Learning to Recognize and Reproduce Abstract Actions from Proprioception

Karl F. MacDorman*† Rawichote Chalodhorn† Hiroshi Ishiguro†
*Frontier Research Center
†Department of Adaptive Machine Systems
Graduate School of Engineering
Osaka University
Suita, Osaka 565-0871 Japan
kfm@ed.ams.eng.osaka-u.ac.jp

Abstract

We explore two controversial hypotheses through robotic implementation: (1) *Processes involved in recognition and response are tightly coupled both in their operation and epigenesis; and* (2) *processes involved in symbol emergence should respect the integrity of recognition and response while exploiting the fundamental periodicity of biological motion. To that end, this paper proposes a method of recognizing and generating motion patterns based on nonlinear principal component neural networks that are constrained to model both periodic and transitional movements. The method is evaluated by an examination of its ability to segment and generalize different kinds of soccer playing activity during a RoboCup match.*

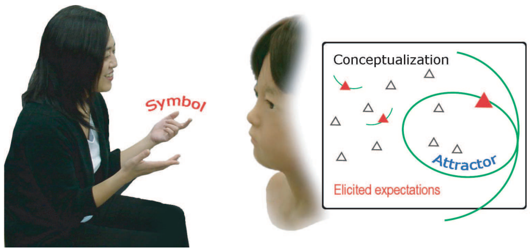

Figure 1. In the proposed approach, a neural network learns each kind of periodic or transitional movement in order to recognize and to generate it. Recent sensorimotor data elicit activity in corresponding networks, which segment the data and produce appropriate anticipatory responses. Active networks constitute an organism's conceptualization of the world since they embody expectations, derived from experience, about the outcomes of acts and what leads to what. It is assumed that behavior is purposive: affective appraisals guide the system toward desired states.

1 Introduction

Complex organisms recognize their relation to their surroundings and act accordingly. The above sentence sounds like a truism owing in part to the almost ubiquitous distinction between recognition and response in academic disciplines. Engineering has successfully developed pattern recognition and control as independent fields, and cognitive psychology and neuroscience often distinguish between sensory and motor processing with researchers specializing in one area or the other. Nevertheless, in some sense recognition and response entail one another. Recognizing an object, action, or sign is largely a matter of recognizing what it does for us and what we can do with it. Indeed, much of what we perceive can be described in terms of potential actions. *Doing* and *seeing* cannot so readily be distinguished because we acquaint ourselves with our world through what we do and our actions drive what distinctions we learn to make. None of this is meant to deny that we can experimentally isolate purely motor centers in the brain from purely sensory ones, but rather to assert that these centers are intimately linked both in their everyday operation and in their epigenetic development. Thus, as scientists and engineers, we may have reified the distinction between recognition and response, when their main difference is merely in descriptive focus.

In this paper, we will entertain and begin to explore two controversially and, as yet, unproven hypotheses: First, there is an integrity of recognition and response. We recognize an object or event largely because it elicits expectation about what we can do with it — or at least piggybacks on those kinds of expectations. In addition, these expectations are expressed in terms of (or decontextualized from) how motor signals transform sensory data. Second, biological motion is fundamentally periodic. To put it simply, patterns

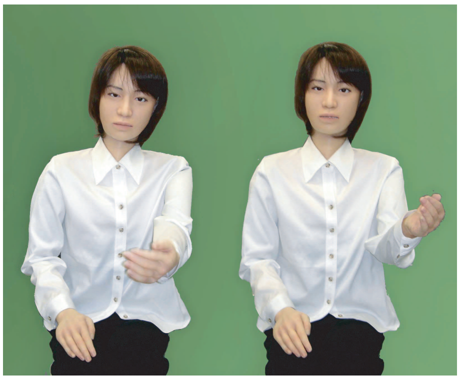

Figure 2. Actroid, the actress android, has 33 motors, which are driven by compressed air, to move its head, neck, arms, body, eyes, eyelids, and mouth. Actroid can make smooth and natural movements, including large and small gestures. Actroid has touch sensors in the arms and can access floor and infrared sensors and video cameras placed in the environment.

repeat. (If they did not, there would be little point in learning.) That is as much a function of the 'hardware' as it is the often routine nature of existence. Joints, for example, have a limited range and will eventually return, more or less, to a given configuration. Moreover, bodies have certain preferred states: for people walking is a more efficient means of locomotion than flailing about randomly. All gaits exhibit a certain periodicity as do many gestures and vocalizations.

