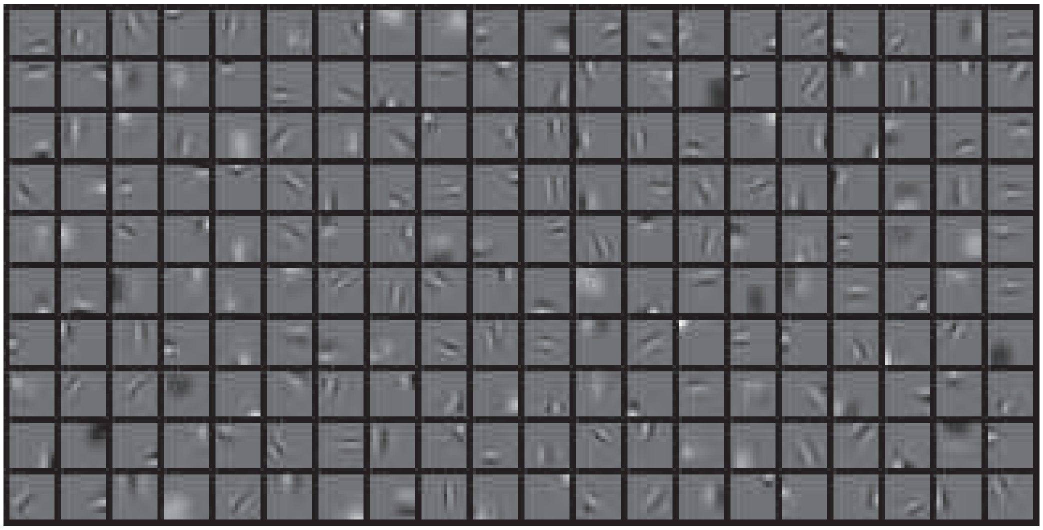

Figure 3.6: Tuning curves of neurons in the model. These learned representations look much like the tuning curves found in primary visual cortex (V1).

detector-like shape.[5] These tuning curves are used to represent the input digits, and then the remaining layers of the network are trained to capture the statistics of that input. By the time we get to the highest level of the hierarchy, we have a much smaller (i.e., compressed) representation summarizing what has been presented to the retina. This compressed representation is a semantic pointer.

Notably, the highest layer of this network has 50 nodes, which, because these are not neurons, means that the state space is 50-dimensional. It is this 50D space that contains the semantic pointers whose contents tell us about the presented digits. Clearly, this representation does not contain all of the information available in early visual areas. Instead, it is a summary that usefully supports this object recognition task. This network, like most in machine vision, can score less than 2% error on the 10,000 test digits which have not been used to train the network (i.e., they classify about 200 wrong). In fact, these models outperform humans on this and other similar tasks (Chaaban and Scheessele, 2007).

So, these compressed representations (i.e., semantic pointers), like the lexical ones discussed previously, can capture important information that can be used to classify the input. In both cases, it is shallow comparisons between the semantic

[5]Training such networks on natural images has often been shown to result in V1-like tuning (Olshausen and Field, 1996).

pointers that result in the classificiation. So, in both cases, the semantic pointers themselves carry shallow semantic information. However, there are two important differences between these semantic pointers and the lexical ones. First, these were generated based on raw images. This, of course, is a more biologically relevant input stream than text: we do not have organs for directly detecting text. Consequently, the 50D pointers are grounded in a visual input. If we have a way of treating these pointers like symbols, then we have found a natural solution to the symbol grounding problem.

Second, and more importantly, these 50D pointers can be used to drive deep semantics. That is, we can, in a sense, run the model 'backwards' to decode, or unpack, the meaning of a 50D representation. In other words, we can clamp the semantic pointer representation at the top level of the network and then generate an input image at the lowest level.[6] Several examples of this process are shown in figure 3.7.

This figure demonstrates that a lot of the detail of an input is in fact captured by the semantic pointer representation. Subtleties of the way particular letters are drawn, such as whether an '8' is slanted or not, can be reconstructed from the pointer that such a figure generates. It is these subtleties that caputre the deep semantics of this representation. However, it is obviously not always the case that we have a precisely drawn '8' in mind when we talk about the number '8'. That is, we might want to have access to the deep visual semantics of a pointer, when it is generated by an auditory input. In such a case, we can still use a semantic pointer, and the natural choice is the 'average' of the pointers associated with a category (see figure 3.7c). This can be thought of as a prototype of the category.[7] If other instances of the category need to be generated, small random movements around this prototype will result in a variety of examples.

