ESANN'2006 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2006, d-side publi., ISBN 2-930307-06-4.

Adaptive Sensor Modelling and Classification using a Continuous Restricted Boltzmann Machine (CRBM)

Tong Boon Tang and Alan F. Murray *

The University of Edinburgh, School of Engineering and Electronics
King's Buildings, Mayfield Road, Edinburgh EH9 3JL, United Kingdom.

Abstract. This paper presents a neural approach to sensor modelling and classification as the basis of local data fusion in a wireless sensor network. Data distributions are non-Gaussian. Data clusters are sufficiently complex that the classification problem is markedly non-linear. We prove that a Continuous Restricted Boltzmann Machine can model complex data distributions and can autocalibrate against real sensor drift. To highlight the adaptation, two trained but subsequently non-adaptive neural classifiers (SLP and MLP) were employed as benchmarks.

1 Introduction

Driven by the recent advances in microsensor technology, many multisensor microsystems have been developed for different applications. For example, in the Wireless Integrated Network Sensors (WINS) array [1], low cost sensing devices are deployed for environmental/machinery monitoring in place of an expensive fully-wired system. To overcome the narrow communication network bandwidth, sensor signals are processed locally. Events of interest provoke an alarm to a basestation for further decision making. Local data fusion (typically classication) uses a neural network because it allows data fusion at all levels (signal, pixel, feature and symbol).

In this paper, we focus on local data fusion using of a generative model, the "Continuous Restricted Boltzmann Machine" (CRBM) and a Single Layer Perceptron (SLP). The former extracts salient features by modelling the integrated sensors, while the latter performs binary classification on the extracted features. The CRBM was developed specifically to fuse data at signal level and to have a hardware-amenable architecture [2].

The CRBM has one visible and one hidden layer with a symmetric inter-layer weight matrix $\{\mathbf{W}\}$. Each stochastic neuron j takes the following form:

$$s_j = tanh\left[a_j\left(\sum_i w_{ij}s_i + \sigma \cdot N_j(0,1)\right)\right], \tag{1}$$

where s_i refers to input from neuron i, and $N_j(0,1)$ represents a unit Gaussian noise with zero mean. The noise component $\sigma \cdot N_j(0,1)$ allows the CRBM to perform probabilistic analogue computation via Gibbs sampling method. The noise

*This project is financially supported by Scottish Higher Education Funding Council (Grant number: RDG 130) and EPSRC grant GR/R47318.

control parameter a_j controls the slope of the sigmoid function, such that the behaviour of a neuron j is either deterministic (small a_j), continuous-stochastic (moderate a_j), or binary-stochastic (large a_j). Both $\{a_j\}$ and $\{w_{ij}\}$ are trained by a "Minimizing Contrastive Divergence" (MCD) learning rule [3]. The simplified MCD learning rule [2] requires only addition and multiplication, and is therefore more hardware-amenable.

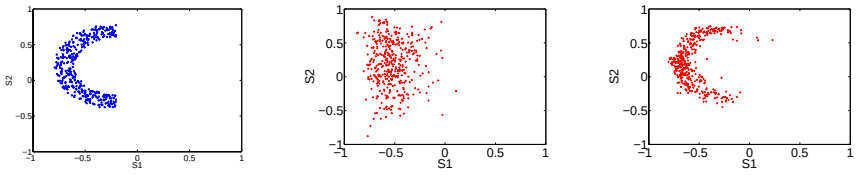
The objectives of this paper are two-fold: 1) to demonstrate that the CRBM can model non-Gaussian distributions and 2) to illustrate the advantage of adaptive sensor fusion in a dynamic environment. The paper is organized as follows. Section 2 discusses how to model non-Gaussian distributions with a CRBM. In this case, the learning is evaluated based on the reconstruction model and a binary classification with a SLP connected to the hidden units of the CRBM (Section 3). Section 4 examines the CRBM's modelling capability in both static and dynamic environments. A conclusion is provided at the end of the paper.

