

winner-take-all readout given that the representation was constrained to have a uniquely defined mapping (i.e. in the scenario the assumption is made that the object to be tracked by the robot is always at the same elevation from the floor). Given that the model learns the relations in data space through optimisation processes [Hsieh, 2000] can use learned curves to simply project available sensory values through the learned function to get the second value, as the scale is preserved. Albeit its capability to precisely extract nonlinear relations from high-dimensional random datasets [Mandal et al., 2013] cannot provide any readout mechanisms to support a proper decoded representation of the extracted relations. This is due to the fact that the method cannot recover the sign and scale of the relations.

5.2.2 Analysis of the basic model

In the following section we introduce the features of the basic sensory relation learning model. The model acquires samples from the two sensory streams, encodes them in distributed populations of neurons (i.e. activation pattern), and then learns the correlation patterns between the two distributed representations (i.e. co-activation).

For the basic scenario we consider a bimodal relation learning problem. Each input sensory stream is encoded by a SOM composed of 100 neurons distributed in a one-dimensional lattice. We use a one-dimensional representation to encode single subsequent samples from the input stream and provides a sufficient substrate to extract and represent the input data distribution (i.e. through the shape and density of neurons' tuning curves). Each input sample elicits a distributed activation pattern across the network such that each neuron responds proportionally to the distance between his preferred value and the input sample value. If the activation patterns in each of the input SOMs are correlated, the Hebbian linkage between the two networks will enhance the links between highly activated neurons. Subsequent samples will determine the enhancement for correlated structure in the two input signals and depression for un-correlated modes.

As previously mentioned the correlation learning rule enhances correlated neural activities by strengthening synaptic weights following the original Hebbian postulate. This formulation only allows for an increase in synaptic weight between synchronously firing neurons. To prevent unlimited growth, it is necessary to extend the Hebb's rule to allow for weight decreases when neurons fire asynchronously using a covariance learning rule. In our experiments we used two rules for extracting the sensory relation, namely covariance learning and Oja's local PCA learning [Chen et al., 2007], both providing relatively similar results, with insignificant differences in computational implementations, but similar impact on the precision of the representation. In the case of the covariance learning rule, the synaptic strength between neurons i and j in populations p and q , respectively, is given by

$$\Delta w_{cross,i,j}^p(k) = \eta(k)(a_i^p(k) - \bar{a}_i^p(k))(a_j^q(k) - \bar{a}_j^q(k)), \quad (5.12)$$

where if we take a time average of the change in synaptic weight,

$$\bar{w}_{cross,i,j}^p(k) = \eta(k)(\overline{a_i^p(k)a_j^q(k)} - \bar{a}_i^p(k)\bar{a}_j^q(k)), \quad (5.13)$$

the first term on the right-hand side denotes the Hebbian synapse and the second term may be viewed as an activity-dependent threshold that changes with the product of time-

averaged pre- and postsynaptic activity levels. If, on average, the presynaptic activity $a_i^p(k)$ is independent on the postsynaptic activity $a_j^q(k)$, namely $\overline{a_i^p(k)a_j^q(k)} - \overline{a_i^p(k)}\overline{a_j^q(k)}$, then no change in synaptic weight should occur. As a special case of the covariance learning rule, Oja's local PCA learning, is a local and computationally efficient learning rule, keeping the Euclidean norm of a neuron's incoming synaptic weight vector at unity. The online version of Oja's rule used in our work assumes the weight update is given by

$$w_{cross,i,j}^p(k+1) = \frac{w_{cross,i,j}^p(k) + \eta(k)a_i^p(k)a_j^p(k)}{\sqrt{\sum_{l=1}^N (w_{cross,l,j}^p(k) + \eta(k)a_l^p(k)a_j^p(k))^2}}, \quad (5.14)$$