Because semantic pointers can be unpacked to provide detailed perceptual information, it should be clear why I have chosen to call them 'pointers'. However, the dereferencing procedure depends on having the full perceptual hierarchy

[6]This does not suggest that the brain somehow recreates retinal images (there are no neurons that project from cortex to retina). Instead, figure 3.7 shows the retinal images that are consistent with the unpacked cortical representations. The deep semantic information at these non-retinal levels is accessible to the rest of cortex. In the brain, the unpacking would stop sooner, but could still be carried out to as low a level as necessary for the task at hand. This is one reason why seeing an image is not the same as imagining one, no matter how detailed the imagining.

[7]Note that the '2' prototype is a combination of twos with straight bottoms and twos with loopy bottoms. This may suggest simply taking the mean is a mistake, and that there are two subcategories here. Dealing with these interesting, but additional complexities is beyond the scope of the present discussion.

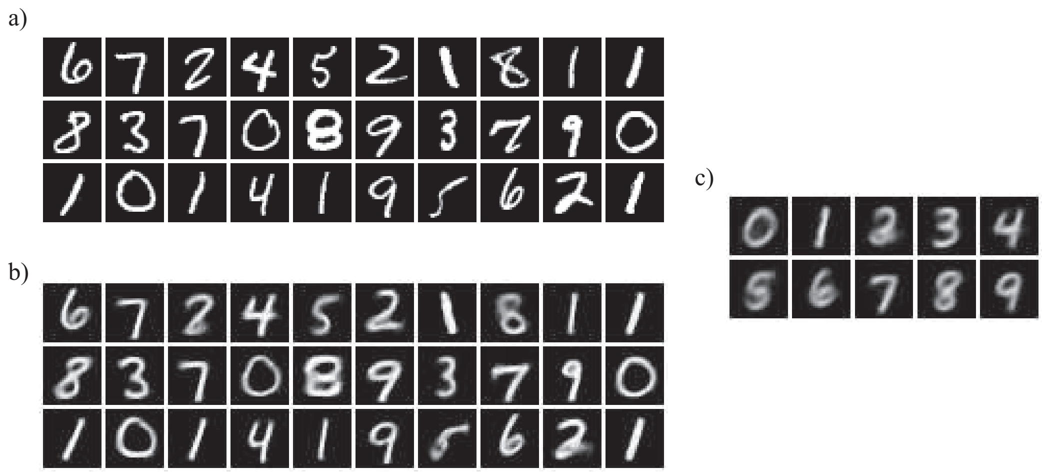

Figure 3.7: Unpacking semantic pointers. a) The original input of several randomly chosen images. These are compressed into a 50D representation at the highest level of the model. b) The resulting images from unpacking the semantic pointers. The same 50D representations generated by the network are clamped at the highest-level, and the rest of the network is used to generate a guess as to what input would produce that semantic pointer. c) The unpacked mean semantic pointer for each category.

available. This maps well onto the fact that during deep semantic processing, percepetual areas are active in human subjects, as discussed earlier. It is, of course, a top-down procedure that allows these deep, grounded semantics to be regenerated.

In addition, I have not said how the pointers themselves can be used in symbol-like structured representations (this is the purpose of the next chapter). But, if that story is plausible, then the overall shape of the SPA should be clear. Semantic pointers, generated by grounded perceptual processing, can be 'stripped-off' of that processing and used to both carry shallow semantics and be treated like a symbol. If deep semantics are needed, the semantic pointer can be used to clamp the top layer of the perceptual network that gave rise to it, and the network can re-generate the deep semantics.

In the terminology of the NEF, this story is one at the level of the state space. That is, these pointers and processing are defined in vector spaces that are represented by neurons. We can apply the NEF principles to construct such a model in spiking neurons, to ensure that it will function on a biological substrate, and to allow more direct comparison of our model with neuroscientific data.

- put in spiking version of the model here???, look at some sample tuning curves (compare to V1 tuning curve from before)
- show the shallow/deep distinction by comparing the above fig, with one that plots their relative location in 2D space... (clustering that gets shallow semantics)???
- this model, like cortex has fewer neurons active at higher layers... (get citations from bruce), fewer in IT than earlier for same repn.
- other bio considerations consistent?

