Deep unsupervised network for multimodal perception, representation and classification

Alain Droniou, Serena Ivaldi, Olivier Sigaud

Deep Unsupervised Network for Multimodal Perception, Representation and Classification

Alain Droniou[a,b,*], Serena Ivaldi[c,d], Olivier Sigaud[a,b]

[a] *Sorbonne universités, UPMC Univ Paris 06, ISIR UMR7222, Paris, France*
[b] *CNRS, Institut des Systèmes Intelligents et de Robotique UMR7222, Paris, France*
[c] *Inria, Villers-lès-Nancy, F-54600, France*
[d] *Intelligent Autonomous Systems Lab, FB-Informatik, TU Darmstadt, Germany*

Abstract

In this paper, we tackle the problem of multimodal learning for autonomous robots. Autonomous robots interacting with humans in an evolving environment need the ability to acquire knowledge from their multiple perceptual channels in an unsupervised way. Most of the approaches in the literature exploit engineered methods to process each perceptual modality. In contrast, robots should be able to acquire their own features from the raw sensors, leveraging the information elicited by interaction with their environment: learning from their sensorimotor experience would result in a more efficient strategy in a life-long perspective. To this end, we propose an architecture based on deep networks, which is used by the humanoid robot iCub to learn a task from multiple perceptual modalities (proprioception, vision, audition). By structuring high-dimensional, multimodal information into a set of distinct sub-manifolds in a fully unsupervised way, it performs a substantial dimensionality reduction by providing both a symbolic representation of data and a fine discrimination between two similar stimuli. Moreover, the proposed network is able to exploit multimodal correlations to improve the representation of each modality alone.

Keywords: Unsupervised learning, Multimodal perception, Deep Learning, Developmental robotics

1. Introduction

A major issue for autonomous robots consists in extracting high-level knowledge from raw perception. This knowledge is critical to allow the robot to interact with the environment, realize specific tasks and learn useful skills.

*Corresponding author (Institut des Systèmes Intelligents et de Robotique, Université Pierre et Marie Curie, Pyramide - T55/65, CC 173 - 4 Place Jussieu, 75005 Paris, France, +33 1 44 27 63 87)

Email addresses: droniou@isir.upmc.fr (Alain Droniou), serena.ivaldi@inria.fr (Serena Ivaldi), olivier.sigaud@upmc.fr (Olivier Sigaud)

The usual approach to this problem consists in developing some dedicated feature extractors such as shape or color descriptors [10], which are fed to a dimensionality reduction technique such as bag-of-features [77]; their output is matched with some pre-existing, handcrafted symbolic knowledge such as ontologies, extracted from large databases or from the Internet [86], which can be used for cognitive planning [47]. In these paradigms, robot actions are used more to match robot sensory signal to high-level abstractions instead of being used to explore in the sensory space to automatically discover these abstractions, reducing the need for prior models of the environment.

These approaches are very diffused also in the developmental robotics community. Hierarchical processing is widely used, and each step of the pipeline can be performed by several algorithms picked from an abundant literature (e.g. [56, 83, 48]). As we discussed in [33], they are far from epigenetic principles at the base of developmental robotics, where knowledge is shaped from experience in an incremental way.

Six fundamental principles for the development of embodied intelligence were defined by [78]: multimodality, incremental development, physical interaction with the environment, exploration, social guidance and symbolic language acquisition. At least three of them are addressed in an inadequate way by the previously described approaches:

- Multimodality: in classical solutions, the features extraction techniques are often sensor specific. This requires either ad-hoc algorithms to fuse information from different sensors at an early stage, which is not always possible and requires much prior knowledge, or a late multimodal fusion which results in poor multimodal interactions. Sensor specific algorithms also limit the portability of the methods and their application to different robotics setups. Several works attribute to multimodality a central role for intelligence, memory [12], recall [13] and category extraction [14].

- Incremental development: unlike machine learning tasks where the whole training set is fixed at the start of learning, infants receive a constant flow of new stimuli during their development. In particular, encountered stimuli are strongly influenced by parental caregiving along with the development of the sensory system, and infants progressively learn to understand more complex stimuli. Usual robotics approaches, using a fixed set of features such as SIFT/SURF points or hand-crafted shape descriptors, do not allow the lowest perception level to evolve through learning. A limited set of features strongly limits the capacity of the robot to act in unconstrained environments, whereas a very fine perception at lowest level may prevent an efficient bootstrap of learning [1].

- Symbolic language acquisition: if the symbolic power of words is present in ontology-based approaches, [78] argues that

  > *[Children] initial progress in language learning is surely built on multimodal clusters and categories emergent in the infant's*

> *interactions in the world. Nonetheless, progress at first is tentative, slow, and fragile. For the 6 months or longer after the first word, children acquire subsequent words very slowly, and often seem to lose previously acquired ones. [Later], most children become very rapid word learners. [...] During this time, they seem to need only to hear a word used to label a single object to know the whole class of things to which the word refers. [...] The evidence from both experimental studies and computational models indicates that children learn these regularities as they slowly learn their first words and that this learning then* creates *their ability to learn words in one trial.*

The acquisition of first words is critical for subsequent development, and the "symbolic babbling" experienced by infants during this period seems to play an important role. It is worth noting that the first words acquisition is concomitant with the acquisition of the very concept of *word*, i.e. the capacity to segment and categorize a continuous auditory flow. This highlights the importance of the "multimodal clusters and categories emergent in the infant's interactions in the world" put forward by [78] (see also [15] for a discussion on the misleading way of using symbolic words as labels for developmental learning). It is not sufficient to provide pre-determined symbolic labels to a learning agent, but learning this representation may be a key for the exponential growth of knowledge observed during infant development. This may be partly due to top-down interactions which can guide low-level processings towards relevant features from the environment [21].

The issue of linking raw perception to high level concepts has been extensively addressed by Goldstone & Barsalou in [25], who argue that a common representational system must underlie both perception and conception. In the perceptual symbol system framework [2], concepts consist of patterns of neural activity corresponding to some selected aspects of perceptual experience. The major property is to empower simulation competences which can be used to combine and manipulate symbols and give raise to creativity and inference [3]. This feature fits naturally within the biological framework of Convergence-Divergence Zones (see [55] for a recent review) which states that multimodality is at the core of intelligence, and that high level representations are not copies of perceptual stimuli, but only minimal records needed to drive the reconstruction of these stimuli in early cortices.

These ideas have been successfully integrated into computational models by several authors (e.g. [36, 58, 57]), yet using simplified stimuli: for instance, in [57] auditory flow was pre-processed to provide a collection of explicit words. Multimodality is often considered in a language learning scheme (e.g. [89, 82]), where the language grounding process leverages the input from other modalities. However, these works generally assume high level capacities for language computation, such as built-in speech recognition, word extraction, etc. which are in

contradiction with the previously described symbolic language acquisition point. The presented work does not target explicitly the language learning issue, as we develop a generic framework which makes no assumption about the nature of input modalities. Thus, it can be used to relate the auditory modality to other sensorimotor perceptions, which is a first step towards fully autonomous language grounding.

Regarding the above principles, deep neural networks constitute an attractive candidate for developmental and cognitive robotics. They are indeed able to learn a hierarchical representation of data in a fully unsupervised way [29, 45], and have natural generative properties at the core of perceptual symbol systems. Mainly applied to images [45, 39, 11], deep learning techniques have been extended without much modifications to other modalities, such as text [79, 28], sound [27] or limb trajectories [81]. They are also successfully applied to multimodal input [64].

A temptation could be to use such algorithms as a particular stage of a perception pipeline, as aforementioned. The learned representations are actually very relevant in order to match corresponding stimuli with symbolic concepts [29, 75]. However, the symbolic language acquisition point reminds us that not only features extraction is important for knowledge development, but also "symbolic babbling". Providing explicit labels constrains this exploration and is very different from the unsupervised (though socially guided) language acquisition process [15]. Unfortunately, explicit labels are usually required in deep learning literature to build this symbolic mapping.

In this work, we propose an extension of deep neural networks, able to structure a high dimensional, possibly multimodal input in a completely unsupervised way, which produces a representation of data as classes along with a coordinate system representing admissible variations in each class. Following the manifold hypothesis for classification [61], this architecture learns to identify and to represent different sub-manifolds in a high dimensional input. We show the effectiveness of our solution on a multimodal dataset acquired on the humanoid robot iCub [62].

The paper is organized as follows. We first present the deep learning framework, and related work on features learning and multimodal learning in Section 2. In Section 3, we present the proposed architecture. We perform evaluation experiments in Section 4 and discuss the results in Section 5.

2. Related Work

In this section, we first present the deep learning paradigm. Then, we review related work on features learning and multimodal fusion.

2.1. Deep networks

It is well-known [30] that a neural network with a single hidden layer can approximate almost any function with arbitrary precision[1]. However, the number of required hidden units can grow exponentially when the input dimensionality increases, when the function becomes more irregular or when the desired precision increases. On the other hand, using multiple levels of hidden layers decreases the number of units required to approximate a large set of functions (from exponential to linear complexity [17]) through a factorization of representations, but are difficult to train because of the vanishing gradient problem [6, 24] and prone to fall in very bad local optima [18].

To overcome this issue, [29] proposed to pre-train each layer to learn a good representation of its input. Different possible pre-training algorithms are surveyed in [5]. Since our work involves only auto-encoders, we restrict ourselves to the description of this family of algorithms. A general review on deep networks is proposed in [4]. We focus in this section on theoretical aspects of deep networks, while we describe different applications later on.

To learn a good representations of their input, auto-encoders are trained to minimize the reconstruction error of input data. Given a visible input $\mathbf{v}$ and a hidden layer $\mathbf{h}$, they learn an encoding[2]

$$\mathbf{h} = \sigma(W_1\mathbf{v} + \mathbf{b_h}) \tag{1}$$

where $\mathbf{b_h}$ is a constant bias term and σ is usually a non-linear activation function such as a sigmoid ($\sigma(x) = \frac{1}{1+\exp(-x)}$) or a rectified linear unit ($\sigma(x) = x$ if $x > 0$, $\sigma(x) = 0$ otherwise). This encoding is then decoded to reconstruct the input

$$\hat{\mathbf{v}} = \sigma(W_2\mathbf{h} + \mathbf{b_v}). \tag{2}$$

Auto-encoders are trained by backpropagating the reconstruction error (e.g. $||\mathbf{v} - \hat{\mathbf{v}}||_2^2$, or the cross-entropy in the binary input case). Several regularization techniques have been proposed, among which:

- Weight tying: weights for encoding and decoding are tied, i.e. $W_2 = W_1^\top$.
- Denoising auto-encoders [85]: the input is first corrupted by noise (e.g. randomly setting some input units to zero), and the reconstruction error is measured either compared to the non-corrupted input, or to the same input corrupted with independent noise.
- Sparse auto-encoders [44]: a sparsity constraint is added to the hidden layer activity.
- Contractive auto-encoders [74]: a penalty cost is added to penalize the Jacobian of the hidden layer w.r.t. the input. This aims to contract

[1] The result is established for any Borel-measurable function on a finite dimensional space to another in [30].

[2] In the following, we denote vectors with bold letters, and matrices with capital letters.

the learned representation along the relevant dimensions to represent the input. Higher-order contractive auto-encoders [73] also penalize higher-order derivatives.

After the pre-training stage where each layer is trained separately, a global fine-tuning can be performed, according to either a similar reconstruction error cost on the input dataset or a task specific cost function (e.g. [76]). Another possibility is to stack on top of the network a classical supervised algorithm (e.g. a support vector machine) which takes as input the activity of the top layer of the deep network [80].

