Computational approaches to sensorimotor transformations

Alexandre Pouget[1] and Lawrence H. Snyder[2]

[1] Department of Brain and Cognitive Sciences, University of Rochester, Rochester, New York 14627, USA

[2] Department of Anatomy and Neurobiology, Washington University, 660 S. Euclid Avenue, St. Louis, Missouri 63110, USA

Correspondence should be addressed to A.P. (alex@bcs.rochester.edu)

Behaviors such as sensing an object and then moving your eyes or your hand toward it require that sensory information be used to help generate a motor command, a process known as a sensorimotor transformation. Here we review models of sensorimotor transformations that use a flexible intermediate representation that relies on basis functions. The use of basis functions as an intermediate is borrowed from the theory of nonlinear function approximation. We show that this approach provides a unifying insight into the neural basis of three crucial aspects of sensorimotor transformations, namely, computation, learning and short-term memory. This mathematical formalism is consistent with the responses of cortical neurons and provides a fresh perspective on the issue of frames of reference in spatial representations.

The term 'sensorimotor transformation' refers to the process by which sensory stimuli are converted into motor commands. This process is crucial to any biological organism or artificial system that possesses the ability to react to the environment. Accordingly, this topic has attracted considerable attention in neuroscience as well as engineering over the last 30 years.

A typical example of such a transformation is reaching with the hand toward a visual stimulus. In this case, as for most sensorimotor transformations, two issues must be resolved. First, one must determine the configuration of the arm that will bring the hand to the spatial location of the visual stimulus (kinematics). The second problem is specifying and controlling the application of force to determine the movement trajectory (dynamics)[1,2]. This review focuses almost exclusively on kinematics (see Wolpert and Ghahramani, this issue, for models of movement dynamics).

Our goal is to provide an overview of the basis function approach to sensorimotor transformations. In this approach, sensory information is recoded into a flexible intermediate representation to facilitate the transformation into a motor command. This has the advantage of explaining how the same neurons can be engaged in three seemingly distinct aspects of sensorimotor transformations, namely, computation, learning and short-term memory. We first review the theory behind representing and transforming spatial information using basis functions. Next we describe how these transformations can be learned using biologically plausible algorithms. Finally, we explain how to implement short-term spatial memory and updating of motor plans using these representations. In each case, we examine the extent to which models relying on basis functions are consistent with known neurobiology. Remarkably, all three tasks—computation, learning and short-term memory of spatial representations—can be efficiently handled using a neural architecture derived from the basis function approach. As we will see, a basis function representation is a form of population code, and we argue that the exceptional computational versatility of basis functions may explain why population codes are so ubiquitous in the brain.

Basis functions for sensorimotor transformations

Sensorimotor transformations are often formalized in terms of coordinate transformations. For instance, to reach for an object currently in view, the brain must compute the changes in joint angles of the arm that will bring the hand to the desired spatial location. This computation requires combining visual information—the retinal or eye-centered coordinates of the object—with signals related to the posture of body parts, such as the position of the eyes in the head (eye position), the position of the head with respect to the trunk (head position) and the starting position of the arm. We refer to such positional signals as 'posture signals'. In this way, we can recast a coordinate transformation as the computation of the value of a particular function. This function takes visual and postural signals as input and produces as output the set of changes in joint angles required to solve the task (for example, bring the hand to the target). Recasting the coordinate transformation as computing the value of a function makes it easier to generate and test biologically inspired models of this aspect of brain function.

We will adopt a vectorial notation for the signals being transformed. V is a vector that encodes an object's location in eye-centered space. It has three components, which correspond, respectively, to the image's azimuth and elevation on the retina and the object's distance from the retina. This is clearly not the form of the representation used by the brain, but the vector format is not important at this stage (see below). We use similar notations, P and J, for the posture signals and the change in joint coordinates. A coordinate transform can then be written as a function, f(), mapping V and P onto J: J = f(V,P).

It is useful to divide all functions into two classes, linear and nonlinear (Fig. 1). Sensorimotor transformations almost exclusively belong to the second class. The nonlinearity arises from the geometry of our joints. The change in spatial location of the hand that results from bending the elbow depends not only on the amplitude of the elbow movement, but also on the state of the shoulder joint. As a result, a neural network implementation of a sensorimotor transformation requires at least three layers. There must be at least one intermediate layer (the so-called 'hidden layer') to