While models like the ones presented here present specific examples of how semantic pointers may be generated, it is important to keep in mind the limitations of such models. For instance, there is probably much more of relevance regarding biological structure that should inform how we construct our statistical models that has not yet been taken into account. While these models mimic the basic hierarchical structure of perceptual systems, and can approximately be mapped onto observed anatomical connections, most such models do not have connections that skip layers of the hierarchy, as observed in cortex. In addition, many other processes important for recognition, such as attention, tend to be excluded from current state-of-the-art models.[8] Finally, there remain many questions regarding how all aspects of such models could be implemented by biological networks: Can spiking networks compute the integrals necessary to *learn* these models? Can the learning of these models be done by biologically plausible learning rules? Does each neuron in the brain represent a different variable in the statistical model? The NEF model presented above does not address learning (although see section ???). Nevertheless, the crucial features of such models that I am suggesting are relevant for understanding deep semantics are: 1) the construction of representations of the perceptual input that capture important statistical relationships; and 2) that such representations are 'compressed' (i.e., lower dimensional) representations of the input.

3.5 Deep semantics for action

In fact, we can find these same features in biological representations used to drive motor states. Of course, the task for motor systems seems much different than that for perceptual systems. The motor system does not need to classify presented stimuli, but rather to direct a highly nonlinear system towards a desired state. So, perceptual systems need to go from a high-dimensional, ambiguous state (e.g., images generated by highly nonlinear environmental processes) to a much smaller set of states (e.g., object categories). Motor systems need to go from a small set of states (e.g., desired pointing targets) to a high-dimensional, ambiguous state (e.g., any one of the possible configurations of muscle tensions over the body that results in pointing to a target). These tasks seem to be almost exact *opposites*.

However, there is a lot of shared by these opposites: both need to map low- to high-dimensional states; both need to deal with complexity and nonlinearity in doing so; both need to deal with uncertainty and ambiguity in doing so; and both need to share information between past and future in doing so. In mathematical terms, problems which share their structure in this kind of way are called 'dual problems'. It is useful to identify duality between problems because if one kind of problem can be solved, then so can the other.

As a simple example, consider the relationship that exists between a cube and an octahedron (see figure 3.8). Notice that there are the same number of sides in a cube as vertices in an octahedron and vice versa. As well, both have twelve edges, connecting the relevant vertices/faces in a structurally analogous manner. These

[8] Though we and others have been considering the inclusion of attention-like processes (Tang and Eliasmith, 2010).

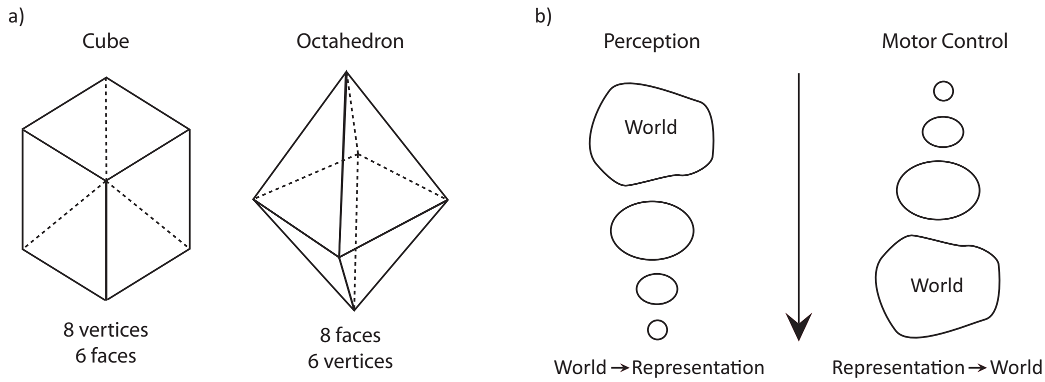

Figure 3.8: Dual problems. a) The cube and the octahedron provide a simple example. Both figures share a similar geometric framework, but the number of vertices and faces of the two shapes are reversed. b) The perceptual and motor systems are much more complicated, but can also be treated as duals. The perceptual system encodes high-dimensional, non-linear information using a hierarchical framework to establish a compressed representation with a lower dimensionality than the original information. The motor control system reverses this to determine high-dimensional control signals from a compressed, low-dimensional signal.

two solids are thus duals. If we pose a problem for one of these solids – What is the volume of the smallest sphere that intersects all vertices (faces)? – then a solution for one of the solids provides a solution for the other. This is true as long as we swap the relevant structural elements (i.e., faces and vertices).