In order to test the functionality of the basic model we fed the network with correlated artificial sensory datasets. Each sensory dataset contained 1500 samples and followed different data distributions. As the proposed model comprises multiple learning and adaptation processes we often varied the input data distribution, such that we were able to analyse the behaviour and performance of the network by feeding data with uniform, nonuniform, or mixed probability distributions. An overview of some notable experiments is given in Figure 5.12. In the first experiment we feed sensory data with a hidden linear sensory relation with nonuniform data distribution (e.g. convex probability distribution), Figure 5.12 a left panel. The network extracts the relation and encodes it in the strength of the Hebbian links and in the tuning curves of each input SOM neuron. Higher density areas in the input space are characterised by narrower tuning curves and wider areas by broader ones. Consistent with the learned sensory data distribution, the network allocates more neurons to represent areas with a higher density (i.e. narrow tuning curve), and less neurons for coarser represented areas in the input space. The capability to encode the density of the data distribution can be used to define reliability maps of the sensors, and subsequently used in fault detection and accommodation.

In a second experiment we feed sensory data with a hidden nonlinear sensory relation (i.e. second order power-law) following a nonuniform data distribution (e.g. convex and powerlaw probability distributions), Figure 5.12 b left panel. Similar to the first scenario we observe that the network extracts the hidden relation, sensory data distribution, and judiciously allocates neurons for a consistent representation. Furthermore, we observe that the learned tuning curves' shapes and densities are uneven (heterogeneous), providing a non-equidistant tiling of the input space, and representing the irregularities and variability describing real-world data. In the current and all the other experiments we performed the representation method produced comparable results with [Ganguli et al., 2014]. We consider that the proposed approach in the thesis provides an alternative formulation of the efficient coding hypothesis for a neural population encoding a scalar stimulus variable drawn from an unknown prior distribution. In [Ganguli et al., 2014] the information-maximizing solution provided precise and intuitive predictions of the relationship between sensory prior, physiology, and perception: more frequently occurring stimuli should be encoded with a proportionally higher number of cells and a proportionally higher perceptual sensitivity for the frequently occurring stimuli. Our model was able to unsupervisedly obtain representations consistent with the predictions.

In the third third experiment we fed uniformly distributed data in the $[-1, 1]$ interval implementing a nonlinear periodic function (i.e. sine wave). Tiling evenly the entire input