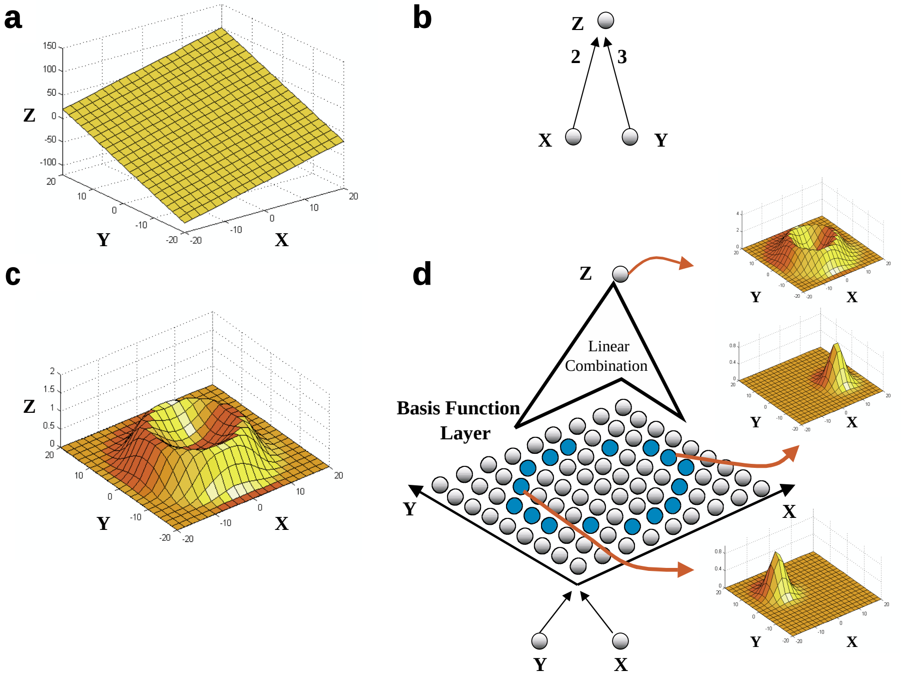

Fig. 1. Linear and nonlinear functions. (**a**) The linear function $z = 2x + 3y$. In general, a function is linear if it can be written as a weighted sum of its input variables plus a constant. All other functions are nonlinear. Linear functions form lines (one variable), planes (two variables, as shown here) or hyperplanes (more than two variables). (**b**) Linear functions can be implemented in two-layer networks. The network shown corresponds to the linear function in (**a**). (**c**) The nonlinear function $z = \exp(-(x^2 + y^2 - 100)^2/1000)$. Nonlinear functions are not planar and can form surfaces that can be arbitrarily complex, such as a circular ridge. (**d**) A neural network implementation of the nonlinear function in (**c**) using Gaussian basis functions in the intermediate representation (**Box 1**). The basis function units are organized so as to form a map in the x–y plane. Right, two representative response functions of these basis function units. The activity of the output unit is obtained by taking a linear sum of the basis function units. In this example, the weights of the blue units onto the output unit are set to one, whereas all the other units have a weight of zero. As a result, the output unit mathematically sums a set of Gaussian functions arranged along a circle in the x–y plane. This leads to the response function (right), which is similar to the circular ridge in (**c**).

recode the sensory inputs before they can be transformed into motor commands (Box 1; Fig. 1d). One of the challenges in computational neuroscience has been to identify intermediate representations that are both biologically plausible and computationally efficient for these nonlinear mappings.

One solution involves using intermediate units that compute basis functions[3–5], because most functions of interest can be approximated using a linear combination of basis functions. The best known basis set is that used by the Fourier transform: any function can be expressed as the linear sum of a series of cosine and sine functions of arbitrary amplitudes and frequencies. Many other functions can be used to form basis sets (Box 1; Fig. 1d).

When applied to sensorimotor transformations, and in particular to the example of reaching toward a visual target, the idea is that the reaching motor command J can be obtained by taking a weighted sum of N basis functions $\{B_i(V,P)\}_{i=1}^{N}$ of the visual and posture signals, V and P:

$$J = \sum_{i=1}^{N} w_i B_i\,(V,P). \qquad (1)$$

The set of weights, $\{w_i\}_{i=1}^{N}$, is specific to the reaching motor command being computed and, as we will see later, can be determined using simple learning rules.

Many choices are available for the basis functions. For instance, one can use the set of all Gaussian functions of V and P, which are a subset of a larger family known as radial basis functions (RBF) [3]. The network in Fig. 1d is an example of a radial basis function network in which the variables considered are x and y instead of V and P. This type of representation is also sometime called a population code, that is, a code in which the variables are encoded through the activity of a large population of neurons with overlapping bell-shape tuning curves. Population codes may be ubiquitous in the nervous system because they provide basis sets.

An alternative choice for a basis set is the product of a Gaussian function of the eye-centered position of an object (V) and a sigmoid function of eye position (P). In an idealized neuron that performs this calculation (Fig. 2a), the receptive field is eye-centered, that is, it remains at the same location relative to the fovea regardless of eye position (Fig. 2b). However, response gain (that is, its amplitude) changes with eye position.

From a biological point of view, one problem with Eq. 1 is the format of the input and output vectors. For instance, we used polar coordinates for vector V, yet no such vector has been explicitly identified in the cortex. Instead, the visual position of objects is encoded by the activity of a large number of binocular neurons forming the retinotopic maps in the early visual areas. This does not mean that we cannot use the basis function framework. We simply replace the vector V with a new vector, V^A, which has as many components as there are neurons in the retinotopic map; each component corresponds to the activity (for example, firing rate) of one neuron. Likewise, the vectors P and J can be replaced by the corresponding neuronal patterns of activities P^A and J^A (Fig. 2c). Many network models of sensorimotor transformations rely on such basis function representations in their intermediate layer[6–9].

Biological plausibility of the basis function approach

The basis function approach requires that the tuning curves of neurons in intermediate stages of computation provide a basis function set. A set of functions constitutes a basis set if certain requirements are met. First, the functions must combine their inputs, for example, the visual input, V, and the posture signals, P, nonlinearly and so that they cannot be decomposed into separate functions of V and functions of P. This rules out linear functions and functions of the type $B_i(V,P) = C_i(V) + D_i(P)$. Furthermore, the functions must be able to fully cover the range of possible input values[5]. In other words, there must be units with all possible combinations of selectivity for visual and posture signals.

Neurons whose response can be described by a Gaussian function of retinal location multiplied by a sigmoidal function of eye position would qualify (Fig. 2a and b)[5]. Many such gain-

Box 1. Implementing functions with neural networks—the basis function approach.