Why does this matter for understanding perception and motor control? Because precisely this dual relationship has been shown to exist between statistical models of perceptual processes and optimal control models of motor processes Todorov (2007, 2009). Figure 3.8 suggests a mapping between perceptual and motor systems that takes advantage of this duality. From an architecture point of view, this duality is very useful because it means that there is nothing different in kind about perceptual and motor systems. From the perspective the SPA in particular, this means that semantic pointers can play the same role in both perception and action. Much of the remaining discussion in this section describes why and how these mappings can be made.

To begin, like the perceptual system, the motor system is commonly taken to be hierarchical (see figure 3.9). Typically, we think of information as flowing down rather than up the motor hierarchy. For instance, suppose you would like to move your hand towards a given target object. Once the desired goal state has been generated, it is provided to the cortical motor system. The first level of this system may then determine which direction the arm must move in order to reach that goal state.[9] Once the direction of movement has been determined, it can be used as a control signal for a controller that is lower in the hierarchy. Of course, specification of the direction of movement does not determine how torques need to be applied at the joints of the arm in order to realize that movement. This, then, would be the function of the next lower level in the control hierarchy. Once the specific forces needed to move the arm are determined, the specific tensions that need to be produced in the muscles in order to realize those forces must be generated by the activity of motor neurons. In short, as the motor command becomes more and more specific to the particular circumstance (e.g., including the particular part of the body that is being moved, the current orientation of the body, the medium through which the body is moving, etc.), lower levels of controllers are recruited to determine the appropriate control signal for the next lower level. Ultimately, this results in activity in motor neurons that cause muscles to contract and our bodies move.

Notably, just as it is inaccurate to think of information as flowing only 'up' the perceptual hierarchy, so is it a mistake to think of this picture of motor control as being in one direction. As well, in both systems, there are many connections that skip hierarchical levels, so the flow of information is extremely complex. Nevertheless, the main problems to be tackled are very similar. So, just as we can begin our characterization of perception as thinking of higher levels in the perceptual hierarchy as constructing models of the levels below them, so we can think of higher levels in the motor hierarchy as having models of the lower levels. Higher levels can then use such models to determine an appropriate control signal to affect the behavior of that lower level.

There is good evidence for this kind of control structure in the brain Wolpert and Kawato (1998); Kawato (1995); Oztop et al. (2006). That is, a control structure in which higher levels of the hierarchy have explicit models of the behavior of lower levels. When we attempt to specify these models, a simplification in our characterization of perception becomes problematic: we assumed that the models were static. In motor control, there is no such thing as non-temporal processing. Time is unavoidable. Embracing this observation actually renders the connection between perceptual and motor hierarchies even deeper: higher levels in both

[9] This description is highly simplified, and proposes representations and movement decompositions that probably do not occur in the motor system. Identifying more realistic representations would require a much lengthier discussion, but would not add to the main point of this section.

hierarchies must model the statistics *and* the dynamics of the levels below them.

In the previous perceptual model, the statistics at higher-levels capture the regularities at lower levels. In the neural model, these were evident as neural tuning curves that were used to represent the perceptual space. In motor control, these same kinds of regularities are often called 'synergies' Bernstein (1967); Lee (1984). Synergies are useful organizations of sets of movements that are often ellicited together. As before, we would expect the tuning curves of neural models to reflect these synergies for the representation of the space of motor actions. These representations need to be learned based on the statistics of the space they are representing, as in the perceptual model above.

- Evidence/arg that higher levels are lower-dimensional

What is not addressed by that simple perceptual model is dynamics.[10] The particular dynamics of the system are likely to affect which synergies are most useful for controlling action. So, the representational and dynamical aspects of the system are tightly coupled. In fact, it is natural to describe this connection in a way captured by the third principle of the NEF: the dynamical models are defined over the representational state space. That is, the dynamical models of a given level of the hierarchy are defined using the synergies of that level.[11] The resulting dynamics then drive lower levels, and feedback from lower levels can inform these higher-level models. So, despite the added complexity of dynamics, the general structure of the motor system is hierarchical.

Consequently, like the simpler perceptual model, there are two main features of the motor system: those (dynamical) representations capture important statistical relationships; and higher-level representations are compressed. I suspect, but will not consider the details here, that the dynamical motor hierarchy is a better model for perceptual processing (including object recognition) than the kinds of static models considered above. Nevertheless, that main conceptual points rele-

[10]For effective control, dynamical models are typically broken into two components: a forward model and an inverse model. A forward model is one which predicts the next state of the system given the current state and the control signal. An inverse model performs the opposite task, providing a control signal that can move between two given states of the system. For simplicity, I discuss both using the term 'dynamical model', as this level of detail is beyond the current scope.