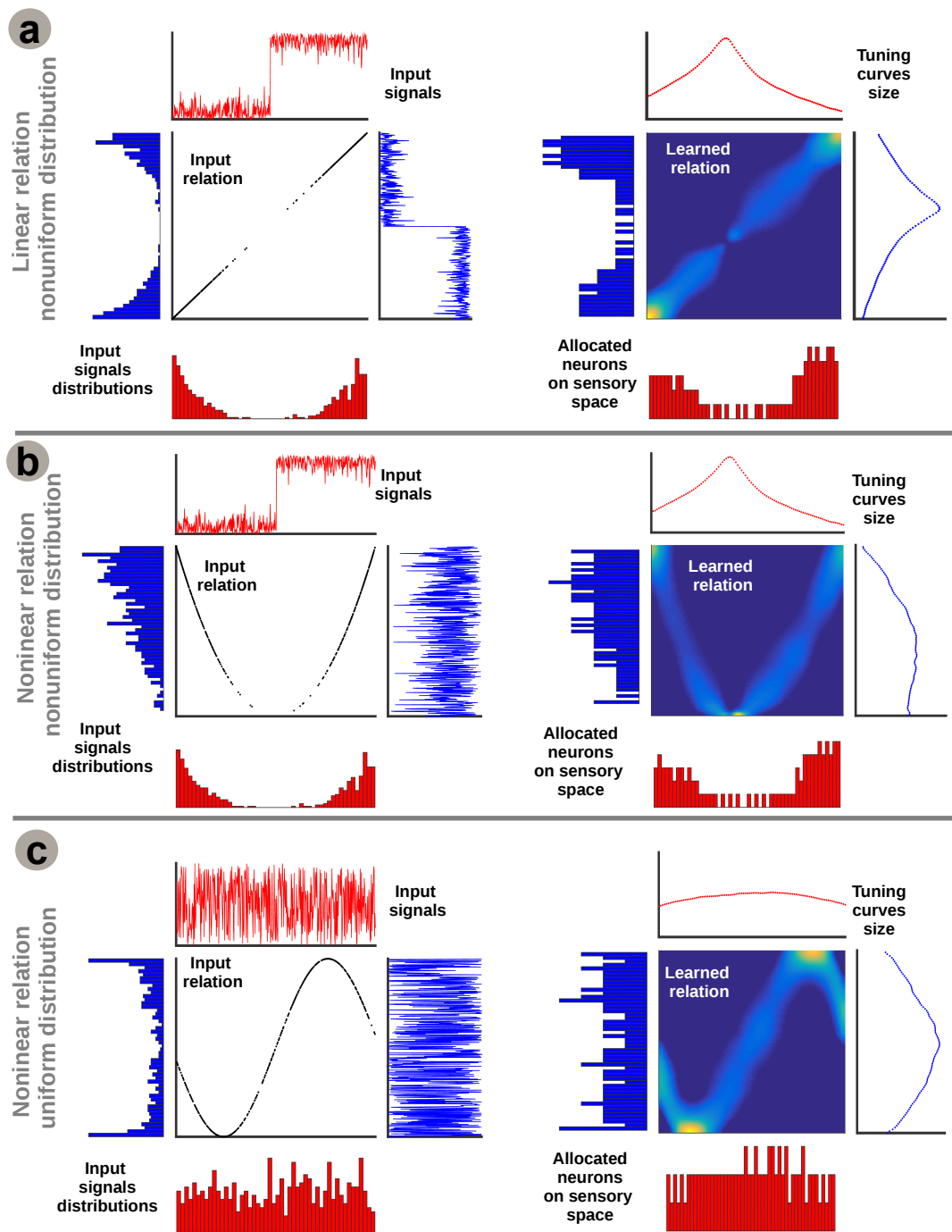


Fig. 5.12: Analysis of the basic model in a bimodal scenario. Different hidden relations and data distributions: input data and its probability distribution (left); learned relation and allocated resources (i.e. neurons) according to input distributions (right). a) Linear sensory relation with nonuniform data distribution; b) Nonlinear sensory relation with nonuniform data distribution; c) Nonlinear sensory relation with uniform data distribution.

space, the network allocates neurons uniformly, such that each region of the input space is equally represented.

Given incoming streams of correlated sensory data, each input SOM uses cooperation, competition, and adaptation (plasticity) to learn and represent the input data statistics in a heterogeneous population code. The representation process is jointly evolving with the relation extraction process, such that, through Hebbian learning, the network learns the underlying relation between the data, given that the input are efficiently represented in the SOM. Subsequently, the network uses the stable state (the learned relation) for cue integration, such that the learned relation (weight matrix) imposes the constraint on the possible values a sensory stream can have. At this stage, during cue integration each sensory modality representation will do its best to keep consistency with all the relations it is involved in, subject to the constraint imposed by the relation encoded by the weight vector.

An important aspect is that the network models synaptogenesis, such that initially the SOM projection weights are 0, and it doesn't need any prior information about the span of the input data distribution. This aspect, as well as the fact that the two concurrent processes evolve simultaneously, is consistent with the processes known to explain development in cortical circuitry. Featuring biologically plausible mechanisms the network increases its robustness capabilities (i.e. adapt to incoming streams of sensory data by enhancing / penalizing contributions) as on the longer timescale the input representation process adapts the structure for the faster sensor fusion process. After relaxing in a stable state the network contains a fully informative representation of the input data and the learned sensory relation. In order to make use of the learned relation we developed a simple readout mechanism. Given the ordered representation of the input data space onto the SOM lattices, one can find the corresponding real-world values by finding the best (optimal) solution of a cost function of maximal sensory elicited activation given input patterns. Bounding the value of the cost function with learned preferred values, a simple optimization method decodes the corresponding sensor value.

5.2.3 Inference and fault tolerance capabilities

After the learning process, the network stores a stable representation of the hidden relation between the two sensory inputs considered during training. By considering only one input sensory source, the network can infer the corresponding quantity for the missing source by using the learned co-activation pattern stored in the Hebbian linkage.