Linear functions can be easily implemented with two-layer neural networks. For instance, the function $z = 2x + 3y$ requires only two input units, one output unit and two connections with weights 2 and 3 (Fig. 1b). Nonlinear functions, on the other hand, generally require a network with at least one intermediate layer. Several options are available for the intermediate representation but one solution, known as basis functions, is particularly relevant in the context of sensorimotor transformations. Basis functions are building blocks that, when combined linearly, can approximate any nonlinear function. For instance (Fig. 1d), one can use Gaussian functions of x and y as a basis set. A network containing a large number of these units, each representing a Gaussian, form a two-dimensional map of the x–y plane. The activity of the single output unit is determined by taking a weighted sum of the activity of the basis function units. If we set the weights of the units shown in blue to one, and all the others to zero, the resulting function—the sum of the Gaussian functions corresponding to the blue unit—looks like a circular ridge very similar to the one obtained by plotting the nonlinear function, $z = \exp(-(x^2 + y^2 - 100)^2/1000)$ (compare Fig. 1c and d).

The same idea can be applied to sensorimotor transformations. However, unlike our example, sensorimotor transformations involve more than two input variables (see text). Similarly, the output is also multidimensional. Yet these high-dimensional transformations can be computed in a way exactly analogous to the low-dimensional computation illustrated here.

modulated neurons are found in the parietal lobe, where there are neurons with all possible combinations of visual and eye position selectivities[10]. Gain modulations between sensory and posture signals are also observed in occipital[11–15] and premotor cortices[16], suggesting that basis function representations may be widely used.

In many early papers, gain modulation by posture signals was reported to be linear, not sigmoidal. This is clearly incompatible with the basis function hypothesis, as basis functions require nonlinear tuning curves. These experiments, however, were designed to detect an effect, not to distinguish the precise form of the gain field. A linear model of gain fields was simple and lent itself most easily to statistical testing. However, recent experiments[17] and new analyses[5] reveal significant nonlinearities consistent with sigmoidal modulation. This conclusion is based on data from the parietal cortex, but given the similarities among gain fields throughout the cortex, it is reasonable to think that it applies to most gain fields.

Another line of evidence in support of the basis function approach comes from the study of hemineglect patients with right parietal cortex lesions. These patients tend to ignore sensory stimuli located on their left[18]. 'Left', however, can be defined with respect to multiple frames of reference; it could be the left side with respect to the eyes (that is, the left visual field), head or body. For example, consider a subject who turns his head to the left of a stimulus that lies directly in front of him, but then moves his eyes far to the right. The stimulus will now lie on the midline with respect to the body, to the right with respect to the head, and to the left with respect to the eyes. By assessing neglect using a variety of body postures and stimulus locations, one can attempt to determine the relevant frame of reference for a given patient's neglect. Interestingly, such experiments show that neglect often affects multiple frames of reference (for review, see ref. 19).

This observation fits well with one property of basis function representations, namely, that they encode location in multiple frames of reference simultaneously. For instance, a basis function representation integrating a visual input with eye position signals (Fig. 2c) represents the location of objects in eye- and head-centered frames of reference simultaneously[5]. Indeed, to recover the position of an object in, say, head-centered coordinates, one must compute a function of the eye-centered position of the object as well as the current eye and head positions. As for any other function, this can be done with a simple linear transformation of the activity of the basis function units. As a result, a simulated lesion of a basis function representation can explain why hemineglect affects several frames of reference across a variety of tasks[19].

The multiplicity of frames of reference is one of the most dis-

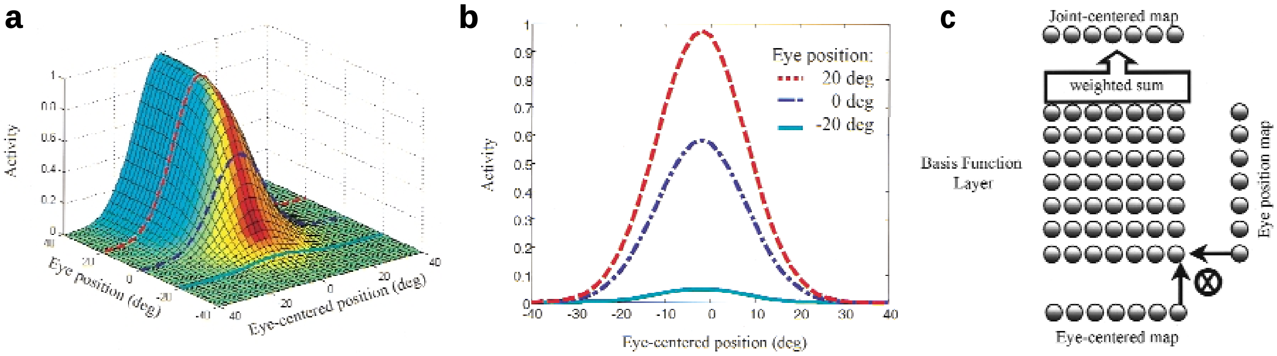

Fig. 2. Basis function units. (**a**) The response function of a basis function unit computing a product of a Gaussian function of retinal location (eye-centered position) multiplied by a sigmoidal function of eye position. (**b**) A mapping of the retinotopic receptive field derived from a unit with the properties in (**a**) for three different eye positions. Bold lines in (**a**) correspond to the three curves shown here. The receptive field always peaks at the same retinal location, but the gain (or amplitude) of the response varies with eye position. Gain modulations similar to this are found in many cortical areas, from V1 to the premotor cortex. (**c**) A neural network model for nonlinear sensorimotor transformations using basis functions. The input layers encode the retinal location of an object and the current eye position, whereas the output layer encodes the change in joint angles of the arm. Other inputs signals are needed to compute the change in joint angles of the arm, such as head position, starting hand location and so on, but for simplicity we show only the visual and eye position inputs. The tuning curve of the eye-centered and eye-position units are assumed to follow Gaussian and sigmoid functions, respectively. This nonlinear sensorimotor transformation requires an intermediate layer. In the case illustrated here, the intermediate layer uses basis function units. Each unit computes the product of the activity of one input unit from the eye-centered map and one input unit from the eye position map. This leads to the response of the basis function units (**a**, **b**). The transformation from the basis function to the output units involves a simple linear transformation, namely, a weighted sum of the activity of the basis functions units. This is the main advantage of this approach: once the basis functions are computed, nonlinear transformations become linear.