2.2. Feature learning

Many features learning techniques have been developed. They usually aim at learning atoms providing an efficient coding of the dataset. The techniques mainly differ by the constraints imposed on the atoms. Principal Components Analysis (PCA) corresponds to a linear coding of input, whereas forcing each data to be represented by only one atom leads to clustering algorithms. Sparse coding techniques [66, 44] code each data by a small number of atoms, while other constraints such as non negativity of coding is used for instance by non-negative matrix factorization [43]. Other non-linear techniques [9] and incremental dimensionality reduction techniques have also been developed [38, 41, 90].

2.2.1. Dimensionality reduction with deep networks

Deep networks have been widely applied as a dimensionality reduction technique. In fact, one nice property of deep architectures is their capacity to learn hierarchical features of the input. For instance, [45] shows that a deep network trained on images of faces, cars, airplanes and motorbikes learns Gabor-like features at the lowest level, which are progressively combined into more abstract representations towards global prototypes of each class at highest levels.

Such an approach is useful to efficiently learn manifolds and reduce the dimensionality of the data. In [29], the authors showed the greater capacity of deep architectures to extract meaningful dimensions from large datasets, compared with classical dimensionality reduction techniques such as PCA.

Some architectures, such as the above mentioned contractive auto-encoders, explicitly encourage the network to learn along the most meaningful dimensions of the dataset. In [72], the authors use contractive auto-encoders to extract an "atlas" of the tangent planes of the input data manifolds through the singular values of the Jacobian $J = \frac{\partial h}{\partial v}$. These tangent planes are then used to reduce the distance between neighbor points on the manifold. This can be done explicitly for a k-nearest neighbors approach, by defining an appropriate distance between two points, or implicitly by adding a penalty term for a fine-tuning of the network to shrink the representation of input data along these planes.

In [70], the authors propose a network able to disentangle factors of variation, thus allowing to traverse the manifold by fixing some factors and varying the others. However, since the network uses binary units, the manifold traversal actually corresponds to a walk on the vertices of an hypercube. This requires to

use several units for each factor of variation, and does not represent a continuous parametrization of the manifold (or necessitates an exponential number of units to discretize the parametrization with an increasing precision). Moreover, it does not classify data according to different sub-manifolds, but rather learn a unique global manifold for the whole dataset. It is unclear whether using a different number of units for each latent factor of variation could help to have a more or less clear discretization of some factors compared to the others, which could be considered as different classes.

2.2.2. Clustering

Clustering algorithms are attractive regarding the categorical perception effect [26] and the symbolic language acquisition [78]. Most of the proposed algorithms [34] are variants of the k-means algorithm [49] or hierarchical clustering [87]. Other algorithms such as spectral clustering [63] exhibit good performances, but rely only on relationships between points and do not provide features which would be characteristic of each class. Kohonen-like algorithms also rely explicitly on a neighborhood relationship between clusters [37]. The critical point for all these algorithms is the definition of a suitable metric depending on the addressed task. Usually, on high dimensional, redundant data (such as images), simple metrics such as euclidean distance do not capture efficiently the similarity between two data points. To avoid complex metrics, a possibility is to learn a representation of data which extracts discriminative features on which simple distances can be more relevant than on raw data. This argues in favor of the use of another feature extraction technique providing a compressed representation of data. A popular approach is the use of bag-of-features [65], but other dimensionality reduction techniques can be used.

Dimensionality reduction properties of deep architectures have been used for classification. They usually consist of an unsupervisedly pre-trained deep network on top of which a layer dedicated to classification is added, e.g. a multi-layer perceptron or a support vector machine [80]. A fine-tuning of the whole network is then performed to optimize a classification loss function depending on the top layer. Smarter algorithms can be used to obtain hierarchical classification and one-shot learning of new concepts [75]. In [54], the authors use a gated network to learn "style" features from a labeled dataset. This network learns some features shared among different classes and learns how each class is defined as a particular combination of some of these features. The authors show that sharing features improves the performance of the network compared to a non-sharing approach. This sharing is a result of the factorization of gated connections, as introduced in [53]. The presence or absence of each learned feature is indicated by boolean variables in a hidden layer. For k hidden units, the authors show that this model is equivalent to a mixture of 2^k logistic classifiers with shared weights. This makes the network efficient for classification, but requires supervised training. To our knowledge, using deep networks for unsupervised classification has not been studied extensively.

Using a dimensionality reduction technique as an input for a clustering algorithm rely on the underlying hypothesis that different concepts are defined

by distinct sub-manifolds in the raw input space. This has been hypothesized in [9, 61, 74]:

- *Unsupervised manifold hypothesis*: real world data in high dimensional spaces is likely to concentrate in the vicinity of non-linear sub-manifolds of much lower dimensionality.
- *Manifold hypothesis for classification*: data from different classes is likely to concentrate along different sub-manifolds, separated by low density regions of the input space.

The validity of these hypotheses is still an open question, but it seems reasonable to consider that natural data depends on much fewer variables than the dimensionality of sensors. For instance, the number of muscles whose activities define the appearance of a face (about 50), is much smaller than the number of cones on the retina which are activated by looking at this face (few millions), or the number of pixels on a good quality picture. Moreover, these hypotheses provide a natural definition of categories and have been shown to efficiently reduce the complexity of algorithms [61].

2.3. Multimodal fusion

There are many evidences that the brain is strongly influenced by multimodal sensations: in the McGurk effect [52] for instance, the perception of a syllable differs depending on the presence of visual only, auditory only, or both visual and auditory stimuli. The rubber hand effect [7] is another example, where a coupling between visual and tactile stimuli modifies the perceived body. These examples show that multimodal fusion is not only a "high level" processing, but that processings of different modalities are intertwined. These cross-modal interactions are supported by anatomical evidences [20, 19] and lesion-based studies [12, 13, 14]. They are at the core of the Convergence-Divergence Zones framework [13, 55]. This theory posits the existence of association cortices receiving inputs from different sensorimotor cortices. These association cortices save the minimal amount of information needed to regenerate the neural activity pattern corresponding to different perceptual experiences *inside the sensorimotor cortices*. Thus, a sensorimotor cortex from one modality can stimulate an imaginary perception in another sensorimotor cortex through the activation of one or more association cortices.

Some works have addressed the issue of multimodal learning. In [50], the authors use non-negative matrix factorization to learn a joint representation of gestures and spoken words. They show that the learned representations can be used to retrieve one modality given the other (e.g. retrieve the gesture corresponding to a spoken sentence) and that they acquire a semantic content through a high mutual information with respect to semantic labels. The learned representations can also be used to efficiently classify data. However, this classification is done by feeding the algorithm with ground-truth labels.

In [64], the authors use a deep network to learn a joint representation of visual and auditory input corresponding to spoken syllables. The network is

also able to retrieve one modality given the other, but the authors also show that it can reproduce the McGurk effect: after training the network on spoken syllables, they train a supervised classifier to distinguish between syllables `ba`, `ga` and `da`. Then, they show that when a visual `ga` is presented with an auditory `ba`, the network classifies it as a `da` most of the time, as observed in humans.

In [58, 57, 46], a Hebbian-like learning rule is used to associate the activation of several self-organizing maps, as suggested by the Convergence-Divergence Zones framework. This association can then be used to influence the self-organization of monomodal maps [46]. The use of self-organizing maps to learn multimodal associations has been studied by several authors [88, 68, 35, 71, 84, 40]. If these architectures are good at learning crossmodal associations, they suffer from the curse of dimensionality: projecting high dimensional data to two or three dimensions by preserving local topology (on which usually rely crossmodal associations) is difficult. A hierarchy of self-organizing maps is used in [40] to reduce the dimensionality of input, using the coordinates of the most active unit of each monomodal map as input of the multimodal map. In this case, two similar monomodal stimuli have to be represented by two units close enough to each other, which is problematic when the underlying physical phenomenon is high dimensional (for instance the visual appearance of the hand which is determined by 9 degrees of freedom for the iCub humanoid robot they use). The fact that they use only three different poses of the hand (from the *rock, paper, scissors* game) with a simple movement interpolation between them is probably an important factor for the success of their approach with a 3-dimensional self-organizing map.

Similarly, in [15, 16] De Sa & Ballard derive an Hebbian-like learning rule from a *disagreement minimization* framework. In their scheme, each modality is coded by a codebook obtained through a SOM-like algorithm, and all codebook spaces (for each modality) are further clustered into several categories by a linear classifier. Then, the winning unit for one modality is used as a label for the other modality to train the classifier. This results in a joint learning of multimodal categories. The authors achieve good results by applying this algorithm to audio-visual clustering of syllables using lip movements and spoken sounds. In particular, they show that using multimodal input increases the correct classification rate for each modality alone. However, based on self-organizing maps, this algorithm is prone to the same weaknesses as above models. In particular, representing each modality by a single winning unit before the multimodal fusion necessitates several iterations of codebook learning and multimodal classification to achieve good performances.

In [59], Nakamura et al. learn a joint probability distribution of features from different modalities. They show that by adding more modalities, the algorithm is able to learn a clustering closer from the human labeling. However, they use hand-crafted features with a relatively small number of dimensions. They extend their work in [60] to cope with complex categories structures, for instance when an object belongs to several categories such as *toy* and *soft*. To do this, they run their clustering algorithm several times and select relevant clusterings based on correlations with words utterances from the verbal description of objects. The

use of these words utterances can be seen as a supervision signal. Moreover they do not address the difference between "sharp" categories and continuous traits (an object is or is not a *toy*, but can be more or less *soft*).

2.4. Summary

Many algorithms have been proposed to process high-dimensional data. On the one hand, dimensionality reduction techniques provide a compressed representation of data in an unsupervised way but do not usually provide a symbolic representation suitable for reasoning and planning, while classification algorithms require some supervision. On the other hand, clustering techniques, which fail in high-dimensional spaces when used with simple distances, can be used on top of a dimensionality reduction technique. However, stacking two different algorithms makes it difficult to build synergies between them (how clustering can influence dimensionality reduction along relevant dimensions, and *vice versa*) and constrains top-down interactions.

Neural networks are attractive: deep networks are good at dimensionality reduction, and clustering can be achieved through competitive processes (e.g. self-organizing maps). Moreover, since standard neural networks make few assumptions about the nature of their input signal, different modalities can be fed to a same layer of units, endowing the network with natural multimodal properties.

3. Architecture

This section introduces the proposed architecture. First, a monomodal version is described. Then, the generalization of the architecture to an arbitrary number of modalities is explained.

We choose the auto-encoder paradigm to cluster high-dimensional data. This paradigm is used both for reducing the dimensionality of data (in a standard auto-encoding approach) and for clustering, using a softmax activation function to introduce competition between units. To increase the representational power of the architecture, we extend the network with an additional auto-encoder layer whose weights depend on the learned clustering. This allows our architecture to learn several manifolds in parallel and deeply intricate clustering and dimensionality reduction.

3.1. Monomodal network

The proposed network is outlined in Fig. 1. It first identifies clusters in the input data using a softmax layer. Then, for each cluster, the network tries to learn a representation of the corresponding manifold. Based on a global loss function, the network refines its clustering criteria along with the manifold representations. The input layer of the architecture can be either raw data or top level output of a deep architecture, for instance when working on images.