Given one input sample from the input sensory stream, the network computes the elicited activity in the input SOM population (pre-synaptic neurons). The resulting activity pattern is projected through the Hebbian linkage to compute the post-synaptic activation pattern in the output SOM population. Due to the all-to-all connectivity pattern, the activity of a single neuron in the output population is given by the sum of (Hebbian) weighted activity values in the input population. The resulting output activation pattern will peak at the most active (post-synaptic) neuron given the pre-synaptic input pattern. The position in the SOM lattice and the corresponding activation value are subsequently used for decoding the population activation pattern and recover the real-world sensory value.

We developed two methods for decoding the activity pattern and extract the corresponding real-world value. The first method is a naïve decoder, which simply computes a term to finely tune the preferred value of the most active (winning) neuron towards a more precise estimate. Given samples from the input stream $x(t)$ the most active neuron has index i in the output SOM, a preferred value $w_{in,i}^p$ and $\xi_i^p(k)$ tuning curve size, the corresponding increment term is given by

$$d_{fi}^p(k) = \sqrt{2\xi_i^p(k)^2 \log(\sqrt{2\pi}a_i^p(k)\xi_i^p(k)^2)}. \quad (5.15)$$

Depending on the position of the winning neuron in the N-dimensional lattice the recovered value $y(t)$ is computed as

$$y(t) = \begin{cases} w_{in,i}^p + d_{fi}^p, & \text{if } i \geq \frac{N}{2} \\ w_{in,i}^p - d_{fi}^p, & \text{if } i < \frac{N}{2} \end{cases}$$

A second, more precise, decoding mechanism is based on an optimisation method to recover the real-world value given known bounds in the input space. The bounds are obtained as minimum and maximum of a cost function of the distance between the current preferred value of the winner neuron and the input sample. The optimiser is based on Brent's method [Brent, 2013] which uses a recursive method to find the global optimum of a function for which the analytical form of its derivative is not available or too complex. Using this approach, after applying the input sensory stream and finding the winner in the input SOM population, the decoding decision is based on the position of the winner. Two bounds (i.e. left and right) are defined with respect to the winner's position such that the recovered value is obtained by running the algorithm between the preferred values of the neurons with indices given by the bounds. The method is not guaranteed to converge to global minima (of the cost function) and it's not immune to boundary effects, if winners are placed at the extremes of the SOM population.

In order to emphasize the capabilities of the two decoding mechanisms we provide in Figure 5.13 a brief analysis for some of the sensory learning scenarios previously used in the chapter. As one can see the decoding performance is satisfactory, yet the recovered values lie around the correct input pattern. By analysing the learned representation stored in the Hebbian matrix we noticed that, due to the asymmetric neighbourhood function in the input SOMs, the activity will saturate at the edges of the latent representation space. This behaviour is also visible in the co-activation pattern, such that the higher activity values characterise the bounds of the Hebbian representation towards the edges. Both decoding mechanisms assume that by applying one input to the network and projecting the sensory elicited activity pattern on the Hebbian matrix we can extract a plausible activation pattern for the missing sensory modality.

When decoding the activity pattern both approaches provided a relatively similar recovered probability distribution shape. This interesting behaviour relates the boundary effects in the SOM representation and Hebbian co-activation pattern with the extracted sensory data distribution learned from the data. Inspecting both decoders' probability distributions we observed that if the input data is uniformly distributed decoders' output is biased. The resulting distributions have a convex profile, concentrating a large number of

Given a learned relation, we apply samples from the input space on one input, project it through the Hebbian matrix and get an activity pattern to decode. The peak of the Gaussian activity pattern corresponds to one value in the output space.