2 Sensor Modelling

To perform binary classification with a neural network, typically the sensor data distributions will first be encoded by a neural model and then one or more hyperplane(s) defined by the model will be used to separate one class of data from another. While the noise, which causes the dispersion in the distributions, is often taken to be Gaussian, this is not valid in many real applications. For example, strong non-Gaussian radio-frequency interference is unavoidable in landmine detection [4], and the emitted signals by sources are often non-Gaussian in localization of multiple sources [5]. Therefore, we have extended our own previous work [6] to non-Gaussian modelling problems.

Modelling non-Gaussian distributions with Gaussian experts is a non-trivial task. To illustrate this, we modelled a 2-D non-Gaussian data distribution (Fig.1a) by using a CRBM with five (Gaussian) hidden units. The learning rates for $\{w_{ij}\}$ and $\{a_j\}$ were 0.2, while all noise scaling factors σ were 0.3. After 5000 training epochs, the learning result (*reconstruction*) was poor (Fig.1b). Although a_j in visible layer autonomously annealed from its initial large value (i.e. 6.0), the model was trapped in a local minima with final $a_j(S1, S2) = [1.50\ 2.87]$, where S1 and S2 are two sensory inputs to the CRBM. One solution is to reduce the learning rate for a_j to 0.01, at the expense of much longer training period ($\geq$ 200000 epochs). A more computationally-efficient approach is to fix a_j at a low value (e.g. 0.1), ie. lowering and controlling the noise component in Eq.1 and hence the spread of the individual distributions. As shown in Fig.1c, the resulting reconstructed distribution is better-defined.

Upon completion of training, the CRBM is configured to adapt to drift in its baseline without *Castatrophic Interference* (CI) [7]. All the learning in a_j and w_{ij} except w_{0j} and w_{i0} is discontinued [6]. Such configuration ensures that 1) the distributions' general features are retained, 2) the distributions' baselines encoded by w_{i0} can adapt to drift, and 3) a consistent representation of drifting data is presented (with updated w_{0j}) to subsequent layer.



(a) (b) (c)

Fig. 1: (a) Non-Gaussian training data, (b) reconstruction data after 5000 training epochs with a_j free running, and (c) reconstruction data after 5000 training epochs with a_j fixed at 0.1.

3 Binary Classification

While the CRBM has a simple, hardware-amenable learning rule, it is difficult to measure the success of learning quantitatively. The CRBM shares the characteristic of the Product of Experts (PoE) [3] that it is not possible to compute the normalized constant $\log Z$, where Z is the sum of the probabilities of *all possible* CRBM data vectors for a set of $\{w_{ij}\}$ and $\{a_j\}$.

An alternative way to evaluate the quality of CRBM learning is to look at the *reconstruction*. In this context, *reconstruction* refers to the distribution generated by current CRBM at the visible layer after N-step Gibbs sampling. Typically, the encoded distribution will emerge within ten steps, with the input data for sampling chosen to be a continuous uniform distribution on (0,1). Then the reconstruction is compared with the training data distribution to give a "score" to the learning process.

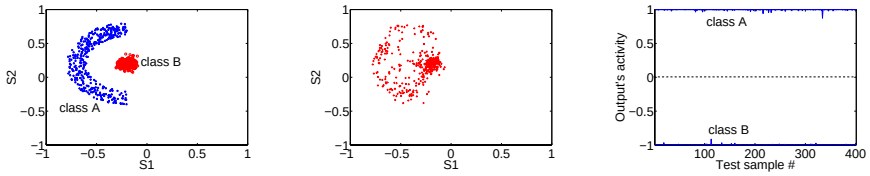
The extent to which the CRBM has encoded the salient features of the data may also be probed by performing classification on the hidden layer activities of the CRBM. An SLP with sigmoidal activation function is trained with the features extracted by a learnt CRBM. The SLP is clamped to '+1' during the training of class A data, and to '-1' during class B. Quantifying the learning result can be obtained by thresholding the activities of the SLP with respect to a set of test data and calculating the number of correctly classified test data.

4 Simulation Results

4.1 Modelling : Artificial Data

A CRBM with 7 hidden units was trained for 5000 epochs to encode two artificial clusters with 400 samples for each class (Fig.2a). All the noise scaling factors and the learning rates were set empirically to 0.4 and 0.2 respectively to achieve a balance between the model's convergence rate and details into modelling the distributions. Whilst the a_j in visible layer was fixed at 0.1, its counterpart in hidden layer was trained, to allow the different hidden units to perform disparate functions.