This paper proposes a method of generalizing, recognizing, and generating patterns of behavior based on nonlinear principal component neural networks that are constrained to model both periodic and transitional movements. Each network is abstracted from a particular kind of movement. Learning is competitive because sensorimotor patterns that one network cannot learn will be assigned to another network, and redundant networks will be eliminated and their corresponding data reassigned to the most plausible alternative. Recognition is also competitive because proprioceptive data is associated with the network that best predicts it. (The data can be purely kinematic or dynamic depending on the dimensions of the sensorimotor phase space.) Since each network can recognize, learn, and generalize a particular type of motion and generate its generalization, the integrity of recognition and response are maintained. These generalizations are grounded in sensorimotor experience. They can be varied, depending on the networks' parametrization. They may be viewed as a kind of protosymbol. While we do not claim that the networks have neural analogues, we believe the brain must be able to implement similar functions.

1.1 The emergence of signs in communication

In one vein, we are exploring the application of periodically-constrained NLPCA neural networks to vocal and gesture recognition and generation. Our aim is to develop robots whose activity is capable of supporting the emergence of shared signs during communication. Signs take on meaning in a given situation and relationship, as influenced by an individual's emotional responses and motivation (see Figure 1). They reflect mutual expectations that develop over the course of many interactions. We hypothesize that signs provide developmental scaffolding for symbol emergence. For infants, the caregiver's intentions are key to fostering the development of shared signs.

We believe that periodically-constrained NLPCA neural networks could be one of the embedded mechanisms that support the development of shared signs. We are testing this hypothesis by comparing the behavior generalized by these neural networks with Vicon motion capture data from mother-infant interactions.[1] The results of behavioral studies are applied to the android robot, Actroid, which has 33 degrees of freedom (See Figure 2).

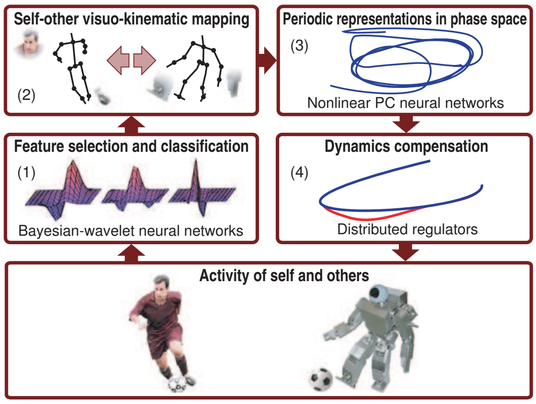

Figure 3. Periodic nonlinear principal component networks may characterize motion patterns in a much larger system for recognizing, learning, and responding behavior.

1.2 Mimesis loop

In a separate vein, we are applying NLPCA neural networks to the learning of cooperative behavior in robot soccer. Although techniques from reinforcement learning can be borrowed to guide a robot's behavior toward goals, they cannot be directly applied to the state space of a humanoid robot because of its enormous size. The approach outlined

[1] From this we have ascertained that certain important micro-behaviors that make movement seem lifelike may have been overlooked in the approach outlined here, and we are starting to develop a micro-behavior filter.

in this paper can vastly reduce the size of the state space by segmenting it into different kinds of movements. A mimesis loop [3] may be used to capture many aspects of the sort of imitation involved in learning to play soccer and other sports. This paper addresses one aspect of the mimesis loop: the abstraction of a robot's own kinematic motions from its proprioceptive experience. Figure 3 roughly outlines how a mimesis loop might be realized in a soccer playing robot. Attentional mechanisms direct the robot's sensors toward the body parts of other players, and the robot maps successfully recognized body parts onto its own body schema. This paper introduces a method to abstract the robot's own kinematic patterns: our segmentation algorithm allocates proprioceptive data among periodic temporally-constrained nonlinear principal component neural networks (NLPCNNs) as they form appropriate generalizations.

The robot can use NLPCNNs to recognize the activities of other players, if the mapping from their bodies to its own has already been derived by some other method. Since each network correspond to a particular type of motion in a proprioceptive phase space, it can act as a protosymbol. Thus, the robot would be able to recognize the behavior of others because it has grounded their behavior in terms of its own body.

Although periodic NLPCNNs may be used to generate motion patterns, the robot must continuously respond to unexpected perturbations. There are a number of approaches to this control problem that do not require an explicit model. For example, distributed regulators [2] could set up flow vectors around learned trajectories, thus, converting them into basins of attraction in a phase space of possible actions.

