Simultaneous concept formation driven by predictability

Alexander Gepperth and Louis-Charles Caron
Department of Electronics and Computer Science
École Nationale Superieure de Techniques Avancées
858 Boulevard des Maréchaux, 91762 Palaiseau, France
Email: alexander.gepperth@ensta-paristech.fr

***Abstract*—This study is conducted in the context of developmental learning in embodied agents who have multiple data sources (sensors) at their disposal. We describe an online learning method that simultaneously discovers "meaningful" concepts in the associated processing streams, extending methods such as PCA, SOM or sparse coding to the multimodal case. In addition to the avoidance of redundancies in the concepts derived from single modalities, we claim that "meaningful" concepts are those who have statistical relations *across modalities*. This is a reasonable claim because measurements by different sensors often have common cause in the external world and therefore carry correlated information. To capture such cross-modal relations while avoiding redundancy of concepts, we propose a set of interacting self-organization processes which are modulated by local predictability. To validate the fundamental applicability of the method, we conduct a plausible simulation experiment with synthetic data and find that those concepts which are predictable from other modalities successively "grow", i.e., become over-represented, whereas concepts that are not predictable become systematically under-represented. We conclude the article by a discussion of applicability in real-world robotics scenarios.**

I. Introduction

The autonomous formation of representations is a currently very active research topic in developmental robotics[1], [2], [3], [4]. Such concepts may be formed at low abstraction levels (and thus usually be termed "features") or at high abstraction levels (where they tend to be termed "concepts"). While it is generally agreed that concepts derived from a single information source should be encouraged to be diverse, as it is the case in sparse coding[5], ICA[6] or competitive learning approaches[7], biological and behavioral evidence suggests a great deal of correlations between concepts derived from different sources. As individual sources are usually corrupted by (structured) noise, issues of multisensory integration become crucial for stable perception and performance, as can be seen in audio-visual facilitation[8], contour integration[9] and multisensory integration[10]. Such sources may be different sensory inputs (vision/touch, vision/audition etc.), results of divergent processing (ventral/dorsal processing in visual cortex) or even different locations on a retinotopic surface such as V1. It has furthermore been shown in various experiments that humans are able to integrate multi-sensory cues in a fashion that is close to being Bayes-optimal[10].

In this contribution, we therefore deal with the problem of how multisensory features or concepts may be formed that are particularly suited for performing multisensory integration. Such concepts must be statistically related across sensory modalities while being non-redundant within their own modality. For achieving this in an online learning process, we propose a variant of the PROPRE (projection-prediction) algorithm[11]: PROPRE is a neural learning method which uses *projection* to map input stimuli to a two-dimensional neural representation ("induced representation") while using *prediction* from another neural representation ("reference representation") to modulate the adaptation of the projection step[1]. This results in selectivity to distinct patterns whose presence can be inferred from activity in the reference representation.

For the present study, we assume two sensory modalities providing input stimuli which are treated by two independent PROPRE instances, resulting in two induced representations containing concept-sensitive neurons. As can be seen from Fig. 1, the prediction steps of each PROPRE instance use the induced representations of the other instance as reference representations, thus realizing a mutually self-organizing concept formation process.

A. Related work

Our focus on predictability is motivated by a conceptual work [12], arguing that symbolic quantities should be diverse on the one hand, and on the other hand be defined by their *power to predict* other quantities. Our work differs from [12] in that we focus on quantities that *can be predicted*, and in that we propose and evaluate a concrete algorithm. We focus on predictability because we find that it can be naturally incorporated into local learning algorithms, whereas using predictive power necessarily involves bi-directional non-local operations.

A conceptually similar approach, which is moreover implemented in a robotic agent, is presented in [13]. This work extracts multimodal concepts from feature vector arising from visual and haptic processing streams by concatenating them and subjecting the resulting vector to principal components analysis (PCA). By construction of PCA, the basis vectors of the resulting transformation will be those whose multimodal

[1]The term "prediction" is meant to indicate "meaningful inference", irrespective of time. More specifically, given quantities A and B, it indicates that a distribution for B is estimated from the value of A.