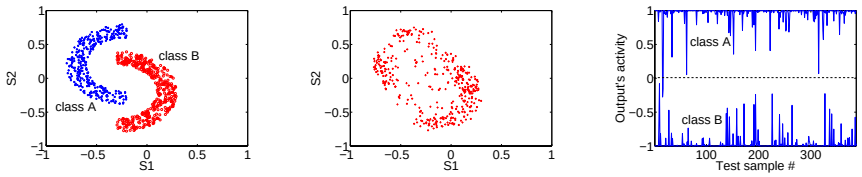
After 5000 training epochs, Fig.2b, the CRBM has modelled what is a reasonably challenging pair of distributions with some success. The response/activity of the trained SLP to a test dataset with 400 samples for each class (Fig.2c) also shows that the combined CRBM and SLP can classify the two clusters with 100% accuracy, while an SLP alone clearly could not do so.



(a) (b) (c)

Fig. 2: (a) Training data where Class A is non-Gaussian and Class B is Gaussian, (b) reconstruction after 5000 training epochs, and (c) the trained SLP's response to test data.

The simulation was re-run to model another pair of distributions (Fig.3a). As these clusters are more complicated than previous pair, a longer training period was required. Fig.3b shows the reconstruction after 20000 training epochs. The CRBM models the training data distributions with some success once more, while the combined CRBM and SLP classify test data with 100% accuracy (Fig.3c).



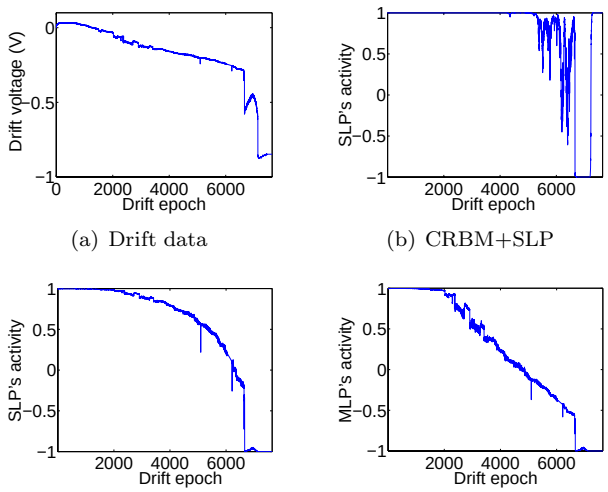
(a) (b) (c)

Fig. 3: (a) Training data where both classes are non-Gaussian, (b) reconstruction after 20000 training epochs, and (c) the trained SLP's response to test data.

4.2 Drift Tracking : Real Sensor data

In our Lab-in-a-Pill application [8], the pH sensors were subject to noise (due to interfering ions) and sensor drift, especially in the harsh environment of the GastroIntestinal (GI) tract. An example of real drift dataset over 20hrs is illustrated in Fig.4a. With the sampling frequency at 0.1Hz, every 10s is defined as one drift epoch. For this drift simulation, the training data is as in Fig.3a. A CRBM with 7 hidden units was trained to model it in a static environment. Subsequently, the trained CRBM was fed with drifting data in which sensor *S1* drifted towards its upper limit ('+1') and sensor *S2* in the opposite direction.

Moreover, all drifting data were taken from class A, as a fully-balanced representation of the data distributions is not always available in real applications. During the 7644 drift epochs, the CRBM's weight was allowed to adapt, at a learning rate of 0.01, to the sensor drift with the configuration described in Section 2. The linear classifier (i.e. the SLP) was not subjected to training during the drift simulation.



(a) Drift data (b) CRBM+SLP

(c) SLP (d) MLP

Fig. 4: (a) Real drift data with constant pH level and (b-d) the activity of output unit for each algorithm over the 7644 drift epochs.