1.3 Outline

This paper is organized as follows. Section 2 extends an NLPCNN with periodic and temporal constraints. Section 3 presents a method of assigning observations to NLPCNNs to segment proprioceptive data. Section 4 reports experimental results using NLPCNNs to characterize the behavior of a Fujitsu HOAP-1 humanoid robot that has been developed to play RoboCup soccer.

2 A periodic nonlinear principal component neural network

The human body has 244 degrees of freedom [15] and a vast array of proprioceptors. Excluding the hands, a humanoid robot generally has at least 20 degrees of freedom — and far more dimensions are required to describe its dynamics precisely. However, many approaches to controlling the dynamics of a robot are only tractable when sensory data is encoded in fewer dimensions (e.g., [9]). Fortunately, from the standpoint of a particular activity, the effective dimensionality may be much lower.

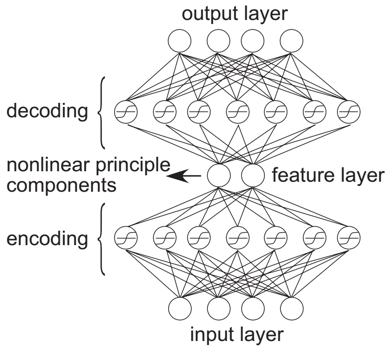

Figure 4. Target values presented at the output layer of a nonlinear principal component neural network are identical to input values. Nonlinear units comprise the encoding and decoding layers, while either linear or nonlinear units comprise the feature and output layers.

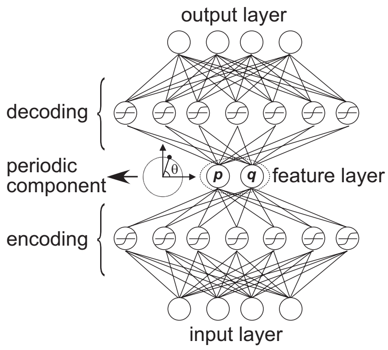

Figure 5. An NLPCA neural network with the activations of nodes p and q constrained to lie on the unit circle.

Given a coding function $f : \mathbb{R}^N \mapsto \mathbb{R}^P$ and decoding function $g : \mathbb{R}^P \mapsto \mathbb{R}^N$ that belong to the sets of continuous nonlinear functions $\mathcal{C}$ and $\mathcal{D}$, respectively, where $P < N$, nonlinear principle component networks minimize the error function E

$$\|\vec{x} - g(f(\vec{x}))\|^2, \quad \vec{x} \in \mathbb{R}^N$$

resulting in P principal components $[y_1 \cdots y_p] = f(\vec{x})$. Kramer (1991) first solved this problem by training a multilayer perceptron similar to the one shown in Figure 4 using the backpropagation of error, although a second order method such as conjugant gradient analysis converges to a solution faster for many large data sets. Tatani and Nakamura (2003) were the first to apply an NLPCNN to human and humanoid motions, though for dimensionality reduction only.

Nonlinear principal components analysis, unlike PCA (Karhunen-Loève expansion), which is a special case where

$\mathcal{C}$ and $\mathcal{D}$ are linear, does not have a unique solution, and no known computational method is guaranteed to find any of the globally optimal solutions. Nevertheless, for a 20-DoF humanoid robot, a hierarchically-constructed[2] NLPCNN has been shown to minimize error several times more than PCA when reducing to two-to-five dimensions [13].

2.1 The periodicity constraint

Because the coding function f of an NLPCNN is continuous, *(1)* projections to a curve or surface of lower dimensionality are suboptimal; *(2)* the curve or surface cannot intersect itself (e.g., be elliptical or annular); and *(3)* projections do not accurately represent discontinuities [8]. However, since the physical processes underlying motion data are continuous, discontinuities do not need to be modelled. Discontinuities caused by optimal projections can create instabilities for control algorithms (e.g., they allow points along the axis of symmetry of a parabola to be projected to either side of the parabola). Moreover, an NLPCNN with a circular node (Ridella et al., 1995, 1997) at the feature layer can learn self-intersecting curves and surfaces.