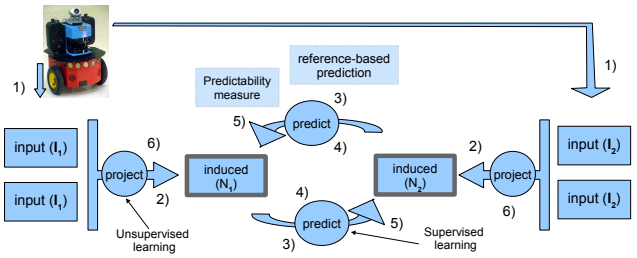

Fig. 1. Schematic overview of the used PROPRE learning algorithm for two sensory modalities. Note that learning in both modalities is completely symmetric, with neither modality having a special role for the other. The numbers indicate the order in which steps are executed within a single PROPRE iteration. Please see text for details of the algorithm.

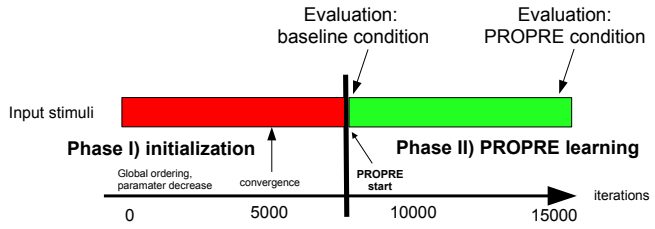

Fig. 2. Time course of PROPRE learning. In phase I, PROPRE learning is not applied: all input samples contribute to the formation of the induced representation, therefore the induced representation is formed by a pure SOM algorithm. In phase II, PROPRE learning is enabled: the current pattern contributes to learning only if activity in the induced representation is predictable enough. Time points of performance evaluations (see text) are given on the upper side of the diagram.

components are maximally correlated, which can be used to improve a multisensory classification task. The main difference to our approach is that we aim at online learning in a behaving agent, and that our approach maintains separate representations in different modalities which are however aligned to each other.

II. Methods

In the whole article, we will work with two-dimensional distributions on a discrete grid, denoted "representations", and we will express "neural activity" in an arbitrary representation X by $z^X(\vec{x},t)$. Synaptic connections between positions $\vec{x}$ and $\vec{y}$ in representations X and Y are denoted by $w^{X-Y}_{\vec{x}\vec{y}}$.

A. Workflow of a single PROPRE iteration

The presented algorithm is a priori independent of the embedding into any kind of robotic system; its only assumption is that all input representations I periodically receive new values in a synchronous manner. In most robotic agents, such values are provided at regular intervals by sensors updating their perception of the world, or by subsequent computations on new sensory inputs. Every time this happens, one PROPRE *iteration* is executed, consisting of the following steps which are also illustrated in Fig. 1. We use the terms and symbols given in Fig. 1.

1) new data is fed into input representations I_k
2) **projection:** activity in the two induced representations N_k, $k = 1,2$ is formed by projection of the input representations I_k, see Sec. II-C
3) **prediction:** based on activity in the induced representations, the reference-based predictions $\mathrm{pr}_{N_k \to N_{k'}}$, $k \neq k'$ are computed as described in Sec. II-C
4) **update of prediction:** the two predictive mappings between the induced representations N_1, N_2 are updated based on the accuracy of the prediction $\mathrm{pr}_{N_1 \to N_2}$ and vice versa, see Sec. II-C
5) **calculation of predictability**: predictability measures are computed from $\mathrm{pr}_{N_{k'} \to N_k}, k \neq k'$, resulting in update indicators $\lambda_k(t) \equiv \lambda_k(\mathrm{pr}_{N_{k'} \to N_k}, t)$, see Sec.II-D
6) **update of projection:** the projections $I_k \to N_k$ are updated (see Sec. II-C). The learning rate of the adaptations is gated by the update indicators $\lambda_k(t)$ which effectively determine whether the projections $I_k \to N_k$ are adapted with the current pattern or not
7) **update of evaluation measures**, see Sec. II-E

Details for each step of a single PROPRE iteration are described later in this section; for the time being, we focus on the question how a complete training session consisting of many iterations is conducted.