Fig.4b depicts the SLP's response/activity to the drifting data. As only class A data was fed, the SLP's activity should always be at '+1'. With the proposed configuration, the CRBM managed to track the drift autonomously for at least first 5000 drift epochs (Fig.4b). However, as the drift was further increased, the CRBM was unable to compensate. At around 6500^{th} drift epoch, the abrupt drift step (caused by a malfunction in the reference electrode [8]) resulted the CRBM to fail completely as expected. Sets of 400 test samples for each class were also used to evaluate the autocalibration by the CRBM at different drift epochs. Classification accuracy degraded from 100% (at start of the simulation), to 96.28% (after 4000 epochs), and to 89.83% (after further 2000 epochs).

To highlight the importance of the adaptivity feature, two trained but subsequently non-adaptive neural classifiers, namely another SLP and a multilayer perceptrons (MLP) network, were used as benchmarks. Both classifiers were trained and fed with the same datasets as the combined CRBM and SLP. After 20000 training epochs, the SLP achieved a mean square error (MSE) of 1.378×10^{-1} and an accuracy of 95.20%, as this is a not a linearly-separable task. As predicted, the SLP lost track of the drift over time (Fig.4c). Its accuracy dropped to 86.53% and to 78.80% at 40000^{th} and 6000^{th} epochs respectively.

A 3-layer MLP was employed and had 7 neurons in each hidden layer. The MSE was 1.102×10^{-4} and the classification accuracy was 100% after 20000 training epochs. Fig.4d depicts the response of the MLP's output unit to the drifting data. A sharper fall in the response was observed and this could be explained by the use of higher-order (hence better-defined) hyperplanes by the MLP, as compared with the SLP. When tested with datasets at different drift epochs, the MLP's accuracy dropped to 85.66% and to 74.18% at 40000^{th} and 6000^{th} epochs respectively, again falling more drastically than the SLP.

5 Conclusion

We have investigated sensor modelling on non-Gaussian data distributions with a CRBM in two separate simulations. The reconstructions were good and test data classification 100% accurate. Additionally, we also applied the combined CRBM and SLP to real drifting pH data in a dynamic environment. Simulation result showed that the CRBM with the proposed configuration was able to track drift for the first 5000 drift epochs but subsequently lost track (due to the use of fixed learning rate) as the drift increased further. The drift simulation was rerun with two benchmarks (SLP and MLP). The results showed that the adaptive CRBM could out-perform them. This is primarily because the benchmarks were not in any way adaptive to drift. As seen above, the learning rate can play a crucial role in ensuring that an appropriate level of adaptation occurs. Thus, our future work will be implementating an adaptive learning rate in the CRBM.

References

[1] G.J. Pottie, W.J. Kaiser, Wireless integrated network sensors, *Communications of the ACM*, 43(5):51-58, 2000.

[2] H. Chen, A.F. Murray, A Continuous Restricted Boltzmann Machine with an Implementable Training Algorithm, *IEE Proceedings on Vision, Image and Signal Processing*, 150(3):153-158, 2003.

[3] G.E. Hinton, Training Products of Experts by Minimizing Contrastive Divergence. Technical Report, Gatsby Computational Neuroscience Unit, University College of London, UK, GCNU TR 2000-004, 2000.

[4] Y. Tan, S.L. Tantum, L.M. Collins, Cramer-Rao Lower Bound for Estimating Quadrupole Resonance Signals in Non-Gaussian Noise, *IEEE Signal Processing Letters*, 11(5):490-493, 2004.

[5] R. Kozick, B. Sadler, R. Blum, Array Processing in Non-Gaussian Noise with the EM Algorithm, *International Conference on Acoustics, Speech, and Signal Processing*, pages 1997-2000, Seattle, Washington, USA, 1998.

[6] T.B. Tang, H. Chen, A.F. Murray, Adaptive, integrated sensor processing to compensate for drift and uncertainty: a stochastic 'neural' approach, *IEE Proceedings on Nanobiotechnology*, 151(1):28-34, 2004.

[7] M. McCloskey, N.J. Cohen, Catastrophic interference in connectionist networks: The sequential learning problem, *The Psychology of Learning and Motivation*, 24:109-165, 1989.

[8] T.B. Tang, *et al.*, Toward a Miniature Wireless Integrated Multisensor Microsystem for Industrial and Biomedical Applications, *IEEE Sensors Journal: Special Issue on Integrated Multisensor Systems and Signal Processing*, 2(6):628-635, 2002.