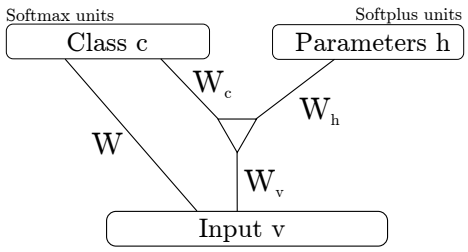

Figure 1: Unsupervised classification network. The network aims at clustering input data at the softmax layer using the W matrix. Based on this clustering, a gated network learns to represent the underlying manifold. Gated connections make it possible to share features W_{v} and W_{h} between all classes and reduce the number of parameters to learn. The input is intended to be the output of a standard deep network to handle high-dimensional, raw perception.

3.1.1. Manifold learning

As for auto-encoders, our goal is to learn to reconstruct the input of the network[3] $\hat{\mathbf{v}} = \sigma(W^T\mathbf{h})$ where the hidden layer $\mathbf{h}$ of standard auto-encoders corresponds in our case to the parameters layer. One way to create meaningful representations is to create a bottleneck, by setting the number of hidden units much smaller than the number of visible ones. In a supervised scheme, we could attribute a different weight matrix W to each class, and the hidden layer would learn a different parametrization for each class. In this case, we would have $\hat{\mathbf{v}} = \sigma(W_{c_i}^T\mathbf{h})$ where c_i denotes the class of $\mathbf{v}$.

In an unsupervised approach, we use a simple softmax layer to cluster input data. The estimated class is given by $c_i = \sigma_{max}^i(W\mathbf{v}) = \frac{\exp((W\mathbf{v})_i)}{\sum_j \exp((W\mathbf{v})_j)}$. We denote by $\mathbf{c}$ the vector whose i-th value c_i can be seen as the probability that sample $\mathbf{v}$ belongs to class i. Then, one way to build different weight matrices for each class is $W = \sum_i c_i W_i$: the class vector "gates" the connection matrix W. However, this approach requires to learn $n_c \times n_v \times n_h$ parameters, where n_v, n_h and n_c are respectively the number of visible units, hidden units and classes,.

By factorizing this gating architecture [53], we can reduce the number of learned parameters. In this scheme, the hidden layer becomes

$$\mathbf{h} = \sigma_+(W_h((W_c\mathbf{c}) * (W_v\mathbf{v}))) \tag{3}$$

and the reconstruction of the input is given by

$$\hat{\mathbf{v}} = \sigma(W_v^T((W_c\mathbf{c}) * (W_h^T\mathbf{h}))) \tag{4}$$

where $*$ denotes the element-wise product, σ_+ the softplus function ($\sigma_+(x) = \log(1 + \exp(x))$) and σ is the activation function of the input layer, which can

[3]Biases introduced in Section 2.1 (Eqs. 1 and 2) are omitted in the following equations for the sake of clarity.

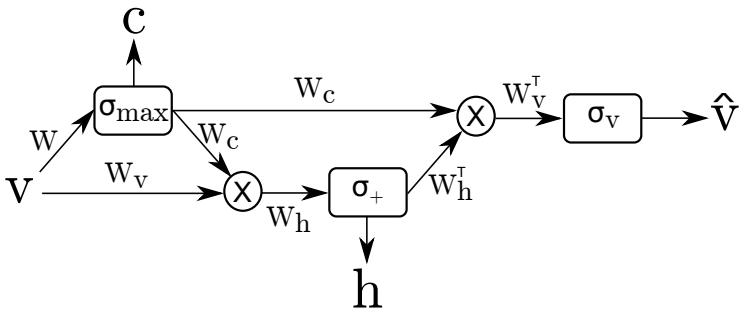

Figure 2: Information flow in the network to compute the class $\mathbf{c}$, the parameters $\mathbf{h}$ and reconstruction $\hat{\mathbf{v}}$ of an input $\mathbf{v}$.

depend on the input data. Matrices W_c and W_v project the class layer and the visible layer on a "factor" layer, which is projected on the hidden layer by W_h (Fig. 2). If we note n_f the number of factor units, we thus have $(n_h+n_c+n_v)\times n_f$ parameters. In the case where n_f is in the same range as n_v or n_h, this greatly reduces the number of parameters to learn, thus making learning easier.

Learning smooth manifolds, using softplus units at the hidden layer, differentiates our work from the supervised classification algorithm of [54], whose architecture looks similar. However, this previous work learns features at the hidden layer which are "on" or "off" depending on the input. This greatly simplifies the expression of $p(\mathbf{c}|\mathbf{v})$ and makes the exact computation of the normalization term tractable, but cannot be used to learn a continuous manifold representation. This also differentiates our work from [70] since we learn a continuous parametrization of several manifolds, instead of a discretized parametrization of one larger manifold.

The reconstruction $\hat{\mathbf{v}}$ of an input $\mathbf{v}$ is eventually given by:

$$\hat{\mathbf{v}} = \sigma_v(W_v^T((W_c\sigma_{max}(W\mathbf{v})) * (W_h^T\sigma_+(W_h((W_c\sigma_{max}(W\mathbf{v})) * (W_v\mathbf{v})))))). \quad (5)$$

Then, a gradient descent on the global reconstruction error tends to learn a correct representation of the classes (matrices W_h, W_c and W_f) and fine-tunes the classification matrix W to focus on relevant attributes of each class. If the input layer is the output of another network, the gradient descent can also propagate in this other network to fine-tune the input according to the classification task.

3.1.2. Clustering regularization

Given the softmax classification layer, the algorithm could use a distributed representation of input over all units. To force the algorithm to build sharp clusters, we add a Gaussian noise on the activation of the classification layer, before applying the softmax function. Thus the network cannot rely on fine combinations of mid-activated units to represent a variety of different inputs. The influence of the noise is detailed in Section 4.

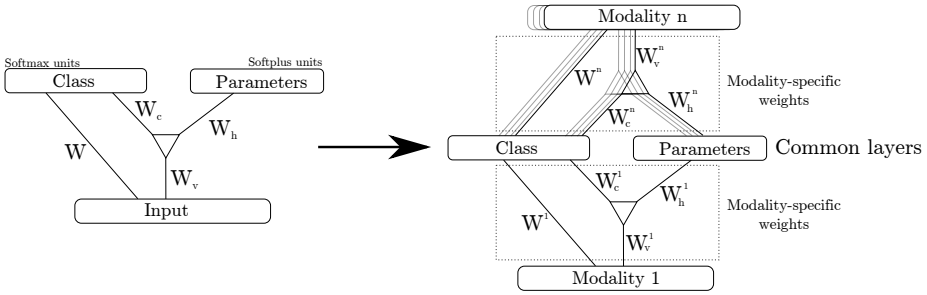

Figure 3: The architecture generalizes to an arbitrary number of modalities by duplicating the input layers while sharing the same softmax and softplus layers (*Modality 1* has been separated for clarity only, all modalities are equivalent in the network). This constrains the network to learn a joint representation of different modalities, which have to be clustered into a single space (one softmax unit active at one time), but the network can allocate different softplus units to one or several modalities, according to their correlations and the number of available units.

3.2. Multimodal generalization

The generalization of the monomodal network to several modalities is straightforward: the softmax and softplus layers are shared between all modalities, as illustrated in Fig.3. Thus, as in [59], the network has to learn a joint representation of different modalities. Since the softmax layer tends to make only one unit active at the time, it forces the network to associate input from different modalities to the same "concept". However, different units of the softplus layer can specialize on the representation of details from a single modality. This gives more flexibility to the network: for instance a visual input of a cat and an auditory input corresponding to the word "cat" should be associated to each other in the same cluster, but there is no reason to deduce the exact pronunciation of the word from the visual input. On the contrary, sharing the softplus units between different modalities makes sense for instance in a writing task: a good representation of the proprioceptive trajectory can be deduced from the visual appearance of the written letter, and *vice versa*. Sharing the whole layer between all modalities lets the network allocate the resources depending on the amount of mutual information between modalities.

3.2.1. Multimodal regularization

When one modality is much more noisy or irregular than another one, the network can learn to classify the input based on solely one modality. The learned network is then unable to exploit a partial input where this modality is absent. To avoid such a behavior, a solution consists in assigning a random weight to each modality for each class and each training sample. For two modalities, the softmax layer activation becomes:

$$\mathbf{c} = \sigma_{max}\left(\Omega * (W_1\mathbf{v_1}) + (1-\Omega) * (W_2\mathbf{v_2}) + \eta\right) \tag{6}$$

where Ω is a matrix of independent uniform random numbers between 0 and 1, $\mathbf{v_1}$ and $\mathbf{v_2}$ correspond to the visible inputs of both modalities, W_1 and W_2 are their corresponding classification matrices, and η is the regularization noise (see Section 3.1.2).

By randomly weighting modalities, the network cannot rely anymore on a single modality (whose weight may be null for some samples). Moreover, being trained on the reconstruction cost of both modalities, the network is forced to behave identically whatever the weight of each modality is, in particular when one modality is quashed by a null weight, or when both modalities have the same weight (of 0.5). Therefore, the network has to learn classification matrices W_1 and W_2 which produce similar projections of both modalities as input for the classification layer.

Extending to n modalities is straightforward, taking n random matrices whose sum is normalized such that each term is equal to one.

4. Experiments

In this section, we illustrate the properties of the architecture. First, we carry on a comprehensive analysis of the monomodal network to assess what is the influence of the parameters and which manifolds are learned. Then, we illustrate the performance of the multimodal network with two and three modalities. We study the influence of several modalities on the performance, and show how the network behaves when partial information is fed as input.

4.1. Training the network

For all experiments, we consider a network in which each modality is represented through a single-layer auto-encoder, on top of which the proposed network is added (Fig. 4). The whole network is trained incrementally: first, we train auto-encoders of each modality for 3000 time steps. In a second stage, we train the proposed network for 3000 additional time steps. Finally, the whole network in Fig. 4 is trained to reconstruct its raw input for another 4000 time steps[4]. We use a denoising auto-encoder approach [85], in which each modality is corrupted by a zero-masking noise of 30% (30% randomly chosen input is set to 0) and the reconstruction error is measured with respect to the non-corrupted input.

Unless otherwise stated, we use 10 classes (softmax units) and 2 parameters (softplus units) and the number of factors (n_f, Section 3.1.1) is equal to the hidden layer size for each modality.

[4]The learning rate was fixed at 0.001 for weight matrices and 0.0001 for biases, with a momentum of 0.9, such that no instability was visible on the learning curve. No attempt has been made to optimize the duration of each stage, the number of time steps was chosen empirically on few runs such that the reconstruction cost reached a plateau at the end of each stage (less than 5% improvement on 1000 time steps).

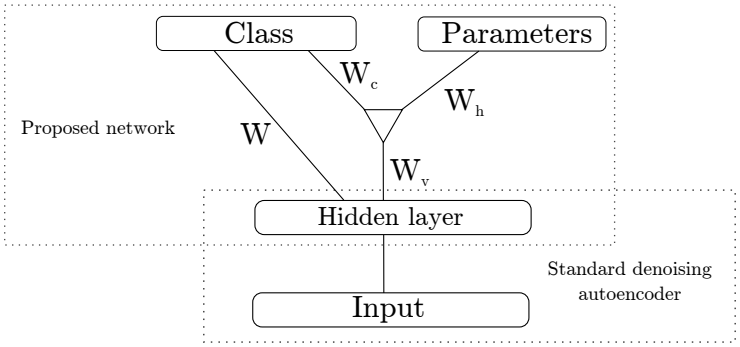

Figure 4: Architecture of the network used for the experiments (only one modality is represented). The input is first encoded by a standard denoising auto-encoder, whose output is used as input of the network presented in previous sections.