B. Time course of PROPRE learning

PROPRE learning consists of numerous iterations which are executed sequentially during operation of the embedding system (online learning), be it real or simulated as in this contribution. However, although PROPRE learning is a fully online operation, it requires a suitable initialization from input stimuli, i.e, an induced representation where preliminary selectivities have already been formed. Therefore, training is conducted in two phases (see also Fig. 2): in phase I (initialization), the update indicators are kept at $\lambda_k \equiv 1$ which causes all inputs to be considered in the updating of the projection algorithm. During this phase, the values for SOM radius $r(t)$ and learning rate $\tilde{\epsilon}^{\mathrm{proj}}(t)$ are gradually decreased from initially large to very small stable values $\tilde{\epsilon}^{\mathrm{proj}}_{\infty}$, r_{∞}, after an initial phase of "global ordering" where large initial values are used (see [14]). This is done in such a way that stable values are reached and maintained well before the end of phase I.

In phase II, the update indicators $\lambda_k(t)$ are computed from the reference-based predictions $\mathrm{pr}_{N_{k'} \to N_k}$. In both phases, the learning rates of the projection steps are computed as: $\epsilon^{\mathrm{proj}}_k(t) = \lambda_k(t)\tilde{\epsilon}^{\mathrm{proj}}(t)$. Performance evaluations are taken at two distinct times during phase II: first, at its very start when SOM learning has converged and PROPRE learning has not yet started (*baseline condition*), and second, just before the end of phase II (*PROPRE condition*). This is done in order to compare the effect of PROPRE learning to a purely input-driven SOM approach (baseline condition -vs- PROPRE condition).

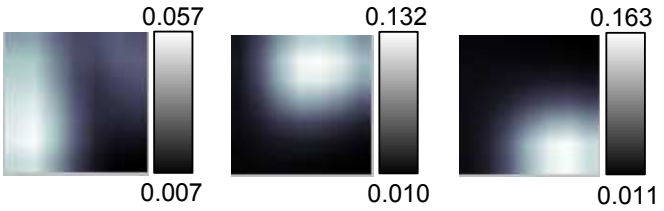

Fig. 3. Examples of reference-based predictions $\mathrm{pr}_{N_{k'}\to N_k}(\vec{y},t)$ indicating different predictabilities of the corresponding induced representations N_k. Please note the different value ranges indicated by the colorbars. The left image exhibits low predictability as compared to the middle and right images.

C. The projection and prediction steps

We choose to implement the prediction step by logistic regression (LR)[15] operating between the representations N_k, $N_{k'}$ by adapting the connection weights $w^{R-N}_{\vec{x}\vec{y}}$ that ideally achieve an error-free mapping. Using the weights, LR computes an estimate of N_k given $N_{k'}$, $\mathrm{pr}_{N_{k'}\to N_k}(\vec{y},t)$, which we term "reference-based prediction". The following training of the connection weights $w^{R-N}_{\vec{x}\vec{y}}$ is governed by a single parameter, the learning rate ϵ^{pred}.

The projection step is realized using the SOM algorithm proposed in [14], thus obtaining a topology-preserving projection from the possibly high-dimensional input representation I_k to the induced representation N_k. A SOM step consists of data transmission and weight adaptation. In our notation, the data transmission step reads $z^N(\vec{y},t) = \sum_{\vec{x}} w^{I-N}_{\vec{x}\vec{y}} z^I(\vec{x},t)$. Subsequently, weight adaptation and normalization is performed based on the results of the projection step, governed by the current learning rate $\epsilon^{\text{proj}}_k(t) = \lambda_k(\mathrm{pr}_{N_{k'}\to N_k},t)\tilde{\epsilon}^{\text{proj}}(t)$ and neighbourhood radius $r \equiv r(t)$:

$$w^{I-N}_{\vec{x}\vec{y}}(t+1) = \mathrm{norm}(w^{I_k-N_k}_{\vec{x}\vec{y}}(t) + \\ + \epsilon^{\text{proj}}_k(t)\ g^{\sigma}_{\vec{y}^*}(\vec{y})[w^{I_k-N_k}_{\vec{x}\vec{y}}(t) - z^{I_k}(\vec{x},t)]) \quad (1)$$

where $g^{r}_{\vec{y}^*}(\vec{y}) = \mathrm{e}^{-\frac{(\vec{y}-\vec{y}^*)^2}{2r(t)^2}}$, $\vec{y}^* = \mathrm{argmax}_{\vec{y}} z^N(\vec{y})$.