[11]There is some issue here about drawing boundaries to determine the input and output of these models. Essentially, a control (error) signal can be thought of as generated at a given hierarchical level and then mapped to a lower (higher) level, or the mapping and generation may be considered concurrent in which case the dynamical model maps synergies at different levels. It is possible that both happen in the nervous system, but I believe the former is easier to conceptualize.

vant to the SPA remain. Let us now consider in more detail, how the SPA relates to motor control.

While working in my lab, Travis Dewolf (2010) recently proposed a framework that integrates known neuroanatomy with a hierarchical control characterization of the function of the motor system. This framework is called the Neural Optimal Control Hierarchy (NOCH), and a simplified version is shown in figure 3.9, which has been tailored to arm control. Models based on NOCH are able to explain the effects of various motor system perturbations including Huntington's disease, Parkinson's disease, and cerebellar damage. The task I consider here is reaching in a plane towards a target (see figure ???).

Figure ??? demonstrates the functioning of the model.

- ???emphasize that he goes from lower to higher dimensional and uses multiple hierarchical layers
- just travis'... downloadable nengo model? uses non-learned synergies

As can be seen from figure 3.9, the architecture is not a pure hierarchy as in the perceptual case. Nevertheless, low-dimensional, high-level representations in premotor cortex (PM) can be used to drive the high-dimensional, low-level spinal cord to affect movement. While simple, this motor control example can be used to give a sense of how semantic pointers capture the distinction between deep and shallow semantic processing in the motor system. As in the perceptual case, the low-dimensional space can be 'dereferenced' by the remainder of the system into a movement. In figure 3.11 similar high-level representations are shown to result in similar movements, suggesting that they define a semantic space. Indeed, the work of Georgopoulos can be seen as an attempt to map this space (see section 2.5, especially figure 2.11).

In cases that demand deep semantic processing (e.g., that require estimating the precise configuration of joints, etc.) those high-level semantic pointers can be used to 'run' the models in the motor system in order to internally generate more information about the nature of reaching movements. Recall that in the previous section we identified two important features of semantic pointers: they capture higher-order relationships; and they are compressed representations. It should now be clear how these features are realized by the semantic pointers generated by the motor system. First, the semantic pointers capture higher-order relationships between states of the body because they can be 'dereferenced' in order to coordinate those states for successful motor action. And second, the representations at the highest level of the motor hierarchy are lower dimensional than those