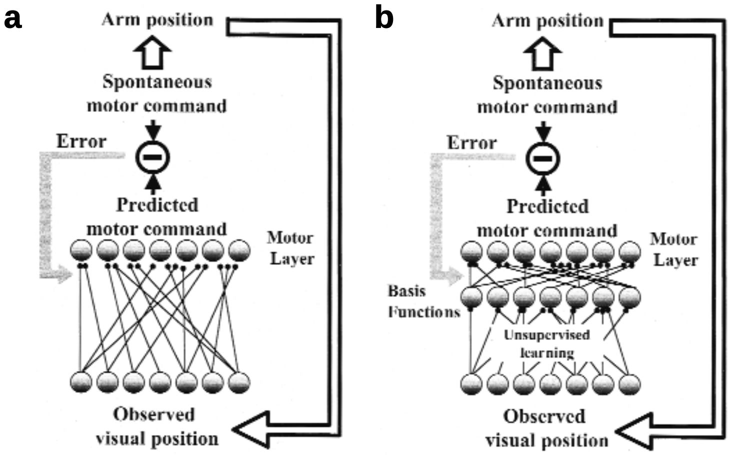

Fig. 3. Learning sensorimotor transformations in neural networks. (**a**) Learning motor commands with spontaneous movements. The motor layer generates a spontaneous pattern of activity. This activity is fed into the arm, resulting in a new arm position. The position of the arm is observed in the input sensory layer. The sensory activity is passed through the weights to compute a predicted motor command. The weights can then be adjusted according to an error signal obtained by computing the difference between the spontaneous and predicted motor command. By repeating this procedure many times, one can learn the appropriate sensorimotor mapping. Note that the same motor units are involved in all steps: generating and remembering the spontaneous motor command, computing the predicted motor command and computing the error signal. In most models, the neuronal mechanisms underlying these various steps are not specified; the motor units are simply assumed to be able to perform the required computations. (**b**) When the transformation is nonlinear, as is the case for arm movements, an intermediate layer of units is required. If basis functions are used in the intermediate layer, learning can be done in two stages. The weights to the basis function can be learned with an unsupervised, or self-organizing rule because the basis functions only depend on the input, not on the motor command computed in the output layer. The weights to the output layer are equivalent to the ones in the network in (**a**) and can be learned using the same error signal.

tinguishing properties of basis function representations. In more traditional approaches to spatial representations, object position is represented in maps using one particular frame of reference. Multiple frames of reference require multiple maps, and a neuron can only contribute to one frame of reference, specific to its map. By contrast, in a basis function map, each neuron contributes to multiple frames of reference. Thus, basis function neurons are ideally placed to coordinate different behaviors, such as moving the eyes and hand to the same object, even though these movements must be programmed in distinct coordinates.

Learning sensorimotor transformations

A few sensorimotor transformations (such as the eyeblink reflex) may already be wired at birth and require little training. In most cases, however, the mapping from sensory to motor coordinates must be learned and updated through life, as eyes, arms and other body parts change in size and weight. As before, we focus exclusively on the issue of coordinate transformations. How do we learn and maintain a mapping of sensory coordinates of objects into motor coordinates? Piaget[20] proposed that babies learn by associating spontaneous motor commands with the sensory consequences of those spontaneous actions. Consider how this would apply to a two-layer network used to control arm movements. We assume that the input layer encodes the visual location of the hand, whereas the output layer represents reaching motor commands in joint-centered coordinates. (Motor commands actually require changes in joint angles, but for simplicity we will consider absolute angles.) On each trial, the network generates a spontaneous pattern of activity in the motor layer. This pattern is fed to the arm, which moves accordingly, and the network receives visual feedback of the resulting hand position. At this point, the system can learn to associate the patterns in the sensory and output layers. In particular, the Hebb rule can be used to increase the weights between co-active sensory and motor units[21].

A further refinement is to treat the position of the hand after the spontaneous movement as a target to be reached. The idea is to first compute the motor command that the network would have generated if it had aimed for that location from the start. We call this the 'predicted' motor command. (Note that this movement is not actually executed.) We can then compare this predicted command to the original spontaneous one. Because the spontaneous command is precisely the command that brings the hand to the current location, we should adjust the network weights to make the predicted motor command closer to the spontaneous one. To compute the predicted motor command, we use the visually determined hand location after the spontaneous movement as a network input, and use the current weights to compute the activity of the motor units. If the predicted and spontaneous commands are the same, no learning is required. If they differ, then the difference between the spontaneous and predicted motor commands can be used as an error signal to adjust the weights (Fig. 3a). For instance, one could use a learning rule known as the delta rule[22], which takes the form $\delta w_{ij} = \alpha a_i (a_j^* - a_j)$, where δw_{ij} is the change in the weight between the presynaptic sensory unit i and postsynaptic motor unit j, α is a learning rate, a_i is the activity of the presynaptic unit, a_j^* is the spontaneous postsynaptic motor activity, and a_j is the predicted postsynaptic motor activity.

This strategy works well if the sensorimotor transformation is linear—if it can be implemented in a two-layer network (Box 1)—such as learning to make an eye movement to a visual target. Indeed, the retinal location of a target and the saccade vector required to acquire that target are identical. The transformation from a sensory map (for example, V1) to a motor map (superior colliculus) is therefore an identity mapping, which is a linear transformation[21].