As proposed in the original SOM algorithm [14], neighborhood radius and learning rate are time-dependent. Initially they are kept constant at large values for $t < t_0$ ("global ordering"), whereas a decay constant ρ governs their slow decrease for $t > t_0$, and is always chosen such that learning rate and neighborhood radius approach their asymptotic stable values $\tilde{\epsilon}^{\text{proj}}_\infty, r_\infty$ at time t_1, after which they are kept constant:

$$\tilde{\epsilon}^{\text{proj}}(t) = \begin{cases} \tilde{\epsilon}^{\text{proj}}_0 & t < t_0 \\ \tilde{\epsilon}^{\text{proj}}_0 e^{-\rho t} & t_0 < t < t_1 \\ \tilde{\epsilon}^{\text{proj}}_\infty & t > t_1 \end{cases} \quad (2)$$

$$r(t) = \begin{cases} r_0 & t < t_0 \\ r_0 e^{-\rho t} & t_0 < t < t_1 \\ r_\infty & t > t_1 \end{cases}$$

In order to extract a simple interpretation from the output of the projection step z^{N_k}, and also to non-linearly suppress noise, we perform non-maxima suppression by simply setting $z^{N_k}(\vec{y},t) \to g^{r}_{\vec{y}^*}(\vec{y})$.

D. Quantification of predictability

A key concept is the quantification of predictability, which is used to decide whether the projection step should be updated with the current input or not. We define easy-to-compute, online *predictability measures* π_k and the *update indicators* λ_k as follows:

$$\mu^{\text{min/max}}_k(t) = (1-\alpha)\ {}^{\text{min}}_{\text{max}}\mathrm{pr}_{N_{k'}\to N_k}(\vec{x},t-1) + \\ + \alpha\ {}^{\text{min}}_{\text{max}} pr_{N_{k'}\to N_k}(\vec{x},t), k \neq k' \quad (3)$$

$$\pi_k(\mathrm{pr}_{N_{k'}\to N_k},t) = \max\ \mathrm{pr}_{N_{k'}\to N_k}(\vec{x},t) - \min\ \mathrm{pr}_{N_{k'}\to N_k}(\vec{x},t)$$

$$\lambda_k(\mathrm{pr}_{N_{k'}\to N_k},t) = \begin{cases} 1 & \pi_k(\mathrm{pr}_{N_{k'}\to N_k},t) > \kappa\times \\ & \times[\mu^{\text{max}}_k(t) - \mu^{\text{min}}_k(t)] \\ 0 & \text{otherwise} \end{cases}$$

Here $\mu^{\text{max}}_k, \mu^{\text{min}}_k$ are running averages (at a time scale α) over the minimal and maximal values of $\mathrm{pr}_{N_{k'}\to N_k}$. Put simply, the relative measure $\pi_k > 0$ checks whether the (spatial) variability of the current reference-based prediction $\mathrm{pr}_{N_{k'}\to N_k}$ is at least κ times the average (spatial) variability of previous predictions, with $\kappa > 0$ being a free parameter of the model. This is motivated by the fact that zero variability indicates no predictability since all elements of the induced representation are equally likely to be active, whereas large variability indicates predictability since some elements are much more likely to be active than others.

E. Evaluation measures

In order to quantify the effect of PROPRE learning on the induced representations N_k, we assume that patterns in the input representations I_k can be, for evaluation purposes, grouped into well-defined *classes* $c = 0, 1 \ldots$. Based on this assumption, we introduce a measure that determines, for each neuron in N_k, to which class of inputs (see, e.g., Fig. 4), if any, it preferentially responds to. For a given class c, we define in analogy to our notation $z^{N_k}(\vec{x},t)$ for neural activity in N_k, a *class preference measure* $p^{N_k}_c(\vec{x},t)$ as

$$m^{N_k}_c(\vec{x},t) = E(z^{N_k}(\vec{x},t)|c) \quad (4)$$

$$p^{N_k}_c(\vec{x},t) = \begin{cases} c & \text{if } m^{N_k}_c(\vec{x},t) > \xi m^{N_k}_k(\vec{x},t), k \neq c \\ -1 & \text{else} \end{cases} \quad (5)$$

The factor ξ on the right-hand side is an arbitrary threshold, determining how much the class preference measure for class c must exceed that of all other classes before we say that a neuron prefers class c. We always use a value of $\xi = 2$ in the presented work.