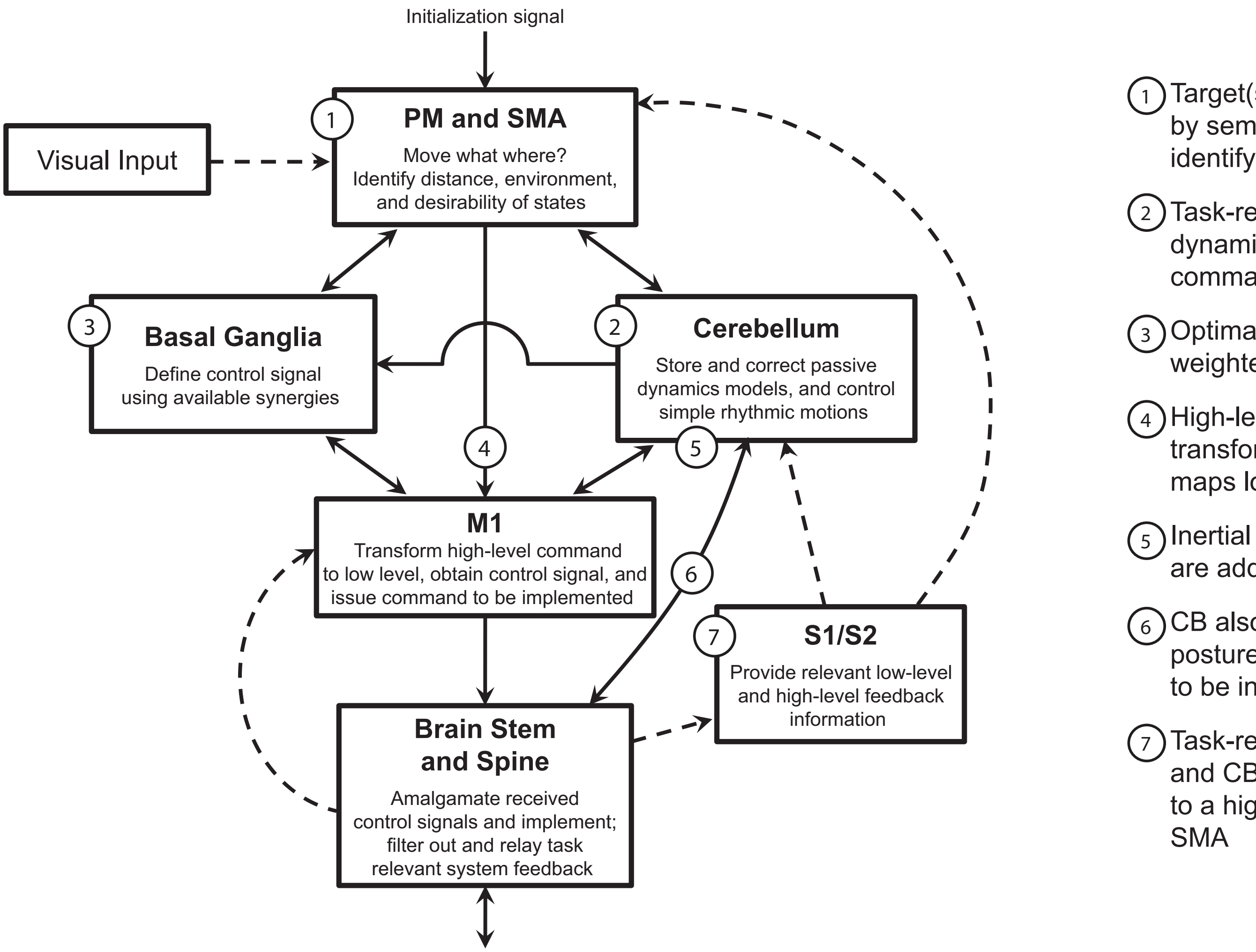

Figure 3.9: Neural optimal control hierarchy (NOCH). This framework maps neuroantomy onto optimal control in a manner which allows the controller to interact with semantic pointers. Abbreviations: PM - premotor area; SMA - supplementary motor area; BG - basal ganglia; M1 - primary motor area; CB - cerebellum; S1 - primary somatosensory area; S2 - secondary somatosensory area. This diagram is a simplified version of NOCH presented in Dewolf (2010).

Figure 3.10: A simple reaching model. ...???show several reaches, and the high and low level control signals? (to emphasize high and low dimensionality)

Figure 3.11: A semantic motor space. This figure shows how similar high-level semantic pointers can generate similar low-level movements.

at the lowest level. As a result, they are usefully considered as compressed representations.

This simple example suggests that semantic pointers can be found playing similar roles in both the perceptual and motor systems. There are, of course, important differences between the perceptual and motor models I have presented above. Most obviously, the perceptual model does not include dynamics, and the motor model does not learn the lower-level representations (i.e., synergies). There is ongoing research that attempts to address both of these challenges,[12] including work in my own lab. Given that the theoretical characterization of the problem is clear, however, it seems reasonable to suggest that subsequent developments will preserve the role of semantic pointers in the preceding characterization.

3.6 Meaningful conclusions

My discussion of the SPA in the context of perception and action has adopted the usual approach of considering these aspects of behaviour somewhat independently. However, it is hopefully clear from the above discussion that action and perception are not only tightly conceptually linked, but also mutually serve to capture deep semantics in the SPA. The parallel stories provided above for action and perception are intended to lay the groundwork for more cognitive consideration of semantics. I leave that task until the end of the next chapter (section 4.7), after considering how semantic pointers can be used in a symbol-like manner. First, I would like to consider the relationship between action and perception in the SPA.

Notice that both characterizations depend on the hierarchy being used in both directions. In the forward direction (up the hierarchy), the perceptual hierarchy allows classification of visual stimuli, in the reverse direction it allows the generation of deep semantics. For the motor hiearchy, the forward direction (down the hierarchy) allows for the control of a sophisticated body using simple commands. The reverse direction allows for the generation of synergies that allow the simple commands to be effective. So it is clear that in both cases, the semantic pointers

[12]For instance, Rod Grupen, Rolf Pfiefer, Dana Kulic, and other roboticists have been working on these issues from a motor control and reinforcement learning perspective. Geoff Hinton, Yann LeCun, Karl Friston and others have been adopting a more statistical modelling oriented approach.

operating at the top of the hierarchy are always 'pointing to a memory', and can be used in the hierarchy to allow the memory to be ellicited. Of course, this use of the term 'memory' is, unlike a in a computer, necessarily constructive. We can make movements we have never made before, and we can imagine handwriting we have never seen before. This observation supports the notion that perceptual and motor systems are well-characterized as constructing statistical models of their past.