Unfortunately, however, most sensorimotor transformations are nonlinear, and the networks that compute them require at least one intermediate layer. We showed above that a good choice for the intermediate representation is to use basis functions. This turns out to be a good choice for learning as well. Indeed, with basis functions, we can decompose learning into two independent stages: first, learning the basis functions and, second, learning the transformation from basis functions to motor commands (Fig. 3b).

The basis functions can be learned via a purely unsupervised learning rule. In other words, they can be learned without regard to the motor commands being computed—before the baby even starts to move his arm. Indeed, because by definition any basis function set can be used to construct any motor command (Eq. 1), the choice of a basis set is independent of the motor commands to be learned. The choice is constrained instead by general considerations about the computational properties of the basis functions, such as their robustness to noise or their efficiency during learning[3], as well as considerations about biological plausibility. Gaussian and sigmoid functions are often a good choice in both respects. It is also crucial that the basis functions tile the entire range of input values encountered, that is, they must form a map in the

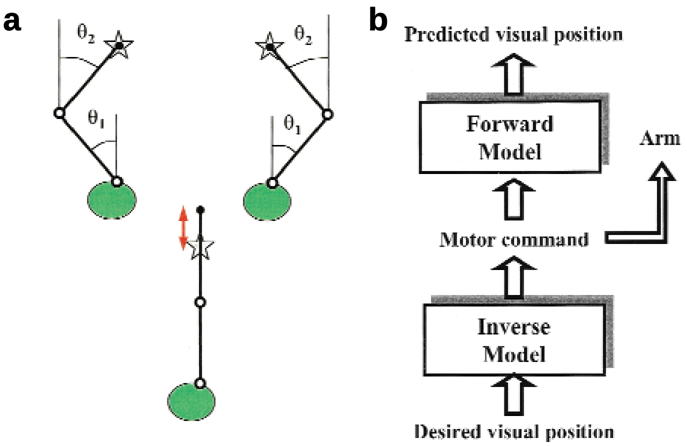

Fig. 4. One-to-many sensorimotor transformations. (**a**) To reach for the location indicated by the star, a two-joint arm working in a two-dimensional plane can adopt exactly two configurations (top). The transformation in this case is from one sensory input into one of two motor commands. In the case illustrated here, the two arm configurations are mirror images of each other, that is, the joint angles have the same amplitude but reversed signs. As a result, the average of angle θ_1 and θ_2 over the two configurations is zero. The average configuration (bottom) would therefore overshoot the target (red arrow). Because averaging is precisely what the network in Fig. 3b does during training, this transformation cannot be learned properly with this kind of networks. (**b**) One solution is to use two networks, one for the sensorimotor transformation (the inverse model) and one for predicting the sensory consequences of the motor command (the forward model). Each of these networks typically involves three or more layers such as the one shown in Fig. 3b. Learning in this system is achieved by learning the forward model (for example, through the generation of spontaneous motor commands) and then using the forward model to train the inverse model.

input space considered, like the one shown in Fig. 2c. This can be done by using variations of the Hebb and delta learning rules with an additional term to enforce global competition, to ensure that each neuron learns distinct basis functions[23–25].

The second problem—learning the transformation from basis functions to motor commands—is easy because motor commands are linear combinations of basis functions (Fig. 2c). We only need to learn the linear weights in Eq. 1, which can be done as outlined above with a simple delta rule[7,8] (Fig. 3b). However, even a three-layer network trained with a combination of delta rules and unsupervised learning does not always suffice. If a nonlinear sensorimotor transformation is a one-to-many mapping, then a network trained with delta rules will converge on the average of all possible solutions, which is not necessarily itself a solution. Consider a planar two-joint arm whose segments are of equal length (Fig. 4a). There are exactly two configurations that will reach a target lying less than two segment lengths from the origin. A network trained using the delta rule would converge on the average of these two configurations, which unfortunately would overshoot the target. A similar problem would arise in the primate arm.

Jordan and Rumelhart[26] proposed a solution to this problem. They use two networks: one to compute the required movement, and one to map that movement into a prediction of the resulting visual location of the hand (Fig. 4b). In engineering terms, these two networks constitute an inverse and a forward model, respectively. The forward model can be thought of as an internal model of the arm; it attempts to predict the perceptual consequence of one's action (see Wolpert and Ghahramani, this issue, for experimental evidence supporting forward models in the brain). One role of the forward model is to ensure that the system does not average across commands leading to the same sensory outcome[26].

Each model requires three or more layers, as both the forward and inverse models are typically nonlinear. In principle, one can use basis function networks and the learning strategy described above. Jordan and Rumelhart instead used the backpropagation algorithm[27], which allows efficient propagation of the error signal through the network layers but is not believed to be biologically plausible.

Thanks to computational modeling, we now have a better idea of how Piaget's proposal for motor learning could be implemented in biologically plausible neural networks using simple learning rules. The basis function hypothesis, in particular, is well suited to the learning of complex nonlinear sensorimotor transformations. Recent work on the parietal cortex has suggested how these models can be tested[28,29].

Short-term memory and updating motor plans

As shown above, representations using basis functions are useful for computing and learning sensorimotor transformations because they provide a basis set. It turns out that these representations are also efficient for short-term or working memory, the ability to remember a location for a few hundred milliseconds to a few seconds. Short-term memory is closely related to the problem of updating motor plans.

Consider a retinotopic map, that is, a population of neurons with Gaussian tuning curves for the eye-centered location of an object. In such a map, the appearance of an object triggers a 'hill' of activity centered at the location of the stimulus (Fig. 5a). This hill can serve as a memory of the object if it can be maintained over time after the object disappears[30]. Much data supports the notion that this is indeed how short-term memory is manifested in the cortex[30–34]. (Note that the retinotopic map need not be arranged topographically on the cortical surface. For example, area LIP seems to contain such a map, inasmuch as the response fields of the neurons completely tile visual sensory space, even though neurons coding adjacent locations do not necessarily lie next to one another.)