This measure can be used to determine, for a given class c, the number of neurons in N_k that respond preferentially to c:

$$\chi_k(t,c) = \frac{1}{c} \sum_{p^{N_k}_c(\vec{x},t)\neq -1} \quad (6)$$

In practice, the class-conditional expectation value of eqn. (4) is computed over finite time intervals T. If, for each $t' \in [t -$

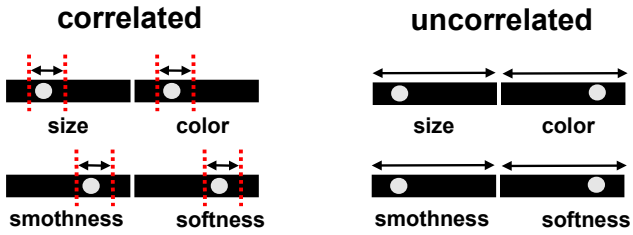

Fig. 4. Input statistics used in experiments. Shown are value ranges and realization examples for the artificial input representations I_k for $t \in [8000, 15000]$.

$T, t]$, class(t) represents the "real" input class, the expectation value can be approximated by

$$E(z^{N_k}(\vec{x},t)|c) = \frac{1}{N_c} \sum_{t'=t-T}^{t} z^{N_k}(\vec{x},t')\delta_{\text{class}(t'),c} \qquad (7)$$

where N_c indicates the occurrences of class c within the last T iterations, and δ_{ab} is the Kronecker symbol.

As we are interested to know how neurons in the induced representation N_k change their sensitivities, we will use $p_c^{N_k}(\vec{x},t)$ and $\chi_k(t,c)$ to quantify the effect of PROPRE learning.

III. Experiments

A. Synthetic input data

Our choice of synthetic stimuli is motivated primarily by simplicity as we above all want to show the fundamental feasibility of our algorithm. Imagine, as implemented in [13], a robot that can both see and touch objects, and extract simple descriptors from both modalities.

We define the artificial visual properties of "size" and "color", and the artificial haptic properties of smoothness and softness, both of which are drawn from a modality-specific probability distribution. Imagine furthermore that these properties are correlated for some objects, and uncorrelated for others, see Fig. 4. The task of our learning algorithm would now be to extract and represent, in each modality, those object properties that have correlations to the respective other modality. Reflecting this, we define two *classes* of inputs that occur with a certain probability: the *correlated class* ($p = 0.1$) and the *uncorrelated class* ($p = 0.9$) by specifying five random variables ν_k: ν_0 governing class occurrence, $\nu_{1,2}$ representing inputs to the haptic modality, and $\nu_{3,4}$ representing inputs to the visual modality. In accordance with what was said before, the $\nu_{1,2,3,4}$ are chosen such that there are statistical relations between them for $\nu_0 =$ "correlated" and no relations for the opposite class.

$$\begin{aligned} &\nu_0 \sim \mathcal{U}(0,1) \\ &\nu_{1,2} \sim \mathcal{U}(0.2,0.3), \nu_{3,4} \sim \mathcal{U}(0.7,0.8) \text{ if } \nu_0 \in [0.0,0.1] \\ &\nu_{1,2} \sim \mathcal{U}(0,1), \nu_{3,4} \sim \mathcal{U}(0,1) \text{ if } \nu_0 \in [0.1,1.0] \end{aligned} \qquad (8)$$

Here, $\mathcal{U}(a,b)$ denotes the uniform distribution in the interval $[a,b]$. After the ν_i are generated, $\nu_{1,2,3,4}$ are encoded into four corresponding two-dimensional blocks realizing the representations $I_{1,2,3,4}$. The size of the blocks is fixed such that a single number $\nu_i \in [0,1]$ can be comfortably encoded into each by creating a Gaussian $g_{\vec{y}^*}^{\sigma=1.5}(\vec{y})$ at $\vec{y}^* = (32{*}\nu_i(t), 3)^T$. Our choice of block dimensions of 32x7=214 elements is a compromise between considerations of simulation speed and representational accuracy. While larger blocks can represent more different values of the ν_i, the speed of the whole simulation suffers if the blocks get too large. Our choice gives a rather limited value resolution to $I_{1,2,3,4}$, but without adverse effects; larger blocks are always possible but do not affect the outcome of the simulation except for making it slower.