In the above examples, I have discussed a number of different ways of constructing these models – of producing semantic pointers. It is worth reiterating that these suggestions are not an essential part of the SPA itself. What is central to the architecture is that there are low-dimensional summaries of these high-dimensional statistical models generated by the system. This central feature also highlights another defeasible assumption of the above examples: that the perceptual and motor models are generated independently.

As has been argued forcefully in much of the recent literature[13] it is a mistake to think of biological systems as processing perceptual information and then processing motor information. Rather, both are processed concurrently, and inform one another 'all the way up' the hierarchies. Consequently, it is more appropriate to think of the semantics of items at the top of both hierarchies as having concurrent perceptual and motor semantics.

- (for this to be true and included here, need an integrated example later that does this – see GUM) In fact, if the motor and perceptual models are appropriate constructed, the same semantic pointer can be used to drive both the motor and percpetual systems. Thus, the accessed semantics of the pointer depends on the model that s used to dereference it.

This is a natural, but precise, way to specify the kinds of interaction between perception and action that have often been argued for. It may remind some of the notion of an 'affordance', introduced by psychologist James Gibson (1977). Affordances are action possibilities that are determined by the relationships between an organism and its environment. Gibson suggested that affordances are automatically picked up by animals in their natural environments, and provide a better characterization of perception (as linked to action), than traditional views.

Several robotics researchers have embraced these ideas, and found them useful starting points for building interactive perception/action systems (Scheier and

[13] I can by no means give comprehensive coverage of the many researchers who have made these sorts of arguments, which can be found in robotics, philosophy, neuroscience, and psycholgoy. Some useful starting points include Brooks (1991); Churchland et al. (1994); Port and van Gelder (1995); Regan and Noë (2001).

Pfeifer, 1995; Ballard, 1991). Similarly, the notion, while imprecise, seems to relate to the information captured by a semantic pointer when embedded in the SPA. Notably, these pointers can be used as direct links between action and perception that depend on the motor and perceptual experiences, of the organism. Consequently, their semantics is directly tied to both the environment and body in which they are generated. There are, undoubtedly, many differences as well (e.g., semantic pointers do not seem to be 'directly perceived' in the sense championed by Gibson for affordances). Nevertheless, the similarities may help relate semantic pointers to concepts familiar in psychology.

As well, affordances highlight the important interplay between perception and action. Motor control, after all, is only as good as the information it gathers from the environment. As a result, it is somewhat awkward to attempt to describe a motor controller without discussing perceptual input. It is much more appropriate to conceive of the entire perception/action system as being a series of nested controllers, rather than a feed-in and feed-out hierarchy. As depicted in figure 3.12, nested controllers can be generated by inter-leaving the hierarchies of the kind described above.[14] What should be immediately evident from this new structure, is that the process of perceiving and controlling becomes much more dynamic, interacting at many levels of what was previously conceived of as a hierarchy. This, of course, makes the system much more complicated, which is why it was more convenient to describe it as two hierarchies. And, indeed, the real system is more complicated still, with connections that skip levels of the hierarchy, and multiple perceptual paths interacting with a single level of the controller hierarchy. The question of relevance is: does identifying this more sophisticated structure change our story about semantic pointers?

Given the suggested genesis and use of semantic pointers, I believe the answer is no. Recall that the way the highest level representations in the motor and perceptual hierarchies were generated, was by characterizing statistical models that identify the relationships between the data of interest. Whether these models are influenced solely by perceptual or motor processes, or whether they are influenced by perceptual and motor processes, may change the nature of those relationships, but will not change the effective methods for characterizing those relationships. It will, then, still be the case that dereferencing a perceptual representation for deep semantic processing results in identifying finer perceptual details not available in

[14]The observation that perceptual and motor cortex are both hierarchically structured and mutually interacting is hardly a new new one (Fuster, 2000), what is new, I believe, is the computational specification, the biological implementation of the computations, and the integration into a cognitive hierarchy.