4.2. Classifying MNIST

To study the influence of different factors on the network, we use the MNIST dataset, composed of 20x20 pictures of handwritten digits, which allows us to perform an easy analysis of what is learned by the network: from a human point of view, there are ten natural classes, and the structure of the pictures is simple enough to be able to interpret the learned features. For all experiments, we repeat each setting 10 times. The hidden layer contains 100 units.

First, we train the network on a dataset composed of 100 samples of each digit and we test it on a dataset composed of 1000 samples of each digit. We study the influence of the amount of noise added to the softmax layer (see Section 3.1.2). We add a Gaussian noise centered on 0 and vary its standard deviation (referred as the "amount of noise" in the following). Figure 5 plots the mean activation of the most active unit for each sample, and the classification performance measured with the adjusted Rand index [31] is represented in Fig. 6.

Figures 5 and 6 show an optimal trade-off between classification sharpness and accuracy for an amount of noise of 2. We use this value for the following experiments.

The number of softmax units can be considered as an important *a priori* knowledge embedded in the architecture. Thus, we investigate the behavior of the network for different numbers of units. Figure 7 shows the influence of the number of softmax units on classification performance. The network is not too sensitive to this parameter: the classification performance with 10 units (which corresponds to the expected number of clusters in the MNIST dataset) is similar to the performance of the network with 100 units. This differentiates the network from the simple k-means algorithm, which reaches a close performance with 10 clusters, but whose performance decreases significantly when the number of clusters increases.

Figure 8 actually shows the ability of the network to use a number of clusters smaller than the number of provided units. In our experiment on the MNIST dataset, it seems to converge towards about 25 clusters.

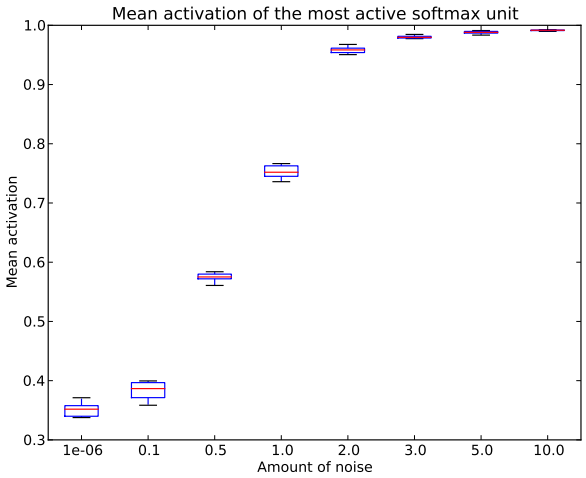

Figure 5: Mean activation of the most active softmax unit. Each box corresponds to 10 runs. For a noise greater than 2, the network classifies sharply: the mean activation is above 0.95. The amount of noise corresponds to the standard deviation of the Gaussian noise.

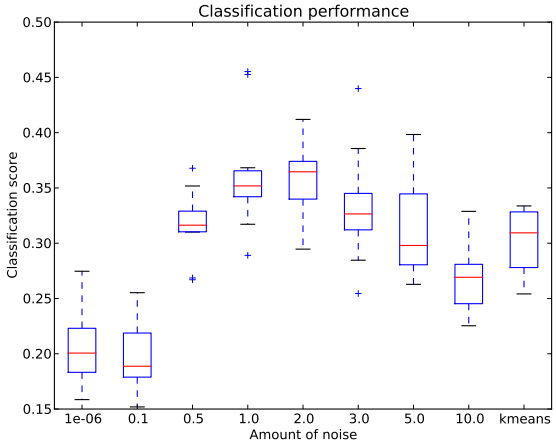

Figure 6: Classification score measured with the adjusted Rand index. A score of 1 indicates a perfect matching with ground truth label whereas a score of 0 corresponds to a random classification. For each sample, the predicted label corresponds to the most active softmax unit. The baseline corresponds to 10 runs of a k-means algorithm, initialized with 10 random samples from the dataset.

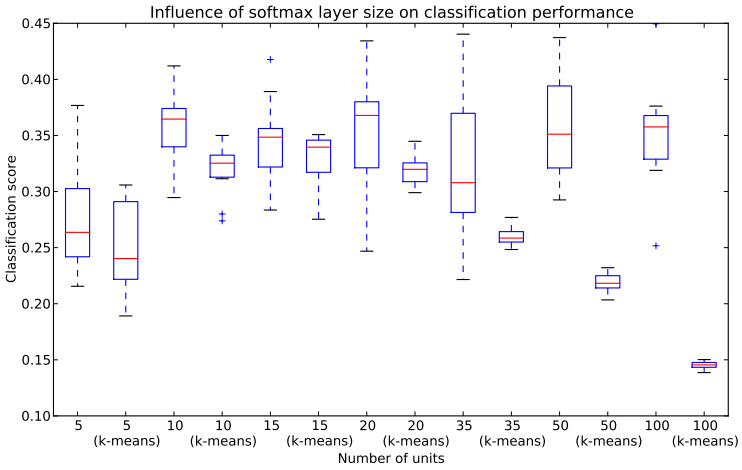

Figure 7: Influence of the number of softmax units on classification performance. For a number of clusters different from ground truth, the performance of the proposed architecture is more regular than for k-means.

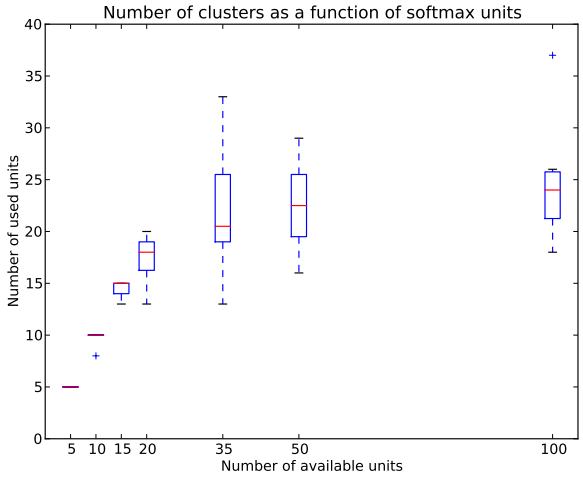

Figure 8: Number of softmax units used by the network depending on the number of available units. A unit is considered to be used if the activation of the unit is higher than the activation of the others for at least one input from the dataset.

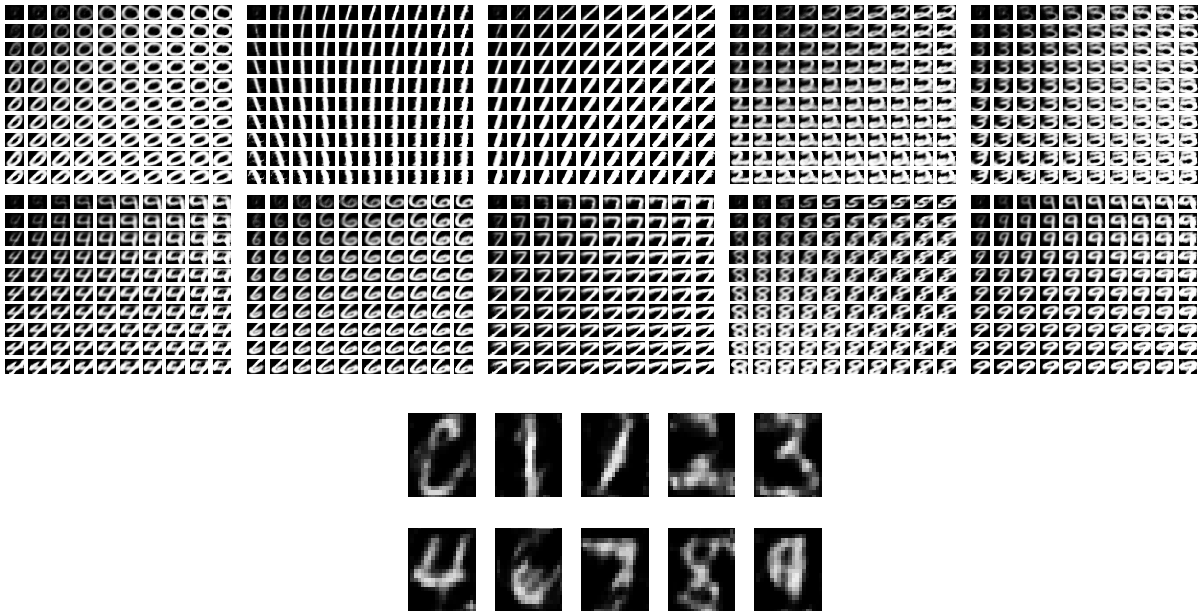

Figure 9: Representations learned by a network with 10 softmax units and 2 softplus units, and the corresponding prototypes. Each manifold is obtained by fixing one softmax unit to 1 and varying the value of the softplus units. The prototypes correspond to the pictures which activate the most each softmax unit.

Finally, we study the representations learned by the network. Figure 9 shows the pictures which activate the most each softmax unit, i.e. the discriminative features used by the network to classify input data, along with the corresponding manifolds learned by the network. These manifolds are obtained by fixing one softmax unit to one and varying the activation of the softplus units. Since scaling all softplus units by the same coefficient results in a more or less contrasted reconstruction (Eq. 4), one dimension of the manifolds corresponds to brightness. Figure 11 corresponds to a network with 10 softmax units and 3 softplus units for which both other dimensions have been represented.

Figure 10 corresponds to a network with no softplus units. Compared to Fig. 9, it shows the synergies between clustering and manifold learning: the softmax layer can focus on discriminative features of each class, while the softplus layer learns to represent inner-class variations.

When 3 softplus units are used, Fig. 11 shows that the clusters are further from natural digits than with 2 softplus units. However, the continuity between adjacent pictures on the manifolds is still preserved. This motivates the fact that this modality alone may not be sufficient to explain the natural classes on which people agree, since several clustering into sub-manifolds are possible. For that reason, we add another modality in the following experiments.

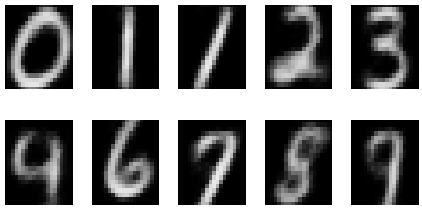

Figure 10: Prototypes learned by a network with no softplus layer. In this case, the top layer consists in only 10 softmax units, trained to reconstruct the input according to the standard auto-encoder paradigm. The prototypes are much closer from the mean instance of each class than in Fig. 9, where they focus on discriminative features.

4.3. Mixing visual input with proprioception

We first add proprioception as a second modality. To address a task similar to the previous one, we perform a task involving digits. We use the iCub humanoid robot [62] to collect a new dataset. We record the joint velocities of the six degrees of freedom of the right arm while it is moved by a human operator in a writing task. Figure 12 illustrates the experimental setup. After each digit, we record the pictures from the iCub cameras. We record a total of 760 samples.

Joint trajectories are normalized to the same duration, resulting in 100 timesteps which give a 6x100 dimensional vector as input. The pictures are processed offline to increase contrast and crop the pictures to a 20x20 pixels bounding box of the digits (in order to make them similar to the MNIST dataset). First, we select only the red channel of the pictures, revert contrast and crop the pictures to their bottom half, corresponding to the whiteboard. This brings out the digits written in green as white on a black background. Then, we compute the center of mass of white pixels and resize the 50x50 pixels windows around the center to a 20x20 pixels image. Figure 13 shows a raw image captured from the cameras and the picture after this preprocessing. Figure 14 shows some picture samples after offline processing[5].