Kirby and Miranda (1996) constrained the activation values of a pair of nodes p and q in the feature layer of an NLPCNN to fall on the unit circle, thus acting as a single angular variable:

$$r = \sqrt{y_p^2 + y_q^2}, \;\; y_p \leftarrow y_p/r, \;\; y_q \leftarrow y_q/r$$

The delta values for backpropagation of the circular node-pair are calculated by the chain rule [4], resulting in the update rule

$$\delta_p \leftarrow (\delta_p y_q - \delta_q y_p) y_q / r^3, \;\; \delta_q \leftarrow (\delta_q y_p - \delta_p y_q) y_p / r^3$$

at the feature layer.

The hyperbolic tangent and other antisymmetric functions (i.e., $\varphi(x) = -\varphi(x)$) are generally preferred to the logistic function as the sigmoid in part because they are compatible with standard optimizations [6].[3] In addition, antisymmetric units can more easily be replaced with linear or circular units in the feature layer, since these units can produce negative activations. We propose using a slightly flatter antisymmetric function for the sigmoidal units with a similar response characteristic to $tanh$ (see Fig. 6). The advantage of this node is that it can be converted to a circular node-pair while still making use of its perviously learned weights.

[2]The joint encoder dimensionality of limbs is independently reduced, then the arms and the legs are paired and their dimensionality further reduced, and then finally the dimensionality of the entire body.

[3]These include mean cancellation, linear decorrelation using the K-L expansion, and covariance equalization.

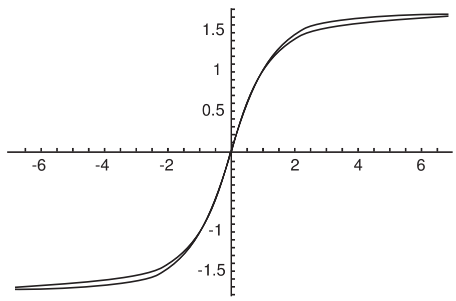

Figure 6. The popular hyperbolic tangent activation function $y \leftarrow 1.7159 \tanh(\frac{2}{3} y)$ can be approximated by a pair of circular nodes where the activation of the second node y_q is fixed at $\sqrt{1.9443}$ and the activation of the first node is calculated accordingly $y_p \leftarrow 1.7159 y_p / \sqrt{y_p^2 + 1.9443}$.

2.2 The temporal constraint

Neither linear nor nonlinear principal components analysis represent the time, relative time, or order in which data are collected.[4] This information, when available, can be used to reduce the number of layers and free parameters (i.e., weights) in the network and thereby its risk of converging slowly or settling into a solution that is only locally optimal. Since the activations y_p and y_q of the circular node-pair in the feature layer in effect represent a single free parameter, the angle θ, if θ is known, we can train the encoding and decoding subnetworks separately by presenting $k\cos(\theta)$ and $k\sin(\theta)$ as target output values for the encoding subnetwork and as input values for the decoding network.[5] Once a single period of data has been collected, temporal values can be converted to angular values $\theta = 2\pi \frac{t_k - t_0}{t_n - t_0}$ for data collected at any arbitrary time t_k during a period, starting at t_0 and ending at t_n. A network may similarly learn transitions between periodic movements when using a linear or sigmoidal activation node in the feature layer because these open-curve transitions do not restrict us to using nodes capable of forming a closed curve.[6] NLPCNNs with a circular feature node remain useful to identify the period of a motion pattern, especially when the pattern is irregular and, thus, begins and ends at points that are somewhat far from each other.

[4]Although a temporal dimension could be added to an autoassociative network, one drawback for online learning is that this dimension would need to be continuously rescaled as more data is collected to keep it within the activation range of the nodes.

[5]$k \approx 1.7$ for zero-mean data with variance equal to 1 based on principles discussed in [6].

[6]$y_{target} = 2k(\frac{t_k - t_0}{t_n - t_0} - \frac{1}{2})$, with $k \approx 1.4$.

3 Automatic segmentation

We conceived of the automatic segmentation problem as the problem of uniquely assigning data points to nonlinear principal component neural networks. It is possible to partition the points without reference to the predictions of the networks.[7] However, for our method each network's performance influences segmentation with more networks assigned to regions that are difficult to learn.

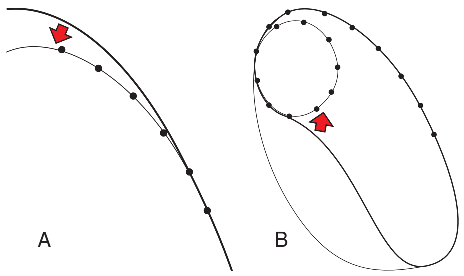

Figure 7. The thick line shows the output of an NLPCNN and the thin line shows the underlying distribution. The dots are data points. **A.** Before learning converges, allowing the network to learn data points despite a high prediction error accelerates learning. **B.** However, after convergence, it leads to segmentation errors.