To maintain a hill of activity in a network of units with Gaussian tuning curves, one needs to add lateral connections with specific weights. Because Gaussian tuning curves are basis function, it is particularly easy to find the appropriate values analytically or to learn them with biologically plausible learning procedures such as the delta rule[35]. A typical solution is to use weights that generate local excitation and long-range inhibition. Such connections seem to exist in the cortex and are extensively used in computational modeling of cortical circuits (for example, ref. 36). The resulting networks are simply basis function maps to which lateral connections have been added. Computation of motor commands and maintenance of spatial short-term memory can therefore be done by the same basis function neurons. This could explain why many neurons showing gain modulation in the parietal cortex also show memory activity (L.H.S., unpublished data). The addition of lateral connections does not alter in any way the computational properties of the basis function map; indeed, lateral connections can contribute to computation of the basis functions themselves[37] or serve as optimal nonlinear noise filters[38].

One requirement for a memory system in retinotopic coordinates is that activity be updated to take into account each eye or head movement. Consider a memory saccade task in which the target appears 15° to the right of the fixation point and is then extinguished. If the eyes are then moved 10° to the right before acquiring the target, the internal representation of the target loca-

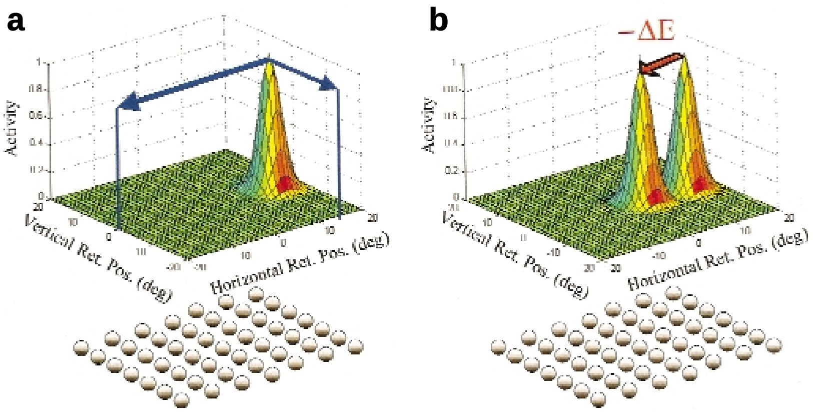

Fig. 5. Short-term memory networks for sensorimotor transformations. (**a**) A hill of activity (top) in a network organized in a two-dimensional retinotopic map (bottom). If the hill is a stable state for the network (that is, if it maintains itself over time), then the position of the peak of the hill can be used as a memory for retinal location. In the case illustrated here, the network is storing the value (15^o, 0^o), as indicated by the arrows. (**b**) To remain spatially accurate, a memory of the retinal location of an object must be updated after each eye movement. This updating can take place in such memory networks by displacing the hill by an amount equal to minus the eye displacement (–Δ**E**). The response to a +10^o horizontal eye movement is shown.

tion must be updated because the target itself is no longer visible. In this particular case, the invisible target would now lie 5^o to the right of the fixation point. More generally, if R is the object's retinal location and the eyes move by ΔE, the new retinal location is R – ΔE. Neurophysiological evidence indicates that such updating does occur in the brain, not only for eye-centered memory maps in response to eye movements[32,39,40] but also for head-centered maps in response to head movements[41].

This updating mechanism is closely related to the problem of updating motor plans. Indeed, the eye movement required to foveate an object is simply equal to the retinal location of the object, from which it follows that remembering a retinal location is mathematically equivalent to remembering an eye movement. Therefore, the updating mechanism we have just considered can also be interpreted as updating a motor plan for an eye movement. Interestingly, updated memory activity is often observed in neurons that also show presaccadic activity[40,42].

If hills of activity are used for short-term memory, updating in this example would require moving the hill from the position corresponding to 15^o right to the position corresponding to 5^o right (Fig. 5b). As long as an eye-velocity or eye-displacement signal is available to the network, this updating mechanism is very easy to implement with the networks above. For instance, the lateral connections can be modified such that the network moves the hills with a velocity proportional to the negative of the eye velocity[35]. Other ideas have been explored as well, but available experimental data do not constrain which of these variations are used in the brain[43–45]. In particular, different schemes may be used in areas with and without topographic maps.

Discussion

Our understanding of the neural basis of sensorimotor transformations has made outstanding progress over the last 20 years, in part because of new experimental data but also thanks to the development of the theoretical ideas we have reviewed in this paper. This is a prime example in which theoretical and experimental approaches have been successfully integrated in neuroscience.

Perhaps the most remarkable observation that has come out of this research is that seemingly distinct problems, such as computing, learning and remembering sensorimotor transformations can be handled by the same neural architecture. As we have seen, networks of units with bell-shaped or sigmoidal tuning curves for sensory and posture signals are perfectly suited to all three tasks. The key property is that units with bell-shaped tuning curves provide basis functions, which, when combined linearly, make it easy to compute and learn nonlinear mappings. The simple addition of appropriate lateral connections then adds information storage (memory) plus the ability to update stored information after each change in posture.

Clearly, basis function networks are only the beginning of the story. Unresolved issues abound, starting with the problem that basis function representations are subject to combinatorial explosion, that is, the number of neurons required increases exponentially with the number of signals being integrated. Hence a basis function map using 10 neurons per signal and integrating 12 signals would require 10^{12} neurons, more than total number of neurons available in the cortex. One solution is to use multiple modules of basis functions. For instance, one could use two maps, connected in a hierarchical fashion, in which the first map integrates 6 signals and the second map the remaining 6 signals, for a total of 2×10^6, which is close to, if not less than, the total number of neurons available in a single cortical area.