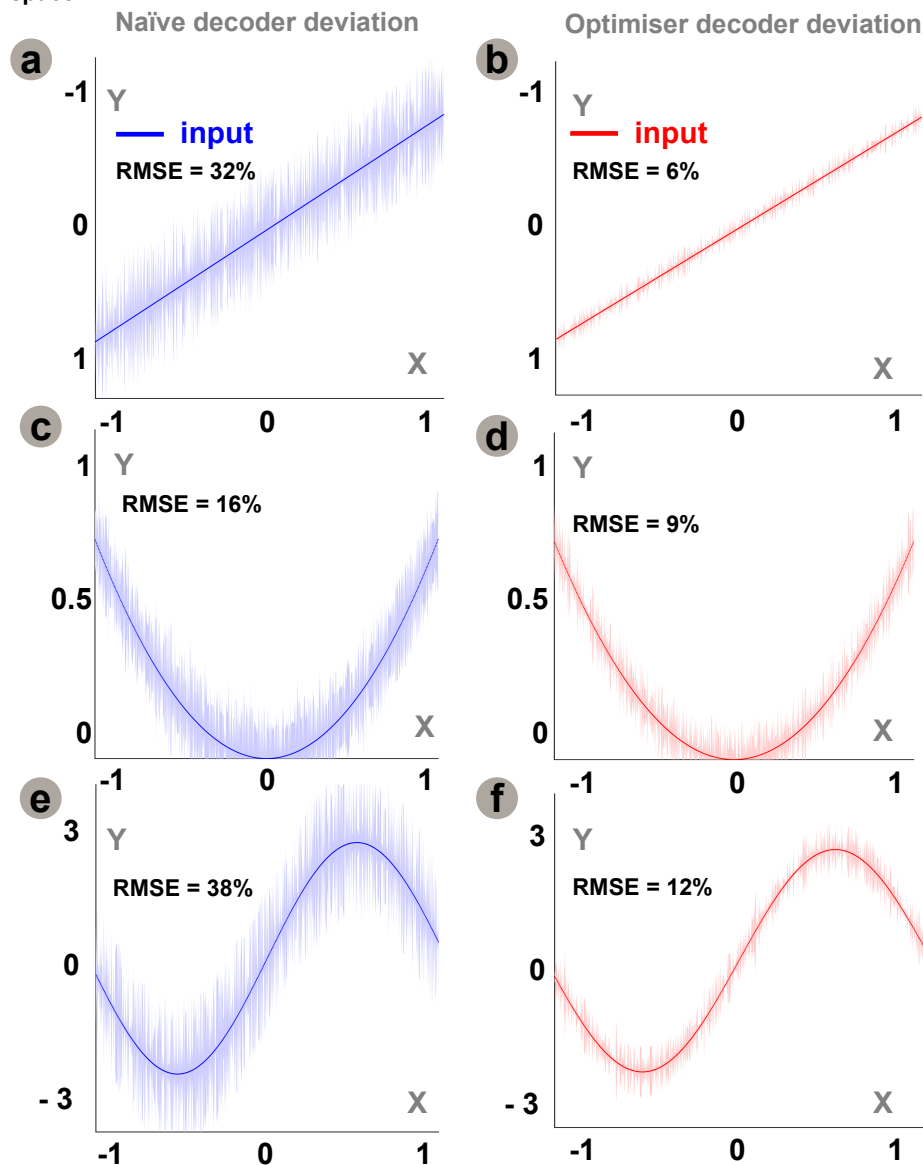


Fig. 5.13: Analysis of decoder performance for various types of sensory data relations. Performance of the naïve decoder for: a) Linear input, b) Nonlinear input symmetric input, c) Nonlinear periodic input; Performance of the optimised decoder: a) Linear input, b) Nonlinear input symmetric input, c) Nonlinear periodic input.

samples towards the edges of the histogram with a large variance, while precisely decoded areas follow a relatively uniform distribution. We notice that the optimiser based decoder, although more complex, provides better recovery results (smaller RMSE is better), such that the deviation is relatively small for linear relations (RMSE: 0.0613) in comparison with the naïve approach, which provides really imprecise recovery values (RMSE: 0.3247). For nonlinear relations the optimiser decoder is performing relatively well (RMSE: 0.0912), overtaking the naïve decoder, which surprisingly performs better than in the linear case,