As the robot begins to move, the first network is assigned some minimal number of data points (e.g., joint-angle vectors), and its training begins with those points. This gets the network's learning started quickly and provides it with enough information to determine the orientation and curvature of the trajectory. If the average prediction error of the data points assigned to a network is below some threshold, the network is assigned additional data points until that threshold has been reached. At that point, data points will be assigned to another network, and a network will be created, if it does not already exist. To avoid instabilities, only a single data point may shift its assignment from one network to another after each training cycle.

Since a network is allowed to learn more data points as long as its average prediction error per point is low enough, it may learn most data points well but exhibit slack near peripheral or recently learned data points. At the start of learning, the network should be challenged to learn data points even when its prediction error is large (see Fig. 7A). As learning converges, however, the slack leads to segmentation errors (see Fig. 7B). Therefore, we alter the method of segmentation once the network nears convergence (as determined by Bayesian methods [7] or crossvalidation) so that

$j \leftarrow 1,\ bucket \leftarrow 1,\ E \leftarrow 0$
$\forall \vec{x_i}$ {
  train$(network_j, \vec{x_i})$
  $E_i = \|\vec{x_i} - g(f(\vec{x_i}))\|^2,\ E \leftarrow E + E_i$
  if ($bucket > B_{max} \vee$
    (learning?$(network_j) \wedge E/bucket > E_{max}$) $\vee$
    $E_i > E_{i+1}$)
    $j \leftarrow j+1,\ bucket \leftarrow 1,\ E \leftarrow 0$ }

Listing 1: Pseudocode for segmentation.

a network may acquire neighboring points if its prediction error for those points is lower that the network currently assigned to those points.

4 Humanoid experiments

This section shows the result of automatic segmentation and neural network learning. We assess the accuracy of the result based on a manual segmentation of the data points and an analysis of how they are allocated among the networks.

First, we recorded motion data while a HOAP-1 humanoid robot played soccer in accordance with a hard-coded program [1]. Each data point is constituted by a 20-dimensional vector of joint angles. A standard (noncircular) NLPCNN reduced the dimensionality of the data from 20 to 3, which was determined to be the intrinsic dimensionality of the data by the ISOMAP procedure [14] We then applied our algorithm to segment, generalize, and generate humanoid motion.

Our algorithm uniquely assigned the data points among a number of circularly-constrained NLPCNNs. Each of the networks learned a periodic motion pattern by conjugate gradients. Our algorithm successfully generalized five out of six primary motion patterns: walking forward, turning

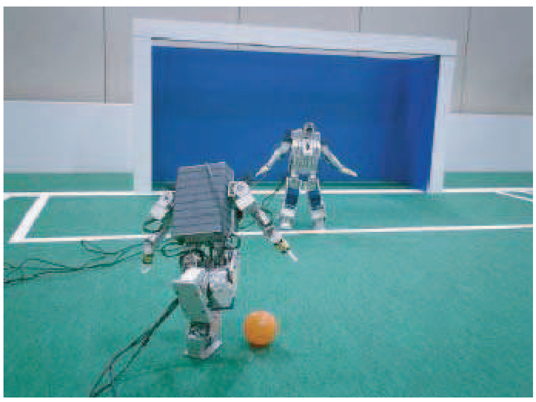

Figure 8. Fujitsu HOAP-1 robots are playing RoboCup soccer.

[7]For example, data points may be partitioned at the point at which a trajectory most closely doubles back on itself, if the distance between the two paths is within a certain threshold and the paths then diverge beyond another threshold.