Since this is a simulation-based work, we have full knowledge of object classes even at training time. We stress, however, that in the general case, classes need to be known only for evaluation purposes: the learning algorithms do not, in any way, depend on this information.

B. Implementation and parameter values

In all experiments, we simulate for 15000 iterations using induced representations of 10×10 elements; phase I of PROPRE training (see Fig.2) is always conducted for 8000 pattern presentations, where radius and learning rate are decreased from initial values of $\tilde{\epsilon}^{\text{proj}}(t < t_0 = 800) = 0.8$ and $r(t < t_0 = 800) = 10/2$, until values of $\epsilon_\infty^{\text{proj}} = 0.02$ and $r_\infty = 10/6$ are reached approximately at $t = t_1 = 5000$. This parameter decrease, as described in Sec. II-C, is performed using decay constants of $\rho_\epsilon = 0.0007$ for the learning rate, and $\rho_r = 0.00025$ for the neighborhood radius.

For $t > t_1 = 5000$, which includes the last part of phase I and all of phase II, we use $\tilde{\epsilon}^{\text{proj}}(t) \equiv \epsilon_\infty^{\text{proj}}$ and $r(t) \equiv r_\infty$. The learning constant of the prediction step is always kept at $\epsilon_k^{\text{pred}} = 0.01$, and the time constant of the running average calculation of eqn. (3) is chosen as $\alpha = 0.002$. For the predictability threshold in eqn. (3), we use $\kappa = 1.1$, and weight vectors of projection and prediction are randomly initialized to small values in the interval $[-0.001, 0.001]$. The interval T used to measure the class averages $\chi_k(t,c)$ is set to $T = 600$ iterations.

All the experiments described here are implemented in Python and C using the OpenCV library to accelerate neural network execution and learning. On a standard off-the-shelf PC with a 2GHz multicore processor, it is possible to simulate roughly 20 iterations per second while running the whole simulation on a single CPU core[2].

C. Results

When conducting the simulation with the configuration, the synthetic inputs and the parameters indicated in the previous sections, there are two indicators of PROPRE's performance: the class preference measure $p_c^{N_k}(\vec{x},t)$ and the percentage if neurons that, in representation N_k, respond preferentially to class c, $\chi_k(t,c)$. The development of both these measures over the duration of the experiment is shown in Figs. 5 and 6. Both

[2]The Python/C code of the simulation will be made available on the site www.gepperth.net

Fig. 5. Development of the class preference measure $p_c^{N_k}(\vec{x}, t)$ for both induced representations. The diagrams show the induced representations N where neural selectivity for the correlated class is indicated by light gray, selectivity for the uncorrelated class by dark gray, and no clear selectivity by black. Left: start of PROPRE learning at $t = 8000$. Middle: $t = 12000$. right: end of PROPRE learning at $t = 15000$. Notable is a clear shift of selectivities from the uncorrelated but more frequent one to the correlated class.

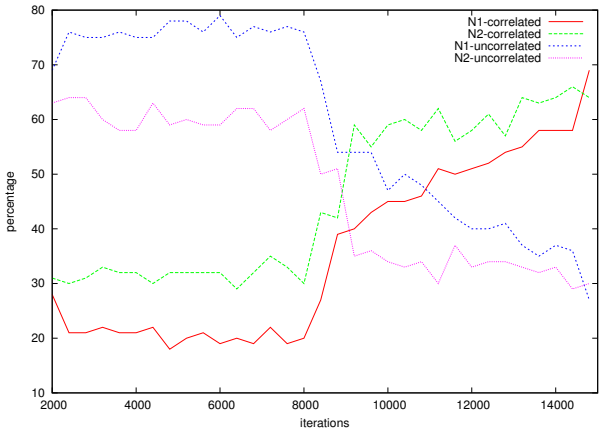

Fig. 6. Percentage of neurons in both induced representations that are sensitive to the correlated class (red and green curves) and to the uncorrelated class (violet and blue curves). The deterioration of selectivities to the uncorrelated class is notable right from the start of PROPRE learning at $t = 8000$.