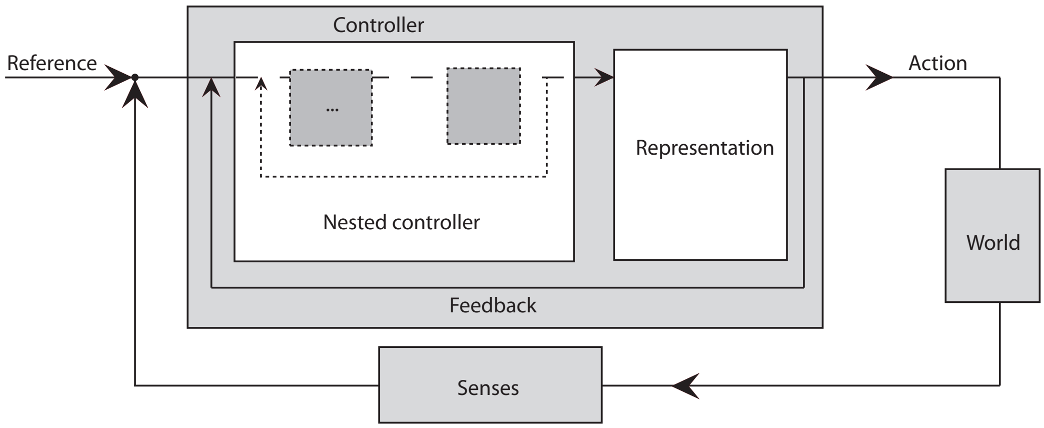

Figure 3.12: SPA as nested controllers. A controller is a component of a dynamic system that reacts to an input to produce a certain output which will influence the larger system. Feedback from the resulting change is then added to the input to the controller, allowing it to dynamically adjust its output. Representing the SPA as nested controllers implies that at every level of the semantic hierarchy there will be such a component influencing the behaviour of that level of the hierarchy based on the dynamics of the local representation of the system.

the higher-level (semantic pointer) representation. It may also turn out that the semantic information includes information about relevant motor activities. This, in fact, has been observed in the psychological data (Barsalou, 2009). So, semantic pointers will still be compressed and capture higher-order relationships, it just may be that their contents are neither strictly perceptual nor strictly motor. Instead, the perceptual/motor divide is one that is occasionally convenient for theorizing, but not built into the semantics of our conceptual system.

We should also be careful, however, not to overstate the closeness of the relationship between perception and action. Dissociation between impairment to visually guided motor control and object identification has been well-established (Goodale and Milner, 1992). Some patients, with damage to the dorsal visual pathways, can identify objects but not reach appropriately for them. Others, with damage to the ventral visual pathways, can reach appropriately, but not classify objects. This suggests two things: 1) that motor action and visual perception can come apart; and 2) that the object classification model presented earlier only models part of the visual system (the ventral stream), at best. Again, the relevant point here is not about the completeness of the models, but that the subtle, complex character of action and perception in biological systems is consistent with the assumptions of the SPA. Capturing the appropriate degree of integration and distinctness of action and perception is a task independent of generating semantic pointers, so long as there are high-level representations with their assumed properties.

So, semantic pointers are compressed representations of motor/perceptual information found within a nested control structure. The story I have told so far is intended to characterize semantics, both deep and shallow, as related to semantic pointers. What remains to be described in order to make semantic pointers cognitively relevant, is how they can actually be used to encode complex syntactic structures. That is, how are these perceptual/motor vectors used to generate the kinds of language-like representations that underlie high-level cognition? Answering this question is the purpose of the next chapter.

3.7 Nengo: Neural computations

The previous tutorial on neural representation is a useful first step to understanding biological cognition but, in order to result in interesting behavior, neural representations must be transformed in various ways. In fact, such transformations, especially when they are non-linear, often result in new and interesting representations.

In this tutorial, we examine how principle 2 from section 2.3.2 can be exploited in Nengo to compute transformations. This discussion builds on the previous tutorial on neural representation (section 2.5), so it may be helpful to review that material. The following discussion is broken into two sections, the first on linear transformations and the second on nonlinear transformations. The tutorial focuses on scalar transformations, but the methods generalize readily to any level of the representational hierarchy (see section 2.4), as is briefly discussed.

Linear transformations

As formulated, principle 2 makes it clear that transformation is an extension of representation. In particular, transformations rely on the same encoding, but exploit a different weighting during decoding than is used when representing a variable. In fact, there is one transformation that comes ‘for free’ – that is, as a basic property of single neurons – and that is addition.

- do addition with scalars

- do multiplication with scalars
- mention how to do arbitrariy linear transformations for sure (refer back to this in later sections... using the matrix 'M' for the transformation).

Nonlinear transformations

- can point ahead to discussion of dendritic nonlienarities in chp 4?
- start with nonlinear fcns of one scalar.
- ?show that multiplication works well with about 100 neurons in each population.
- goto nonlienar fucns of a vector, which allows mutliplicaiton of scalars, and note how these are actually the same thing.