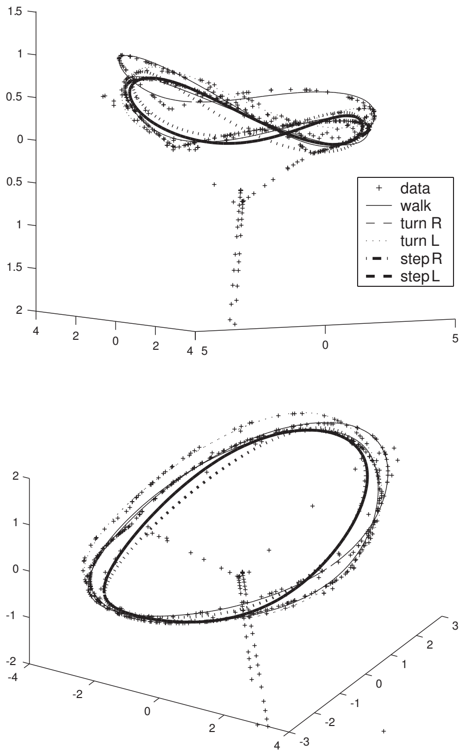

Figure 9. Recognized motion patterns embedded in the dimensions of the first three nonlinear principal components of the raw proprioceptive data. The top and bottom plots differ only in the viewpoint used for visualization.

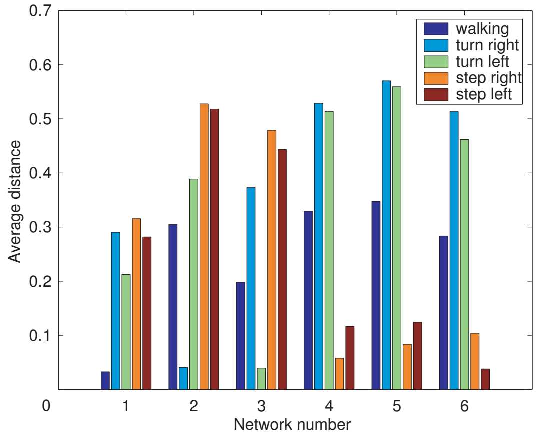

Figure 10. The average distance between the prediction of a network trained on manually segmented data and each of the automatically generated networks.

right or left, and side-stepping to the right or left. It failed to generalize as a single periodic trajectory the kicking motion, which has a highly irregular, self-intersecting shape. However, human subjects were also unable to determine the kicking trajectory from the data points.

Figure 9 shows that the automatic segmentation algorithm successfully employed circular NLPCNNs to separate and generalize five of the periodic motions. (The open-curve segmentation of transitions between periodic motions are omitted for clarity.) The periodic trajectories were generated by varying from 0 to 2π the angular parameter θ_i at the bottleneck layer of each of the circularly-constrained networks and mapping the result to the output layer for display. This demonstrates our method's capacity to generate periodic motions.

We calculated statistics based on running the automatic segmentation for 20 trails. The algorithm resulted in five decoding subnetworks for 45% of the trials, which is the most parsimonious solution. It resulted in six subnetworks for 50% of the trials, and seven for the remaining 5%.

Since the data was generated by the predefined behavior modules used by the Osaka University team in the 2003 RoboCup humanoid competition, each data point was already labelled and could be segmented into the five types of motion that had been successfully abstracted. To assess the accuracy of the automatic segmentation algorithm, we manually assigned the data points corresponding to each type of motion to five periodic temporally constrained NLPCNNs. Figure 10 shows the average distance between the prediction for each of these networks and each of the networks resulting from automatic segmentation.

The lowest bar indicates which pattern the networks, numbered 1 to 6 best match in terms of least average distance. Hence, the first network represents walking; the second represents turning right; the third turning left; the fourth and fifth sidestepping right; and the sixth sidestepping left. The fact that the fifth network is redundant, abstracting the same type of motion as the fourth, does not prevent the abstracted actions from supporting the mastery of soccer or some other task. Both networks can be used. The algorithm's capacity to reduce a vast amount of complex, raw data to just a few states may help reinforcement learning approaches to finesse the curse of dimensionality [12].

In a separate run of the learning and segmentation algorithm, the motion sequence of recorded data during soccer playing was walking forward, turning right, turning left, walking forward, sidestepping to the right, sidestepping to the left, and kicking. We counted the number of point belonging to each network before and after removing redundant networks. Redundant networks were removed by means of linear integration. The angular value θ was varied from 0 to 2π at the bottleneck layer of one network to obtain its predicted output. This value was fed into another network to obtain its predicted value. If the integral of the sum of the squared distances of the predicted outputs