It is clear that the brain indeed uses multiple cortical modules for sensorimotor transformations, and it will be interesting to identify the computational principles underlying this modular architecture[46,47]. Another problem will be to understand how these circuits handle neuronal noise. Neurons are known to be noisy. This implies that we must not only worry about how they compute but also how they do so efficiently or reliably in the presence of noise[38]. These questions promise to provide interesting issues for computational neuroscience to address for years to come.

Acknowledgements

A.P. is supported by a Young Investigator Award from ONR and fellowships from the Sloan Foundation and the McDonnell-Pew foundation. L.H.S. is supported by fellowships from the Sloan and Klingenstein foundations, and by NEI. We thank Daphne Bavelier and Suliann Ben Hamed for comments on earlier versions of this manuscript.

Received 30 May; accepted 4 October 2000

1. Tresch, M., Saltiel, P. & Bizzi, E. The construction of movement by the spinal cord. *Nat. Neurosci.* 2, 162–167 (1999).
2. Todorov, E. Direct cortical control of muscle activation in voluntary arm movements: a model. *Nat. Neurosci.* 3, 391–398 (2000).
3. Poggio, T. A theory of how the brain might work. *Cold Spring Harbor Symp. Quant. Biol.* 55, 899–910 (1990).
4. Pouget, A. & Sejnowski, T. A neural model of the cortical representation of egocentric distance. *Cereb. Cortex* 4, 314–329 (1994).
5. Pouget, A. & Sejnowski, T. Spatial transformations in the parietal cortex using basis functions. *J. Cogn. Neurosci.* 9, 222–237 (1997).
6. Groh, J. & Sparks, D. Two models for transforming auditory signals from head-centered to eye-centered coordinates. *Biol. Cybern.* 67, 291–302 (1992).
7. Burnod, Y. *et al.* Visuomotor transformations underlying arm movements toward visual targets: a neural network model of cerebral cortical operations. *J. Neurosci.* 12, 1435–1453 (1992).
8. Salinas, E. & Abbot, L. Transfer of coded information from sensory to motor networks. *J. Neurosci.* 15, 6461–6474 (1995).

9. Zipser, D. & Andersen, R. A back-propagation programmed network that stimulates reponse properties of a subset of posterior parietal neurons. *Nature* 331, 679–684 (1988).
10. Andersen, R., Essick, G. & Siegel, R. Encoding of spatial location by posterior parietal neurons. *Science* 230, 456–458 (1985).
11. Trotter, Y., Celebrini, S., Stricanne, B., Thorpe, S. & Imbert, M. Neural processing of stereopsis as a function of viewing distance in primate visual area V1. *J. Neurophysiol.* 76, 2872–2885 (1997).
12. Trotter, Y. & Celebrini, S. Gaze direction controls response gain in primary visual-cortex neurons. *Nature* 398, 239–242 (1999).
13. Galletti, C. & Battaglini, P. Gaze-dependent visual neurons in area {V3a} of monkey prestriate cortex. *J. Neurosci.* 9, 1112–1125 (1989).
14. Bremmer, F., Ilg, U., Thiele, A., Distler, C. & Hoffman, K. Eye position effects in monkey cortex. I: Visual and pursuit-related activity in extrastriate areas MT and MST. *J. Neurophysiol.* 77, 944–961 (1997).
15. Cumming, B. & Parker, A. Binocular neurons in V1 of awake monkeys are selective for absolute, not relative, disparity. *J. Neurosci.* 19, 5602–5618 (1999).
16. Boussaoud, D., Barth, T. & Wise, S. Effects of gaze on apparent visual responses of frontal cortex neurons. *Exp. Brain Res.* 93, 423–434 (1993).
17. Squatrito, S. & Maioli, M. Gaze field properties of eye position neurones in areas MST and 7a of macaque monkey. *Vis. Neurosci.* 13, 385–398 (1996).
18. Vallar, G. Spatial hemineglect in humans. *Trends Cogn. Sci.* 2, 87–97 (1998).
19. Pouget, A., Deneve, S. & Sejnowski, T. Frames of reference in hemineglect: a computational approach. *Prog. Brain Res.* 121, 81–97 (1999).
20. Piaget, J. *The Origins of Intelligence in Children* (The Norton Library, New York, 1952).
21. Kuperstein, M. Neural model of adaptative hand-eye coordination for single postures. *Science* 239, 1308–1311 (1988).
22. Widrow, B. & Hoff, M. E. in *Conference proceedings of WESCON*, 96–104 (1960).
23. Moody, J. & Darken, C. Fast learning in networks of locally-tuned processing units. *Neural Comput.* 1, 281–294 (1989).
24. Hinton, G. & Brown, A. in *Neural Information Processing Systems* vol. 12, 122–128 (MIT Press, Cambridge Massachusetts, 2000).
25. Olshausen, B. A. & Field, D. J. Sparse coding with an overcomplete basis set: a strategy employed by V1? *Vision Res.* 37, 3311–3325 (1997).
26. Jordan, M. & Rumelhart, D. Forward models: supervised learning with a distal teacher. *Cognit. Sci.* 16, 307–354 (1990).
27. Rumelhart, D., Hinton, G. & Williams, R. in *Parallel Distributed Processing* (eds. Rumelhart, D., McClelland, J. & Group, P. R.) 318–362 (MIT Press, Cambridge, Massachusetts, 1986).
28. Desmurget, M. *et al.* Role of the posterior parietal cortex in updating reaching movements to a visual target. *Nat. Neurosci.* 2, 563–567 (1999).
29. Wolpert, D. M., Goodbody, S. J. & Husain, M. Maintaining internal representations: the role of the human superior parietal lobe. *Nat. Neurosci.* 1, 529–533 (1998).
30. Amit, D. The hebbian paradigm reintegrated — local reverberations as internal representations. *Behav. Brain Sci.* 18, 617–626 (1995).
31. Fuster, J. *Memory in the Cerebral Cortex: An Empirical Approach to Neural Networks in the Human and Nonhuman Primate* (MIT Press, Cambridge, Massachusetts, 1995).
32. Goldberg, M. & Bruce, C. Primate frontal eye fields. III. Maintenance of a spatially accurate saccade signal. *J. Neurophysiol.* 64, 489–508 (1990).
33. Gnadt, J. & Mays, L. Neurons in monkey parietal area LIP are tuned for eye-movement parameters in three-dimensional space. *J. Neurophysiol.* 73, 280–297 (1995).
34. Funahashi, S., Bruce, C. & Goldman-Rakic, P. Dorsolateral prefrontal lesions and oculomotor delayed response performance: evidence for mnemonic “scotomas”. *J. Neurosci.* 13, 1479–1497 (1993).
35. Zhang, K. Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: a theory. *J. Neurosci.* 16, 2112–2126 (1996).
36. Somers, D. C., Nelson, S. B. & Sur, M. An emergent model of orientation selectivity in cat visual cortical simple cells. *J. Neurosci.* 15, 5448–5465 (1995).
37. Salinas, E. & Abbott, L. F. A model of multiplicative neural responses in parietal cortex. *Proc. Natl. Acad. Sci. USA* 93, 11956–11961 (1996).
38. Deneve, S., Latham, P. & Pouget, A. Reading population codes: A neural implementation of ideal observers. *Nat. Neurosci.* 2, 740–745 (1999).
39. Walker, M., Fitzgibbon, E. & Goldberg, M. Neurons in the monkey superior colliculus predict the visual result of impending saccadic eye movements. *J. Neurophysiol.* 73, 1988–2003 (1995).
40. Mazzoni, P., Bracewell, R., Barash, S. & Andersen, R. Motor intention activity in the macaque’s lateral intraparietal area. I. Dissociation of motor plan from sensory memory. *J. Neurophysiol.* 76, 1439–1456 (1996).
41. Graziano, M., Hu, X. & Gross, C. Coding the locations of objects in the dark. *Science* 277, 239–241 (1997).
42. Duhamel, J. R., Colby, C. L. & Goldberg, M. E. The updating of the representation of visual space in parietal cortex by intended eye movements. *Science* 255, 90–92 (1992).
43. Droulez, J. & Berthoz, A. A neural model of sensoritopic maps with predictive short-term memory properties. *Proc. Natl. Acad. Sci. USA* 88, 9653–9657 (1991).
44. Dominey, P. & Arbib, M. A cortico-subcortical model for the generation of spatially accurate sequential saccades. *Cereb. Cortex* 2, 153–175 (1992).
45. Seung, H. How the brain keeps the eyes still. *Proc. Natl. Acad. Sci. USA* 93, 13339–13344 (1996).
46. Snyder, L., Batista, A. & Andersen, R. Coding of intention in the posterior parietal cortex. *Nature* 386, 167–170 (1997).
47. Snyder, L., Grieve, K., Brotchie, P. & Andersen, R. Separate body- and world-referenced representations of visual space in parietal cortex. *Nature* 394, 887–891 (1998).