indicators show that, with the start of PROPRE learning, the percentage of neurons (in both induced representations N_1 and N_2) that respond preferentially to the correlated class rises strongly at the expense of neurons that respond to the uncorrelated class of inputs. In Fig. 5 it may also be discerned that this growth process is topologically organized, i.e., newly "converted" neurons are in close vicinity to those who already prefer the correlated class. This is an effect of the topological representation property of the SOM algorithm we use for the projection steps. Lastly, we find that the process saturates after some time, and an equilibrium is attained where the percentage of neurons that prefer the correlated class does no longer grow. This can be seen in both Fig. 5 and Fig. 6: the indicators in both figures do not change too much any more between $t = 12000$ and $t = 15000$, especially when comparing to the rapid change between $t = 8000$ and $t = 9000$.

An interpretation and discussion of these findings will be given in the following section.

IV. Discussion

First of all, we will summarize and interpret the result stated in the previous section. Evidently, PROPRE learning causes a massive shift in neural stimulus preferences from the uncorrelated class of inputs to the correlated one (see Fig. 4).

Fig. 7. Visualization of projection weights for induced representation N_1 (top) and N_2 (bottom) at $t = 15000$. Both diagrams contain 20x10 blocks where two blocks at positions $(2i, j)$ and $(2i + 1, j)$ jointly indicate the stimulus preference of the neuron at grid position (i, j) in an induced representation. Please note the strong over-representation of the correlated class of stimuli in both set of weights, reflecting the stimulus preferences of neurons in both induced representations. This can be directly compared to Fig. 4. Equally, the strong topological ordering of induced representations is apparent.

The effect is actually very quick to manifest itself: as can be seen from fig. 6, about half of the observed effect takes places in the first 1000 iterations of the algorithm. This is remarkably efficient as the correlated class is more less frequent than the uncorrelated class. Nevertheless, PROPRE learning saturates at the end of the learning interval, leading to a new equilibrium between class-sensitive neurons.

It is furthermore interesting that the percentage of neurons that prefer the uncorrelated class diminishes, but does not go to zero: at the end of PROPRE learning, there is still a small amount of neurons that respond to the uncorrelated class (see Fig. 7). The reason for this is "cross-talk" between the classes: there is a small probability that the randomly generated stimulus values for the uncorrelated class in, e.g., I_1, will be in the range occupied by the correlated class, therefore triggering learning for the projection to N_2. When there are several or many classes of differently correlated inputs, such "cross-talk" should be a frequent phenomenon which is in fact beneficial: it prevents the induced representations from becoming degenerate by the complete suppression of selectivities to certain classes. Such a suppression would be undesirable since, when a class is suppressed completely, it cannot ever be recovered by PROPRE learning in case the input statistics change and make it worthy of representation.

Another important point to make is the autonomous control of learning by the PROPRE algorithm: as we are dealing with online learning which "sees" each input only once, the relative frequency of classes has an enormous influence on the outcome of learning. For the synthetic inputs used here, where the uncorrelated class is 9 times more probable than the correlated class, the correlated class would be strongly underrepresented without an instance that decides, depending on the intrinsic properties of each input, whether learning should be performed or not. By this autonomous control mechanism, PROPRE becomes largely independent of relative frequency considerations which is an important issue in real-world scenarios where the "interesting" relations are often

buried in a large amount of noise.

Moreover, as claimed in the introduction, it is obvious from Fig. 7 that PROPRE learning achieves, in addition to correlated concepts across modalities, a sufficient dissimilarity in single modalities as expressed by the stimulus preferences of neurons in both induced representations, N_1 and N_2. This is less visible for neurons sensitive to the correlated class because its stimuli lie in a very narrow value range as given by eqn. (8), but obvious for neurons sensitive to the uncorrelated class.

We conducted the experiments with two "sensory modalities", but it should be stressed that the basic PROPRE mechanism can trivially be extended to an arbitrary number of modalities by making the reference-based predictions to the k-th induced representation N_k the sum of predictions originating from all other modalities: $\sum_{k'} \mathrm{pr}_{N_{k'} \to N_k}$. Whether the detection of correlated patterns still works as in the presented case will have to be determined by experiments, but we are confident that at least those patterns which are correlated in more than two modalities can be discovered.