We still use 100 units for the auto-encoder applied to pictures, and 150 units for the auto-encoder applied on trajectories.

4.3.1. Two modalities are better than one

We compare the classification performance of the network on the dataset recorded on iCub when only one of the two modalities is used, or when both modalities are used. Figure 15 compares the performances in each case.

[5] Using full-size images is possible, for instance with convolutional networks as input for the proposed architecture (see Section 5.4 for a discussion). Here, we choose to crop the images for the sake of clarity and to reduce the running time of experiments. Furthermore, we make them similar to the MNIST dataset to re-use the same parameters.

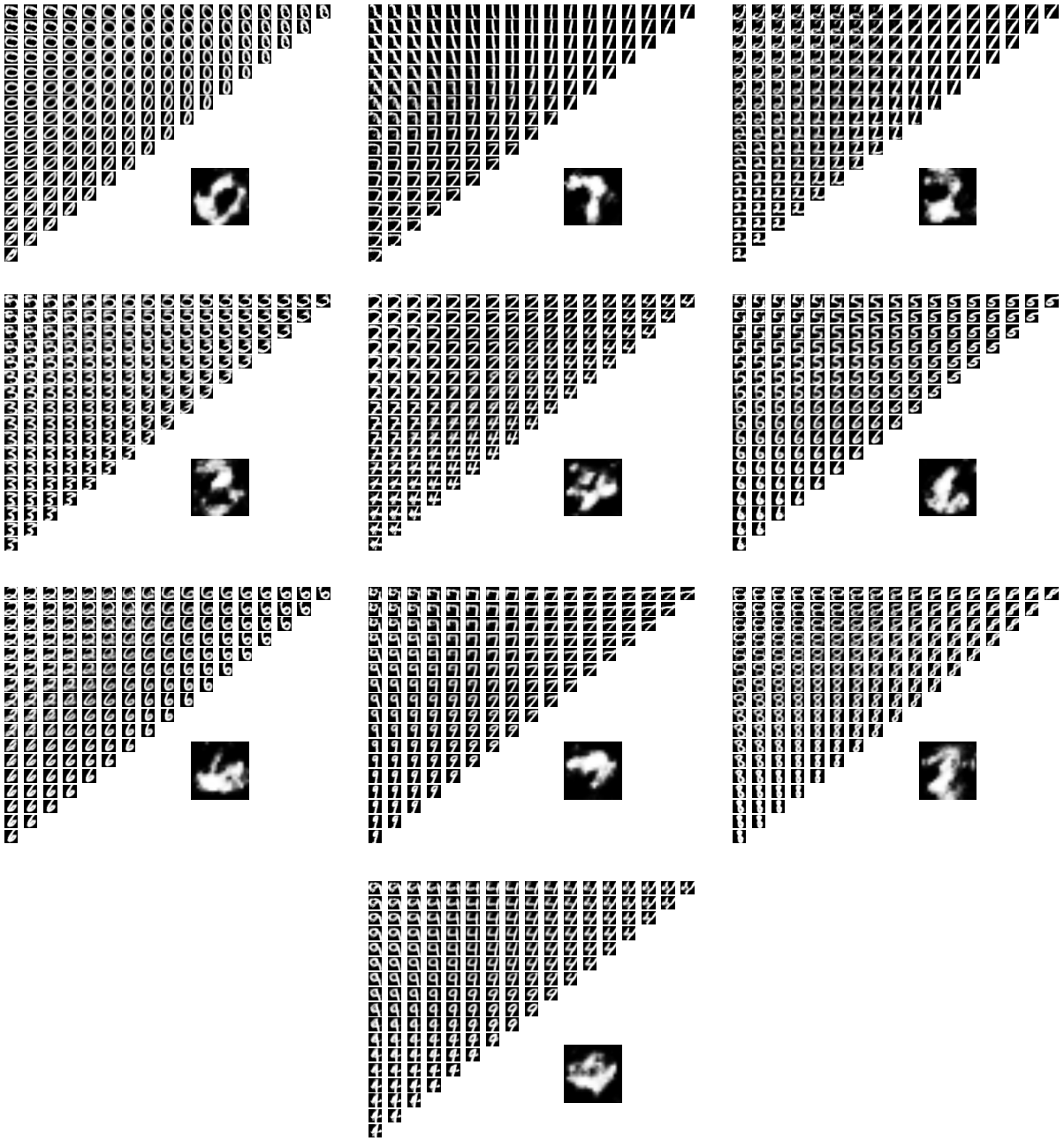

Figure 11: Representations learned by a network with 10 softmax units and 3 softplus units. Each manifold is obtained by fixing one softmax unit to one and varying the activation of the softplus units with a constant total activation (from Eq. 4, scaling all units by the same coefficient results in a more or less bright reconstruction). Each manifold is represented along with its corresponding prototype (picture which activate the most the corresponding softmax unit).

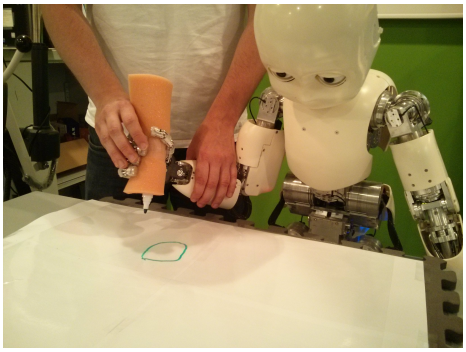

Figure 12: Experimental setup. The iCub robot is in zero torque control [32] while a human experimenter moves its arm to write digits. The joints trajectories are recorded along with the pictures from the robot's cameras for each digit.

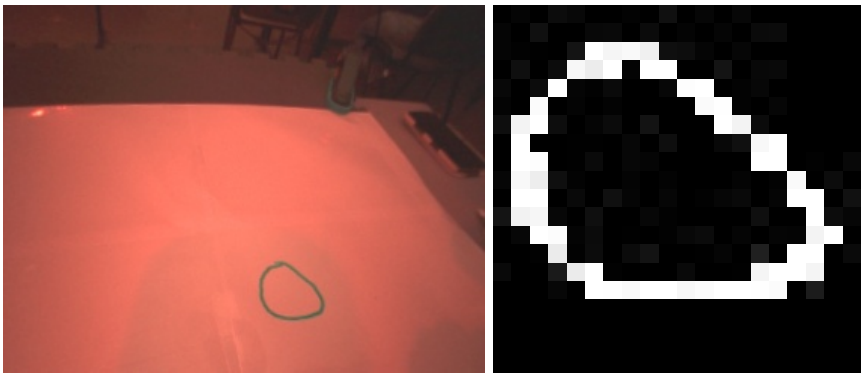

Figure 13: Left: image captured from the iCub camera. Right: same picture after preprocessing.

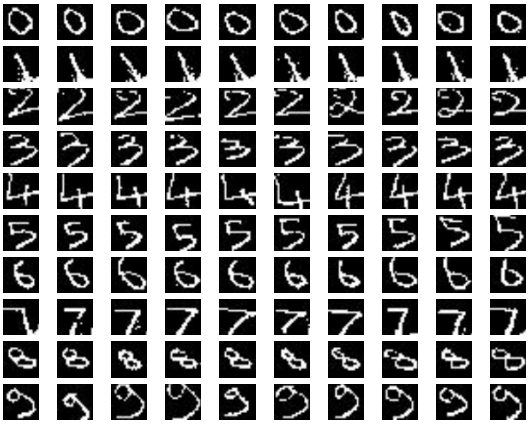

Figure 14: Samples of pictures recorded from the iCub robot after offline preprocessing. A total of 760 pictures has been recorded.

4.3.2. Predicting one modality from the other one

The network can use a partial input where only one modality is available to infer the value of the missing modality. To test its capacities, we train the network with both modalities on 700 samples from our dataset, then test the network on the 60 remaining samples. For each modality, we compute the reconstruction error of one modality given the other (see Fig. 16). Figure 17 shows the reconstruction of the picture given the trajectory. We can see that only one digit 2 is clearly misclassified as a 3. Similarly, Fig. 18 shows the trajectories (computed with a kinematic model of the robot) inferred from pictures. In this case, there are more misclassifications: two 0 are classified as 6 and 2, a 2 is classified as a 4 and a 3 is classified as a 9 (4 errors over 60 samples).

4.4. Extension to three modalities

In this section, we extend the network to three modalities, adding an auditory input. We record a human speaker pronouncing the name of the 10 digits, 76 times per digit. The input then consists of a rough spectrogram of the audio file, computed with a time window of 42ms and 50% overlapping. The auto-encoder contains 150 hidden units. To have the same size for all spectrograms, they were filled with "0" to match the length of the longest one (corresponding to approximately 1.3 second of speech). Figure 19 shows some recorded spectrograms. Each spectrogram corresponds to a vector of 620 values.

Figure 20 shows the classification performance of the network when the three modalities are used, depending on the available input. It shows that classification performance of speech alone is in between pictures and trajectories (median classification score of 0.4 for pictures, 0.0 for trajectories and 0.18 for speech).

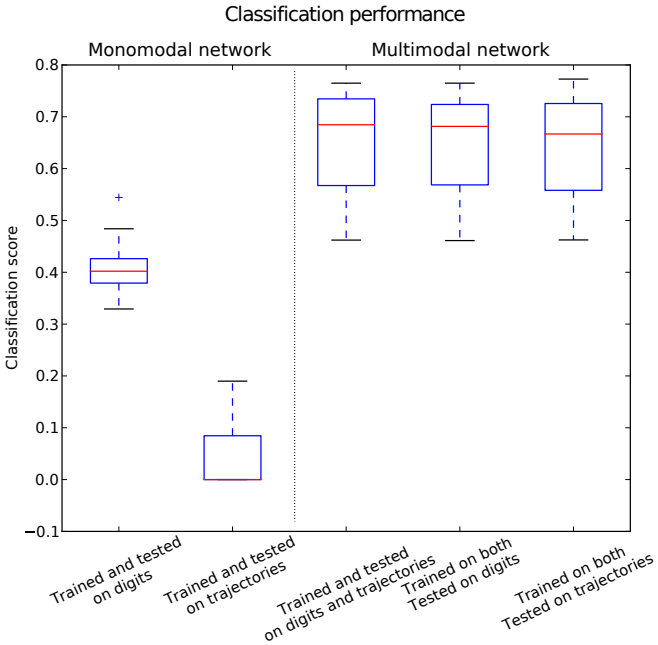

Figure 15: Classification performance with multimodal inputs. Using several modalities increases the classification performance of the algorithm, even if using each modality alone results in poor classification. In particular, mainly due to sampling noise, trajectories are difficult to classify separately, but nevertheless provide useful information in combination with pictures. Moreover, classification performance on trajectories alone is greatly increased when the network has been trained with bimodal inputs.