Viewpoint • Models identify hidden assumptions

It is not only theorists who make models. All biologists work with explicit or implicit ‘word models’ that describe their vision of how a system works. One of the most important functions of theoretical and computational neuroscience is to translate these word models into more rigorous statements that can be checked for consistency, robustness and generalization through calculations and/or numerical simulations.

The process of turning a word model into a formal mathematical model invariably forces the experimentalist to confront his or her hidden assumptions. I have often found that I have ‘skipped steps’ in my thinking that were only revealed when we sat down to construct a formal model. It is easy to tell ‘just so stories’ about cells, circuits and behavior, and discussion sections of journal articles are filled with them, but the exercise of trying to instantiate the assertions in those stories makes the missing links in all of our data and understanding pop into view.

Models offer a solution to one of the hardest problems in experimental biology: how far to generalize from the data one has collected. Neuroscientists work on an array of cells and circuits in lobsters, flies, fish, birds, rats, mice, monkeys and humans. Many of the ‘mistakes’ in neuroscience come from inappropriate generalizations from observations made in one system, or under a given set of conditions. Experimental work I did with Scott Hooper showed that when an oscillatory neuron was electrically coupled to a non-oscillatory cell, the two-cell network had a lower frequency than the isolated oscillator. We initially assumed that this was a general statement, but later learned from theoretical work that, depending on the properties of the oscillator, either an increase or decrease in frequency could be obtained. We had correctly understood our data, but we were unaware that the other case was possible because it did not occur in the particular system we were studying. This is at the core of the usefulness of theory for an experimentalist: it helps us know when we have found only a piece of the answer, and when we have understood the full set of possible outcomes from a given set of conditions.

Finally, theory is legitimized dreaming. We all became neuroscientists out of a deep desire to explore the mysteries of how the brain works. Most of us who do experimental work spend our days preoccupied with the myriad and mundane details that are so crucial to doing experiments and analyzing data. I came of age as a scientist believing that my career would be over if I were ever wrong. For me, participating in the development of highly speculative models was akin to learning to drive as a teenager. In both cases, I remember the thrill of the freedom of the open road (and some of the trepidation of getting lost or getting a flat tire). Speculative models suggest possibilities beyond those found in one’s laboratory, and can produce just that altered outlook that can send one on a new and exciting path.

Eve Marder
Volen Center, MS 013, Brandeis University, Waltham, Massachusetts 02454, USA
e-mail: marder@volen.brandeis.edu