Lastly, one might be tempted to think that the synthetic inputs to the PROPRE algorithm that we used are overly simplistic to represent real-world computation. Such may be argued, but in fact this depends very much on the position in a processing hierarchy that is considered. For example, our previous work on environment perception in road traffic scenarios [16] suggests that simple representations are beneficial at the highest stages of unimodal processing hierarchies because they allow the learning of powerful object-scene relations using simple algorithms. Summarizing, although PROPRE does not require simple stimuli (in fact the dimensionality of our stimuli is rather high), we feel that the simple synthetic problem is not too far away from what one would encounter in a real robotic agent.

V. Summary and conclusion

We showed that, without reference to explicit supervision, an bi-laterally coupled neural learning algorithm can at the same time extract concepts that are dissimilar within their own modality, while being correlated across modalities. As we discussed in the introduction, such a property may turn out to be very valuable when stabilizing robotic perception by multimodal integration. The algorithm is stable in the sense that it reaches a non-degenerate equilibrium state, and it is efficient in the sense that it does not require an enormous amount of computational resources, and converges very quickly.

What we presented here was a proof of concept: in the future, we will aim to implement the PROPRE algorithm on a real robotic agent (see, e.g., [17]) and test its performance in such a setting to determine whether and what modifications must be made for robust real-world operation.

References

[1] P-Y Oudeyer. Developmental robotics. In NM Seel, editor, *Encyclopedia of the Sciences of Learning*, Springer Reference Series. Springer, 2011.

[2] S Kirstein, H Wersing, and E Körner. Towards autonomous bootstrapping for life-long learning categorization tasks. In *IJCNN*, pages 1–8, 2010.

[3] B Ridge, D Skocaj, and A Leonardis. A system for learning basic object affordances using a self-organizing map. In T Asfour and R Dillmann, editors, *Proceedings of the 1st Int. Conf. on Cognitive Systems*, 2008.

[4] B Ridge, D Skočaj, and A Leonardis. Self-supervised cross-modal online learning of basic object affordances for developmental robotic systems. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, USA, May 2010.

[5] BA Olshausen and DJ Fieldt. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

[6] A Hyvarinen. *Independent component analysis*. John Wiley & Sons., 2001.

[7] G Deco and ET Rolls. A neurodynamical cortical model of visual attention and invariant object recognition. *Vision Res*, 44(6):621–642, Mar 2004.

[8] MJ Tyler and MJ Spivey. Spoken language comprehension improves the efficiency of visual search. In *Proceedings of the 23rd Annual Conference of the Cognitive Science Society*, 2001.

[9] R Bauer and S Heinze. Contour integration in striate cortex: Classic cell responses or cooperative selection? *Experimental Brain Research*, 2002.

[10] MO Ernst and MS Banks. Humans integrate visual and haptic information in a statistically optimal fashion. *Nature*, 415(6870):429–433, Jan 2002.

[11] A Gepperth. Efficient online bootstrapping of sensory representations. *Neural Networks*, 2012. submitted.

[12] P König and N Krüger. Symbols as self-emergent entities in an optimization process of feature extraction and predictions. *Biological Cybernetics*, 94, 2006.

[13] O Kroemer, CH Lampert, and J Peters. Learning dynamic tactile sensing with robust vision-based training. *IEEE Transactions on Robotics*, 27(3), 2011.

[14] T Kohonen. Self-organized formation of topologically correct feature maps. *Biol. Cybernet.*, 43:59–69, 1982.

[15] CM Bishop. *Pattern recognition and machine learning*. Springer-Verlag, New York, 2006.

[16] A Gepperth, S Rebhan, S Hasler, and J Fritsch. Biased competition in visual processing hierarchies: A learning approach using multiple cues. *Cognitive Computation*, 3(1):146–166, 2011.

[17] I Jebari, S Bazeille, E Battesti, H Tekaya, M Klein, A Tapus, D Filliat, C Meyer, S Ieng, R Benosman, E Cizeron, J.-C Mamanna, and B Pothier. Multi-sensor semantic mapping and exploration of indoor environments. In *Proceedings of the 3rd International Conference on Technologies for Practical Robot Applications (TePRA)*, 2011.