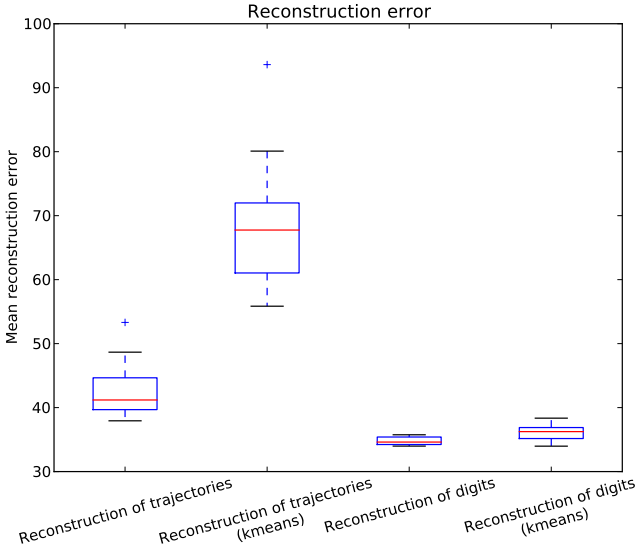

Figure 16: Reconstruction error of one modality given the other one. The network is trained with both modalities as input, and then asked to reconstruct one modality given only the other one. We compare the performance with the k-means algorithm which is trained on a concatenation of both modalities and then tested providing one modality as input, taking the closest cluster centroid using a euclidean distance taking into account only the provided modality, and considering the reconstruction of the other modality as the other part of the centroid. As a baseline, if we use the average of all inputs as the reconstruction, we obtain a mean error of 42.1 for digits and a mean error of 106.3 for trajectories.

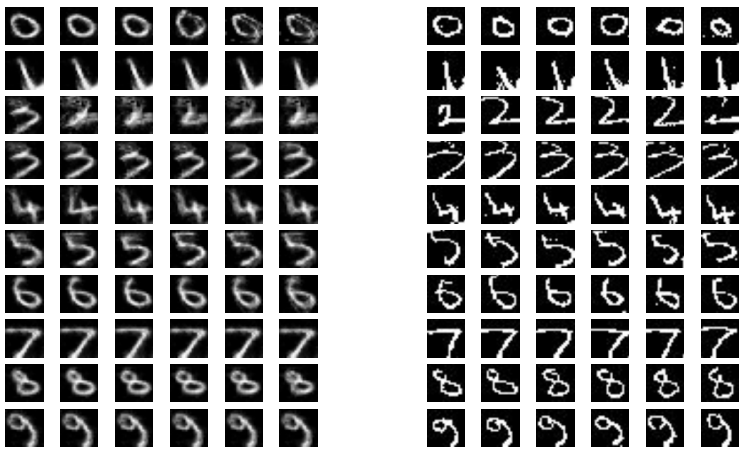

Figure 17: Left: pictures inferred by the network given trajectories (not in the training set). Right: ground truth.

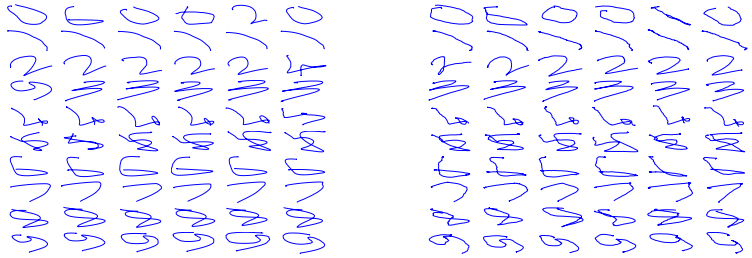

Figure 18: Left: trajectories inferred by the network given pictures (not in the training set). Right: ground truth.

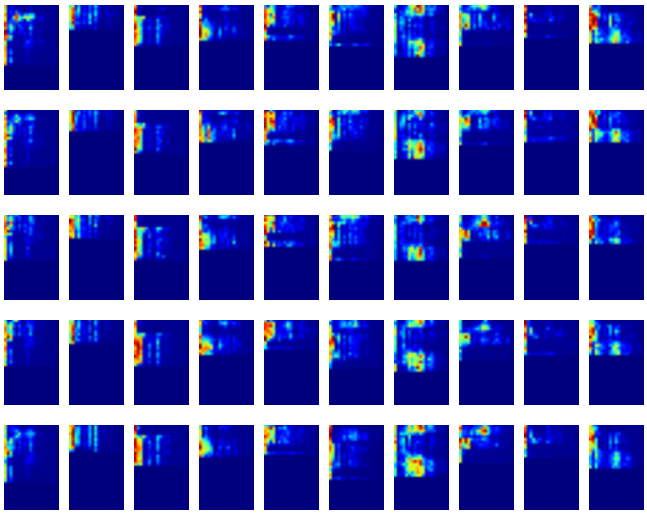

Figure 19: Samples of spectrograms used as third modality. Each column corresponds to a number from 0 (left) to 9 (right).

However, as in Fig. 15 on two modalities, training the network with three modalities increases the classification performance for any combination of modalities.

5. Discussion

We first discuss the results presented in the previous section and then highlight some perspectives and future work.

5.1. Clustering

The first experiment studies the influence of two important parameters: the amount of regularization noise and the number of softmax units. As explained in Section 3.1.2, Fig. 5 shows that without noise the network tends to learn a distributed representation over the softmax units: the mean activation of the most active softmax unit for each sample is about one third. Increasing the amount of noise results in a more discriminative representation. Thus, for a Gaussian noise with a standard deviation of 10, the mean activation of the most active unit is above 0.99. However, Fig. 6 shows that classification performance increases with a small amount of noise, but decreases when the standard deviation is bigger than 2. This is explained by the fact that a huge amount of noise results in a fully random selection of softmax units which prevents the network from learning prototypes of classes. The optimal amount of noise is therefore a

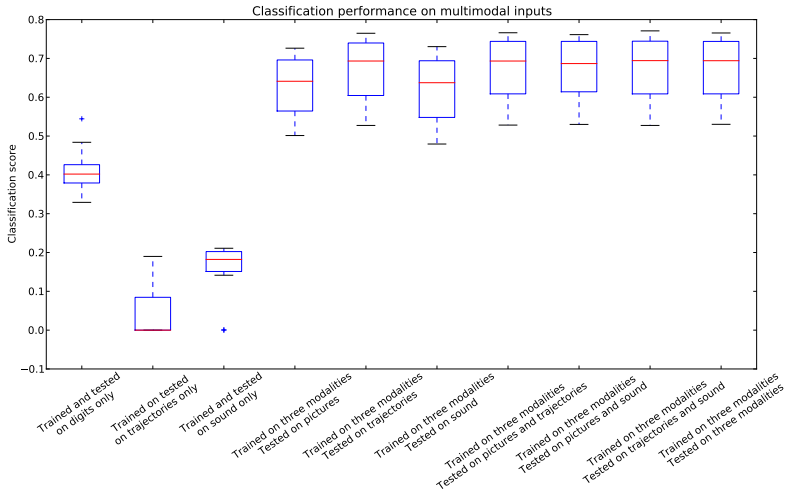

Figure 20: Classification performance of the network trained on three modalities and tested on different subsets of modalities.

trade-off between clustering sharpness and classification performance. Different factors may influence its optimal value, in particular the dataset and the initial weights of the network (i.e. what is the mean input activation of each softmax unit, compared with the mean value of noise) and the learning rate (i.e. the ability for the network to change weights fast enough to break the symmetry induced by a random selection of units).

Figure 8 highlights the ability of the network to use a smaller number of clusters than the number of provided units. This results from several factors. First, the regularization noise forces the network to learn discriminative prototypes of each class in order to make a sharp decision able to counter the effect of the noise. This prevents the network from being very sensitive to small variations of input data. Second, learning a parametrization of the data through a distinct softplus layer lets the network progressively absorb these small variations into the parametrization, allowing the softmax layer to focus on really discriminative features common to all the corresponding samples. Thus, a new unit can be activated only by an input which is far enough from all previous samples. This property is interesting in a life-long learning perspective, since a large number of softmax units can be provided from the beginning, without impacting the performance too much and allowing the network to learn new categories when new stimuli are encountered.

Remarkably, the distinction between the discriminative decision and the generative property, which uses different weight matrices, allows the classification to focus on subparts of the stimuli, as opposed to codebook and self-organizing maps approaches (e.g. [16, 57]). This feature is also characteristic of perceptual symbols systems [2], in which view concepts consist of patterns of selected

aspects of perception (for instance the concept of *chair* does not rely on the perceived color).

5.2. Manifold learning

The last experiment on the MNIST dataset investigates the representation learned by the network. First, Fig. 9 corresponds to a network with two softplus units. In this case, both prototypes and manifolds are easily interpretable in terms of natural classes of digits. Most of them are highly specific (0, 1, 2, 6, 7, 9), others mix some digits (3 and 5, 4 and 9, 8 and 5). In particular, no cluster is dedicated to the digit 5 which is represented either with the digit 3 or the digit 8.

This capacity to represent each data with a coordinate system on a submanifold could be used by a symbolic reasoning system, considering classes as symbols and coordinate values as traits. This is more powerful than representations obtained by self-organizing maps, as in [57, 40], which provide a symbolic representation through the most active unit, but for which discrimination between two similar stimuli corresponding to the same unit would use the distance in the original space. Interestingly, our approach relates to the Convergence-Divergence Zones framework [55], which states that high level representations are not copies of raw perception, but are instead the minimal record needed to reconstruct the approximation of original perceptions in the early cortices.

Adding a third softplus unit (Fig. 11) provides a higher representational power to the network and more variations can be represented around a single prototype. The classification is less interpretable in terms of natural digits, but the continuity between adjacent pictures on each manifold is still present and the learned representation is still meaningful according to the manifold hypotheses.

5.3. Multimodal fusion

In further experiments, we studied how multimodal input influences the classification performance. We first added proprioception, through arm joints velocities recorded during a writing task. Figure 15 shows that pictures alone and trajectories alone were quite difficult to classify. In particular, trajectories are noisy, with an important sampling noise, which explains the very poor performance of the network. However, when the network is trained on both modalities, it is able to learn a much better classification, closer from natural digits classes (the adjusted Rand score is equal to 1 in the case of a perfect match between learned classification and ground truth labels). This result is similar to the one obtained by [59] with latent Dirichlet allocation. However, our approach is more general in the sense that it does not rely on predefined codebooks to encode each input.

More importantly, training the network on both modalities improves the representation of each modality alone, as was observed in [15]. Indeed, after learning, classification performance does not depend anymore on the input modality: the score is nearly the same with only trajectories, only pictures, or both modalities. The same effect is observed when a third modality is added

(Fig. 20). This is in accordance with the observation in humans that multimodality helps to understand each modality alone [23].

Another property of the network is its capacity to infer one modality given the other one, as in [40], which is one of the core properties of Convergence-Divergence Zones [55]. Given one modality alone, the network is able to infer a classification and a parametrization which can be used to reconstruct the missing modality. Figure 16 shows that the reconstruction is more specific than the one obtained through k-means centroids. Moreover, the reconstruction of misclassified digits in Figs. 17 and 18 shows that little variations of the input can produce a complete switch of perception, thanks to the use of two distinct layers for representation. This is reminiscent of the categorical perception effect [26], in which the perceived distance between two stimuli is not related to the true physical distance between stimuli but depends on some internal representations.

5.4. Perspectives and future work

As explained in Section 5.2, the number of softplus units has a strong influence on the classification learned by the network. Future work will consist in endowing the network with the capacity to dynamically select the number of relevant variables.

Another pending issue is the fact that the classification relies on a single pattern of activation. Thus, it cannot handle the case where the same cluster should be represented by a disjunction of different patterns. However, when there exists a modality in which the cluster can be represented by a single prototype, this modality can be used to fuse different prototypes from other modalities into a single pattern. However, this yields a unique pattern of activation which combines all different patterns into a unique one. This prevents from learning exclusive disjunctions. In the case where an exclusive disjunction is necessary, this requires to replace the classification mechanism (based on W in the proposed network) by a more complex one. Sum-product networks [69] could be an interesting source of inspiration for this replacement.