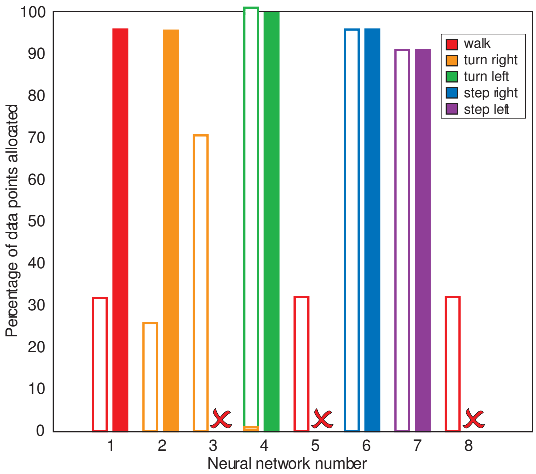

Figure 11. The percentage of data points allocated to each network before and after eliminating redundant networks and reassigning their data.

was less than a threshold, one network was removed and its points reassigned to the other network (see Figure 11). This method removed all redundant networks.

5 Conclusion

Our proposed algorithm abstracted five out of six types of humanoid motion through a process that combines learning and data point assignment among multiple neural networks. The networks perform periodic, temporally-constrained nonlinear principal component analysis. The decoding subnetworks generate motion patterns that accurately correspond to the five motions without including outliers caused by nondeterministic perturbations in the data. During 45% of training episodes, the algorithm generated no redundant networks; a redundant network appeared in 50% of the training episodes, and two appeared in 5% of them. Although the fourth and fifth networks represent the same type of motion, this does not prevent them from serving as action symbols for learning a complex task. By means of linear integration, we were able to remove redundant networks according to the proximity of their predictions.

A kind of behavior can be recognized by selecting the network that best predicts joint-angle values. It can be generated by varying the value of θ in the bottleneck layer. This shows the effectiveness of a tight coupling between recognition and response since the same networks may be used for both processes and they developed by the same mechanisms. The significance of periodicity may be more limited, however. Some motions are not periodic, and in the experiment the kicking motion, although it occurs repeatedly, was difficult to segment because of its highly irregular, self-intersecting shape.

References

[1] R. Chalodhorn, M. Aono, J. Ooga, M. Ogino, and M. Asada. Osaka University "Senchans 2003". In B. Browning, D. Polani, A. Bonarini, and K. Yoshida, editors, *RoboCup-2003: Robot Soccer World Cup VII*. Springer Verlag, 2003.

[2] T. Fujii. A new approach to the LQ design from the viewpoint of the inverse regulator problem. *IEEE Transactions on Automatic Control*, 32(11):995–1004, 1987.

[3] T. Inamura, I. Toshima, and Y. Nakamura. Acquiring motion elements for bidirectional computation of motion recognition and generation. In B. Siciliano and P. Dario, editors, *Experimental Robotics VIII*, pages 372–381. Springer, 2003.

[4] M. J. Kirby and R. Miranda. Circular nodes in neural networks. *Neural Computation*, 8(2):390–402, 1996.

[5] M. A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *Journal of the American Institute of Chemical Engineers*, 37(2):233–243, 1991.

[6] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller. Efficient BackProp. In G. B. Orr and K.-R. Müller, editors, *Neural Networks: Tricks of the Trade*, pages 1–44. Springer, 1998.

[7] D. J. MacKay. Probable networks and plausible predictions: A review of practical Bayesian methods for supervised neural networks. *Network: Computation in Neural Systems*, 6:469–505, 1995.

[8] E. C. Malthouse. Limitations of nonlinear PCA as performed with generic neural networks. *IEEE Transactions on Neural Networks*, 9(1):165–173, 1998.

[9] M. Okada, K. Tatani, and Y. Nakamura. Polynomial design of the nonlinear dynamics for the brain-like information processing of whole body motion. In *IEEE International Conference on Robotics and Automation*, pages 1410–1415, 2002.

[10] S. Ridella, S. Rovetta, and R. Zunino. Adaptive internal representation in circular back-propagation networks. *Neural Computing and Applications*, 3(4):222–333, 1995.

[11] S. Ridella, S. Rovetta, and R. Zunino. Circular backpropagation networks for classification. *IEEE Transaction on Neural Networks*, 8(1):84–97, 1997.

[12] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, Massachusetts, 1998.

[13] K. Tatani and Y. Nakamura. Dimensionality reduction and reproduction with hierarchical NLPCA neural networks extracting common space of multiple humanoid motion patterns. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1927–1932, Taipei, Taiwan, September 2003.

[14] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[15] V. M. Zatsiorsky. *Kinematics of Human Motion*. Human Kinetics, Urbana Champaign, 2002.