Another open question is the number of modalities to be used as input for our algorithm. As we showed, this number is theoretically unlimited. However, when more modalities are added, it becomes highly probable that some modalities are not correlated with the others, especially in a multi-task scenario (e.g. touching an object while looking at and listening to a dog barking). Several solutions can be investigated. First, a better multimodal fusion strategy could be developed. In the proposed network, each modality is weighted at random for each stimuli, since it assumes that all modalities always agree and that the classification decision can be taken from anyone of them. This approach can be seen as a constrained version of the "disagreement minimization framework" [15], that could be relaxed. Another idea, inspired by the Convergence-Divergence Zones [55], would be to use several instantiations of the proposed network and limit the number of input modalities for each instance to two or three. All these networks could then be chained using each time one modality as a pivot (see Fig. 21). Thus, stimulating one modality could still flow through the whole network and re-activate corresponding stimuli in other modalities, but, during

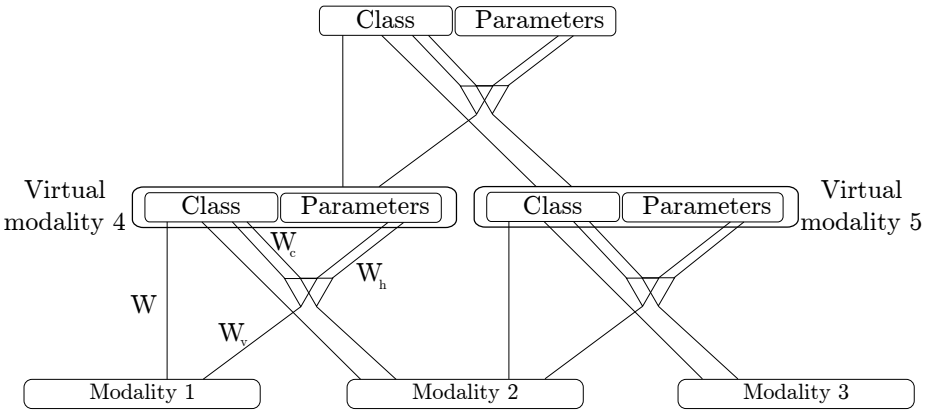

Figure 21: Our network can take as input any number of modalities. However, from a statistical perspective (see text for details), it may be more interesting to chain several networks, each network taking only two modalities as input. Several architectures can be investigated: either each combination of two modalities has its own network (in this figure, it would require one more network between modality 1 and modality 3), or some modalities can act as "pivots" (e.g. modality 2 in this figure). In the latter case, "pivots" modalities are used to spread activation from one modality to another. In this figure, activation of modality 1 can activate a corresponding reconstruction of modality 2, which would itself induce an activation in modality 3. This figure also illustrates how our network can be stacked, considering the output of one network as a "virtual" modality used as input for a higher level network. A similar architecture can be used for convolutional networks, in which modalities 1 to 3 would correspond to the output of a lower level network applied for instance on different subparts of a picture.

training, the effect of uncorrelated modalities would be restricted to a limited zone of influence.

Stacking the proposed network into several hierarchical levels seems also necessary in a long term perspective. For instance, it would be interesting to have a global network able to decompose the "concept" of a face as a combination of two eyes, one nose and one mouth, which would be themselves "concepts" from a lower layer. For this reason, using the proposed architecture on top of a standard convolutional network [45, 42] may not be an optimal solution, since it would force the network to categorize each image as a whole. This rather suggests to fully integrate the proposed network into a convolutional architecture, which would progressively categorize subparts of an image in a hierarchical way. In this view, a first layer of the proposed network would be applied to subparts of the images, to extract local concepts, such as *eye*, *nose*, etc. Thus, each subpart would be represented by a classification vector corresponding to the "concept", along with a parameter vector specifying its coordinates along the concept manifold. Another layer could then take the output from several subparts as input (as in a standard convolutional architecture), and learn to represent higher level concepts (see Fig. 21). This relies on the assumption that the manifold hypothesis holds using such a hierarchical representation. This hypothesis is reasonable: considering the case of images, it states that there are some constraints on the

relative values of the outputs of the network applied to different subparts of an image. Perceiving an eye at one location actually constrains what can be perceived at other locations, it terms of other low level concepts (we expect another eye, a nose and a mouth, but not a gear for instance), and in terms of relative parameter values of each of these concepts (for instance we expect both eyes to have similar appearance, i.e. similar parameters).

Moreover, as in the Convergence-Divergence Zones framework, the proposed network could be used in a hierarchical architecture such that the monomodal version is applied to the lower levels of representations, and that higher layers progressively integrate multimodal representations. Several hierarchical levels could also be used to transform parameters values from lower layers into concepts in upper layers. For instance, smoothness is at the same time a continuous traits of objects but also a concept (which can correspond to different ranges of haptic perceptions depending on the object being manipulated).

The proposed network has been presented in a very generic fashion. However, much work remains to be done to determine the most efficient and appropriate architecture to handle a rich and complex sensorimotor experience close to the one handled by humans.

A shortcoming of the presented work is the pre-processing of temporal data into static inputs: the trajectories are segmented, normalized to the same duration and linearized into a unique vector, speech data is provided as a full spectrogram. Future work will focus on developing an architecture able to learn to segment temporal series and extract relevant patterns in an unsupervised way. This is a crucial aspect particularly for speech segmentation and word recognition, to apply the proposed network to fully unsupervised language grounding tasks. The temporal aspects of the signals are important to enable learning of low-level sensorimotor contingencies, for instance between short joint trajectories and their effects on drawing. Learning such contingencies is crucial to efficiently extrapolate knowledge of new classes of stimuli. As an example, consider a transfer task where learning to write digits could be helpful to infer the trajectories for an alphabet writing task.

A disincentive to using the proposed network for developmental robotics is that it is trained through several, sequential stages which seem at odds with the developmental, life-long learning paradigm. The sequential training of each layer is designed to provide a good initialization of the weights to avoid poor minima and speed-up learning [29]. Thus, it is not incompatible with a life-long learning process applied to the whole network, after an initial stage of sequential pre-training. Recent works on artificial curiosity and intrinsic motivation also suggest that sequential training is helpful for developmental learning [1] and may account for early developmental stages. Furthermore, some techniques have been proposed recently to avoid this pre-training (e.g. [51]).

A second objection concerns the use of autoencoders, which require to be trained on i.i.d. datasets. As a counter-argument, it has been shown that learning new tasks or new stimuli in a neural network can lead to the so-called *catastrophic forgetting* [22], i.e. a perturbation of previously learned tasks so significant that the network becomes inefficient. However, some works have started

addressing this issue in the context of deep networks. First, [67] proposes an organization in which new data are affected to the training of different subnetworks according to their current capacities. Another interesting approach was proposed by [8]: since deep networks are trained to generate their input data, these generated data can be used to extend the dataset. Thus, new incoming data can be mixed with reconstructions of previously learned data (generated on the fly without important storage requirements) to avoid catastrophic forgetting. This idea has still to be much further developed, but is a promising approach to use deep networks in life-long learning settings.

6. Conclusion

In this paper, we proposed an architecture which structures high-dimensional, multimodal information into a set of distinct sub-manifolds. It performs a substantial dimensionality reduction by providing both a symbolic representation of data with a softmax classification layer and a fine discrimination between two similar stimuli through a parametrization of underlying sub-manifolds. Moreover, the proposed network is able to exploit multimodal correlations to improve the representation of each modality alone.

Our work opens several perspectives. First, it paves the way for robots using cheap, off-the-shelf sensors providing a noisy but redundant flow of information, decreasing the need for carefully hand-crafted, task-oriented perception. Second, it can contribute to natural human-robot interaction allowing very loose teaching signal (such as spoken words). In particular, for programming by demonstration, this could be used to relax the interaction scenario. Last but not least, considering proprioception as one among other modalities, this enables a tight coupling between action and perception, closing the *action-perception* loop at a very low level.

References

[1] A. Baranes and P.-Y. Oudeyer. The Interaction of Maturational Constraints and Intrinsic Motivations in Active Motor Development. In *ICDL - EpiRob*, volume 2, pages 1–8. IEEE, 2011.

[2] L. W. Barsalou. Perceptual symbol systems. *Behavioral and Brain Sciences*, 22:577–660, 8 1999.

[3] L. W. Barsalou and J. J. Prinz. Mundane creativity in perceptual symbol systems. In T. B. Ward, S. M. Smith, and J. Vaid, editors, *Conceptual structures and processes: Emergence, discovery, and change*. American Psychological Association, 1997.

[4] Y. Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127, 2009.

[5] Y. Bengio, A. Courville, P. Vincent, and U. Montreal. Representation Learning: A Review and New Perspectives. *arXiv*, 1206.5538v2:1–34, 2012.

[6] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *Neural Networks*, 5(2):157–166, 1994.

[7] M. Botvinick and J. Cohen. Rubber hands 'feel' touch that eyes see. *Nature*, 391(6669):756, Feb. 1998.

[8] R. Calandra, T. Raiko, M. P. Deisenroth, and F. M. Pouzols. Learning deep belief networks from non-stationary streams. In *Artificial Neural Networks and Machine Learning–ICANN 2012*, pages 379–386. Springer, 2012.

[9] L. Cayton. Algorithms for manifold learning. Technical report, University of California, San Diego, 2005.

[10] C. Ciliberto, S. R. Fanello, M. Santoro, L. Natale, G. Metta, and L. Rosasco. On the impact of learning hierarchical representations for visual recognition in robotics. In *IROS*, pages 3759–3764. IEEE, 2013.

[11] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber. Deep Big Multilayer Perceptrons For Digit Recognition. *Neural Networks Tricks of the Trade*, 1:581–598, 2012.

[12] A. R. Damasio. The brain binds entities and events by multiregional activation from convergence zones. *Neural Computation*, 1(1):123–132, 1989.

[13] A. R. Damasio. Time-locked multiregional retroactivation: A systems-level proposal for the neural substrates of recall and recognition. *Cognition*, 33(1):25–62, 1989.

[14] A. R. Damasio. Category-related recognition defects as a clue to the neural substrates of knowledge. *Trends in neurosciences*, 13(3):95–98, 1990.

[15] V. R. De Sa and D. H. Ballard. Perceptual learning from cross-modal feedback. *Psychology of learning and motivation*, 36:309–351, 1997.

[16] V. R. De Sa and D. H. Ballard. Category learning through multimodality sensing. *Neural Computation*, 10(5):1097–1117, 1998.

[17] O. Delalleau and Y. Bengio. Shallow versus Deep Sum-Product Networks. In *NIPS*, pages 666–674, 2011.

[18] D. Erhan, P.-A. Manzagol, Y. Bengio, S. Bengio, and P. Vincent. The difficulty of training deep architectures and the effect of unsupervised pretraining. In *AISTAT*, volume 5, 2009.

[19] A. Falchier, S. Clavagnier, P. Barone, and H. Kennedy. Anatomical evidence of multimodal integration in primate striate cortex. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 22(13):5749–59, July 2002.

[20] A. Fort, C. Delpuech, J. Pernier, and M. H. Giard. Early auditory-visual interactions in human cortex during nonredundant target identification. *Brain research. Cognitive brain research*, 14(1):20–30, June 2002.

[21] E. Freeman, J. Driver, D. Sagi, and L. Zhaoping. Top-down modulation of lateral interactions in early vision: Does attention affect integration of the whole or just perception of the parts? *Current Biology*, 13(11):985–989, 2003.

[22] R. M. French. Catastrophic forgetting in connectionist networks. *Encyclopedia of Cognitive Science*, 1991.

[23] M. H. Giard and F. Peronnet. Auditory-visual integration during multimodal object recognition in humans: A behavioral and electrophysiological study. *J. Cognitive Neuroscience*, 11(5):473–490, Sept. 1999.

[24] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics (AISTATS10)*, volume 9, pages 249–256, 2010.

[25] R. Goldstone and L. Barsalou. Reuniting perception and conception. *Cognition*, 65(2-3):231–262, 1998.

[26] R. L. Goldstone and A. T. Hendrickson. Categorical perception. *Wiley Interdisciplinary Reviews: Cognitive Science*, 1(1):69–78, 2010.

[27] A. Graves, A. rahman Mohamed, and G. E. Hinton. Speech recognition with deep recurrent neural networks. In *ICASSP*, pages 6645–6649. IEEE, 2013.

[28] M. Hermans and B. Schrauwen. Training and analysing deep recurrent neural networks. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 190–198, 2013.

[29] G. E. Hinton and R. R. Salakhutdinov. Reducing the Dimensionality of Data with Neural Networks. *Science*, 313(5786):504–507, 2006.

[30] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[31] L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[32] S. Ivaldi, M. Fumagalli, M. Randazzo, F. Nori, G. Metta, and G. Sandini. Computing robot internal/external wrenches by means of inertial, tactile and f/t sensors: theory and implementation on the icub. In *Proc. of the 11th IEEE-RAS International Conference on Humanoid Robots - HUMANOIDS*, pages 521–528, Bled, Slovenia, 2011.

[33] S. Ivaldi, S. M. Nguyen, N. Lyubova, A. Droniou, V. Padois, D. Filliat, and O. Sigaud. Object learning through active exploration. *IEEE Transactions on Autonomous Mental Development*, pages 1–18, 2013.

[34] A. Jain, M. Murty, and P. Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[35] M. Johnsson, C. Balkenius, and G. Hesslow. Associative Self-organizing Map. *IJCCI*, pages 363–370, 2009.

[36] D. Joyce, L. Richards, A. Cangelosi, and K. Coventry. On the foundations of perceptual symbol systems: Specifying embodied representations via connectionism. In F. Detje, D. Dorner, and H. Schaub, editors, *The Logic of Cognitive Systems. Proceedings of the Fifth International Conference on Cognitive Modeling*, pages 147–152, Universitatsverlag Bamberg, 2003.

[37] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological cybernetics*, 69:59–69, 1982.

[38] O. Kouropteva, O. Okun, and M. Pietikäinen. Incremental locally linear embedding. *Pattern recognition*, 38(10):1764–1767, 2005.

[39] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In P. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25, pages 1106–1114, 2012.

[40] S. Lallee and P. Dominey. Multi-modal convergence maps: From body schema and self-representation to mental imagery. *Adaptive Behavior*, 21:274–285, 2013.

[41] M. H. C. Law and A. K. Jain. Incremental nonlinear dimensionality reduction by manifold learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(3):377–91, Mar. 2006.

[42] Y. LeCun, K. Kavukvuoglu, and C. Farabet. Convolutional Networks and Applications in Vision. In *Proc. International Symposium on Circuits and Systems (ISCAS'10)*, pages 253–256. IEEE, 2010.

[43] D. Lee and S. Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, volume 13, pages 556–562, 2001.

[44] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems*, 2006.

[45] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 609–616, New York, NY, USA, 2009. ACM.

[46] M. Lefort, Y. Boniface, and B. Girau. Self-organization of neural maps using a modulated BCM rule within a multimodal architecture. In *BICS*, 2010.

[47] S. Lemaignan, R. Ros, L. Msenlechner, R. Alami, and M. Beetz. Oro, a knowledge management platform for cognitive architectures in robotics. In *IROS*, pages 3548–3553. IEEE, 2010.

[48] N. Lyubova and D. Filliat. Developmental Approach for Interactive Object Discovery. In *Int. Joint Conf. on Neural Networks*, 2012.

[49] J. Macqueen. Some methods for classification and analysis. In *Berkeley Symposium on Mathematical Statistics and Probability*, volume 233, pages 281–297, 1967.

[50] O. Mangin and P.-Y. Oudeyer. Learning semantic components from subsymbolic multimodal perception. In *Third Joint International Conference on Development and Learning and Epigenetic Robotics (ICDL)*, pages 1–7. IEEE, 2013.

[51] J. Martens. Deep learning via hessian-free optimization. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 735–742, 2010.

[52] H. Mcgurck and J. W. Macdonald. Hearing lips and seeing voices. *Nature*, 264(246-248), 1976.

[53] R. Memisevic and G. E. Hinton. Learning to Represent Spatial Transformations with Factored Higher-Order Boltzmann Machines. *Neural Computation*, 22(6):1473–1492, 2010.

[54] R. Memisevic, C. Zach, G. Hinton, and M. Pollefeys. Gated Softmax Classification. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23, pages 1603–1611, 2010.

[55] K. Meyer and A. Damasio. Convergence and divergence in a neural architecture for recognition and memory. *Trends in neurosciences*, 32(7):376–382, 2009.

[56] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor. Learning Object Affordances: From Sensory–Motor Coordination to Imitation. *IEEE Transactions on Robotics*, 24(1):15–26, 2008.

[57] A. F. Morse, T. Belpaeme, A. Cangelosi, and L. B. Smith. Thinking with your body: Modelling spatial biases in categorization using a real humanoid robot. In *Proc. of 2010 annual meeting of the Cognitive Science Society. Portland, USA*, pages 1362–1368, 2010.

[58] A. F. Morse, J. De Greeff, T. Belpeame, and A. Cangelosi. Epigenetic robotics architecture (era). *Autonomous Mental Development, IEEE Transactions on*, 2(4):325–339, 2010.

[59] T. Nakamura, T. Nagai, and N. Iwahashi. Grounding of word meanings in multimodal concepts using lda. In *International Conference on Intelligent Robots and Systems*, pages 3943–3948. IEEE, 2009.

[60] T. Nakamura, T. Nagai, and N. Iwahashi. Bag of multimodal lda models for concept formation. In *International Conference on Robotics and Automation*, pages 6233–6238. IEEE, 2011.

[61] H. Narayanan and S. Mitter. Sample complexity of testing the manifold hypothesis. *Advances in Neural Information Processing*, 2010.

[62] L. Natale, F. Nori, G. Metta, M. Fumagalli, S. Ivaldi, U. Pattacini, M. Randazzo, A. Schmitz, and G. G. Sandini. The icub platform: a tool for studying intrinsically motivated learning. In *Intrinsically motivated learning in natural and artificial systems - Ed. Baldassarre, G.and Mirolli, M.*, pages 433–458. Springer-Verlag, 2013.

[63] A. Y. Ng, M. I. Jordan, Y. Weiss, et al. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, volume 14, pages 849–856, 2001.

[64] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal Deep Learning. In *International Conference on Machine Learning*, pages 689–696, Bellevue, USA, 2011.

[65] S. O'Hara and B. A. Draper. Introduction to the bag of features paradigm for image classification and retrieval. *arXiv preprint arXiv:1101.3354*, 2011.

[66] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997.

[67] L. Pape, F. Gomez, M. Ring, and J. Schmidhuber. Modular deep belief networks that do not forget. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 1191–1198. IEEE, 2011.

[68] A. Papliski and L. Gustafsson. Multimodal feedforward self-organizing maps. *Computational Intelligence and Security*, pages 81–88, 2005.

[69] H. Poon and P. Domingos. Sum-Product Networks: A New Deep Architecture. In *UAI*, pages 337–346, 2011.

[70] S. Reed and H. Lee. Learning Deep Representations via Multiplicative Interactions between Factors of Variation. In *NIPS Workshop*, 2013.

[71] B. Ridge, D. Skocaj, and A. Leonardis. Self-supervised cross-modal online learning of basic object affordances for developmental robotic systems. In *International Conference on Robotics and Automation*, pages 5047–5054. IEEE, 2010.

[72] S. Rifai, Y. N. Dauphin, P. Vincent, Y. Bengio, and X. Muller. The Manifold Tangent Classifier. In J. Shawe-Taylor, R. S. Zemel, P. Bartlett, F. C. N. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, pages 2294–2302, 2011.

[73] S. Rifai, G. Mesnil, P. Vincent, X. Muller, Y. Bengio, Y. Dauphin, and X. Glorot. Higher Order Contractive Auto-Encoder. In *ECML/PKDD (2)*, pages 645–660, 2011.

[74] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio. Contractive Auto-Encoders: Explicit Invariance During Feature Extraction. In *Proceedings of the 28th International Conference on Machine Learning*, pages 833–840, 2011.

[75] R. R. Salakhutdinov, J. Tenenbaum, and A. Torralba. Learning to Learn with Compound HD Models. In J. Shawe-Taylor, R. S. Zemel, P. Bartlett, F. C. N. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24, pages 2061–2069, 2011.

[76] A. Salman and K. Chen. Exploring speaker-specific characteristics with deep learning. In *The 2011 International Joint Conference on Neural Networks*, pages 103–110. IEEE, July 2011.

[77] A. Schneider, J. Sturm, C. Stachniss, M. Reisert, H. Burkhardt, and W. Burgard. Object identification with tactile sensors using bag-of-features. In *International Conference on Intelligent Robots and Systems*, pages 243–248. IEEE, Oct 2009.

[78] L. Smith and M. Gasser. The development of embodied cognition: Six lessons from babies. *Artificial Life*, 11:13–30, 2005.

[79] R. Socher, E. H. Huang, J. Pennington, A. Y. Ng, and C. D. Manning. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. C. N. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, pages 801–809, 2011.

[80] A. Stuhlsatz, J. Lippel, and T. Zielke. Discriminative feature extraction with Deep Neural Networks. In *IJCNN*, pages 1–8. IEEE, 2010.

[81] G. W. Taylor, G. E. Hinton, and S. T. Roweis. Two Distributed-State Models For Generating High-Dimensional Time Series. *J. Mach. Learn. Res.*, 12:1025–1068, 2011.

[82] V. Tikhanoff, A. Cangelosi, and G. Metta. Integration of speech and action in humanoid robots: icub simulation experiments. *Autonomous Mental Development, IEEE Transactions on*, 3(1):17–29, 2011.

[83] E. Ugur, E. Sahin, and E. Oztop. Affordance learning from range data for multi-step planning. In *EpiRob*, 2009.

[84] M. Vavrečka and I. Farkaš. A Multimodal Connectionist Architecture for Unsupervised Grounding of Spatial Language. *Cognitive Computation*, pages 1–12, 2013.

[85] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning - ICML '08*, pages 1096–1103, New York, New York, USA, 2008. ACM Press.

[86] M. Waibel, M. Beetz, J. Civera, R. D'Andrea, J. Elfring, D. Gálvez-López, K. Häussermann, R. Janssen, J. Montiel, A. Perzylo, B. Schießle, M. Tenorth, O. Zweigle, and R. van de Molengraft. RoboEarth - A World Wide Web for Robots. *IEEE Robotics and Automation Magazine (Special Issue Towards a WWW for Robots)*, 2011.

[87] J. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

[88] S. Wermter, C. Weber, M. Elshaw, C. Panchev, H. Erwin, and F. Pulvermüller. Towards multimodal neural robot learning. *Robotics and Autonomous Systems*, 47(2-3):171–175, June 2004.

[89] C. Yu and D. H. Ballard. On the integration of grounding language and learning objects. In *AAAI*, volume 4, pages 488–493, 2004.

[90] H. Zhao, P. C. Yuen, and J. T. Kwok. A novel incremental principal component analysis and its application for face recognition. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, 36(4):873–86, Aug. 2006.